Transformer RoBERTa vs. TF-IDF for websites content-based classification

Lahcen Yamoun, Zahia Guessoum, Christophe Girard

▶ **To cite this version:**

Lahcen Yamoun, Zahia Guessoum, Christophe Girard. Transformer RoBERTa vs. TF-IDF for websites content-based classification. Deep Learning meets Ontologies and Natural Language Processing, International Workshop in conjunction with ESWC, 2022, Hersonissos, Greece. hal-03725602

HAL Id: hal-03725602
https://hal.science/hal-03725602

Submitted on 22 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Transformer RoBERTa vs. TF-IDF for websites content-based classification

YAMOUN Lahcen[1,2], GUESSOUM Zahia[1,3], and GIRARD Christophe[2]

[1] CReSTIC EA 3804, University of Reims Champagne Ardenne, France
[2] Efficient IP, La Garenne-Colombes, France
[3] LIP6 UMR 7606, Sorbonne University, France

Abstract. The web is growing day by day, and with it the need to categorize websites, for different purposes that can serve the end user, or the providers, among others web filtering. Different classification techniques can be used, such as the exploitation of the textual content of a website. In this paper, we use RoBERTa transformer, a variant of BERT, in its pre-trained version without finetuning, to represent websites. We compare these embeddings with TF-IDF features. Two approaches are studied, a mono-multiclassification into 16 classes, and a binary approach with the one vs. all strategy, and different classifiers are tested, the best by results being a 3 layers fully connected neural network. Tests show better results with RoBERTa embeddings compared to TF-IDF features. They provide an accuracy of 68% for the mono-multiclassification, and an average of 90.69% for the binary classifications with an accuracy of 100% for the pornographic websites.

Keywords: Web classification · RoBERTa transformer · TF-IDF · Sentence embeddings · Text classification.

1 Introduction

The WEB today is present everywhere and in a large way. It is used as an interface for multiple services, such as the cloud and the email. A variety of subjects can be thus found on the WEB, and different needs arise from it. Hence the categorization of web pages in order to facilitate the exploitation, at the end-user level, of the important amount of related data, for example only choose preferred topics; facilitate the processing and the knowledge extraction from data at the service vendor level; and allow the filtering of content according to pre-established criteria, for example the restriction of access to adult or hateful content.

The aim of this paper is to classify websites. This machine learning task, i.e., classification, is a supervised task. To achieve this task, a model learns from input data that has been labeled. In the case of the WEB, the labeling can take different forms depending on the context and the need, for example a classification into malicious or reliable websites, or a classification according to the topics found in the websites, and it is the latter that we studied in this

work. The input can also take different forms. In the case of classification of the WEB according to the content, and more precisely according to the text, different studies make different proposals to make the representation, in this work we study the contribution of the embedding of a pre-trained RoBERTa without finetuning compared to TF-IDF features. At the best of our knowledge, our work is the first to use RoBERTa for the task of classifying WEB, and the second to use transformers for this task, the first being the authors of [6], where BERT is used.

In this paper, we first define TF-IDF and RoBERTa, and describe related works and analyse the different approaches to classify websites. We then introduce our approach, explain the way we built the database, and describe the general architecture used for the classification. We will end by discussing the obtained results in terms of mono-multiclassification into 16 categories, and in the 16 binary classifications according to the one vs. all strategy.

Our work can be used in the filtering of websites, since we have reached a validation accuracy of 100% on the classification of adult content websites.

2 Definitions

In this section, we will briefly give the essential definitions for the rest of the paper.

TF-IDF: *term frequency, inverse document frequency*, an alternative to term frequency feature vector. It's a formula that aims to define the importance of a keyword or a phrase within a document or a web page. TF-IDF weights calculated from equation 1 intend to give higher weights to terms which appear in fewer documents and lower weights to terms occurring in many documents. This is achieved by multiplying a term's frequency described in Equation 2 by an inverse document frequency (IDF) factor described in Equation 3.

