SMU Data Science Review

Volume 6
Number 2 *Summer 2022*

Article 14

2022

Phishing Detection Using Natural Language Processing and Machine Learning

Apurv Mittal
Southern Methodist University, apurvmittal@gmail.com

Dr Daniel Engels
dwe@alum.mit.edu

Harsha Kommanapalli
harshanaidu@yahoo.com

Ravi Sivaraman
Southern Methodist University, rsivaraman@smu.edu

Taifur Chowdhury
Southern Methodist University, tchowdhury@smu.edu

Phishing Detection Using Natural Language Processing and Machine Learning

Tai Chowdhury[1], Ravi Sivaraman[1], Apurv Mittal[1], Daniel W. Engels[2], Harsha Kommanapalli[3]

[1] Master of Science in Data Science
Southern Methodist University
Dallas, TX 75275 USA
[2] AT&T Virtualization Center, SMU, Dallas, TX
[3] Meta, Menlo Park, CA
[1] {tchowdhury, apurvm, rsivaraman}@smu.edu
[2] dwe@alum.mit.edu
[3] harshanaidu@yahoo.com

Abstract. Phishing emails are a primary mode of entry for attackers into an organization. A successful phishing attempt leads to unauthorized access to sensitive information and systems. However, automatically identifying phishing emails is often difficult since many phishing emails have composite features such as body text and metadata that are nearly indistinguishable from valid emails. This paper presents a novel machine learning-based framework, the DARTH framework, that characterizes and combines multiple models, with one model for each composite feature, that enables the accurate identification of phishing emails. The framework analyses each composite feature independently utilizing a multi-faceted approach using Natural Language Processing (NLP) and neural network-based techniques and combines the results of these analyses to classify the emails as malicious or legitimate. Utilizing the framework on more than 150,000 emails and training data from multiple sources, including the authors' emails and phishtank.com, resulted in the precision (correct identification of malicious observations to the total prediction of malicious observations) of 99.97% with an f-score of 99.98% and accurately identifying phishing emails 99.98% of the time. Utilizing multiple machine learning techniques combined in an ensemble approach across a range of composite features yields highly accurate identification of phishing emails.

1 Introduction

Phishing is a method of stealing private and sensitive information using deceptive emails, websites, and text messages. The attackers utilize social engineering approaches to entice people to perform actions, such as clicking on a hyperlink, that leads to malware installation or stealing personal information. To this end, attackers often pretend to be someone from a reputable organization and use fraudulent techniques to steal online users' data, such as passwords or credit card information. The

improvements in cybersecurity protections, to the point, that humans are the weakest link in the cybersecurity chain have been attributed to the advancement of social engineering attacks such as phishing. According to research from IRONSCALES (2021), 81% of the organization that participated in the survey have dealt with phishing attacks [14]. According to Verizon's 2021 Data Breach Investigations Report (DBIR), around 25% of all data breaches involve phishing, and 85% involve a human element [27]. The rise of social media has complicated the issue even further as the attackers use sophisticated tools to carry out these attacks. Hackers use LinkedIn to create faux messages, making up 47% of social media phishing attempts [14]. The cost of the data breach on LinkedIn alone was $4.42 Billion in 2021 [14]. This affects individuals to organizations; it has both privacy and financial implications.

Anti-Phishing Working Group (APWG) is an organization that collects, analyzes, and exchanges a list of credential URLs. It publishes a quarterly report on phishing activities across the globe. The number of phishing attacks has doubled from 2020 to 2021. More than 260,000 reported phishing attacks in July 2021 [41]. Webmail continues to be among the top methods of phishing attempts. There has been an increase in phishing attacks on named brands, from 400 per July to 700 in September 2021. There are state laws for penalizing criminals for phishing attacks. Anti-phishing Act of 2005 imposes fines and imprisonment for up to five years or both for a person involved with phishing attacks [26]. California leads the way in having a strong state law on phishing attacks.

Currently, most phishing attack detection methods are purely one-method approaches. This type of method may not be effective in detecting sophisticated phishing attacks. Most experts use two types of phishing detection systems, list-based detection systems and Machine Learning-Based Detection Systems [6]. A blocklist of URLs is created in a list-based system to identify malicious links using URL metadata gathered from phishing detection systems, user notifications, third-party organizations, and other cybersecurity platforms. Blacklist-based methods have a low false-positive rate compared to machine learning-based approaches. The success rate is about 20% [6]. This method requires constant URL updates to the blocklist database, worsening the problem.

Recent research on phishing detection focuses on machine learning techniques like Artificial Neural Networks (ANN) [9], Bayesian Additive Regression Trees (BART) [6], Graph Convolutional Networks (GCN), and Natural Language Processing (NLP) [8] for feature detection like various attributes in the observed dataset. These research papers primarily focused on URL metadata, with few analyzing the email texts [9]. Several previous research papers focused mainly either on URL metadata or email texts. This creates an opportunity to research and create a holistic model which inherits multiple techniques on various aspects of malicious emails like URLs, attachments, images, senders, body text, etc., to identify phishing attempts effectively.

This research employs various modeling techniques to detect sophisticated phishing attacks, including bagging and boosting modeling techniques. One of the significant challenges of phishing detection is the preprocessing of text in the URLs and email body. Boosting techniques like Extreme Gradient Boosting (XGBoost) handles large dataset for text preprocessing, extract essential features, and handle noise properly for phishing classification. In another industry research, Support Vector Machine (SVM) has been used for phishing classifications as it can combine statistical

framework and other combinations such as user behavior features to create a model that can yield accuracy scores of >97% [22].

This research uses Natural Language Processing (NLP) techniques, clustering, and neural network-based machine learning models to identify phishing attempts by analyzing the email content before users access it. This research recommends a set of processes rather than relying on a single method to address sophisticated attacks. It targets phishing attempts holistically by using a multi-faceted approach that analyzes the embedded URLs, email body, sender's information, email attachments, and other email metadata to classify malicious emails. This research brings incremental improvements to the existing models. Institutions and researchers interested in the security of email communication can use the output from this multi-faceted approach.

This study analyzes English language body text and assigns scores based on the text characteristics persuading users to access the malicious content. The phishing email text is classified into two major categories, "Masquerade-ness" and "Urgent-ness." "Masquerade-ness" is a phishing email characteristic that urges the receiver to click the URLs with less analytical thought. To aid such behavior, such emails masquerade themselves as a famous brand through phony advertising attractive to the receiver. This masquerading behavior is measured from NLP analytics using Sentence Vectors.

Similarly, "Urgent-ness" is a phishing email characteristic that urges receivers to access the malicious content by creating a false sense of urgency (like the receiver needs to click now to get the deal, etc.). This "Urgent-ness" from the email text is measured using Sentence Vectors. These sentence vectors are fed to neural networks as features to detect phishing emails.

In addition to NLP, this research uses Neural Network modeling techniques to detect accuracy improvements. This technique performs better for sophisticated attacks in which blacklisting, heuristic detection, and visual similarity methods do not perform well in terms of detection [18]. Current techniques require more manual processes and human intervention, which becomes inefficient for faster detection of sophisticated attacks. Zhu et al. (2020) mention that these methods allow attackers to cut through the constricted filters and rules. Neural Network models can address these problems by using robust historical datasets to create a model that reduces manual inputs for phishing detection. There are several types of modeling techniques.

Feed Forward Neural Network (FFNN), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Recurrent Neuron Network (RNN), and Ensemble Neural Network (ENN) are some of the crucial neural networks for models that have been used for phishing websites and email detection [17]. ANN is a neural network model, which is a self-structured neural network. It mimics the human brain's neural network, where several neurons or hidden layers are connected to pass information from the input layer to the output layer. This model has been highly used for URL-based phishing detection as it provides high accuracy scores [9]. FFNN is another popular neural network model. Soon et al. (2020) have mentioned the increased usage of FFNN since it has a history of producing accuracy scores of 95% or up [17]. It helps create an effective modeling relationship between input layers and output layers through feedforward neural networks [18]. ENN is another powerful modeling technique that gathers multiple neural network models to detect attacks using covariance matrices. The matrices are calculated by collecting the output's average, maximum, and minimum values and providing the final score using majority votes [18].

CNN modeling techniques can deal with some of the complex issues with new and sophisticated phishing detection. It is a fully connected artificial neural network that can read images and handwritten data for image detection. It consists of several coevolutionary, max-pooling, or fully connected layers [20]. Coevolutionary layers can detect "chrematistic features" in images [20]. These layers can help see phishing attacks by analyzing URLs.

The model improves detection by adding more embedded layers. The model also performs well with NLP, where it can classify the attacks with a higher accuracy score by adjusting the representation of words in URLs [20]. And lastly, there is the Recurrent Neural Network (RNN) which uses sequential data to predict words or speech in language translation and detection. RNN takes characters from URLs as input and sequentially analyze them for each URL to study pattern for attack detection. The classification model is built using Least Square Time Series.

