Stacked LSTM With Cross Attention Validation For Spam Detection

1st Daniel Sadig
Neural Networks (EE8204)

Markham, ON
daniel.sadig@torontomu.ca

***Abstract*—Spam emails are unrequited messages sent to a recipient, often with the malicious intent of acquiring sensitive information. By identifying standard modalities among various spam emails, many machine-learning techniques have successfully determined which messages are spam and which are not. Despite their success, modern approaches only look at the morphology or syntax of the text, as opposed to the semantic meaning. When trained on these low-level features, the morphology, and the syntax, it is easy for an adversary to create adversarial examples. An adversarial example is any alteration made to the original email that causes misclassification. Thus, this paper proposes a novel approach to identifying spam email based on sentiment analysis. Obtain a more meaningful text representation using a bidirectional encoded transformer (BERT) for word embeddings. For feature extraction, we will use a stacked long-term short-term memory encoder (LSTM) with a cross-attention layer to extract more semantically relevant features. Focusing on the semantic relevance of a corpus circumvents black-box's attacks. Our experiments show that using a pre-trained Bert word embedding coupled with a stacked LSTM result in a four percent increase in accuracy.**

I. Introduction

In the ever-evolving digital communication landscape, emails remain indispensable for interpersonal communication. As their popularity steadily rises, it has piqued the interest of malicious actors who seek to exploit the platform and its users. Adversaries intend to deceive a recipient into divulging sensitive information (i.e., username, password, or credit card details). These deceitful attacks are known as *phishing* emails [18], an attempt to obtain sensitive information from an entity while hiding one's true intent and identity. Phishing attacks aim to undermine the platform's integrity while simultaneously jeopardizing recipients' security. This facade has cost American companies $20.5 billion in 2022 [19], making spam detection a highly sought-after area of study. Spam detection is preemptively identifying and notifying recipients of phishing emails.

To mitigate the risk of phishing emails, the current literature has employed many supervised [6-10] and unsupervised [1-4] machine learning techniques. Although unsupervised learning techniques have obtained high accuracy for in-data distribution (i.i.d) [17], they are highly susceptible to adversarial attacks [14]. Adversarial attacks are variants of email features (i.e., characters, strings, images, or URLs) that cause misclassification. Since unsupervised learning techniques rely on clustering data to generate a probability distribution, it is easy for adversaries to create adversarial examples that distort these distributions. To counteract these attacks, looking at the sentiment of the features rather than feature-mapping them to obtain a probability distribution is essential. By utilizing supervised learning techniques for feature extraction and classification, we create a model that classifies a corpus based on sentiment rather than features.

This paper looks at ways to increase the accuracy of supervised spam detection algorithms by focusing on sentiment analysis instead of feature mapping. Sentiment analysis identifies semantic meaning from text, whereas feature mapping looks at the commonality between varying spam emails. Using a stacked Long Term Short Term Memory (LSTM) model to perform feature extraction, we can train the model on higher-level features such as syntax and semantics [11]. Stacked encoders encode different information at each level, with the bottom level being the most fundamental character representation of text and the top level being the most intricate, the semantic representation. To ensure more semantically relevant feature extractions, we will use a cross-attention layer to update the parameters of the lowest encoder, similar to what is seen in 1. However, unlike the SRCLA [11] approach, our model will generate word embeddings using a fine-tuned directional encoder (BERT) model, figure 2. BERT has yielded more accurate results than prior word embedding techniques [9].

II. Literature Review

Before the development of DNN's statistical approaches, such as Bayes' theorem [2] were used for email spam detection, Bayes theorem (1) was used as a part of the Naïve Bayes algorithm. Naïve Bayes algorithm [1] is a statistical approach that relies on certain features in an email to identify the email as spam or genuine.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{1}$$

Bayes theorem assumes words are independent of each other. Therefore, the Naïve Bayes algorithm starts by breaking text into unigrams/features; these unigrams are then given a number; this approach mimics that of a bag of words approach. To classify a text as spam, the equation (1) is performed; if

$P(Spam|Feature) < P(Non-spam|Feature)$, the text will be classified as spam. Statistical approaches, such as *Bayesian spam detection* [1], rely heavily on training data; thus, they cannot understand the semantic sentiment of the text. They are only helpful on in distribution data (i.d.d) but perform poorly on out of distribution data (o.d.d). Out-of-distribution data is categorized as any foreign colloquialism, (i.e., bi-grams or tri-grams), that are not part of the training data. This makes it easy for adversaries to perturb an email such that the model misclassifies it.

