Classification of Warranted and Unwarranted Spam: Improving User-Feedback Trained Spam Filters

Eric Burton Samuel Martin
eric.burton.martin@colostate.edu
Department of Computer Science
Fort Collins, CO, USA

ABSTRACT

Building upon my prior research, which introduced a nuanced approach to spam classification and supplied a brand-new ’warranted spam’ dataset, I will be working towards addressing the future work section by developing a classification model to distinguish between warranted and unwarranted spam. My previous work highlighted the potential issues with user-feedback based spam filters and the biases they harbor. With the use of the newly created ’warranted spam’ dataset, combined with the Bruce Guenter "unwarranted spam" dataset [4], I have presented two classification models that work together to decide if an email is warranted or unwarranted.

The first model utilizes the capabilities of the BigBird transformer introduced by Google to tokenize and create embeddings of our textual data which is then fed into a long short-term memory recurrent neural network (RNN/LSTM) to capture sequential dependencies and context from the emails and output a binary classification. The second model is a simple feed-forward neural network (NN) that is trained on various non-textual features of our email datasets to determine a classification. These two models proceed to average their prediction values to arrive at a final classification. Utilizing both RNN/LSTM and traditional NN models to differentiate between warranted and unwarranted spam allows us to analyze both the textual and non-textual content of emails to make a more well rounded and robust decision.

Current spam filters like Google’s use machine learning models that heavily rely on user-feedback to train their models. Yet a large amount of perceived spam is actually warranted and benign commercial correspondence that could be safely unsubscribed from. This can lead to the spam filter’s weights to become focused on identifying the features of warranted commercial correspondence and lose the ability to identify the nuanced features that malicious unwarranted spam present. To address this, the ultimate goal of this research is to create a classification model that will be utilized in the step between when a user reports an email as spam and the flagged mail is used to update the spam filter model. Our classification model would take the user-flagged mail, classify it, and if it is classified as warranted spam it will not be used to update the spam filter ML model, but if it is classified as unwarranted spam, it will be used to updated the filter. In both cases the email provider will still provide rules to filter this flagged mail but the main spam filter will become more and more focused on truly malicious unwarranted spam and less biased on benign and annoying commercial correspondence.

CCS CONCEPTS

• **Information systems** → **Document filtering**; **Information extraction**; • **Security and privacy** → *Information flow control*; **Usability in security and privacy**; **Economics of security and privacy**; **Social aspects of security and privacy**; • **Computing methodologies** → *Machine learning*; *Natural language processing*.

KEYWORDS

Warranted Spam, Unwarranted Spam, Spam Detection, Dataset, Spam, Email Filtering, Spam Classification, Machine Learning, Natural Language Processing

ACM Reference Format:
Eric Burton Samuel Martin. 2023. Classification of Warranted and Unwarranted Spam: Improving User-Feedback Trained Spam Filters. In *Natural Language Processing, August 21 – December 13, 2023, Fort Collins, CO, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the digital age, the prevailing perception of spam has evolved, yet the definitions commonly used to describe it are becoming more and more vague. Terms like “unsolicited,” “irrelevant,” “inappropriate,” “unwanted,” and “commercial messages” are frequently used in these definitions. However, this broad and somewhat ambiguous characterization is increasingly working against the effectiveness of spam filters, which are crucial in safeguarding against malicious unwarranted spam. To enhance the resilience of these filters, it is imperative to revisit and refocus on the definition of ’unsolicited.’ The term ’unsolicited’ implies messages that are ’not asked for or requested.’ Surprisingly, a significant portion of what is currently perceived as spam includes messages that recipients have, in some form, consented to receive. This consent may be explicit, as seen in newsletter subscriptions, or implicit, often buried in the fine print of terms and conditions, online purchases, or free services. In such instances, users might inadvertently agree to future communications. These emails, though potentially unwanted, is warranted as they are sent with a degree of prior consent. This nuanced understanding is crucial in refining spam filters to more effectively distinguish between benign, albeit unsolicited, communications and genuinely malicious unwarranted spam.

This paper introduces a nuanced distinction between ’warranted spam’ and ’unwarranted spam.’ We define ’warranted spam’ as legitimate communications that recipients have consensually opted

CS542 ’23, August 21 – December 13, 2023, Fort Collins, CO, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

into, knowingly or not, originating from credible sources, and providing clear, safe opt-out options. In contrast, 'unwarranted spam' encompasses unsolicited and often malicious messages sent without the recipient's consent, where unsubscribing may be ineffective or even harmful.