The data for this study has been acquired from PhishTank.com, Mendeley Data [15], and NapierOne [16]. Phishtank has datasets that break down URLs into different features that detect malicious emails. The data is confirmed phishing attempts, gathered collaboratively by the registered users, and further reviewed by PhishTank operated by *Cisco Talos Intelligence Group*. Mendeley Data is the dataset Hannousse et al. (2021) prepared with confirmed malicious and legitimate URLs with their domain and sub-domain classifications [15]. NapierOne provides a dataset of documents often sent as attachments with malicious contents. NapierOne is managed by the School of Computing at Edinburgh Napier University [16].

This paper presents the DARTH framework, a novel, multi-faceted solution to the email phishing detection problem. DARTH deconstructs an email by the email composite features such as body text and metadata that are nearly indistinguishable from valid emails. Each composite feature is analyzed by its respective neural network model, and an Ensemble Neural Network (ENN) utilizes the output of these models to determine phishing classification. The exemplary multi-faceted DARTH method presented in this paper uses the following composite features: email body text, the entropy of attached files, metadata of email, and embedded URLs contained anywhere within the email.

2 Literature Review

Traditionally, phishing detection research has focused on methods for automated phishing detection. This section presents related work covering different aspects of phishing detection. This section begins with a brief history of phishing and an overview of the most common phishing detection methods. Researchers have tackled this problem differently over time. Some researchers have focused on machine learning models, while others have focused on manual add-ins and natural language processing elements on email text.

2.1 Origin and Types of Phishing

As defined in Merriam-Webster, phishing is "the practice of tricking Internet users into revealing personal or confidential information which can then be used illicitly" [**Error! Reference source not found.**].

Table 1: Types of Phishing and their brief description

Phishing Type	Description
Standard Phishing	Stealing sensitive information by pretending to be an authorized person or an organization. It is not a targeted attack and can be conducted for a large group or for a mass attack.
Malware Phishing	It introduces bugs/viruses into the victim's machine and network by convincing a user to click a link or download an attachment to install the malware. It is currently one of the most widely used form of a phishing attack.
Spear Phishing	In contrast to the standard phishing where many users are attacked at once, spear-phishing is a targeted attack towards a big target like CEOs, Celebrities, etc. This requires intense research of the potential victim to convince them into engaging with the scam.
Smishing	SMS + Phishing = SMISHING. In this type of attacks, the SMS or text messages are used to deliver the malicious links to the unsuspecting user. The links are often short of the actual URLs.
Search Engine Phishing	In this technique, the fraudulent sites are injected into the search results often in the form of paid ads.
Vishing	Vishing is a method where a hacker contacts the user over a phone call pretending to be from a known organization and tries to extract the sensitive financial information from the user like banking and credit card details.
Pharming	It is a technically sophisticated form of phishing using the internet's domain name system (DNS). Pharming reroutes legitimate web traffic to a spoofed page without the user's knowledge, often to steal valuable information.
Clone Phishing	In clone phishing attackers make changes to an existing email, resulting in a nearly identical (cloned) email but with malicious URLs and attachments. This requires a compromise of an email account.
Man-In-The-Middle (MITM)	Man-In-The-Middle (MITM) attack is when a hacker eavesdrops into conversation among the two or more individuals. Hackers create a public Wi-Fi network which unsuspecting users join allowing the attackers to capture information and transmit incorrect information including malware to the involved parties.
Business Email Compromise	Business Email Compromise (BEC) involves a phony email usually claiming to be an urgent request for payment or purchase from someone within or associated with a target's company.
Malvertising	In case of Malvertising the attackers post a malicious advertisement on the legitimate websites. The animation or video or links within the

	advertisement has links to the malicious software to steal information from the users.

The term "phishing" was coined by a then-teenager named Koceilah Rekouche [32]. Rekouche developed the first phishing attack. With a small group of teenagers, Rekouche developed the AOHell software designed to steal the passwords of America Online (AOL) users [32]. It was arguably the first phishing software, and it was used for stealing passwords and credit card information beginning in January 1995. AOHell's phishing system was made publicly available, its release leading to many other automated phishing systems over the years [32].

Started by teenagers and adopted by several other amateurs, phishing activity spread from AOL to other networks. Slowly, professional criminals took notice of this phishing activity and got involved in phishing schemes. Although phishing started small, it became one of the major cyber security threats worldwide, leading to significant financial losses to individuals, corporations, and even governments [32].

Phishing, which started as a very basic technology, soon became sophisticated methodical attacks. As organizations began building algorithms to identify phishing attempts, hackers continued to invent new ways to evade the detection. Phishing attackers have constantly developed new techniques to hide their phishing attacks like Smishing, Spear phishing, Malware phishing, and Malvertising.

Humans are the weakest link in the phishing scheme as they can be easily manipulated for information or duped into clicking on malicious links via social engineering techniques.

2.2 URL-Based Phishing Detection

Past studies have used methods of detecting phishing attacks using URLs. Dutta et al. (2021) mention that phishing techniques are mainly classified as technical subterfuge and Social Engineering. Technical subterfuge such as Keylogging DNS poisoning uses a tool to attack, while social engineering such as Spear phishing whaling tricks victims into accessing a compromised URL [2]. Dutta et al. (2021) evaluate the detection of social engineering phishing attempts delivered via email. Haynes et al. (2021) propose a lightweight phishing detection system to identify phishing URLs. They have used NLP transformers and applied ANN (Artificial Neural Networks) [9]. Haynes et al. (2021) suggest that the models may predict if the website is phishing or not, just using the texts in the URL by applying transformers on the texts. Haynes et al. (2021) propose this idea to improve the speed of creating and validating models as an edge compared to other phishing techniques [9]. While Haynes et al. (2021) have provided groundbreaking works for the Neural Network Modeling technique. There is an opportunity to use the Neural Network modeling technique for email body text-based Natural Language Processing.

Existing phishing techniques are based on source code that scrapes web pages' content. Machine learning techniques require essential manual feature engineering and do not detect new phishing offenses effectively. Aljofey et al. (2020), in their research on *Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL,* introduced a deep learning model that uses a convolutional neural network (CNN) to evaluate the URLs of the websites to identify malicious sites and

potential for phishing. It captures the sequential pattern of URL strings without prior knowledge about phishing and uses the sequential features for faster classification. For performance metrics, the model accuracy is compared with traditional models using hand-crafted character embedding, character level TF-IDF, and character level count vector features [13]. Using the convolutional neural network model and character level TF-IDF analysis is crucial for this study as both techniques are important supplementary methods to build the entire process. Each of the previous studies has investigated both ways individually. This study includes both approaches to create a set of sophisticated phishing attack detection techniques.

In traditional machine learning techniques, website URLs are first analyzed with different hand-crafted features to improve detection accuracy. URLs are analyzed to perform feature adaptation from phishing websites. Using these features, the engineers constructed the training set using labeled features. On the other hand, the convolutional neural network (CNN) model requires less human effect. It identifies individual characters from URLs based on prescribed character vocabulary and then represents each character as a fixed-length vector using one-hot encoding [13]. The model identifies similar characters that can be unnoticeable in website URLs. One of the significant advantages of this model is that it does not have to rely on third-party services for detection. The study provides a unique modeling technique that can aid the build the new proposed set of models. However, the study has not investigated multiple modeling techniques or included any Natural Language Processing techniques.

A Recurrent Neural Network (RNN) is a neural network that uses sequential or time-series data for language translation, natural language processing, speech detection, and image detection. All parameters are the same across each hidden layer, meaning weights are the same on each node. These features have been helpful for the precise detection of phishing attacks. Neeharika et al. (2021) have conducted a study that found that this neural network model has a high accuracy rate. Also, they did not have to create extra features for model building [21]. The model takes characters from sequentially listed URLs and predicts whether the URL is part of a phishing attack or not using Least Squares Time Series units [21]. This research can be convenient for the proposed study because of its performance and feature of eliminating manual inputs for feature creation. The proposed research also plans to explore the email body text, and RNN has the potential for detecting a particular set of sequence data that may be linked to phishing attacks.

Ensemble Neural Network (ENN) is a method where many neural networks are used to solve a problem. Multiple neural networks, regression, and classification neural networks are analyzed. Findings reveal that numerous neural network ensembles are a better fit. The optimization process uses covariance matrices calculated by the maximum likelihood algorithm under the Bayesian framework. The network does not calculate the gradient, which allows it to utilize complicated neural models and loss functions [29]. The appropriate networks are selected from the available set of neural networks to achieve an effective ensemble, using an approach called Genetic Algorithm-based Selective Neural Network Ensemble (GASEN). GASEN term proposed by Zhi-Hua et al. (2002), trains neural networks, assigns random weights to the networks, evolves, and employs a genetic algorithm to find the better fit among available networks. The study by Zhi-Hua et al. (2002) shows that compared to bagging and boosting, ENN can create a better neural network with smaller sizes [28].