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D) \tag{1}$$

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2}$$

$$idf(t, D) = -log(\frac{|d \in D : t \in d|}{|D|}) \tag{3}$$

In Equation 1, $tfidf(t, d, D)$ is the weight of the term t in the document d according to the corpus of documents D. In Equation 2, $tf(t, d)$ is the relative frequency of term t in document d. In Equation 3, the inverse document frequency is a measure of how much information the term t provides, i.e., if it is common or rare across the corpus D. It is the logarithmically scaled inverse fraction of the documents that contain the term t.

Different other variants exist to calculate the term frequency and the inverse document frequency.

Transformers: A transformer is a deep learning model that adopts the mechanism of self-attention [21], differentially weighting the significance of each part of the input data. It is used primarily in the fields of natural language processing (NLP) and computer vision. For a given sentence for example, the model extracts features for each word using a self-attention mechanism to figure out how important all the other words in the sentence are with regard to the word itself. Like recurrent neural networks (RNNs), transformers are designed to handle sequential input data, such as natural language, for tasks such as translation and text summarization. However, unlike RNN-like models, no recurrent units are used to obtain these features, since transformers use only weighted sums and activation values, and they do not necessarily process data in order, thus, they can be very parallelizable and efficient.

BERT: Bidirectional Encoder Representations from Transformers (BERT) [7] is a transformer-based machine learning technique for NLP pre-training developed by Google. BERT, at its core, is a transformer language model with a variable number of encoder layers and self-attention heads. The architecture is almost identical to the original transformer implementation in [21]. It is a revolutionary technique that achieved state-of-the-art results on a range of NLP tasks while relying on unannotated text drawn from the web, as opposed to a language corpus that's been labeled specifically for a given task. The technique has since become popular both as an NLP research baseline and as a final task architecture.

RoBERTa: A Robustly Optimized BERT pre-training Approach (RoBERTa) [13], is an improved version of BERT, where key hyper-parameters are modified, some pre-training tasks are omitted, and pre-training is done with higher learning rates and much larger mini-batches.

3 Related Work

In contrast to text classification where different methods are well established and comparisons are made, website classification is relatively less pronounced, and the number of works is much reduced. This being said, the latter has been treated in different ways in the literature, basically, there are old works that looked at the classification of URLs as strings, and more recent works that classified web pages according to a mixture of several criteria, according to [18] these criteria fall in one of the following categories: content-based features, including text, images, styles and scripts; blacklist features, meaning wether the website is present in blacklists or not; lexical features; host-based features; third-party features, as example Alexa ranking.

From the DMOZ database, the idea of using an n-gram with different ranks was proposed to constitute the features of the urls [4]. In this work, SVM, Naive Bayes, Maximum Entropy gave comparable results. In [16] all possible 3-grams have been used, to constitute a space of tokens independent of the dataset, Term-Frequency has been used afterwards with SVM and Maximum Entropy to classify the urls. In [12], the authors focused on creating better embeddings for urls, and

opted for what they called URLNET architecture, a character and word level CNNs. With this technique, even unknown words have a unique representation due to the character level CNN. They then used these embeddings for binary classification of urls into malicious or legitimate url, without proceeding to a stopwords removal. 78 lexical features were introduced in [14] and used in [10] to classify urls as malicious or legitimate. Random Forest gave better results than neural networks.

URL-based features are found in many works, including [17] [3] [23] [19] [9] [20]. The authors combined the URL features with other features, in [17] among other set of features, we find anomaly features for the malicious websites classification, in [3] The authors claim to have proposed 3 new features that significantly improve the classification of malicious websites, which are: Google page rank, Google position of website title, and Alexa rank. Still for the classification of malicious websites, in [23] authors proposed and studied the impact, advantages and disadvantages of 58 new features based on url, online characteristics, and webpage content characteristics.

