Transformers against SMS spam

Mitch Abdon
University of California, Berkeley
mitchabdon@berkeley.edu

Tanya Flint
University of California, Berkeley
tanyaflint@berkeley.edu

Abstract

This study is motivated by the rampant *smishing* attacks in the Philippines. The SMS messages offer dubious jobs, products, and cash deals attempting to scam victims into providing sensitive information. SMS messaging remains an important medium of communication in developing countries, thus effective filtering of SMS spam messages could have wider social impact. Despite limited local-specific training data, our pre-trained transformer-based language model achieved f1-score 0.9805 and accuracy 0.9920. The languid response in addressing spam SMS is not of lack of available technology but rather on weak incentives due to scale and profits. Regulation therefore has a crucial role to plug this market failure. This could include institutionalizing SIM registration and incentivizing industry collaboration.

1 Introduction

In 2021, rampant short-message-service (SMS) messages offering dubious jobs, products, and cash deals led to investigation in the Philippines telecommunications industry (NPC, 2021). More than just annoying, the SMS spams are threats to data privacy and potential source of serious monetary losses as most are smishing attacks that attempt to scam victims into providing sensitive information and bane to legitimate advertisers. And, with reports that international crime groups were behind the scams, this could also be elevated as a national security concern.

Despite drives to block SMS spam (Calonzo, 2022; Yap, 2021), regulators admit the limited capacity, despite the Data Privacy Act, to trace the source of spam SMS, particularly in the absence of mandatory SIM card registration in the country (Dumlao-Abadilla and Gascon, 2021). As of writing, we continue to receive unfiltered spam messages.

SMS messaging or texting remains an important medium for communication, especially in developing countries. Texting via mobile network is still widely used due to limited coverage of broadband service that make internet-based messaging apps (e.g., Watsapp, Viber, Telegram) inaccessible or more expensive. In the Philippines, 73% still prefer to use SMS despite the trend in the rest of Southeast Asia is declining (Wavecell, 2019; Interactive, 2016). Moreover, market studies find SMS more engaging, i.e., consumers are more likely to open, read, and react to SMS than emails thereby offering a better marketing platform (Pemberton, 2016). Thus, effective filtering of SMS spam messages could have wider social impact by minimizing smishing attacks before they could reach and be read by potential victims. But why has there been not much action against SMS spam compared to email spam?

We use a unique set of data (Section 2.2) that not only increases the number of spam samples, making the training data more balanced, but also allows the models to be trained on local data. We implemented three deep learning model architectures on this data (Section 3): Convolutional Neural Network (CNN), our baseline model; Long- Short-Term Memory (LSTM); and Bidirectional Encoder Representations from Transformers (BERT). The performance of these models are measured against accuracy and f1-score (Section 4). We also compare these results with selected recent studies on SMS spam classification (Section 2.1).

2 Background

Spam SMS are similar to spam emails in that both are unsolicited and unwanted and mostly with the intent to scam recipients to reveal sensitive information for monetary gains. In addition, both are free or relatively very cheap services with a wide

consumer reach. But SMS and emails also differ in a number of ways, in particular the business models and the scale at which their service providers operate.

The email service market is essentially global with Google (Gmail) and Microsoft (Outlook) together with 63% of market share on the business side (email hosting) and Gmail dominates as the most popular email provider among the general population with 1.8 billion users or 45% share (Press, 2022). On the contrary, SMS are provided by country-based telecommunication networks providing mobile communication services domestically, e.g. AT&T, Verizon, T-Mobile in the US and Smart and Globe in the Philippines.

Because the main business of email service providers is hosting, there is a strong incentive to invest in research and infrastructure to develop sophisticated filtering of unwanted and potentially harmful emails for their clients. Google, for example, reportedly detects and filters 99.9% spam and phishing emails (Dada et al., 2019). Can this level of accuracy be achieved for SMS spam as well? Maybe but unlikely be provided by mobile service providers on their own as they profit from aggregators sending messages in bulk, whether spam or ham.