Soon et al. (2020) have used ENN, RNN, and FFNN on phishing datasets and have produced several highly accurate models using different hyperparameters. The researchers have run all three models using two scenarios to improve accuracy detection. In the first scenario, all three models have been executed using a range of 1-18 input layers. All the models have been performed for the second scenario using a 0.001 – 0.1 learning rate. The final experiment shows that ENN has produced a better accuracy score than RNN and FFNN. The researchers have concluded that ENN requires fewer neurons than the other two models. A lower learning rate produces better results for phishing detection due to its ability to reduce error [17].

Various Machine Learning techniques have been used to identify phishing attempts. Dharani et al. (2021) proposed using Machine Learning Methods such as Random Forest Algorithm and Extreme Gradient Boosting (XGBoost) Algorithm for efficient and accurate phishing website detection on its Uniform Resource Locator [5]. Most research focused on identifying phishing attempts by evaluating the URLs rather than the email content and patterns. Another similar study, Akinyelu et al. (2014), in the paper *Classification of Phishing Email Using Random Forest Machine Learning Technique,* mentions that most tools and techniques are used to flag emails to identify phishing emails. In contrast, phishing detection tools are not standard [3]. Most phishing detection techniques involve scanning URLs through block-listed [4] sets of URLs previously flagged as malicious. While these studies present essential aspects of the Random Forest classification modeling technique for the URL portion of the emailing system, these studies are based on one method that can be ineffective for detecting early phishing. Prakash et al. compare multiple modeling techniques and Random Forest to build the optimal machine learning model. This research study also investigates other avenues of the emailing system, such as the text body of the email processing using Natural Language Processing.

Abu-Nimeh et al. (2009*)*, in their work, *Distributed Phishing Detection by Applying Variable Selection Using Bayesian Additive Regression Trees,* focused on detecting phishing emails on mobile devices. They have used distributed detection techniques applying variable selection using Bayesian additive regression trees. The study notes that BART improves accuracy when combined with other machine learning classifiers. The study concludes that future work is necessary. However, BART can be a tool to improve accuracy [6].

2.3 Natural Language Processing on Text

While other studies have focused on a single method*,* Ramanathan, & Wechsler, H. (2012) propose a multi-layered methodology called phishGILLNET. The researchers applied the methods three times: Fisher similarity, second Adaboost, and third used NLP techniques on misspelled words to identify phishing [10]. Ramanathan et al. used a large dataset of public corpus emails (about 400,000) to conduct the study and noted outstanding results. 10 The paper mentioned those social media users, such as Internet Messages, chat, blog posts, etc., could apply the phishGILLNET methods [10].

Attackers continually evolve their methods to evade advances in protection and exploit newly discovered vulnerabilities or events. The current anti-phishing products use a combination of blocklist, heuristic, visual, and machine learning to detect the attacks. Sahingoz et al. (2019) promote using classification algorithms and natural

language processing base features to see malicious links and emails in real-time rather than from a list of databases. Experiments used a newly constructed test dataset utilizing the Random Forrest model with NLP-based features that created an accuracy rate of 99.98% [11]. The study can be helpful as it combines machine learning techniques with Natural Language Processing. However, it still lacks research on how an integrated approach can improve phishing detection using URLs and text body text.

Since phishing detection is a classification case, Sahingoz et al. (2019) deploy a model which extracts keywords using the "frequency-inverse document" algorithm. The drawback of this technique is that the model is helpful with the English language. Also, it tends to produce many false positives, although the model has a high accuracy rate. The model determines website legitimacy, detects possible target domains using a search engine, and determines whether the domains in the query are legitimate or not. It can also study offline websites using a support vector machine. The Adaptive Regularization of Weights algorithm is used for fraud detection on online websites. These are all non-linear approaches for detecting phishing attacks [11].

Sanglerdsinlapachai et al. (2010) have added a new dimension to the literature by focusing on domain top page similarity. Their research, *"Using Domain Top-Page Similarity Feature in Machine Learning-Based Web Phishing Detection,"* explored domain top page similarity to detect any new phishing websites. Sanglerdsinlapachai et al. (2010) note the high success rate of detecting phishing, though the samples were tiny [7]. On the other hand, Abbasi et al. (2021) argue that the root cause of the problem is the internet users' lack of ability to identify malicious emails or products. The study introduced the phishing funnel model (PFM). It is a design artifact that predicts users' susceptibility to phishing outlets such as websites, emails, etc. [12]. For the target variable, the research focused on user behavior regarding interaction with phishing attacks instead of predicting whether the links of websites or emails are related to phishing attacks. Using over 1,200 employees and around 49,000 phishing interactions, the model has outperformed other existing models, reducing the number of phishing attacks as it made users classify incoming emails and attachments as malicious items [12].

Despite the large sample size, the question is, can we avoid human errors by training machines to identify phishing. Alhogail & Alsabih et al. (2021) research seemed to look for the answer. Alhogail & Alsabih et al. (2021) have emphasized the importance of using machine learning methods to detect phishing instead of relying on humans. In their studies on *Applying machine learning and natural language processing to detect phishing emails,* they propose a deep learning method using Graph convolutional network (GCN) and natural language processing on the email body text. The method is more efficient at detecting zero-day phishing emails than other methods [8]. However, the Alhogail & Alsabih et al. (2021) concluded that more study is necessary to confirm the findings.

Regarding PFM, the proposed research will address some of the gaps that need to be addressed. Alhogail & Alsabih et al. (2021) analysis brings up three research gaps. First, prior works have not investigated the details of user behavior as a target variable. Second, previous studies have focused on "single decision," such as binary classification of the malicious or non-malicious status. Third, prior models did not emphasize tools much when studying users' susceptibility to attacks [12]. This research paper addresses the second gap in binary classification's "single decision."

2.4 Email Attachment Phishing Detection

Cybercriminals target users by sending malicious attachments through emails. The attachments are sent through different file formats such as pdf, SVG, XML, JSON, etc. Users often download the files by mistake, containing malware that installs automatically on their devices. This allows the attackers to gain personal information through fraudulent activities such as transferring money from victims' banks, stealing trade secrets from organizations, and even threatening individuals for different motives. Machine learning models can be instrumental in detecting attacks through attachments. Akinyelu et al. (2014) have studied branches in emails using Random Forest models to improve the accuracy rate for phishing detection. This technique has produced an accuracy rate of 99.7% compared to other machine learning models that have made 97% [3].

Phishing attacks lure individuals to access malicious email content, including attachments and links. Attackers often use one or more of the phishing techniques listed in Table 2 to persuade users to access the information by downloading an attachment (malware) or simply clicking on a link that installs malware into the victim's system [35].

Table 2. Attachment-based phishing techniques [35]

Techniques	Description
Authority	Attackers claim to be someone from reputable organizations to ask for the victim's information
Urgency	Attackers ask victims to respond to a claim in an urgent manner
Reciprocity	Attackers claim to favor victims using stated service
Social Proof	Attackers try to gain information by saying others have responded to the claim.
Reward	Attackers offer a reward to the victims for a response.
Loss	Attackers claim victims will deal with some form of loss if they don't respond.
Scarcity	Attackers offer a limited amount of opportunity to the victims such as a claim for the first 20 responders.

Using Table 2, Williams et al. (2018) mentioned two theoretical frameworks that have been applied to detect these attacks. The Suspicion, Cognition, and Automaticity Model (SCAM) is a theory that takes a company's users' knowledge, beliefs, and habits to analyze users' susceptibility to phishing attacks [35]. Protection Motivation Theory (PMT) is a theory that has been applied to generic security behavior to understand users' perceptions of these types of attacks [35]. For the research, Williams et al. (2018) have created two hypotheses – 1) users will respond to authority-based attacks, and 2) will respond to urgency-based attacks. Given the results from the statistical modeling, both hypotheses have been validated with a z score of 72.68, $p < 0.001$ for hypothesis 1, and a z score of 39.12, $p < 0.001$ for hypothesis 2.

2.5 Moving to a Multi-Faceted Approach

Prior studies are highly based on analyzing URLs and body text separately[11, 5]. Abbasi et al. (2021) introduced a phishing detection study using user behavior only [12]. These phishing detection methods have been primarily unidirectional since the Abbasi et al. have analyzed phishing detection of emails using unique composite features along with NLP or machine learning model separately. For example, Dharani et al. (2021) have used the non-neural network model XGBoost on text data from URLs only [5]. Furthermore, others, such as Aljofey et al. (2020), have used neural network models such as Convolutional Neural Network on URL data [13]. These studies are primarily unidirectional as traditional research have used one feature and one technique to analyze phishing attacks. Few studies have studied email features using combination techniques such as NLP transformation and Neural Networks. Haynes et al. (2021) used NLP for URL text preprocessing and Artificial Neural Network for phishing attack detection [9]. Sahingoz et al. (2019) have analyzed the emails' body text using NLP preprocessing and random forest modeling techniques for phishing attack detection [11]. Although these simplistic multi-faceted studies have added more tools to the algorithm for phishing attack detection, this research still lacks a comprehensive algorithm design that includes all the composite features of email, such as body text, URLs, metadata, and attachments for analysis and detection. Comprehensive design is crucial as attackers constantly upgrade and build more sophisticated techniques to target people for stealing confidential information or other cyber-attacks. Relying on a single method is not a reliable long-term solution as attackers may overcome that check. Using as much information available to design a phishing detection model is a promising approach. This multi-faceted phishing detection approach utilizes various composite features of the emails in the algorithm for data processing and modeling for attack detection.