Different works have been based on the textual content of the websites to make the classification, we have summarized them in Table 1. In [11], in order to build a database, the authors extracted texts from a list of websites, and then according to a list of keywords per class they automatically classified their database via a voting system. To train the classification model they considered BoW embeddings and a feature selection method, and concluded that a fully connected two layer network gives better results than SVM or random forrest models. We also find this idea of using keywords for classification in [15]. LSTMs were used in [5], the authors used text and some meta-tags of websites: title, description, keywords. 5 layers LSTM model was trained on a maximum length sequences of 100, with BoW embeddings to achieve almost 85% accuracy on a classification of 23 classes database. The work [6] is, to the best of our knowledge and according to the authors themselves, the only one using transformers (among others BERT) for the task of website classification. The authors did not give details on which text they used from the websites, and whether they used meta-tags or not, however, on the 5000best dataset [1], containing 5000 websites classified in 32 categories, they compared the results obtained from BERT, from an LSTM with pretrained GloVe embeddings, and from an LSTM with pretrained GloVe embeddings and char level embeddings concatenated. BERT gave the best results, 67.81% in terms of accuracy.

4 Proposed Approach

In this work, we tested several classification models on a database of 3023 websites categorized into 16 classes, mainly a fully connected neural network with 3 layers (input, hidden layer, output). Two classification approaches were developed, the first one being a mono-multi-classification, and the second one, being different binary classifiers for each category, proceeding by the one vs. all strategy.

Work	Task	Technique	Dataset	Results
[11]	Classification of blogs into: technical, fashion and news	Bag of Words for features, and logistic regression-recursive feature elimination. For classification the authors compared support vector machine, random forrest and a 2 hidden layers fully connected network.	1870 blog websites categorized automatically	The fully connected network gave the best accuracy: 94.36%
[5]	Multi classification to 23 categories	They extracted from websites texts and metatags: title, description and keywords. For classification they used a 5 layers LSTM with GloVe embeddings.	A subset of Roksit's database [2]: 887195 samples categorized in 23 categories	Accuracy: 86.18%
[15]	Classification of news articles to predefined categories such as sport and economy	Extraction of keywords from the news feed and comparison with predetermined keywords obtained from WordNET library [8].	12020 articles from various Indian news web portal	Mean F1 measure: 10.5% Mean Precision: 46.9% Mean Recall: 5.98%
[6]	Classification of websites to 32 categories	3 models were proposed: LSTM with pretrained GloVe embeddings, LSTM with pretrained Glove embeddings and char level embeddings, BERT	5000best dataset [1]	Accuracy: 67.81%

Table 1. Table summarizing the work done for the classification of websites based on textual content.

To represent the websites, two embedding policies have been implemented: the embeddings resulting from the RoBERTa model, and TF-IDF embeddings to make comparison and see if there is a benefit.

In the following, we start by detailing the used data, and the way we collected, preprocessed them, and the augmentation techniques we applied. We then explain the architecture of the classification model.

4.1 Data Collection and Preprocessing

To constitute the pairs (website, category), we referred to the 5000best [1] website regrouping 5000 things, including websites. The websites are chosen according to the popularity, and are classified into 32 categories, such as Porn, News,

Technology, etc. However, we encountered two problems with this database, the first one is the presence of several down websites, and the second one is that the database is strongly unbalanced as shown in the Figure 1.