2.1 Related work

The structure of text messages fundamentally differ from emails, thereby posing additional challenges in discriminating spams from hams. For one, SMS messages are short at 160 characters, while emails can be anywhere from a one word "hello" to an essay long letter. For another, text messages are generally informal and conversational. The former limits the context that can be embedded in the model and the latter further abstracts the language for machines, for instance from the use of slang, idioms, and local dialects.

In Table 1, we present the performance, based on accuracy (Acc) and f1-score, of various models used in select recent studies classifying spam SMS. Whenever available, we report the macro f1-score as our interest is in the minority class (spam). All studies included in Table 1 used the same UCI data that we augmented and employed in this paper, which is described in Section 2.2.

Jain et al. (2022) implemented traditional machine learning (ML) classification algorithms. Their results show that Support Vector Machines (SVM) performed the best, followed by Logistic Regression (LR), and then Naive Bayes (NB). Raga and L (2022), in addition to traditional ML algorithms, included one deep learning method, LSTM. They find that, among the traditional ML classifiers, NB performed the poorest with respect to accuracy and SVM performed the best, consistent with Jain et al. (2022). However, the deep neural network-based LSTM architecture outperformed all else with accuracy of 98.5% and f1-score of 93.7%, albeit slightly lower that what Jain et al. (2022) achieved with SVM.

Study	Model	Acc	f1
Jain et al. (2022)	SVM	98.8	95.4[a]
Raga and L (2022)	LSTM	98.5	93.7[b]
Jain et al. (2019)	LSTM	99.0	99.2[a]
Roy et al. (2019)	CNN	99.4	98.9[c]
Rifat et al. (2022)	DBERT	98.7	97
Tida and Hsu (2022)	BERT	98	94.0[a]

[a] Not specified if macro or weighted.

[b] Calculated from reported precision and recall.

[c] Calculated average of reported class f1-scores.

Table 1: Survey of recent literature

Roy et al. (2019) noted the increasing use of deep neural networks for spam filtering. Jain et al. (2019), for example, used the LSTM architecture with pre-trained word embedding layers from Word2Vec, WordNet, or ConceptNet and achieved 99.0% accuracy and 99.2% f1-score, the highest in Table 1. The higher f1-score reported by Jain et al. (2019) compared to accuracy, while not impossible, is curious but could also imply that the model, despite the skewed dataset, performs well in predicting the minority class as well.

Raga and L (2022), Jain et al. (2022), and Jain et al. (2019) did not account for the highly imbalanced data with 87-13% ham-spam ratio. Also, Raga and L (2022) did not explicitly report f1-scores, which is more relevant in this case. The ones reported in Table 1 are calculated based on reported precision and recall.

Roy et al. (2019), on the other hand, addressed the data imbalance by oversampling the minority class, in this case spam SMS, or via Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2011). Using a synthetically balanced data, Roy et al. (2019) achieved 99.4% accuracy with CNN architecture and 96.8% with LSTM.

More recently, researchers have began using transformer neural network architectures for filtering spam SMS, such as Rifat et al. (2022) and Tida and Hsu (2022) that used some version of pre-trained BERT. But both made no adjustments to account for data imbalance. Rifat et al. (2022) added calculated features, such as number of words and message length. Tida and Hsu (2022), on the other hand, fine-tuned their model by adding dropout layers, batch normalizations layers, etc. These BERT-based models, however, performed poorly when compared to the reported performance of the LSTM and CNN architectures in Table 1.

Results from previous studies (Table 1) show that while neural network architectures perform better than traditional ML algorithms, there is still no consensus on which type of architecture and what modifications work best in filtering SMS spam. This study builds on these rich SMS classification literature and offers a further exploration of deep neural network architecture and leveraging the pre-trained layer and self-attention technique to produce context-based text embeddings, such as BERT (Devlin et al., 2018), instead of a static text representations such as Word2Vec. The two differences that make this study unique is combined CNN architecture with pre-trained BERT and expanded data with more SMS spam samples.

2.2 Data

This study uses the same benchmark data set from the UCI Machine Learning Repository (UCI) used in most SMS classification studies. UCI is a set of 747 spam and 4,827 ham text messages collected from various sources (Almeida et al., 2011). One challenge with this data is that this is not specific to the Philippines, for which we intend to apply the results, and that the messages in it are over a decade old. This paper is unique in using an expanded set of data with current SMS spam collected in the Philippines.

