IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 1, 2019, accepted November 16, 2019, date of publication November 20, 2019, date of current version December 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2954791

A Comprehensive Survey for Intelligent Spam Email Detection

ASIF KARIM, SAMI AZAM, BHARANIDHARAN SHANMUGAM, KRISHNAN KANNOORPATTI, AND MAMOUN ALAZAB
College of Engineering, IT and Environment, Charles Darwin University, Casuarina, NT 0810, Australia
Corresponding author: Asif Karim (asif.karim@cdu.edu.au)

ABSTRACT The tremendously growing problem of phishing e-mail, also known as spam including spear phishing or spam borne malware, has demanded a need for reliable intelligent anti-spam e-mail filters. This survey paper describes a focused literature survey of Artificial Intelligence (AI) and Machine Learning (ML) methods for intelligent spam email detection, which we believe can help in developing appropriate countermeasures. In this paper, we considered 4 parts in the email's structure that can be used for intelligent analysis: (A) Headers Provide Routing Information, contain mail transfer agents (MTA) that provide information like email and IP address of each sender and recipient of where the email originated and what stopovers, and final destination. (B) The SMTP Envelope, containing mail exchangers' identification, originating source and destination domains\users. (C) First part of SMTP Data, containing information like from, to, date, subject – appearing in most email clients (D) Second part of SMTP Data, containing email body including text content, and attachment. Based on the number the relevance of an emerging intelligent method, papers representing each method were identified, read, and summarized. Insightful findings, challenges and research problems are disclosed in this paper. This comprehensive survey paves the way for future research endeavors addressing theoretical and empirical aspects related to intelligent spam email detection.

INDEX TERMS Machine learning, phishing attack, spear phishing, spam detection, spam email, spam filtering.

I. INTRODUCTION

Email spamming refers to the act of distributing unsolicited messages, optionally sent in bulk, using email; whereas emails of the opposite nature are known as ham, or useful emails [1]. The word ''spam'' came into existence from ''Shoulder Pork HAM'', a canned precooked meat marketed in 1937, and eventually with the passage of time, digital mailing junks have taken the word [2].

Spam emails are propagated by the spammers for simple marketing purposes to unfold more malicious activities such as financial disruption and reputational damage, both in personal and institutional front. The practice of spamming is now spreading rapidly in other digital communication channels as well.

Financial motivation is one of the primary reasons for the spammers and it has been estimated that spammers earn around USD 3.5 million from spam every year [3].

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

A. RELEVANT SPAM EMAIL STATISTICS

In the following subsections, we will highlight some current worldwide statistical observations. Besides, some country-specific metrics will also be discussed.

The statistics relating to the adoption of email as a means for communication is quite staggering. As of 2017, there were nearly 5.5 billion email accounts which are actively in use [4], this number is projected to grow over 5.5 billion in 2019 [5]; nearly one third of the population are estimated to use email by the dawn of 2019 [5]. As of 2018 approximately 236 billion emails are exchanged daily [6], of which around 53.5% are just spams [4]. In fact, 2018 saw an average of 14.5 billion spam emails daily [3]. FBI recently reported a loss of USD 12.5 Billion to business email consumers in 2018 incurred by spam emails [7]. The financial loss incurred by the businesses due to this spamming attack may just skyrocket in few years' time, hitting an accumulated figure of around USD 257 Billion from 2012 by the mid of 2020 [3]. The estimated yearly damage will be around USD 20.5 Billion [3].

TABLE 1. Financial loss incurred in australian markets due to digital scams.

Year	Total Loss (AUD)	Losses in Top Three Categories (AUD)
2018	107,025,301	• Investment Scams – 38,846,635 • Dating & Romance – 24,648,024 • False Billing (Fake Invoices) – 5,512,502
2017	90,928,622	• Investment Scams – 31,327,476 • Dating & Romance – 20,530,578 • Other Business & Employment – 5,270,948
2016	83,561,599	• Dating & Romance – 25,480,351 • Investment Scams – 23,631,338 • Advanced Pay & Fee Fraud – 6,499,604

United States has traditionally been the largest source of spam, however, in recent times it is not the case anymore. Though there were legislations such as CAN-SPAM (Controlling the Assault of Non-Solicited Pornography and Marketing Act) to protect the users, it did not have the expected deterrent effect on the spammers [8]. USA houses world's top 70% spam gangs, responsible for coordinated worldwide spamming [3].

Scamwatch reports [9] portrays a grim figures in financial losses for Australian consumers due to verities of scam types, primarily carried out through phones and emails in the last three years as portrayed in Table 1.

As discussed in Table 1, the trend is heading upwards each year for digital theft and email spams will only rise due to the increasing adoption of this media as mentioned in the above mentioned statistics. Investment scams basically offers fraudulent but promising business opportunities in exchange of significant amount of money, while dating scams victimizes individuals looking for romantic partners in digital spaces. When comes to delivering malware to propagate such scams, emails are still the primary choice for the scammers. Recent reports indicate Australian businesses and consumers already lost nearly AUD 56,000 due to email fraud just within the first two and half months of 2019 [10].

As of April 2019, Brazil and Russia have conveniently overtaken USA and China (another substantial spam originating country), to produce approximately 16% and 14% of total volume of worldwide spam [11].

B. RESEARCH MOTIVATION

The motivation behind this research initiative is to address a gap that has risen over time in the field of spam email detection. The current solutions are mostly lagging behind the innovativeness the spammers are constantly bringing in, which heavily justifies the emergence of machine learning based anti-spam propositions. This review work critically evaluates number of such reasonably recent solutions and provides insights into ways upon which further improvement can be obtained. The paper also discusses a number of existing non-machine learning based frameworks to highlight the loopholes and the current state of affairs, this also signifies why machine learning automated procedures should be the approach of the newly developed systems.

There are some other good review papers available on the topic that discussed anti-spam frameworks in general, but as the field is expanding fast with lots of novel and automated ideas of spam email detection, we deemed it is necessary to orchestrate a comprehensive review paper that will analyze the state of the art developments as no other contemporary paper strictly focusses on current and recent trends and solutions geared specifically towards phishing spam email using machine learning algorithms. This paper also provides an exclusive detailed analytical insights based on the reviews. The insights clearly identify multiple gaps that can be addressed using machine learning principles.- showing the general future direction of research in this domain.

C. SCOPE OF SPAM EMAILS ANALYSED

Dissecting and critically analyzing scholarly research work on all types of spam email is itself a mammoth task and often impossible in a single survey attempt. Bearing that in mind, this paper primarily focusses on the intelligent and automated solutions devised against malicious spam emails. Particularly on the following:

1) Containing malicious links
2) Containing malicious attachment
3) Phishing attempts
4) Phishing and Spoofing campaigns

This survey work excludes studies that addresses 'Only' the marketing email spam.

D. RESEARCH METHODOLOGY

The papers for evaluation have been selected based on the objective of this research attempt. We have looked into several papers selected based on the listed index terms and thoroughly analyzed the presented method, whether it has effectively used machine learning principles; how robust and impactful the proposed solution really is; and finally the degree of modification required to address the drawback(s) the solution may exhibit. Only the works showing significant impact and intelligent automation have been selected as those were deemed promising for further research. The other two sections dealing with a small number of static and bio-inspired techniques have been mainly added to highlight the current state of email spam frameworks and diversity in research directions.

E. STRUCTURE OF THE PAPER

This survey paper has been structured in such a way so that the necessary background for the studies analysed are addressed first. Section II details out the parts of an email, and how the spammers take advantage of these various parts to craft verities of spam attacks on users, that is the types of email spam. Though this review paper intents to evaluate the machine learning based solutions aimed primarily for phishing and spoofing attacks, Section III will discuss number of general purpose non AI based spam detection systems and frameworks that do not rely on Machine Learning principles.

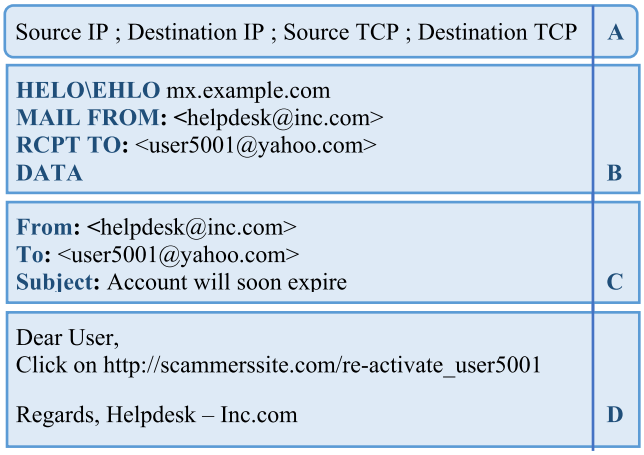

FIGURE 1. Email data parts.

It is important to have a look into such propositions to better understand where we stand at current times against spam emails and the necessity to bring in automated intelligence into the existing tools and emerging processes. The section after that (Section IV) is based on several Bio-inspired and Machine Learning based approaches for spam classification and detection. Section V will have a detailed discussion on insights that have been gained as the result of the critical review done in Section IV. Section VI clearly highlights the future direction of the research followed by the conclusion.

II. ANATOMY OF EMAIL AND TYPES OF SPAM ATTACK

The nuisance of spamming will inevitably find its way into almost all sorts of digital communication mediums in use at the present era. Among these, spamming through emails has always been one of the most exploited arena for the fraudsters. This section depicts a detailed structure of the email itself and the various attacking techniques adopted by the scammers. Email data parts are composed of several different blocks as illustrated in Fig 1 [24], while Table 2 summarizes the blocks.

A. DISCUSSION ON TYPES OF SPAM ATTACK

There are number of attacks that are being constantly bombarded on users worldwide, such as email spoofing [12], [13], phishing [14], [15], variants of phishing attack like spear phishing, clone phishing, whaling, covert redirect etc. Besides, spoofing and phishing attacks, variations such as clickjacking has also emerged within spam emails. Hackers have even gone the distance as to hide the text behind images to battle the anti-spam programs [16]. Chung-Man [17] demonstrated that in particular, phishing alerts may lead to a considerable negative return on stocks or market value of global firms. Others have indicated the destructive effects on companies whose email messages had been marked spams by the anti-spam systems, but were, in fact, not actually spams, an expensive instance of False Positive detection [18]. Below are some discussions on several common attacks.

TABLE 2. Email data parts explanation.

Data Parts	Identified As	Explanation
A	TCP/IP Header	The message payloads are wrapped inside TCP/IP packets, source and destination IPs are the most important information here
B	SMTP Envelop	Starts with containing mail exchangers' identification (may contain. MAIL FROM is the originating source and the recipient's address goes to RCPT TO segment. These two are actual source and destination email addresses. However, these are optional and can actually be forged. EHLO and DATA are two of the SMTP commands
C	SMTP Header	The 'From' and 'To' fields in this part are for viewing only (appears in email clients), scammers may fill this with fraud data to hide the actual route. By default these take the values from the Envelope
D	Body	A sub-part of SMTP Header. Email content goes in this part, including attachments

1) EMAIL PHISHING

Email Phishing is one of the most common ways of carrying out spam attacks on senders, and achieved through manipulating data part C (The 'From' field) or B. The aim is to fashion the message in such a way so that it appears to have been sent from someone or somewhere, often known to user, other than the actual source [13]. Spammers also tamper with the domain that is passed in the HELO statement, so that it seems the mail has originated from some known domain [7]. This indicates spoofing may occur in data part B (SMTP Envelop) as well. A Malaysian oil distribution company suffered a substantial financial loss over USD 1 Million in 2017 due to email spoofing [19].

2) SPEAR PHISHING

In practical terms, ''Spear Phishing' is a form of general email phishing family, in that it deceives with legitimate-looking messages [20]. A phishing scam may optionally provide a link to a bogus website where the end-user is required to enter sensitive financial and personal information [14], [15]. It operates on the body of the email, that is, data part D. Such type of spear phishing can also contain attachment which can carry malicious malwares [21]. It is also possible, basically using social engineering tricks, to craft the message in such a way, without malicious links or attachments, that the user will forced to take certain steps based on the content of the email, which will ultimately benefit the scammers.

The spam in case of Spear Phishing is crafted using personal information about the user, often gathered though social engineering methods [14], and sent from, what appears to be, a trusted source. These types of attacks are often harder to detect with traditional filters due to the sophisticated personalization of the look as well as the content. Spear phishing can be used to generate a form of serious attack known as ''Advanced Persistent Threats'' (APT), such as GhostNET

and Stuxnet [22]. Just in 2018, sensitive financial information of employees of the ABC Bus Company (USA) had been compromised due to phishing email scams [23].

3) WHALING

A variation of phishing attacks; in this form, the attack is often directed towards high level officials of a company. In the case of whaling or 'CEO Fraud attack', the impersonating web page/email will adopt a more serious executive-level form. As it works in the data part D as well, the content will be created targeting mostly an upper manager or a senior executive who has some high level clearance inside the organization and are almost always either an urgent executive issue - affecting the whole of the company, or a customer complaint. The emails will be sourced from a fake origin, disguising as a legitimate business establishment (same or other company) or even the CEO of the host company itself [24]. The risks and dangerous are similar to that of the other forms of phishing and spoofing.

Europe's principal electrical cable and wire manufacturer, Leone AG, lost a massive €40 million due to a sophisticated corporate email scam using a combination of spear phishing and whaling attacks [25]. In a recent Whaling attack in 2018, French cinema chain 'Pathé' lost over USD 21 Million [26].

III. NON-AI BASED CURRENT ANTI-SPAM SYSTEMS

Following are number of common anti-spam frameworks, most of which are available under different platforms such as standalone software programs or online based solutions. These do not adopt any AI based approaches.

A. SERVER AUTHORIZATION/AUTHENTICATION SCHEMES

Following are some of the notable Server Authorization/Authentication Schemes.

1) DOMAINKEYS IDENTIFIED MAIL (DKIM)

One of the most complicated frameworks that are in circulation these days [34]. The entire process is implemented through a public key encryption. However, due to the reasonably low adoption of this rather formidable framework by the ESPs [35], a certain email, with a nil DKIM field, cannot be marked as confirmed spam. DKIM is also susceptible to spoofing [36]. DKIM operates in both part C and D [Fig. 1].

'Public Key Encryption (PKE)' method is considered as one of the concrete encryption techniques designed till now. Generally two (2) keys are involved in the process [37]. 'Public Key', one of the two keys allocated to each party, and is published in an open directory in a place where anyone can easily search for it, for example by email addresses. Then there is a 'Private Key', a secret key maintain by each party. Several steps are involved before a successful encryption and decryption cycle is completed using PKE [37].

- Find P and Q, two large (e.g., 1024-bit) prime numbers.
- Choose E such that $E > 1$, $E < PQ$, and E and *(P-1)(Q-1)* are relatively prime, meaning they have no prime factors in common. E does not have to be prime, but it must be odd. *(P-1)(Q-1)* cannot be prime because it is an even number.
- Compute D such that $DE = 1\ (mod\ (P-1)(Q-1))$
- The encryption function is $C = (T\text{^}E)$ mod PQ, where C is the ciphertext (a positive integer), T is the plaintext (a positive integer). The message being encrypted, T, must be less than the modulus, PQ.
- The decryption function is $T = (C\text{^}D)\ mod\ PQ$, where C is the ciphertext (a positive integer), T is the plaintext (a positive integer).

The public key is the pair (PQ, E) while D is the private key. A major advantage of this cryptography is that one can publish ones public key freely, because there are no known easy methods of calculating D, P, or Q given only (PQ, E) - the public key. Besides, popular Email Service Providers (ESP) like Gmail now provides End-to-End encryption facility through S/MIME (Secure/Multipurpose Internet Mail Extensions), which itself is based on Public Key Encryption. However, such type of cryptography is often slower than other available methods.

2) SENDER POLICY FRAMEWORK (SPF)

Sender Policy Framework (SPF) now a days has become one of the critical email authentication mechanisms, often used along with DKIM. However, this technology itself is a standalone framework and is an email validation protocol architected to detect and block email spoofing by providing a system to allow receiving mail exchangers to authenticate that the incoming mail from a domain indeed has arrived from an IP address authorized by that domain's administrator [38]. SPF basically prevents the scammers to distribute emails on someone else's behalf.

The receiving server of the incoming message will look for the SPF record of the sender server along with the message. The SPF record will have a list of allowable IPs from which emails messages of that specific sender (or user) are allowed to originate. So, in case the list do not contain the IP address of the server that sent the message to the receiving server, the receiving server will not allow the message to pass through [38]. SPF works in both part A and B.

Even though not all mail servers implemented SPF as of now, but the adoption of the technology is rapidly gaining pace with time.

B. PROPOSITIONS BASED ON ARCHITECTURAL MODIFICATION

The simple and unassuming design of SMTP has long been held responsible for a range of spam attacks. Bandav *et al.* [39] state that hackers even spoof the 'Date' field in SMTP header to keep their spam emails on top of receiver's Inbox, so that immediate attention can be gained. The authors have also suggested that ESP's should employ a dedicated 'Time Stamping Server' to authenticate the sending date for every email. Number of other researchers have proposed alteration (for instance, modifying some of the SMTP transactional steps) in the blueprint of SMTP to make it more

secure and robust as the preferred choice for mail transfer protocol. However, such steps are not always practicable for multiple reasons as discussed later. This section will highlight few of such commendable research undertakings and a discussion on the hindering bottlenecks.

A pre-acceptance test of emails has been discussed by Esquivel *et al.* [40], which works by analysing the features of individual SMTP transactions such as EHLO/HELO message sequences. These can be further divided into different categories based on the working mechanism such as 'Protocol Defects'. Protocol Defect can detect any extra suspicious data blocks in the input buffer before the EHLO/HELO message transaction takes place.

The work done Bajaj *et al.* [41] suggest that filters in the spam blocking network servers should use the facility of detecting suspicious behaviour patterns of VoIP spam callers- which can be built into the signalling protocols used in VoIP, such as SIP (Session Initiation Protocol). SIP is an Application Layer protocol- heavily used to create, modify, and terminate a multimedia session (streaming videos, online games, instant messaging etc.) over the Internet Protocol [42]. SIP can also use Message Digest (MD5) authentication for security purposes. To detect suspicious behavior, SIP can inherently apply automated frameworks that can analyze the message to determine whether the message is syntactically wrong, have no apparent meaning, hard to interpret or may lead to a deadlock [43].

The above discussed studies are fine examples of impressive steps in the direction of fortifying the SMTP framework at the root level. On the other hand, the very popularity and adoption of SMTP at the first instance as the de-facto protocol of choice for email communication has established some strict deterrents. For instance, a slight modification to the protocol may introduce a wave of changes to other intertwined enabling services needed for successful mail delivery, both in regards to efficiency and usefulness [44]. Thus such structural modifications, required at the core infrastructure of email communication, will surely introduce operational complexities. This constrain of the SMTP has been an issue long since and hackers and spammers have exploited the drawbacks from a very early stage [44].

C. COLLABORATIVE MODELS

Under collaborative spam filtering modeling strategies, each message is delivered to a number of recipients. A specific message will most certainly be received and judged by another user. Collaborative models exhibit the process of capturing, recording, and querying these early judgments. Over time these collections become significant enough to stamp a verdict on a certain email. A number of techniques are in circulation to achieve various steps of a successful collaborative framework [46].

1) CRYPTOGRAPHIC HASHING

A highly successful method in earlier days of email communication, where large email vendors (Hotmail, Yahoo!, AOL etc.) mathematically calculated an alphanumeric strings of 32 to 128 characters, known as signature of the email (the Hash value), to store it in a database [47]. The vendors work on the idea that spammers will send out a burst of spam emails to achieve their target and some of these spam emails will reach to their honeypot accounts, that is, account that had been set up specifically to catch spam emails. These vendors also rely on the fact that the generated signature will be largely different for spam to that of non-spam emails [48]. Therefore, soon as the signature pattern matches to that of the spam pattern, it is added to a database for spam signature, and as other emails arrive at any of the other customer accounts, those are instantly discarded if found spam - by matching the signature (calculated using the exact same method, from header and body, thus the method works in both part C and D [Fig. 1]) to that of the one stored in the database, provided the record is found. Vendors supply the database to other Email Service Providers (ESP) and thus once an email is identified as spam, it is updated in several of these databases positioned throughout the globe.

Message Digest 5 (MD5) was one of the popular choices for cryptographic hashing. It is a cryptographic algorithm that accepts an input of any length and generates a message digest that is 128 bits long, often known as the ''fingerprint'' or ''hash'' of the input. MD5 is quite useful when a potentially long message needs to be processed and/or compared quickly.

The problem with such technique is that spammers have already succeeded, in a rather constant basis, in devising tools that can actually break the hashing algorithms. Further, the issue of database update is also a lingering bottleneck for quite sometimes now as it is being automatically updated but in a delayed nature, and that window is enough for a lot of fraudsters [50]. Furthermore, if that database itself is hacked, then it is curtains for the ESPs. Thus the technique has seen eroded accuracy over the years [51]. SHA-3 is a recent development in progress to replace MD5 and its close variations altogether.

2) FUZZY HASHING

As illustrated by Chen *et al.* [54], researchers have used Hashing principles (such as Fuzzy Hashing) to detect spam campaigns by clustering emails on the basis of similar goals.

Fuzzy Hashing can effectively be used to measure the resemblance of two sequences of characters by calculating scores based on similarity on the spam messages. Fuzzy hashing relies on both 'Traditional Hash' function (for instance, MD5) and a 'Rolling Hash' function. The Rolling Hash value of a string $M = m_0 ... m_{n-1}$ can be obtained from (1), where k is the modulo and $0 \leq a < k$ the base. Both k and b are usually prime number and do not have any prime factors in common.

$$h(M) = (\sum_{i=1}^{n-1} a^{n-1-i} m_i) mod\ k \tag{1}$$

In the work of Chen *et al.* [54], the Rolling hash simply divides the input into arbitrary sized pieces. These pieces are then hashed with the traditional hash function. The concatenation of the hash values obtained after hashing all the pieces forms the 'Fuzzy Hash Value' of the given content. Hash values are often considerably compact than the original string of characters, as they produce fixed length output, irrespective of the length of input [55]. In this way, contents that are not exactly identical, but slightly differ in some way, can still be grouped under the same hood.