The proposed multi-faceted study introduces a new method called DARTH Framework. It helps us to address the gaps that currently exist in phishing detection. It contains clear target variables, creates models to lower users' susceptibility to attacks, and relies on multiple methods to design models for phishing attack detection. The framework fills the gap in the existing literature for early phishing detection.

This research combines multiple machine learning modeling techniques and machine learning on all the available avenues of emailing systems. The study hypothesizes that the DARTH framework can combine the natural language processing of email text and machine learning algorithms on the metadata to identify phishing email attempts.

3 Methods

As covered in Section 2, there is a gap in phishing detection techniques and prior research. Most research focuses on one aspect of detecting phishing emails, while few studies attempted a simplistic multi-feature approach. In this research, the proposed algorithm addresses the problem by including multiple composite email features,

preprocessing them, and executing simulation to predict whether emails are phishing or legitimate. In this Section, the intrinsic details of the DARTH framework are presented.

3.1 The DARTH Framework

The novel DARTH framework breaks emails into several parts such as the body texts, the embedded URLs in the emails, the email headers, and email attachment metadata. The data extracted from the emails are first processed to vectorize the data and add composite data to add more features to the data. The pre-processed data are analyzed through various individual machine learning models. As a final step, the output from the individual models is fed into an ensemble neural network. The output of the ensemble model is to predict if the is phishing or legitimate. An important aspect of the DARTH framework is that the individual models like URLs and attachments are trained on an external dataset published in prior studies and research, more details are covered in Section 3.5. The framework is flexible and allows the addition of any new neural network models on individual composite features of the emails or a more complex model to improve the accuracy of the output. Figure 1 explains the DARTH framework and its various facets. Sections 3.2 to 3.5 covers each aspect of the DARTH framework in detail.

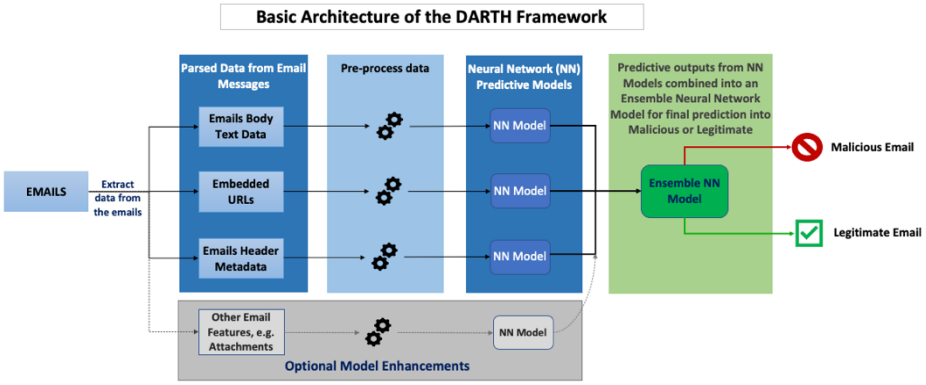

Fig. 1. The DARTH Framework Basic Architecture

3.2 The Ensemble Model

The final layer of the DARTH Framework is the ensemble model which takes inputs from various other models, primarily three different models predicting the output of the email being malicious or legitimate based on an individual composite feature of the email data. Darth Ensemble is a two-layer ensemble, where it takes the output of the email body text model, the embedded URLs model predictions, and the prediction output on the email's header metadata to the final layer. The model is built to take the inputs from the email attachments as well as an additional model. This ensemble model is the ensemble of other small ensemble models and the results from each model are evaluated in detail in Sections 4 and 5.

3.3 Feature Models

As discussed in Sections 3.1 and 3.2, there are several individual composite features of the emails which are evaluated in detail to get the individual predictions prior to combining them into an ensemble model. Over 50,000 emails were evaluated with a mix of legitimate and malicious emails. Sections 3.3.1 to Section 3.3.4 covers each composite feature in detail.

The composite feature models are neural network models as it was recorded to give superior results. However, additional modeling techniques like Logistic Regression, Random Forest, and XGBoost model are also created to compare the results to that of the neural network model. A neural network model is designed just like the human brain where information is gathered, processed in neurons, and then can provide predictions in terms of the categorical variable or continuous variable. There are three types of layers – input, dense, and output. The input layer captures all the data, the dense layer processes the data and learns from the dataset, and the output layer provides prediction. Deep learning is a subset of machine learning which uses dense layers in the neural network to learn from the data at a granular level. Each layer contains neurons that learn from the data and assigns weights to all the composite features. The output layer classifies whether an email is a phishing or not.

3.3.1 Email Body Text Model

The email body text was evaluated in detail as it has the most profound impact on the receiver of the email. Email texts can trick an unsuspecting individual to access malicious content by clicking on the embedded links, downloading attachments, or sharing sensitive personal details with the attacker. Phishing emails often have one of two important aspects, they "masquerade" the actual identity of the sender to be someone trusted or they create an "urgency" in the mind of the receiver of the email to take quick action without thinking a lot about the authenticity of the email. For example, an email from a well-known e-commerce website telling the receiver that their order was canceled due to a problem, and they must click on the link to confirm their payment details. The user may access the link and provide sensitive information like password and credit card details to the spurious webpage.

The email text data is analyzed using non-parametric methods like clustering to group them into similar groups using the KMeans clustering technique to understand if there is a pattern to the phishing emails and text can be used to identify such attempts. The pattern of such malicious emails includes words like "click now", "urgent", "immediately", "now" etc.

Various NLP techniques were employed to analyze the texts like "word2vec", "topic modeling" and "BERT". The final model was built using the transfer learning from the BERT [31], which is a pretrained English NLP model and is used to classify the email text as a phishing attempt or legitimate. A common email phishing technique is posing as Amazon for a deep discount on a product or an official email from Microsoft. Such phishing attempts then hide their actual URLs under the popular domain names for Microsoft it can be *Microsoft-sales-nk.com,* to lure users into accessing the link thinking it's from Microsoft. This sentiment is captured from body text and sent as a composite feature to neural networks. As discussed earlier, a common

technique is to create a sense of urgency by urging receivers to click on links with little thought. The urgency of the texts can be measured quantitatively using Natural Language Processing. This composite feature is passed on to neural networks as additional data. Thus, two new features are introduced, “Masquerade-ness” and “Urgent-ness”.

BERT is short for “Bidirectional Encoder Representations from Transformers”, it’s a deep learning model trained upon the Wikipedia articles. It is a bidirectional model which helps in analyzing the text in both directions of the target word such that it can predict the previous text as well as the next text based on the surrounding words. Due to its training on a huge dataset of Wikipedia articles, it is extremely powerful and has been used in the industry for various tasks like sentiment analysis, text prediction, chatbots, auto-completion of queries and email, etc. Using BERT as transfer learning proves to be a powerful tool in predicting the outcome of the email texts malicious or legitimate.

3.3.2 Embedded URLs Model

This featured model is built upon the embedded URLs in the emails. One of the major patterns noticed as part of email text clustering and topic modeling of phishing emails is that the users are urged into accessing a link embedded in the email. It's pertinent that the URLs to be analyzed as a perfectly normal-looking email from a trusted sender may contain a malicious link. In the event of man-in-the-middle attacks, where an attacker might access the conversations and relay an updated message with a malicious link, or in the event of an account being compromised the emails received may appear to be trustworthy but may have the malicious content. It's essential to analyze each URL even if the receiver trusts the sender. The URLs have various vital features like subdomains, top domains, suffixes, age of the URLs, etc. The features of the URLs were trained on an independent dataset with verified phishing from phishtank.com website data and Hannousse et al. (2021) published dataset. This model allows models to be created using external data and added to the ensemble. Ensembling models trained with external data provided valuable information to the neural networks to better detect phishing from the externally trained models.

The embedded URL data from the email were analyzed through the pre-trained models, including Logistic Regression, XGBoost, and Neural Network models, to predict the legitimate or malicious links. This model is to identify the malicious emails solely based on the probability of the embedded URLs being malicious or not. The output from the neural network model is sent further into the ensemble model.