As mentioned above, text messaging is generally informal and could include local-specific language, idioms, and slang. Thus, the model must be trained on these as well. To address the lack of country-specific data, we crowdsourced spam messages via a Google Form to augment the training set with local samples. We have collected 533 spam messages, after deduplication, from the Philippines (PH) in June 2022. In total, we have 6,107 samples of 1,280 spams and 4,827 hams split into 4,885 training set and 1,222 test set (Table 2). This expanded data brings the spam-ham ratio to 21-79%, an improvement from 13-87%, albeit still imbalanced. In one of our experiments, we try to account for data imbalance to see whether this would improve performance.

Source	Spam	Ham	Spam+Ham
UCI	747	4,827	5,574
PH	533	-	553
UCI+PH	1,280	4,827	6,107
Train	1,017	3,868	4,885
Test	263	959	1,222

Table 2: Distribution of spam and ham

Another limitation is that we have only crowdsourced for spam messages and would have to rely entirely on the UCI data for ham data which are in English and collected over a decade ago. While this may be problematic, the structure of actual conversations may not have changed drastically. Moreover, the influence of Americans in Filipino culture and language from their 48-year occupation during the first half of the 21st century lingers through today. For example, only a fifth of the PH messages are in Filipino based on Google's language detection algorithm. Nevertheless, the presence of some Filipino texts adds to the complexity in modeling. To mitigate this issue and see if this makes a difference, we implemented our models on auto-translated PH spam text into English using Google Translate.

We also note the potential bias introduced into the data from the way it was collected. The additional spam data were collected via a Google Form sent to personal networks of one author who is based in Manila and was re-shared to their own networks. Each respondent may send multiple submission and each form submission may contain up to 5 spam messages that are self-reported by the recipient thus dependent on what the respondent consider as spam.

2.3 Sample messages

Enumerated below are a couple of examples of spam from the UCI dataset (A), another set of examples from the PH dataset (B) and a set of hams (C).

A. Spam (UCI)

1. *Congratulations - Thanks to a good friend U have WON the £2,000 Xmas prize. 2 claim is easy, just call 08712103738 NOW! Only 10p per minute. BT-national-rate*

2. *Guess what! Somebody you know secretly fancies you! Wanna find out who it is? Give us a call on 09065394973 from Landline DATEBox1282EssexCM61XN 150p/min 18*

B. Spam (PH)

1. *I am a SHOPEE hiring manager. You are invited to become a full-time employee. h0me work. Daily salary: 3000-9999P. contact WS: https://bit.ly/3GoGHdt*

2. *Para sa limitado lamang mkha ng 5000p bonus Magister n po kyo. CLICK where.name/fgv*
[Google translation: For only limited mkha of 5000p bonus magister n po kyo. CLICK WHERE.NAME/FGV]

C. Ham (UCI)

1. *Where r e meeting tmr*?

2. *HIYA COMIN 2 BRISTOL 1 ST WEEK IN APRIL. LES GOT OFF + RUDI ON NEW YRS EVE BUT I WAS SNORING.THEY WERE DRUNK! U BAK AT COLLEGE YET? MY WORK SENDS INK 2 BATH.*

From the sample texts above, we observe that spam messages from UCI and PH both attempt to lure recipients to respond in return for some reward. One glaring difference is the means through which the recipient would respond—UCI spams encourage the recipient to call, while the PH spams to click—reflecting changing preferences over time.

Ham messages, on the other hand, are conversational, as expected. We also observe the limitations of auto-translation where the input *words* are heavily truncated.

3 Methods

In this study, we used the pre-trained transformer-based language model BERT combined with the CNN (BERT+CNN) architecture on the expanded SMS data described in Section 2.2, and compared its performance with results from BERT with a fully connected neural network (BERT+FCN) and simpler deep learning methods such as CNN (baseline) and LSTM architectures with Word2Vec embeddings.

