Journal of Theoretical and Applied Information Technology
30th September 2023. Vol.101. No 18
© 2023 Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MACHINE LEARNING MODEL TO IMPROVE CLASSIFICATION PERFORMANCE IN THE PROCESS OF DETECTING PHISHING URLS IN QR CODES

BETTY HERLINA[1], HARYONO SOEPARNO[2]

Computer Science Departement, BINUS Graduate Program - Master of Computer Science,
Bina Nusantara University, Jakarta 11480, Indonesia[1]
Computer Science Departement, BINUS Graduate Program – Doctor of Computer Science,
Bina Nusantara University, Jakarta, 11480, Indonesia[2]
betty.herlina@binus.ac.id[1], haryono@binus.edu[2]

ABSTRACT

This paper discusses the model of detection of phishing URLs in QR codes using Machine Learning techniques. Phishing URLs are URLs that resemble real websites created by cybercriminals to obtain user information for profit. The dataset is collected from various sources, with some important features based on address bar, domain and HTML. The six Machine Learningalgorithms including Decision Trees, RandomForests, SVM, Multilayer Perceptron, Autoencoder Neural Network and XGBoost. The experimental results show that the XGBoostt algorithm provides the highestdetection accuracy with 92% accuracy.

Keywords: *Urls Phishing, Qr Code, Python,Machine Learning, Security Cyber.*

1. INTRODUCTION

QR code technology has developed significantly during the COVID19 pandemic [1], as people are encouraged to avoid physical contact and maintain physical distancing to reduce the spread of the virus. The use of QR codes is not only useful for suppressing the spread of the COVID19 virus, but also to make it easier and save time for users to share information by scanning QR codes. In previous research [2], has proposed Secure QR Code (SQRC) technology in an authentication system through QR code verification using the RSA public key cryptographic algorithm, to increase QR code security. However, this does not rule out the emergence of cyber crimes related to phishing on QR codes. Basically, human failure [3] is the main factor in phishing crimes, as a result, personal information from users is stolen and used by criminals to take advantage. This is not only financially detrimental, but also personal data that is misused has a bad impact.

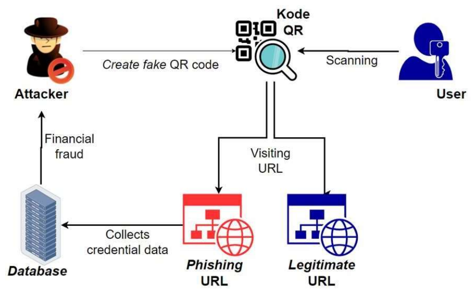


Figure 1. 1 Phishing attack mechanism

Phishing attacks on QR codes are presented in (Figure 1.1), namely, when an attacker creates a fake QR code containing a URL that redirects to a fake website [4]. After that, the QR code is distributed for users to scan it without suspicion. Then the user is asked to log in or enter personal information. The user unknowingly provides this information, believing they have scanned a valid QR code. So that attackers can collect personal information for identity theft and financial crimes [5].

Based on the APWG report [6] for the third quarter of 2022, APWG observed a new record of 1,270,883 phishing attacks (Figure 1.2), this value makes the third quarter (Q3) the worst period in APWG's observations. So to add a layer of security to QR codes, a machine learning method is needed to detect actual phishing URLs [7].

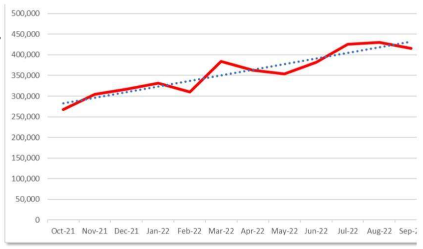


Figure 1. 2 Record of phishing attacks

Basically a QR code is an easy way to share URLs. Based on survey results [10] it was found that when registering on the login page as many as 67% of correspondents registered using their social media account. This indicates that social engineering is a major factor in the success of phishing attacks on QR codes.