3.3.3 Email Headers Model

A crucial part of every email is the header section which contains important information about the email and can help determine if the email can be malicious. Even though it's an integral part of any email, the content of the header is not immediately visible to the user and is easy to ignore. The headers of the email consist of a large amount of information such as the sender details, the email's route to get to the inbox (computers’ addresses that an email may have been transferred through), MIME-version (Multipurpose Internet Mail Extension), and the attachment counts, etc. This

information was used as test data to predict whether the email was suspicious to be malicious. The original model is built on the train data from the confirmed malicious emails.

The header data was analyzed using Logistic Regression, a Random Forest model, and a Neural Network model. The results were compared, and the neural network model output was sent further to the ensemble model for final prediction. This model predicts whether the email is malicious or legitimate solely on the header information. The email may contain several tens of headers, but for this analysis, only the first 11 headers per email were used.

3.3.4 Additional Models

Some additional models can also be added to the framework, for example, email attachments. This model analyzes the entropy of the email attachments and compares that to the typical entropy of such file types. If there is a significant difference in the entropy of the attachment compared to the expected entropy, then it can be a malicious email.

NapierOne has published a large dataset of malicious files of different types. A small subset of the NapierOne dataset was used to calculate the entropy of the different file types. The entropy measures the randomness of the data in a file and if the entropy value is higher than expected it could be due to any hidden executables in the simple file types (like text files). The entropy of different file types is calculated and published in Figure 2. It should be noted that if the entropy values of certain files type are not within the threshold doesn't necessarily mean that it's malicious, however, this is important information that must be accounted for to identify phishing attempts.

Though the documents were analyzed, the email attachments are not part of the final ensemble model.

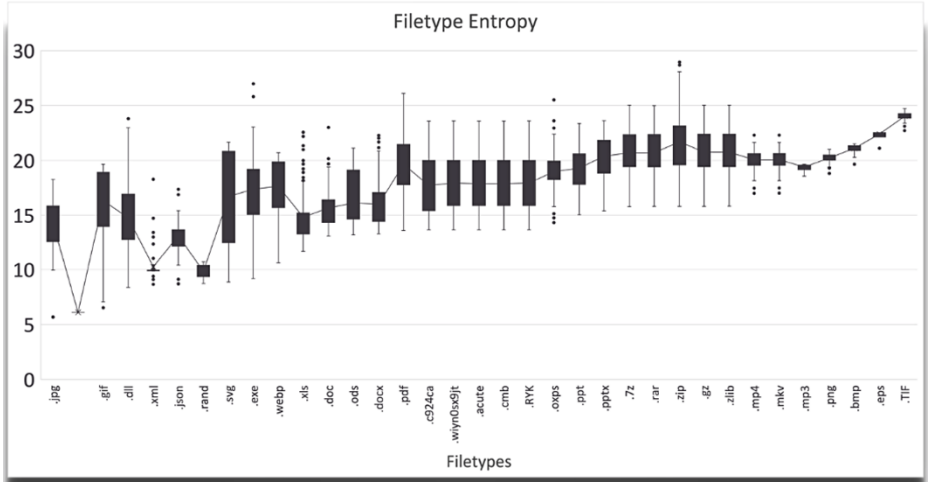

Fig. 2. Calculated Entropy for each file type shows how the entropy varies by the type of file with non-malicious content

3.4 Parsed and Processed Data

One of the major aspects of any data analysis is data processing. It is important that data is appropriately handled to extract as much information as possible. The models, such as URL models and attachments, were trained on externally published data; see Section 3.5 for the details of the data sources. However, recent emails with phishing attempts are required to test the models accurately. The authors' personal emails were used for model building and testing. To use the data appropriately, it's required to parse the information from the emails, for example: extract the email body text, parse out the embedded URLs and separate the header information.

3.4.1 Email Data Extraction

Email data is comprised of a .msg filetype that stores the entire content of an email in text format. This format of files can be downloaded from email providers as .mbox files. As such, each .msg file contains the entire data of an email. Data is parsed to read the email headers, body text, attachment counts, and, if there are any attachments, the file type.

For email body text, the email is scanned for the content type. If the body is plain text, then the entire body is used as text. However, if the body text is in HTML format, all visible texts are harvested and stored as plain text. Similarly, if the text is base64 encoded, it's first decoded then the text is stored. The stored text is pre-processed before running any further models.

The header information is extracted from the email headers and stored as a dataset with multiple features as the header information. Similarly, each email is scanned for any URLs. Once the URLs are identified, they are stored for further processing covered in Section 3.4.2.

Since data extraction is a performance-intensive process, SMU’s ManeFrame II HPC (High-Performance Computing) was extensively used to complete the data extraction.

3.4.2 Preprocessing Data

The email body text analysis requires Natural Language Processing (NLP) techniques. This requires the text data to be tokenized for any further research. Tokenization breaks down text into words which are called tokens. It establishes the meaning and context of the text by analyzing the sequences of the words. A new feature is added to the data frame, containing word token counts from the text. Using the new feature, we can notice the frequency of different token lengths for both legitimate and malicious emails. Malicious emails generally have 100 or few tokens, as shown in Figure 3.

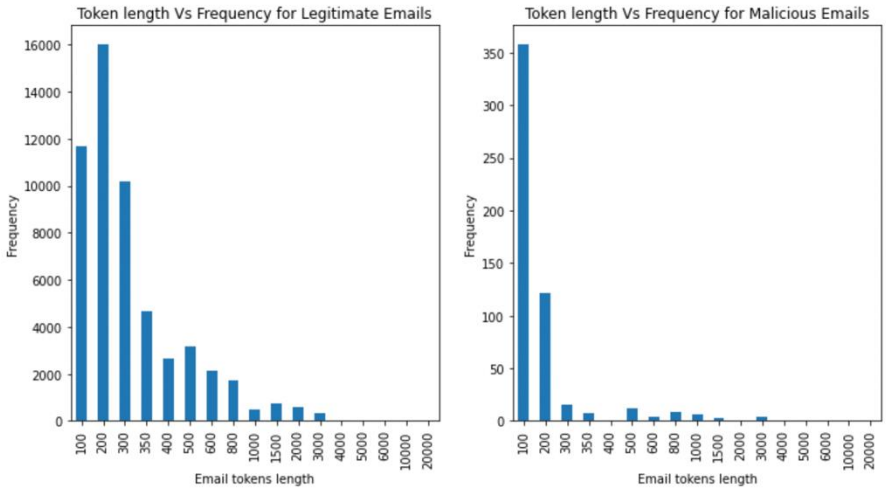

Fig. 3. Email token length by the legitimate and malicious emails. The malicious emails have lower token length compared to those of legitimate emails.

Once the URLs are extracted from the emails, a master list of URLs is created. These URLs have lots of useful metadata which is extracted during pre-processing. The extracted information has more than 50 features about the URLs, the features include top-level domains, subdomains, and suffixes. The pre-processed URL data is sent to the various models to predict if the URL is suspected to be malicious or not. The predicted outcome is stored against each URL. Any new URL extracted from the email is scanned through the master list to capture the predicted outcome for the existing URLs and if a new URL is found, it's pre-processed similarly to other URLs and added to the master list.

3.5 Data Sources

The data is sourced from various reputed places to design the DARTH framework. The training data is taken from different data sources like phishtank.com and Hannousse et al. (2021) published a URLs dataset. NapierOne has published a dataset of malicious documents which is useful in calculating the entropy of such files. For emails, the personal emails of the authors have been used to manually identify the phishing attempts to be used as a test and train dataset for the model. The details of various data sources used are listed in Table 3.

Table 3: Dataset and sources with the description of each data source.

Data Sources	Description
phishtank.com	Phishtank.com is an internet community website where phishing data is published for anyone to download. The website is managed by Cisco Talos Intelligence data. It is an open-source platform for any of its registered users to submit URLs suspected of phishing. The Cisco team verifies the submitted request and any additional information provided along with the request. If Cisco teams deem it to be phishing, then the link is then added to a list of phishing websites. There are currently about 4900 confirmed phishing URLs available at phishtank.com.
URLs dataset from the research paper Web page phishing detection *[15].*	Hannousse et al. (2021) published a URL dataset with the research paper *Web page phishing detection* [15]. This dataset has an equal number of phishing and non-phishing URLs and the URL metadata. This dataset includes various features of the URLs including domain, sub-domain, age of the domain, number of hits etc.
UCI Spam dataset	The UCI dataset is a list of emails that are classified as phishing and non-phishing email with email metadata [30]. The texts are analyzed for NLP. The attachments and URLs available in the email is used against the respective models for URLs and documents.
NapierOne Mixed File Dataset	NapierOne Mixed File Dataset [16] published a list of file types and 5000 files of each file type. In addition, the list contains some common ransomware affected/encrypted files of the same files in those 5000 examples. This study used only non-ransomware-affected files to calculate the entropy of a typical file type.
Author's personal emails with phishing attempts	Authors' emails with confirmed phishing and phishing attempts are downloaded and read by a Python script along with the metadata, email body text, attachments, and embedded URLs. The data is used as model verification for NLP-based text classifier models as well as URL verification against the phishtank.com dataset.
Sample text messages to train models in detecting "urgent-ness" in the messages	To train the BERT model to identify the emails urging users to act swiftly with little thought to the content of the email, it was required to capture sample email text to identify such attempts. Sample email text taken from the below sources were used to train the model to identify the "urgentness" in the emails. 1. Mobile Ecosystem Forum (Feb'2022): https://mobileecosystemforum.com/2022/02/18/top-five-text-message-scams-in-2021/ 2. Panda Security: https://www.pandasecurity.com/en/mediacenter/security/text-message-scams/

4 Results

The DARTH framework for phishing email detection contains an ensemble model, which is composed of four neural network models using email body text, embedded URLs, email metadata, and attachment datasets. To evaluate the performance of this model, six other ensemble models are created using all four datasets individually and a combination of those datasets. Ensemble model 1 uses the email body text dataset. The data has been preprocessed using NLP and trained with the BERT modeling technique for phishing detection. Ensemble Model 2 uses an embedded URL dataset, and Ensemble Model 3 uses an email metadata dataset. All three of those models are built using the neural network modeling technique.

