On the performance of lightweight models in fine-grained classification of malicious URLs

Chia Yu Hong A0235341N[1], **Eric Lee Ying Yao A0230337N**[1], **Hoang Huu Chinh A0243326L**[1], **Justin Widodo A0234942A**[1]

[1]School of Computing, National University of Singapore
{e0727341, e0694459, e0817828, e0726942}@u.nus.edu

Abstract

The proliferation of malicious URLs poses a significant threat to cybersecurity, leading to financial losses, data breaches, and compromised user privacy. According to the Singapore Cybersecurity Agency's 2022 Cyber Landscape report[1], phishing, malware, and defacement attacks remain prevalent in Singapore, with a notable 184% rise in phishing attacks from 2021. Furthermore, as outlined by the report, these trends are mirrored globally. In light of evolving threats, traditional detection methods may be increasingly ineffective, necessitating the exploration of machine learning to develop adaptive classification techniques. We propose a self-contained, lightweight and user-friendly URL classification model designed to be implemented as a browser extension, operating silently in the background as users browse the internet. The model focuses solely on the lexical contents, structure, and semantics of URLs, without relying on requests to the URL or querying of external databases. This approach would address the pressing need for efficient and effective URL classification in the face of growing cybersecurity threats. The project is accessible at a publicly available repository[2].

Introduction

The ever-growing prevalence of cyber threats necessitates efficient and adaptive methods for safeguarding users against malicious URLs. These URLs often aim to steal sensitive information or infect devices with malware. The traditional approach held by default browser features such as Google Chrome's Safe Browsing merely checks URLs against existing databases[3]. Hence, we propose a browser extension that leverages machine learning models to adaptively classify URLs as users surf the internet.

We summarize our contributions as follows:

- We explore the performance of several lightweight models in fine-grained classification of URLs.
- We propose the use of additional static lexical features on top of features identified in prior work.
- We explore features that exploit the use of Pretrained Language Models (PLMs) in lighter models, with reference to SOTA methods proposed in prior work.
- We propose an interpretable ensemble model that leverages the difference in performance of Random Forest classifiers on different negative classes to better generalize on unseen URLs

The proposed browser extension is designed to reach a broad audience. Hence to ensure widespread accessibility and usability, we can categorize the information displayed into two tiers. In doing so, this allows the extension to accommodate with users of varying technical understanding.

At the simple tier, users will receive clear and concise results indicating whether the URL they entered is classified as anything other than benign. This straightforward presentation should provide immediate guidance to users and make them aware of the potential threats. Additionally, simple explanations accompanying each classification will educate users on the significance of different the classifications.

For users with a more technical background, the complex tier offers deeper insights into the classification process. It will provide detailed information on the confidences of the models involved, as well as explanations of how the model classified the URL through the different features of the submodels in the ensemble model. This will provide the users with some level of explainability to answer why the model classified the URL in such a manner.

Related Work

In exploring the performance of lightweight classification models for malicious URL detection, we review some prior work on feature extraction from URL strings. Additionally, we review prior work proposing classification models of different sizes and analyse their performance on classification tasks of different granularity.

Lexical feature extraction

Joshi et al. 2019 proposes a machine learning ensemble classification model that extracts static lexical features from a URL string. An assumption is made that lexical features differ between malicious and benign URL strings. The use of static lexical features is motivated by safety, as execution is

[1] https://www.csa.gov.sg/Tips-Resource/publications/2023/singapore-cyber-landscape-2022

[2] https://github.com/autumn-sonata/CS3264-Project

[3] https://developers.google.com/safe-browsing/

avoided. The lexical features, by URL component, explored for the model are as follows:

- **URL** - length; number of semicolons, underscores, question marks, equals, ampersands;
- **Top level domain** - presence in suspicious list;
- **Primary domain** - contains IP; length; number of digits; number of non-alphanumeric characters; number of hyphens; number of @s; presence in top 100 Alexa domains;
- **Subdomain** - number of dots; number of subdomains;
- **Path** - number of '//'; number of subdirectories; presence of '%20' in path; presence of uppercase directories; number of special characters; number of zeroes; ratio of uppercase to lowercase characters;
- **Parameters** - length;
- **Query** - number of queries;

Results show that the use of trigram-based features with lexical features achieves a low False Negative Rate (FNR) as desired, with a good False Positive Rate (FPR) and accuracy. The proposed Random Forest classifier using these features achieved a good balance of 0.38 FNR and 0.92 accuracy with a maximum depth of 20. Tuning of other hyperparameters did not affect accuracy significantly.

