To what extent can bi-directional transformer-based neural networks be applied to correctly classify malicious phishing URLs?

April 30, 2024

Word Count: 4073

1. Introduction

Phishing attacks are a technique to acquire sensitive data, such as passwords and bank account numbers, through fraudulent solicitation in email or on a website, in which the attacker impersonates a legitimate business or person (National Institute of Standards and Technology). They exploit system weaknesses through users, making them especially vulnerable to attackers. For example, even if a system is secure enough to protect against password theft, an attacker may provide a user with a Hypertext Transfer Protocol (HTTP) link that could make them leak their password under the assumption they are updating it. In addition, the domain name of the uniform resource locator (URL) could be manipulated through technical vulnerabilities (e.g. Domain Name System cache poisoning) which can create very convincing phishing links (Khonji et al., 2013). With the average cost of a data breach reaching $4.35 million and increasing by nearly 13%, finding solutions is more important than ever (IBM, 2022).

Efforts to combat these phishing attacks have used a range of techniques, with one of the earliest and easiest being blacklists. Blacklists are databases that record known malicious phishing URLs and domain names that are continuously updated as more are reported through open-source communities such as OpenPhish. However, the changing nature of phishing attacks makes this method quickly outdated and ineffective. As a result, more recent approaches to phishing prevention have focused on artificial intelligence. Specifically, machine learning–a subset of artificial intelligence–has emerged as a leading technique due to its ability to detect phishing attacks that have never been seen before due to its ability to analyze and detect patterns in past attacks that it has been trained on.

This paper focuses on deep learning, a field within machine learning, that focuses on more complex architectures based on artificial neural networks. Neural networks are advanced

machine learning algorithms that can detect complex patterns in large amounts of data and make predictions based on the information they have been trained on. These systems are designed to mimic the structure of the human brain, with interconnected layers of "neurons" that transmit information to each other. An artificial neural network typically consists of an input layer, one or more hidden layers, and an output layer, each with a specific number of neurons connected to preceding and following layers. There are types based on artificial neural networks such as convolutional neural networks and recurrent neural networks as they can extract and learn even deeper features in the data such as the order in which characters are put in an HTTP link. Specifically, this paper explores the use of applying transformer-based neural networks on the task of classifying malicious phishing URLs.

2. Literature Review

2.1 Traditional Phishing Prevention Approaches

The two main traditional phishing prevention approaches are user training (users can be better educated to understand phishing attacks and identify them) and software classification (programs classify phishing and legitimate messages on behalf of the user). However, with educational training, users failed to detect nearly thirty percent of phishing attacks even when trained with the best-performing user awareness program (Sheng et al., 2010).

Within software classification, there are blacklists, heuristic, and machine learning approaches. Blacklists as previously mentioned are updated lists of previously detected phishing URLs, which are ineffective against zero-hour phishing attacks (phishing attacks that exploit a vulnerability that has zero days to be fixed) as they are not on these blacklists until approximately twelve hours after the fact.

Heuristics are rules that are used to determine whether a form of media such as an email, website, or URL is a part of a phishing attempt. They identify common characteristics of phishing attempts like the structure of URLs, and any patterns within domains such as how long it has been since it was registered or registration details that have been linked to fraudulent activities. Heuristics can also utilize algorithms to inspect payloads of protocols like HTTP or SMTP and prevent users from visiting malicious sites. However, these algorithms can only be developed for a certain amount of attacks, and still suffer the same problem of not being able to generalize to new threats (Sahoo et al., 2019).

2.2 Machine Learning

Machine learning is a subfield of artificial intelligence that focuses on building algorithms that learn based on data that is provided (Oracle Cloud). In an attempt to prevent phishing, this approach uses features like the URL, internet protocol address, text content, and source code of the website to classify malicious phishing attempts. Algorithms have primarily focused on classic machine-learning techniques. For instance, one of the most effective is support vector machines which find an optimal plane in a multi-dimensional space (each dimension being a characteristic used for training). The goal of support vector machines is to have one “side” of the hyperplane hold malicious phishing attacks, and the other side hold benign phishing attacks. Additionally, decision trees, k-means, and k-nearest neighbors have been utilized in the task of classifying phishing attacks, but deep learning remains relatively unexplored, leaving a gap that needs more research (Khonji et al., 2013).

