BERT-Based Models for Phishing Detection

Milita Songailaitė [1,2], Eglė Kankevičiūtė [1,2], Bohdan Zhyhun [1,2] and Justina Mandravickaitė [1,2]

[1] *Vytautas Magnus University, Kaunas, Lithuania*
[2] *Centre for Applied Research and Development (CARD), Kaunas, Lithuania*

Abstract

In this paper we report the application of BERT-based models for phishing detection in emails. We fine-tuned 3 BERT-based models (DistilBERT, TinyBERT and RoBERTa) for the task. All the fine-tuned models attained scores above 0.985 for each metric (accuracy, precision, recall and F1-score). Nevertheless, the RoBERTa model demonstrated the highest classification scores across all metrics, indicating that it can classify the selected phishing data with the utmost accuracy. The models from each BERT architecture have then been assessed more deeply via using them in pseudo-real-life situation. For this purpose, we created an entirely new dataset from the actual phishing emails and used text augmentation techniques to increase their quantity. DistilBERT and RoBERTa models produced very similar outcomes, i.e., most of the emails were classified correctly. However, as DistilBERT uses fewer resources and performs better than the RoBERTa model, it has been regarded as the best model for detecting phishing emails in our case. The TinyBERT variant had the worst results as its size was insufficient for learning to categorize emails and detect phishing.

Keywords

phishing detection, transformers, BERT, DistilBERT, TinyBERT, RoBERTa, cybersecurity, transfer learning

1. Introduction

People are becoming more and more involved in the digital world, which contributes to the pervasive issue of phishing, a sort of cyber-attack [20]. User data is frequently stolen using this method as the attackers' primary strategy is to pose as reliable entities to collect sensitive or private information from their victims [17]. Such an attack might take the form of emails, messages, phony website visits, etc. as the victim is persuaded to open a malicious link, which may install malware, damage the system, or reveal private data. A phishing attack can have severe consequences, such as identity theft, money loss, or other negative outcomes [24].

Phishing attacks are often initiated through emails that appear to be from appropriate sources, such as banks, government authorities, or company management [17]. As these emails contain links that take recipients to fraudulent webpages that imitate legitimate ones, the attacker acquires access to the victim's accounts after the (s)he submits their login credentials or other personal information, which may lead to financial loss or identity theft. Phishing attempts can also lead to the theft of private company information, damage a company's brand, and cause stakeholders and customers to lose faith in it [19]. Moreover, phishing is frequently used to attack governmental systems as a part of significant attacks, such as advanced persistent threat (APT) events [23]. Therefore, the accounts of government employees can be hacked and allow the attackers to get over security barriers, spread malware, or have access to secured data [2].

There are several methods and tools that people have commonly used for phishing detection. Software programs called email filters examine incoming emails and eliminates the ones that may be

28th Conference on Information Society and University Studies (IVUS'2023), May 12, 2023, Kaunas, Lithuania
EMAIL: justina.mandravickaite@vdu.lt (J. Mandravickaitė);

phishing emails [18]. These filters use such techniques as phishing email address blacklist or examination of the email's content. Anti-phishing toolbars is another technique for phishing detection as they provide alerts or prevent access to phishing websites [19]. Another technique for phishing detection is URL analysis to find potentially harmful or questionable information [20]. Such tools may examine the domain name or URL path to match it to a known phishing website. However, user awareness and education are the most effective strategies for phishing detection as well as prevention. Users who are aware of the indications and dangers of phishing attempts, including suspicious email sender addresses and requests for personal data [21], can take the necessary precautions. Another strategy (MFA) is the use of two-factor authentication (2FA) or multifactor authentication [22]. This strengthens the security of the authentication process by a second level of security, such as a code sent to the user's mobile phone in addition to a password [1].

Focusing on phishing email detection, a variety of methods have been used for development of solutions for this task. In recent years deep learning approaches have become popular for phishing detection. Deep learning has such benefits as automated feature extraction, reduced reliance on data pre-processing, extraction of high-dimensional features, and increased accuracy, therefore its application is increasing in various areas, including phishing detection [17]. Such architectures as Convolutional Neural Network (CNN) [25-26], Recurrent Neural Network (RNN) [27], [33], Long Short-Term Memory (LSTM) [28-29], Gated Recurrent Unit (GRU) [30], Multi-Layer Perceptron (MLP) [31], etc. have been used for phishing detection. LSTM and BiLSTM are considered the most widely applied deep learning approaches in phishing detection [17]. Also, transformers architecture for phishing detection was utilised as well, e.g., for developing CatBERT [34], which is a modified BERT [5] model, capable of identifying social engineering emails.

