# Investigating the Performance of Deep Learning in Detecting Phishing Website

Noor Dayana binti Zulkifli
Universiti Kuala Lumpur
Malaysian Institute of Information Technology
1016 Jalan Sultan Ismail, 50250 Kuala Lumpur
ndayana.zulkifli@s.unikl.edu.my

Noormadinah binti Allias
Universiti Kuala Lumpur
Malaysian Institute of Information Technology
1016 Jalan Sultan Ismail, 50250 Kuala Lumpur
noormadinah@unikl.edu.my

***Abstract*—This project investigates the performance of deep learning in detecting the phishing websites. Phishing website is one of the security threats in the Internet that can disturb any web services. The phishing websites aim to steal user's information, such as usernames, passwords, credit cards including financial details. It is very difficult to identify the difference between phishing and legitimate website. Therefore, the purpose of this project is to investigate the performance of machine learning in classifying the phishing website by using a deep learning model. The deep learning algorithm to learn the URL features identified in the dataset during the training process. The other purpose is to analyze the performance of the technique and model in term of accuracy, precision and recall. The result was compared between the other two classification models which is Random Forest and Decision Tree. The result of accuracy achieved for deep learning model is higher than the other models. The result from this project shows that the model and technique proposed outperforms in terms of accuracy and low error rate.**

***Keywords—Machine Learning, Deep Learning, Phishing Website, Phishing Detection***

## I. INTRODUCTION

In response to the increase of phishing attack, various detection techniques have been performed to create an anti-phishing solution. The techniques use for detecting and identify phishing attack such as blacklist-based, hybrid and heuristics-based [1]and so forth. All of those techniques use their own method to identify the network attack. The

Blacklist-based technique maintains a uniform resource locator (URL) list of sites that are classified as phishing sites [1]; if a page requested by a user is present in that list, the connection is blocked. This technique is commonly used and has a low false-positive rate; however, its accuracy is determined by the quality of the list that is maintained. Consequently, it has the disadvantage of being unable to detect temporary phishing sites.

A hybrid approach classifier like is a combination of different types of model such as Support Vector Machine (SVM) and Nearest Mean Classification (NMC) [2]. The performance from this model is still not achieve a high accuracy to get the best prediction of URL classes.

In conclusion, although there were many studies done before to provide a solution to detect phishing, the approaches mentioned above share a common disadvantage which is the approaches provide inaccurate results in the final evaluation. The hybrid approach takes a lot of time because this technique has many layers to perform to make the final result. As mentioned, the blacklist technique relies on the blacklist dataset to classified the URLs and this technique must be performing a few times and the blacklist dataset need to be regularly update to avoid inaccurate result.

## II. RELATED WORKS

In this article, the authors [3] performs a classification technique to identify the phishing-sites. This research was divided with two phases. The first phase is to perform a few classification models chosen. Based on all of the model performed, the researcher chooses the 3 top model based on the training accuracy. In second phase, this project combined the 3 model with the weak model. The combination of those model is to create a hybrid model. A dataset used in this project contains of 11055 instances with 30 attributes including IP address, domain link, sub-domain, DNS record Google index and many more. The higher accuracy based on this article is 97.75% for two combination models which is Bayesian Network with IBk and J48 with IBk.

According to author [4], machine learning is a platform that can understand, distribute all the issues such as prediction, find a better solution, adapt a new fast information. This paper experiments a few ways to deploy machine learning in order to enhance heuristic and metaheuristic algorithm. This paper limited to combinatorial optimization problem (COP). It is involved with simpler algorithm to make a simple solution using a less dataset. This framework can be use or suiTable for individual dataset to create a heuristic model. Furthermore, this article provides with an explanation to choose the right architecture and presenting several ideas to give an optimal impact on machine learning approaches.

The previous researcher [5] tell about the detection of phishing website using deep learning framework. The researcher introduced Deep Belief Network (DBN) to detect phishing websites and discuss the detection model and algorithm for DBN. DBN is a deep learning models, each of which is a restricted type of Boltzmann machine that contains a layer of visible units that representing the data. The paper contributes by

train the DBN model to get the appropriate parameters for detection in the small data set. The paper using an IP address as and interactive features to identify the URLs and it detect the flow of IP from ISP (Internet Service Provider). The result achieved based on the IP flows proved that the detection can detect the true positive result in average of 90%.