Furthermore, this study proposes a phishing detection model on QR codes using machine learning effectively. The dataset used is in the form of a URL. While the classification process in machine learning produces output in the form of dataset classification into three classes, namely URL phishing, URL Legitimate and URL suspicious. Initially, this model compared the performance results of six machine learning algorithms, namely, Extreme Gradient Boosting (XGBoost), Super Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Autoencoder Neural Network (AENN) and Multi Layer Perceptron (MLP). The algorithm is considered good enough to carry out the classification. In the end, the best algorithm will be developed for the detection system so that it can carry out the QR code detection process optimally.

2. LITERATURE RIVIEW

QR code technology is applied to data sharing, payment, information and authentication [8]. From a QR code extracted automatically into more detailed information. On research [9] revealed facts regarding the confidentiality of information when using QR codes. Technically QR codes are protected by encryption algorithms. However, this does not guarantee that the QR code is safe from phishing risks.

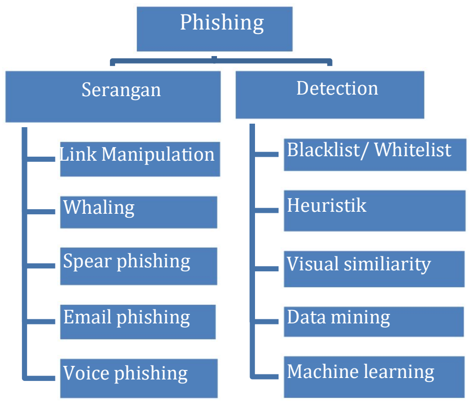


Figure 2. 1 Attack and detection classification diagram

Several surveys have presented various detection techniques, classification algorithms and solutions for detecting phishing attacks (Figure 2.1). However, each approach has its limitations. Phishing attacks are divided into link manipulation, spear phishing, whaling and clone phishing. Meanwhile, phishing detection techniques include blacklist/whitelist, heuristics, visual similarity, machine learning and data mining [11]. Thus, further research will focus on the phishing detection approach to machine learning-based QR codes.

Several machine learning algorithms used in classification techniques are as follows:

2.1 Decision tree (DT)

A decision tree is a model that studies hierarchies in the form of if/else questions that lead to the right results quickly and accurately [12]. The algorithm will search all possible tests to find the target variable.

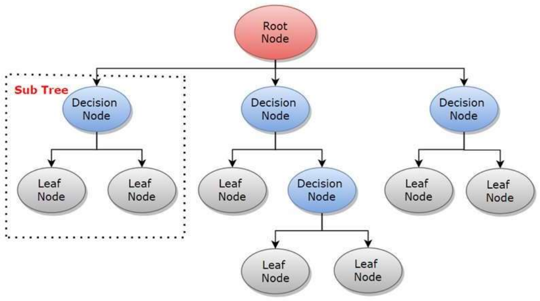


Figure 2. 2 Decision Tree Architecture

Decision trees work very well in classification problems. In this case, the algorithm categorizes the URL dataset into different classes, namely phishing, legitimate, and suspicious. The classification mechanism is determined by several features, enabling the decision tree to make accurate predictions.

$$\text{Gain}(S,A) = \text{Entropy} \sum_{i=1}^{n} * \text{Entropy (Si)} \quad (1)$$

In equation (1) S is a set. While A is an attribute of the set S. The number of partitions on attribute A is defined by n. The tree structure represents a hierarchy that graphically resembles a tree (Figure 2.2). The root node is the highest in the hierarchy and has connected subtrees.

2.2 Random Forest (RF)

A random forest is a collection of several decision trees, then averaging the results of each decision tree. This can reduce the amount of overfitting, to get the best target variable [13].

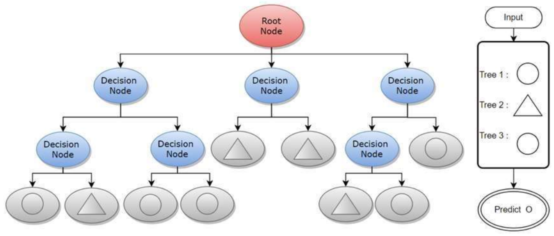


Figure 2. 3 Random forest architecture

The visualization of the random forest algorithm is presented in Figure 2. In this figure, a root node is depicted with multiple decision trees, each producing different outcomes. The results from each tree are averaged to obtain the prediction results generated by the random forest algorithm.

$$l(y) = argmax_c\left(\sum_{n=1}^{N} I_{h_n(y)=c}\right) \quad (2)$$

In equation (2) where I is an indicator function. Meanwhile, h_n is the nth tree generated from the random forest model. The input is processed based on the predicted results of each tree which is calculated by the average value (Figure 2.3).