Current spam datasets, such as Ling-Spam [2], SpamAssassin [9], Enron-Spam [8], TREC07 [3], CSDMC [1], and others, often contain data that is outdated and indiscriminately mix warranted and unwarranted spam. This amalgamation hinders the development of precise and effective filtering solutions, as it blurs the distinct characteristics of each spam type. This issue is critical, as unwarranted spam is not merely an annoyance but a conduit for phishing, malware, and other cyber threats. Furthermore, unwarranted affiliate marketing spam is increasingly used for financial gain, potentially damaging the reputations of associated companies.

Recognizing these challenges, our research aims to develop a modern dataset of warranted spam, painstakingly curated through manual sign-ups for email correspondence, akin to a typical user's experience. This dataset is designed to complement existing unwarranted spam datasets, like the one maintained by Bruce Guenter [4], offering a more comprehensive understanding of spam.

For individual users, this refined approach to spam detection promises reduced exposure to malicious content. For businesses, particularly those relying on email marketing, our dataset offers potential for improved deliverability, increased customer engagement, and compliance with regulations like the CAN-SPAM Act and GDPR. Crucially, it also aids in preserving brand integrity, ensuring that legitimate communications are less likely to be misclassified as spam, which can erode customer trust and tarnish reputations.

Current spam filters like Google's use machine learning models that heavily rely on user-feedback to train their models. Yet a large amount of perceived spam is actually warranted and benign commercial correspondence that could be safely unsubscribed from. This can lead to the spam filter's weights to become focused on identifying the features of warranted commercial correspondence and lose the ability to identify the nuanced features that malicious unwarranted spam present. To address this, the ultimate goal of this research is to create a classification model that will be utilized in the step between when a user reports an email as spam and the flagged mail is used to update the spam filter model. Our classification model would take the user-flagged mail, classify it, and if it is classified as warranted spam it will not be used to update the spam filter ML model, but if it is classified as unwarrantes spam, it will be used to updated the filter. In both cases the email provider will still provide rules to filter this flagged mail but the main spam filter will become more and more focused on truly malicious unwarranted spam and less biased on benign and annoying commercial correspondence.

2 GOALS

To address the issues presented in the introduction we have set the following goals:

1. Increase the effectiveness of current spam filters that rely on user-feedback by decreasing the amount of malicious spam that bypasses modern spam filters that rely on user-feedback to train their models.
2. Produce a classification model that can discriminate between benign warranted spam and malicious unwarranted spam.

By completing these goals we can decrease spam filter bias against commercial correspondence and simultaneously harden spam filters to focus on malicious spam.

3 MOTIVATION

Our work was inspired by a chief information security officer who has stated that affiliate marketing and survey-based spam have been consistently bypassing the corporate spam filters of the past two companies they have worked for. They had presented to us research of their own that showed that spammers have incentive to send unwarranted affiliate marketing and survey-based emails in bulk since they can collect affiliate commissions, obtain and sell user-data, and can deploy malware on victim's machines. The recipients aren't the only victims though. The brands that are associated with these spammers, who are breaking the terms of their affiliate agreements, risk tarnished reputations as the recipients of the unsolicited mail begin to perceive their companies as 'spammy' and untrustworthy.

4 CONJECTURE

Our conjecture explores the hypothesis that spam filters relying on user feedback for training may inadvertently contribute to inefficiencies in spam detection. This issue primarily arises from the nature of user interactions with their email and the subsequent impact on the training data of spam filter models.

4.1 Reliance on User Feedback

The tendency of users to flag solicited benign emails as spam (warranted spam), rather than unsubscribing, introduces noise into the spam filter model. This incorrect labeling of spam can potentially over-weight patterns of authentic commercial correspondence, leading to a skewed understanding of what constitutes malicious unsolicited spam (unwarranted spam). As Google's spam filters illustrate below, user feedback plays a key role in shaping spam detection models.