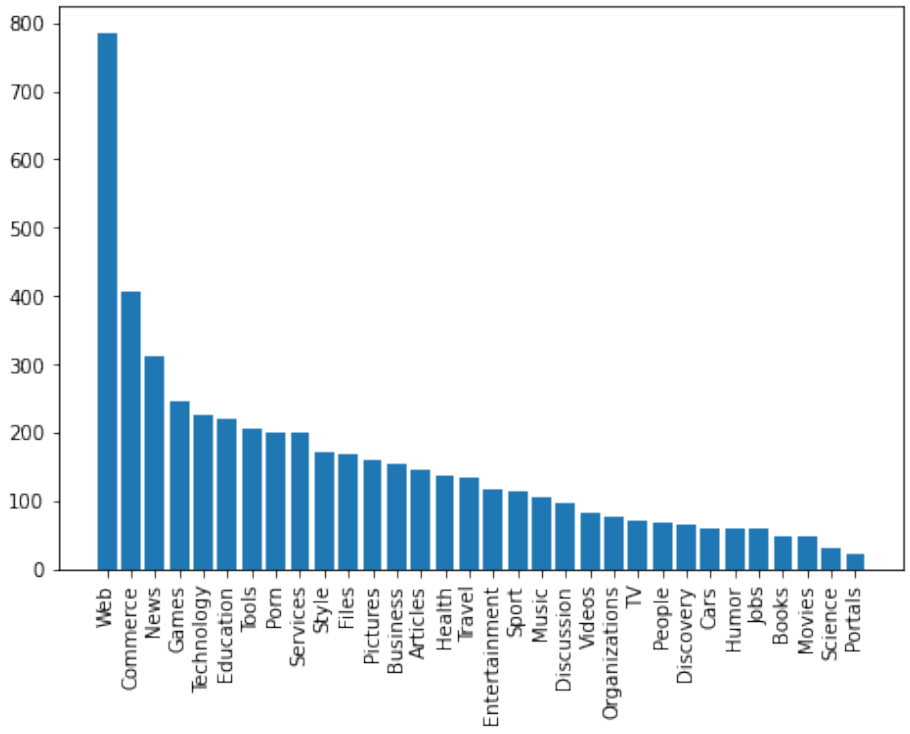

Fig. 1. Categories distribution according to 5000best [1].

So, a first work to prepare our dataset was to eliminate the dead links. It was enough to use a script making HTTP get requests and react according to the responses status. Then, in order to extract the textual content of the well-reachable websites, we use the playwright browser automation framework under NodeJS to do the scraping. The aim of using a browser automator instead of using old scrapping techniques is to allow the generation of the website at the client level by allowing the execution of Javascript, what we call client side rendering, thus having a more precise rendering.

Then, we keep only the websites belonging to the best 16 categories in terms of the number of websites. Since a category like Portals which contains only 10 websites after dead-links elimination is practically impossible to be incorporated into our training pipeline with a category like Web which contains more than 750 websites.

The textual data taken from each website are the description and keyword metadata, the page title, the titles from h1 to h5, the texts forming a link, and then the texts under the tags div, span, p. All these texts are concatenated and separated by spaces.

We so obtain a database of 3023 rows and 3 columns: website URL, website textual content, category. Then we applied back translation and Easy Data Augmentation (EDA) techniques [22] on 70% of the data dedicated to the training (2117 as train set, and 906 as validation set):

1. Back translation: we started by translating the texts into a language other than the original, i.e., English, we chose for this aim French, and then we made a back translation of the result into English, this allows a reformulation of the text. For that we used the Google translation API.
2. Synonym Replacement (SR): a technique in which we replace a random proportion of words by their synonyms. For the synonyms we opted to use the WordNet corpus, and the proportion was fixed to 1/5 of the number of total text words which are not stop words.
3. Random Insertion (RI): finds a random synonym of a random word in the sentence that is not a stop word. Insert that synonym into a random position in the sentence, and repeat this n time, n being a parameter. In our work, we set n to 1/10 the number of non stop words in the text.
4. Random Deletion (RD): randomly removes each word in the sentence with probability p, in our work we set p to 1/10.
5. Random Swap (RS): randomly chooses two words in the sentence and swaps their positions, we do this n times, n being a parameter, in our work we set it to 1/10 the number of non stop words in each text.

The result is: 12702 rows for training containing 2117 original data and 10585 augmented data. The remaining 906 lines with no augmentation were used as validation data.

To train the binary classifiers, we take for each of the 16 categories all the positive samples, and randomly the same number of negative samples.

4.2 Proposed Architecture

In our work, from each input, we apply a text cleaning, followed by a tokenization. For each result we create a representative vector for the text, and this vector is then injected into the classification model. This classical pipeline is given in Figure 2.

Fig. 2. Classification architecture.