Harisinghaney, Dixit, Gupta, and Arora wanted to see how well statistical approaches such as KNN, Naïve Bayes classifier, and DBSCAN could incorporate both text and image features to identify spam [10]. K-Nearest Neighbours (KNN), is an unsupervised learning technique that looks at the similarity between features for classification. Naïve Bayes classifier uses the Bayes theorem 1 and a threshold to classify the corpses. Lastly, Reverse DBSCAN is a clustering approach that may reveal outliers that are indicative of spam mail. The objective of this research was to combine text and image features to obtain a multimodal representation of each email corpus. By doing this the authors hoped to achieve a broad range of spam characterization, alleviating the *false positive* results. Using the *Enron corpus's* dataset, they tested their hypothesis and realized they obtained high precision, sensitivity, specificity, and accuracy for all supervised learning techniques. Thus, a multimodal representation of emails reveals more relevant features and is useful for classification tasks such as email spam detection.

Since statistical approaches do not have a way to extract relevant features, they require a sizeable, unbiased training set. This means that training time is high and calculations at inference time are high because the algorithm can not filter out irrelevant features. The Bayesian therom was coupled with Support Vector Mahines (SVM's) to circumvent this issue. SVMs perform feature extraction of spam email datasets [3]. Also, SVM's provide more relevant features aside from just uni-grams, for instance SVM's look at the metadata, word frequencies and structural attributes of the email. Once the SVM performs feature extraction, we train the SVM to classify the selected features along a hyperplane. Then, using the Baysiean theorem, we find the conditional probability of each feature. Now, both the SVM and the Naive Bayes algorithm have separately classified the e-mail, we must coagulate there responses. To do this, an *ensemble approach is used*, an ensemble is a collection of independently trained networks that each produce a result. Inorder to reach a decision different voting mechanisms can by used (i.e., PAET [4], averaging, voting etc...), for this particular instance an averaging approch was used. We produced a final result by summing the probabilities of both the SVM and the Bayes classifier. If the final result exceeds the threshold, we can classify the email as spam.

Gao et al., devised an unsupervised learning method that creates a similarity graph based on online social network (OSN) wall posts [16]. The procedure consists of using semantic similarity to identify all mutually exclusive wall posts; this means the post has no affiliation association relevant to the user. Then, using behavioural cues, identify if the email/wall post is harmful or harmless. This technique is known as FBCluster [15]. To identify a malicious post/email, the SVM will look for data that is distributed or sent out in bursts or intervals. To see if the corpora are spam, apply behavioral filters; this filter looks for a given corpus's underlying intent or semantic meaning. Despite the initial success of this unsupervised learning technique, in 2013 Enhua et al., revealed that the FBCluster approach yields a high flase positive rate of 39.3%, meaning a lot of non-spam emails were getting marked as spam and therefore where never being read. Enhua et al., alleviated this issue by applying a *Sybil Defense-based Spam detection (SD2)*; this technique successfully separated Spam and non-spam classes.

Tak and Taposwi, proposed a seven-step query-based email spam detection approach using Artificial Neural Networks (ANN) [5]. Each mailing user has their own knowledge base comprising previous spam and non-spam emails, email senders, and server locations. Firstly, they would *analyze the mailing content*, using the equation shown in 2, if the resulting value was higher than the given threshold value S_t, the email is marked as spam. In equation 2 NM_p is the total number of exactly matched words of incoming spam with the p-th spam mail, N_p : total number of words in the incoming p-th mail, and P : the total number of recent emails with are available in the corresponding mailing list.

$$Matchingcontent = max.(NM_1/N_1, ..., NM_p/N_p) \quad (2)$$

If it is below the threshold, check if the *sender is within the trusted knowledge base*, if so mark the email as real. If not, check the *knowledge base for spam words, senders and sender locations*; each user has their own set of spam words, senders and sender locations. Lastly, use an ANN to check for the *missbehavious of incoming mail*, using an ANN to learn the complex patterns of mailing servers such that it can make intelligent and efficient decisions based on the incoming mail. Using cross-validation between the ANN and the aforementioned steps, we can verify the sender's legitimacy.