Phishing is a major threat that can seriously hurt both people and businesses. Detecting and preventing phishing attacks is critical to protect sensitive information and prevent a variety of losses. Email filters, anti-phishing toolbars, machine learning, URL analysis tools, and user education are a few techniques and tools that have been utilized for phishing detection. But despite this, all tools and methods need to be improved and supplemented, since an increasing number of new means of influencing systems are being invented [3]. In this paper we report the application of BERT-based models for phishing detection in emails. The rest of paper is structured as follows: ***Data*** briefly introduces data we used for our experiments; ***Methods*** describes methods and base models we used in our experimentation; ***Experimental Setup*** presents the set of parameters we used for fine-tuning the selected BERT-based models; ***Results*** reports results of our experiments and assessment of the fine-tuned models; the final section ends the paper with ***Conclusions***.

2. Data

The starting dataset used in the experiments consisted of 1086 phishing email messages. All messages were anonymized using pseudonymization tools and methods. Each email message was assigned a unique ID during the data preparation stage. Information about links and attachments in the dataset is presented separately. In total, there are 1510 links and 190 attachments in the dataset.

The email messages were divided into the following elements during the data preparation stage:

- Sent at: date and time of sending of the email
- Subject: email subject
- From email: sender's email address
- From name: sender's name
- Reply to: name for reply to the email
- Return path: real email address for reply to the email
- Category: thematic category of the email message
- Risk: degree of risk (evaluated by an expert)
- Risk source: source of risk level evaluation (evaluated by an expert)

- Link: number of links in the email message
- Attachments count: number of attachments in the email message
- Plaintext: email message text.

To understand phishing emails better, we explored their distribution. In the pie charts below, all emails are grouped and analysed according to different classification schemes: a general classification, a technical classification based on the data theft techniques used in the emails, and the target of the attack. We based our general classification of phishing emails on [10] and technical and the target of the attack classifications – on [11]. Therefore, **Figure 1** presents constitution of starting dataset by the general classification. Two-thirds (66.7%) of all the data were attributed to the category of domain or brand impersonation (originally distinguished as separate categories, i.e., domain impersonation and brand impersonation, but merged for simplicity under the label of impersonation). The undefined category (7.73% of emails) consists of emails which could not be classified as a specific type of data theft. A small proportion of the entire dataset was classified as belonging to the categories of extortion, whaling (targeted phishing attack, aimed at senior executives [12]), business email compromise, and ransomware. Since the content of emails belonging to the spear phishing (personalized form of email phishing [10]), lateral phishing (a hijacked corporate account is used to send phishing emails to other users [13]), and account takeover categories is particularly sensitive, and these categories generally encompass a data theft process rather than individual emails, these data theft types were not included in the final experimental dataset.

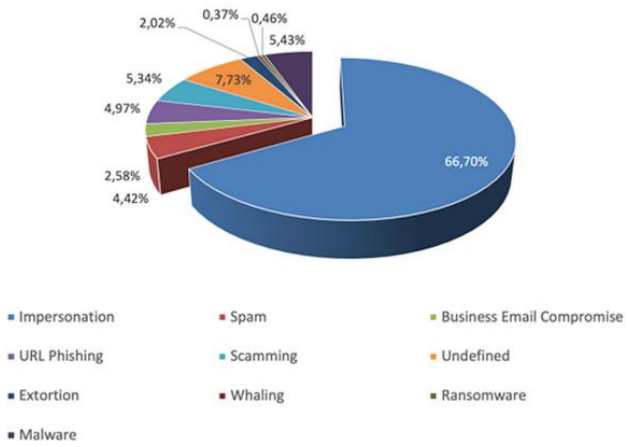

Figure 1: Distribution of thematic emails by the general classification of data theft types.