The research [54] takes on the view that the emails from same campaign will have a higher similarity score among each other while the scores will be far apart among emails from different spam campaigns. It has also been shown that emails from same campaign have similar sort of URL or email address. A lacking of the work is that it does not address concerns regarding 'Asymmetric Distance Computation' [56], where the cluster distance score may become non-deterministic if the order of input changes [57].

Due to several drawbacks of MD5, in particular a slight change in input dramatically alters the corresponding hash value, which is not always desired, as often the message content of multiple spam email varies slightly, but those are still considered spam and often from same campaign, the application of a locally-sensitive hashing algorithm, known as 'Nilsimsa' [49] has grown considerably for hashing purposes, it generates a score from 0 (dissimilar objects) to 128 (very similar objects or identical). Nilsimsa uses a 5-byte fixed-size sliding window that analyses the input on a byte-by-byte before generating trigrams (group of three consecutive characters) of probable combinations of the input characters. The trigrams map into a 256-bit array to produce the desired hash [49]. Another supervised k-NN based close variation of Nilsimsa, known as TLSH (Trend Locality Sensitive Hashing) has garnered much attention these days [205]. It provides a similarity score between 0 and 1000+, where any score <=100 will identify the two entities being similar to each other, mostly originating from same source [205]. Projects such as SSDEEP, a Context Triggered Piecewise Hashing program, is also an important addition to this field [205].

3) DISTRIBUTED CHECKSUM CLEARINGHOUSE (DCC)

Distributed Checksum Clearinghouse (DCC), another hash sharing framework against spam emails, works by counting how many times a specific message has been reported as spam. It takes a checksum of the message body and stores it in a clearinghouse or server [52]. Thus with every additional reporting to the clearinghouse of a message being spam, the checksum count increases by 1. Bulk mail in this way can confidently be identified because the response number and checksum count are usually lot higher. The checksums are fuzzy in nature and oftentimes multiple DCC servers participate in the checksum exchange process [53]. It operates in both Part A, B and D [Fig. 1].

4) GREYLISTING

Greylisting takes on the view that a legitimate sender, will resend the email if the initial attempt is unsuccessful, while the spammers will just move on to the next sender and will not bother to check whether the email has been delivered. However, this approach can simply be bypassed by resending the spam email [41].

5) DNS BLACKLISTING AND WHITELISTING

DNS (Domain Name Server) Blacklisting is carried out in two different flavours. First one involves maintaining a list of mail-server IPs identified as spam originator or propagator in a centralized database [44], [48], [58]. The other way is to mark spam based on Uniform Resource Identifiers (URIs), usually domain names or websites; the blacklists then consist of such malicious URIs [59]. These blacklists or databases of known spamming IP or domains can then be given access by the administrator either for free or with a price. The email server using this service will execute an additional DNS query on the host that is sending the message to determine the source status; in this case the queried DNS server will be the one provided by the DNS Blacklisting service. However, all such blacklists suffer from the inability of early detection of malicious phishing URLs on the wake of the attack because their database update process is not fast enough [60].

The issues with Blacklists are that the spammers can frequently alter source address [58]. Also the source address itself can be spoofed as mentioned earlier. In case of blacklists composed of URIs, spammers continuously set up cheap new domains before starting a fresh cycle of mass spamming, leaving the blacklists very little time to react instantly. Additionally, the list is often quite slow to be updated and thus rather ineffectual against phishing email threats that banks on user-visits at short-lived phishing websites.

Whitelisting is the practice of maintaining a list of mail-servers that are only administered by confirmed legitimate administrators, or to accept content from bona fide users. Different organizations have such whitelists of their own to make things easier for the customers. Blacklisting and Whitelisting operates in both part A and B [Fig. 1]

When it comes to Whitelisting and Blacklisting of spamming sources, the Spamhaus Project, initiated in 1998, has become a workhorse in this arena [61]. A number of ISPs and email servers use the lists to reduce the amount of spam that reaches their users. Spamhaus also provides information on certain domains and main server for intentionally providing a *Spam Support Service* for Profit. The Project currently has over 600 million subscribers [61].

6) SOCIAL TRUST BASED SOLUTIONS

'Social Trust' is a layer that is capable of providing a measure of the system's belief that a host is distributing spam emails. Other nodes, that do not have spam identification processes installed, can actually receive the notion of these 'Trust' enabled nodes and can take appropriate action [62].

TABLE 3. Few Regex rules for spam filtering.

RegEx	Objective
Header{Subject} =~ m.{0,2}o.{0,2}r.{0,2}t.{0,2}g.{0,2}a.{0,2}g.{0,2}e not Header{From} =~ \S+	Filter out all messages having various variants of the word 'mortgage' in the subject line. Deleting messages where the sender's address is nil
Header{To} =~ \b<?([\w\-.]{2})[^@,]*@.*(?:\b<?\1[^@,]*@.*){3}	Blocking messages having several addresses in the 'To' field that start with the same characters
^207\.23\.100\.([5-9][0-9]|1([0-4][0-9]|50))$	Filtering out all emails sourced from IPs in the range from 207.23.100.50 to 207.23.100.150

Sirivianos *et al.* [62] envisioned a framework based on social trust rooted in 'Online Social Networks (OSN)'. This system, known as 'SocialFilter', aims at accumulating the experience of a number of spam detectors, which in a sense, according to the author, ''democratizes'' the mitigation of spam. It is a graph-based solution where the reports received are assessed for trustworthiness. This assessment is used to gauge a value which echoes the system's conviction that the reported host is actually spamming. The performance is enhanced by 1.5%-2% than few other comparative products.

Lin *et al.* [63] argued that an authentic sender maintains ties with a reasonably small social circle, that is, they have noted a legitimate sender often communicates with a small number of accounts multiple times, mostly the ones that are in user's social circle. However, spamming accounts tend to just communicate very few times (most cases just once) with an account but their actions are far wider, that is tend to spam thousands of accounts. Authors have employed 'Bloom Filters' to develop the statistics over some time. Bloom Filters, a space-efficient probabilistic data structure, were introduced by Burton Howard Bloom in 1970 [64]. This study assumes botnets are behind rapid spamming, but the framework proposed is unable to identify exact spamming bots.

D. HEURISTIC FILTERING MODELS

These are Rule-based static filtering systems that can be extremely efficient to downright inefficient (poor accuracy) depending upon how versatile the rules are and how frequently these are being updated [65], [66].

1) REGULAR EXPRESSION (REGEX) BASED FILTERING SYSTEMS

Rules are developed mostly using Regular Expressions as depicted in Table 3. Scores are assigned for each of the matched rules and the total value is calculated to check if it tops a pre-set threshold value, which indicates that the email is indeed spam [65], [66].

A regular expression (or regex) is a pattern that describes a certain portion of text for the purpose of mostly string matching [67]. For instance the pattern '\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6}\b' can be used to look for an email address in a set of texts. Programmers can also use this pattern to check the validity of an entered email address, regardless of programming languages. Regex is incredibly useful in finding out strings of almost any pattern.

Heuristic systems are fast and easy to install, but in case the scammers are able to get a hold of the ruleset, they can very easily craft messages to avoid the filtering system [68]. Regex based methods work on part A, C and D [Fig. 1].

E. CONTENT BASED APPROACHES

These systems primarily relies on the examination of the body or content of the email. Several well-known techniques are used for such spam filtration systems as discussed below.

1) REGULAR CONTENT FILTERING SYSTEMS

A common class of spam detectors for quite some time now has been the 'Content based Filtering' method and several of its variations. In these systems, a thorough analysis is done on the host message to find out patterns in message texts, these are then matched with predefined and confirmed spam patterns and a score is recorded. A decision of spam or ham is taken after comparing the cumulative score against a threshold value [69]. The typical example of content based filtering systems is the 'Rule Based Expert Systems'. Such type of classification can be applied when the classes in consideration are static, and their components can cater for feature-wise distinguishability [70].

Fig 2 shows the approach graphically where it can be seen that some keywords have been designated as markers for spam email content ('cheap' and 'mortgage' in this case). A certain 'Weight' has been assigned to these words depending upon, mostly, general rate or frequency of the word appearing in confirmed spam emails. These values are then summed up to derive the 'Cumulative Score', which is then compared against a 'Threshold Value' (1.0 in this case). Once this Cumulative Score overtakes the Threshold Value, the email is marked as confirmed spam.

Despite being highly impactful, the system suffers from 'Context Sensitivity', meaning the actual intended message and background of the discussion may not be taken into account. The method fails to take into consideration the context of the content, thus emails having discussions or educational message on negative entities, for example 'Viagra', may be flagged as spam.

2) CONTEXT SENSITIVE PROPOSALS

To address contextual issues found at content based filtering approach, Laorden *et al.* [71], in his approach of using of semantics in spam filtering by introducing a pre-processing step of 'Word Sense Disambiguation (WSD)' [71], argued that WSD is an important pre-processing steps which can increase the accuracy rapidly with majority of the techniques. WSD deals with solving the problem of determining the most appropriate 'sense' (meaning) of the word under

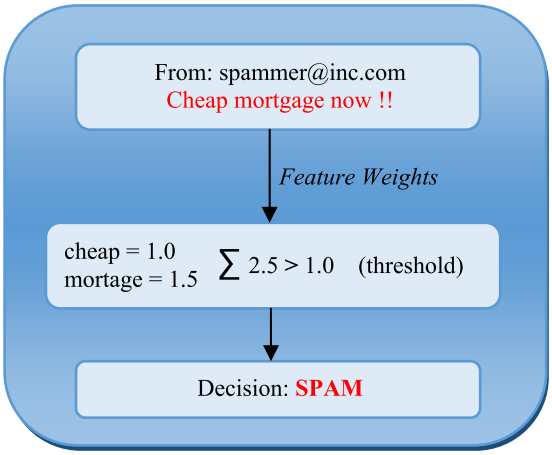

FIGURE 2. Content based filtering.

particular context. Laorden *et al.* [71] have, in fact, disambiguated the terms using 'Part of Speech' (POS) Tagging before constructing the 'Vector Space Model (VSM)'. However, the work does not address word collocation.

POS Tagging and VSM are two of the most used frameworks for both automated and non-automated email filtering system. POS Tagging refers to the process of classifying words into their respective parts of speech [72]. The two most common algorithms are 'Stochastic Tagging' (uses 'Probability' measure) and 'Rules Based Tagging' (use contextual information to assign tags to ambiguous or unknown words).

The Vector Space Model (VSM), is a popular algebraic model primarily employed for the representation of text documents and also incorporated in number of spam detection models. The model is built in several steps, which initiates with a weight being assigned to each term found in the collection of documents [73]. Oftentimes the weight is equal to the frequency of occurrence of the term t throughout the document d. This arrangement is called Term Frequency *(tf)* and denoted using $tf_{t,d}$. Now as all the terms are not equally significant, in a view to apply some form of scaling down into the weights, the inverse of another term, Document Frequency *(df)* is introduced. *df* is basically the total number of documents in the collection where the term t can be located [73], and thus denoted as df_t.

If the total number of documents in the concerned corpus is identified as N, the inverse document frequency (idf_t) of the term in question t can be obtained using (2) [73].

$$idf_t = log(\frac{N}{df_t}) \tag{2}$$

Finally, both Term Frequency and Inverse Document Frequency is combined to cement the composite document-wise weight for each of the terms using (3) [73].

$$tf - idf_{t,d} = tf_{t,d} x log(\frac{N}{df_t}) \tag{3}$$

VSM is often employed in building a vocabulary of most impactful words that may have an effective weight in differentiating the content type. One point to note is in case of long documents, the performance of VSM may often show the need for significant improvement.

3) FUZZY LOGIC BASED SYSTEMS

Fuzzy logic was introduced by Lotfi Zadeh as a mechanism to process imprecise data. A Fuzzy Controller itself is glued together by three linked segments [75], and consequently, Che *et al.* [74] have developed their novel algorithm on the back of Fuzzy Controller principle, having three distinct but interlinked segments [74]. The authors brought multiple angles into consideration as it used elements from social engineering practices, fuzzy control and semantic web to devise a novel algorithm to tackle phishing email.

The first part of the algorithm builds up the semantic web database which establishes the relationships between event and words (similar meaning words are grouped together). The events are specific keywords (from email content and subject, excluding prepositions) that insists the user to take some action (the aim of the phishing email). The second part is building the category database which is used to classify phishing emails. To achieve the target it first goes through an Even-Pair generation process, where, using the semantic database built into the earlier step, words are converted to related events; and two events are fused together to form a pair [74]. These pairs will then be inputted to a Fuzzy control function to determine the closest category. Finally, the last stage adds suggestions to users on categorizations of the new incoming emails based on logic that are derived primarily out of the above steps [74].

The framework puts highest emphasis on the content of the email rather than header or domain information.

F. SOURCE BASED FILTERING FRAMEWORKS

Identifying the validity of the source of email has been proven quite important to detect the class of the email in question. Following are few most common techniques.

1) IP BASED FILTERING

According to Hu *et al.* [76], 'Source based Filtering', especially using IP address, has also been popular and effective to a certain degree, as it is quite difficult for even the spammers to work around the IP address of the spam and thus if certain range of IP addresses can be identified as malicious, these emails can then be blocked from mass distribution. Further, IP addresses reveal geographic locations as well, and it is a well-known fact that countries from certain geographical boundaries are a mass source of spam, thus emails from those areas may be considered as spam with high degree of confidence, even though there might be issues as discussed earlier regarding such country based filtering.

Source based filtering has also been used to tackle Botnets. Spammers have tried to use Botnets to the maximum effect for automating high volume spam dispersion operation with speed. Wanrooij and Pras [77] and Stringhini et al. [78] pointed out the fact that Low Volume Spammers (LVS) are relatively harder to detect than High Volume Spammers

(HVS), due to usage of botnets. Examining the working mechanism of botnets, Wanrooij and Pras [77] have proposed an assumption, termed as 'Bad Neighbourhood', they also suggested filtering using core attributes such as IP addresses and any machine-readable hyperlinks in the email itself. The performance has shown superiority over traditional frameworks such as SpamAssassin, mainly because of lesser execution complexity and very low false positive rate [77]. However, the work needs to be tested longer. The authors have faced issues like URI (Uniform Resource Identifier, URIs and URLs are often interchangeably used) Blacklisting bottlenecks; such occurrences also require addressing.

2) DNS LOOKUP SYSTEMS

Even though such a process is not fully guaranteed to get the junk out, but oftentimes it can be a strong indication for further checking. It works by looking up if a record for the domain name, from which the email claims to have originated (the part after '@') does exist (the "A Record"). If it is not, then there is reason to doubt the validity of the email, as oftentimes such spamming domains are short-lived [79]. However, it suffers from the fact that the FROM field of the header can also be spoofed as discussed before. Thus the scammer can just simply put a closely related valid domain. The method is mostly applicable to part C [Fig. 1].

G. OTHER FRAMEWORKS

There are few other propositions available which are either not so commonly implemented or in an emerging state.

1) COUNTRY BASED FILTERING

Certain email servers often entirely block email streams from certain countries as certain geographical boundaries are often a mass source of spam [76]. This techniques may have high false positive rate of detection as even though spammers in certain countries probably are more active than others, but lots of benign and legitimate emails can circulate the World Wide Web as well from those nations. The method is mostly applicable to part A [Fig. 1].

2) PEER TO PERR INFRASTRUCTURE

Bradbury [80] shed lights on a different approach known as 'Bitmessaging', based on the similar proof-of-work concept [81] that is being used in Bitcoin transactions. The framework relies on the BitMessage peer-to-peer communication protocol, and uses completely decentralized and encrypted network. On the contrary, one usability drawback is the nature of BitMessage addresses, which are rather complicated and unintuitive long alphanumeric strings that are difficult for a normal user to deal with. The system also has some scalability concerns as it is not yet fully compatible with the existing email infrastructure. The write-up also discusses a fundamentally different email system known as Dark Mail [80].

H. COMBINING EXISTING TECHNIQUES

There are open-source and commercial products available for spam detection that decide whether an email is spam based on the filtration results obtained via multiple filters. SpamAssassin and Zerospam are two of such most used products.

1) SPAMASSASIN

SpamAssassin is a free and open-source anti-spam product that has garnered several positive reviews over the years for its effectiveness and simplicity of installation. The product uses a number of above-discussed techniques for filtration purposes, such as DNS-based blacklists and DNS-based whitelists, Heuristic based checks, Fuzzy-checksum-based spam detection, SPF, DKIM and Bayesian filtering [82].

2) ZEROSPAM

Zerospam is a widely used commercial software that has also gained some grounds in effective spam detection [83]. It also uses a number of existing techniques such as IP address and Domain Check, Attachment and URL Scanning, Heuristic based filtering and Bayesian filtering [84].

The performance for these software has demonstrated regular fluctuations. A common problem with a number of commercial and open-source solutions are the lack of detectability in case of some form of phishing and word obfuscation. Besides, difficulty in implementation and usability complications are oftentimes observed.

That being said, the targeted scope of this paper prohibits a detailed discussion on several other commercial and open-source software available that work at different capacities by combining several existing techniques discussed above.

Fig. 3 illustrates an overall interconnection between email data parts and different spam detection techniques discussed till now, while Table 4 tabulates a summarized view of majority of the above-discussed research works and the reported results from these studies. The table also highlights the key points and shortcomings of some of the established spam detection techniques. The overall table links these different techniques and research initiative to the anatomy of spam [Fig. 1] to better underscore where in the very structure of an email these frameworks may belong. The colored lines in Fig. 3 links the respective method to the corresponding email data part, whereas the dotted lines signify under which category the respective techniques lie.

IV. ARTIFICIAL INTELLIGENCE BASED DETECTION AND CLASSIFICATION TECHNIQUES

The endeavour for successful detection and classification of spam emails have been going on for quite sometimes now. Over time number of successful methods had been devised, but with time many of these could not face the witty changes the spammers bring into their crafts on a rather continuous basis.

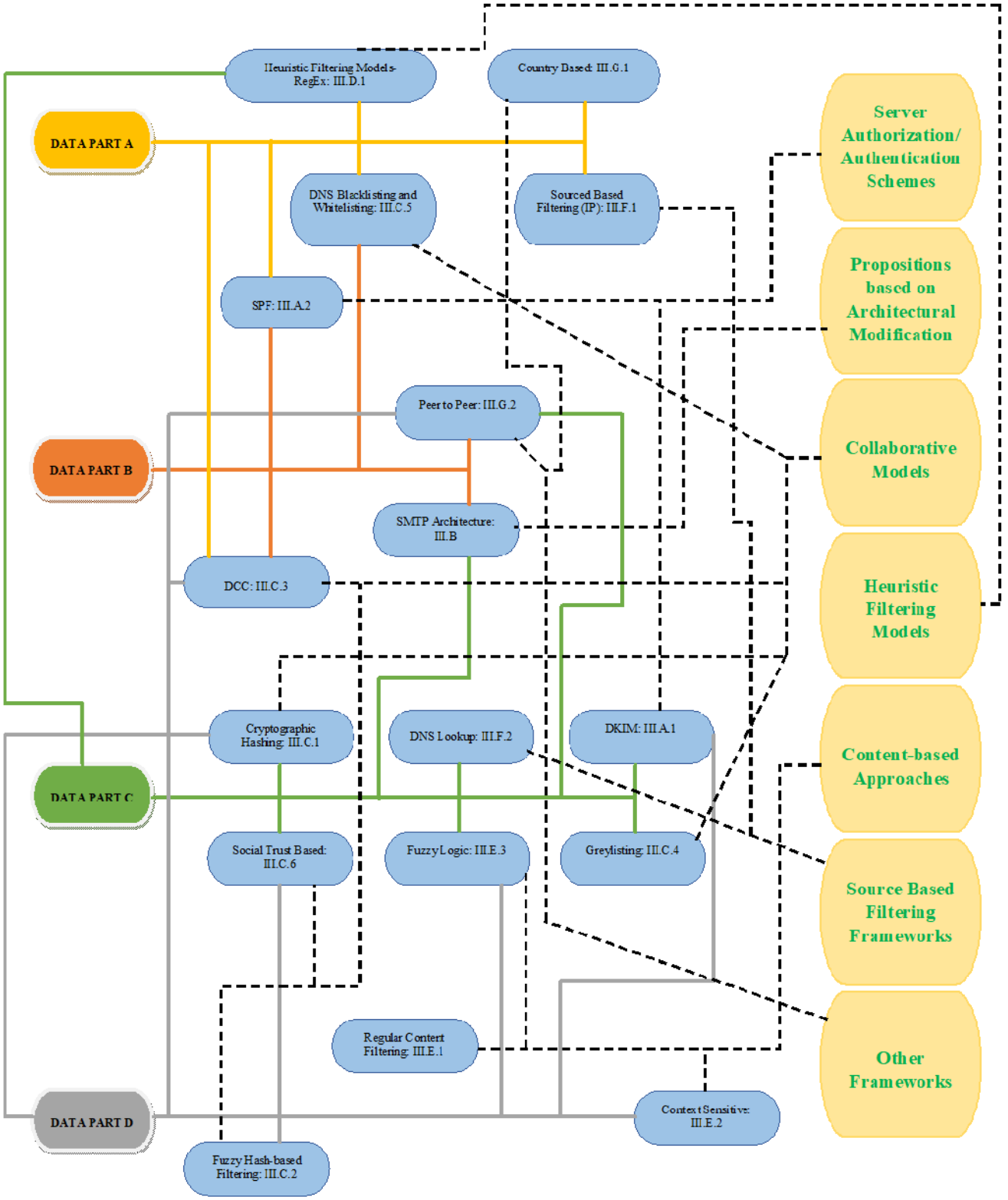

FIGURE 3. The interconnection between email data parts and different spam detection techniques.

A. SYSTEMS BASED ON BIO-INSPIRED INTELLIGENCE

These are computational algorithms motivated by inherent behavious and mechanisms often observed within the various natural living beings [85]. This is an emerging field of study, and consequently number of different algorithms are coming up with time. The following section illustrates some of the

TABLE 4. A summary of some of the above-discussed spam detection techniques.