The advantages of using Random Forest in detecting phishing URLs are being able to handle complex data with many features, work stably and have consistent performance. The algorithm is also reliable in handling data that is not balanced between the number of phishing URLs and legitimate URLs by using techniques such as class balancing or minority class oversampling.

Basically, this algorithm can also overcome dependencies and interactions between features in URL data. In addition, the random forest is able to capture correlations between several features which have quite complex compounding effects. So that it can produce more accurate predictions.

2.3. Super Vector Machine (SVM)

Support Vector Machines is an algorithmic model that analyzes data based on a training set, each data to be classified into a different class. The SVM algorithm is a non-probabilistic binary linear classifier [14]. The main scope of SVM is to find a hyperplane with the largest margin indicator.

$$f(\phi(x)) = sign(w.\phi(x) + b = sign\left(\sum_{i=1}^{N} \alpha_i\, y_i\ \phi(x_i)^T.\phi(x) + b\right) \quad (3)$$

The main process of the SVM method is to determine a hyperplane <w, x> + b = 0 which xj data. Data class yi = {+1, -1}, with maximum margin. Margin is the distance between the hyperplane to each data on a particular label. This hyperplane becomes the decision function f(x) to classify the two labels. Furthermore, w is defined as the weight value and "x" as the input variable. While b is the bias value. Based on (Figure 2.4) it can be visualized that the dataset is classified into two, namely circles and triangles. SVM works to separate the two data points (super vector).

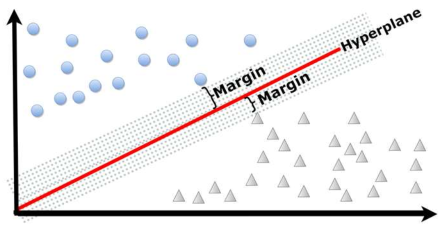


Figure 2. 4 Super Vector Machine visualization

2.4. Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting is one of the boosting techniques. Boosting itself is an in demand method in machine learning that is used to reduce analysis errors and improve the prediction accuracy of the target used [15]. The XGBoost process begins by designing a gradient driven Decision Tree set to increase speed and performance. Each subsequent tree formation depends on the previous tree [16]. The first tree has an indication that the resulting accuracy is not optimal in classifying. So that weight updates are needed in making new trees to optimize prediction accuracy.

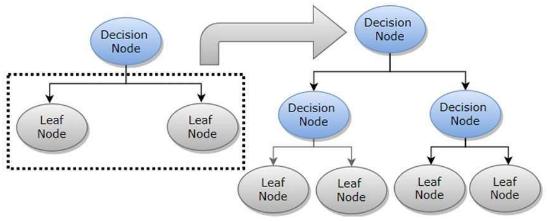


Figure 2. 5 XGBoost illustration

The XGBoost method requires an objective function as a parameter in assessing the quality of the model. The function consists of missing training values (L), regulation values (Ω) and model parameters (θ). Where the value of the data (y_i) is compared with the value of the predicted results ($\hat{y}_i$), while n defines the number of iterations performed.

$$obj(\theta) = L(\theta) + \Omega(\theta) \quad (4)$$

$$L(\theta) = \left(\sum_{i=1}^{N} l(y_i, \breve{y}_i)\right) \quad (5)$$

XGBoost is a utilization of the gradient boosting technique. Gradient boosting itself involves the combination of several simple predictive models such as the Decision tree. After calculating the gradient, the error function is monitored for optimization of the new model. In creating a new model important parameters must be evaluated. So that it can reduce residual errors from previous model iterations.

2.5. Autoencoder Neural Network (AENN)

Autoencoder Neural Network is an unsupervised learning technique that comprises an encoder and a decoder. The encoder is the part that functions to compress the important features of the input into a lower representation in the hidden space by reducing the size of the dimensions (Figure 2.6).