2.3 Deep Lexarning

Deep learning is a subfield of machine learning based on artificial neural networks. These systems are designed to mimic the structure of the human brain, with interconnected layers of "neurons" that transmit information to each other. A singular neuron takes in inputs, performs various mathematical operations, and produces an output. Each input is multiplied by a certain weight (randomly initialized) from the input layer and is all added together in a weighted sum along with a bias term (typically initialized at 0). This sum is then passed through an activation function, which plays an integral role in neural networks by introducing nonlinear transformations that add necessary complexity. This process for each neuron is repeated for each layer in the network until the output layer in which another activation function is applied depending on the number of output neurons.

In 2018, a team of researchers from the Singapore Management University proposed an end-to-end deep learning framework for classifying malicious phishing URLs–URLNet: a deep learning framework to learn URL embeddings for the detection of malicious URLs directly from the sequence. In order to use URLs as training data, they convert them to a list of numbers called a vector which are called "embeddings". The closer two of these vectors are to each other when projected onto a multi-dimensional plane, the closer their URLs are in terms of structure. After converting the URLs to embeddings, a convolutional neural network–a type of neural network that learns features through filters that convolve around the input data–is applied to both the characters and words of the URL string to learn in "a jointly optimized framework". Through this combined approach, the model captures semantic information which is not possible through traditional machine-learning methods. For their data, they scraped VirusTotal to collect a total of 15,000,000 malicious and benign URLs. The researchers evaluate the performance of their

method by finding the area under the receiver operating characteristic curve, which quantifies how well their model is performing while taking an aggregate measure across all possible classification thresholds. They found that their combined word and character-level convolution approach outperformed all baselines with an area under receiver-operator characteristic of 0.9929, which is a very high score for this type of metric (Le, Hung et al. 2018).

However, the deep learning landscape has shifted since then towards another type of neural network: transformer neural networks. In 2017, the Google Brain team proposed a novel network architecture for deep learning on sequence data. The transformer, based solely on "attention" mechanisms, gets rid of the recurrence and convolution that has dominated the machine learning space. Attention is a set of values that shows how compatible each part of the sequence is with one another. Transformers rely solely on these attention mechanisms. Advantages of this approach include having a theoretically infinite attention window, allowing for better modeling in longer texts, and faster computation, as using solely attention lends itself to parallel computing. Although this research was originally applied to just language translation, researchers have applied transformers to everything related to natural language processing, and have achieved state-of-the-art performance on various tasks from sentiment analysis to text classification (Vaswani et al., 2017).

There have been a few implementations of transformers in phishing classification since then. In 2021, Xu introduced using transformers for malicious URL detection. They then describe their proposed solution in detail, and explain how they train their transformer model. The performance of the transformer model is compared to six machine learning models: Decision Tree, Random Forest, Multi-layer Perceptrons, XGBoost, Support Vector Machines, and Auto Encoders. The results show that they outperformed these traditional methods with

transformer models, with a variety of performance metrics including a testing accuracy of 0.973, precision of 0.984, recall of 0.962, and an F1-score of 0.973 (Xu, 2021).

Additionally, there has been some work done in bi-directional transformer neural networks, which consist of two separate transformer models. One of these transformer models processes the input sequence from the left, while the other processes it from the right. A team from Microsoft Research finetuned bi-directional transformer models along with creating a custom "URLTran" transformer and training it on ~3,000,000 phishing URLs scraped from the internet. Their system uses tokenizers instead of extracting lexical features from the URLs, which transforms a URL into a list of characters. The tokenizer converts the URLs to a token sequence, which is then mapped to an embedding vector to act as the input sequence for the transformer. For the evaluation of these transformer models, they use the receiver-operator characteristic curve, accuracy, precision, recall, and F1 scores. The results show that URLTran significantly outperforms recent baselines, with a wide range of very low false positive rates (Maneriker, 2021). However, there are many types of bi-directional transformer neural networks that were not implemented in this study.