Hierarchical feature extraction with PLMs

Liu et al. 2023 proposes the use of hierarchical feature extraction, layer-aware attention, and spatial pyramid pooling for improved feature extraction from unprocessed URL strings, on top of the CharBERT character-aware pretrained model, for malicious URL detection.

The hierarchical extraction and capturing of features of URLs at local and global levels demonstrated notable improvements in performance over prior models. The model reached 0.9352 accuracy with minimal training data in binary classification, and up to 0.9839 accuracy with more training data. In multi-class classification of malicious URLs (benign, defacement, phishing, malicious), the model achieved an accuracy of up to 0.9838, outperforming prior models, which deteriorate in performance in multi-class classification.

Data

Three publicly available datasets were used in the evaluation of the proposed methods, differing based on the granularity of classification of malicious URLs.

Multi-Class Dataset

The multi-class dataset was obtained from the Kaggle Malicious URLs dataset.[4] This dataset is classifies URLs into one of four classes, benign (428,103), defacement (96,457), phishing (94,111), and malware (32,520). URL data is sourced from a variety of sources, such as the URL dataset (ISCX-URL2016)[5], Malware domain black list dataset[6], and the PhishStorm dataset[7]. As one of the few datasets that classify URL data to this level of granularity, this dataset serves as the focus of our investigation on the finer classification of Malicious URLs.

Benign Dataset

The benign dataset was obtained from the Kaggle Malicious And Benign URLs dataset.[8] This dataset classifies URLs as benign (345,738) or malicious (104,438). URL data for this dataset was acquired from various sources, such as PhishTank. The benign URLs from this dataset were used to extensively test the positive classification (benign) of URLs.

Phishing Dataset

The phishing dataset was obtained from the PhishTank database.[9] URLs in this dataset are classified as phishing (59,282). This dataset serves to extensively test the negative classification (phishing) of URLs.

The choice of each dataset is motivated by their focus on either multi-class or binary classification of malicious URLs. The multi-class dataset is used to investigate feature extraction and modelling for fine-grained classification. Binary datasets are used for cross-dataset validation, and serves to examine the ability of the proposed methods to generalize on unseen data. As reliable data on URLs classified as defacement and malware is scarce, cross-dataset validation focused on benign and phishing data as falsely classifying benign URLs as malicious would be detrimental to users, and phishing URLs were observed to have significantly higher FNRs among negative classifications in multi-class classification.

Methodology

In developing the proposed methods for URL classification, small models were prioritised, with a focus on static feature extraction, classification speed, and a lack of reliance on external databases. These considerations are motivated by several factors:

- **Usability** - the proposed model should be usable in practical settings. A small model that is able to classify URLs quickly is more suited for practical use. A motivating example would be the deployment of the model as a browser extension; a lightweight model would be able to be deployed with relative ease, and quick classification would encourage users to utilize the model classification in aiding their decision on the safety of visiting a particular URL.

[4] https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

[5] https://www.unb.ca/cic/datasets/url-2016.html

[6] http://www.malwaredomains.com/wordpress/?page_id=66

[7] https://research.aalto.fi/en/datasets/phishstorm--phishing--legitimate-url-dataset(f49465b2-c68a-4182-9171-075f0ed797d5).html

[8] https://www.kaggle.com/datasets/siddharthkumar25/malicious-and-benign-urls

[9] https://phishtank.org/developer_info.php

- **Safety** - feature extraction should avoid making any requests to the URL, in the case that the URL is malicious. Static feature extraction is useful in this case as there is no interaction with the URL itself, and prevents any potential harm to the user.
- **Maintainability** - the model should be able to achieve a consistent performance with minimal maintainability. A lack of reliance on external databases contributes to this factor, as storing an up-to-date list of malicious or benign URLs would require frequent updates to the model.
- **Interpretability** - one concern in model development was the interpretability of model performance. Models which allow some form of analysis on the significance of features used in classifying malicious URLs were preferred as they allowed some insight into what features of a URL contribute to a particular classification.

Feature Extraction

Static feature extraction from URLs focused on two aspects, lexical feature extraction based on URL components, and tokenized embeddings of URLs.