Text cleaning consists in keeping only alphanumeric characters. We eliminate punctuation and line breaks by replacing them with spaces. We also eliminate stop words even if there is a loss of useful context in the case of RoBERTa, but we did it with a perspective of having a maximum of important words within the limit of the RoBERTa maximum sequence length, which is 512 tokens. The list of stop words we used is the one proposed by the Python library: Nltk. Texts are lowercased after that.

For tokenization, byte-level BPE tokenizer is used for the RoBERTa model, and to obtain TF-IDF embeddings, words are first derived by splitting texts by spaces.

To extract the embeddings with RoBERTa model, we use the frozen pre-trained model, since the size of the database, and the computational power we have, do not allow finetuning. The CLS token of the last layer is used to represent the input text. To obtain the TF-IDF embeddings, we use the top 10000 features ordered by term frequency across the corpus.

As classifier, we use a 3 layered fully connected network, a first layer of length 768 in the case of RoBERTa embeddings, and 10000 in the case of TF-IDF. A hidden layer of length 64 in the case of the mono-multiclassification and of 32 in the case of binary classification as the experiments we show that these are the sizes that give the best results. And a last layer for the output, of size 16 for the multiclassification, or 2 for the binary classification.

We use ReLU activation for the hidden layer, and softmax for the output. A dropout with probability 0.15 was used for the hidden layer. The cross entropy error function is used in its weighted version. The Adam optimizer is used with an initial learning rate of 0.01, this learning rate is divided by 10 every 30 epochs.

The training is performed in a minimum of 100 epochs with an early stopping strategy, and a maximum of 200 epochs. The training batch size is 16.

5 Results and Discussion

We implemented two approaches to test the added value of RoBERTa embeddings compared to TF-IDF features. The first one is a mono-multiclassification task into 16 classes; and the second one is different binary classification tasks for each of the 16 classes following the one vs. all strategy. Table 2 and Table 3 highlight the main results, and show that RoBERTa embeddings clearly outperform TF-IDF features in the binary classifications, with a difference in accuracy of +5.41%. For mono-multiclassification the results obtained with the RoBERTa and TF-IDF embeddings were practically the same.

Table 2 shows that the classification model gives better results for some classes compared to others, e.g. we get 100% validation accuracy for the *Porn* category, but only 79.70% for the *Articles* category. This is due to the nature of the category itself: categorizing a page containing pornographic content is relatively easier, since a lot of keywords make the detection simple, which is not true for the *Article* category, where the web page contains several pieces of

Label/Embeddings	RoBERTa embeddings	TF-IDF features
Porn	**100.00%**	92.70%
Services	**83.30%**	77.10%
Games	92.00%	**93.80%**
Pictures	**92.20%**	81.20%
Tools	86.70%	**87.50%**
Web	86.70%	**90.10%**
Articles	**79.70%**	78.10%
Technology	**91.70%**	80.20%
News	90.20%	**92.20%**
Sport	**96.90%**	93.80%
Style	**97.90%**	85.90%
Education	86.50%	**87.50%**
Business	**92.20%**	79.70%
Commerce	85.60%	85.60%
Entertainment	**95.80%**	78.10%
Health	**93.80%**	81.20%
AVG	**90.70%**	85.29%

Table 2. Accuracies of the binary classifications with the one vs. all strategy. Last line is the average of all the accuracies

	Accuracy	Top-3 Accuracy
RoBERTa embeddings	68%	88.83%
TF-IDF features	68.2%	86.7%

Table 3. Mono-multiclassification results.

information potentially belonging to several different topics, resulting in considerable noise and a degradation of the classification results. Also, by consulting the classification results on the validation data, we found examples where for instance a *sports articles* site is classified in the *Sport* category but the ground truth label is *Articles*. There are plenty of examples of this kind, which shows the limit of mono-classification for this kind of tasks.

It has been confirmed that using binary classifications gives better results than a mono-multiclassification as shown in the Table 4, the cost being obviously that we must train as many classifiers and make as many inferences as classes. Also, the fully connected 3-layer neural network proves to be the best approach with RoBERTa embeddings compared to other classical classifiers, such as Nearest Neighbors (NN), Support Vector Machine (SVM), Gaussian Process (GP), Decision Trees (DT), Random Forest (RF), AdaBoost, Naive Bayes (NB), Logistic Regression (LR) (see Table 5).