> "Simply put, to protect users at scale, we rely on machine learning powered by user feedback to catch spam ... it easier to adapt quickly to ever-changing spam tactics. Gmail employs a number of AI-driven filters that determine what gets marked as spam. These filters look at a variety of signals, including characteristics of the IP address, domains/subdomains, whether bulk senders are authenticated, and user input. User feedback, such as when a user marks a certain email as spam or signals they want a sender's emails in their inbox, is key to this filtering process, and our filters learn from user actions. "

[5]

4.2 Bias in Training Data

The effectiveness of a model is inherently tied to the quality of its training data. When users misreport legitimate emails as spam, it introduces a bias against these emails. This can lead to higher false positive rates for commercial mail, where legitimate communications are mistakenly identified as spam.

4.3 Impact on Spam Detection Effectiveness

Relying heavily on user-reported spam can create blind spots in detection. Over-tuning to this type of spam may cause the model to miss subtle cues of real malicious unsolicited spam, allowing it to slip through due to the learned biases.

4.4 Nuanced Approach for Effective Spam Detection

To address these challenges, a nuanced approach is proposed:

- Introduce two new classification labels for spam: **'warranted'** and **'unwarranted.'**
- Generate a model to distinguish between these two classifications.
- Classify all user-flagged spam as 'warranted' or 'unwarranted.'
  - If classified as warranted, do not use it to update the spam-filter model, but add a rule to filter it.
  - If classified as unwarranted, use it in future spam-filter model updates.

As Google's spam filters illustrate, user feedback plays a key role in shaping spam detection models. However, this reliance can inadvertently lead to biases that compromise the effectiveness of these models in distinguishing between truly malicious spam and benign commercial correspondence. [5].

5 DATASET GENERATION METHODOLOGY

In order to create a classification model we need to have access to labeled warranted spam and unwarranted spam. We will be utilizing the Bruce Guenter Spam Archive [4] as our unwarranted spam but needed to establish our own warranted spam dataset. This section details that process. We have provided three datasets to the public in our Warranted Spam Archive[1].

5.1 Warranted Spam Email Datasets

To host our warranted spam, we have created three accounts.

1. **PRIMARY@gmail.com** - The primary dataset. This account meticulously registers to websites found in the website repository[1] and employs the '+' feature in Gmail to trace the original source of each email. For example, PRIMARY+nike@gmail.com would be used to register for Nike.com.
2. **AD-HOC@gmail.com** - A supplementary dataset that, unlike the primary account, does not maintain a record of each sign-up. This dataset is designed for convenient, ad-hoc sign-ups encountered during researchers' daily lives outside of working hours.
3. **FORWARD@outlook.com** - This account has the single purpose of receiving emails forwarded by the primary Gmail account. This incorporates email headers generated by Microsoft Outlook into the dataset.

All emails received that are categorized as spam by the email provider are forwarded to a dedicated "Labelled Spam" folder for easy identification and segregation for future analysis. This allows researchers to investigate features that can cause commercial warranted spam to be classified as spam.

5.2 Website Repository

For the primary email account, we created a diverse website repository[1] spanning 28 categories that were systematically chosen to maximize affiliate communications, marketing emails, and newsletters. Each category contains more than 70 websites that were generated utilizing ChatGPT4 and Google and vetted by the team. A few examples of the categories within the repository are Retail, Travel, Finance, Health, Sports, Food, Beauty, Fashion, News, Crypto, and Job Search. The repository contains the website name, URL, unique email address provided to the site (using the '+' feature), whether registration was successful, and comments if applicable.

5.3 Sign-up Methodology

For the primary account, each site in the website repository[1] was manually visited and surveyed for newsletter sign-ups, pop-ups, or account creation procedures that included an opt-in email system. Once a place for an email address was found, a unique email address employing the '+' feature was supplied and if possible, the highest email frequency was chosen. For the ad-hoc account, researchers provide the email address when they encounter a sign-up form in their daily lives. The ad-hoc account does not use the website repository as a reference. Therefore, the primary and ad-hoc accounts may have some overlapping sign-ups.

5.4 Sign-up Automation

Automation of the sign-up process has proven difficult due to the diverse and complex registration requirements across different websites, including CAPTCHA challenges, phone number verification, and unique form structures designed to deter automated interactions. We are currently working on an automatic sign-up bot for websites that have simple email input fields to help dramatically increase our dataset size.