	Objective	Email Data Part [Fig. 1]	Approach	Results Reported	Additional Comments\Shortcomings
Laorden *et al.* [71]	Addressing the Contextual Sensitivity	D	Introducing Word Sense Disambiguation (WSD) as a pre-processing step	An increase of Precision by 2% - 6% with certain Machine Learning algorithms on Ling Spam and TREC dataset	Does not address word collocation, application of POS labelling often degrades performance
Khanna *et al.* [69]	Analysis of inbound and outbound email data to understand spam pattern	D	Content Filtering techniques, with focus on the Statistical Filter	The deployment of Statistical Content Filters resulted in the detection of 6,696 spam emails, 74% of total spam emails detected. The remaining 26% were detected using other types of Content Filters	Apart from Content Filtering, the other methods have only been scantly discussed
Chen *et al.* [54]	Identifying Spam Campaigns using Fuzzy Hashing	D	Application of Fuzzy Hashing algorithm to cluster a dataset of 550K spam emails	The top Spam Campaign successfully identified had 8,630 spam emails over a period of 467 days	Does not provide any measure against 'Asymmetric Distance Computation'
Sirivianos *et al.* [62]	Building an Online Social Network based trust-aware collaborative spam mitigation framework	C, D	Using Sybil-resilient trust inference and social links among the connecting nodes to assess trustworthiness of a spam report	Improves the performance by 1.5%-2% than that of the comparative social trust based products in detecting spam emails	The comparing product is able to detect more spam, although with Higher False Positive
Wanrooij and Pras [77]	Detection of spam without analysing the content of the email	A	Analyse originating IP addresses and web links embedded into the email body, and then compare these to multiple real-time blacklists	The False Positive rate was just 1 against SpamAssassin's 36	Concerns regarding URI Blacklisting
Lin *et al.* [63]	Develop a system to track down spamming Botnets	C	The statistical core of the method has been orchestrated using Bloom Filter	The framework attained a Precision and Recall of 97% and 96% respectively	Not fully able to identify the exact spamming bots
Mori *et al.* [40]	Develop a system to analyze EHLO/HELO message sequences	B	Analyses the features of individual SMTP transactions such as EHLO/HELO message sequences. These can be further divided into different categories based on the working mechanism such as 'Protocol Defects'	Most notably, the resulting framework can detect any extra suspicious data blocks in the input buffer before the EHLO/HELO message transaction takes place	Difficult to implement any significant modification in SMTP infrastructure
Bradbury [80]	Highlighting the problems of existing email system and discusses few innovative messaging frameworks	B, C, D	BitMessage and Dark Mail	If implemented successfully and widely adopted, the email systems and routing networks can be largely secure with a significant reduction in Phishing email	Not yet fully compliant with the existing email infrastructure
Hashing Cryptographic [Section III.C.1]	Store the signature of spam emails and do required comparison while new emails arrive	C, D	Use a locally-sensitive hashing algorithm to mathematically calculate the 32 to 128 bit hash (signature) of the email and store it. Calculate the hash for newly arrived email and do a comparison with the stored ones, if found spam, also store the new has	Low False Positive rate and highly effective especially against Spam Campaigns	Oftentimes the hash can easily be broken. Besides, Hash Database update is rather slow and obtained accuracy can be quite low

TABLE 4. *(Continued.)* **A summary of some of the above-discussed spam detection techniques.**

Heuristics-based Filtering Systems [Section III.D.1]	Filter spam based on various static rules	A, C, D	Develop different regex rules to from various properties and constraints belonging to email header and body. Apply those rules to determine the status of an incoming email	Fast efficient processing and easy to implement	An insecure ruleset can easily render the system useless and rules need to be frequently updated as scammers change patterns
DNS Blacklisting [Section III.C.5]	Maintain a list in a database of confirmed spam originating IPs and Domains	A, B	The database is shared among different parties and legitimate mail servers query the DNS Blacklists before delivering an incoming email to the user	Requires minimum tools to get up and running. Besides, management of the system is quite easy and flexible	Enlisting the malicious IP and Domain can often take some time. Also requires judicious judgement to maintain an acceptable level of operation
DNS Whitelisting [Section III.C.5]	Maintain a list in a database of confirmed ham originating IPs	A, B	The idea is to maintain a list of mail-servers that are only administered by confirmed legitimate administrators, or to accept content from bonafide users	More secure than a DNS Blacklist. Also less False Positive than DNS Blacklists. Another plus point is that it is easily customizable	Requires more time than a DNS Blacklist to manage. In addition, installation can be complicated based on the environment
DNS Lookup System [Section III.A.6.2]	The validity of the sending domain is verified beforehand	C	It works by looking up if a record for the domain name, from which the email claims to have originated (the part after '@') does exist (the "A Record"). If it is not, then there is reason to doubt the validity of the email	Can be an useful indication for further verification	Can be easily bypassed by the scammers
Greylisting [Section III.C.4]	Wait for an incoming email to arrive multirole times to confirm its legitimacy	C	Greylisting takes on the view that a legitimate sender, will resend the email if the initial attempt is unsuccessful, while the spammers will just move on to the next sender	Aids other spam detection programs to increase accuracy of detection. Additionally, works with minimum of tweaking once installed. Come with easy management as there is no blacklists or ruleset to be frequently updated	The delay cause to deliver the message sometimes can be a bottleneck for urgent messages. Can be resource intensive depending upon the environment
Country Based Filtering [Section III.G.1]	Block emails from certain geographical locations based on previous spam origination record	A	Certain email servers often entirely block email streams from certain countries as certain geographical boundaries are often a mass source of spam	High number of spam emails can be detected in a short time	High False Positive rate
DKIM [Section III.A.1]	Public Key Encryption and Decryption dependent system	C, D	Digital Signatures based on Public Key Cryptography is used to validate the email	Increases the trust level of the email. Besides, a properly formulated DKIM signature confirms the legitimacy of the originating domain that the message the claims to be from	Not such a high adoption rate as of now. DKIM verified message still can easily be spoofed in a fashion known as 'Replay Attack". One other side issue is Implementing DKIM may not always be straightforward

TABLE 4. *(Continued.)* **A summary of some of the above-discussed spam detection techniques.**

SPF [Section III.A.2]	An email validation protocol	A, B	The SPF record of a domain prevents the scammers to distribute emails on someone else's behalf	SPF record for a domain significantly reduces the chance of being flagged an email as spam and curtails attempts by spammers to spoof that domain	SPF is an emerging technology and will still take some time to become ubiquitous
DCC [Section III.C.3]	A hash sharing spam email detecting framework	A, B, D	It works by counting how many times a specific message has been reported as spam. It takes a checksum of the message body and stores it in a clearinghouse or server and the count is increased, maintained and shared among other DCC servers	Can confidently determine whether an email is sent in bulk as spam	Setting up the usage of public DCC servers can sometimes be problematic due to either incorrect firewall settings or if the mail server processes in excess of 100,000 messages per day [52].

spam detection systems based upon such evolutionary and biology based computational algorithms.

1) GENETIC ALGORITHM BASED SYSTEMS

Ruano-Ordás *et al.* [86] argued that application of automatically generated regular expressions (regex) can be one significantly strong method in identifying messages that have been obfuscated by the spammers. The work compactly illustrate groups of sentences from compromised emails that follow a suspicious pattern. Thus the idea can be deployed as a local content based filtering system. It can also be shared in a P2P network for a collaborative approach in combating spams. The paper takes on the view that Bio-inspired Evolutionary Algorithms, such as Genetic Programming, should be used to generate the regular expressions. Genetic programming is based upon a subset of Evolutionary Algorithms, known as Genetic Algorithm. It is a search heuristic that is based upon the 'Theory of Natural Evolution' [87]. The work also presented a reasonably effective software, developed taking the drawbacks and limitations of some other contemporary similar systems into consideration; the system has been termed as 'DiscoverRegex'. A key improvement over the research of Conrad [88], claimed to have been achieved by the work, is the 'Fitness Function', an essential segment of any Genetic Algorithm based solution. Thus the proposed DiscoverRegex uses (4) for the Fitness Function.

$$fitness(i) = matches(i, spam)X(\frac{10}{length(i) + 1} + 1) \quad (4)$$

matches (i, spam) denotes for the number of spam messages that match regular expression *i* and *length (i)* represents the size of the generated pattern.

The results shows improvement over other software packages. However, the work only described generating regular expressions from the content of the spam subject header but not the body of the spam. Thus this enhancement needs to be incorporated to make the system fully complete.

2) NSA AND PSO BASED SYSTEMS

Idris *et al.* [89] and Idris *et al.* [90] discussed proposition where other bio-inspired algorithms, such as 'Negative Selection Algorithm (NSA)' [91], improved and reinforced with the addition of 'Particle Swarm Optimization (PSO)' and 'Differential Evolution (DE)' have been put into action. NSA is inspired by the self-nonself discrimination behaviour commonly observed in the mammalian acquired immune system [92]. PSO has been developed based on the social foraging behaviour observed in some animals such as schooling behaviour of fish and flocking behaviour of birds [93] and Differential Evolution is a metaheuristic that attempts to gradually optimize a given problem by multiple iterative passes over a candidate solution with regard to a given measure of quality. It can work with very high dimensional dataset, without always guaranteeing an optimum solution [93]. The combined approach discussed in this work shows increased performance than a standalone NSA based system. Idris *et al.* [89] achieved an increase of accuracy around 7%-9% than that of standalone NSA, especially over 1000 detectors.

However, both of these studies [89], [90] does not seem to address the behaviour of the proposed model in regards to the gaps in the understanding of few issues with the Particle Swarm Optimization algorithm, such as getting trapped in local minima and Heterogeneity [94].

This problem regarding local minima may also crop up in number of traditional nonlinear optimization algorithms. In this case the function (typically a 'cost function' in Machine Learning) produces a greater value at every other point in a neighbourhood around that local minimum than the local minimum itself. On the contrary, the global minimum of a function results in the minimization of the function on its entire domain, and not just on a neighbourhood of the minimum [95]. The ideal result of the function should be the global minima, or at least quite close to it. There are always one global minima but there can be multiple local minima.

3) OTHER RELATED SYSTEMS

An impressive numbers of other experimentation done with biologically inspired algorithms indeed delivered some more interesting outcomes besides the above discussed ones when comes to spam detection. Zhu and Tan [96] proposed a

'Biological Immune System (BIS)' based model where 'Local Concentration (LC) based Feature Extraction' approach has been adopted for the development of the anti-spam model. Such LC approach is thought to be able to effectively determine position-correlated information from a message by transmuting each area of a message to a corresponding LC feature. The proposition tends to divide the message content into the size of a fixed length window that goes through (slides) each chunk of the divided content. However, if the length of message content itself is shorter than the length of this sliding window, then the performance degrades. The study reported an accuracy and precision of over 96% with the size of sliding window set to 150 characters per window.

B. MACHINE LEARNING BASED SYSTEMS

Machine Learning (ML) is the engineering steps formulated in a view to make the computational instruments to act without being explicitly programmed. Machine Learning can be a great boon to tackle the spam issue primarily because of its ability to evolve and tune itself with time, and counter a key bottleneck ingrained in other classes of spam detection mechanism – 'Concept Drift'. Researchers pointed out that the contents and operating mechanism of spam emails change over time so the techniques that work now, may render useless in near future due to the change in structure and content of these spam emails; this phenomena is called Concept Drift [97], [98]. Wang *et al.* [99] conducted a statistical analysis of spam emails over a period of 15 years (1998 – 2013) and demonstrated how spammers adopt changes in not only spam contents, but also in the delivery mechanism.

The following sections will discuss a number of such technique and the results obtained once tried and tested on different spam corpora. However, before that, some Machine Learning based terminologies must be briefly discussed such as types of algorithms and associated benefits as majority of the models utilize these algorithms or its close variations.

Supervised Machine Learning Algorithms: Systems utilizing Supervised Machine Learning algorithms tends to learn from a set of labelled data, where the possible output for the corresponding input is already given [100]. The algorithm tends to go over this set of data (learns) and eventually builds up the 'Idea' or probabilistic mapping between the nature of input and most likely output (the result). Supervised Learning can be branched out into two different subtypes, Classification and Regression [100]. Supervised algorithms that, in most cases, produces outputs of categorical nature, are said to be classification algorithm, for instance: Spam or Ham, whereas, supervised algorithm that predicts outputs of continuous numerical value, are denoted as regression algorithm, for example: \$1000-\$5000, 50°F etc.

Unsupervised Machine Learning Algorithms: As the name suggests, unsupervised learning refers to the fact that the model will not have any labelled data to work with, and thus no training will be provided. Based on the dataset, the algorithms will try to figure out common features within a group of items and will rearrange the data points in clusters based on the commonality [101]. Alongside clustering, another type of unsupervised learning is 'Association Rule Learning (ASL)'; it finds pattern in large datasets based on some measure of interesting properties. For example, to deduce an activity pattern of an individual. Equation (5) can be deployed as following, where P and Q belongs to a set of items R [102]. ASL has also been used in recent times as an aid to develop supervised classification models [103].

$$\begin{aligned} &\mathrm{P} \Rightarrow \mathrm{Q}, \quad \text{where } \mathrm{P}, \mathrm{Q} \subseteq R \\ &\{\text{day} \in (\textit{weekends, public\_holidays}), \text{weather} \in (\textit{sunny})\} \\ &\qquad \Rightarrow \{\textit{fishing}\} \end{aligned} \tag{5}$$

The above example can be stated in general terms as, when its *weekend* or *public holiday* and the weather is *sunny*, the individual spends time on *fishing*.

Semi-supervised Machine Learning Algorithms: It is an amalgamation of both supervised and unsupervised learning. Oftentimes it has been seen that in a collection of large amount of input data, only a limited volume is actually labelled; semi-supervised learning algorithms work well in such scenarios [104].

Reinforcement Learning: In a Reinforcement learning system, the agent is capable of learning the pathways on the fly using a temporal learning scheme, without supervision. Agent is the entity that decides what action (A_t) to take. The system works on the basis of trial and error, where depending upon the action of the agent, a positive or negative feedback (R_{t+1}) is provided at the next instance [105].

1) PERFORMANCE MEASUREMENT

Most often the performance of Machine Learning models are calculated using various measures such as Accuracy, Precision, Recall, F-Measure, Receiver Operator Characteristic (ROC) Plot and Receiver Operator Characteristic (ROC) Area to name a few.

These measurements are mostly determined using *True Positive (TP)* – when the model correctly predicts the class, for instance classifying a spam as 'spam', *True Negative (TN)* – when the model correctly predicts the opposite class, for instance classifying a ham as 'not spam'. *False Positive (FP)* – when the model incorrectly predicts the class, for instance classifying a ham as 'spam', *False Negative (FN)* – when the model incorrectly predicts the opposite class, for instance classifying a spam as 'ham'. Table 5 describes the key terminologies of performance measurement of a model.

2) FEATURE SELECTION AND ENGINEERING

In any Machine Learning based model, 'Feature Selection [106], [107] and Feature Engineering' is a really crucial task as it is used to derive new and novel features from the existing ones to better facilitate the subsequent learning and generalization steps if a Machine Learning based algorithm is deployed to build a model [108]. The performance of the built model can often drastically improve if an intelligent

TABLE 5. Definitions Of some performance measuring terms.

Definition	Formula
Accuracy: The ratio of True Positives and True Negative combined against the total number of instances analysed.	$\frac{tp+tn}{tp+tn+fp+fn}$
Precision: It determines the percentage of identifications that are actually correct out of the total amount of cases claimed as correct.	$\frac{tp}{tp+fp}$
Recall: It determines the percentage of identifications that are actually correct out of the total amount of cases that should have been identified as correct.	$\frac{tp}{tp+fn}$

and intuitive Feature Selection and Engineering phase can be executed beforehand.

Scores of Machine Learning based systems [109]–[113] carry out from many of email headers parameters and content while designing the model to draw inference based on feature data that are not from expected spectrum.

3) HIGHLIGHTING SOME KEY SUPERVISED AND UNSUPERVISED ALGORITHMS

There are a number of algorithms in use in each of the categories briefly discussed above. Table 6 highlights few strengths and weaknesses of some of the primary algorithms that will be discussed in this study.

4) SUPERVISED LEARNING BASED PROPOSITIONS

This section will dissect a number of Machine Learning based research attempts which are primarily supervised by nature. These include one or more supervised algorithms in a view to develop an automated spam detection framework.

a: ARTIFICIAL NEURAL NETWORK BASED FRAMEWORKS

Artificial Neural Networks (henceforth ANN) are built using artificial neurons, modelled after neurons of biological brains. Depending upon the system, the total number of artificial neurons could be from few dozens to many thousands. These are connected in a series of layers, and divided into Input, Hidden and Output Layer.

The connection between the neurons, or often called units, is represented by a number called 'Weight'. Weights can both be positive and negative, meaning either they excite or suppresses another neuron. Normally information passes from Input Layer, through Hidden Later(s) to Output Layer, it is called a 'Feedforward' arrangement [114]. ANNs 'learn', most commonly, through a process called 'Backpropagation'. In this model, the produced output of the network is compared or matched with the one that should instead have been produced. The difference is then taken to adjust the weights between the connections, in this case starting from the output layer, to hidden layer(s) up until input layer, hence the term 'Backpropagation'. Over many iterations, eventually the network is able to produce a sufficiently accurate and acceptable result [114].

As can be seen from Fig. 4, an artificial neuron summing up all the weights (w_1-w_n) from inputs (x_1-x_n) before

TABLE 6. A summary of some useful machine learning algorithms.

Algorithm	Strengths	Weakness
	Supervised	
Artificial Neural Networks	Performs better than other algorithms, especially with very large datasets	Theoretical Foundation is weaker than other algorithms
	Efficient Feature Engineering	Computationally expensive and performance is not always up to the scratch for problems having smaller dataset
Naïve Bayes	Works well with missing or noisy data	The assumption that all feature are independent of each other
	Can operate with both continuous and discrete data	Inefficiencies in handling imbalanced datasets
	Highly scalable, fast and easy to implement	
Decision Trees	Virtually no hyper-parameters to be tuned for optimization	The prediction model can become inconsistent with minor variance in data
	Significantly easier to explain and visualize	Noisy training dataset can bring about significant overfitting
	Can handle both categorical and numerical data; besides performs appreciably with large datasets	Results can be interpreted in many forms or shapes, often leading to a difficulty in choosing the best one
Adaboost	Easy to implement for multi-class classification	Computationally expensive
	Handles overfitting rather efficiently	Increases the complexity of the classification rather substantially
Logistic Regression	Can be conveniently and intuitively interpreted	Not suitable for non-linear problems
	Able to function well even if the input features are not scaled	Cannot be used for predicting continuous outcome
	Can efficiently work with high dimensional by producing dependable results	Requires the data points to be independent of each other, which may not be the case for every problem at hand
Support Vector Machine (SVM)	For imbalanced dataset, SVM is a good option	Tuning the hyper-parameters can be quite complex
	SVM generalizes rather well as minor changes in data points do not affect the model much	Training time can be significantly higher especially on large datasets
	Resistant to overfitting	Not optimally designed for Multi-class classification

TABLE 6. *(Continued.)* **A summary of some useful machine learning algorithms.**

Unsupervised		
K-Nearest Neighbour	Robust and effective against noisy dataset	Determining the optimum value of K often becomes cumbersome
	Can be used for both classification and regression problems	Computation cost is comparatively higher that some other algorithms
	Handles multi-class problems effectively	Extreme sensitivity to outliers in the data points
K-Means Clustering	Straightforward implementation	High sensitivity to scaling though standardization or normalization
	Implementation is easier and has a low computational cost	Requires multiple runs to get a reasonably optimum value for K
Expectation Maximization	Can efficiently handle missing values	The convergence time can be lengthy
	Does not assume clusters to be of any specific geometry; unlike K-Means	High computational cost is a barrier to its widespread deployment

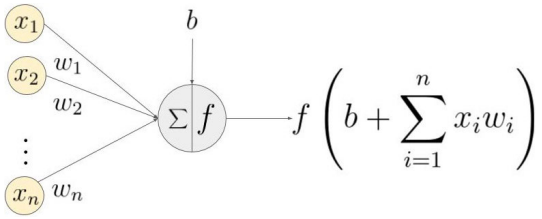

FIGURE 4. An artificial neuron with its activation function [30].

an 'Activation Function' f, together with a bias b, decides whether the neuron should fire (process and pass the information through); and if it does, then what will be its strength. An Activation Function also normalizes the output of a neuron especially after several runs.

Nosseir *et al.* [115] developed ANN based classifier to identify unacceptable and acceptable words from the message content of an email. The multi-neural network classifiers deal with words from the email body after the words have been pre-processed to remove the stop words (articles, prepositions) and noises (obfuscated words such as I∗n$u∗rènce or misspellings) along with the application of stemming process to extract the word root using Porter's Algorithm [116]. One of the major concern for the system is that it was tested on a small scale database of words and should further be tested on a larger setting before using a blacklist and whitelist content filter. The derived accuracy from the measurements provided has been found to be 99.87% for a five-character ANN.

Malge and Chaware [117] deployed tokenization, stop words removal and stemming before feeding the result in feature extraction algorithm. The obtained set of words is

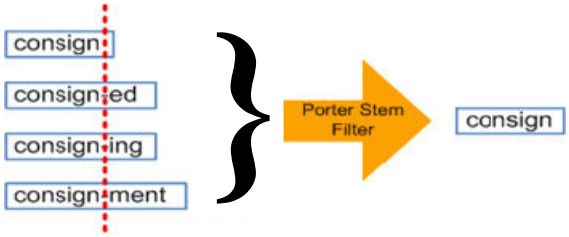

FIGURE 5. Porter's Stemmer.

evaluated based on the statistics past occurrences to mark an email spam or ham. The proposed framework achieves a recall measure of nearly 95%. To prepare the training set, words are collected from a collection of junk emails and are separated as good or bad as well as the length is also taken into consideration. For example, the three neural networks consecutively works on a set of words of length three, four and five characters. Words are further put into three different groups, namely marketing purposes, commercial, financial, and pornography; a weight is assigned for each of the groups depending upon the importance to the end user. The result shows low true negative and high false positive percentages. An advantage of this technique is that the user has flexibility in deciding the kinds of email more intrusive to him or her than some other categories, and thus set the weight accordingly. This technique is reasonably effective against word obfuscation as well as simple phishing attempts. Having said that, the system is limited to 'Bag of Words' approaches [118] and thus easy for the spammers to adopt other evasive workarounds.

There are various ways to carry out stemming [119], of which Porter's algorithm [116] found most success. A stemming algorithm retrieves the stem of a word. Fig. 5 illustrates the stemming logic rather intuitively.

b: DEEP LEARNING BASED FRAMEWORKS

Deep Learning is a subclass of Machine Learning that can learn in supervised and in unsupervised fashion. It employs cascading layer of processing units (nonlinear) for feature extraction and transformation. The output of each of these layers is fed into the next consecutive layer as input and these layers (often processing different levels of abstraction) construct a hierarchy of concepts [28], [120]. Algorithms such as DeepSVM [79], Convolutional Neural Networks [121] (henceforth CNN), Deep Neural Network, Deep Boltzmann Machine are few of the developments that are based upon the principle of Deep Learning.