Lexical feature extraction is motivated by their performance in binary classification of malicious URLs explored in Joshi et al. 2019, and analysis on how such features perform in multi-class classification tasks. This focus on the extraction of component features of URLs is based on the intuition that different classes of URLs would exhibit differences in lexical features. Features proposed in Joshi et al. 2019 that were explored are shown in Table 1. Features requiring a lookup in an external list were not considered. The use of n-grams, motivated by the performance improvement with URL trigrams in Joshi et al. 2019 was explored, however, performance deteriorated in every case of n-grams from n=1 to n=3, likely due to significant variation in URL structure. n-gram features were thus excluded. Additional features extracted through further exploration are shown in Table 2.

Tokenized embeddings are motivated by the PLM based architecture proposed in Liu et al. 2023. The embeddings are extracted with the DistilBERT Tokenizer,[11] which utilizes wordpiece tokenization based on a learned vocabulary to decompose a string into components. Similar to the proposed lexical feature extraction, the intent is the exploitation of component differences in different classes of URLs for classification with lightweight PLMs.

Models

Two types models were considered for evaluation, a Random Forest of Decision Tree Classifiers,[12] and a pretrained DistilBERT (cased) model.[13]

Component	Feature
URL	Length ’;’, ’_’, ’?’, ’=’, ’&’ count Digit to letter ratio
Primary domain	Contains IP Length Digit count Non-Alphanumeric count ’-’ count ’@’ count
Subdomain	’.’ count Subdomain count
Path	’//’ count Subdirectory count Contains ’%20’ Contains uppercase directory Contains character directory Special character count ’0’ count Uppercase to lowercase ratio
Parameters	Length
Query	Query count

Table 1: Joshi et al. 2019 Features

Feature
’www’ count
Digit count
’.’ count
’/’ count
’-’ count
’=’ count
’&’ count
’_’ count
Parameter count
URL entropy[10]
Domain extension

Table 2: Additional Features

Random Forest Classifier: A Random Forest of Decision Tree classifiers was used to classify malicious URLs based on the extracted lexical features. This classifier was chosen as it averages over a number of Decision Tree classifiers to help prevent overfitting and improve accuracy, which would be useful in the classification of URLs as they tend to be noisy. The classifiers are also interpretable as it is possible to trace the decision path of the classifiers in the Random Forest to identify which features were considered in the classification. Furthermore, the classifier is able to output a classification quickly, which suits the use case of our model.

DistilBERT (cased): A pretrained DistilBERT (cased) model was chosen as part of an exploration of SOTA models and how they compare to less complex classification models. This choice was motivated by the results of Liu et al. 2023, and seeks to analyse the performance of a comparatively simple, lightweight, transformer-based BERT model

[10] https://towardsdatascience.com/predicting-the-maliciousness-of-urls-24e12067be5

[11] https://huggingface.co/docs/transformers/en/model_doc/distilbert#transformers.DistilBertTokenizer

[12] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[13] https://huggingface.co/distilbert/distilbert-base-cased

in multi-class classification tasks. Additionally, this serves as an exploration of the effectiveness of transfer learning of PLMs in classifying URLs, as the semantic and contextual relation between the components of a URL differs from similar relations between tokens in natural language, which such PLMs are trained on. The cased version of DistilBERT was chosen over the uncased version as the casing of text in URLs is a feature that could be exploited in classification.

Experiments

Experiments are conducted in a Python 3.11.8 environment. Performance is evaluated with a priority on FNR and Accuracy, with Precision, Recall and F1-score metrics also being collected. FNR was prioritised as false negative classification of malicious URLs would be detrimental to users. Similarly, accuracy was prioritised as misclassification of positive or negative classifications would be detrimental. Fine-tuning of DistilBERT also used CUDA 10.8, and was performed on Kaggle with P100 GPUs.

Setup

Hyperparameters for Random Forest Classifiers were found with GridSearchCV, and were tuned at 50 estimators and a max depth of 20. Fine-tuning of DistilBERT was done with a batch size of 32, and the AdamW optimizer with a learning rate of 2e-5, over 3 epochs.

The Kaggle Malicious URLs dataset is split into training (80%) and test (20%) sets. Feature extraction is applied on all data based on model type, with lexical feature extraction for Random Forest models and tokenized embeddings for DistilBERT. Each model is trained on the training data before being evaluated on the test set.