5.5 Data Cleansing and Management

We developed several Python scripts for data management and cleansing. These scripts perform tasks such as extracting individual emails from large .mbox files, organizing emails chronologically, scrubbing sensitive recipient data, extracting key metrics from collected emails, including sender domain, unsubscribe link count, presence of tracking pixels, and authentication results, among others, and updating the warranted spam archive website [1].

We noticed a substantial difference in storage consumption between the primary Gmail account and its corresponding Outlook account, which receives emails forwarded from the primary account. Even with a comparable number of emails, the storage used by Outlook significantly exceeds that of Gmail as seen in Table 1.

[1] EbMartin. 2023. Warranted Spam Archive. https://www.cs.colostate.edu/~ebmartin/warrantedSpamDataSet/

	Primary	Ad-Hoc	Forwarded
Provider	Gmail	Gmail	Outlook
Instantiation	3-May-23	31-Mar-23	18-May-23
GB	14.12	4.93	15 (Max)
Total Emails	164.8K	71.2K	54.4K
Spam	1.4K (2.0%)	1.4K (2.90%)	1.4K (2.60%)

Table 1: Summary of statistics from creation date until 12-Nov-2023 for each email Account used in dataset collection.

6 CLASSIFICATION METHODOLOGY

For our classification we are utilizing the combined predictions of two models. One model is trained on textual data and features BigBird (as a pretrained encoder) [12] and an RNN/LSTM model and the second model is trained on non-texutal features and employs a simple feed-forward neural network.

6.1 Data Preprocessing and Tokenization

The raw email dataset underwent a series of preprocessing steps, including cleaning, handling missing values, and feature extraction. The textual content of emails (body, subject, and comments) had to undergo heavy cleaning due to the tendency of spam and commercial correspondence to utilize various obfuscation techniques, encodings, and fail to abide by multipart email etiquette (declaring a boundary value and failing to use it). To extract features and text from the raw email files we utilized the mailparser tool provided by SpamScope [11]. Its capability to handle different email formats and extract a wide range of data, including defects, made it a great choice for processing our complex spam datasets.

Once the text was cleansed it was then tokenized using the BigBird tokenizer which is a sparse-attention based transformer which extends Transformer based models, such as BERT to much longer sequences [12]. Due to the length of the emails averaging at around 3000 words the BigBird tokenizer was chosen to convert the text into a sequence of input IDs and attention masks, suitable for RNN/LSTM processing.

We also extracted various non-textual features from the emails that we felt may be representative of warranted and unwarranted spam for use in the NN model. The features we have extracted are 'num plain text blocks', 'num html text blocks', 'num unmanaged text blocks', 'defects count', 'defects categories count', 'number of unsubscribe links', 'number of undecodable characters', 'tracking pixel present', 'total num of images', 'total links in email', 'email size (bytes)', 'dkim signature present', 'unsafe link count', 'unsafe image url count', 'unsafe button url count', 'link url count', 'image url count', 'button url count', and 'unsafe to safe link ratio'.

6.2 Model Architecture

Two distinct models are employed: an RNN with LSTM layers and a Feed-Forward Neural Network (NN).

6.2.1 ***RNN/LSTM Model Architecture.*** The Recurrent Neural Network (RNN) model employed in this research is an LSTM (Long Short-Term Memory) variant inspired by work peformed by Topbas et al. in 2021 [7]., designed to handle sequences of textual data. The model architecture comprises the following key components:

- **Embedding Layer:** Maps the input tokens, represented as indices, to dense vectors of a specified size (embedding dimension). This layer is essential for capturing the semantic relationship between words in the input text.
- **LSTM Layer:** A bidirectional LSTM layer processes the embedded text, capturing both forward and backward dependencies in the sequence. The LSTM's ability to retain long-term dependencies makes it well-suited for text data.
- **Fully Connected Layer:** Transforms the output of the LSTM layer to the desired output dimension. This layer acts as the classifier in the context of the task.

The model's forward pass involves the sequential processing of data through these layers. Firstly, input tokens are embedded and then fed into the bidirectional LSTM. The LSTM's output at the final time step is passed to the fully connected layer for classification. The model is optimized using the Adam optimizer with a defined learning rate, and the loss is computed using CrossEntropyLoss, suitable for classification tasks.