Seth and Biswas [122] introduced Deep Learning techniques, such as CNN to tackle spam emails based on images and spam content. To classify e-mails containing both image and text, the authors have proposed two multi-modal architectures. Each of these architectures combines both image and text classifiers, producing an output class. The first architecture works on the basis of 'Feature Fusion' while the other mines the rules between the two classifiers as well as uses class probabilities. It has been reported that the later

projected higher accuracy of 98.11%; but the dataset is really a small one of just over 1500 images, whereas CNNs require hugely enriched and larger datasets to produce results that can be generalized over multiple different instances. Moreover, the Dropout rate, a regularization technique discussed at length in [123], has been fixed at 0.5, which may not be the optimal value for every situation, as stated in the work of [124]. The study needed to project the effects of other dropout rates before settling into 0.5.

Shang and Zhang [125] also deployed CNNs for image classification in spam emails. However, CNNs sometimes do not perform well on real world images partly due to the noises of different sorts that distort the image, but the paper did not discuss on such issues.

Barushka and Hajek [126], have demonstrated that reduction of features often reduces accuracy and precision as well as recall, although ANN and Decision Tree showed inspiring performance with a significantly reduced feature set. The research work [126] also argues that 'Shallow Neural Networks' are a poor fit for handling high dimensional data yet computationally expensive, unless some other advanced techniques, such as, 'Dropout Regularization' [127] and 'Rectified Liner Unit (ReLU)' - a popular Activation Function, are combined. Such methods can address some critical spam filtering limitations such as optimization convergence to non-optimal local minimum, an example of a problem arising out of overfitting and high-dimensional data. Thus the authors have proposed a model of spam filter that integrates a high-dimensional N-Gram term frequency–inverse document frequency (*tf-idf*) feature selection. The proposition is composed of a modified distribution-based balancing algorithm [128], and a 'Regularized Deep Multi-Layer Perceptron Neural Network' model with Rectified Linear Units, in a view to capture intricate high-dimensional data features. Multilayer Perceptron (henceforth MLP) is a class of feed-forward ANN. The model does not require any dimensionality reduction and was tested on four different datasets. N-Gram is a contiguous sequence of n terms or items from a sample of speech or text. The results show Deep Neural Networks can be quite promising and the model outperforms number of spam filters commonly in use, showing an accuracy of 98.76% on Enron dataset. The main limitation of [126] is the framework is exceedingly computationally intensive than some of the common techniques available and thus its performance as a spam filter in a more standardized computing hardware is questionable.

'Overfitting' can be a quite critical aspect of a Machine Learning based model, when it models the training data too well, to an extent that it negatively impacts the performance of the model on new data [129], [217].

As seen in the above study and will be highlighted in the studies discussed forward, 'High Dimensionality' of feature space (too many attributes) is a recurring problem in number of Machine Learning dependent models especially that uses Bayesian techniques. With the increase of dimensionality, the complexity rises exponentially; this problematic issue is known as 'Curse of Dimensionality' [130]. To circumvent Curse of Dimensionality many filters perform some degree of 'Dimensionality Reduction' before applying the anti-spam filter to classify incoming messages. Dimensionality Reduction also limits overfitting. With every single addition of dimension d, data increases exponentially i.e. n^d, where n is the number of data points at the start, underscoring the augmentation in complexity that Curse of Dimensionality introduces.

c: NAÏVE BAYES BASED PROPOSITIONS

Another popular supervised algorithm is Naïve Bayes (henceforth NB), developed on Bayes' Rule. Bayes' Rule, introduced by Thomas Bayes, attempts to derive the probability of an event with the help of some prior knowledge of that event-related condition [131]. For instance, if someone's sprinting speed is related to body weight, then with the application of this Bayes' rule, the body weight can be applied to determine individual's sprinting speed more accurately than that of determining sprinting speed without the knowledge of body weight.

Mahdavinejad *et al.* [132] stated that NB classifiers require a limited number of data points for training purposes, they are reliable and considerably faster, as well as efficient in dealing with high-dimensional data points. Bielza and Naga [133] also pointed to similar directions as it argued the Bayesian network classifiers, using some form of Naïve Bayes algorithm, in general are far superior than other pattern recognition classifiers in terms of algorithm efficiency and effectiveness in learning a model from a dataset. However, NB considers features to be completely independent [134], which, in any practical application, is not always the case in majority of the situations.

Zhou *et al.* [135] mentioned that number of research works have used a ternary approach (Spam, Ham and Unsure) of determining whether a mail is a spam or not, using, in most cases, NB classifier. Their proposed modification enhances the calculation and interpretation of the required thresholds, which has been determined in the earlier developed systems just on the intuitive understanding to define ternary email categories. The authors have employed 'Decision-Theoretic Rough Set (DTRS)' models with NB classification to regularize this computation of the threshold value. The result did show significant improvement in 'Cost-sensitivity' (a 'loss function' is regarded as the 'costs' of making classification decisions) in grouping emails into spam, ham and 'suspect' from three different datasets. Despite demonstrating weighted accuracy of 90.05% (assuming that misclassifying a legitimate email as spam is 9 times more costly than the opposite), the proposition doesn't solve the issue of automatically classifying an email as spam, as the user still has to make a decision from the group of email marked as ''Suspect'', and this leaves a potential possibility of error in judgement from the part of the user.

Qingsong and Ting [136] worked on 'Mutual Information Feature Selection' algorithm and introduced Word Frequency

Factor and Average Word Frequency factor to improve upon the application of algorithm on both Chinese and English language corpuses. Emails were classified using NB algorithm and the result indeed showed some improvement. Nevertheless, the Chinese classification showed substandard result than that of the English one; probable cause of which would be the lack of efficient methods in Chinese word segmentation and pre-processing.

Jatana and Sharma [137] presented an improvement of NB algorithm by introducing a fragmented and encoded database technique somewhat similar to Radix Sorting algorithm. The traditional NB based approaches employ simple tokenization of words, that is, only extraction and storage of words as token into some database. The authors, instead, proposed to encode the words using ASCII values and store those in distributed database, in a sorted order, for faster processing. The words are encoded by taking the ASCII values of the alphabets and then finding the difference (absolute) of consecutive words. For example, to tokenize the word 'Speed', the authors have used the following principle after changing it to lower case:

$$\begin{aligned} s - p &= \text{abs}(115 - 112) = 3 \\ p - e &= \text{abs}(112 - 101) = 11 \\ e - e &= \text{abs}(101 - 101) = 0 \\ e - d &= \text{abs}(101 - 100) = 1 \end{aligned}$$

So the code for token 'Speed' is 31101. Now the distributed database is broken into 26 different sets (0-25). The encoded tokens are stored in these databases based on the difference of the first two characters, which is in this case is '3'. Thus the token 31101 gets stored into dataset numbered '3'. Tokens are in this way stored in sorted order in the repository. Binary search is used for searching purposes to determine the top K token with highest probabilities.

The researchers [137] claim that it enhanced execution speed of the algorithm nearly six times, tested on LingSpam and SpamAssassin. Nonetheless the work has more room for improvements, for instance if hash functions are used in sorting [137].

Ranganayakulu and Chellappan [138] considered host based and lexical features, Age of domain and Page Rank to classify URL within the email body to be malicious or not. With a rather minimal of feature set, Bayshean classifiers have been put into use for classification purposes. The classifier deals with a training dataset of malicious phishing URLs and legitimate URLs. The probability for each of the features to occur in the dataset is calculated and their respective scores are obtained through cumulative addition. Finally if the cumulative score crosses the threshold value, the system determines a malicious phishing URL is present in the email, and thus it is a spam.

The above system [138] demonstrated an FPR (False Positive Rate) of 0.4% and TPR (True Positive Rate) of 92.8%. Though the framework is quite compact, nonetheless, the 'Page Rank' features has been suspended by Google as of now, moreover the logic behind calculating the domain age is not so clear. Similar observation goes for the 'Number of Dots' parameter.

Hayat *et al.* [97] in their work introduced a framework, on top of traditional NB, that showed improved performance on a simulated future direction over implementation that uses NB classification in a straightforward way. The work compares a batch of emails to that of the old ones, and if the distributions seem considerably distinct, the mechanism stipulates 'Concept Drift' has taken place, and updates itself based on the hybridization of the two concerned models. The resulting model displayed an improvement of 8%-9% in terms of accuracy over multinomial NB. However, the research initiative needs to be more adaptive in the sense that instead of judging a group of emails for the occurrence of Concept Drift, the author feels it should be checked after every single email and the system should update itself accordingly if the need be; but that might hamper the usual performance [97].

Lee *et al.* [139], used Weighted Naïve Bayes (WNB) along with natural language features such as Parts of Speech (POS) tagging [72] to formulate a spam filter that only examines the subject header. The system transforms a subject line into a feature vector $x = (x_1, x_2, \ldots, x_n)$, where x_i is the value of feature X_i. Alongside POS, to determine the value of x_i, both Bag-of-Words [139] method and statistical features of the subject such as total length, case and composition have been utilized. With every input of new feature vector x, denoting h and s as the ham and spam class respectively, the algorithm predicts that x is in class c_l according to the current training set X, as shown in (6). In (6), $p(c_l|n)$ is the posterior probability of modified WNB [139].

$$l = arg \max_{l \in \{h,s\}} p(c_l|x) \tag{6}$$

The framework achieved 95.74% accuracy on Enron dataset. On the contrary, POS tagging is rather ineffective against those subject headers that contain word obfuscation [72] and animated contents.

d: DECISION TREE BASED PROPOSITIONS

One of the most impactful algorithms in the field of Machine Learning is the Decision Tree (henceforth DT) algorithm. DT based 'Learning', in most cases commonly employs an upside-down tree based progression method. DT can be used to resolve both classification and regression problems. [140]. The growth of the tree from the root node starts by deciding upon a 'Best Feature' or 'Best Attribute' from the set of available attributes, and then by applying splitting.

In majority of the instances, the selection of 'Best Attribute' is done through the calculation of two more measurements, 'Entropy' as shown in (7), and subsequently calculating the Information Gain, shown in (8). The 'best attribute' is the one that imparts most information. Entropy defines how homogeneous, or the lack of, the dataset is and Information Gain is the change in Entropy of an attribute, usually a reduction [140].

$$E(D) = -P(_{positive})log_2P(_{positive}) - P(_{negative})log_2P(_{negative}) \tag{7}$$

Equation (9) calculates the Entropy E, of a dataset D, which holds the positive and negative 'Decision Attributes'.

$$Gain\ (Attribute\ X) = Entropy\ (Decision\ Attribute\ Y) - Entropy(X, Y) \quad (8)$$

Gain is calculated for each of the features or attributes and the one with the highest value is selected as it provides most information gain. The whole process is repeated for sub-branches of the tree to eventually complete the DT.

Ouyang *et al.* [141] developed frameworks based on DT and another algorithm known as 'Rulefit' [142], in order to carry out a comprehensive empirical study into the efficacy of using packet and flow features in the detection of spam emails from a single-enterprise perspective. The flow based analysis critically examines an email using different methods, such as DNS Blacklisting, filters that works on SYN packet features, filters based on key traffic characteristics and finally content analysis. Addition of each of the stages in the processing adds more overhead and thus computational complexity increases. A message is marked as spam when any of the layers confidently labels it as spam. Researchers claim that the proposed work on network level filtering for spam detection can greatly reduce the workload for a more intensive content level filtering.

Sheikhalishahi *et al.* [143] envisioned a preliminary approach to a new algorithm known as 'Categorical Clustering Tree (CCTree)', which is based on DT. CCTRee extracts a set of categorical features such as size of email, number of embedded links, attachment information, HTML characteristics etc. to build a tree of clusters. The researchers argued that it has a simpler approach in dealing with the task in hand, thus the complexity is quite low comparing to some other approaches. On the other hand, the research attempt is yet to be tested and implemented on a large-scale dataset, and it only showed some theoretical underpinning where the low complexity and easy-to-understand representation of the chosen features have been highlighted as the key strength of the proposed CCTree algorithm.

To address the issue of 'Concept Drift', Sheu *et al.* [144] designed a DT-based framework that works in conjunction with 'Incremental Learning' of spam keywords, set up in an online environment for continuous enrichment. The precision attained at the point of publication is 95.5%.

e: RANDOM FOREST BASED FRAMEWORKS

Random Forest (henceforth RF) is one of the most successful supervised classification and regression techniques based on ensemble learning. It operates by constructing an entire forest from multitudes of random and uncorrelated Decision Trees during training segment [145]. Ensemble learning methods employ multiple learning algorithms to come up with an optimal predictive analytics, which can perform better than any of the individual model's prediction [145]. RF may incur additional complexity in its calculation as it uses a lot more features than a standalone DT, but generally it does produce higher accuracy in dealing with unseen datasets.

The ensemble method upon which RF is founded is known as 'Bootstrap Aggregation', a powerful, yet simple algorithm. Bootstrap Aggregation can address overfitting arising out of high variance of algorithms such as DT.

Tran *et al.* [146] indicated that limited work has been done on detecting malicious contents in spam emails, in the form of malignant URL or harmful attachments (malware). Coders for malware are relentlessly developing novel and clever techniques for transforming binary code that cannot be detected by anti-virus scanners, and their level of sophistication is growing with time [29]. The proposed model extracts many different features in a rather time-efficient manner from email content and metadata, without using external tools. Some of the header and content features used are quite unique, and according to the authors have not been used elsewhere. RF has been applied to measure the effectiveness of the selected features. Nevertheless, the authors have suggested the system does not always perform well enough against detecting malicious URLs in spam emails, but rather effective against detecting potentially hazardous attachments.

The proposed model of Shams and Mercer [147] extracted features such as words, length of words and documents etc. and carried out the classification task on four different data sets such as CSDMC2010, SpamAssassin, LingSpam and Enron with multiple classification algorithms. The authors claim that classifiers generated using meta-learning algorithm ('Bagging' in this case) performs better than probabilistic and tree based models. The Bagging model demonstrated average accuracy of 94.75% across all four datasets. The algorithm is a close variation of RF. Regardless of reasonable accuracy, the work does not address the issue of high dimensionality and the associated increase in the complexity of the proposed methods.

f: LOGISTIC REGRESSION BASED SOLUTIONS

Logistic Regression (henceforth LR) is another simple, yet very useful supervised approach, applicable to a wide range of binary classification problems, for instance, predicting binary-valued labels for a data point z such that $z^{(i)} \in \{0, 1\}$, such probabilities can be calculated from (10) and (11).

$$P(z = 1|x) = \frac{1}{1 + exp(-\theta^T x)} \quad (9)$$

$$P(z = 0|x) = 1 - P(z = 1|x) \quad (10)$$

The right hand side of (9) will 'squash' the value of θ^T within the range of 0 to 1 so that it can be interpreted as a Probability [148]. Over the years, apart from scientific research, LR has been widely adopted in many different fields such as marketing, health care and economics to name a few [149]. Both the equation signify how likely the probability is to fall within the value 0 and 1 strictly.

Pawar and Patil [150] demonstrated for a small dataset of less than a thousand, a regular LR model performs best (accuracy of 98%), however, as the research suggests, to keep

the performance consistence with the growth of dataset, another version of LR, the Multiple Instance Logistic Regression (MILR) should be used, as it demonstrated a consistent accuracy within the tight range of 93.3% to 94.6% (up to 2500 data points).

g: SUPPORT VECTOR MACHINE BASED PROPOSITIONS

Support Vector Machine (henceforth SVM), a well-established supervised learning technique used for classification, was originally proposed by Alexey Chervonenkis and Vladimir N. Vapnik in 1963 [151].

A hyperplane creates the differentiating classes by analysing various features found in the dataset. The SVM can work in any number of dimensions. In a $\mathbb{R}^2$ (2-dimensional) workspace, a hyperplane is a line, in $\mathbb{R}^3$ it is a plane and in $\mathbb{R}^n$ it is termed as 'hyperplane'. The algorithm may identify several hyperplanes. But the optimum one would be the one that has the maximum distance from the training datasets of each of the classes. Given a specific hyperplane, the computed distance from it to the nearest data points of both sides can be used to draw the margin, There should never be any data points inside the margins. The bigger the margins, the better the model will generalize with unseen data. Support Vectors, required to calculate the margins, are the data points 'On' or 'Closest' to the margins [152]. Though SVM is a supervised techniques, but work has also been done to use it in unsupervised clustering [153], [154]. SVM has the unique ability to transform non-linearly separable data to a new linearly separable data by a mechanism known as 'kernel trick' [154].

Similar to the study of [126], Diale *et al.* [155] demonstrated that while using SVM for email classification, optimising the kernel type and kernel parameters are of utmost importance. The authors have indicated that varying feature extraction and feature selection techniques for SVM often bring about the need for employing different kernel functions for optimum performance. They have also concluded that increasing number of features available for feature selection and extraction resulted in better performance, that is, there is a positive correlation. This research attempt primarily works with the words from email body for feature engineering; but excludes other forms of features, such as header, URL and domain information. Thus the obtained results are only accurate within a limited boundary of circumstances.

A combined model has been envisioned by Amayri and Bouguila [156] where both textual and visual (images) information from emails have been combined and simultaneously put into action in detecting spams. The framework is based on building probabilistic SVM kernels from mixture of Langevin distributions [156]. For the textual features, certain header information, such as FROM, REPLY-TO, Cc, Bcc and TO fields have been consulted along with email content and subject. For the visual part, texts in the embedded images have been extracted using OCR, and certain visual features of the images have also been included in the feature vector. For the SVM kernel, the authors have experimented with three different flavours and came into conclusion that Bhattacharyya Kernel (BK) works best. The framework attained an accuracy of around 92% when both images and texts are considered from an email, however, the accuracy declines for spam which is based solely on image or text.

Deep Support Vector Machine (DeepSVM) are known to perform better than CNNs and standalone SVMs due to a design improvement where instead of single layer, any number of layers can be used with kernel functions. Roy *et al.* [157] proposed an application of Deep SVMs in spam classification. The lower level SVMs carry out the task of feature extraction while the highest level SVM performs the actual prediction using the extracted features. The authors have also compared the performance with ANN and regular SVM. The model based on Deep SVM showed highest accuracy of 92.8%.

h: ADABOOST BASED PROPOSITIONS

Boosting basically means to combine a number of simple learners (classifiers that produces an accuracy just above 50%) to formulate a highly accurate prediction. AdaBoost (Adaptive Boosting) sets different weights to both samples and classifiers [158]. This enforces the classifiers to put concentrated focus on observations that are rather difficult to accurately classify. The formula for the final classifier is shown at (11).

$$H(p) = +/- (\sum_{k=1}^{K} \alpha_k h_k(p)) \tag{11}$$

Equation (11) is a linear combination of all of the weak classifiers (simple learners), where K is the total number of weak classifiers, $h_k(p)$ is the output of weak classifier t (can only be -1 or 1). $\boldsymbol{\alpha}_k$ is the weight applied to classifier k. The final decision is derived by looking at the sign (+/-) of (13).

The research done by Varghese and Dhanya [159] attempted to develop a filter using Parts of Speech (POS) Tag, Bigram POS Tag, Bag-of-Word (BoW)s and Bigram Bag-of-Word (BoW)s. It has been detected that POS tags and Bigram POS Tag features demonstrated better output using AdaBoost as the classifier; the experimentation achieved a False Positive Rate of 0. On the contrary, [159] suffers from the same issue due to POS Tagging as discussed earlier. In addition, as pointed out in [160], Adaboost as a classifier might incur issues such as high computational cost and non-scalability. Apart from this, the work does not address any header information, leaving a loophole for number of different types of spam emails.

i: K-NEAREST NEIGHBOUR BASED SOLUTIONS

K-Nearest Neighbour (henceforth KNN) is widely used classification technique that boasts a commendable balance among several important criterion such as predictive ability, intuitive interpretability and time required for calculation (for a moderately rich dataset). Though algorithms such as RF does have higher capability in prediction, but lags behind in few other parameters. Unsurprisingly, industry adoption

of KNN is quite high. KNN can often be used to formulate regression models, but that is not very common. KNN uses 'Euclidian Distance' to determine the distance between two data points (X^n and X^m) as shown in (12) [161].

$$Dist(X^n, X^m) = \sqrt{\sum_{i=1}^{D} X_i^n - X_i^m)^2} \tag{12}$$

On the hindsight, instead of using Euclidean Distance, Sharma and Suryawanshi [162] have proposed 'Spearman Correlation' [163] as the distance measure for KNN based classification as shown in (13). X and Y are training and testing tuple respectively while n is the number of observations. The values of d_{ij} usually lies between 1 and -1.

$$d_{ij} = 1 - \frac{6 \sum_{i=1}^{n} (rank(X_i) - rank(Y_j))^2}{n(n^2 - 1)} \tag{13}$$

The changes have shown some enhancements over regular KNN model with nearly 50% improvement in accuracy (97.44% in 80%-20% Train and Test ratio). A limitation of the study is the size of the dataset, having just over 4000 data points. KNN often needs a rather large dataset to produce a rather stable model with realistic accuracy. Besides, a bit of elaboration was needed for fixing the value of K as 3. The authors have also expects the study to be used in conjunction with other more robust and complete spam filtering frameworks.

j: MULTI-ALGORITHM SUPERVISED SYSTEMS

A number of interesting propositions have been put forward that employ more than one supervised algorithms in different segment of the framework to develop the final model. This section will highlights some of such recent solutions that, mostly is a hybrid of the above discussed algorithms.

In his study, Wang [164] proposed a heterogeneous ensemble approach for spam detection composed of DT, NB and Bayesian Net algorithm. Heterogeneous ensembles composed of methodologically different learning algorithms. The study have also discussed multiple procedures for algorithm selection in building the ensembles. The researchers have compared the framework with homogenous ensemble techniques and found their approach to be performing better with an accuracy of 94%.