Sharma and Kaur realized that modern spam detection techniques have poor accuracy and precision. To mitigate this, they proposed a spam detection technique that used SVMs and a Radial Basis function (RBF). The RBF is a kernel function that maps data into a higher plane such that it becomes suitable for non-linear classification tasks. By mapping data points to a higher dimension, we allow the SVM to operate in a higher dimension without the need for complex transformed feature vector calculations. Thereby ensures reduced computational complexity and a more accurate decision-making process.

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \quad (3)$$

- $K(x, y)$ is the RBF kernel function.
- $\|x - y\|$ represents the Euclidean distance between the input vectors x and y.

- σ is a parameter that controls the width of the Gaussian distribution. It determines the influence of each training example on the decision boundary.

Equation 3 demonstrates how the RBF function can be used to map a 2-dimensional vector, (x,y), into a higher dimension. Therefore, once a spam word dictionary is generated, use an SVM to extract features of the spam emails/words, similar to [3]. Next, using the RBF method, map the features to a higher dimiesional space, in this space train the SVM to identify spam and non-spam features successfully. Sharma and Kaur noted that this method yielded more accurate and precise results than previous works.

Seth and Biswa designed a *multidimensional spam classification technique that utilized DNNs* [7].They realized that scammers use a combination of text, images, and other content to evade traditional filters. Using different scamming techniques, scammers have introduced unique discrepancies between spam emails, meaning they can no longer be linearly classified. Multimodal spam classification involves integrating information from different modalities, therefore, Seth and Biswa use DNN's [8] to extract complex patterns and representations from multimodal data. Thus, each email is represented as a combination of different modalities later combined into a single comprehensive representation. Using this data representation, they can train DNNs to learn intricate relationships between various types of content. This approach yields higher accuracy and precision, it also made it harder to create advesarial examples that would fool the model.

Lan, Hao, Xia, Qian, and Li, expressed concern for the security of machine learning techniques. Moreover, they devised a way to quantify how adversarial sampling impacts spam detection techniques [14]. The goal of an adversary is to attack the model's specificity such that it misclassifies an input. As previous works show, spam detection techniques concatenate several textual and visual features to form one vector representation of the corpa. Therefore, if an adversary can determine what features impact the model's decision, they can successfully create a false negative result. The model confidence will degrade if the adversary iteratively repeats this process (i.e., adversarial sampling). Their paper discusses quantifying the model vulnerability and the cost of a perturbation. A perturbation is any distortion of the original spam email. equation 4 show how to calculate the cost of an advesarial attack, x_i, is the original email, and $\hat{x}_i$ is the perturbed email, the larger the distance the greater the cost.

$$d(x_i, \hat{x}_i) = \sqrt{\sum_{k=1}^{n} (x_i^k - \hat{x}_i^k)^2} \tag{4}$$

let l be the number of perturbed features needed to generate $\hat{x}_i$ and d denotes the distance between the original and perturbed email. Then, the total cost equation is $C(x_i, \hat{x}_i) = (l, d)$. Quantifying the cost needed to misclassify an email shows how vulnerable current spam-detection approaches are.

Lan, Hao, Xia, Qian, and Li, proposed a stacked residual recurrent neural network (RNN) with cross-layer attention to extrapolate more semantic features from a corpous [11] wich they named **SRCLA**. Previous approaches focus on combining different features to create different modalities, however, this paper aims to distinguish features by there type. Stacked encoders encode different information at each