In the BERT+CNN model architecture we used the full sequence of the last hidden vector from BERT, excluding the [CLS] and [SEP] tokens as input for the CNN model with multiple convolution layers, global max pooling layer, and a dense fully connected layer with dropout.

Our task is different than the original Language Model task that BERT was trained on and we intended to retain more semantic information of the SMS text input so we chose to use the full hidden layer output in this case since the CNN architecture may handle the complexity better by slicing the input into kernels. By using the entire hidden sequence we can disregard the first [CLS] token that contains information about the hidden layer and we also take out the last [SEP] before passing the output into the convolution layers.

After the global max pooling layer, we added a fully connected layer with dropout before the classification. Previous work has also found it optimal to use this regularization parameter to reduce complexity between links in the fully connected layer of the model (Roy et al., 2019).

The BERT+FCN model is similar in architecture but uses the [CLS] token pooler output as an input for the dense fully connected layers. The pooler output is essentially the [CLS] token that contains the last hidden layer information from BERT with additional linear layer and Tanh activation function. [CLS] token is generally used for classification problems and we found the pooler output of BERT sufficient in this architecture by reducing some complexity that we pass into following fully connected layers. The dense layer and dropout is followed by a classification layer.

The difference between our CNN architecture in the BERT+CNN and our baseline CNN model is that we use two fully connected layers. For the most part the CNN architecture follows the BERT+CNN architecture.

LSTM model follows a simple architecture with RNN embedding layer passing the last output from the RNN calculation into an LSTM layer. Followed by a single fully connected dense layer and classification layer.

Table 3 provides a summary list of the models implemented in this paper. These are further ex-

plained in Sections 3.1 and 3.2.

Models
Baseline
1. CNN
LSTM and basic BERT
2. LSTM
3. BERT + FCN
4. BERT + CNN
Finetuning BERT
5. BERT + FCN 3e
6. BERT + FCN 3e un
7. BERT + FCN 5e un
8. BERT + CNN 3e un
9. BERT + CNN 3e un kernel
10. BERT + CNN 3e un filters
Cross Validation
11. BERT + FCN 1e un
12. BERT + FCN 1e 5e un
13. BERT + CNN 3e un kernel
14. BERT + CNN 3e un kernel 160len

Note: e-epoch; un-unfreeze; 160len-max_length=160.

Table 3: Models

3.1 Fine-tuning BERT

We fine-tuned the base-cased BERT models by adjusting the number of epochs, kernel sizes, filters and training on all BERT layers and see whether these will improve performance.

Epochs. For this text classification task we had to adjust number of epochs without overfitting as the text is shorter and may be less complex. We found that running models on more than 3 epochs led to overfitting in most cases. BERT with fully connected network gets incrementally better on 5 epochs whereas BERT with CNN architecture needed less passes through the network.

Kernel sizes. The BERT with CNN model was most sensitive to adjustment in kernel sizes. We saw an improvement after adjusting the sizes from [3, 5, 10, 20] to [2, 3, 4, 5]. The choice for the number of kernels were influenced by previous work on this task using CNN with static embeddings (Roy et al., 2019). Most text messages are short and we theorize that kernel sizes of 10 and 20 were least effective as the kernel window is to wide. This step increased the f1-score by a couple points.

Filters. Increasing the number of output filters in the convolution didn't make a difference and we're seeing similar classification reports. Indicating that limiting the number of filters is sufficient for this task.

BERT layers. In the BERT model there are 12 layers that we kept set to the default pre-trained weights in our baseline models. We saw an improvement without much overfitting when training on the entire network. This may be explained by the fact that SMS corpus is different than the original BERT training text corpus with more slang and abbreviations that require fine-tuning BERT layers.

Dropout. We used a dropout parameter of 0.3 to reduce the complexity of the fully connected layer.

3.2 Experiments

We experimented on 3 subsets of our expanded data, that is, we trained the data first on just the original UCI (UCI), second on the expanded data as collected (+PH), and third on the expanded data where the Filipino texts were auto-translated into English (+PHTr) before training.

To account for possibility of removing complexity by subsetting the UCI dataset with crowdsourced texts. We took steps to check for bias that may potentially introduce a signal that is solely based on the way we collected the data. For that reason we ran all our models on the UCI data alone and compared performance. While we expected the performance to be slightly under our expanded data we checked for any drastic differences in performance.