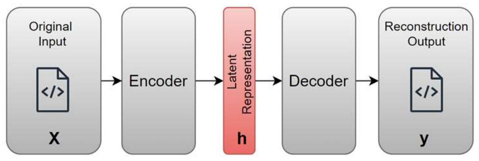


Figure 2. 6 Correlation of Encoder and Decoder

The function used is h=f(x). Meanwhile, the decoder is a part that has the opposite function, namely to reconstruct data in the latent-space representation to the original form of the data at the beginning of the input. The function used is y=g(h).

In general, the AENN structure has the same number of input neurons as its output. But it has less middle layer (hidden layer). So it looks like a bottleneck as shown in (Figure 2.7). The hidden layer of a neural network will have fewer neurons than input/output neurons. So a method is needed to code from the input neurons to a smaller number of hidden neurons, as well as presenting the reconstructed hidden neurons to the output neurons. The predictor (x) and output (y) are exactly the same in an automatic encoder. Based on references from research [17] related to the AENN algorithm, the method applied is gradient-based with a backpropagation process. The backpropagation algorithm in AENN calculates the gradient of the error function from the original data and the resulting data. If the error function is not optimal, then the gradient will adjust the weight and bias values for further prediction calculations.

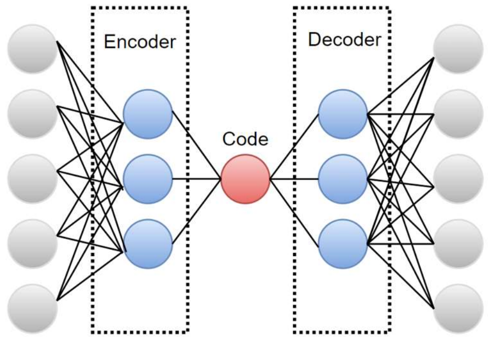


Figure 2. 7 AENN architecture

The backpropagation process in AENN involves several important steps, namely, feed forward, error calculation, gradient calculation and backpropagation. Feed forward is a process starting from data input to going through the encoding and decoding process. Error calculations can use the Mean Square Error (MSE) so you can monitor the difference between the original data and the reconstructed data.

The gradient calculation as a parameter evaluates the difference between the original data and the reconstructed data. Additionally backpropagation counts down the gradient before the target variable through the decoder and encoder layers. So the model can update the bias and weights. This is what underlies the backpropagation algorithm method to be able to work independently by adjusting the weight and bias based on the prediction error.

2.6. Multilayer Perceptron (MLP)

Multilayer perceptron is a type of artificial neural network inspired by human neural networks [18]. Just like neural networks in humans, the MLP Algorithm is composed of several layers of neurons that are connected to each other. Each connected neuron receives data from the previous neuron layer so that it produces output according to the activation function applied. The activation function facilitates the non-linearity of the model to model elusive patterns. So that it can study complex relationships between layers.

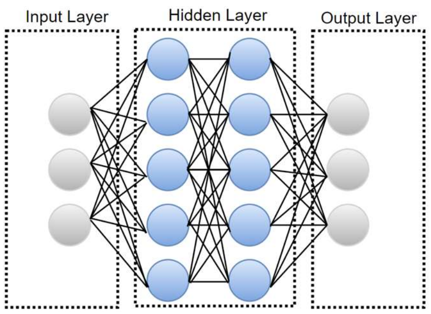


Figure 2. 8 Multilayer Perceptron architecture

The MLP algorithm consists of an input layer, hidden layer and output layer (Figure 2.8). The input layer accepts input data. Then the hidden layer has an important role in complex data transformation. While the output layer as a result of the target variable.

In addition to the machine learning method, another study [19] presents a model using a fuzzy logic algorithm with an accuracy of 82.9%. Another solution to reduce the risk of phishing attacks is to implement a scanner application that is able to read QR codes and redirect URLs through user validation and confirmation [20]. However, this study has limited features in detecting URLs. These features are displaying redirected URLs and sending requests to third parties, namely virus total.

Phishing research using Random Forest and Decision Tree algorithms is further discussed [21] by comparing accuracy. However, this paper has not measured the recall and precision parameters. These parameters were examined by researchers [22] using the ANN, SVM, Decision Tree and Random Forest algorithms.