Similar to [164], *Large et al.* [165] also suggested that heterogeneous ensemble-based spam filtering frameworks perform better. However, the researchers argued that instead of simple tree based ensemble techniques used in [164], the more advanced ones, based on slightly complex algorithms such as RF, Rotation Forest, Deep Neural Network and Support Vector Machine, can actually perform better in varieties of scenarios. The claim can also be substantiated from Shuaib *et al.* [166] where the researchers have reviewed number of classification algorithms and found Rotation Forest to be performing better than some other common algorithms with an accuracy of 94.2% on the Spambase dataset. However, the authors have used a 66% split, rather than the more traditional 80%-20%, without really explaining the rationale behind the choice.

DT based systems often tend to have high sensitivity to noise and overfitting. This issue has been highlighted in the work of Wijaya and Bisri [167]. To tackle such issue, the researchers have added a regular LR to the process. In this hybrid spam detection system, data is fed into an LR module before passing through the DT based segment. The reported accuracy is 91.67%. The work does not use any feature engineering methods, and simply uses all the available aspects as features. This simplicity gives the framework effortless execution, but makes the accuracy less realistic.

It has been stated by Nizamani *et al.* [168] that efficient and advanced feature selection weights more than the types of classification algorithms used when comes to identifying deceitful emails. The authors have employed SVM, J48 Decision Tree (implemented in Java), CCM (Cardiac Contractility Modulation) and NB classifiers together with various carefully designed features and disseminates the idea that frequency based features generally achieve top accuracy, 96% in their study. The work only deals with the contents of the fraudulent emails for feature extraction, ignoring the header, which is also an important aspect that needs to be considered. Besides, Alsmadi and Alhami [169] argued that better false positive rate can be acquired through the deployment of N-Gram based clustering and classification than employing any other algorithms, even the one discussed in [168].

Feng *et al.* [170] offered a hybrid model composed of NB and SVM, attaining an accuracy of around 91.5% with a training set of 8,000 samples. The framework tries to reduce dependency issue among features as much as possible - commonly observed in NB based models. The study aims to extend its functionality towards image spam as one of the future improvements. However, we believe the authors [170] should also include header and domain information in its analysis of spam emails.

Islam and Abawajy [171] developed a multitier classifier where an email is checked for an accurate labelling in the first two tires, and if any misclassification occurs (initial two tires giving out conflicting labelling), it is then sent to third tier. The choice of algorithms (SVM, AdaBoost and NB) picked for each of the tires has been decided after juggling the selected algorithms and their respective tiers. A strength of this technique is that the processing among tiers transpire in parallel, unlike some other ensemble based multitier classifiers. The model returned a high accuracy rate (around 96.8%) with low rate of false positive detection.

A behavior-based mechanism has been discussed by Hamid and Abawajy [172] to detect phishing emails using hybrid feature selection approach. They have deployed 4 different classifiers (Bayes Net, AdaBoost, Decision Table and Random Forest) to mine sender's behaviour, in a view to find out whether the source is a legitimate one. Sender's behaviour is further broken down into two subgroups: Unique Sender (US) and Unique Domain (UD). The inputs to the

Sender Behaviour algorithm are the domain message-id (DMID) and lists of email sender (ES). The system showed an average accuracy of 93% with only 7 features from a rather limited set of 3000 data points. On the other hand, a similar framework [173], achieving a slender advantage in terms of accuracy, used a high number of features - 43.

The framework developed by More and Kulkarni [174] and tested on Enron dataset using NB and RF demonstrated an accuracy of 96.87%. The system employs text analysis of the email body using NB, and categorizes the words in several linguistic features as well specific spam words. If it is found that the message body contains over 5% spam words, it is flagged as Spam. Besides, the same set of emails are passed through a classification system built on RF that uses the following Polynomial Kernel Function as shown in (14). X and Y are vectors of features derived from test or training samples, and C is constant.

$$K(X, Y) = (X^T Y + C)^2 \tag{14}$$

The obtained result has also been compared with ANN and LR built model (following the same Kernel Function). An improvement can be added if the issue of high dimensionality and the associated increase in the complexity of the proposed methods can be explained in detail.

Islam and Xiang [8] developed a promising email classification technique based on data filtering method. The work broached an innovative filtering technique using a modified 'Instance Selection Method (ISM)' to cut down on the least valuable data instances from training model and then classify the test data. The aim of ISM, enhanced by NB, is to identify which instances (examples, patterns) in email corpora should be selected as representatives of the entire dataset, without significant loss of information. Several algorithms have been tried and the model displayed an accuracy of 96.5%. However, according to the authors, the system needs to have the capability to handle incoming emails to address Concept Drift [8].

k: SUPERVISED SYSTEMS DISCUSSING PERFORMANCE OF DIFFERENT ALGORITHMS

The core of the above discussed systems are either built with a single supervised algorithm or multiple ones. Below are a discussion on single-algorithm frameworks where multiple algorithms have been individually tested to design the primary classifier of the system, and based on the performance, the best one has been chosen to finalize the classifier.

The proposed binary classification model named 'Sentinel' by Shams and Mercer [147] utilized features of Natural Language Processing before developing the classifier with multiple supervised algorithms in a view to evaluate performance of each of those algorithms. Among the five algorithms tested, RF, Adaboost, Bagged Random Forest, SVM and NB, Adaboost and Bagged Random Forest performed equally best on four different spam datasets. However, real time training and response latency have not been considered as well as performance against Concept Drift [97] is yet unknown.

Aski *et al.* [176] brought forward a rule based framework where 23 meticulously chosen features have been selected from a personally compiled spam databases and each of these criterion have been scored to get a total value, which was subsequently compared to a threshold value to finally label an email as spam or ham. MLP, NB classifier and C4.5 Decision Tree classifier have been used to train the model as well as the individual model's performances have been compared; of which the MLP based model scored accuracy around 99% [176]. The MLP based model propagates information by activating input neurons that contains labelled values. The Activation of neurons is calculated either in the middle or output layer using (15), where a_i represents the activation level of neuron i; j denotes neuron set of the previous layer; W_{ij} is the weight of the link between neuron j and I, and O_j is the output of neuron j.

$$a_i = \sigma(\sum_j W_{ij} O_j) \tag{15}$$

However, the small testing dataset (750 spam and ham in total) is somewhat limits the wide acceptance of the results obtained for this study. Thus an effective performance measure in terms of memory and time footprint for large scale datasets is yet to be determined, also, the study does not mention how the model will perform against certain critical attacks such as spear phishing.

As illustrated in earlier works, 'Baysian Probability Theorem' has been the choice for handling uncertainty in datasets. However, the work of Zhang *et al.* [177] rather argued the 'Dempster-Shafer (D-S) theory of evidence' [178] is better equipped than Baysian probability while using statistical classification. Uncertainty can arise in number of regards in the analysis of spam corpuses such as assigning missing values to features. In D-S theory, given a domain $\boldsymbol{\alpha}$, a probability mass is assigned separately to each subset of $\boldsymbol{\alpha}$, whereas in classical probability theory, this probability mass is assigned to each individual elements. Such an assignment is called a Basic Probability Assignment or *BPA* [179]. The researchers have selected 5 most representative header features of spam corpus after appropriate quantification. Their D-S integrated classification model found ANN to be one of the most effective classification algorithms along with NB.

Ergin and Isik [180] highlighted the fact that spam is not only a problem in emails based on English language, but also non-English speakers also have to deal with the issue. The work in question demonstrated a Turkish spam filtering system developed with the aid of DT and ANN as classifiers, while 'Mutual Information (MI)' method has been deployed for feature selection. ANN attained an accuracy of 91.08%. Though the study states the superiority of Mutual Information (MI) over more widely applied technique - 'Information Gain (IG)', the extensive study by George [181] found otherwise, where it has been concluded the performance of Mutual Information is not up to the scratch, mainly due to its sensitivity to probability estimation error.

Sharaff *et al.* [182] conducted experimentation on a processed Enron dataset with standard DT, J48 Decision Trees, SVM and BayesNet. The study reported the effectiveness of J48 and BayesNet over SVM.

Sharma and Kaur [183] tested a spam detection framework built upon RBF (Radial Bias Function) Network (a subclass of ANN), where neurons were separately trained to address common spamming techniques. The approach seemed to have increased the performance of RBF and also outperformed SVM. The research resulted in an average accuracy of 99.83% after five consecutive runs. Nonetheless, the dataset of just 1000 words is not comprehensive at all, and the proposed feature extraction method is rather vague.

Saab *et al.* [184] also measures the performance of SVM, Local Mixture SVM, DT and ANN on spambase dataset. While taking into account the full 57 available features, SVM demonstrated the highest precision (93.42%), while ANN the highest accuracy (94.02%). However, this high accuracy was achieved in exchange of the longest training time.

The presence of malicious URL in phishing emails is a key characteristics of spam emails and Vanhoenshoven *et al.* [185] tested the effectiveness of RF in detecting such URLs within spam emails using a publicly available database. The authors came into conclusion that with an accuracy of 97.69%, RF actually performed better than few other classification techniques such as MLP, C4.5 Decision Tree, SVM and NB. Features were ranked with Pearson Correlation Coefficient' [186] for selection. Qaroush *et al.* [187] also justified the superiority of RF (reported accuracy of 99.27%) by comparing its performance against several other classification methods while building the classifier using various important email header features.

A study based on semantic method has been introduced by Bhagat *et al.* [188] using Wordnet ontology [189] as well as some 'Similarity Measures' to reduce the high number of extracted features. 'Path Length Measure' has been chosen as the most suitable algorithm for determining the similarity measures. Path Length Measure derives the semantic similarity of a pair of concepts. The calculation starts by counting the number of nodes along the shortest path between the concepts which can be found in the 'is-a' hierarchies of WordNet. In general, the path similarity score is inversely proportional to the number of nodes along the shortest path between the two words. Equation (16) summarizes the nature of the derived score where *w1* and *w2* are the two terms

$$PATH\ (w1, w2) = \frac{1}{length(w1, w2)} \tag{16}$$

This resulted reduction of high number of features also reduces space and time complexity. *tf-idf* has been used for feature updates while feature selection is done with Principal Component Analysis (PCS). Multiple supervised algorithms have been used for classification evaluation and the system projected an average accuracy of over 90% with considerable dimensionality reduction of feature set; LR found to be performing optimally. Nevertheless, the reduced feature set of 70 is still a bit too large and more effective testing against phishing attacks is required.

Bhagat and Moawad [190] carried out similar semantic based implementations. The resulting reduction of feature set was around 37%, with LR showing optimal performance with an accuracy of 96% while RF performed the least, demonstrating accuracy around 85%. The somewhat similar study of Bhagat *et al.* [191] attained a feature reduction rate of 43.5% through the stemming of the email body on Enron dataset. Multiple classifiers have been tested and SVM and LR performed comparably better than other classifiers with LR showing an accuracy of 97.7%.

Nonetheless, both [190] and [191] suffers from contextual ambiguity issues. Ambiguity refers to the fact that a sentence in context may indicate multiple meaning, for example, ''There was not a single man at the party'', can be interpreted as *I)* Absence of bachelors at the party *II)* Absence of men altogether [192]. The right conclusion can be deduced upon analysing the context within which the sentence has been used.

Besides the above studies, Almeida *et al.* [193] conceived a process of expanding short texts, often found in SMS spams, but could sometimes be seen in spam emails too. The authors argued that when the original text is too short and mostly filled with abbreviations and idioms, it can be harder to apply any sort of classification algorithm on it, most because the feature set is also extremely limited. Feature Engineering is also difficult out of this limited initial feature set. Their proposed normalization and expansion method is based on semantic dictionaries, lexicography and highly effective techniques for semantic analysis and disambiguation. The study can also generate novel attributes to feed into any classification algorithm. The statistical evaluation done on the output showed promising directions. However, the researchers concluded more thorough testing and performance measurements are required.

Méndez *et al.* [194] devised a semantic-based feature selection approach. The first critical segment of the proposed method is e-mail topic extractor and guesser and the other one is computing the topic-related significance of each feature. To guess the topic of the email, the researchers have semantically grouped terms into more generic topics, that is, each of the topic has a bunch of related terms under it, and the more the terms are found in the content from a certain category, the higher the likelihood of the email being belonging to that topic. These root level of topic is taken from the Wordnet Lexical Database. The logic ensures each email may actually fall under multiple topic. The set of topic comprises both spam and ham groups. Finally, it is then determined whether the email contains higher number of spam topic, in which case it is declared as spam. The model has been evaluated against several common Machine Learning Algorithms for benchmarking. The proposed 'Topic Guessing' technique showed significant improvement especially in terms of performance. However, the authors feel that the manual specification of root topic level needs further attention.

5) UNSUPERVISED LEARNING BASED PROPOSITIONS

This section will analyse a number of Machine Learning based research attempts which are primarily unsupervised by nature. These include one or multiple unsupervised algorithms to develop automated spam detection framework.

a: K-MEANS CLUSTRING BASED FRAMEWORKS

K-means Clustering is one particularly useful, simple and popular algorithm which intends to group similar data points together in a view to finding the underlying pattern. The algorithm produces the final output through iterative refinement. The number of groups is denoted by K, and iteratively each data point is assigned to one of these groups of clusters based on the identified similarities among the features [195].

Determining the optimum value for K, the total number of clusters, which needs to be inputted for the algorithm to work, can sometimes be tricky and users often run the system multiple times with different values of K to compare the results. Several methods exist for getting a reasonably solid approximation of K [195].

In their work, Basavaraju and Prabhakar [196] proposed system that employs the text clustering based on 'Vector Space Model (VSM)'. The method performed reasonably well on identifying spam emails. Representation of data is done using a VSM and data reduction has been achieved through a custom developed Clustering techniques using the features of K-Means algorithm and BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm, achieving an accuracy of around 76%. This study uses raw words from the documents to develop the VSM, A point of concern for the system is that in case of spammers using character variations, such as disguising the word insurance as I∗n$u∗rènce, it will be difficult for the framework to work correctly.

A content based approach has been put forward by Laorden *et al.* [197]. The proposition works on anomaly detection to spam filtering by comparing features such 'Word Frequency' to that of a dataset of ham, or valid email. The inspected email, if shows considerable deviation from a normal scale, will be considered as spam. The techniques utilizes an algorithm known as 'Quality Threshold (QT)', which basically falls into the category of Partitional Clustering algorithms [198], a close variation of K-Means Clustering, giving an edge in reducing the number of vectors in the dataset used as normality. This attempt also lessens the processing overhead significantly. On the contrary, the system may render ineffective against the usage of language features such as Synonyms, Hyponyms [199] and Metonymy. The study achieved a weighted accuracy of 92.27% on LingSpam dataset. *Basnet et al.* [27], also reported similar accuracy of 90.6% using k-means.

'Authorship Attribution' in recent times has become a valuable tool in resolving issues around authorial disputes mainly in historic documents and literatures. Patterns regarding grammatical and syntactic features emerging out of such documents may lead to successful grouping and identification of original authors. Even though emails are highly unstructured, Alazab *et al.* [200] tried to implement the idea on spam detection, especially for phishing campaign identification. The researchers have deployed an Unsupervised Automated Natural Cluster Ensemble (NUANCE) methodology to approximately cluster spam emails. The final clustering is achieved by hierarchically clustering the approximate sets, giving 27 different clusters. Though the system is impressive and achieves improvement in the general direction of 'authorship attribution' in spam campaign detection, however, the intra-dynamics within the campaign groups may go undetected.

Halder *et al.* [201] used clustering algorithms such as K-Means and Expectation Maximization (henceforth EM) on schemas such as stylistic features of emails, for example total number of punctuations and contractions, number of email IDs used in the body etc. The authors have also looked into sematic features, that is, statistical measures of different words used in a batch of emails. Besides, they have also taken the combination of these two approaches into account. The cluster analysis has been carried out on a dataset of 2600 spam emails. It was detected that this method can be successfully deployed to identify writing styles of spam campaigns. Further, prototypes can be built based upon the extracted patterns for future identification of spam emails. K-Means showed 80% success rate when a combined approach is taken while EM projected a success rate of 84.6% while dealing with only semantic features. However, its detection rate drops to 57.4% while using a combined approach. The success rates have been reported in terms of 'Purity' of clusters – which basically projects the quality of the cluster. The accuracy should also be within similar range. The authors' area of concentration has generally been rather narrow as there are number of important features in spam email detection such as email subject headers, URLs composition, detailed header and domain information, attachments etc. which have not been discussed, thus there are rooms for a considerable expansion of this work.

The expectation Maximization (EM) is an effective iterative algorithm that calculates the Maximum Likelihood (ML) estimate in the presence of hidden or missing data [202]. Latent variables, or unobserved variables which cannot be directly measured, but rather can be inferred from the non-latent variables, are often used in an EM models for gauging the best estimation.

b: SELF-ORGANIZING MAP BASED PROPOSITIONS

Self-Organizing Map (SOM), an unsupervised technique that borrows the baseline idea from ANN. However, instead of 'Backpropagation', it uses a process called 'Competitive Learning' to produce a two-dimensional map of input space with higher dimension [203]. It is conceptualized that in Competitive Learning, output neurons are in competition to respond to input patterns. At training stage, the output unit that is able to provide the highest activation to a given

input pattern is brought closer to the input pattern, whereas the rest of the neurons are left unchanged. The process is repeated number of times, eventually forming clusters of closely related data points [204].

Porras *et al.* [206] declared that several benefits can be obtained if SOM is used instead of KNN for clustering, such as eradication of inputting the number of clusters as one of the parameters to the algorithm. Instead SOM can use a threshold, a radius-based boundary, to manipulate the algorithm's sensitivity. Further, topological aspect of the similarity among several clusters can be observed with much ease. Multiple filtering systems work in unison for spam detection in this model. On the contrary, according to the author, SOM calculation complexity may make it less than optimum for datasets that are limited both in the aspect of size and diversity. The experiment has been carried out on a dataset of 6047 email messages.

Cabrera-León *et al.* [207] introduced another SOM based system where they used 13 different categories for the emails. The researchers had started by a 4-stage preprocessing of emails (both spam and ham). First stage batch-extracted all the emails' subject and content and filled whitespaces with alphanumeric characters. The second stage removed all the stop words and calculated raw Term Frequency measure along with some other metadata (spam\jam) to the processing. The following stage built a 13-dimensional integer array to hold the themes and categorize the above-processed texts. The last preprocessing phase added 'weights' to the words of each of the 13 categories. The model was then built using SOM (with 'Batch' learning method), finally, a threshold value was used to label the clusters accordingly. The framework recorded an accuracy of 94.4%. A concern was noted by the authors in the performance of the model against newer and off-topic emails, where the accuracy did get affected, leaving more room for improvement.

c: PRINCIPAL COMPONENT ANALYSIS (PCA) BASED FRAMEWORKS

PCA is a statistical framework that works extremely well in most cases for Dimensionality Reduction in such a fashion where the maximum variations of the dataset can be retained [208]. PCA is also a valuable tool in building Predictive Models. The system is an 'Orthogonal Linear Transformation' that transmutes the normalized inputted data to a new coordinate system [209].

Dagher and Antoun [210] deployed four different scenarios regarding feature pre-processing using PCA (Principal Component Analysis). Out of the four, two notable ones are - representing ham and spam emails using same and different set of features. It has been reported that PCA performs best when both the classes of emails are represented using same features, having an accuracy of 94.5%. On the hindsight, depending upon the best selected features, the other three scenarios may as well perform differently. Besides there is no mention of the fact how the features have been selected to begin with. In addition, the spam dataset is rather limited in number of emails it holds.

A variation of standard PCA, termed as 'PCAII', has been broached by Gomez *et al.* [211], where the features of both the classes under analysis are combined together. Few variations of 'Latent Dirichlet Allocation' (LDA) have also been proposed in this work. The modified algorithm had been applied on TREC 2007 spam corpus and the output showed a balanced and stable performance regardless of dimensionality reduction.

6) SEMI-SUPERVISED LEARNING BASED SYSTEMS

Semi-supervised spam filtering systems have also demonstrated promise, even though not many attempts have been taken to construct such systems yet. This section will shed some light on few of such frameworks.

Las-Casas *et al.* [103] inspired a technique called 'SpaDes' (Spammer Detection at the Source), which works by analysing SMTP metrics such as number of distinct SMTP servers targeted, number of observed SMTP transactions, average geodesic distance to destination, average transaction size (in bytes) and average SMTP transaction inter-arrival time (IAT). These SMTP metrics are studied via a Machine Learning algorithm known as 'Active Lazy Associative Classification' (ALAC) [212] to build a prediction model. Associative classification method aims to amalgamate supervised classification and unsupervised association rule mining techniques in order to build a model known as associative classifier. Though the proposition did show reasonably satisfactory performance, however, it has been reported that over time the system did not produce consistent performance, due to the changes in behaviour of the spammers' way of carrying out spamming with time. The role and impact of Machine Learning based algorithms in the detection of spam emails will be further discussed in the following sections.

Smadi *et al.* [15] presented a framework, 'Phishing Email Detection System (PEDS)' based on both supervised and unsupervised techniques in conjunction with reinforcement learning methodology, which gives the system an increased ability to adapt itself based on the detected changes and modifications in the environment. The target of the system is Zero-Day Phishing attacks [15]. The core of the system, 'Feature Evaluation and Reduction (FEaR)' algorithm, can select and rank the important features from emails dynamically based on the environmental parameters. FEaR is based on Regression Tree (RT) algorithm, a subtype of Decision Tree. Immediately after the execution of FEaR, another novel algorithm, DENNuRL (Dynamic Evolving Neural Network using Reinforcement Learning) will take over to allow the core Three-Layer Neural Network of PEDS to evolve dynamically and build the optimum Neural Network possible. DENNuRL has the element of Reinforcement Learning where the degree of 'Reward' has been linked with the Mean Square Error (MSE) of the Neural Network in (16) and (17). In (16), n is the number of emails used in the evaluation process, oi is the output for an email and ti is the desired target for the same

email.

$$MSE = \frac{\sum_{i=1}^{n} (o_i - t_i)^2}{n} \quad (17)$$

$$Reward = \frac{1}{MSE} \quad (18)$$

Though the achieved accuracy rate is 99.05%, some of the features that have been used are rather unconventional; for example 'BodyDearWord', 'BodyNumChars', 'BodyNumWords', 'NumLinkNonASCII', and 'ContainScript'. These features do not have any real significance on whether a mail is spam. The authors have not argued the inclusion of these, which leaves a scope for improvement.