2.4 Conclusion

Currently, much of the literature surrounding phishing detection using machine learning focuses on rule-based detection like the IP address and actual HTML content, which is theoretically more accurate but not as feasible due to the number of features required to classify a message as phishing or not. This led to the idea of URL evaluation, which only requires the link to the website and less dependence on visiting the website. Algorithms have primarily focused on non-deep learning techniques like support vector machines and decision trees which leaves a

gap that has yet to be explored. Recently, some work has been done using primitive types of artificial neural networks such as perceptrons, convolutional neural networks, and recurrent neural networks. The few papers that do implement transformer neural networks for phishing classification call for more research, especially in bi-directional transformer neural networks due to the recency of this discovery. In my research, I will explore this gap by conducting experimental research comparing the performance of a variety of bi-directional transformer neural networks in classifying malicious phishing URLs.

3. Methodology

This section outlines how the performance of different bi-directional transformer neural networks will be evaluated in detecting phishing URLs. The research method used is experimental research with the variable that is manipulated being the type of bi-directional transformer neural network used. The transformer models used in this experiment are BERT, ALBERT, RoBERTa, and DistilBERT, which will be tested on sddataset of malicious phishing URLs. The performance of these transformer neural network models will be assessed on a variety of metrics including F1-Score, Accuracy, Precision, and Recall.

3.1 Dataset

The dataset used in this experiment was PhishStorm, which was built for the paper "PhishStorm: Detecting Phishing with Streaming Analytics". The dataset includes a total of 96, 018 URLs, with 48,009 benign URLs and 48,009 malicious phishing URLs. It is in the format of a comma-separated values file with a domain column (records the URL) and a label column (0 for benign, 1 for phishing) (Marchal, 2014). Eighty percent of this dataset is used for training the transformer neural networks, and the other twenty percent is used solely for evaluation as it is

data the neural network models have not seen before. This dataset was selected due to the technique which was used to collect URLs. Malicious phishing URLs were collected via open-source phishing collection websites such as PhishTank and VirusTotal, while benign URLs were collected from the Open Directory Project which contains previously legitimate URLs that were discarded. Additionally, this dataset is balanced (equal number of benign and malignant URLs) which allows certain performance metrics such as accuracy to effectively portray the effectiveness of the transformer models.

3.2 Model Creation

```
def create_encodings(row, tokenizer):
   url = row['domain']
   url = str(url)
   url = ' '.join(url.split())

   encodings = tokenizer(url, padding="max_length",
                         truncation=True, max_length=256)
   label = int(row['label'])

   encodings['label'] = label
   encodings['text'] = url

   return encodings
models = ['bert-base-uncased', 'albert-base-v2', 'roberta-base',
'distilbert-base-uncased']

CURRENT_MODEL = models[0]
model = AutoModelForSequenceClassification.from_pretrained(CURRENT_MODEL,
num_labels=2)
```

Figure 1: Primary Python Code for Transformer Model Instantiation

A variety of transformer neural network models are chosen, BERT, ALBERT, RoBERTa, DistilBERT, XLM, and FlauBERT. These neural networks are chosen for their very diverse

architectures and training mechanisms, which provide a broad spectrum when compared. Each model has been shown to have unique characteristics and capabilities that could influence their performance in classifying malicious phishing URLs.

For each of the selected models, the matching pre-trained model and tokenizer are imported using the Hugging Face Transformer library. The tokenizer converts the URLs into a suitable format for input to the model, which includes encoding the URLs into IDs and attention masks. This ensures that the URLs that are fed into the transformers align with what each model expects. Each of the models is then instantiated with pre-trained parameters and two output labels.