All other models are different combinations of the above three models into an ensemble neural network model to predict whether the email is malicious or legitimate. The ensemble of Model 1 and Model 2 is called Model 4 which consists of the predictions based on the BERT model for body texts and the predictions based on the embedded links in the email. Model 5 is a combination of model 3 and model 2 which includes predictions based on the email headers and the embedded links. Similarly, model 6 is an ensemble of Models 1 and 3. And the final model is the ensemble of all three models which considers the predictions based on the email body texts, embedded links in the emails, and the header information captured from the emails. This is called ensemble Model 7. The results from all seven models were compared to identify the best model with the highest accuracy and precision. The models were tested on a test dataset to determine the accuracy and other metrics of the model proficiency. Table 4 defines all seven models and the steps for training those models.

As discussed previously, several models were created with different datasets to predict whether the data were malicious or legitimate for the respective data. In the end, all the predictions and composite features from individual models were combined in an ensemble model to accurately identify whether the emails were phishing or not. The results from these models with their Accuracy, Precision, and F Score are listed in Table 5. This also includes the results from prior studies by the respective authors in a similar field and is relevant to the DARTH framework presented in this paper.

In the listed results, the most essential metrics are precision and F-Score as the target feature is imbalanced. *Precision* and *F-score* are important metrics for performance evaluation for predicting imbalanced features because it breaks down both of its scores for each class – 0 being legitimate and 1 being malicious. Precision tells how well the model has predicted over correct and incorrect predictions for each class. Recall tells us the number of true positives has been found over the number of true positives in the population. F-Score is the weighted mean of both recall and precision metrics.

Table 4: List of Ensemble Models and the details about the inputs to the model

Ensemble Model	Input Feature Models	Notes
Ensemble Model 1 (EM1): Email Body Text	Body Text NN	Trained with email body text dataset. Model outputs and results are from the trained model utilizing transfer learning from BERT. Output also includes urgency prediction based on the model trained on external data.
Ensemble Model 2 (EM2): Embedded URLs	URLs NN	Trained with URLs dataset and its metadata. The model was trained on the external dataset and was used to predict the embedded URLs from the emails.
Ensemble Model 3 (EM3): Metadata	Metadata NN	Trained with email header metadata obtained from the email dataset. Predicted outputs are used for further ensemble models.
Ensemble Model 4 (EM4): Email Texts and Embedded URLs	Email Text NN and Embedded URLs NN	Pre-trained models from earlier steps were used in the ensemble NN model including the predictions from the respective models for phishing detection.
Ensemble Model 5 (EM5): Metadata and URLs	Embedded URLs NN, Header Metadata NN	Pre-trained models with URL and Metadata datasets. Prediction outputs from those models are used in the ensemble model for phishing email prediction.
Ensemble Model 6 (EM6): Metadata and Body Text	Header Metadata NN, Body Text NN	Pre-trained models with email body text (BERT) and Metadata datasets. Prediction outputs from those models used in the ensemble model for phishing email prediction
Ensemble Model 7 (EM7): Body Text, URLs, Metadata	Body Text NN, Embedded URLs NN, Metadata NN	The final model utilizes inputs from pre-trained models and their predicted output for this ensemble model to detect phishing emails.

To evaluate the model's effectiveness, Table 5 presents accuracy, precision, and f-scores for all seven models. Also, scores from other relevant research projects are presented in Table 5. The DARTH framework with an ensemble of models utilizing all email composite features provides high accuracy and precision results. The framework is Ensemble Model 7, producing an accuracy score of over 99%. The model consistently performs better than the other models with individual email features and other published studies. The performance scores of the models are listed in Table 5. As previously mentioned, six other ensemble models have been created to evaluate the framework.

Table 5: Results of various models computed and in comparison, to previously published research studies by the respective author

Category	Models	Accuracy	Precision	F-Score
Email Body Text	**EM1: Email Body Text**	**96.00**	**96.00**	**96.00**
	Ahogail et al. (2021) [8] NLP and Graph Convolutional Network on Email Body Text	98.20	98.20	98.20
	Ramanathan et al. (2012) [10] Topic Modeling plus Adaboost on Email Body Text	97.00	NA	100.00
Embedded URLs	**EM2: Embedded URLs**	**92.00**	**92.00**	**92.00**
	Haynes et al. (2021) [9] Bert on URL	96.30	96.90	96.30
	Aljofey et al. (2020) [13] CNN using URL	95.20	95.00	95.20
	Dharani et al. (2021) [5] XGBoost and Random on URL	93.70	93.80	92.80
	Sahingoz et al. (2019) [11] Random Forest and NLP on URL	98.00	97.00	98.00
Ensemble Model - Metadata	**EM3: Metadata**	**98.00**	**98.00**	**98.00**
Ensemble Model - Body Text and URLs	**EM4: Email Body Text and Embedded URLs**	**97.39**	**97.66**	**97.38**
Ensemble Model - Metadata and URLs	**EM5: Metadata and URLs**	**99.95**	**99.93**	**99.96**
	Soon et al. (2020) [17] ENN - URL and Metadata	94.20	NA	NA
Ensemble Model - Metadata and Body Text	**EM6: Metadata and Body Text**	**99.94**	**99.93**	**99.94**
Ensemble Model - Body Text, Metadata and URL	**EM7: Body Text, URLs, and Metadata**	**99.98**	**99.97**	**99.98**

EM1: Email Body Text, EM2: Embedded URLs, and EM3: Metadata models are based on individual composite features of emails, such as email body text, URLs, and metadata. These models performed at 96%, 92%, and 98% accuracy, precision, and f-score, respectively. Ensemble Model 4, which utilizes email body text and URLs, has an accuracy score of 97.39%. Ensemble model 5, which utilizes metadata and URLs,

has an accuracy score of 99.95%. Ensemble model 6, which utilizes metadata and body text, also has produced an accuracy score of 99.94%. Ensemble Model 5 and Ensemble Model 6 scored higher than Ensemble model 4. Both Ensemble Model 5 and Ensemble Model 6 utilize metadata, unlike Ensemble Model 4. All multi-faceted ensemble models have higher accuracy and precision scores compared to that individual composite feature models. The model performance accuracy scores are presented in Figure 4. The results show that not all composite feature combinations yield similar scores. Among all the composite feature models, the ensemble models with metadata as a composite feature yields higher accuracy and precision scores.

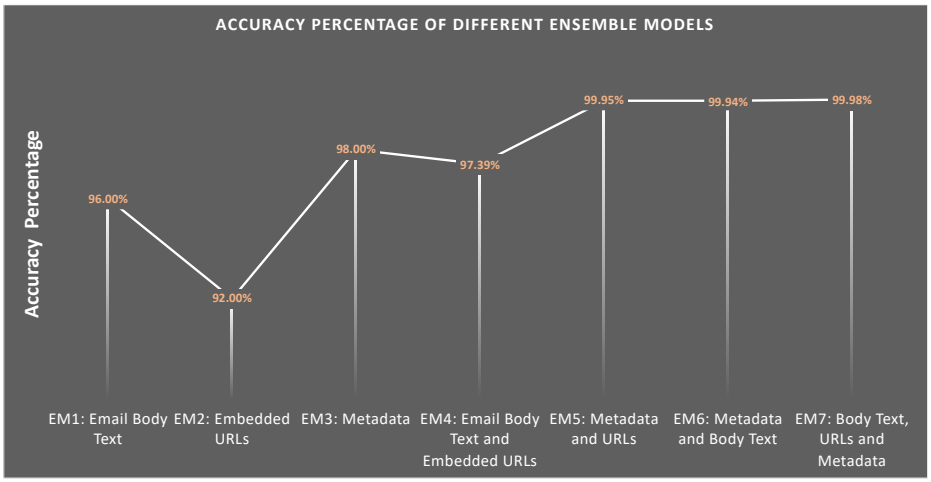

Fig. 4. Accuracy percentage distribution of each ensemble model

5 Discussion

Various models under the DARTH Framework and their results are mentioned under Section 4. The interpretation of those results is covered in detail under Section 5.1. Any research conducted has a responsibility toward society, and the ethics of said research and its possible implications must be discussed in detail. Section 5.2 talks about the ethical considerations as part of the study of the DARTH framework.