According to the techniques used in the emails for data theft, all messages were classified into the following categories:

1. Ransomware - malicious software that demands payment in exchange for returning control of a victim's data.
2. Trojan horse and content injection - the use of malware that appears to be legitimate software but is designed to disrupt, damage, or gain unauthorized access to a computer system.
3. Keylogger and screen logger - software designed to track and record the keys struck on a keyboard or the images displayed on a screen.
4. Man-in-the-middle attacks - instances where cybercriminals intercept target email accounts, gain access to them, and monitor or manipulate information exchanged for malicious purposes.
5. Social engineering - a form of manipulation that involves deceiving users into divulging confidential information or downloading malicious software.
6. Scams - fraudulent schemes aimed at tricking individuals into providing personal information or making financial transactions.
7. Undefined category - emails that could not be attributed to a specific data theft technique.

The structure of the dataset according to the data theft techniques used in the emails is presented in **Figure 2**. As can be seen from the results presented in the diagram, most emails (66.15%) were identified as using the social engineering technique. Emails that would have used keylogger and screen

logger could not be obtained. Therefore, emails of these categories were not included in the dataset we used for the experiments.

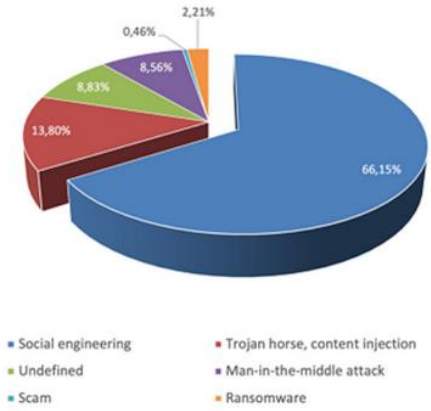

Figure 2: Distribution of thematic emails by the general classification of data theft types.

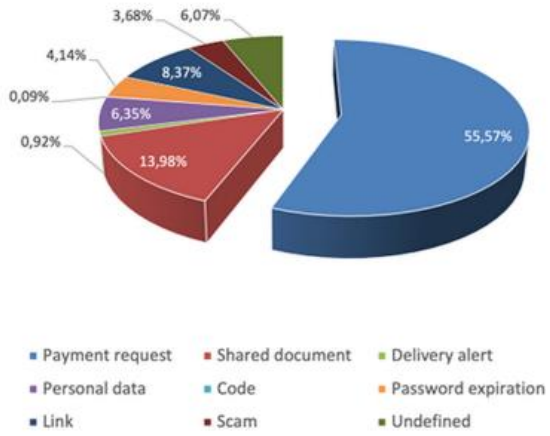

Figure 3: Distribution of thematic emails based on identified targets of data theft in emails.

According to the attack target, all emails were divided into the categories presented in **Figure 3**. As the data presented in the diagram shows, more than half (55.57%) of the analyzed emails were identified as payment requests. Shared documents were identified as a target in 13.98% of the emails, while links were identified as a target in 8.37% of the emails. Personal data was identified as a target in 6.35% of the emails, delivery alerts were identified as a target in 6.07% of the emails, and password expiration was identified as a target in 4.14% of the emails. The targets for the scam, undefined and code categories were only identified in a small fraction of the analyzed emails.

Finally, after analysis and filtering, the starting dataset for fine-tuning pretrained models was complemented with 5323 phishing email messages and 6403 neutral or "ham" email messages from publicly available sources[2].

3. Methods

The phishing detection task was performed using the transfer learning methodology [14]. It involves using a pre-trained language model as a starting point for training a new model on a specific task. This approach is particularly effective when working with small datasets or when training a model for a very specific task [15]. In our case, we chose three popular pretrained deep learning models for the English

[2] We used public datasets available at https://github.com/TanusreeSharma/phishingdata-Analysis and https://github.com/KostasKoutrou/Text_Phishing_Email_ML_Classification

language: DistilBERT, TinyBERT and RoBERTa transformer models to fine-tune them for the task of phishing detection in email messages.

DistilBERT model

Transformer BERT (*Bidirectional Encoder Representations from Transformers*) is a deep learning model based on the attention mechanism [4], which is usually applied to solve various language technology problems [5]. This model works on the principles of transfer learning [6]. A neural network is trained to generate word embeddings, which are then used as input functions for models that solve mainstream language technology tasks. One of the most significant advantages of the BERT architecture models over other neural network models is understanding the context between words in the text. The model learns the context using the attention mechanism characteristic of transformer models, which consists of encoding and decoding mechanisms [4].