3.3 Model Training and Evaluation

```
from sklearn.metrics import accuracy_score,
precision_recall_fscore_support

def compute_metrics(pred):
   labels = pred.label_ids
   preds = pred.predictions.argmax(-1)
   precision, recall, f1, _ = precision_recall_fscore_support(labels,
preds, average='binary')
   acc = accuracy_score(labels, preds)

   return {
       'accuracy': acc,
       'f1': f1,
       'precision': precision,
       'recall': recall
   }
```

Figure 2: Evaluation Metric Function

To train each of the transformer neural networks, the CSV dataset must be loaded into a data frame using the Pandas module, and then converted into a Hugging Face dataset. The URLs

are then tokenized using the functions defined earlier to ensure proper training of the networks. The dataset is then split into training and testing with an 80-20 split. Before training, the performance metrics used in the evaluation are implemented into a function, which includes precision, recall, f1, and accuracy using the sklearn module.

```
from transformers import TrainingArguments, Trainer

training_args = TrainingArguments(
   output_dir='./models',
   num_train_epochs=1,
   evaluation_strategy="epoch",
   save_strategy="epoch",
   load_best_model_at_end=True)

trainer = Trainer(
   model=model,
   args=training_args,
   train_dataset=train_hg,
   eval_dataset=valid_hg,
   compute_metrics=compute_metrics)

trainer.train()
trainer.evaluate()
```

Figure 3: Transformer Training and Evaluation Code

Next, define the hyperparameters used when training (epochs, batch size, etc.) and use the Trainer function from the transformers module to train and evaluate each of the transformer neural networks instantiated earlier. Record all of the performance metrics returned by each of the transformer neural networks for statistical analysis.

4. Results

This research aims to evaluate the performance of various bi-directional transformer neural networks in their ability to classify malicious URLs. Specifically, four models were

compared: BERT, AlBERT, RoBERTa, and Distilbert, across performance metrics including accuracy, F1 score, precision, and recall. Additionally, each model's training and validation losses were also recorded. This section will analyze the results of these comparisons and focus on what each of the performance metrics reveals about each model in classifying malicious URLs.

	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
BERT	0.054	0.058049	0.985925	0.985926	0.992965	0.978986
ALBERT	0.1273	0.123741	0.972893	0.972746	0.985138	0.960663
RoBERTa	0.6697	0.758862	0.502997	0	0	0
Distilbert	0.0552	0.06561	0.985091	0.985107	0.991094	0.979193