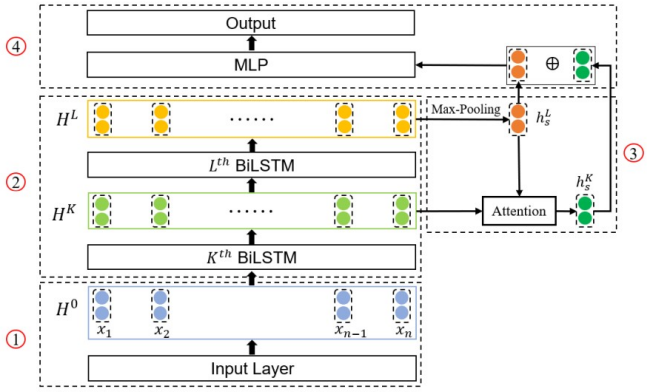

FIGURE 3. The model structure of SRCLA. ① indicates the word embedding; ② indicates the stacked residual BiLSTM; ③ indicates the Cross-Layer Attention; ④ indicates the MLP classifier.

Fig. 1. The model structure for SRCLA [11]

layer [12-13]; the bottom layer is the most straightforward representation of the corpus, the morphology, and the top layer is the more complex representation of the data, the semantic representation. Their proposition was by using the higher-level features, the semantic representation of the text, to train the lower-level features, the model would be able to extract more semantic relevance from corpora. Figure 1 shows their model; firstly, a word embedding technique is used to represent the corpora as a low-dimensional vector. Phase 2 comprises the stacked residual BiLSTM, where semantic meaning is extracted from the low dimensional vector, and next is the *cross attention layer*. This SRCLA design was tested on eight different datasets and achieved state-of-the-art performance on five of them.

In recent years Guo, Mustafaoglu, and Koundal, analyzed the impact of using a bidirectional encoder representation transformer (BERT) [9]. A transformer uses a self-attention mechanism to establish relationships between words in a sentence; A BERT transformer comprises a stack of encoder transformers. BERT has a unique classification token (CLS) which embed-eds a string of inputs, these tokens are then passed through a feed-forward network with a self-attention layer. This process results in the following word embeddings, figure 2; these word embeddings are then used as inputs into a classification algorithm. Guo, Mustafaoglu, and Koundal used a pre-trained BERT model to perform feature extraction on a given corpus. Once the word embedding was retrieved, this became a classification problem. The supervised classification techniques were trained on the provided feature set, the set contained from the BERT encoding. This approach yielded higher accuracy, precision and recall than all other previous supervised learning approches.

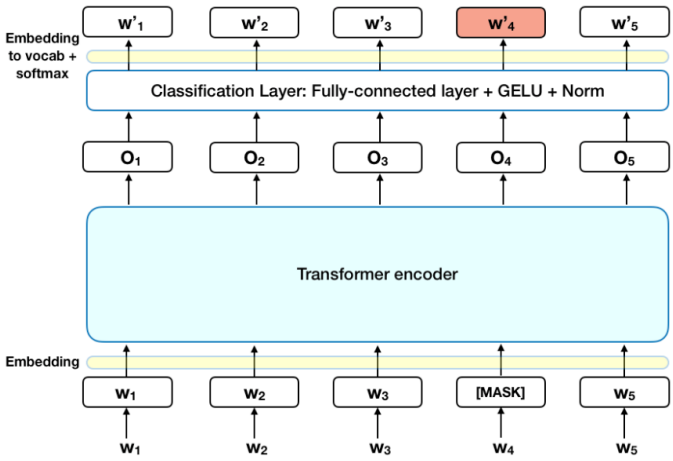

Fig. 2. Representation of BERT embeddings for NLP tasks

III. Dataset

For our dataset, we will use a *Spam Mails dataset* [20] from Kaggle. Figure 3, shows two sample entries in the dataset. It consists of a total of 4993 unique entries, 71% of the entries are marked as *'ham'*, meaning they are not spam emails, the remaining 29% are *'spam'* emails. The target column is the *'#label_num'* column, where 0 indicates *'ham'* and 1 indicates *'spam'* To separate the data into their respective training and