Another study by Hassan [213] investigated the effect of combining text clustering using K-Means algorithm with various supervised classification algorithms. Some of the features from clustering space have been shared with classification module to gauge the degree of improvement in classification. However, the outcome portrayed an insignificant gain which is not really viable against the added computational complexity. Table 7 tabulates a summarized view of a 'sample' of 42 studies drawn from the Machine Learning based techniques discussed in this section. Out of this 42 studies, 28 are supervised, 6 are semi-supervised and 8 and unsupervised.

The semi-supervised model put forward by Padhiyar and Rekh [214] has been built upon KNN and NB. The model claims to achieve improved classification accuracy than a standalone NB or KNN based methods. But further inspection shows that probably the work will not be highly accurate when availability of initial labelled documents are limited. As explained earlier, labelled documents contain labelled data; that is data for which the target answer is already known. However, the study proposes Expectation Maximization (EM) to be added to handle the scarcity of labelled data, but it has not been implemented in the study. Further, the implemented framework lacks an effective feature selection and pre-processing module.

Though KNN works well in majority of the cases, but Chakrabarty and Roy [215] highlighted few issues with the logic behind the calculation of the similarity measure; which creates the requirement of high memory usage and complicates the calculation which eventually puts pressure on the system resources. To address such issues, the authors have proposed an amalgamation of KNN and unsupervised Minimum Spanning Tree.

The system showed to have attained an accuracy of 75%. They have also reduced the size of the training set and assigned weights to different training samples to indicate the degree of importance of each of the sample. The system is rather tied to the directory structure of individual's email settings and more works are needed to make it flexible and usable for different types of email management systems.

Debarr and Wechsler [216] experimented with both supervised and unsupervised frameworks to generate a hybrid model known as Random Boost. The system uses random feature selection to improve upon the performance of the Logit Boost algorithm. Random Boost is more like an extension of RF. Its runtime complexity is around one-fourth one-fourth to that of the RF (with comparable accuracy) and it also reduces the training time of Logit Boost quite significantly.

Meng et al. [33] used Multi-view datasets along with disagreement-based semi-supervised learning to build a framework. Multi-view datasets means to have more than one dataset, composed of different features, but selected from the same data source. The endeavour takes motivation from the fact that there are number of problems in supervised classification which often hinders the practicability of these systems, such as data labelling. Disagreement-based Semi-supervised learning on the other hand is well equipped to handle both labelled and unlabelled data. In this types of semi-supervised learning, multiple learners actively collaborate to analyse a set of unlabelled data; the disagreement among these ''learners'' plays a key role in the final outcome [32]. The proposed method gives an accuracy of a bit over 85%. But as suggested by [32], Disagreement-based Semi-supervised learning, at the current state, is not really safe in the sense that oftentimes the exploitation of unlabelled data may result in adverse effect on the model's performance.

V. AN ANALYTICAL DISSECTION OF THE STUDIES CARRIED OUT AND FUTURE SPAM DETECTION RESEARCH DIRECTIONS

This section will shed lights on some key insights that can be derived from the above critically analyzed studies. We will start with the general, non-AI based spam detection techniques discussed in Section *III*.

A. KEY INSIGHTS FROM GENERAL NON-AUTOMATED SPAM DETECTION FRAMEWORKS

From Fig. 6, we can see how the non-automated anti-spam systems discussed in Section *III* have been designed around different subsections of email infrastructure.

Clearly, email data parts C and D [Fig. 1] have been exploited a lot more than other parts. Thus these systems have relied upon email content, subject, sender and receiver information, while other header features such as IP source and destination along with SMTP transactional fields have been used in a bit lesser degree.

From Table 4 we can see that around 50% of the non-automated techniques depend upon multiple data parts of an email.

B. KEY INSIGHTS FROM MACHINE LEARNING BASED SPAM DETECTION FRAMEWORKS

Based on the information presented in Table 7, a number of useful perceptions can be obtained on the nature and trend of research done on Machine Learning based spam

TABLE 7. A summary of machine learning based techniques.

First_Author - Year	Model Types			Primary Algorithms Used																Result Reported	
	Supervised	Semi-Supervised	Unsupervised	K-means	Naïve Bayes	RBF	J48 DT	MLP	BayesNet	ANN	C4.5 DT	Logistic Regression	AdaBoost	SVM	KNN	Random Forest	CNN	EM	SOM	OTHERS	Accuracy (%)	Others (**P** = Purity. **A** = AUC, **R** = Recall, **F** = FPR)
Smadi - 2018 [15]		♦								•											95.05	
Basavaraju - 2010 [196]			♦	•																•	76	
Laorden - 2014 [197]			♦																	•	92.27	
Padhiyar - 2014 [214]		♦			•										•							--
Roy - 2017 [157]	♦									•				•							92.8	
Nosseir - 2013 [115]	♦									•											99.87	
Aski - 2016 [176]	♦				•			•			•										99	
Shams - 2013 [147]	♦				•								•	•		•				•	94.75	
Islam - 2010 [8]			♦	•																•	96.5	
Amayri - 2012 [156]		♦												•							92	
Nizamani - 2014 [168]	♦				•		•							•							96	
Sharma - 2016 [183]	♦					•								•							99.83	
Saab - 2014 [184]	♦						•			•				•							94.02	
Halder - 2011 [201]			♦	•														•			≈ 80	80% (P)
Bahgat - 2018 [188]	♦				•	•	•							•		•					90	
Debarr - 2012 [216]		♦														•				•		+5% (A)
Bahgat - 2016 [190]	♦				•		•					•		•							96	
Bahgat - 2015 [191]	♦											•		•							97.7	
Pawar - 2015 [150]	♦											•									94.6	
Va'shoven - 2016 [185]	♦				•			•			•			•	•	•					97.69	
Qaroush - 2012 [187]	♦				•						•					•				•	99.27	
Malge - 2016 [117]	♦									•												95% (R)
Sharma - 2016 [162]		♦													•						97.44	
Feng - 2016 [170]	♦				•									•							91.5	
Chakrabarty - 2017 [215]			♦												•						75	
Zhou - 2013 [135]	♦				•																90.05	
Shams - 2013 [175]	♦				•									•		•				•	95.70	
More - 2013 [174]	♦				•					•		•				•					96.87	
Wang - 2010 [164]	♦				•				•		•										94	
Shuaib - 2018 [166]	♦				•	•	•	•				•								•	94.2	
Lee - 2017 [139]	♦				•																95.74	
Basnet - 2008 [27]			♦	•																	90.6	
Wijaya - 2016 [167]	♦										•										91.67	
Ergin - 2014 [180]	♦									•										•	91.08	
Islam - 2013 [171]	♦				•								•	•							96.8	
Hamid - 2011 [172]	♦								•				•			•				•	97	
Seth - 2017 [122]	♦																•				98.11	
Li - 2014 [33]		♦																		•	85	
Sharaff - 2015 [182]	♦						•		•		•			•								--
Varghese - 2017 [159]	♦												•							•		0% (F)
Porras – 2011 [206]			♦																•			--
Cabrera-León - 2016 [207]			♦																•		94.4	
∑	**28**	**6**	**8**	**4**	**16**	**3**	**6**	**3**	**3**	**7**	**6**	**5**	**4**	**14**	**4**	**8**	**1**	**1**	**2**	**12**		

detection techniques since circa 2010. Facts deduced from the sample can decidedly aid in understanding the direction of research that had been conducted over the years. Some of the obtained insights will also indicate potential future research directions.

1) FINDING A: HIGH ADOPTION OF SUPERVISED TECHNIQUES

The PI chart on Fig. 7 demonstrates the high adoption of supervised techniques in developing or benchmarking such anti-spam systems, with 67% of the selected sample.

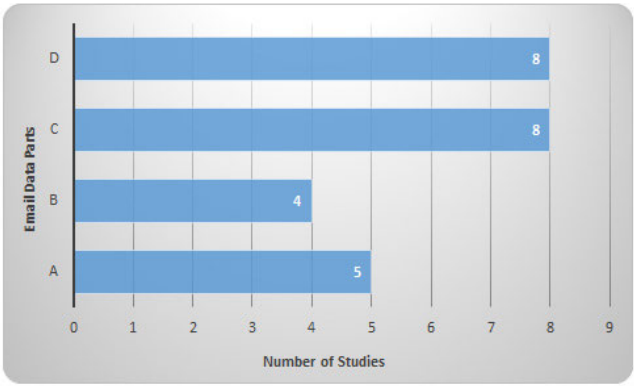

FIGURE 6. Number of non-automated spam detection studies in relation to various email data parts.

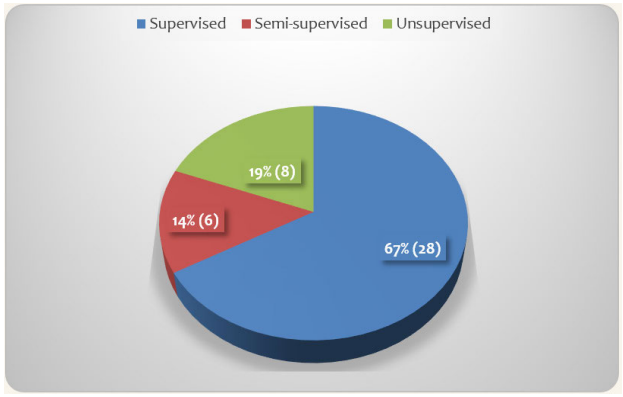

FIGURE 7. Proportion of types of frameworks.

As suggested by Fig. 7, supervised approaches have been the first choice for the researchers and developers, clearly signifying the fact that there is high degree of room and opportunity to expand the research into both semi-supervised, unsupervised and even reinforcement based models.

True form of Artificial Intelligence can only be achieved through non-supervised learning, so there is clearly a need to investigate and develop this wing of Machine Learning for anti-spam systems.

2) FINDING B: PROBABLE REASON BEHIND MARGINAL ADOPTION OF UNSUPERVISED AND SEMI-SUPERVISED ALGORITHMS

The reason for lower adoption rate for Semi-supervised and unsupervised method may be explained through some statistical studies on the outcome (Accuracy in this case) they provide. Although the sample in Table 7 shows disproportionate number of research works for these two types of methods in comparison to supervised learning, but the following Scatterplots and Standard Deviation calculations may shed some light on the underlying cause.

The scatterplots have been laid out in Fig. 8 (supervised), 9 (semi-supervised) and 10 (unsupervised). It can clearly be seen from the three plots that accuracy for supervised learning (Fig. 8) has been within a tightly closed range, with lesser variation; meaning the outcomes are mostly consistent.

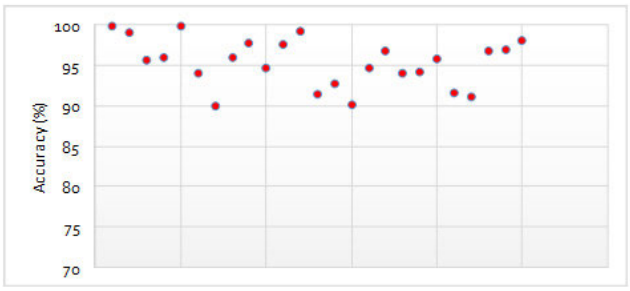

FIGURE 8. Scatterplo for accuracy of supervised methods (sd: 2.97).

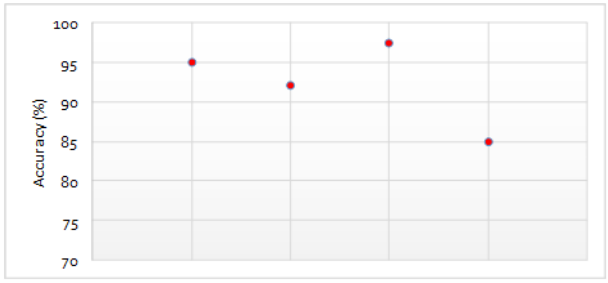

FIGURE 9. Scatterplot for accuracy of semi-supervised methods (sd: 5.39).

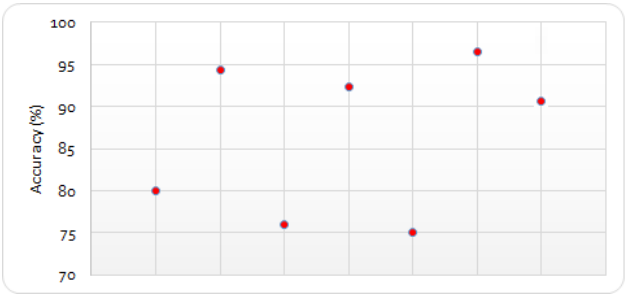

FIGURE 10. Scatterplo for accuracy of unsupervised methods (sd: ≈ 9.20).

The average accuracy is also around min-nineties which is quite acceptable.

On the contrary, scatterplots for semi-supervised (Fig. 9) and unsupervised (Fig. 10) demonstrates the grouping of accuracy is not that tightly maintained; which means the results are not consistent and can vary widely, thus incurs less confidence among the researchers and developers alike. Nonetheless, with the innovation of new algorithms and techniques for unsupervised and semi-supervised methods, such high variance should come down.

The Standard Deviation (SD) for supervised, semi-supervised and unsupervised frameworks have been calculated as 2.97, 5.39 and ≈9.20 respectively from the reported accuracy. The values clearly confirm the findings from the scatterplots.

It is clear from **Finding A** and **B** that there are high degree of opportunities to work with non-supervised spam detection frameworks in a view to bring its performance to a comparable level (or even better) to that of supervised methods, as unsupervised learning do hold few distinct advantages over supervised learning, such as the easier availability of

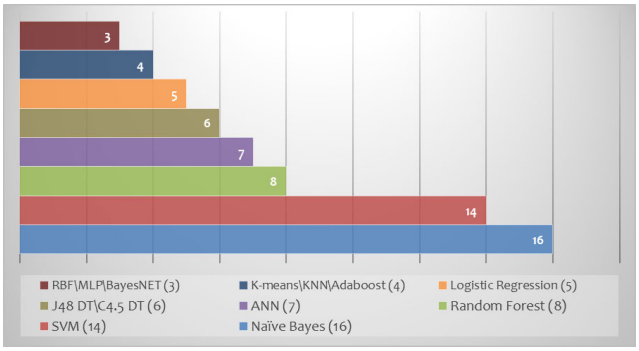

FIGURE 11. Showing the total number of studies in which each of the algorithms has been used (at least 3 studies).

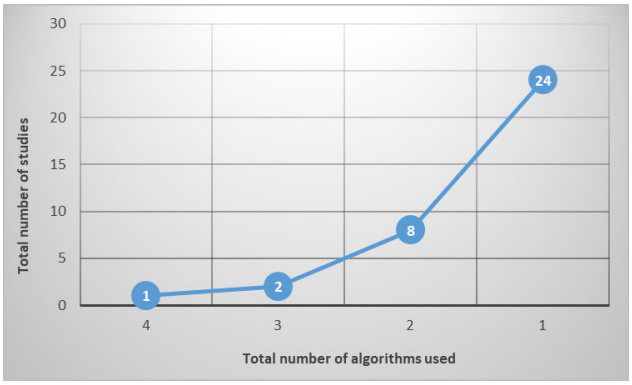

FIGURE 12. Grouping the studies based on the number of algorithms used to build the primary model.

unlabelled data than labelled ones and lesser computational complexity.

3) FINDING C: ALGORITHMIC PREFERENCES

The Bar chart on Fig. 11 primarily illustrates the prevalence of supervised algorithms such as Naïve Bayes and SVM.

It is understood that with the probable rise in unsupervised and semi-supervised system, these number should change as well. Another point to note that along with a more complex and resource-consuming algorithm such as SVM, a simpler and easier algorithm such as Naïve Bayes also has its applications in a wide varieties of settings.

4) FINDING D: PROPORTION OF SINGLE ALGORITHM SYSTEMS AGAINST MULTI-ALGORITHM FRAMEWORKS

Machine Learning based models can have a single algorithm in its core, or multiple algorithms may be used to formulate a working model. A subset of 35 studies from Table 7 has been drawn out to understand the design patterns. Fig. 12 summarizes the finding. In case of sole algorithm systems, oftentimes the study behind these frameworks carried out a comparison analysis using multiple algorithms after the model has been implemented, but the primary framework for the model had been built upon a single Machine Learning algorithm.

As can be seen from Fig.12, number of frameworks based on a single algorithm outnumbers multi-algorithm models conveniently. It is clear that research initiatives on hybrid systems can be something that may need more attention.

In fact, the relation to the two variables, total number of algorithm used (p) and total number of studies undertaken (q) has almost a near-perfect inverse correlation. Using Pearson's correlation coefficient [186], r, as shown in (19), the value of p comes out as -0.91, which clearly points towards the sharp negative correlation between p and q.

$$r = \frac{\sum (p - \overline{p})(q - \overline{q})}{\sqrt{\sum (p - \overline{p})^2 \sum (q - \overline{q})^2}} \tag{19}$$

Thus more research on hybrid systems can be a possible area of future research that can be investigated as indicated by **Finding D**.

5) FINDING E: APPLICATION OF HEADER AND DOMAIN FEATURES

Out of the 58 studies evaluated throughout Section *III.C*, only 9 of those (or ≈14%) have used some form of header or domain features (excluding subject field) while designing spam detection systems using Machine Learning algorithms. Such an observation highlights an opening where more gravity and careful analysis can be applied regarding header, domain and even URL based features of an email. Besides, the frameworks that have used these features, have worked with only limited set of it, and often left out a number of useful ones such as the '*Received from:*' header fields, '*Age of domain*' to name a few.

From **Finding E** we can underscore the fact that the future spam detection frameworks may consider evaluating a number of these useful header, URL and domain features simultaneously to formulate an efficient and effective set of features through appropriate feature engineering.

6) FINDING F: HANDLING OF CONCEPT DRIFT

Again, out of 58 recent research initiatives, only 2 [97], [144], have worked on the issue of Concept Drift with automated principles, which is just ≈1.15%. This highlights a strong research prospect as addressing Concept Drift is something that makes machine learning based filtering systems to stand apart from the traditional static ones.

7) FINDING G: UNDER-ANALYSING THE EMAIL CONTENT

Almost all the studies that work with email content to detect spam emails, especially the phishing ones, rely on word-based clustering or classification models and the degree of closeness of these clusters or classification models to high-probability spam words. The approach is reasonably logical, however, in modern times, the spammers create these phishing emails in the light of several psychological aspects of users' mindset.

In general, a well-crafted phishing email is modelled as closely as possible after the Fig. 13, where the message body may contain words or phrases related to Finance and Personal issues, with 'Subject' header holding phrase that

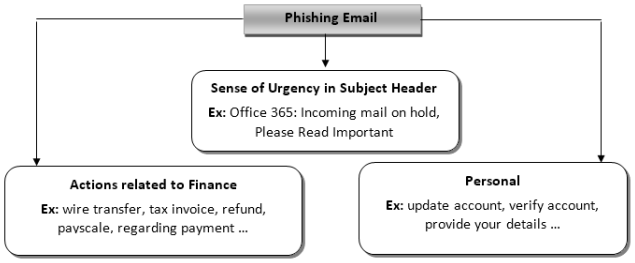

FIGURE 13. Nature of an effective phishing email.

will definitely put the user in a position where he\she will feel the urge to open the email at an earliest instance possible.

Thus for the content analysis to be effective, along with normal word-based analysis, we believe an automated mechanism is also required which will be able to detect, from multiple angles, how closely an email matches to that of the above discussed structure in Fig. 13 before finally labelling it as an instance of phishing email. Such an approach has not been seen in our studies of the modern content based analysis techniques, and we believe more research is required in this area of content analysis (along with the subject header).

Certainly, to be effective against spammers and fraudsters, a Machine Learning based framework, if fully leveraged, should be able to counter all of these key issues as much as possible. Therefore, we believed that the future research should encompass the directions that have been identified in the previous section.

VI. FUTURE WORK

The survey work presented in this paper discussed the types and implication of spam emails on modern society and commerce. A multitude of spam detection frameworks – both Machine Learning based and regular non-automated ones, have been dissected critically to depict a complete picture of the current development and future direction of the field. It is expected that in near future adequate development will branch out in lesser explored arena of Machine Learning based spam identification propositions. It is reasonably clear from the reviews that the currently emerging frameworks, even though using automated machine leaning based solutions, are often not equipped to deal with the multiple angles from which an email spam threat can spread. Thus the future direction of research in this field should be to develop anti-spam software that can simultaneously battle against multiple types of email spamming, considering multiple angles of attack as discussed above, with a single installation of the software.

VII. CONCLUSION

After a thorough analysis, the study results in several different observations especially in the realm of Machine Learning based proposition. It is noted that high adoption of supervised approaches is quite obvious, the reason behind this turns out to be a better consistency in the performance of the model. It has also been highlighted that certain algorithms, such as SVM and Naïve Bayes are in high demand. We have also came into conclusion that single-algorithm anti-spam systems are quite common thus the potentiality of research into hybrid and multi-algorithm systems is quite promising. Besides, research that focusses on email header features excluding the 'subject' field, URLs within the email body and sender domain information need to substantially increase. Another important area that needs increasing attention is the addressing of 'Concept Drift', which would definitely make a system to perform optimally under gradual modification in spamming techniques and motives. In addition, the current way of dealing spam emails of phishing nature is not the most efficient as described, thus requires a more innovative approach that will take into account the different angles of the problem.

A point of concern is that despite several admonishments from multiple bodies, governments of number of leading countries in the world have fell short in forming effective regulations that can really have a lasting impact on this issue [31]. Nevertheless, the actions to strengthen cybersecurity have seen greater gravity in recent times, resulting in the increased research and streamlined availability of funding in this field. Thus it can be expected that a formidable framework, equipped with measures against the drawbacks highlighted in this study, will soon become available for commercial and personal deployment.