Table 1: Performance Metrics for Transformers

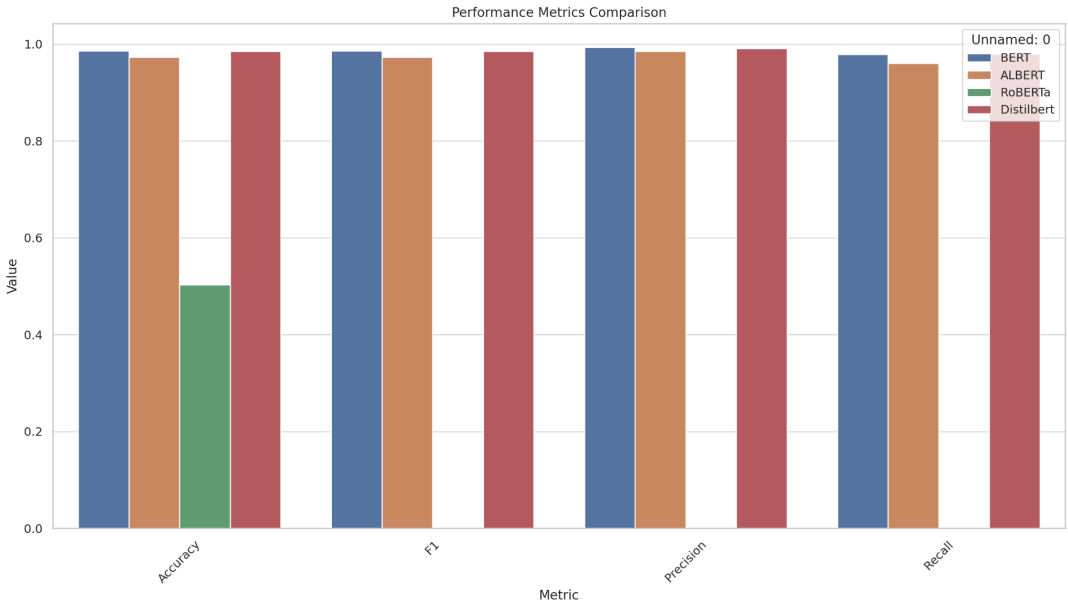

Figure 1: Comparison of Performance between Models Bar Graph

4.1 Accuracy

Accuracy is the most basic in evaluating the performance of machine learning models, as it is easy to understand and interpret. In the context of this project, accuracy measures the proportion of URLs that are identified correctly as either malicious or benign, which includes both true positives (malicious URLs correctly identified as malicious) and true negatives (benign URLs correctly identified as benign) out of all evaluated URLs.

In the comparison of bi-directional transformer neural networks for malicious URL classification, as shown in Figure 1, the BERT model achieved the highest accuracy with 98.59%, and Distilbert had 98.51%. ALBERT had a slightly lower accuracy of 97.29%, which could offer a balance between computational efficiency and performance, where computing power is a concern (like smartphones). However, RoBERTa underperformed with an accuracy of 50.30%, indicating that it is around as good as a random guess rather than a machine learning model.

However, since our dataset is not perfectly balanced (the number of malicious URLs does not equal the number of benign URLs) the performance comparison should not stop at accuracy. For example, if 90% of the dataset comprised malicious URLs, a model that "learned" to always predict a URL to be malicious would achieve an accuracy of at least 90%.

4.2 Precision

Precision is defined as the ratio of true positives (malicious URLs correctly identified as malicious) to the total predicted positives (malicious URLs correctly identified + benign URLs incorrectly identified as malicious). This is an important metric to avoid misidentifying benign URLs as phishing, which could lead to blocking a legitimate or important website.

The BERT model achieved the highest precision with 99.30% and was again closely followed by Distilbert and ALBERT with precisions of 99.11% and 98.51% respectively. These precision scores show the models' ability to accurately predict malicious URLs with a very low false positive rate. Conversely, RoBERTa's had a very small precision metric, which kept in line with the accuracy mentioned earlier. This further highlights the challenges that it faces in this classification task.

4.3 Recall

In contrast to precision, recall is the ratio of true positives (malicious URLs correctly identified as malicious) to all positives in the dataset (malicious URLs correctly identified as malicious + malicious URLs incorrectly identified as benign). Recall in this scenario can record the effectiveness of each model in detecting true phishing links without missing any potential risks.

Similar to precision and accuracy, BERT showed the highest recall with 97.90% and was again closely followed by Distilbert and ALBERT with recalls of 97.92% and 96.07% respectively. These high recall scores show the ability to correctly identify almost all malicious URLs, and minimize the threat of a phishing site going undetected. RoBERTa's recall was still much lower compared to the others, which further emphasizes the limitations of the RoBERTa model for the malicious URL classification task.

4.4 F1-Score

In machine learning, F1-Score is the primary metric used when evaluating models. It takes the harmonic mean of precision and recall, which therefore takes into account both false

positives and false negatives. In this case, it balances accurately identifying phishing URLs and minimizing the amount of phishing URLs that are missed.

The BERT model achieved the highest F1-score with 98.59% and was followed by Distilbert and ALBERT with F1-scores of 98.51% and 97.27% respectively. However, RoBERTa still performed terribly in comparison to the other models and shows that significant adjustments are needed.