Cross-Dataset validation is done with models trained on the Kaggle Malicious URLs dataset to analyse the ability of the models to generalize. The benign URLs from the Kaggle Malicious And Benign URLs dataset was used to validate the ability of the trained models to generalize on benign URLs. Phishing URLs from the PhishTank database were used to do the same for phishing URLs.

The following models were explored:

- ***rf-general*** is a RandomForestClassifier that uses features in Table 2, in addition to the following lexical features from Table 1: URL - Length, Digit to letter ratio; Subdomain - Subdomain count; Primary domain - Digit count, Non-alphanumeric count; Path - Contains uppercase directory, Special character count, Uppercase to lowercase ratio;
- ***rf-minimal*** uses a minimal set of features in *rf-general*, URL entropy, Domain extension, and '/' count, that achieves reasonable performance with a RandomForestClassifier.
- ***rf-lexical*** uses features in Table 1 with a RandomForestClassifier and serves as a baseline for comparison, as well as a exploration on the performance of selected features in Joshi et al. 2019 on multi-class classification tasks.
- ***rf-ensemble*** is an ensemble model of *rf-general*, *rf-minimal* and *rf-lexical*. As each model was observed to perform differently in cross-dataset validation, *rf-ensemble* makes a decision based on the output of selected models in an attempt to generalize better on unseen URLs via an algorithmic approach. This approach was also considered as each individual model is relatively small, and is able to output classifications quickly. The model attempts to leverage differences in constituent model performance on different negative classifications to balance overall performance in fine-grained negative classification.
- ***DistilBERT*** is a lightweight model pre-trained on the same corpus as the larger BERT model, with the goal of learning the same representation while being faster at downstream tasks. The cased version was used to account for casing in URLs, and DistilBertForSequenceClassification, which adds a classification head to the model, was used to adapt the model for the classification of URLs.

Multi-class Classification Results

The results on the Kaggle Malicious URLs dataset are shown in Table 3. The *rf-lexical* baseline originally intended for binary classification of URLs performed well in mutli-class classification, achieving an overall accuracy of 0.9031. It was able to achieve a low FNR of 0.0355 for positive (benign) classifications, but seemed to strugggle with the finer classification of negative classifications, with a 0.1578 FNR for defacement, 0.3030 FNR for phishing, and 0.1313 FNR for malware.

This trend seems to be common across the different classifiers, with FNRs for benign being low, and FNRs for the negative classifications being higher, with phishing FNRs being the highest.

rf-minimal was able to achieve comparable FNRs in benign and malware classification at 0.0469 and 0.1976 respectively with only 3 features, significantly less than the 21 features used in the baseline. FNRs of defacement and phishing saw significant deterioration, suggesting that the selected features were insufficient in differentiating URLs of these classes.

The *rf-general* model saw improvements in FNR across all classes, with 0.0121, 0.0587, 0.1927 and 0.0984 FNRs for benign, defacement, phishing and malware respectively. The relative improvement in FNR over the base model was noticeably lower for phishing and malware classes, at 0.6360 and 0.7494 of the *rf-lexical* FNRs respectively, compared to 0.3408 and 0.3720 of the benign and defacement FNRs in the baseline. Overall accuracy also saw an improvement of 0.0476 over the baseline at 0.9507, with per-class accuracies seeing improvements overall, with benign accuracy seeing the biggest improvement of 0.0461 over the baseline at 0.9594. Notably, in comparison with other Random Forest models, *rf-general* achieved the lowest FNRs in defacement, phishing, and malware, and achieved the highest accuracies across all classes.

rf-ensemble performance also saw improvements over the *rf-lexical* baseline, with noticeable improvements in benign and defacement FNRs at 0.0071 and 0.0973, and slight improvements in phishing and malware NFRs. The ensemble