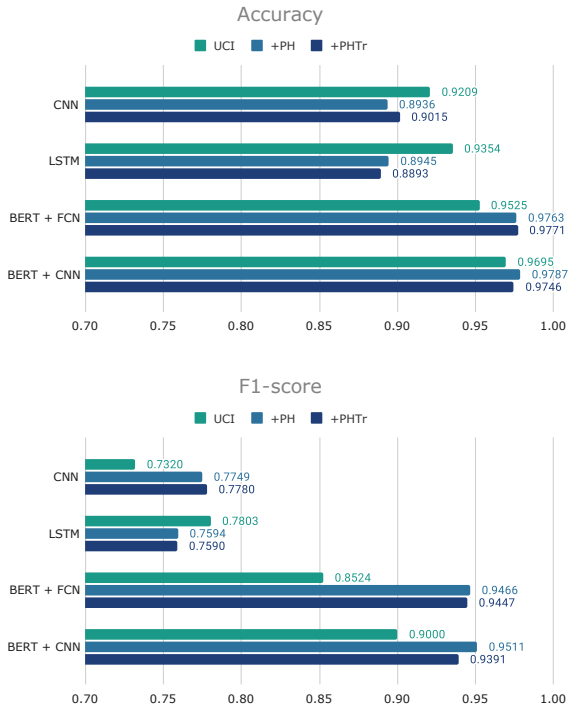
We also attempted to account for the imbalanced data by introducing class weights into the models such that the classifier will put more attention to the fewer spam samples. The calculated weights for spams is 0.63 and for hams is 2.40, consistent with the 21-79% sample distribution.

We present and discussed the results of our base model and modifications in Section 4 below.

4 Results and discussion

Figure 1 shows the performance from our baseline CNN model, LSTM, and the base BERT models. Accuracy (top panel) and f1-scores (bottom panel) are presented for the different subset of data they were trained on. The best performing model is BERT+CNN using the expanded but untranslated data (+PH) in both f1-score (95.11%) and accuracy (97.87%), and the least performing is LSTM on translated messages (+PHTr).

Figure 1: Baseline CNN, LSTM, and basic BERT



We observe also the following:

1. The transformer-based BERT models outperformed CNN and LSTM models, both in accuracy and f1-scores;
2. The additional spam data improved the accuracy of BERT-based models but deteriorated that of CNN and LSTM;
3. f1-scores imply that the transformer-based models are less sensitive to imbalanced data;
4. Auto-translation only marginally improved the accuracy of CNN and BERT+FCN and reduced the f1-scores of all except CNN.

Transformer-based models worked better compared to CNN and LSTM architectures is in part due to the the differences in the vocabulary. BERT uses WordPiece (or subword) tokenizer compared to the word-level tokenizer used in Word2Vec. For another, the Word2Vec embeddings we used with CNN and LSTM, in contrast to BERT embeddings, are independent of context, that is, it represents the same word with the same embedding vector regardless of context. Thus, the BERT Wordpiece tokenizer may have covered new vocabulary, even those in the raw messages with Filipino texts, that are not in Word2Vec. Nevertheless, the presented results are before tuning and we assume these results are expected to improve after all models are tuned. In this study we focused on BERT performance.

Table 5 presents the results after model fine-tuning and cross-validation. The BERT+CNN (3e un kernel) on translated data achieved the best f1-score at 0.9805±0.0054 with accuracy 0.9920±0.0023 (Table 4). Adjusting for class weights, the same model performed the best, but slightly less, with f1-score 0.9764±0.0084 and accuracy 0.9902±0.0037 (not reported in Table 4). The results from fine-tuned models (Tables 6 and 7) are included in the Annex section.