When we retrained and tested again the models with no augmentation, the average accuracy dropped by 4% for the one vs. all strategy, thus its importance.

In comparison to [6], the only work using transformers for the task of web classification and based on 5000best dataset [1], our multiclassifiers, based on BERT or TF-IDF respectively, gave slightly better results. We reached an accu-

Label	One vs. all classifiers	Mono-multiclassification model
Articles	79.70%	21.74%
Business	92.20%	65.22%
Commerce	85.60%	60.27%
Education	86.50%	84.62%
Entertainment	95.80%	66.67%
Games	92%	72.06%
Health	93.80%	69.23%
News	90.20%	78.16%
Pictures	92%	69.70%
Porn	100%	85.45%
Services	83.30%	62.96%
Sport	96.90%	87.50%
Style	97.90%	65.62%
Technology	91.70%	70.37%
Tools	86.70%	51.11%
Web	86.70%	62.31%
Avg	90.69%	67.06%

Table 4. Classification accuracy of each class, using the classifier trained on the mono-multiclassification task, and the binary classifiers trained with the one vs. all strategy.

3 layers FCN	NN	SVM	GP	DT	RF	AdaBoost	NB	LR
90.70%	75.58%	84.91%	87.03%	69.84%	70.38%	80.07%	79.81%	82.98%

Table 5. Accuracies obtained from RoBERTa embeddings as input to different classifiers, among others: a 3 layers fully connected network.

racy of 68% and 68.2% respectively, and the authors of [6] reached an accuracy of 67.81%.

6 Conclusion

Websites' classification is a complex task that can be tackled in several ways by using the text of the web page. In this work, we showed the interest of using RoBERTa transformer, a variant of BERT, to represent web pages in a first step and then exploit them to make the classification. We proceeded to a mono-multiclassification in 16 different classes, and also to binary classifications with the strategy one vs. all. Pre-trained and non-finetuned RoBERTa embeddings provided better results compared to classical TF-IDF features. Moreover, we showed that using a 3-layer fully connected neural network as a classifier gives better results than classical machine learning classifiers, such as SVM or Logistic Regression. It is preferable to opt for binary classifiers with the one vs. all strategy, since it improves significantly the results compared those of a mono-multiclassification model. However, there is a trade-off, since one has to train many classifiers and make as many inferences as classes. We mention that we

reached a validation accuracy of 100% for the classification of adult content websites when using a binary classifier. Thus, our work can be used for website's adult content filtering.

As a perspective, it is important to build a less noisy database that is not mono-categorized, since a web page can belong to both the class *Articles* and *Sport* for example, and this multi-categorization aspect was absent in our database. It would also be interesting to test other transformers, and to finetune them on a larger database, to have better embeddings, which will surely end up by improving the results.