DistilBERT is a variant of the BERT model that has been optimized for smaller size and faster performance. It achieves this by employing a process called knowledge distillation, where a smaller model learns from the predictions and representations of a larger pre-trained model [7]. This is a common method for developing low resource Large Language Models. The process involves pre-training a large BERT model, fine-tuning it on a task, selecting a sub-network, training a small model, and applying knowledge distillation to allow the small model to learn from the large model's predictions and representations. This produces a smaller model that performs similarly to the larger one, making it suitable for resource-constrained applications or those requiring faster inference times.

TinyBERT model

Similarly to DistilBERT, TinyBERT also uses knowledge compression methodology to achieve faster model performance [8]. However, there are several key differences between two models:

1. DistilBERT is already a smaller version of BERT, but TinyBERT is even smaller, with a size of only a few hundred megabytes, making it ideal for low resource development.
2. DistilBERT uses a technique called knowledge distillation to transfer the knowledge learned from a larger pre-trained model like BERT to a smaller model. TinyBERT, on the other hand, uses a similar approach called "teacher-student" learning, where the smaller model is trained to mimic the behavior of a larger model by matching the outputs of the two models on the same inputs.
3. DistilBERT is trained on a combination of unlabeled and labeled data, while TinyBERT is trained only on labeled data, making it more efficient for specific tasks.

Overall, both models require less resources and produce comparable model predictions compared to the BERT model. As a result, it was decided to test both for the phishing detection task.

RoBERTa model

RoBERTa (*Robustly Optimized BERT approach*) extends the BERT language masking approach, in which the system learns to predict the masked text portions within unlabeled language samples. RoBERTa modifies critical hyperparameters in the BERT, such as removing BERT's next-sentence prediction objective, and it was trained with much bigger mini-batches and learning rates [9]. This enables RoBERTa to outperform BERT on the masked learning goal, resulting in superior downstream task performance. Furthermore, RoBERTa was trained on a larger and more diverse corpus of data, enabling the model to comprehend complex information that may span a longer time period. This is particularly significant in the context of phishing detection, where the content of messages may change over time. Finally, unlike BERT, which always masks out the same tokens during pre-training, RoBERTa uses dynamic masking. This means that the model is trained to predict masked tokens based on the surrounding context, making it better at handling out-of-vocabulary words.

Fine-tuning the models

The second step of a transfer learning methodology is fine-tuning the pretrained BERT language models. At this stage, the already pre-trained model is learning how to classify the given data based on the training data [16]. The process begins by initializing the BERT model with pre-trained weights on a large corpus of text data. Then, a new classification layer is added on top of the pre-trained model, which is trained on the specific task using labeled data. During training, the weights of the pre-trained model are updated along with the weights of the classification layer. Once training is complete, the fine-tuned model can be used to predict the classification of new text inputs. Fine-tuning with BERT models has demonstrated to be extremely successful in achieving exceptional results on a wide range of natural language processing tasks.

4. Experimental Setup

The phishing detection model was built with transfer learning approach outlined in the ***Methods*** section and trained on data explained in the ***Data*** section. The three base models (DistilBERT[3], TinyBERT[4] and RoBERTa[5]) were already pre-trained on a large English language corpus by the creators of these models. Then we fine-tuned these models to classify phishing email data.

The fine-tuning was done three times for each of the base large language models using different random weights initiation seed. Overall, nine models were fine-tuned to classify phishing emails into two classes – phishing or not phishing. The models were fine-tuned for 30 epochs using a variable learning rate that began at 0.001. Each architecture had a distinct training batch size: TinyBERT had 64, DistilBERT had 36, and RoBERTa had 24. That is, the smaller the model, the bigger the batch size we could choose. While training, each of the models were evaluated by looking at the loss function scores. In addition, after each epoch the evaluation step was done, where the model's ability to classify phishing emails was evaluated by four selected classification metrics.