5.1 Discussion of Results

The results, as covered in Section 4, show that the ensemble models with multiple composite features yield much higher accuracy. Individual composite features analysis does give promising results with accuracy above 90%, however, the ensemble models with muti-faceted features are more successful in identifying the malicious emails. The attackers continue to change their tactics to dupe unsuspecting individuals. A single feature-based model is likely to fail in scenarios when attackers make the emails look

even more like legitimate emails. Analyzing and building an ensemble model using several aspects of the received email does provide better results.

One of the crucial aspects of emails is the attachments. The document or file attached to the email can contain malware; otherwise, a legitimate-looking email can install malicious content in the users' system and network. As part of the DARTH framework, it is recommended to add optional models, including attachment analysis. The documents entropy varies by different file types, which can help identify suspicious documents; however, the attachment is not part of the analysis of this study due to the unavailability of malicious test emails with attachments. Results from Figure 4 show the improvement in accuracy scores when multiple composite feature models are combined into an ensemble model. Body text models produced less effective accuracy scores by themselves, but metadata-based models produced results with higher accuracy and precision. This points to the fact that email metadata is an important aspect in the identification of legitimate or malicious emails. The email headers are part of the metadata, and often users ignore that information as it is not typically visible to the common users. The results show that metadata may hold more clues to finding the phishing than other features and is an important feature that plays a significant role in identifying the phishing email with better accuracy in the DARTH framework.

Other traditional research models from Soon et al. (2020), Ahogail et al. (2021), and Haynes et al. (2021) have not scored as high as the EM:5, EM:6, and EM:7 as covered in Table 5. Among the traditional research discussed in Table 5, graph convolutional networks and NLP on email body text for a phishing detection model have produced high accuracy of 98.20% [8], however, the multi-faceted ensemble model (EM:7) yields higher accuracy of 99.98%.

The individual feature models for the DARTH framework are trained using external datasets like Hannousse et al. (2021) dataset to train the URL model. It provides a good baseline for the ensemble models to perform as it learns from previous research and applies to new studies.

The results demonstrate that phishing detection can be improved through a multi-faceted approach. The existing phishing detection tools used in the industry can employ the techniques that are covered as part of the DARTH framework. It can help and thwart phishing attacks on an individual or an organization using such tools. This same idea can be used for many problem domains where adding multiple models have a better outcome than a few highly tuned models.

Composite features like metadata yields more accurate results compared to composite features like URLs and body text. Headers are essential metadata that users typically don’t see but are an important composite feature in detecting phishing.

5.2 Ethics

Algorithmic bias is a field of study under *algorithmic ethics* that analyzes the fairness of an algorithm based upon its probability of errors and compliance with its solution requirements. Ethics encompasses a broad range of topics, and primarily those are social ethics and algorithmic ethics. Algorithm ethics defines how the algorithm needs to behave and act. It should also clearly list the behavior it should avoid for producing outcomes or providing recommendations. It also deals with fairness which helps to

understand and reduce bias in the algorithm. The ethics state that the design of the algorithm should be auditable so the models can be analyzed for further development. One of the normative concerns of ethics is fairness. It deals with the algorithm's trade-off between accuracy and different notions of fairness [38]. Tsamados et al. (2021) describe the fairness of algorithm in four ways – protect categories such as race are not distinctly used for the function of the algorithm, false-positive error and false-negative error are equal for all classes of categorical variables, algorithms are properly 'calibrated' between different classes, and equal probability estimates across all classes of categorical variables [39]. There are some drawbacks to these definitions. One cannot just remove sensitive categorical features such as race and ethnicity from the dataset. Veale et al. (2017) suggest two ways fairness can be preserved in the algorithm.
One is that a third party can audit the dataset and algorithm design to reduce discrimination [1]. The other way is to have the algorithm designers collaborate with other relevant stakeholders who are experts in the domain [1]. Bias tends to arise when there is a lack of fairness. It occurs when algorithm developers deviate from requirements that list out the data collection and algorithm design standards. Removing skewed data, using a biased estimator, or introducing compensatory bias to the algorithm are ways to reduce bias [39]. The algorithm may behave unethically if biases are not reduced.

The Institution for Electrical and Electronics Engineers (IEEE) is an organization that has devoted itself to defining a code of ethics and standards for engineering professionals, most notably in emerging areas such as AI, robotics, and data management. The code of ethics ensures that professionals comply with the company and government rules. One of the organization's standards is IEEE P7003 which deals with algorithm ethics. The standard provides a framework that makes algorithm developers prioritize ethics and communicate with regulators and other stakeholders for any clarity or feedback on the objective or functionality of the application [40]. The proposed algorithm's objective is to identify phishing attacks accurately.

Given this classification problem, the research is subjected to Type I and Type II errors. False-positive (mis-identifying legitimate email as phishing email) and false-negative (mis-identifying phishing email as legitimate email) rates dictate the bias and fairness of the phishing attack detection algorithm. In the proposed framework, the model performance shows that the algorithm is not biased towards predicting legitimate emails over phishing emails. The precision rate for phishing and legitimate email detection is over 99%. The recall rate for phishing and legitimate email detection is also over 99%. The models in the proposed framework sound ethical as the false positive and false negative error rates are low and equal for both classes. Biases often challenge the framework as one can question the algorithm's fairness. The authors have set strict rules and procedures for collecting both legitimate and phishing emails to address this concern. Most importantly, the algorithm should provide enough phishing email data of several types. Since there are more volumes of legitimate emails, the algorithm will have a natural bias towards that class compared to the other class.

Unintended bias and unfairness in algorithm impacts society negatively. Technology improves societies worldwide by bringing efficiency through technological innovations, which benefit people in all aspects of their lives. These innovations occur by scaling and speeding technical advances using an astronomical amount of data. As the volume of sophisticated data grows, the threat of phishing attacks from different

cybercriminal parties worldwide increases. The proposed algorithm from this research can reduce this problem. It can protect a person or entity from revealing personal or sensitive information by mistake or aggressive cyberattacks. The protection can benefit anyone in this digital age. It can address ethics from a practical standpoint as the researchers present the steps to handle data collection, processing, and model building. Researchers also need to make proper judgments in the interest of the public or stakeholders since they deal with a group of people's sensitive and personal information. It is essential due to the public's lack of understanding or misconception of algorithms in detail. First, the proposed framework shows all the steps of collecting private emails. Then it lists out the processes that have been used to wrangle the data for building the model for the algorithm. The algorithm does not require human intervention as one does not need to access these emails for any data processing for phishing detection. There is no purpose for accessing any individual email for building this algorithm. The lack of human intervention addresses a critical aspect of the cybersecurity code of ethics: personal autonomy. As described previously, some existing methods require human intervention to preprocess emails for phishing attack prevention. In a manual process, the scientist may have to access private emails for preprocessing or may end up mistakenly taking a step that may leak the confidential information of the senders and recipients. After detecting the attack, cybersecurity analysts must take manual steps where the individual must take the server down to act on the attack.

Formosa et al. (2021) state that during a manual process like that, the chance of preventing the attack is low [37]. The proposed algorithm eliminates manual processes like this as there is no human intervention for preprocessing, and the suspicious emails never reach the recipients' destination. The elimination of the process benefits everyone as private information never gets leaked. Also, the public and any institution never have to face any threats the attackers pose. All the steps meet the requirement of IEEE standards.

6 Conclusion

The multi-faceted approach of using an ensemble of multiple independent composite feature models yields highly accurate ensemble phishing detection models even when lower quality feature models are used. The novel DARTH framework decomposes email into composite features and allows independent models on each composite feature to be developed and used. The ensemble of the output of these models achieved 99.98% accuracy in detecting phishing emails in our test data.

Adding more composite features improves the accuracy of the ensemble model. Experiments showed that the ensemble model created from two composite features always yielded better results than the models for individual composite feature, and the ensemble model created from three composite feature yielded superior results compared to the ensemble models with two composite features. Ensemble models created using the several models of composite features, DARTH framework is more accurate compared to a single model with only non-composite features as input.

The DARTH framework is usable in any problem domain with identifiable and separable composite features. The framework is particularly useful where the composite

features are disjoint. The phishing email detection problem domain is the exemplary domain for the DARTH framework.