The RNN model leverages LSTM's ability to capture long-term dependencies in sequential data, making it a good choice for processing tokenized complex email content. It includes an embedding layer, bidirectional LSTM layers, and a fully connected output layer. The NN model, designed to process non-textual email features, consists of several fully connected layers with ReLU activations and dropout for regularization.

6.2.2 ***Feed-Forward Neural Network Architecture.*** The feed-forward neural network (NN) in this study is a standard modern feed-forward artificial neural network, designed to handle non-textual features of emails. Its architecture comprises:

- **Input Layer:** Accepts input features matching the count of non-textual features.
- **Fully Connected Layers:** Three linear layers transition the input through 128, 64, and finally to a single output value.
- **Activation Functions:** ReLU activations follow each linear layer, except the last which uses a Sigmoid function for binary classification.
- **Dropout:** Post-activation dropout (rate of 0.7) counters overfitting by omitting random outputs.
- **Regularization:** Weight decay in the Adam optimizer for L2 regularization further helps in preventing overfitting, ensuring the model generalizes well on unseen data.

This NN effectively discerns complex patterns within non-textual email features, aiding in accurately classifying warranted and unwarranted spam emails. The Binary Cross-Entropy Loss function guides the training, with Adam optimizer and weight decay attempting to enhance the model robustness.

6.2.3 ***Combining Model Outputs.*** The outputs of the RNN/LSTM and Feed-Forward Neural Network (NN) models are combined to leverage the strengths of both architectures and allow for a more robust detection methodology. This combination process is as follows:

(1) **Combining Predictions:** Predictions from both models are combined by averaging their probabilities. This merged prediction aims to balance the nuanced textual understanding

of the RNN/LSTM model alongside the NN's classification bsed on the non-textual features.

(2) **Final Decision Making:** A threshold of 0.5 was chosen and applied to the combined predictions to classify emails as warranted or unwarranted spam.

This method can take advantage of the complementary strengths of both models, potentially improving the overall classification accuracy over using either model in isolation.

6.3 Training and Evaluation

Both models are trained on separate aspects of the dataset. The RNN model is trained on tokenized textual data, while the NN model focuses on non-textual features like email size, the presence of tracking pixels, and counts of various elements within the emails. The training involves standard procedures like batch processing, loss computation using cross-entropy (RNN) and binary cross-entropy (NN), and backpropagation. Evaluation metrics include accuracy and the F1 score.

7 RESULTS

7.1 RNN/LSTM Performance

The RNN/LSTM model with a vocabulary size of 50358, embedding dimension of 256, hidden dimension of 256, two-layer architecture, and a binary output was trained with a learning rate of 0.001 using the Adam optimizer. Over three epochs, the model achieved a test loss of 0.0116, an impressive test accuracy of 99.74%, and an F1 score of 0.9987, indicating a highly effective classification performance. This performance is concerning though and we delve deeper into why in the discussion.

7.2 NN Performance

Similarly, the Feedforward Neural Network (NN) was trained using Binary Cross Entropy Loss and Adam optimizer with a learning rate of 0.001 and weight decay for L2 regularization and multiple dropout layers. After three epochs, the NN model reported a loss of 0.1045, an accuracy of 98.91%, and an F1 score of 0.9945, showing either robustness in classifying emails based on non-textual features or signs of overfitting to the data which will be discussed in the discussion.

7.3 Interpretability Analysis

Two interpretable machine learning methods, LIME (Local Interpretable Model-agnostic Explanations) ![10] and SHAP(SHapley Additive exPlanations) [6], were employed to elucidate the decision-making process of the NN model. The LIME analysis, see Figure 1, pinpointed features such as the unsafe-to-safe link ratio and tracking pixel presence as significant.

Similarly, the SHAP summary plot, see Figure 2, shows similar attributes, providing a visual and quantitative understanding of feature influences on model predictions.