## III. METHODOLOGY

In this chapter explained about the methodology process used in this project. It is also referred to as a linear-sequential life cycle model. Waterfall model is model explained the phase of project starting process until the end. This type of model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project.

### *A. Waterfall model*


Figure 1 Waterfall model.

The waterfall model in Figure 1 used for this project has five process need to be follow step by step. It will start with the feasibility study for the related research article and paper to come out with the new propose idea. After conclude with the title and idea, the next phase is requirement phase to collect all the requirement need to proceed with project. Then, the process of modelling and build a deep learning framework using the technique proposed begins. After modelling process, the project will continue to the next phase which is testing phase. This testing phase is a process to evaluate the model approached on classification of URLs. The last phase is analysis; an examine process based on the result collected from the project.

### *B. Proposed method*

The proposed method in Figure 2 below used in this study comprised two phases which is training and testing. In the training phase it will generate the website to extract the URLs features. The features identify and extract using tokenization and vectorization process. After the features extract, the dataset will be split into two set which the training and testing dataset. The best training sets is between 70% to 80% to get the best accurate result for a model.

Based on the Figure 2 below, training process will generate model which create and training the model to find the best solution in performing the classification model. This training process will repeat until the model produce get the highest accuracy with lower loss. In testing process, the testing dataset will be used as sample to predict the accuracy result using the model created in training process. Then after testing dataset been classified, the model will be analyzed based on the result of classification report and classification matrix.

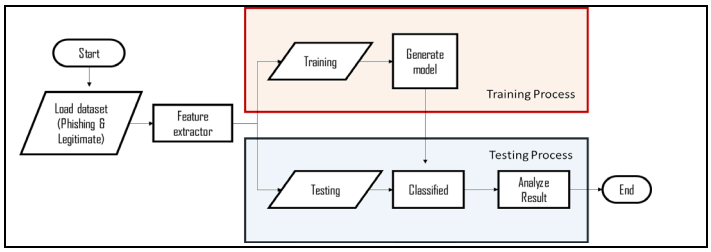


Figure 2 Proposed technique used in this project

### *C. Performance measurement*

One of the purposes of this project is to analyze the performance of the heuristic technique proposed in classification of URLs. This performance measurement will calculate the percentage of prediction during the testing process. This measurement will show the capability of the model in predict the actual value correctly by using the different dataset.

Table 1 Confusion matric

| | | Prediction | |
|---|---|---|---|
| | | Positive | Negative |
| Actual | Positive | True Positive | False Negative |
| | Negative | False Positive | True Negative |

Table 1 shows a confusion matrix used in a measurement calculation of performance for a model. In addition, the prediction will detect the value positive and negative label to be used in performance measurement calculation.

i. True positive = The value of predict result is positive and the actual result also positive.
ii. False positive = The prediction is a positive while the actual result is negative.
iii. True negative = True negative is a value where both of the value from prediction and actual is negative.
iv. False negative = The prediction result is negative and the actual result belongs to positive.

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\textbf{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$

Figure 3 Concept of precision and recall.

Figure 3 is an explanation of terminology used in confusion metric. Confusion matrix used in the calculation to measure the performance of a model in terms of precision, recall and accuracy[6].

a) Precision = Precision term and calculation is a ratio of correct positive prediction to the total.

b) Recall = A calculation of correct positive to the total positive examples.

c) Accuracy = A ratio of correctly predicted example by the total of example

*D. Project Requirement*

a) Computer

The basic requirement for this research is computer as a hardware to download the other software. Computer need 500GB internal hard disk and at least 4GB RAM to support the software and download the dataset. All the dataset and result are kept in this computer to analyzed and evaluate for documentation process.

b) Packages

This research run using python code-language. Python need to be installed in the computer; the script of deep learning project can be run in command prompt. The unnecessary software is Anaconda software, Anaconda is a user-friendly software to run any python script. In Anaconda, it provided with various type of tools which Jupyter Notebook, PyCharm, Spyder, Orange and many more. All the python code in every environment can be used in all tools installed.

c) Dataset

Datasets is the collection data in file have documented by other projects. In this project, the student needs a lot of datasets of URLs to be testes and trained. These datasets are an important element for this project because the machine learning can only be train and learn by this dataset. The various of datasets can be downloaded from the online sources such as Kaggle.com and phishtank sites [7]. These datasets will be set to be train dataset and test dataset. The train dataset will be used in the making of the model framework. While the test dataset was needed in the proses of evaluation to calculate the precision, recall and accuracy.