References

1. M. Veale and R. Binns (2017) Fairer machine learning in the real world: mitigating discrimination without collecting sensitive data. Big Data Soc 4(2):205395171774353. https://doi.org/10.1177/20539 51717743530
2. A. K. Dutta (2021) Detecting phishing websites using machine learning technique. PLoS ONE 16(10): e0258361. https://doi.org/10.1371/journal.pone.0258361
3. A. Akinyelu, and A. Adewumi (2014). Classification of Phishing Email Using Random Forest Machine Learning Technique. Journal of Applied Mathematics. 2014. 10.1155/2014/425731.
4. P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive Block listing to Detect Phishing Attacks," 2010 Proceedings IEEE INFOCOM, 2010, pp. 1-5, DOI: 10.1109/INFCOM.2010.5462216.
5. Detection of Phishing Websites Using Ensemble Machine Learning Approach Dharani M., Soumya Badkul, Kimaya Gharat, Amarsinh Vidhate, and Dhanashri Bhosale ITM Web Conf., 40 (2021) 03012, DOI: https://doi.org/10.1051/itmconf/20214003012
6. S. Abu-Nimeh, D. Nappa, X. Wang and S. Nair (2007) "Distributed Phishing Detection by Applying Variable Selection Using Bayesian Additive Regression Trees." 2009 IEEE International Conference on Communications, IEEE, 2009, pp. 1–5, https://doi.org/10.1109/ICC.2009.5198931.
7. N. Sanglerdsinlapachai and A Rungsawang. "Using Domain Top-Page Similarity Feature in Machine Learning-Based Web Phishing Detection." IEEE, 2010, pp. 187–190, https://doi.org/10.1109/WKDD.2010.108.
8. A. Alhogail and A. Alsabih (2021). Applying machine learning and natural language processing to detect phishing emails. Computers & Security, 110, 102414. https://doi.org/10.1016/j.cose.2021.102414
9. K. Haynes, H. Shirazi, and I. Ray (2021). Lightweight URL-based phishing detection using natural language processing transformers for mobile devices. Procedia Computer Science, 191, 127–134. https://doi.org/10.1016/j.procs.2021.07.040
10. V. Ramanathan and H. Wechsler (2012). phishGILLNET—phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training. EURASIP Journal on Multimedia and Information Security, 2012(1), 1–1. https://doi.org/10.1186/1687-417X-2012-1
11. O.K. Sahingoz, E. Buber, O. Demir, and B. Diri (2019). Machine learning-based phishing detection from URLs. Expert Systems with Applications, 117, 345–357. https://doi.org/10.1016/j.eswa.2018.09.029
12. A. Abbasi, D. Dobolyi, A. Vance, and F. M. Zahedi (2021). The Phishing Funnel Model: A Design Artifact to Predict User Susceptibility to Phishing Websites. Information Systems Research, 32(2), 410–436. https://doi.org/10.1287/isre.2020.0973
13. A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.P. Niyigena (2020). An Effective Phishing Detection Model Based on Character Level Convolutional Neural Network from URL. Electronics (Basel), 9(9), 1514. https://doi.org/10.3390/electronics9091514
14. C. Jones (2022, January 18). 50 phishing stats you should know in 2022. 50 Phishing Stats You Should Know In 2022. Retrieved January 27, 2022, from https://expertinsights.com/insights/50-phishing-stats-you-should-know/

15. A. Hannousse and S. Yahiouche (2021), "Web page phishing detection", Mendeley Data, V3, doi: 10.17632/c2gw7fy2j4.3
16. NapierOne Mixed File Dataset was accessed on February 19, 2022, from https://registry.opendata.aws/napierone.
17. G. K. Soon, C. O. Kim, N. M. Rusli, T. S. Fun, R. Alfred, and T. T. Guan, T. (2020). Comparison of simple feedforward neural network, recurrent neural network, and ensemble neural networks in phishing detection. Journal of Physics. Conference Series, 1502(1), 12033. https://doi.org/10.1088/1742-6596/1502/1/012033
18. Y. Z. Ju, Z. Chen, F. Liu, and X. Fang (2020). DTOF-ANN: An Artificial Neural Network phishing detection model based on Decision Tree and Optimal Features. Applied Soft Computing, 95, 106505. https://doi.org/10.1016/j.asoc.2020.106505
19. F. T. Mohammad and L. McCluskey (2013). Predicting phishing websites based on self-structuring neural network. Neural Computing & Applications, 25(2), 443–458. https://doi.org/10.1007/s00521-013-1490-z
20. W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak (2020). Accurate and fast URL phishing detector: A convolutional neural network approach. Computer Networks (Amsterdam, Netherlands: 1999), 178, 107275. https://doi.org/10.1016/j.comnet.2020.107275
21. N. Kamireddy, K P Ruphaa Sri, B. Vishruthi, and M. S. Anand. (2021). Precise Detection of Phishing URLS Using Recurrent Neural Networks. i-Manager's Journal on Computer Science, 9(1), 21. https://doi.org/10.26634/jcom.9.1.18154
22. M. R. Ahmad, Rafie and S. M. Ghorabie, (2021). Spam detection on Twitter using a support vector machine and users' features by identifying their interactions. Multimedia Tools and Applications, 80(8), 11583–11605. https://doi.org/10.1007/s11042-020-10405-7
23. Grimes. (2017). Hacking the Hacker: Learn from the Experts Who Take down Hackers. John Wiley & Sons, Incorporated.
24. California Enacts Tough Anti-phishing Law; California Gov. Arnold Schwarzenegger has signed anti-phishing legislation into law. (2005). InternetWeek (Manhasset, N.Y.).
25. S. R. Bowman, G. Angeli, C. Potts and C. D. Manning (2015). A large, annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326. Chicago.
26. California Enacts Tough Anti-phishing Law; California Gov. Arnold Schwarzenegger has signed anti-phishing legislation into law. (2005). InternetWeek (Manhasset, N.Y.).
27. Verizon 2021 Data Breach Investigations Report [Online]. Retrieved Feb 9, 2022, from https://www.verizon.com/business/resources/reports/2021-data-breach-investigations-report.pdf
28. Z. H. Zhou, L. Wu, and W. Tang (2002). Ensembling neural networks: many could be better than all. Artificial intelligence, 137(1-2), 239-263.
29. Y. Chen, H. Chang, J. Meng, and D. Zhang (2019). Ensemble Neural Networks (ENN): A gradient-free stochastic method. Neural Networks, 110, 170-185.
30. D. Dua, and C. Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: the University of California, School of Information and Computer Science.
31. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
32. K. Rekouche (2011). Early phishing. *arXiv preprint arXiv:1106.4692*.
33. Merriam-Webster. (n.d.). Phishing. In Merriam-Webster.com dictionary. Retrieved April 9, 2022, from https://www.merriam-webster.com/dictionary/phishing

34. Webroot, Types of Phishing Attacks You Need to Know to Stay Safe [Online]. Retrieved July 21, 2022, from https://mypage.webroot.com/rs/557-FSI-195/images/Webroot_11%20Types%20of%20Phishing_eBook.pdf
35. J. H. Williams, and A. N. Joinson (2018). Exploring susceptibility to phishing in the workplace. International Journal of Human-Computer Studies, 120, 1–13. https://doi.org/10.1016/j.ijhcs.2018.06.004
36. S. Vallor (n.d.). Intro to cybersecurity ethics - an introduction to cybersecurity ethics module author: Shannon. StuDocu. Retrieved May 30, 2022, from https://www.studocu.com/en-us/document/boston-university/information-security/intro-to-cybersecurity-ethics/17140429
37. P. Formosa, M. Wilson and D. Richards (2021). "A Principlist Framework for Cybersecurity Ethics." Computers & Security, vol. 109, Elsevier Ltd, 2021, p. 102382, https://doi.org/10.1016/j.cose.2021.102382.
38. M. Kearns, and A. Roth (2022, March 9). Ethical Algorithm Design Should Guide Technology Regulation. Brookings. Retrieved June 9, 2022, from https://www.brookings.edu/research/ethical-algorithm-design-should-guide-technology-regulation/#footnote-3
39. A. Tsamados, N. Aggarwal, J. Cowls, J. Morley, H. Roberts, M. Taddeo and L. Floridi (2022). "The Ethics of Algorithms: Key Problems and Solutions." AI & Society, vol. 37, no. 1, Springer London, 2021, pp. 215–30, https://doi.org/10.1007/s00146-021-01154-8.
40. IEEE Announces Standards Project Addressing Algorithmic Bias Considerations. (2017, Mar 09). Business Wire https://www.proquest.com/wire-feeds/ieee-announces-standards-project-addressing/docview/1875371352/se-2?accountid=6667
41. Anti-Phishing Working Group (APWG) (2014). Phishing Activity Trends Report, 3rd Quarter 2021 [Online] Retrieved Feb 9, 2022, from https://docs.apwg.org/reports/apwg_trends_report_q3_2021.pdf