Model	Class	FNR	Accuracy	Precision	Recall	F1-score
rf-general	*overall*	-	**0.9507**	0.9503	0.9507	0.9496
	benign	0.0121	**0.9594**	0.9522	0.9879	0.9697
	defacement	**0.0587**	**0.9856**	0.9607	0.9413	0.9509
	phishing	**0.1927**	**0.9616**	0.9153	0.8073	0.8579
	malware	**0.0984**	**0.9948**	0.9946	0.9016	0.9458
rf-minimal	*overall*	-	0.8239	0.8201	0.8238	0.8101
	benign	0.0469	0.8381	0.8270	0.9531	0.8856
	defacement	0.3441	0.9242	0.7959	0.6559	0.7191
	phishing	0.5867	0.8978	0.7684	0.4133	0.5375
	malware	0.1976	0.9878	0.9479	0.8024	0.8691
rf-lexical	*overall*	-	0.9031	0.9018	0.9031	0.9007
	benign	0.0355	0.9133	0.9092	0.9645	0.9360
	defacement	0.1578	0.9629	0.9009	0.8422	0.8705
	phishing	0.3030	0.9369	0.8367	0.6970	0.7605
	malware	0.1313	0.9931	0.9939	0.8687	0.9271
rf-ensemble	*overall*	-	0.9319	0.9337	0.9319	0.9290
	benign	**0.0071**	0.9375	0.9187	0.9929	0.9544
	defacement	0.0973	0.9822	0.9750	0.9027	0.9375
	phishing	0.3011	0.9501	0.9378	0.6989	0.8009
	malware	0.1128	0.9941	0.9954	0.8872	0.9382
DistilBERT	*overall*	-	0.9711	0.9724	0.9711	0.9715
	benign	0.0274	0.9769	0.9920	0.9726	0.9822
	defacement	0.0082	0.9970	0.9883	0.9918	0.9900
	phishing	0.0590	0.9730	0.8795	0.9410	0.9092
	malware	0.0235	0.9953	0.9340	0.9765	0.9548

Table 3: Model Performance on Kaggle Dataset
(*benign, defacement, phishing, malware*)

model also achieved the lowest NFR in benign classification among Random Forest models at 0.0071, a 0.0050 improvement over the next lowest with *rf-general*.

The *DistilBERT* model achieved excellent performance in the multi-class classification task. Despite its relative simplicity compared to the *CharBERT*-based model proposed in Liu et al. 2023, it achieved an overall accuracy of 0.9711, with precision, recall and F1-scores of 0.9724, 0.9711 and 0.9715, which is comparable to the roughly 0.99 overall performance achieved by the Liu et al. 2023 model. Compared to the Random Forest models, the *DistilBERT* model achieved a more consistent FNR and accuracy across different classes, and notably managed to achieve lower FNRs for defacement, phishing, and malware. This suggests that the *DistilBERT* model was able to leverage transfer learning from its pretraining in the classification of malicious URLs, and URL classification can benefit from standard natural language processing techniques as well.

Model	Dataset	FNR	Accuracy	F1-score
rf-general	*benign*	0.9772	0.0228	0.0447
	phishing	**0.1728**	**0.8272**	0.9054
rf-minimal	*benign*	**0.1875**	**0.8125**	0.8966
	phishing	0.8535	0.1465	0.2555
rf-lexical	*benign*	0.2775	0.7225	0.8389
	phishing	0.6945	0.3055	0.4680
rf-ensemble	*benign*	0.4056	0.5944	**0.7456**
	phishing	0.3213	0.6787	**0.8086**
DistilBERT	*benign*	0.9528	0.0471	0.0900
	phishing	0.0947	0.9053	0.9503

Table 4: Model Performance on External Datasets
(*benign, phishing*)

Cross-Dataset Validation Results

Cross-dataset validation results are shown in Table 4. When validated on the benign and phishing datsets, most models were observed to generalize significantly worse on either benign or phishing classification.

rf-general generalized badly on the benign dataset with an FNR of 0.9772 and an accuracy of 0.0228, but managed to achieve a 0.1728 FNR and an accuracy of 0.8272 on the phishing dataset, the best phishing classification performance among the Random Forest models.

rf-minimal achieved good generalization on the benign dataset with an FNR of 0.1875, and an accuracy of 0.8125, the best benign classification performance among Random Forest models. However, as observed, the phishing classification performance suffered at 0.8535 FNR and 0.1465 accuracy.

The *rf-lexical* baseline achieved a more balanced FNR and accuracy compared to the previous Random Forest models at 0.2775 and 0.7225 for benign and 0.6945 and 0.3055 for phishing. However, generalization performance was still skewed heavily in favor of benign classification.

As an ensemble model, *rf-ensemble* was able to leverage the ability of its constituent models to classify certain classes well, and was able to achieve an FNR and accuracy of 0.4056 and 0.5944 for benign and 0.3213 and 0.6786 for phishing classification. It was also able to achieve good, balanced F1-scores at 0.7456 and 0.8086 for benign and phishing respectively.