## IV. RESULT AND DISCUSSION

This chapter presents the findings of this project, which obtain the result from the various analysis. This project is simulated using python code for classification of phishing sites. The main objective is to classified the websites belongs to phishing or legitimate using deep learning framework. The performance will be analysed in terms of precision, recall and accuracy. The result will show the performance of this technique to identify the websites.

A. Deep learning process

The datasets showed in the Figure 4 below is the combination of legitimate and phishing URL data. This dataset is needed in processing the deep learning; the machine learning will be train based on the dataset provided. For machine learning project, the dataset should be as many as possible in order to test the model learnt the pattern.

| | domain | label |
|---|---|---|
| 0 | www.voting-yahoo.com | 1 |
| 1 | www.zvon.org/xxl/WSDL1.1/Output/index.html | 0 |
| 2 | tecportais.com/file-security-update-infonfmati... | 1 |
| 3 | bima.astro.umd.edu/nemo/linuxastro/ | 0 |
| 4 | huarui-tec.com/js/?us.battle.net/login/en/?ref... | 1 |

Figure 4 A column dataset will be used in the model.

All the dataset used will be checked before start the process of learning session. In Figure 4 shows the new dataset which is only the important attribute will be use in the training and testing process. The label attribute is type of the URLs which 0's is phishing URLs and 1's is legitimate URLs that used in performing the training to create a binary classification.

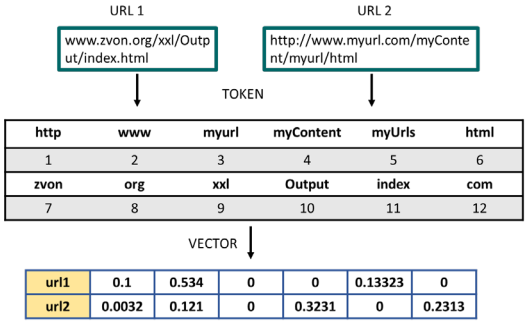


Figure 5 Token and vector process proposed in this project.

The token and vector process will be resulting the value shows in Figure 5. Each URLs in the dataset will be split using the token process above:

a) Token split by slash '/': The token splits every URLs by slash. The slash will determine the URLs domain and subdomain.
b) Token split by dots '.': All the tokens above contain dots will be split again.
c) Token split by dash '-': After the token split by dots, those tokens inspects to find any dash in the URLs.
d) Remove 'com' word. Every 'com' word contained in URLs will be remove as it not important value to classified the result

All the tokens collected in above will be the features or attribute for each URLs. It also can be seen as a filtration process where it can find the same words in all inputs. Then, the vectorization process will turn the all the tokens to an array.

In this project, it used the same dataset file to train, test and predict. The dataset used contain 90000 rows, and it will split by two for the training and testing. 80% for the dataset will be the training process and the others will be the testing process to evaluate the prediction. In order to create the framework, a few elements need to be identified such as the dense, activation type, loss and so forth. The dense is deep neurons in the deep learning model.

The important things in this framework loss element use in it. The loss will declare as binary cross entropy where it will be classified between two type of classification. The binary classification will predict one class and the other will belong to

the other class[8]. For additions, the label attribute which the result of the data is in a binary value, that shows that binary cross entropy will classify the training and testing data into 0 and 1 types. Table 2 below is the detail of the model chose to continue with the phase in this project.

Table 2 Detail of the model proposed

| Model | Description |
|---|---|
| Dataset | 95,000 URLs<br>Training = 76,000<br>Testing = 19,000 |
| Hidden layer | Layer 1 (Dense = 2)<br>Layer2 (Dense = 5) |
| Kernel_initializer | Uniform |
| Activation | Layer 1&2 (relu)<br>Layer 3 (sigmoid) |
| Optimizer | Adam |
| Loss | Binary_crossentropy |
| Metrics | Accuracy |
| Epoch | 10 |

B. Training Result

The training process will shows the result based in every epoch runs in model.fit(). Epoch is a learning cycle used in a deep learning to differ the output in terms of how the modification made in a model to perform better in each cycle. For this proposed model it used 10 epochs to validate the model accuracy.