4.5 Training Loss and Validation Loss

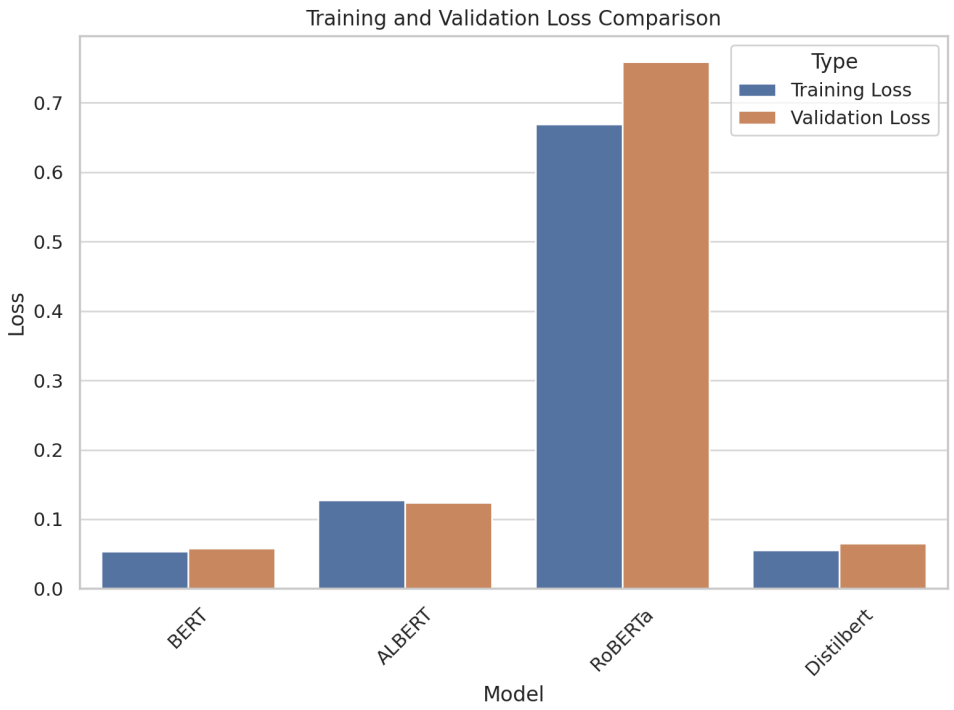

Figure 2: Comparison of Loss Values between Models

Loss is a concept in machine learning that quantifies how well a model is performing when it is training/learning. It calculates the differences between the model's predictions and actual targets from the training data to see how far off it is. The ultimate goal of training is to minimize this loss value, which will improve the model's accuracy. Training loss is determined

by applying a loss function to the training data, and validation loss is the loss function applied on data not used when training, or testing dataset. In both cases, generally, a lower loss indicates better performance.

As seen in Figure 2, BERT and DistilBERT were able to maintain a low loss value during both the training and validation phases, which indicates their ability to learn new data and generalize. ALBERT achieved slightly higher levels of validation loss in comparison to its training loss, but still maintained relatively high performance. However, RoBERTa's high loss values further highlight its underperformance and the need for further tuning to perform better on this task.

4.6 Summary

This section reveals the BERT and Distilbert as the most effective transformer neural networks for classifying malicious URLs, as seen in their consistently high performance throughout accuracy, precision, recall, f1-score, and loss values. ALBERT can be seen as a very viable alternative in scenarios with the need to optimize performance due to computation demands in low-performance devices due to its slightly lower performance across the board. RoBERTa's consistently poor performance indicates the need for adjustments for it be effectively applied.

5. Discussion

The results of this research have shown that bi-directional transformers do have a use in this landscape of cybersecurity. This paper evaluated the effectiveness of various bi-directional transformers–BERT, ALBERT, RoBERTa, and DistilBERT– in classifying malicious URLs. The

performance of these transformer neural network models was assessed on a variety of metrics including F1-Score, Accuracy, Precision, and Recall to determine the best model.

The data gathered shows superior performance by BERT and DistilBERT across all metrics. Both of these transformer models saw high accuracy, precision, recall, and F1 scores, paired with low training and validation loss values, emphasizing their capability in being the most effective at classifying benign and malignant phishing URLs. This could be attributed to the architecture behind each model, as they have some of the highest depth when learning, which seems to be crucial in the context of phishing attempts.