[3] https://huggingface.co/distilbert-base-uncased
[4] https://huggingface.co/prajjwal1/bert-tiny
[5] https://huggingface.co/roberta-base

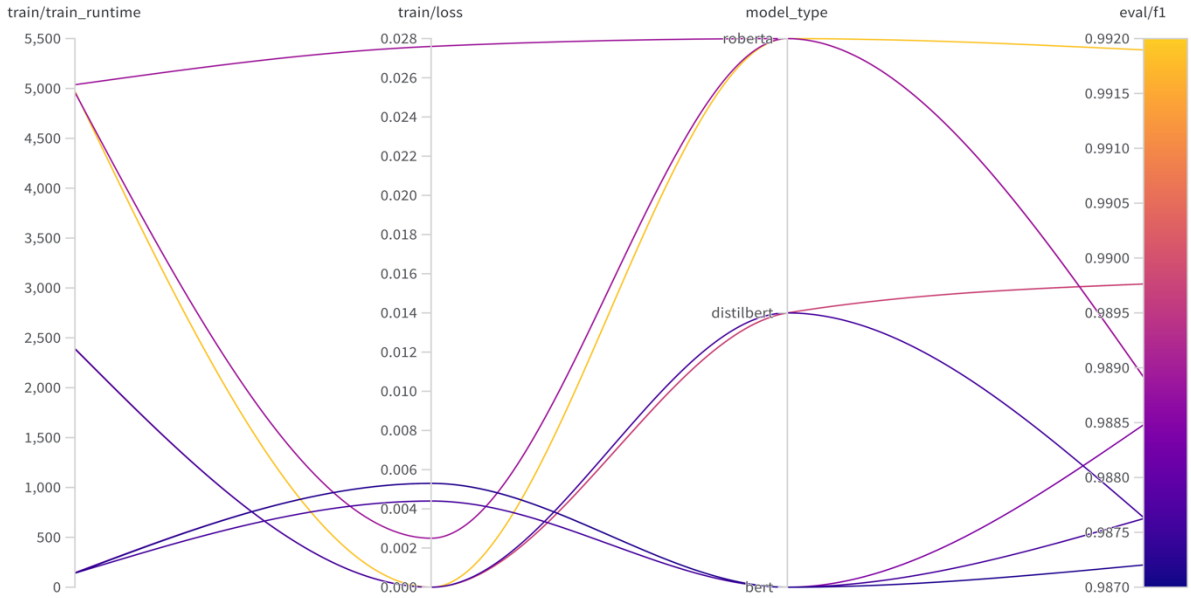

Figure 4: The training parameters comparison between the fine-tuning of all nine classification models. The "roberta" represents RoBERTa models, "distilbert" represents DistilBERT models and "bert" represents "TinyBERT" models.

The comparison of the fine-tuned models' training parameters is given in Figure 4. RoBERTa type models were the longest to train since this model had the most complex architecture and was trained on more data than the two distilled learning type models (DistilBERT and TinyBERT). However, most of the RoBERTa models also had the lowest training loss, which later resulted in higher classification performance. The two distilled learning models both had a lower train runtime. Still, the fastest to fine-tune was the TinyBERT model. However, both models had higher training loss than the larger RoBERTa model.

5. Results

The experiments on the classification of phishing emails were performed using three different BERT model architectures: DistilBERT, TinyBERT and RoBERTa. The models were trained with parameters described in the section ***Experimental Setup***. The initial model results are shown in Figure 5. The models' abilities to classify the emails into two classes (phishing or not phishing) were evaluated with four classification metrics: accuracy, precision, recall and F1-score.

Overall, it can be observed that all of the models exhibit strong performance in classifying phishing emails. All of the fine-tuned models attained scores above 0.985 for each metric. Nevertheless, the RoBERTa model demonstrated the highest classification scores across all metrics, indicating that it can classify the selected phishing data with the utmost accuracy. While DistilBERT and TinyBERT models may not have performed as well as RoBERTa, they do offer the advantage of requiring significantly less computing resources and time to train. This makes them ideal for low resource applications.