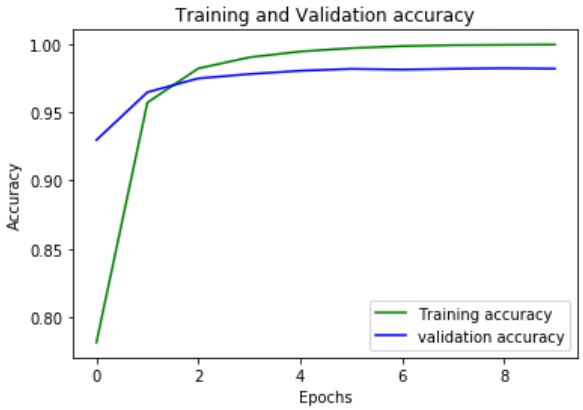


Figure 6 Training and validation accuracy

Figure 6 shows the result of training accuracy for every epoch runs in the model. The graph above shows that the learning process accuracy increase slowly starting from third epoch. The accuracy value for both training and validation from fourth epoch until last epoch is between 0.96 to 1.0. It means that the training process is successful with result almost 100 percent accurate.

Based on the Figure 7 below show both of the training and validation loss decrease from every epoch. From the first and second epoch the loss value dropped from 0.3 to 0.18 for training process. While the validation learning process drop from 0.25 to 0.15 from first to second epoch. Then, the value loss for both training and validation drop slowly between 0.0 to 0.1.

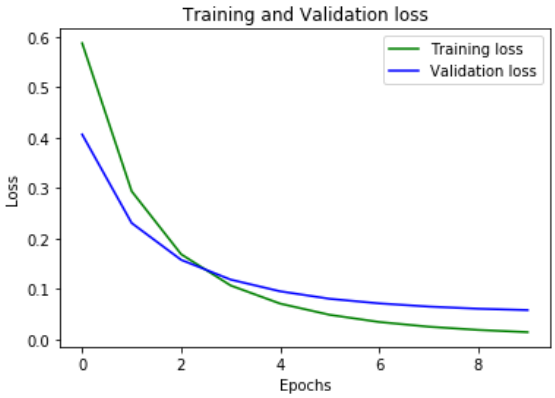


Figure 7 Training and validation loss

This graph shows that the model perform is sTable between the parameter, layer created and neurons set for this learning process. If the loss increase, the model needs to be modified in the number of neurons or feature extractor.

*C. Testing Result*

Based on this measurement the model will identify the true positive, false positive, true negative and false negative. Below is the result of prediction made using this deep learning model.

Table 3 Actual and predict result

| Actual/ Predict | | Predict | |
|---|---|---|---|
| | | Phishing | Legitimate |
| Actual | Phishing | 11017 | 197 |
| | Legitimate | 158 | 7628 |

Table 3 shows the different value from the actual and predict URLs detected using the approach model. The 11,017 is a true prediction for phishing sites and other 158 value is actually the legitimate URLs based on testing dataset. The prediction detected 197 values for legitimate URLs is false and detect 7,628 URLs as the true legitimate URLs. The result in this Table shows that false prediction is low which is 197 for phishing URLs and 158 for legitimate URLs.

Table 4 Classification report

| Class | Precision | Recall | F1-score |
|---|---|---|---|
| 0 | 97% | 98% | 98% |
| 1 | 99% | 98% | 98% |

Classification report shows in Table 4 is used to measure percentage of class detected for testing process. This percentage is measurement for both of class in terms of precision recall and F1-score. F1-measurement is a means of recall and precision

result. The report shows that the model is a good model because of the quality during the learning process. The result from all the perspective is closed to 100%. The lowest value is 97% which to identify the 0's value from total dataset. The highest value is 99% for precision measurement percentage in predict the right 1's value in dataset. The precision and f1-score for both classes is same with value of 98% which means only 2% false to be classified.

### *D. Model Analysis*

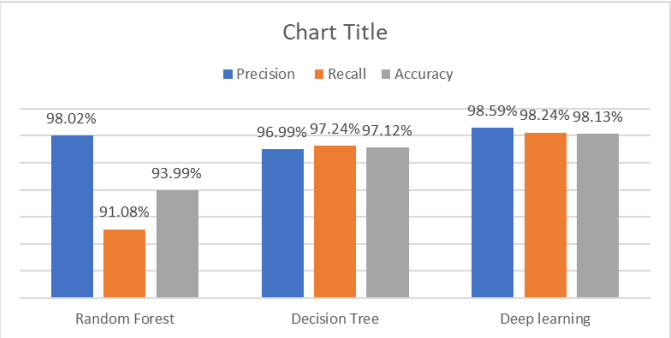


Figure 8 Performance measurement

Based on Figure 8 above is a collected result for three models which is random forest, decision tree and deep learning. The result for random forest is good only at precision term which mean that the model capable on prediction of true legit website because the precision is the percentage of true positive to the actual result. Unfortunately, the recall percentage for this model is low and it effected the value of accuracy; the calculation of true phishing and true legit website from the actual result. Based on the result of this model, it shows that random forest model is not able to identify the correct feature of phishing website.