8 DISCUSSION

(Since this is for a CS class I am going to cut the scientific talk and speak casually in the discussion and use I instead of we now as well. I can convert this into scientific jargon next semester when I finish the research)

Alright, so the models are doing disconcertingly well and I don't like that. I am a bit skeptical because when something seems too good to be true, it often is. I am curious about what's really going on under the hood, especially with the RNN/LSTM model that's

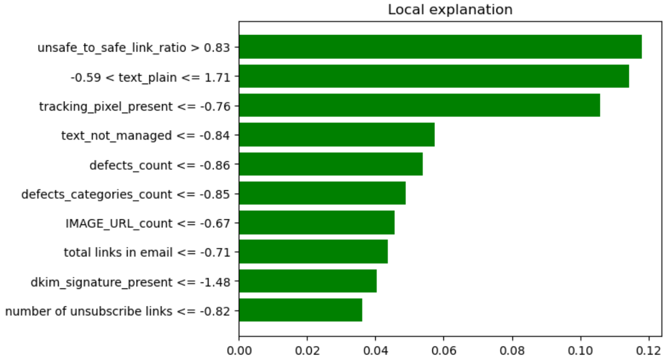

Figure 1: Local explanation model by LIME showcasing the impact of individual features on the neural network's prediction

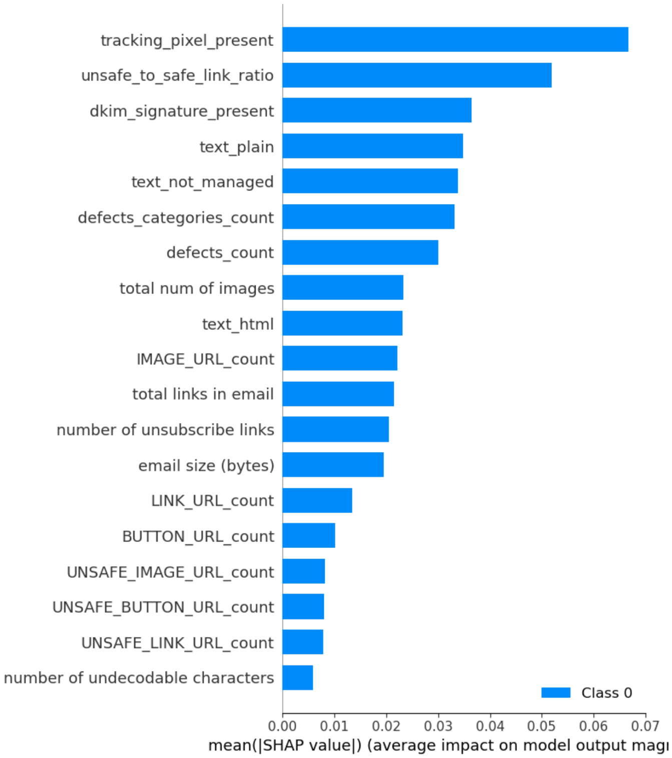

Figure 2: Summary of SHAP values indicating the average impact of each feature on the model's predictions, with tracking pixel presence and link safety ratios among the most influential factors

working on the textual data but all the techinques I am used to using for simple neural networks don't seem to work with these models. So since we are dealing with logits and tokens I did some research and am planning on using Integrated Gradients to try to get a feel for which tokens are doing too much work in the predictions. I want to make sure our model isn't just picking up on a few specific words that make it super accurate. My biggest theory as to why it does so well, which literally just came to mind while writing this is that the mail can mention the recipients names, which in my warranted spam dataset would be "honey" or "honey potter" which are clearly not in the unwarraned dataset.

When people say cleaning the data is 90% of the job, they mean it. I spent so long cleaning this dataset, stripping it of personal information so that people can tamper with the email associated with the dataset, decoding crazy characters, and much more. So now it seems I need to ensure the names associated with the warranted spam dataset email address are cleaned out as well. That won't be difficult though and I am grateful I identified this issue during this project. I will let you know if this was actually the issue when I see you next semester.

Once I identify that the model isn't just 'cheating' to get good results. I plan on testing our models on our Ad-Hoc dataset that contains different warranted spam data than the Primary dataset used in this classification. Along with that dataset, I want to test it on other public spam datasets, albeit they are super outdated, and see if we can find some warranted spam hiding amongst the unwarranted spam. I didn't get to adjust the hyperparameters as much as I would like but due to the time it took to clean all the data alongside how long training the RNN/LSTM took I was happy enough with the outcome since I could think long and hard about it in this writeup.

The NN model took no time at all to train and I was able to identify that it liked to use the following features to make most of its decisions, the presence of tracking pixels, the number of unsafe links (http) to safe link (https) ratio, the presence of a DKIM signature, and if the mail had defects (it claimed to do something and failed to do so... so doing spammy things). So since I know this model doesn't have access to the email recipient names, I am happy with the results so far.