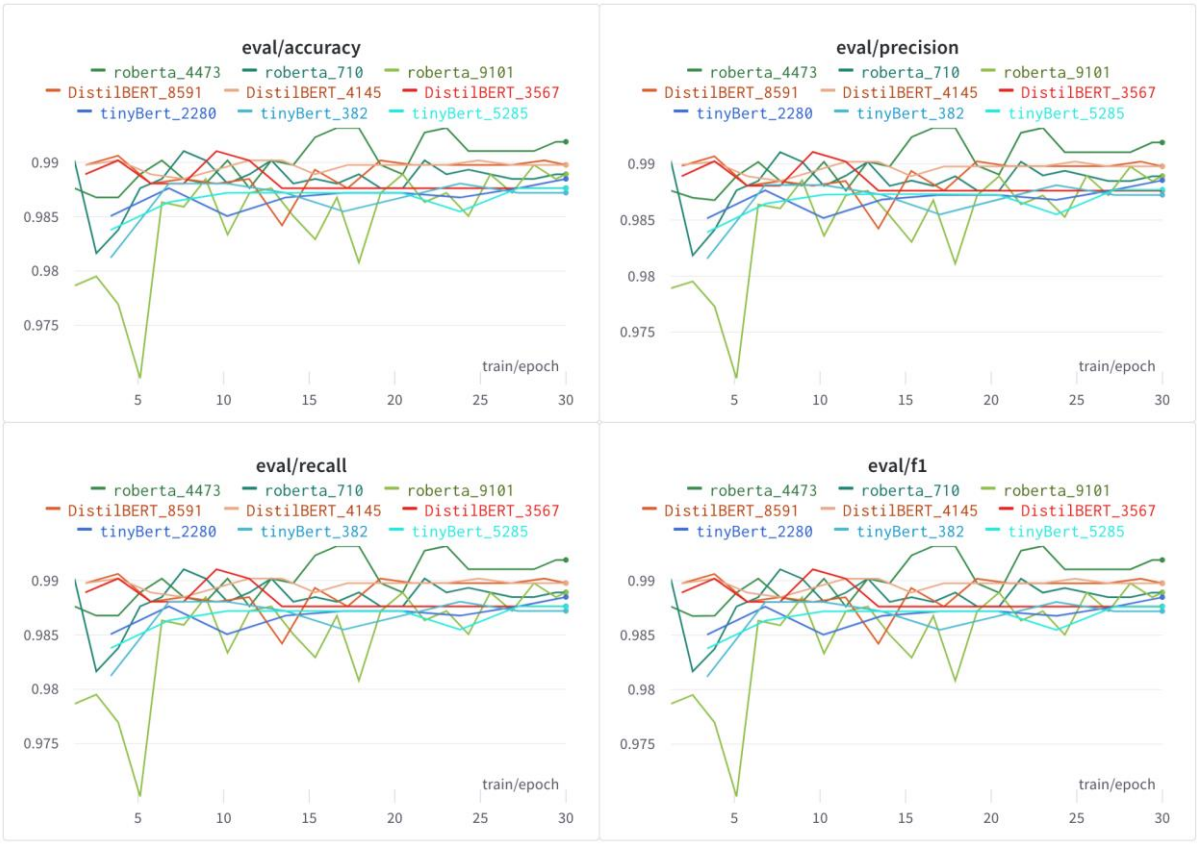

Figure 5: The classification models results according to the four selected classification metrics: accuracy, precision, recall and F1 score.

The next stage in model assessment was to see how well the models classified real-world phishing email data. We created an entirely new dataset from the actual phishing emails we gathered for this purpose. Several text augmentation techniques were used to increase the quantity of gathered emails:

1. The introductions and endings of the letters were rewritten in several ways so that the idea would stay the same. These parts of emails were exchanged thus generating more variations of the same email.
2. A database of fictitious personal information (email addresses, phone numbers, personal identity numbers, and so on) was developed. The same variables were detected in each of the real emails. The emails were augmented with variables from the personal information database, resulting in more phishing emails of the same type.

After the augmentation, there were 8994 phishing emails in the augmented testing database. These emails were then used to test the best models from each BERT architecture. The results are presented in Table 1.

Table 1

The results of best models from each BERT architecture experiment testing with the phishing email database. *Classified correctly* column indicates that the model classified the emails as phishing and the *Classified incorrectly* column indicated that the model classified the letters as not phishing.

Model name	Classified correctly	Classified incorrectly
DistilBERT	8590	404
TinyBERT	5488	3506
RoBERTa	8552	442

DistilBERT and RoBERTa models produced very similar outcomes. Almost all the emails were accurately classified by these models. However, because DistilBERT uses fewer resources and performs better than the RoBERTa model, it is regarded as the best model for detecting phishing emails in our case. The TinyBERT variant had the worst results. Although this BERT design is an improvement over the DistilBERT, the model is also significantly smaller. As a result, the TinyBERT size was insufficient to learn how to categorize different emails and detect phishing.