REFERENCES

[1] O. Saad, A. Darwish, and R. Faraj, "A survey of machine learning techniques for Spam filtering," *Int. J. Comput. Sci. Netw. Secur.*, vol. 12, no. 2, p. 66, Feb. 2012.
[2] M. K. Paswan, P. S. Bala, and G. Aghila, "Spam filtering: Comparative analysis of filtering techniques," in *Proc. Int. Conf. Adv. Eng., Sci. Manage. (ICAESM)*, Mar. 2012, pp. 170–176.
[3] E. Bauer. *15 Outrageous Email Spam Statistics that Still Ring True in 2018*. Accessed: Jul. 20, 2019. [Online]. Available: https://www.propellercrm.com/blog/email-spam-statistics
[4] Statista. *Number of E-mail Users Worldwide From 2017 to 2023*. Accessed: Jul. 24, 2019. [Online]. Available: https://www.statista.com/Leoni AG loses Ă40m in an email scamstatistics/456519/forecast-number-of-active-email-accounts-worldwide/
[5] Y. Cohen, D. Gordon, and D. Hendler, "Early detection of spamming accounts in large-Scale service provider networks," *Knowl.-Based Syst.*, vol. 142, pp. 241–255, Feb. 2018.
[6] Campaign Monitor. *The Shocking Truth About How Many Emails Are Sent*. Accessed: Jul. 25, 2019. [Online]. Available: https://www.campaignmonitor.com/blog/email-marketing/2018/03/shocking-truth-about-how-many-emails-sent/
[7] O. A. Okunade, "Manipulating E-mail server feedback for spam prevention," *Arid Zone J. Eng., Technol. Environ.*, vol. 13, no. 3, pp. 391–399, Jun. 2017.
[8] R. Islam and Y. Xiang, "Email classification using data reduction method," in *Proc. 5th Int. ICST Conf. Commun. Netw. China*, Aug. 2010, pp. 1–5.
[9] Scamwatch. *Scam Statistics*. Accessed: Jul. 15, 2019. [Online]. Available: https://www.scamwatch.gov.au/about-scamwatch/scam-statistics
[10] Scamwatch. *Scam Statistics*. Accessed: Jul. 16, 2019. [Online]. Available: https://www.scamwatch.gov.au/about-scamwatch/scam-statistics?scamid=31&date=2019
[11] A. Test. *Spam Statistics*. Accessed: Jul. 16, 2019. [Online]. Available: https://www.av-test.org/en/statistics/spam/
[12] N. Banu and M. Banu, "A Comprehensive Study of Phishing Attacks," *Int. J. Comput. Sci. Inf. Technol.*, vol. 4, no. 6, pp. 783–786, 2013.
[13] H. Hu and G. Wang, "Revisiting Email spoofing attacks," 2018, *arXiv:1801.00853*. [Online]. Available: https://arxiv.org/abs/1801.00853

[14] R. A. Halaseh and J. Alqatawna, "Analyzing cybercrimes strategies: The case of phishing attack," in *Proc. Cybersecur. Cyberforensics Conf. (CCC)*, Aug. 2016, pp. 82–88.

[15] 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.

[16] 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. 2011.

[17] A. L. Chung-Man, "An analysis of the impact of phishing and anti-phishing related announcements on market value of global firms," HKU, Hong Kong, Tech. Rep., 2009.

[18] N. Raad, G. Alam, B. Zaidan, and A. Zaidan, "Impact of spam advertisement through E-mail: A study to assess the influence of the anti-spam on the E-mail marketing," *Afr. J. Bus. Manage.*, vol. 4, no. 11, pp. 2362–2367, Sep. 2010.

[19] TheStar. *Company Cheated of RM 4.5 Mil Due to Email Spoofing*. Accessed: Jul. 30, 2019. [Online]. Available: https://www.thestar.com.my/news/nation/2017/06/11/kedah-based-company-cheated-due-to-email-spoofing

[20] T. L. Shan, G. N. Samy, B. Shanmugam, S. Azam, K. C. Yeo, and K. Kannoorpatti, "Heuristic systematic model based guidelines for phishing victims," in *Proc. IEEE Annu. India Conf. (INDICON)*, Dec. 2016, pp. 1–6.

[21] H. M. Al-Mashhadi and M. H. Alabiech, "A survey of Email service: Attacks, security methods and protocols," *Int. J. Comput. Appl.*, vol. 162, no. 11, pp. 31–40, 2017.

[22] J. V. Chandra, N. Challa, and S. K. Pasupuleti, "A practical approach to E-mail spam filters to protect data from advanced persistent threat," in *Proc. Int. Conf. Circuit, Power Comput. Technol. (ICCPCT)*, Mar. 2016, pp. 1–5.

[23] *ABC Bus Company*. Accessed: Apr. 5, 2019. [Online]. Available: https://www.doj.nh.gov/consumer/security-breaches/documents/abc-bus-20180302.pdf

[24] B. B. Gupta, N. A. G. Arachchilage, and K. E. Psannis, "Defending against phishing attacks: Taxonomy of methods, current issues and future directions," *Telecommun. Syst.*, vol. 67, no. 2, pp. 247–267, Feb. 2017.

[25] A. Han. *Leoni AG Loses 40m in an Email Scam*. Accessed: Apr. 17, 2019. [Online]. Available: https://www.bankvaultonline.com/news/security-news/leoni-ag-loses-e40m-in-an-email-scam/

[26] M. J. Schwartz. *French Cinema Chain Fires Dutch Executives Over CEO Fraud*. Accessed: Apr. 17, 2019. [Online]. Available: https://www.bankinfosecurity.com/blogs/french-cinema-chain-fires-dutch-executives-over-ceo-fraud-p-2681

[27] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: A machine learning approach," in *Soft Computing Applications in Industry* (Studies in Fuzziness and Soft Computing), vol. 226. Berlin, Germany: Springer-Verlag, 2008, pp. 373–383.

[28] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[29] M. Alazab, "Profiling and classifying the behavior of malicious codes," *J. Syst. Softw.*, vol. 100, pp. 91–102, Feb. 2015.

[30] V. Gupta. *Understanding Feedforward Neural Networks*. Accessed: Jun. 10, 2019. [Online]. Available: www.learnopencv.com/

[31] M. R. Sánchez, T. T. Loon, and V. Victor, "An anti-spam framework for Singapore," *Media Asia*, vol. 30, no. 4, pp. 240–246, 2003.

[32] Z. Zhou, "Disagreement-based semi-supervised learning," *Acta Automatica Sinica*, vol. 39, no. 11, pp. 1871–1878, 2013, doi: 10.3724/sp.j.1004.2013.01871.

[33] W. Li, W. Meng, Z. Tan, and Y. Xiang, "Towards designing an Email classification system using multi-view based semi-supervised learning," in *Proc. IEEE 13th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Sep. 2014, pp. 174–181.

[34] S. Hameed, T. Kloht, and X. Fu, "Identity based Email sender authentication for spam mitigation," in *Proc. 8th Int. Conf. Digit. Inf. Manage. (ICDIM)*, Sep. 2013, pp. 14–19.

[35] E. Calò. *SPF, DKIM and DMARC Brief Explanation and Best Practices*. Accessed: Feb. 21, 2019. [Online]. Available: https://www.endpoint.com/blog/2014/04/15/spf-dkim-and-dmarc-brief-explanation

[36] L. Seltzer. *DKIM: Useless or Just Disappointing*. Accessed: Mar. 8, 2019. [Online]. Available: www.zdnet.com

[37] A. Karim, "A cryptographic application for secure information transfer in a linux network environment," *Amer. J. Eng. Res.*, vol. 5, no. 8, pp. 266–275, 2016.

[38] D. Sipahi, G. Dalkılıç, and M. H. Özcanhan, "Detecting spam through their sender policy framework records," *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 3555–3563, Dec. 2015.

[39] M. T. Banday, F. A. Mir, J. A. Qadri, and N. A. Shah, "Analyzing Internet E-mail date-spoofing," *Digit. Invest.*, vol. 7, nos. 3–4, pp. 145–153, Apr. 2011.

[40] H. Esquivel, A. Akella, and T. Mori, "On the effectiveness of IP reputation for spam filtering," in *Proc. 2nd Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2010, pp. 1–10.

[41] K. S. Bajaj, F. Egbufor, and J. Pieprzyk, "Critical analysis of spam prevention techniques," in *Proc. 3rd Int. Workshop Secur. Commun. Netw. (IWSCN)*, May 2011, pp. 83–87.

[42] R. R. Roy, "Basic session initiation protocol," in *Handbook on Session Initiation Protocol: Networked Multimedia Communications for IP Telephony*. Boca Raton, FL, USA: CRC Press, 2016, pp. 5–166.

[43] R. Ferdous, "Analysis and protection of SIP based services," Ph.D. dissertation, Dept. Inf. Eng. Comput. Sci., Univ. Trento, Trento, Italy, 2014.

[44] G. Caruana and M. Li, "A survey of emerging approaches to spam filtering," *ACM Comput. Surv.*, vol. 44, no. 2, p. 9, Feb. 2012.

[45] P. Sousa, A. Machado, M. Rocha, P. Cortez, and M. Rio, "A collaborative approach for spam detection," in *Proc. 2nd Int. Conf. Evolving Internet*, Sep. 2010, pp. 92–97.

[46] M. Mojdeh, "Personal Email spam filtering with minimal user interaction," Ph.D. dissertation, Dept. Comput. Sci., Univ. Waterloo, Waterloo, ON, Canada, 2012.

[47] M. Prilepok, P. Berek, J. Platos, and V. Snasel, "Spam detection using data compression and signatures," *Cybern. Syst. Int. J.*, vol. 44, nos. 6–7, pp. 533–549, Mar. 2013.

[48] S. Geerthik and P. Anish, "Filtering spam: Current trends and techniques," *Int. J. Mechtron., Elect. Comput. Technol.*, vol. 3, no. 8, pp. 208–223, Jul. 2013.

[49] E. Damiani, S. D. Capitani, D. Vimercati, S. Paraboschi, and P. Samarati, "An open digest-based technique for spam detection," in *Proc. ISCA 17th Int. Conf. Parallel Distrib. Comput. Syst.*, 2004, pp. 559–564.

[50] B. Biggio, G. Fumera, I. Pillai, and F. Roli, "A survey and experimental evaluation of image spam filtering techniques," *Pattern Recognit. Lett.*, vol. 32, no. 10, pp. 1436–1446, Jul. 2011.

[51] Process Software, LLC. *Explanation of Common Spam Filtering Techniques*. Accessed: Feb. 11, 2019. [Online]. Available: http://www.process.com/products/pmas/whitepapers/explanation_filter_techniques.html

[52] Vernon Schryver. *Distributed Checksum Clearinghouses*. Accessed: Oct. 5, 2019. [Online]. Available: https://www.dcc-servers.net/dcc/

[53] H. Wang, R. Zhou, and Y. Wang, "An anti-spam filtering system based on the naive Bayesian classifier and distributed checksum clearinghouse," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Appl.*, Nov. 2009, pp. 128–131.

[54] J. Chen, R. Fontugne, A. Kato, and K. Fukuda, "Clustering spam campaigns with fuzzy hashing," in *Proc. AINTEC Asian Internet Eng. Conf. (AINTEC)*, Nov. 2014, p. 66.

[55] A. Karim, "Multi-layer masking of character data with a visual image key," *Int. J. Comput. Netw. Inf. Secur.*, vol. 10, no. 10, pp. 41–49, Oct. 2017, doi: 10.5815/ijcnis.2017.10.05.

[56] C.-Y. Chiu, A. Prayoonwong, and Y.-C. Liao, "Learning to index for nearest neighbor search," 2018, *arXiv:1807.02962*. [Online]. Available: https://arxiv.org/abs/1807.02962

[57] Y. Li, S. C. Sundaramurthy, A. G. Bardas, X. Ou, D. Caragea, X. Hu, and J. Jang, "Experimental study of fuzzy hashing in malware clustering analysis," in *Proc. 8th Workshop Cyber Secur. Experimentation Test (CSET)*, Berkeley, CA, USA: USENIX Association, 2015, p. 8.

[58] P. Liu and T.-S. Moh, "Content based spam E-mail filtering," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, Nov. 2016, pp. 218–224.

[59] D. Chiba, M. Akiyama, T. Yagi, K. Hato, T. Mori, and S. Goto, "DomainChroma: Building actionable threat intelligence from malicious domain names," *Comput. Secur.*, vol. 77, pp. 138–161, Aug. 2018.

[60] A. Ramachandran, N. Feamster, and S. Vempala, "Filtering spam with behavioral blacklisting," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, Oct. 2007, pp. 342–351.

[61] Spamhaus. *The Spamhaus Project*. Accessed: Oct. 5, 2019. [Online]. Available: www.spamhaus.org

[62] M. Sirivianos, K. Kim, and X. Yang, "SocialFilter: Introducing social trust to collaborative spam mitigation," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 2300–2308.

[63] P.-C. Lin, P.-H. Lin, P.-R. Chiou, and C.-T. Liu, "Detecting spamming activities by network monitoring with Bloom filters," in *Proc. 15th Int. Conf. Adv. Commun. Technol. (ICACT)*, Jan. 2013, pp. 163–168.
[64] *Bloom Filters*. Accessed: May 15, 2019. [Online]. Available: https://llimllib.github.io
[65] P. Revar, A. Shah, J. Patel, and P. Khanpara, "A review on different types of spam filtering techniques," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, pp. 2720–2723, May/Jun. 2017.
[66] S. Khanna, H. Chaudhry, and G. S. Bindra, "Inbound outbound Email traffic analysis and Its SPAM impact," in *Proc. 4th Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Jul. 2012, pp. 181–186.
[67] L. Ilie, "Regular expression matching," in *Encyclopedia of Algorithms*. New York, NY, USA: Springer-Verlag, 2014, pp. 1–6, doi: 10.1007/978-3-642-27848-8_340-2.
[68] Pettingers. *What's Working Now in Spam Assassin*. Accessed: Feb. 13, 2019. [Online]. Available: http://www.pettingers.org/annoyances/sa-rules.html
[69] S. Khanna, H. Chaudhry, and G. S. Bindra, "Inbound outbound Email traffic analysis and its SPAM impact,"in *Proc. 4th Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Jul. 2012, pp. 181–186.
[70] R. K. Kumar, G. Poonkuzhali, and P. Sudhakar, "Comparative study on Email spam classifier using data mining techniques," in *Proc. Int. Multi Conf. Eng. Comput. Scientists*. vol. 1, Mar. 2012, pp. 14–16.
[71] C. Laorden, I. Santos, B. Sanz, G. Alvarez, and P. G. Bringas, "Word sense disambiguation for spam filtering," *Electron. Commerce Res. Appl.*, vol. 11, no. 3, pp. 290–298, 2012.
[72] D. Kumawat and V. Jain, "POS tagging approaches: A comparison," *Int. J. Comput. Appl.*, vol. 118, no. 6, pp. 32–38, Jan. 2015.
[73] R. M. Ravindran and A. S. Thanamani, "K-means document clustering using vector space model," *Bonfring Int. J. Data Mining*, vol. 5, no. 2, pp. 10–14, Jul. 2015.
[74] H. Che, Q. Liu, L. Zou, H. Yang, D. Zhou, and F. Yu, "A content-based phishing Email detection method," *IEEE Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2017, pp. 415–422.
[75] L. A. Zadeh, "Advances in fuzzy systems—Applications and theory," in *Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers By Lotfi A Zadeh*. Singapore: World Scientific Publishing Co Pte Ltd, 1996, pp. 394–432, doi: 10.1142/9789814261302_0021.
[76] Y. Hu, C. Guo, E. W. T. Ngai, M. Liu, and S. Chen, "A scalable intelligent non-content-based spam-filtering framework," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8557–8565, Dec. 2010.
[77] W. V. Wanrooij and A. Pras, "Filtering spam from bad neighborhoods," *Int. J. Netw. Manage.*, vol. 20, no. 6, pp. 433–444, Nov./Dec. 2010.
[78] G. Stringhini, M. Egele, A. Zarras, T. Holz, C. Kruegel, and G. Vigna, "B@bel: Leveraging Email delivery for spam mitigation," in *Proc. 21st USENIX Conf. Secur. Symp.*, 2012, p. 2.
[79] A. Liska and G. Stowe, "Understanding DNS," in *DNS Security*, 2016, pp. 1–23.
[80] D. Bradbury, "Can we make Email secure," *Netw. Secur.*, vol. 3, no. 3, pp. 13–16, Mar. 2014.
[81] Anon. *Proof of Work*. Accessed: May 27, 2019.[Online]. Available: https://en.bitcoin.it/wiki/Proof_of_work
[82] SpamAssassin. *The Apache SpamAssassin*. Accessed: Oct. 6, 2019. [Online]. Available: https://spamassassin.apache.org/
[83] Capterra. Com. *Anti-Spam Software*. Accessed: Oct. 6, 2019. [Online]. Available: www.capterra.com/anti-spam-software/
[84] ZeroSpam. *The ZeroSpam Solution*. Accessed: Oct. 6, 2019. [Online]. Available: https://www.zerospam.ca/en/home/
[85] A. Darwish, "Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications," *Future Comput. Informat. J.*, vol. 3, no. 2, pp. 231–246, Dec. 2018.
[86] D. Ruano-Ordás, F. Fdez-Riverola, and J. R. Méndez, "Using evolutionary computation for discovering spam patterns from E-mail samples," *Inf. Process. Manage.*, vol. 54, no. 2, pp. 303–317, Mar. 2018.
[87] B. Meadows, P. Riddle, C. Skinner, and M. M. Barley, "Evaluating the seeding genetic algorithm," in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science). Springer, 2013, pp. 221–227.
[88] E. Conrad. *Detecting Spam With Genetic Regular Expressions*. London, U.K.: SANS Institute InfoSec Reading Room, 2007.
[89] I. Idris, A. Selamat, N. T. Nguyen, S. Omatu, O. Krejcar, K. Kuca, and M. Penhaker, "A combined negative selection algorithm-particle swarm optimization for an email spam detection system," *Eng. Appl. Artif. Intell.*, vol. 39, pp. 33–44, Mar. 2015.
[90] I. Idris, A. Selamat, and S. Omatu, "Hybrid email spam detection model with negative selection algorithm and differential evolution," *Eng. Appl. Artif. Intell.*, vol. 28, no. 2, pp. 97–110, Jan. 2014.
[91] A. J. Saleh, A. Karim, B. Shanmugam, S. Azam, K. Kannoorpatti, M. Jonkman, and F. D. Boer, "An intelligent spam detection model based on artificial immune system," *Information*, vol. 10, no. 6, p. 209, Jun. 2019.
[92] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, May 2014, pp. 202–212.
[93] J. Brownlee, *Clever Algorithms: Nature-Inspired Programming Recipes*. Abu Dhabi, United Arab Emirates: LuLu, 2012.
[94] S. S. Aote, M. M. Raghuwanshi, and L. Malik, "A brief review on particle swarm optimization: Limitations & future directions," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 5, pp. 196–200, Sep. 2013.
[95] S. Salhi and N. M. Queen, "A hybrid algorithm for identifying global and local minima when optimizing functions with many minima," *Eur. J. Oper. Res.*, vol. 155, no. 1, pp. 51–67, May 2004.
[96] Y. Zhu and Y. Tan, "Extracting discriminative information from e-mail for spam detection inspired by immune system," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–7.
[97] M. Z. Hayat, J. Basiri, L. Seyedhossein, and A. Shakery, "Content-based concept drift detection for Email spam filtering," in *Proc. 5th Int. Symp. Telecommun.*, Dec. 2010, pp. 531–536.
[98] K. Jackowski, B. Krawczyk, and M. Woźniak, "Application of adaptive splitting and selection classifier to the spam filtering problem," *Cybern. Syst. Int. J.*, vol. 44, nos. 6–7, pp. 569–588, 2013.
[99] D. Wang, D. Irani, and C. Pu, "Is Email business dying?: A study on evolution of Email spam over fifteen years," *EAI Endorsed Trans. Collaborative Comput.*, vol. 1, no. 1, p. e3, May 2014.
[100] R. Sathya and A. Abraham, "Comparison of supervised and unsupervised learning algorithms for pattern classification," *Int. J. Adv. Res. Artif. Intell.*, vol. 2, no. 2, pp. 34–38, Feb. 2013.
[101] F. Qian, A. Pathak, Y. C. Hu, Z. M. Mao, and Y. Xie, "A case for unsupervised-learning-based spam filtering," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, p. 367, Dec. 2010.
[102] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, May 1993, pp. 207–216.
[103] P. H. B. Las-Casas, J. M. Almeida, M. A. Gonzalves, D. Guedes, A. Ziviani, and H. T. Marques-Neto, "Adaptive spammer detection at the source network," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 1434–1439.
[104] X. Zhu, "Semi-supervised learning," in *Encyclopedia of Machine Learning and Data Mining*. New York, NY, USA: Springer, 2010, pp. 1142–1147, doi: 10.1007/978-1-4899-7687-1_749.
[105] K.-L. A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk, "A survey on reinforcement learning models and algorithms for traffic signal control," *ACM Comput. Surv.*, vol. 50, no. 3, p. 34, Oct. 2017.
[106] M. A. Karim, J. Currie, and T.-T. Lie, "Dynamic event detection using a distributed feature selection based machine learning approach in a self-healing microgrid," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 4706–4718, Sep. 2018.
[107] D. Hassan, "On Determining the most effective subset of features for detecting phishing Websites," *Int. J. Comput. Appl.*, vol. 122, no. 20, pp. 1–7, Jan. 2015.
[108] *Feature Extraction Foundations and Applications*, Springer-Verlag, New York, NY, USA, 2016.
[109] O. A. Adewumi and A. A. Akinyelu, "A hybrid firefly and support vector machine classifier for phishing email detection," *Kybernetes*, vol. 45, no. 6, pp. 977–994, Jun. 2016.
[110] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2091–2121, 4th Quart, 2013.
[111] I. Qabajeh and F. Thabtah, "An experimental study for assessing email classification attributes using feature selection methods," in *Proc. 3rd Int. Conf. Adv. Comput. Sci. Appl. Technol.*, Dec. 2014, pp. 125–132.
[112] R. Mohammad, T. L. McCluskey, and T. Fadi, "An assessment of features related to phishing Websites using an automated technique," in *Proc. Int. Conf. Internet Technol. Secured Trans.*, London, U.K., Dec. 2012, pp. 492–497.
[113] A. Yasin and A. Abuhasan, "An intelligent classification model for Phishing Email detection," *Int. J. Netw. Secur. Appl.*, vol. 8, no. 4, pp. 55–72, 2016.

[114] T. Rashid, *Make Your Own Neural Network: A Gentle Journey Through the Mathematics of Neural Networks, and Making Your Own Using the Python Computer Language*. Charleston, SC, USA: CreateSpace, 2017.

[115] A. Nosseir, K. Nagati, and I. Taj-Eddin, "Intelligent word-based spam filter detection using multi-neural networks," *Int. J. Comput. Sci. Issues*, vol. 10, no. 2, p. 17, Mar. 2013.