3. METHODS

This section describes the methodology for detecting phishing attacks on QR codes using machine learning and also describes the proposed framework. Basically the number of datasets, feature extraction and selection of machine learning algorithms play an important role in producing a good model. The model proposed in this experiment is presented in a flowchart (Figure 3.1).

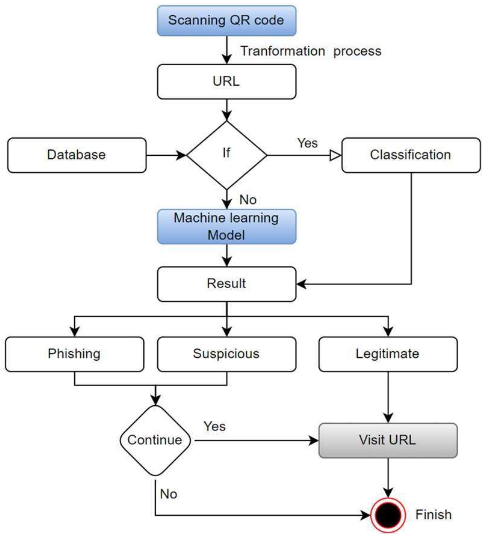
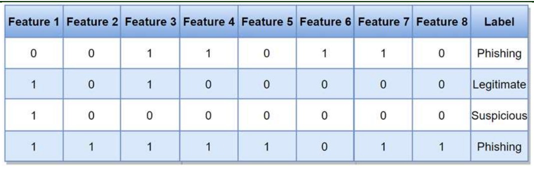


Figure 3. 1 Research design flow chart

In research [23] conducted phishing detection experiments using DNS and IP filtering. But in that paper there is no model with QR code authentication parameters. These parameters were examined by [24]. However, this research has not yet applied the machine learning algorithm method. So in this study it is proposed to detect phishing on QR codes using machine learning methods and are also capable of authenticating.

3.1. Dataset

Relevant datasets will produce good models with optimal target variable results. If the input received is not good, then the results cannot be maximized [25]. The dataset used in this study comes from third party services, namely phistank.com and the University of New Brunswick. The dataset used is as much 50,000 URLs, for more efficient data processing, 6,000 URLs will be taken from all phishing and legitimate URLs randomly.



Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7	Feature 8	Label
0	0	1	1	0	1	1	0	Phishing
1	0	1	0	0	0	0	0	Legitimate
1	0	0	0	0	0	0	0	Suspicious
1	1	1	1	1	0	1	1	Phishing

Figure 3. 2 Phishing datasets

The dataset provided already has the same format, making it easier to integrate the data. Since the dataset is dynamic and is always updated by a third party, the data will be downloaded in real time via the wget (web get) parameter. Then the dataset that has gone through the feature extraction process is converted using the one hot encoding method [26] to csv format (Figure 3.2).

Important features selected from the URL extraction process (Figure 3.3) are IP Address, Status Bar customization, Symbol @, URL Length <54, Domain Urlparse, Depth of URL '/', Redirection '//', http/https, tiny url, prefix suffix, DNS, Age of domain, Website Traffic, End period of domain, Iframe, Disabling right click and Website forwarding. Each of these features has an important role in producing relevant datasets.

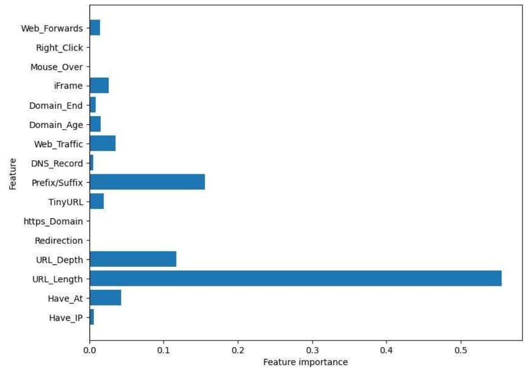


Figure 3. 3 Importance feature

The systematic random sampling method is used in this model to obtain relevant datasets [27]. The datasets are organized and samples are taken at systematic intervals. Furthermore, the data is described based on statistical information in the form of the average value, standard deviation, min, max of each dataframe column (feature). Then validate the value of each attribute that already has a value and no data is empty. Then divide the dataset into two subsets, namely the training set and the testing set. The dataset is separated by a comparison of 80% training data and 20% test data. The selection of the dataset distribution ratio is based on the size of the dataset used in this study and the complexity of the features used. This minimizes the risk of bias that will occur [28].