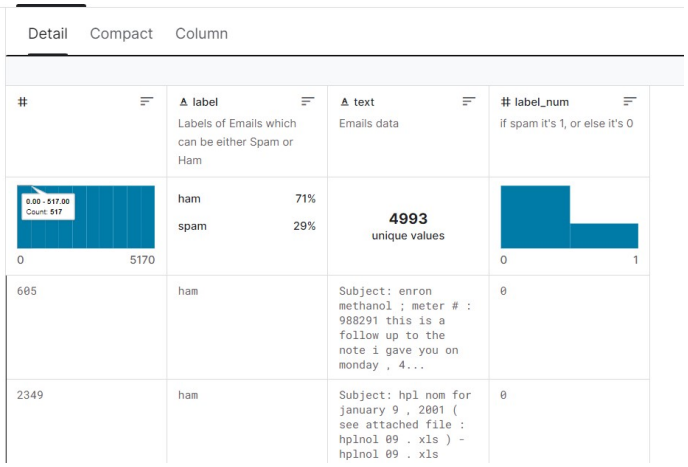

Fig. 3. Spam Mails dataset from Kaggle

testing sets, we will use the python *SKlearn* to set 70% of the data as training and 30% as testing

IV. Problem Statement

Spam emails substantially threaten their recipients' efficiency, privacy, and security. The existing paradigms rely on low-level feature mapping techniques [1-10] for spam detection, which are susceptible to adversarial sampling. This paper aims to enhance the robustness of spam-detection models, mitigating the risk of adversarial attacks. Our proposed model leverages a stacked LSTM with a cross-attention layer, allowing us to obtain high-level feature extractions (i.e., sentiment analysis).

V. Methodology

Spam detection techniques are comprised of three main steps:

- **Word Embedding**: Representing copra as vectors that are easily understood by the computer.
- **Feature Extraction**: Identifying and categorizing a list of features related to spam text
- **Classification**: Classifying an input as spam or not spam using the outputs generated in the feature extractions.

For our word embedding we will use a fine-tuned BERT model. Unlike other word2Vec approaches, BERT generates contextualized word embedding, figure 2, meaning the vector representation for any given word is influenced by the context of the sentence. Guo, Mustafaoglu, and Koundal provided that these contextual word representations allow the model to better identify spam and non-spam text [9]. Although using a BERT model can slow down computations at inference time, the increase in accuracy is worth the trade-off.

For Feature extraction, we use a stacked LSTM with a cross-attention layer, figure 4. Using a stacked LSTM allows us to capture hierarchal patterns and more complex dependencies [12]; each layer provides a more complex representation of the word embeddings. The initial layer is the most basic representation of the text and morphology, whereas the final layers capture the semantic relevance. Using a cross-attention layer, we can train the lower levels on the semantic representation generated by the higher levels in the stack. Thereby capturing more semantically relevant features.

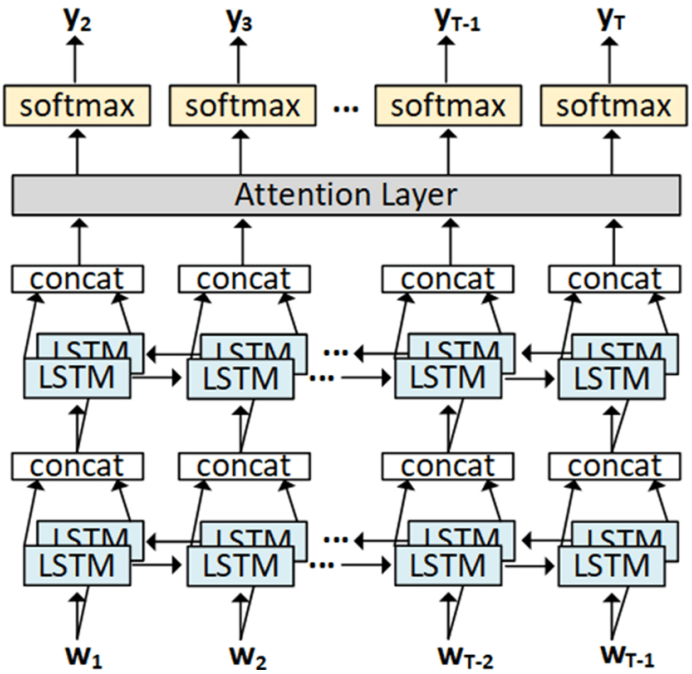

Fig. 4. Stacked LSTM with a cross attention-layer

Once we have embedded the text and extracted the relevant features, we must map them to a given plan. To classify the inputs as spam or non-spam, we will use a KNN, clustering-based algorithm. As previously stated, non-supervised learning approaches are good at classifying the text as spam or not providing the features extracted adequately represent our input data. So, by using a BERT encoder and a stacked LSTM with a cross-attention layer, we can generate the semantically relevant features, thereby mitigating the risk of adversarial examples.