Another unexplored avenue that I would really like to investigate is the application of cosine similarity to compare warranted, unwarranted, and regular mail. By comparing the average vectors of warranted spam, unwarranted spam, and personal emails (that do not fall into warranted and unwarranted categories), I can make a model capable of estimating the classification of emails based on their feature vectors. I think of these average multi-dimensional vectors as fingerprints that can be used to catch sneaky unwarranted malicious spam and give a probability based value as the classification. Alongside that, since we used 28 categories of websites to generate our warranted spam dataset, we can even break these average vectors fingerprints into category specific fingerprints and get finer granularity.

I didn't get to the categorical classification or sentiment analysis just yet but that in in the pipeline for next semester. I am looking forward to using the knowledge I have gained in this class to push this research further and hopefully get a new publication as well.

8.1 Conclusion

The ultimate goal of this research was to create a classification model that will be utilized in the step between when a user reports an email as spam and the flagged mail is used to update the spam filter model. Our classification model would take the user-flagged mail, classify it, and if it is classified as warranted spam it will not be used to update the spam filter ML model, but if it is classified as unwarranted spam, it will be used to updated the filter. In both cases the email provider will still provide rules to filter this flagged mail but the main spam filter will become more and more focused on truly malicious unwarranted spam and less biased on benign and annoying commercial correspondence. I believe we are on the path to making this goal a reality once the topics in the discussion are addressed.

9 THREATS TO VALIDITY

The integrity of our dataset relies on the websites used to amass warranted spam emails. While we sought diversity in our website selection, most were generated via ChatGPT4 (May 3 Version), so there's potential for geographical, cultural, or other biases that might not capture the global spectrum of spam characteristics. Ensuring the confidentiality of the dataset's email accounts remains crucial to prevent unintended external influences, which could adulterate the dataset. Our pursuit of automation also may introduce bias as our bot will have varying constraints, limiting its scope.

ACKNOWLEDGMENTS

This work was supported in part by funding from NSF under Award Numbers DMS 2123761, CNS 1822118, and from AFRL, ARL, Statnett, AMI, NewPush, and Cyber Risk Research and from NIST under Award Number 60NANB23D152.

We extend our gratitude to Evan Anspach, Alex De La Chica, and Tobias Beesley at Colorado State University, Jhin Echon and Adrian Castro Farias at San Diego State University, Emmanuel Gonzalez at California State University, Los Angeles, and the talented high school student Udhirna Krishnamurthy, for their invaluable contributions to the manual sign-up process, which played a pivotal role in the creation of this dataset.

We would also like to acknowledge Colorado State University for the resources and facilities provided by the institution.

Special thanks to the RaysCyberLab for their continuous guidance, time, insight, and expertise to enhance this research.

REFERENCES

[1] 2010. CSDMC2010 SPAM Corpus. http://csmining.org/index.php/spam-email-datasets-.html.

[2] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, George Paliouras, and C.D. Spyropoulos. 2000. Ling-Spam Dataset.

[3] G. V. Cormack. 2007. TREC07 Spam Corpus. In *TREC 2007 Spam Track*.

[4] B. Guenter. 1997-2023. Spam Archive. http://untroubled.org/spam/. Accessed: June 2021.

[5] Neil Kumaran. 2022. Google Spam Filter Description. https://workspace.google.com/blog/identity-and-security/an-overview-of-gmails-spam-filters.

[6] Scott Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. arXiv:1705.07874 [cs.AI]

[7] Vildan Mercan, Akhtar Jamil, Alaa Ali Hameed, Irfan Ahmed Magsi, Sibghatullah Bazai, and Syed Attique Shah. 2021. Hate Speech and Offensive Language Detection from Social Media. In *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*. 1–5. https://doi.org/10.1109/ICECube53880.2021.9628255

[8] V. Metsis. 2006. Enron-Spam Dataset.

[9] Apache SpamAssassin Project. 2005. SpamAssassin. https://spamassassin.apache.org/old/publiccorpus/.

[10] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv:1602.04938 [cs.LG]

[11] SpamScope. [n. d.]. mail-parser. https://github.com/SpamScope/mail-parser. Accessed: [Dec 5, 2023].

[12] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems* 33 (2020).