3.2. Machine Learning Model

Based on previous research [29] compared three methods, consisting of Support Vector Machine (SVM), Naïve Bayes (NB), and Decision Tree (DT) using public datasets published by the UCI Machine Learning Repository (www.uci.edu). This research can optimize the model by selecting the best features, but the process still uses the RapidMiner program. In this study it is proposed to use the Decision tree (DT), Random forest (RF), Super Vector Machine (SVM), Extreme Gradient Boosting (XGBoost), AENN and Multilayer Perceptron (MLP) methods. This data set is classified into two classes namely, phishing (1) or legitimate (0). The model architecture is presented in (Figure 3.4).

The Multilayer Perceptron (MLP) algorithm used in this study uses the python library. The MLP Classifier class is imported from the Neural Network module in the scikit-learn library (sklearn). The regulatory parameter for this model is alpha, which makes it easier for the model to control overfitting potential. While the parameter of the number of hidden layers is 200 neurons.

The Decision Tree (DT) algorithm used in this study uses the python library. The Decision Tree Classifier class is imported from the Tree module in the scikit-learn (sklearn) library. The parameters for this model have a maximum depth of 20.

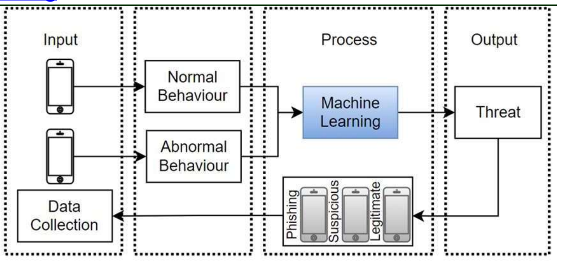


Figure 3. 4 The Proposed Model Architecture

The Extreme Gradient Boosting (XGBoost) algorithm used in this study uses the python library. The XGB Classifier class is imported from the XGBoost library. The learning rate parameter is 0.4 with a maximum depth of 7 per tree.

The Random Forest (RF) algorithm used in this study uses the python library. The Random Forest Classifier class is imported from the Ensemble module in the scikit-learn (sklearn) library. The parameters for this model have a maximum depth of 25.

The Super Vector Machine (SVM) algorithm used in this study uses the python library. The Super Vector Classifier (SVC) class is imported from the SVM module in the scikit-learn (sklearn) library. The type of function from the kernel used is poly (polynomial). The kernel was chosen because it can model a non-linear relationship between features and labels. Furthermore, parameter C to control the trade-off between misclassification and the separation margin is given a value of 1.0. the value is quite moderate to the error of classification. So it is expected to maintain good accuracy in the training data and maintain a fairly wide margin.

The Autoencoder Neural Network (AENN) algorithm used in this study uses the KERAS and Tensorflow libraries in python. The KERAS library is a more intuitive framework for building neural network models [30]. Meanwhile Tensorflow is a powerful platform in numerical computing [31]. The parameters in the AENN model are the dimensions of the input (encoding), input layer, encode layer, code layer, decoder layer and output layer.

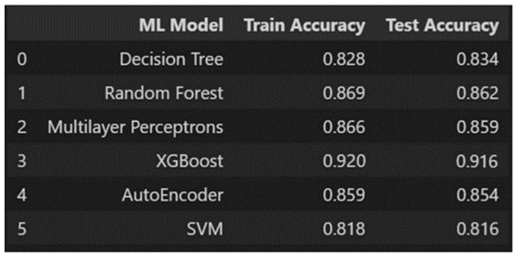
In the context of neural networks,

especially AENN has a fully connected (dense) layer. Dense serves to connect between neurons with layers before (input) and after (output). Each neuron has a weight and bias that is always changing to produce an optimal model. The weight value on the neuron will be multiplied by the input value. While the bias value is added before entering the activation function. The activation function itself used in this model is ReLU (Rectified Linear Unit), because it can support the non-linearity of the output of each neuron. At the input layer, the model can receive data batches with an unspecified number of samples and each sample has an attribute for each feature. The number of epochs used in this AENN model is 10 times. So that the entire dataset will go through the training process 10 times. Each iteration process will be adjusted based on the calculated loss value. While the number of batches for one iteration is 64 data.