[116] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[117] A. Malge and S. M. Chaware, "An efficient framework for spam mail detection in attachments using NLP," *Int. J. Sci. Res.*, vol. 5, no. 6, pp. 1121–1125, May 2016.

[118] Y. Zhang, R. Jin, and Z. Zhou, "Understanding bag-of-words model: A statistical framework," *Int. J. Mach. Learn. Cybern.*, vol. 1, nos. 1–4, pp. 43–52, Dec. 2010.

[119] J. Singh and V. Gupta, "Text Stemming: Approaches, applications, and challenges," *ACM Comput. Surv.*, vol. 49, no. 3, p. 45, Dec. 2016.

[120] L. Deng and D. Yu, *Deep Learning: Methods and Applications*. Boston, MA, USA: Now, 2014.

[121] M. F. A. Foysal, M. S. Islam, A. Karim, and N. Neehal, "Shot-Net: A convolutional neural network for classifying different cricket shots," in *Proc. Int. Conf. Recent Trends Image Process. Pattern Recognit.*, 2019, pp. 111–120.

[122] S. Seth and S. Biswas, "Multimodal spam classification using deep learning techniques," in *Proc. 13th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Dec. 2017, pp. 346–349.

[123] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[124] H. Jansma. *Don't Use Dropout in Convolutional Networks—Towards Data Science*. Accessed: Jul. 30, 2019. [Online]. Available: https://towardsdatascience.com/dont-use-dropout-in-convolutional-networks-81486c823c16

[125] E.-X. Shang and H.-G. Zhang, "Image spam classification based on convolutional neural network," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Jul. 2016, pp. 398–403.

[126] A. Barushka and P. Hajek, "Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks," *Appl. Intell.*, vol. 48, no. 10, pp. 3538–3556, Oct. 2018.

[127] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, "Large-scale multi-label text classification—Revisiting neural networks," in *Machine Learning and Knowledge Discovery in Databases* (Lecture Notes in Computer Science), vol. 10534. Cham, Switzerland: Springer, 2014, pp. 437–452.

[128] P. Bermejo, J. A. Gámez, and J. M. Puerta, "Improving the performance of Naive Bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2072–2080, Mar. 2011.

[129] R. T. Nakatsu, "Information visualizations used to avoid the problem of overfitting in supervised machine learning," in *HCI in Business, Government and Organizations. Supporting Business* (Lecture Notes in Computer Science), vol. 10294. Cham, Switzerland: Springer, 2017, pp. 373–385.

[130] T. A. Almeida, J. Almeida, and A. Yamakami, "Spam filtering: How the dimensionality reduction affects the accuracy of Naive Bayes classifiers," *J. Internet Services Appl.*, vol. 1, no. 3, pp. 183–200, Feb. 2010.

[131] L. Melian and A. Nursikuwagus, "Prediction student eligibility in vocation school with Naïve-Byes decision algorithm," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 407, no. 1, 2018, Art. no. 012140.

[132] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for Internet of Things data analysis: A survey," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 161–175, 2018.

[133] C. Bielza and P. Larrañaga, "Discrete Bayesian network classifiers: A survey," *ACM Comput. Surv.*, vol. 47, no. 1, p. 5, Jul. 2014.

[134] S. Manlangit, S. Azam, B. Shanmugam, K. Kannoorpatti, M. Jonkman, and A. Balasubramaniam, "An efficient method for detecting fraudulent transactions using classification algorithms on an anonymized credit card data set," in *Intelligent Systems Design and Applications* (Advances in Intelligent Systems and Computing), vol. 736. Cham, Switzerland: Springer, 2018, pp. 418–429.

[135] B. Zhou, Y. Yao, and J. Luo, "Cost-sensitive three-way Email spam filtering," *J. Intell. Inf. Syst.*, vol. 42, no. 1, pp. 19–45, Feb. 2013.

[136] L. Ting and Y. Qingsong, "Spam feature selection based on the improved mutual information algorithm," in *Proc. 4th Int. Conf. Multimedia Inf. Netw. Secur.*, Nov. 2012, pp. 67–70.

[137] N. Jatana and K. Sharma, "Bayesian spam classification: Time efficient radix encoded fragmented database approach," in *Proc. Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2014, pp. 939–942.

[138] D. Ranganayakulu and C. C, "Detecting malicious URLs in E-mail—An implementation," *AASRI Procedia*, vol. 4, pp. 125–131, Jan. 2013.

[139] C.-N. Lee, Y.-R. Chen, and W.-G. Tzeng, "An online subject-based spam filter using natural language features," in *Proc. IEEE Conf. Dependable Secure Comput.*, Aug. 2017, pp. 479–487.

[140] S. Hegelich, "Decision trees and random forests: Machine learning techniques to classify rare events," *Eur. Policy Anal.*, vol. 2, no. 1, pp. 98–120, 2016.

[141] T. Ouyang, S. Ray, M. Allman, and M. Rabinovich, "A large-scale empirical analysis of email spam detection through network characteristics in a stand-alone enterprise," *Comput. Netw.*, vol. 59, pp. 101–121, Feb. 2014.

[142] *The R Implementation of Rulefit Learning Algorithm*. Accessed: Jun. 17, 2019. [Online]. Available: www.stat.stanford.edu/jhf/R-RuleFit.html

[143] M. Sheikhalishahi, M. Mejri, and N. Tawbi, "Clustering spam Emails into campaigns," in *Proc. 1st Int. Conf. Inf. Syst. Secur. Privacy*, Feb. 2015, pp. 90–97.

[144] J.-J. Sheu, K.-T. Chu, N.-F. Li, and C.-C. Lee, "An efficient incremental learning mechanism for tracking concept drift in spam filtering," *Plos ONE*, vol. 12, no. 2, Sep. 2017, Art. no. e0171518.

[145] K. Fawagreh, M. M. Gaber, and E. Elyan, "Random forests: From early developments to recent advancements," *Syst. Sci. Control Eng.*, vol. 2, no. 1, pp. 602–609, 2014.

[146] K. N. Tran, M. Alazab, and R. Broadhurst, "Towards a feature rich model for predicting spam Emails containing malicious attachments and URLs," in *Proc. 11th Australas. Data Mining Conf. (AusDM)*, 2014, pp. 1–10.

[147] R. Shams and R. E. Mercer, "Classifying spam Emails using text and readability features," in *Proc. IEEE 13th Int. Conf. Data Mining*, Dec. 2013, pp. 657–666.

[148] S. Sperandei, "Understanding logistic regression analysis," *Biochemia Medica*, vol. 24, no. 1, pp. 12–18, 2014.

[149] C. Constantin, "Using the logistic regression model in supporting decisions of establishing marketing strategies," *Bull. Transilvania Univ. Braşov*. vol. 8, no. 2, p. 57, Jul. 2015.

[150] K. Pawar and M. Patil, "Pattern classification under attack on spam filtering," in *Proc. IEEE Int. Conf. Res. Comput. Intell. Commun. Netw. (ICRCICN)*, Nov. 2015, pp. 197–201.

[151] B. Schoelkopf, *Empirical Inference*. Berlin, Germany: Springer-Verlag, 2016.

[152] J. Nayak, B. Naik, and H. S. Behera, "A comprehensive survey on support vector machine in data mining tasks: Applications & challenges," *Int. J. Database Theory Appl.*, vol. 8, no. 1, pp. 169–186, Dec. 2015.

[153] S. Winters-Hilt and S. Merat, "SVM clustering," *BMC Bioinformatics*, vol. 8, no. S7, 2007.

[154] S. Winters-Hilt, "Clustering via support vector machine boosting with simulated annealing," *Int. J. Comput. Optim.*, vol. 4, no. 1, pp. 53–89, 2017.

[155] M. Diale, C. Van Der Walt, T. Celik, and A. Modupe, "Feature selection and support vector machine hyper-parameter optimisation for spam detection," in *Proc. Pattern Recognit. Assoc. South Africa Robot. Mechatronics Int. Conf. (PRASA-RobMech)*, Nov. 2016, pp. 1–7.

[156] O. Amayri and N. Bouguila, "Content-based spam filtering using hybrid generative discriminative learning of both textual and visual features," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2012, pp. 862–865.

[157] S. S. Roy, A. Sinha, R. Roy, C. Barna, and P. Samui, "Spam Email detection using deep support vector machine, support vector machine and artificial neural network," in *Soft Computing Applications* (Advances in Intelligent Systems and Computing). Berlin, Germany: Springer-Verlag, Apr. 2017, pp. 162–174.

[158] R. Wang, "AdaBoost for feature selection, classification and its relation with SVM, A review," *Phys. Procedia*, vol. 25, pp. 800–807, Jan. 2012.

[159] R. Varghese and K. A. Dhanya, "Efficient feature set for spam Email filtering," in *Proc. IEEE 7th Int. Advance Comput. Conf. (IACC)*, Jan. 2017, pp. 732–737.

[160] H. Zuhair, A. Selmat, and M. Salleh, "The Effect of Feature Selection on Phish Website Detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 10, pp. 221–232, 2016.

[161] L.-Y. Hu, M.-W. Huang, S.-W. Ke, and C.-F. Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets," *SpringerPlus*, vol. 5, no. 1, p. 1304, Aug. 2016.

[162] A. Sharma and A. Suryawanshi, "A novel method for detecting spam Email using KNN classification with spearman correlation as distance measure," *Int. J. Comput. Appl.*, vol. 136, no. 6, pp. 28–35, Feb. 2016.
[163] *Spearman's Rank-Order Correlation*. Accessed: Jul. 15, 2019. [Online]. Available: https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php
[164] W. Wang, "Heterogeneous Bayesian ensembles for classifying spam Emails," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–8.
[165] J. Large, J. Lines, and A. Bagnall, "The heterogeneous ensembles of standard classification algorithms (HESCA): The whole is greater than the sum of its parts," 2017, *arXiv:1710.09220*. [Online]. Available: https://arxiv.org/abs/1710.09220
[166] M. Shuaib, O. Osho, I. Ismaila, and J. K. Alhassan, "Comparative analysis of classification algorithms for Email spam detection," *Int. J. Comput. Netw. Inf. Secur.*, vol. 10, no. 1, pp. 60–67, Aug. 2018.
[167] A. Wijaya and A. Bisri, "Hybrid decision tree and logistic regression classifier for Email spam detection," in *Proc. 8th Int. Conf. Inf. Technol. Elect. Eng. (ICITEE)*, Oct. 2016, pp. 1–4.
[168] S. Nizamani, N. Memon, M. Glasdam, and D. D. Nguyen, "Detection of fraudulent emails by employing advanced feature abundance," *Egyptian Informat. J.*, vol. 15, no. 3, pp. 169–174, Nov. 2014.
[169] I. Alsmadi and I. Alhami, "Clustering and classification of email contents," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 27, no. 1, pp. 46–57, Jan. 2015.
[170] W. Feng, J. Sun, L. Zhang, C. Cao, and Q. Yang, "A support vector machine based naive Bayes algorithm for spam filtering," in *Proc. IEEE 35th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2016, pp. 1–8.
[171] R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 324–335, Jan. 2013.
[172] I. R. A. Hamid and J. Abawajy, "Phishing Email feature selection approach," in *Proc. IEEE 10th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Nov. 2011, pp. 916–921.
[173] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. Anti-Phishing Working Groups 2nd Annu. Ecrime Researchers Summit*, vol. 7, Oct. 2007, pp. 60–69.
[174] S. More and S. A. Kulkarni, "Data mining with machine learning applied for email deception," in *Proc. Int. Conf. Opt. Imag. Sensor Secur. (ICOSS)*, Jul. 2013, pp. 1–4, doi: 10.1109/icoiss.2013.6678403.
[175] R. Shams and R. E. Mercer, "Personalized spam filtering with natural language attributes," in *Proc. 12th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*. Miami, FL, USA: IEEE, Dec. 2013, pp. 127–132.
[176] A. S. Aski and N. K. Sourati, "Proposed efficient algorithm to filter spam using machine learning techniques," *Pacific Sci. Rev. A, Natural Sci. Eng.*, vol. 18, no. 2, pp. 145–149, Jul. 2016.
[177] C. Zhang, X. Su, Y. Hu, Z. Zhang, and Y. Deng, "An evidential spam-filtering framework," *Cybern. Syst. Int. J.*, vol. 47, no. 6, pp. 427–444, Jun. 2016.
[178] J. Kukulies and R. H. Schmitt, "Uncertainty-based test planning using dempster-shafer theory of evidence," in *Proc. 2nd Int. Conf. Syst. Rel. Saf. (ICSRS)*, Dec. 2017, pp. 243–249.
[179] R. Sun and Y. Deng, "A new method to determine generalized basic probability assignment in the open world," *IEEE Access*, vol. 7, no. 1, pp. 52827–52835, 2019.
[180] S. Ergin and S. Isik, "The investigation on the effect of feature vector dimension for spam email detection with a new framework," in *Proc. 9th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Jun. 2014, pp. 1–4.
[181] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *J. Mach. Learn. Res.*, vol. 3, pp. 1289–1305, Mar. 2003.
[182] A. Sharaff, N. K. Nagwani, and A. Dhadse, "Comparative study of classification algorithms for spam Email detection," in *Emerging Research in Computing, Information, Communication and Applications*. Singapore: Springer, 2015, pp. 237–244.
[183] R. Sharma and G. Kaur, "E-mail spam detection using SVM and RBF," *Int. J. Modern Edu. Comput. Sci.*, vol. 8, no. 4, pp. 57–63, Apr. 2016.
[184] S. A. Saab, N. Mitri, and M. Awad, "Ham or spam? A comparative study for some content-based classification algorithms for email filtering," in *Proc. 17th IEEE Medit. Electrotech. Conf. (MELECON)*, Apr. 2014, pp. 339–343, doi: 10.1109/melcon.2014.6820574.
[185] F. Vanhoenshoven, G. Nápoles, R. Falcon, K. Vanhoof, and M. Köppen, "Detecting malicious URLs using machine learning techniques," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–8.
[186] P. Sedgwick, "Pearson's correlation coefficient," *BMJ*, vol. 345, Jul. 2012, Art. no. e4483, doi: 10.1136/bmj.e4483.
[187] A. Qaroush, I. M. Khater, and M. Washaha, "Identifying spam e-mail based-on statistical header features and sender behavior," in *Proc. CUBE Int. Inf. Technol. Conf. (CUBE)*, vol. 12, Sep. 2012, pp. 771–778.
[188] E. M. Bahgat, S. Rady, W. Gad, and I. F. Moawad, "Efficient email classification approach based on semantic methods," *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 3259–3269, Dec. 2018.
[189] L. T. Nguyen and K. M. Huynh, "Using WordNet similarity and translations to create Synsets for ontology-based vietnamese WordNet," in *Proc. 5th IIAI Int. Congr. Adv. Appl. Inform. (IIAI-AAI)*, Jul. 2016, pp. 651–656.
[190] E. M. Bahgat and I. F. Moawad, "Semantic-based feature reduction approach for E-mail classification," in *Proc. Int. Conf. Adv. Intell. Syst. Inform.*, 2016, pp. 53–63.
[191] E. M. Bahgat, S. Rady, and W. Gad, "An E-mail filtering approach using classification techniques," in *Proc. 1st Int. Conf. Adv. Intell. Syst. Inform. (AISI)*, Beni Suef, Egypt, Oct. 2015, pp. 321–331.
[192] M. K. Anjali and A. P. Babu, "Ambiguities in natural language processing," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 5, pp. 392–394, 2014.
[193] T. A. Almeida, T. P. Silva, I. Santos, and J. M. G. Hidalgo, "Text normalization and semantic indexing to enhance instant messaging and SMS spam filtering," *Knowl.-Based Syst.*, vol. 108, pp. 25–32, Sep. 2016.
[194] J. R. Méndez, T. R. Cotos-Yañez, and D. Ruano-Ordás, "A new semantic-based feature selection method for spam filtering," *Appl. Soft Comput.*, vol. 76, pp. 89–104, Mar. 2019.
[195] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, and B. D. Satoto, "Integration K-means clustering method and elbow method for identification of the best customer profile cluster," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 336, no. 1, 2018, Art. no. 012017.
[196] M. Basavaraju and D. R. Prabhakar, "A novel method of spam mail detection using text based clustering approach," *Int. J. Comput. Appl.*, vol. 5, no. 4, pp. 15–25, Aug. 2010.
[197] 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.
[198] U. Kutbay, "Partitional clustering," *Recent Applications in Data Clustering*, 2018, doi: 10.5772/intechopen.75836.
[199] R. D. Kortum, "Hyperonyms and hyponyms," in *Varieties of Tone*. New York, NY, USA: Palgrave Macmillan, 2013, pp. 178–180, doi: 10.1057/9781137263544_23.
[200] M. Alazab, R. Layton, R. Broadhurst, and B. Bouhours, "Malicious spam Emails developments and authorship attribution," in *Proc. 4th Cybercrime Trustworthy Comput. Workshop*. Bundoora, Australia: La Trobe University, Nov. 2013, pp. 58–68.
[201] S. Halder, R. Tiwari, and A. Sprague, "Information extraction from spam emails using stylistic and semantic features to identify spammers," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Aug. 2011, pp. 104–107.
[202] K. Xu, "Expectation-maximization algorithm," in *Encyclopedia of Systems Biology*. New York, NY, USA: Springer-Verlag, 2013, doi: 10.1007/978-1-4419-9863-7_449.
[203] J. M. Hancock, "Self-organizing map (Kohonen Map, SOM)," in *Dictionary of Bioinformatics and Computational Biology*. Hoboken, NJ, USA: Wiley, 2004, doi: 10.1002/0471650129.dob0661.
[204] D. I. Kumar and M. R. Kounte, "Comparative study of self-organizing map and deep self-organizing map using MATLAB," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2016, pp. 1020–1023.
[205] A. Azab, R. Layton, M. Alazab, and J. Oliver, "Mining malware to detect variants," in *Proc. 5th Cybercrime Trustworthy Comput. Conf.*, Nov. 2014, pp. 44–53.
[206] S. Porras, B. Baruque, B. Vaquerizo, and E. Corchado, "Clustering ensemble for spam filtering," in *Hybrid Artificial Intelligent Systems* (Lecture Notes in Computer Science). Springer, 2011, pp. 363–372.
[207] Y. Cabrera-León, P. G. Báez, and C. P. Suárez-Araujo, "Self-organizing Maps in the design of anti-spam filters a proposal based on thematic categories," in *Proc. 8th Int. Joint Conf. Comput. Intell.*, 2016, pp. 1–12.
[208] X. Kong, C. Hu, and Z. Duan, "Generalized principal component analysis," in *Principal Component Analysis Networks and Algorithms*. Singapore: Springer, 2017, pp. 185–233.
[209] I. T. Jolliffe, "Principal component analysis," in *Springer Series in Statistics*, vol. 28, 2nd ed. New York, NY, USA: Springer, 2002, p. 487.

[210] I. Dagher and R. Antoun, "Ham-spam filtering using different PCA scenarios," in *Proc. IEEE Intl Conf. Comput. Sci. Eng. (CSE) IEEE Intl Conf. Embedded Ubiquitous Comput. (EUC) 15th Intl Symp. Distrib. Comput. Appl. Bus. Eng. (DCABES)*, Aug. 2016, pp. 542–545.
[211] J. C. Gomez, E. Boiy, and M.-F. Moens, "Highly discriminative statistical features for email classification," *Knowl. Inf. Syst.*, vol. 31, no. 1, pp. 23–53, 2012.
[212] R. Silva, M. A. Gonçalves, and A. Veloso, "Rule-based active sampling for learning to rank," in *Proc. ECML PKDD*, 2011, pp. 240–255.
[213] D. Hassan, "Investigating the effect of combining text clustering with classification on improving spam email detection," in *Intelligent Systems Design and Applications* (Advances in Intelligent Systems and Computing). Cham, Switzerland: Springer, 2017, pp. 99–107, doi: 10.1007/978-3-319-53480-0_10.
[214] H. Padhiyar and P. Rekh, "An improved expectation maximization based semi-supervised email classification using Naive Bayes and k-nearest neighbor," *Int. J. Comput. Appl.*, vol. 101, no. 6, pp. 7–11, Jan. 2014.
[215] A. Chakrabarty and S. Roy, "An optimized k-NN classifier based on minimum spanning tree for email filtering," in *Proc. 2nd Int. Conf. Bus. Inf. Manage. (ICBIM)*, Jan. 2014, pp. 47–52.
[216] D. Debarr and H. Wechsler, "Spam detection using Random Boost," *Pattern Recognit. Lett.*, vol. 33, no. 10, pp. 1237–1244, Jul. 2012.
[217] M. S. Junayed, A. A. Jeny, S. T. Atik, N. Neehal, A. Karim, S. Azam, and B. Shanmugam, "AcneNet—A deep CNN based classification approach for acne classes," in *Proc. 12th Int. Conf. Inf. Commun. Technol. Syst. (ICTS)*, Jul. 2019, pp. 203–208.

ASIF KARIM is currently a Ph.D. Researcher with Charles Darwin University, Australia. His research interests include machine intelligence and cryptographic communication. He is currently working toward the development of a robust and advanced email filtering system, primarily using machine learning algorithms. He has considerable industry experience in IT, primarily in the field of software engineering.

SAMI AZAM is currently a leading Researcher and a Lecturer with the College of Engineering and IT, Charles Darwin University, Australia. He is actively involved in the research fields relating to computer vision, signal processing, artificial intelligence, and biomedical engineering. He has a number of publications in peer-reviewed journals and international conference proceedings.

BHARANIDHARAN SHANMUGAM is currently a research-intensive Lecturer with the College of Engineering and IT, Charles Darwin University, Australia. He has a large number of publications in several different journals and conference proceedings. His research interest mainly revolves around the field of cybersecurity.

KRISHNAN KANNOORPATTI is currently a Research Active Associate Professor with the College of Engineering, IT and Environment, Charles Darwin University, Australia. In addition of being a stellar academic and innovative researcher, he also has an extensive experience of working with the government bodies in setting up data privacy policies at national and state level.

MAMOUN ALAZAB is currently an Associate Professor with the College of Engineering, IT and Environment, Charles Darwin University, Australia. He is also a cyber-security researcher and practitioner with industry and academic experience. His research is multidisciplinary that focuses on cyber security and digital forensics of computer systems, including current and emerging issues in the cyber environment like cyber-physical systems and the Internet of Things (IoT), by taking into consideration the unique challenges present in these environments, with a focus on cybercrime detection and prevention.