References

1. 5000 best websites, http://5000best.com/websites/, accessed on 25 April 2022
2. Domain categorization, https://www.roksit.com/domain-kategorizasyon/, accessed on 25 April 2022
3. Abunadi, A., Akanbi, O., Zainal, A.: Feature extraction process: A phishing detection approach. In: 2013 13th International Conference on Intellient Systems Design and Applications. pp. 331–335. IEEE, Salangor, Malaysia (Dec 2013). https://doi.org/10.1109/ISDA.2013.6920759, http://ieeexplore.ieee.org/document/6920759/
4. Baykan, E., Henzinger, M., Marian, L., Weber, I.: Purely URL-based topic classification. In: Proceedings of the 18th international conference on World wide web - WWW '09. p. 1109. ACM Press, Madrid, Spain (2009). https://doi.org/10.1145/1526709.1526880, http://portal.acm.org/citation.cfm?doid=1526709.1526880
5. Buber, E., Diri, B.: Web Page Classification Using RNN. Procedia Computer Science **154**, 62–72 (2019). https://doi.org/10.1016/j.procs.2019.06.011, https://linkinghub.elsevier.com/retrieve/pii/S187705091930780X
6. Demirkıran, F., Çayır, A., Ünal, U., Dağ, H.: Website Category Classification Using Fine-tuned BERT Language Model. In: 2020 5th International Conference on Computer Science and Engineering (UBMK). pp. 333–336 (Sep 2020). https://doi.org/10.1109/UBMK50275.2020.9219384
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)
9. Jain, A.K., Gupta, B.B.: Towards detection of phishing websites on client-side using machine learning based approach. Telecommunication Systems **68**(4), 687–700 (Aug 2018). https://doi.org/10.1007/s11235-017-0414-0, https://doi.org/10.1007/s11235-017-0414-0
10. Johnson, C., Khadka, B., Basnet, R.B., Doleck, T.: Towards Detecting and Classifying Malicious URLs Using Deep Learning. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications **11**(4), 31–48 (2020). https://doi.org/10.22667/JOWUA.2020.12.31.031, https://doi.org/10.22667/JOWUA.2020.12.31.031
11. Karthikeyan T., Sekaran, K., Ranjith D., Vinoth Kumar V., Balajee J M: Personalized Content Extraction and Text Classification Using Effective Web Scraping Techniques:. International Journal of Web Portals **11**(2),

41–52 (Jul 2019). https://doi.org/10.4018/IJWP.2019070103, http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJWP.2019070103
12. Le, H., Pham, Q., Sahoo, D., Hoi, S.C.H.: URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection. arXiv:1802.03162 [cs] (Mar 2018), http://arxiv.org/abs/1802.03162, arXiv: 1802.03162
13. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
14. Mamun, M.S.I., Rathore, M.A., Lashkari, A.H., Stakhanova, N., Ghorbani, A.A.: Detecting malicious urls using lexical analysis. In: International Conference on Network and System Security. pp. 467–482. Springer (2016)
15. Patel, A.D., Sharma, Y.K.: Web Page Classification on News Feeds Using Hybrid Technique for Extraction. In: Satapathy, S.C., Joshi, A. (eds.) Information and Communication Technology for Intelligent Systems. pp. 399–405. Smart Innovation, Systems and Technologies, Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-1747-7_38
16. Rajalakshmi, R., Aravindan, C.: Web page classification using n-gram based URL features. In: 2013 Fifth International Conference on Advanced Computing (ICoAC). pp. 15–21. IEEE, Chennai, India (Dec 2013). https://doi.org/10.1109/ICoAC.2013.6921920, http://ieeexplore.ieee.org/document/6921920/
17. Rami, M., Fadi, T., Lee, M.: An assessment of features related to phishing websites using an automated technique. IEEE, Piscataway, NJ (2012)
18. Sahoo, D., Liu, C., Hoi, S.C.H.: Malicious URL Detection using Machine Learning: A Survey. arXiv:1701.07179 [cs] (Aug 2019), http://arxiv.org/abs/1701.07179, arXiv: 1701.07179
19. Shirazi, H., Haefner, K., Ray, I.: Fresh-Phish: A Framework for Auto-Detection of Phishing Websites. In: 2017 IEEE International Conference on Information Reuse and Integration (IRI). pp. 137–143. IEEE, San Diego, CA (Aug 2017). https://doi.org/10.1109/IRI.2017.40, http://ieeexplore.ieee.org/document/8102930/
20. Somesha, M., Pais, A.R., Rao, R.S., Rathour, V.S.: Efficient deep learning techniques for the detection of phishing websites. Sādhanā **45**(1), 165 (Dec 2020). https://doi.org/10.1007/s12046-020-01392-4, https://link.springer.com/10.1007/s12046-020-01392-4
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
22. Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
23. Zuhair, H., Salleh, M., Selamat, A.: HYBRID FEATURES - BASED PREDICTION FOR NOVEL PHISH WEBSITES. Jurnal Teknologi **78**(12-3) (Dec 2016). https://doi.org/10.11113/jt.v78.10026, https://journals.utm.my/jurnalteknologi/article/view/10026