6. Conclusions

In this paper we reported the application of BERT-based models for phishing detection in emails. We fine-tuned 3 BERT-based models (DistilBERT, TinyBERT and RoBERTa) for the task. The fine-tuning was done three times for each of the base large language models using different random weights initiation seed. Overall, nine models were fine-tuned to classify phishing emails into two classes – phishing or not phishing. The models were fine-tuned for 30 epochs using a variable learning rate that began at 0.001. Each architecture had a distinct training batch size: TinyBERT had 64, DistilBERT had 36, and RoBERTa had 24. All the fine-tuned models attained scores above 0.985 for each metric (accuracy, precision, recall and F1-score). Nevertheless, the RoBERTa model demonstrated the highest classification scores across all metrics, indicating that it can classify the selected phishing data with the utmost accuracy. The models from each BERT architecture have then been assessed more deeply via using them in pseudo-real-life situation. For this purpose, we created an entirely new dataset from the actual phishing emails and used text augmentation techniques (the introductions and endings of the letters were rewritten in several ways; a database of fictitious personal information (email addresses, phone numbers, personal identity numbers, and so on) was developed) to increase their quantity. After the augmentation, there were 8994 phishing emails in the augmented testing database. DistilBERT and RoBERTa models produced very similar outcomes, i.e., most of the emails were classified correctly (8590/8994 by DistilBERT and 8552/8994 by RoBERTa). However, as DistilBERT uses fewer resources and performs better than the RoBERTa model, it has been regarded as the best model for detecting phishing emails in our case. The TinyBERT variant had the worst results as its size was insufficient for learning to categorize emails and detect phishing.

Our future plans include experimentation with a more diverse variety of models and datasets. We also plan to explore the application of BERT-based models for the detection of phishing emails, written in non-English languages.

References

[1] A. Pagán, and K. Elleithy, A Multi-Layered Defense Approach to Safeguard Against Ransomware, In 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0942-0947. IEEE, 2021.

[2] Z. A. Wen, Z. Lin, R. Chen, and E. Andersen, What. hack: engaging anti-phishing training through a role-playing phishing simulation game, In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1-12. 2019.

[3] A. Binks, The art of phishing: past, present and future, Computer Fraud & Security 2019, no. 4 (2019) 9-11.

[4] S. Lei, W. Yi, C. Ying, and W. Ruibin, Review of attention mechanism in natural language processing, Data Analysis and Knowledge Discovery 4, no. 5 (2020) 1-14.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, (2018). URL: https://arxiv.org/abs/1810.04805

[6] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf, Transfer learning in natural language processing" in Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials, 2019, pp. 15–18.

[7] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter, 2020. https://doi.org/10.48550/arXiv.1910.01108.