3.3. Evaluation

The method for detecting phishing URLs in QR codes is formulated as a type classification in machine learning. The main input is to extract the URL stored in the QR code. To determine the best classification algorithm, a series of experiments were carried out on six machine learning algorithms, namely, SVM, DT, RF, AES, MLP and XGBOOST. All of these machine learning algorithms are trained using the dataset described in the previous section (III.A). The results of the six algorithms are presented in Table 4.1. In the table it can be seen that the greatest accuracy was obtained by the XGBoost method of 92%.

Table 4.1 Performance Comparison



	ML Model	Train Accuracy	Test Accuracy
0	Decision Tree	0.828	0.834
1	Random Forest	0.869	0.862
2	Multilayer Perceptrons	0.866	0.859
3	XGBoost	0.920	0.916
4	AutoEncoder	0.859	0.854
5	SVM	0.818	0.816

Based on the experimental results in subchapter III.B, it is determined that the algorithm used is XGBoost. So that the application will use the XGBoost algorithm which is embedded into the QR code scanning application model according to the proposed research design in Figure 3.1

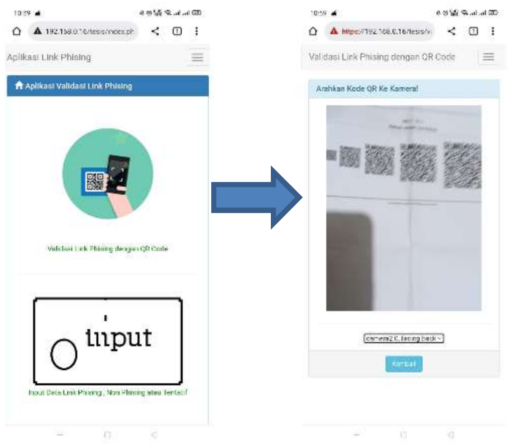
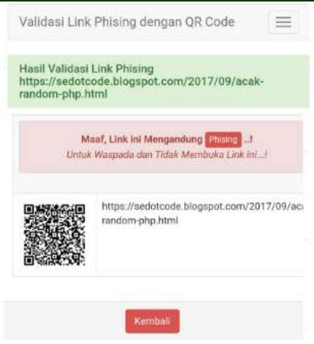

Aplikasi Link Phising
Aplikasi Validasi Link Phising
Validasi Link Phising dengan QR Code
input
Validasi Link Phising dengan QR Code
Arahkan Kode QR Ke Kamera!
Kembali

Figure 4. 1 Visualization Of The Detection System.

In that model a QR code is scanned, then the URL is extracted. Before entering the machine learning process, URL data will be synchronized against a dynamic database associated with the model. If the dynamic database does not find the URL data entered, then the URL data will be processed in the machine learning algorithm. The result of this process produces a target variable, which is a malicious phishing URL or a legitimate URL that is safe to go to. In addition to these two conditions, a "suspicious" condition will appear where this occurs if the dynamic database does not have URL data and the results of the machine learning algorithm are not optimal or doubtful. Application Architecture model is presented in Figure 4.1 and Figure 4.2.


Validasi Link Phising dengan QR Code

Hasil Validasi Link Phising
https://sedotcode.blogspot.com/2017/09/acak-random-php.html

Maaf, Link ini Mengandung Phising ..!
Untuk Waspada dan Tidak Membuka Link ini...!

https://sedotcode.blogspot.com/2017/09/ac
random-php.html

Figure 4. 2Warning Notification Contains Phishing

4. RESULTS AND DISCUSSION

As stated in the previous section, namely research methodology, there is actually a URL dataset as the main research material. The dataset is a dataset of legitimate URLs and phishing URLs. Each of these datasets goes through data preparation, starting from identifying data formats, data cleansing and random data collection. The dataset used in this research is 6,000 legitimate URLs and 6,000 phishing URLs. The dataset is retrieved in real time using the wget parameter (Web Get) because the dataset is always being updated.