The decision tree model performance result based on the Table shows that this model is better than random forest. It shows the performance for this in terms of precision which is got 96.99%, 97.24% for recall and the accuracy got 97.12%. The result percentage between each other is almost close, and the total prediction value of true positive and true legit website is higher than the random forest model. This model proves that it performed better on identification on both class of websites.

The deep learning which is the proposed model in this project with the other two models. Based on Figure 8, the deep learning model got 98.59% higher than random forest model for precision measurement. While, the performance to classify in terms of recall measurement is 98.24%. Based on the Table above, the accuracy for deep learning is 98.13% to measure the correct value for phishing and legitimate class. All of the percentages result shows above is the highest compared to the other two models for all precision, recall and accuracy. Based on the result, deep learning gives the best performance on classification process of the websites. The features extraction where the data have been tokenized and vectorized give an important impact on the process of classification to the model learning the different between each URLs given. The other factors to get the best performance for deep learning is the modification of model which the number of neurons, epoch size for the learning process, and the loss result during training process will give an impact on the model to avoid the error on classification. The deep learning proves that learning and training process for the model gives the model a best performance on classification.

## V. CONCLUSION AND RECOMMENDATION

### *A. Conclusion*

The objective of this project is to perform an analysis for the approach model in terms of accuracy, precision and recall. Based on the study of the few machine learning models and techniques have a few limitations where some of the model not suiTable on processing a large value of dataset such as SVM model. Other than that, the blacklist technique used in deep learning model in previous paper proved that this technique cannot identify the new set of websites that is not in the training list. This project come out with heuristic approach using deep learning model. Based on the experiment result, the model success on predictions of the URLs class belongs to the actual class. With the percentage performance higher than 98% it can conclude that the model able to identify the features nicely.

Finally, this project meets both of the objective which is the first one is to develop the heuristic based approach using deep learning framework. The project able to produce the heuristic technique for URLs classification process in the deep learning model. The second objective is to analyze the performance of proposed model in terms of precision, recall and accuracy. The develop techniques and model have been examined and come out with the best result for accuracy, precision and recall. The result of the proposed model also has been compared with other two models to prove the proposed model is better than the other model.

### *B. Recommendation*

In order the pursue the best learning in the deep learning, this project recommend to create a framework for classified the phishing site using the combination of different dataset which is this project only limited for URLs features. Using a different dataset may provide the most accurate result on the prediction due to the multi approach use in the training.

This models also performs using binary cross-entropy which limited only two classes. Future research can focus on categorical cross-entropy to prove shows that learning process work greatly for machine learning. This categorical can be used to identify of gender, picture and many more.

## REFERENCES

[1] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: A machine learning approach," *Stud. Fuzziness Soft Comput.*, vol. 226, pp. 373–383, 2008.

[3] M. A. U. H. Tahir, S. Asghar, A. Zafar, and S. Gillani, "A Hybrid

Model to Detect Phishing-Sites Using Supervised Learning Algorithms," *Proc. - 2016 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2016*, pp. 1126–1133, 2017.

[4] S. Mirshekarian and D. Sormaz, "Machine Learning Approaches to Learning Heuristics for Combinatorial Optimization Problems," *Procedia Manuf.*, vol. 17, pp. 102–109, 2018.

[5] P. Yi, Y. Guan, F. Zou, Y. Yao, W. Wang, and T. Zhu, "Web phishing detection using a deep learning framework," *Wirel. Commun. Mob. Comput.*, 2018.

[6] B. E. Sananse and T. K. Sarode, "Phishing URL Detection: A Machine Learning And Web Mining-Based Approach," *Int. J. Comput. Appl.*, vol. 123, no. 13, pp. 46–50, 2015.

[7] M. Aburrous, M. A. Hossain, F. Thabatah, and K. Dahal, "Intelligent Phishing website detection system using Fuzzy techniques," in *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications, ICTTA*, 2008.

[8] M. Alazab and S. Fellow, "Malicious URL Detection using Deep Learning," pp. 1–9, 2020.