DistilBERT suffered the same skewed generalization performance with an FNR and accuracy of 0.9529 and 0.0471 for benign, and 0.0947 and 0.9053 for phishing. Compared to the cross-dataset validation performance of the CharBERT-based model proposed in Liu et al. 2023, *DistilBERT* generalized poorly in comparison. This suggests that classifying URLs with a standard PLM fine-tuned on the original dataset may generalize poorly on unseen URLs, perhaps due to inherent differences in URL and natural language structure. Alternatively, the pretraining objective of DistilBERT, where it replicates the same probabilities and states of the teacher model, [14] could hinder its ability in developing a proper representation of the language model, which could be a factor in the poor generalizability of the model in the benign classification task.

Ethics

When considering the ethical implications of our machine-learning experiments, one potential problem is biased or discriminatory outcomes, particularly if the models are trained on biased datasets or if the features selected inadvertently encode biases. This can result in a biased classification of certain URLs, leading to misclassifications with our proposed model that ultimately would impact users' access to resources.

To address this, from the beginning, we need to research and set a high standard for the datasets that we are planning to use for training. Additionally, as we want to put our trained model to use, it should be our responsibility to showcase the transparency of how the browser extension gathers users' URLs and runs classification on it since URL gathering may inadvertently reveal sensitive information about individuals' online activities. To mitigate this risk, we will make our browser extension open-source. This will enhance transparency by allowing external scrutiny of the algorithms and processes happening inside the extension. Moreover, it's also crucial to make sure that the browser extension is secured in a way that user information, including their accessed URLs cannot be exploited by any third party without users' consent.

Conclusion

The experiments conducted showcase the efficacy of different models in the fine-grained classification of URLs. On the Kaggle Dataset, the Random Forest models, *rf-general*, *rf-minimal*, and *rf-lexical*, demonstrate differing performance based on the features utilized. Notably, *rf-general* outperforms the others by incorporating additional features from Table 2, showcasing enhanced accuracy and lower FNRs in defacement, phishing, and malware classifications as compared to the baseline model *rf-lexical*. Conversely, the *rf-minimal* model, characterized by a reduced feature set while retaining acceptable performance, despite demonstrating notable strengths in classifying benign and malware URLs as compared to *rf-lexical*, performs worse than *rf-general*. This discrepancy suggests a relationship between the features considered in *rf-general* but not in *rf-minimal* that is significant in distinguishing between classes of URls.

These observations underscore the pivotal role of feature selection in optimizing model performance, highlighting how the inclusion or exclusion of specific features can significantly impact the ability to accurately classify URLs across different categories. The *rf-ensemble* model, a combination of the *rf-general*, *rf-minimal* and *rf-lexical* models, seeks to capitalize on this, and manages to perform reasonably well on the original dataset, but is overshadowed by *rf-general* in performance, with the exception of FNR, in classifying benign URLs. However, when it comes to external datasets, *rf-ensemble* emerges as the model with the most balanced results, unlike *rf-general* which exhibits significantly high FNR on benign URLs, potentially triggering unnecessary alerts for users when accessing benign URLs. Hence, *rf-ensemble* is most suitable for use in our browser extension as its performance is the most stable and achieves the best generalizability on external datasets or unseen data.

Roles

Chia Yu Hong A0235341N

- Feature extraction
- *rf-lexical* modelling
- Report - *Related Work*, *Data*, *Methodology*, *Experiments*

Eric Lee Ying Yao A0230337N

- Feature extraction
- *rf-general*, *rf-minimal*, *rf-ensemble* modelling

Hoang Huu Chinh A0243326L

- Browser extension development
- Report - *Ethics*, *Conclusion*

Justin Widodo A0234942A

- Sourcing datasets
- Feature extraction
- Hyperparameter tuning
- *DistilBERT* modelling
- Report - *Abstract*, *Introduction*

References

Joshi, A.; Lloyd, L.; Westin, P.; and Seethapathy, S. 2019. Using Lexical Features for Malicious URL Detection – A Machine Learning Approach. arXiv:1910.06277.

Liu, R.; Wang, Y.; Xu, H.; Qin, Z.; Liu, Y.; and Cao, Z. 2023. Malicious URL Detection via Pretrained Language Model Guided Multi-Level Feature Attention Network. arXiv:2311.12372.

[14] https://huggingface.co/docs/transformers/en/model_doc/distilbert