Model	UCI	+PH
BERT + FCN		
5e un	0.9924	0.9898
BERT + CCN		
3e un kernel	0.9933	**0.9920**
3e un kernel 160len	-	0.991

Note: e - epoch; un - unfreeze layers

Table 4: Cross Validation: **Accuracy**

Model	UCI	+PH
BERT + FCN		
5e un	0.9712	0.9753
BERT + CNN		
3e un kernel	0.9742	**0.9805**
3e un kernel 160len	-	0.9782

Note: e - epoch; un - unfreeze layers

Table 5: Cross Validation: **f1-score**

Our best performing models (BERT+CNN 3e un kernel and BERT+FCN 5e un) showed only marginal differences in accuracy (Table 4) between using UCI and the expanded +PH. With BERT+FCN 5e un model showing no difference in f1-score, where the BERT + CNN 3e un kernel model was more sensitive and showed an improved f1-score on the full data by a single point (Table 5). To take a closer look at how our additional data may be different from the UCI data we analyzed BERT tokenizer. The expanded +PH data showed more text messages that contain the [UNK] token, 12 SMS messages in the +PH data and only 1 message in the UCI data with an [UNK] token. This is

a token that is not in the vocabulary and cannot be converted to an ID. Overall these may be insignificant for classification and gaining a diverse, larger and balanced dataset may be a better trade off. Visually inspecting the data also hints at that SMS messages in the +PH data are localized and more complex. There are examples of spam messages that mix characters and numbers such as word "job" is spelled "j0b" with a zero. We assume that's done to pass spam filters. The *innovations* in the structure and content of spam SMS to avoid filtering algorithms is consistent with what was observed in spam emails. For example, the use of images rather than texts. This implies the need for continuous review and updating of spam classification models.

The UCI data alone consistently showed higher accuracy in spite that more SMS messages got misclassified and reported consistently lower f1-score compared to our expanded data. When evaluating our model we reported and compared cross-validated f1-scores over accuracy scores as it may not communicate the correct results. The findings lead us to believe that expanded data is remediating the data imbalance. We tried training on just the UCI data and testing on the expanded +PH data and the f1-score is 2ppt lower than our regular model, which makes for a good argument that UCI data is not sufficient to make predictions in other settings, such as the Philippines.

We found examples where the auto-translation did not work when the language used is not Filipino (or Tagalog) but other Philippine languages:

- *Daghang salamat sa suporta ug pagsalig nga inyong gihatag sa among pamilya. Sinsero ug tinuod nga pagserbisyo ang among ibalik kaninyo. -Sara, Pulong ug Baste* [Thank you for the support and trust you gave to our family. We will give in return our sincere and honest service. -Sara, Pulong, and Baste]

This was incorrectly predicted as ham message by one of the models (e.g., BERT + CNN 3e un), but the best model correctly classified it. This is an example of a model potentially having bias toward classifying messages as ham messages. Because the majority class is ham we're consistently seeing precision being much higher than recall. In particular when looking at messages that got misclassified, BERT + FCN and BERT+CNN models are similarly classifying messages, and the list of messages that are incorrectly labeled are almost identical with slight differences. There are some that BERT+CNN does better at classifying, for example:

- *1Apple/Day=No Doctor. 1Tulsi Leaf/Day=No Cancer. 1Lemon/Day=No Fat. 1Cup Milk/day=No Bone Problms 3 Litres Watr/Day=No Diseases Snd ths 2 Whom U Care..:-)*?

It is possible that the CNN layers are picking up on character patterns better than the dense fully connected layers, and these types of patterns maybe better recognized by CNN layers because the single kernel window is only looking at that pattern one step at a time. In general, BERT + CNN seems to be producing a more balanced classification report with precision and recall being almost equal.

5 Conclusion

Among the SMS spam classification models we implemented, the BERT+CNN on translated data is the best performing model with f1-score 0.9805 ± 0.0054 and accuracy 0.99201 ± 0.0023. This is comparable with existing estimates. However, it is not at par with what was achieved in email filtering. Data collection is key for this task, email spam classification probably performs that well because it's trained on millions of examples.

Further work in this domain will require a more robust data collection and labeling. In the future, for disparate data collection, one can explore the similarity between messages in the data using Sentence-BERT to pick out outliers or look at clusters in more detail. Exploring other models aside from *bert-base-cased*, such as *bertweet* could be an interesting option as it was trained on short tweet sized messages that include more slang and idioms.