To Evaluate our results we will use three metrics: (1) accuracy equation 5, (2) precision equation 6, and (3) recall equation 7.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100\% \quad (5)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}} \quad (6)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} \quad (7)$$

VI. Evaluation

In our evaluation, we analyze how using a Bert textualized word embedding coupled with a stacked LSTM network compares to that of the Naive Bayes classifier [2].

A. Baseline

Before being passed into the baseline model, the data, i.e., the emails, were pre-processed using *stemming* and a *bag of words* approach. Stemming is a normalization technique that produces one term for an aggregation of like terms (i.e., ran, running, or run would be simplified to run). To do this, a dictionary of common words is needed. To remedy this issue, we use the **NLTK** package in Python; this package is comprised of many like terms that our model can exploit. In order to pass the text into our model, we must represent the text as vectors (i.e., tokens); tokens are mathematical representations of text that a neural network can interpret. To do this, we utilize a bag of words approach; this approach creates a token representation for each word in our corpus. Lastly, to simplify the task of the classifier, we use a TF-IDF, equation (10), term frequency-inverse document frequency, to categorize the *importance* of each word in a given corpus.

$$TF(t) = \frac{Numberoftimestermtappearsinadoc}{Totalnumberoftermsinthedoc} \quad (8)$$

$$IDF(t) = \log_e \frac{TotalNumberofDocuments}{Numberofdocumentswithtermtinit} \quad (9)$$

$$TF - IDF = TF(t) * IDF(t) \quad (10)$$

Importance, of term t, indicates how rare the word is; for instance, common phrases like and, the, more, and if, would yield a lower TF-IDF score as they appear frequently in both spam and non-spam text. However, non-common phrases that frequently appear in one category but not the other would yield a higher TF-IDF score. The TF-IDF representation of our corpa is then based on our Naïve Bayes (1). Statistical approaches such as these rely on the presence of certain features [2] to categorize corpora, which makes them vulnerable to white-box attacks. That is why our state-of-the-art approaches attempt to grasp the underlying context of the corpora rather than categorize it based on feature extraction.

B. State-Of-The-Art Approach

Unlike traditional statistical approaches, which rely on feature mapping for classification. Our approach looks at the semantic relevance of the text and aims to classify the corpus based on the semantics. The first step is to generate embeddings based on the semantic relevance of text within a given sentence. To do this, we used a BERT encoder 2 since BERT has been shown to produce more meaningful full embeddings [9]. Next, we used a stacked LSTM structure similar to that of [11]. A stacked LSTM has been shown to encode different representations at each stack. As seen in 1, the bottom layer captures the morphology, while the top layer captures the semantic relevance. However, we use a bi-directional LSTM for the first layer to improve this approach. Bi-LSTMs can use information from both directions of the text; this allows the model to generate more useful semantic representations of text. Although the bottom layer only captures the morphology, all the other layers rely on it to generate their embeddings. In essence, it is the most important of the three layers, so we decided to enhance it using a bi-LSTM. This approach aims to categorize emails by the semantic relevance of text, such that the model is more robust and accurate.

C. Results

Models	**Metrics**		
	Accuracy	**Precision**	**Recall**
Naïve	90%	1.0	0.63
SOA	94.3%	0.9	0.96

TABLE I
Performance Results

Table I shows the comparison between our state-of-the-art approach against that of the traditional statistical approach. We measured accuracy and recall using equations (5) and (7), respectively; we see that our state-of-the-art approach yields better accuracy and precision for this given dataset. However, when we measure precision (6), we see that the traditional statistical approach yields better results.

There is a 4% increase in accuracy, which means that our model has successfully understood the semantic relevance of each corpus and what makes it spam or not. This understanding is indicative of a more adversarial-robust model. Since the model does not look for different modalities in the text and classify them based on those modalities, an adversary must understand the model hyperparameters to craft a successful adversarial attack. Also, the statistical approach yields a low recall score, indicating that many spam emails

have successfully fooled the model. In contrast, the state-of-the-art approach has a near-perfect score for both precision and recall.