[8] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, TinyBERT: Distilling BERT for Natural Language Understanding, arXiv, October 15, 2020. URL: http://arxiv.org/abs/1909.10351.
[9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019. URL: http://arxiv.org/abs/1907.11692
[10] Barracuda, 13 email threat types to know about right now, 2020. URL: https://assets.barracuda.com/assets/docs/dms/Barracuda-eBook_13-email-threats_may2020.pdf
[11] A. Aleroud, and L. Zhou, Phishing environments, techniques, and countermeasures: A survey, Computers & Security 68 (2017): 160-196.
[12] A. Shankar, R. Shetty, and B. Nath, A review on phishing attacks, International Journal of Applied Engineering Research 14, no. 9 (2019): 2171-2175.
[13] G. Ho, A. Cidon, L. Gavish, M. Schweighauser, V. Paxson, S. Savage, G. M. Voelker, and D. Wagner, Detecting and characterizing lateral phishing at scale, In 28th USENIX Security Symposium (USENIX Security 19) (2019): 1273-1290.
[14] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, A comprehensive survey on transfer learning, Proceedings of the IEEE 109, no. 1 (2020): 43-76.
[15] M. A. Bashar, and R. Nayak, Active learning for effectively fine-tuning transfer learning to downstream task, ACM Transactions on Intelligent Systems and Technology (TIST) 12, no. 2 (2021): 1-24.
[16] C. Sun, X. Qiu, Y. Xu, and X. Huang, How to fine-tune bert for text classification?, In Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18, pp. 194-206. Springer International Publishing, 2019.
[17] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, Deep learning for phishing detection: Taxonomy, current challenges and future directions, IEEE Access (2022).
[18] J. Rastenis, S. Ramanauskaitė, I. Suzdalev, K. Tunaitytė, J. Janulevičius, and A. Čenys, Multi-Language spam/Phishing classification by Email Body text: toward automated security Incident investigation, Electronics 10, no. 6 (2021): 668.
[19] S. Chanti, and T. Chithralekha, Classification of anti-phishing solutions, SN Computer Science 1, no. 1 (2020): 11.
[20] S. Salloum, T. Gaber, S. Vadera, and K. Sharan, A systematic literature review on phishing email detection using natural language processing techniques, IEEE Access (2022).
[21] B. Reinheimer, L. Aldag, P. Mayer, M. Mossano, R. Duezguen, B. Lofthouse, T. Von Landesberger, and M. Volkamer, An investigation of phishing awareness and education over time: When and how to best remind users, In Proceedings of the Sixteenth USENIX Conference on Usable Privacy and Security, pp. 259-284. 2020.
[22] N. Sarginson, Securing your remote workforce against new phishing attacks, Computer Fraud & Security 2020, no. 9 (2020) 9-12.
[23] A. Khalid, A. Zainal, M. A. Maarof, and F. A. Ghaleb, Advanced persistent threat detection: A survey, In 2021 3rd International Cyber Resilience Conference (CRC), pp. 1-6. IEEE, 2021.
[24] S. Magdy, Y. Abouelseoud, and M. Mikhail, Efficient spam and phishing emails filtering based on deep learning, Computer Networks 206 (2022): 108826.
[25] S. Y. Yerima, and M. K. Alzaylaee, High accuracy phishing detection based on convolutional neural networks, In 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), pp. 1-6. IEEE, 2020.
[26] R. Vinayakumar, K. P. Soman, P. Poornachandran, V. S. Mohan, and A. D. Kumar, ScaleNet: scalable and hybrid framework for cyber threat situational awareness based on DNS, URL, and email data analysis, Journal of Cyber Security and Mobility 8, no. 2 (2019) 189-240.
[27] E. Castillo, S. Dhaduvai, P. Liu, K.-S. Thakur, A. Dalton, and T. Strzalkowski, Email threat detection using distinct neural network approaches, In Proceedings for the First International Workshop on Social Threats in Online Conversations: Understanding and Management, pp. 48-55. 2020.

[28] P. Verma, A. Goyal, and Y. Gigras, Email phishing: Text classification using natural language processing, Computer Science and Information Technologies 1, no. 1 (2020) 1-12.
[29] Q. Li, M. Cheng, J. Wang, and B. Sun, LSTM based phishing detection for big email data." IEEE transactions on big data 8, no. 1 (2020) 278-288.
[30] M. A. Remmide, F. Boumahdi, and N. Boustia, Phishing Email Detection Using Bi-GRU-CNN Model, In Proceedings of the International Conference on Applied CyberSecurity (ACS) 2021, pp. 71-77. Cham: Springer International Publishing, 2022.
[31] T. O. Oyegoke, K. K. Akomolede, A. G. Aderounmu, and E. R. Adagunodo, A Multi-Layer Perceptron Model for Classification of E-mail Fraud, European Journal of Information Technologies and Computer Science 1, no. 5 (2021) 16-22.
[32] M. Regina, M. Meyer, and S. Goutal, Text Data Augmentation: Towards better detection of spear-phishing emails, arXiv preprint arXiv:2007.02033 (2020).
[33] L. Halgaš, L. I. Agrafiotis, and J. R.C. Nurse, Catching the Phish: Detecting phishing attacks using recurrent neural networks (RNNs), In Information Security Applications: 20th International Conference, WISA 2019, Jeju Island, South Korea, August 21–24, 2019, Revised Selected Papers 20, pp. 219-233. Springer International Publishing, 2020.
[34] Y. Lee, J. Saxe, and R. Harang, CATBERT: Context-aware tiny BERT for detecting social engineering emails, arXiv preprint arXiv:2010.03484 (2020).