In this paper, we demonstrate that existing natural language processing models, specifically neural networks including transformers, perform well in identifying spam SMS received in the Philippines, despite limited local-specific training data.

We argue therefore that the limited effort in addressing spam SMS is not of lack of available technology and methods but on weak incentives for networks to act on them due to scale and profits. Regulation therefore has a crucial role to plug this market failure and minimize the need for enforcement, which is costly. This could include institutionalizing SIM registration, which may not be compulsory but with corresponding benefits, or encouraging industry collaboration.

Acknowledgements

We thank Natalie Ahn for her comprehensive comments and suggestions on the earlier draft of this paper and guidance throughout the process.

References

Tiago A. Almeida, José María Gómez Hidalgo, and Akebo Yamakami. 2011. Machine learning for email spam filtering: review, approaches and open research problems. *DocEng'11, September, 2011*.

Andreo Calonzo. 2022. Philippines' pldt blocks 23 million texts in sign of scam threat. *Bloomberg*.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2011. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813.

Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, and Opeyemi Emmanuel Ajibuwa. 2019. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Doris Dumlao-Abadilla and Melvin Gascon. 2021. Telcos summoned, asked to do more vs. text scams. *Inquirer*, Nov 2021.

Marketing Interactive. 2016. Filipinos' messaging tool of choice: Facebook messenger. *Marketing Interactive*, Feb 2016.

Gauri Jain, Manisha Sharma, and Basant Agarwal. 2019. Optimizing semantic lstm for spam detection. *International Journal of Information Technology*, 11:239–250.

Tarun Jain, Payal Garg, Namita Chalil, Aditya Sinha, Vivek Kumar Verma, and Rishi Gupta. 2022. Sms spam classification using machine learning techniques. *IEEE*, March.

NPC. 2021. Preventive data privacy practices against smishing. *National Privacy Commission PHE Bulletin*, Oct 2021(21).

Chris Pemberton. 2016. Tap into the marketing power of sms. *Gartner*, Nov 2016.

Word Press. 2022. Email usage statistics 2022: How many people use email? *WP Dev Shed*, Apr.

Sarika Raga and Chaitra B L. 2022. Machine learning and deep learning techniques for sms spam detection, accuracy check and comparative study. *International Journal of Journal homepage: Research Publication and Reviews*, 5(3):2601–2604.

Nafiz Rifat, Mostofa Ahsan, Md. Chowdhury, and Rahul Gomes. 2022. Bert against social engineering attack: Phishing text detection. In *2022 IEEE International Conference on Electro Information Technology (eIT)*, pages 1–6.

Pradeep Kumar Roy, Jyoti Prakash Singh, and Snehasish Banerjee. 2019. Deep learning to filter sms spam. *Future Generation Computer Systems*, 102(Jan):524–533.

Vijay Srinivas Tida and Sonya Hsu. 2022. Universal spam detection using transfer learning of bert model. *arXiv*.

Wavecell. 2019. Customer engagement in the philippines: Texting capital of the world. *Wavecell*.

Cecilia Yap. 2021. Millions of spam messages blocked in philippines as scams surge. *Bloomberg*.

Annex: Results from fine-tuned models

Model	UCI	+PH	+PHTr
BERT + FCN			
3e	0.9803	0.9869	0.937
3e un	0.9919	0.9828	0.9885
5e un	0.9901	0.991	0.9877
BERT + CNN			
3e un	0.9874	0.9861	0.9869
3e un kernel	0.9928	0.9918	0.9845
3e un filters		0.9877	0.9771

Note: e - epoch; un - unfreeze layers.

Table 6: Fine-tuned BERT—Accuracy

Model	UCI	+PH	+PHTr
BERT + FCN			
3e	0.9317	0.9698	0.8684
3e un	0.9712	0.9591	0.9734
5e un	0.9655	0.9791	0.9712
BERT + CNN			
3e un	0.9573	0.9674	0.9689
3e un kernel	0.9747	0.9808	0.963
3e un filters		0.9711	0.9487

Note: e - epoch; un - unfreeze layers.

Table 7: Fine-tuned BERT—f1-score