VII. Conclusion

In conclusion, our work focuses on extracting the semantic relevance from a given corpus and classifying it into one of two categories: spam or not spam. Our approach varies from previous techniques as it utilizes a BERT encoder to generate embeddings before passing it to the stacked bi-lstm structure for further refining and classification. Using a bi-lstm in the first layer, we aim to extract more semantically relevant embeddings, which are later fine-tuned in the upper layers of the stacked lstm structure. Our state-of-the-art approach has achieved a 4% increase in accuracy than the traditional statistical approaches. Also, it does not rely on feature mapping, which makes it harder to generate successful perturbations that would fool our model and cause misclassification.

VIII. Future Work

To better evaluate the model's success, we must compare it against the **SRCLA** [11] approach for various datasets. Also, we want to quantify the adversarial robustness of our model by generating text perturbations and seeing how many adversarial examples can fool our state-of-the-art approach.

References

[1] J. J. Eberhardt, "Bayesian spam detection," Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal, vol. 2, no. 1, Mar. 2015.

[2] M. A. Fattah, "A novel statistical feature selection approach for text categorization," Journal of Information Processing Systems, 2017. doi:10.3745/jips.02.0076 .

[3] N. Kumar, S. Sonowal, and Nishant, "Email spam detection using machine learning algorithms," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Jul. 2020.

[4] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. "Semi-supervised knowledge transfer for deep learning from private training data". In: International Conference on Learning Representations(2016).

[5] R. Sharma and G. Kaur, "E-Mail Spam Detection Using SVM and RBF.", International Journal of Modern Education & Computer Science (2016).

[6] G. Kumari Tak and S. Taposwi, "Query Based Approach towards Spam Attacks Using Artificial Neural Network", International Journal of Artificial Intelligence and Application, (2010).

[7] S. Seth and S. Biswas, "Multimodal spam classification using deep learning techniques," in 2017 13th International Conference on SignalImage Technology & Internet-Based Systems (SITIS), IEEE, (2017).

[8] R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning. ACM, (2008)

[9] Y. Guo, Z. Mustafaoglu, and D. Koundal. "Spam detection using bidirectional transformers and machine learning classifier algorithms." Journal of Computational and Cognitive Engineering 2.1 (2023)

[10] A. Harisinghaney, A. Dixit, S. Gupta, A. Arora "Text and image-based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm." 2014 International Conference on Reliability Optimization and Information Technology (ICROIT). IEEE, (2014).

[11] Y. Lan, Y. Hao, K. Xia, B. Qian, and C. Li, "Stacked residual recurrent neural networks with cross-layer attention for text classification, "IEEE Access, vol. 8, pp. 70401–70410, (2020),

[12] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol., (2018),

[13] A. Søgaard and Y. Goldberg, "Deep multi-task learning with low-level tasks supervised at lower layers," in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, vol. 2, (2016)

[14] H. Shirazi, B. Bezawada, I. Ray1, and C. Anderson, "Adversarial Sampling Attacks Against Phishing Detection," 33th IFIP Annual Conference on Data and Applications Security and Privacy (DBSec). Springer International Publishing, (2019)

[15] T. Enhua, G. Lei, C. Songqing, Z. Xiaodong, and Z. Yihong, "Unik: unsupervised social network spamdetection," In: Proceedings of the 22nd ACM International Conference on information & knowledge management, (2013)

[16] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, "Detecting and characterizing social spam campaigns," In Proc. of IMC, (2010).

[17] T. Nallamothu, Phani, and M. Khan. "Machine Learning for SPAM Detection," Asian Journal of Advances in Research, IEEE (2023)

[18] Y. Zhang, Y. Xiao, K. Ghaboosi, J. Zhang, and H. Deng. "A survey of cyber crimes. Security and Communication Networks," (2012)

[19] N. Cveticanin "What's on the other side of your inbox - 20 spam statistics for 2022," (2022)

[20] Venky73. "Spam Mails Dataset," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/venky73/spam-mails-dataset.