On the other hand, ALBERT saw slightly lower performance metrics in comparison to BERT and DisilBERT but still remains a viable alternative in contexts where computational efficiency is a priority. Its slightly lower performance metrics show a trade-off between slightly less effectiveness and less computation demand, potentially making it better in environments with limited computing power.

Contrastingly, RoBERTa's performance was by far the worst compared to its counterparts. This underperformance is highlighted by its accuracy and precision metrics barely exceeding a random guess, which points in the direction of significant adjustments needing to be made for it to be considered viable in this phishing classification task.

The varying metrics of these transformers models indicate the importance of the choice of machine learning model in the context of cybersecurity threats such as phishing detection. Based on the results of this study, it is clear that transformers are valuable in this domain, and the effectiveness vary substantially based on architecture choices and the specific task to which it is applied to. The results of this experiment answer the research question by concluding that

bi-directional transformer-based neural networks can correctly classify malicious phishing URLs to a great extent.

6. Conclusion and Future Directions

The results of this research have contributed to the field of cybersecurity, specifically in the detection of phishing URLs with deep learning. This study evaluated the effectiveness of different transformer neural network models, namely BERT, ALBERT, RoBERTa, and DisilBERT, in classifying malicious URLs. The evaluation of these models focuses on various metrics, including accuracy, F1-score, precision, recall, and loss, to determine the most viable model in a certain phishing detection context. Analysis of the results found superior performance by BERT and DistilBERT across all metrics, and ALBERT as a resource-friendly alternative. The results also indicate the potential for new software to include these bi-directional transformer neural networks, specifically with ALBERT due to its low computational complexity.

Some limitations of this study included the access to computational resources when training. All models were trained using T4 GPUs in Google Colab which seemed to finish training models at a reasonable time for 1 epoch; however, in the real world many more epochs would be needed, which could alter results. Additionally, the methodology of this study only allowed for one variable to be manipulated (the type of bi-directional transformer used), which could limit the potential performance of these transformer models.

Therefore, further research should try to tune the hyperparameters that were set in this project using better GPUs for training deep learning models. Additionally, further research should explore how these machine-learning models can be incorporated into real-world systems,

such as a browser extension that could classify each link before being clicked. Furthermore, hybrid machine learning models and novel transformer architectures could potentially further improve phishing detection solutions.

References

IBM report: Consumers pay the price as data breach costs reach all-time high. IBM Newsroom. (n.d.). https://newsroom.ibm.com/2022-07-27-IBM-Report-Consumers-Pay-the-Price-as-Data-Breach-Costs-Reach-All-Time-High

Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: A literature survey. *IEEE Communications Surveys & Tutorials*, *15*(4), 2091–2121. https://doi.org/10.1109/surv.2013.032213.00009

Le, Hung, et al. URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection. arXiv, 1 Mar. 2018. arXiv.org, https://doi.org/10.48550/arXiv.1802.03162.

Maneriker, P., Stokes, J. W., Lazo, E. G., Carutasu, D., Tajaddodianfar, F., & Gururajan, A. (2021). Urltran: Improving phishing URL detection using Transformers. *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*. https://doi.org/10.1109/milcom52596.2021.9653028

Marchal, S., Francois, J., State, R., & Engel, T. (2014). PhishStorm: Detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management*, *11*(4), 458–471. https://doi.org/10.1109/tnsm.2014.2377295

National Institute of Standards and Technology. (n.d.). *Phishing*. COMPUTER SECURITY RESOURCE CENTER. https://csrc.nist.gov/glossary/term/phishing

Sahoo, Doyen, et al. Malicious URL Detection Using Machine Learning: A Survey. arXiv, 21 Aug. 2019. arXiv.org, http://arxiv.org/abs/1701.07179.

Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L. F., & Downs, J. (2010). Who falls for phish? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/1753326.1753383

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Retrieved from https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

What is machine learning?. Oracle. (n.d.). https://www.oracle.com/artificial-intelligence/machine-learning/what-is-machine-learning/

Xu, P. (2021). A Transformer-based Model to Detect Phishing URLs. ArXiv. /abs/2109.02138