The scenarios proposed in this study include selecting a classification method using a machine learning algorithm to a phishing detection system model on QR codes. The accuracy results in table 4.1 show that the best performance is owned by the XGBoost algorithm. Similar to the XGBoost algorithm, the Random Forest, Decision Tree, Super Vector Machine, Multi Layer Perceptron and Auto Encoder Neural Network algorithms have fairly good accuracy. In the testing process, the method with the best accuracy can be produced, namely the XGBoost method of 92%.

The results of the evaluation found that the accuracy of the XGBoost algorithm was 92%, Random Forest algorithm was 86.9%, Decision Tree algorithm was 82.8%, Super Vector Machine algorithm was 81.8%, Multi Layer Perceptron algorithm was 86.6 % and Auto Encoder Neural Network was 85.9%.

Therefore, when looking at the overall results obtained, it can be seen that there are advantages possessed by the XGBoost algorithm, namely producing more accurate predictions and high computational speed. Apart from that, another advantage that can be shown using the XGBoost algorithm is the ability of ensemble learning. So it can be combined with several other models.

5. Conclusion

From various tests carried out on six machine learning algorithms, the XGBoost algorithm produces better performance compared to the use of other algorithms such as the Random Forest algorithm, Decision Tree, Super Vector Machine, Multi Layer Perceptron and Auto Encoder Neural Network which are processed for classification techniques. The accuracy value generated by the XGBoost algorithm is 86.8%. Because of these results, the use of the XGBoost algorithm which utilizes supervised learning techniques as a classification model has the potential to be used as a solution for developing a classification model for phishing URL detection systems in QR codes.

For further research, it is necessary to experiment with the hybrid model method to maximize the performance of the classification model, especially with regard to the development of phishing detection applications on QR codes.

REFERENCES

[1] T. Wang and F. Jia, "The impact of healthQR code system on older people in China during the COVID-19 outbreak," *Age Ageing*, vol. 50, no. 1, pp. 55–56, 2021.

[2] Md. S. Ahamed and H. Asiful Mustafa, "A Secure QR Code System for Sharing Personal Confidential Information," in *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, IEEE, Jul. 2019, pp. 1–4. doi: 10.1109/IC4ME247184.2019.9036521.

[3] G. Desolda, L. S. Ferro, A. Marrella, T. Catarci, and M. F. Costabile, "Human Factors in Phishing Attacks: A Systematic Literature Review," *ACM Comput Surv*, vol. 54, no. 8, pp. 1–35, Nov. 2022, doi: 10.1145/3469886.

[4] R. Heartfield and G. Loukas, "Protection Against Semantic Social Engineering Attacks," in *Advances in Information Security*, 2018, pp. 99–140. doi: 10.1007/978-3-319-97643-3_4.

[5] K. Abd. Latif, B. Sugiantoro, and Y. Prayudi, "Anti-Qrishing Real-Time Technique on the QR Code Using the

Address Bar-Based and Domain-Based Approach on Smartphone," *International Journal of Cyber-Security and Digital Forensics*, vol. 8, p. 134+, 2019, [Online]. Available: https://link.gale.com/apps/doc/A609412267 /AONE?u=anon~85795513&sid=googleScholar&xid=c29f6e80

[6] "APWG Phishing Report." [Online]. Available: http://www.apwg.org,

[7] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, "A predictive model for phishing detection," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 2, pp. 232–247, Feb. 2022, doi: 10.1016/j.jksuci.2019.12.005.

[8] Z. Deineko, S. Sotnik, and V. Lyashenko, "Usage and Application Prospects QR Codes," 2022.

[9] Z. Deineko, S. Sotnik, and V. Lyashenko, "Confidentiality of Information when Using QR-Coding," *International Journal of Academic Information Systems Research(IJAISR)*, 2022.

[10] F. Sharevski, A. Devine, E. Pieroni, and P. Jachim, "Gone Quishing: A Field Study of Phishing with Malicious QR Codes." 2022.

[11] Merlin. V. Kunju, E. Dainel, H. C. Anthony, and S. Bhelwa, "Evaluation of Phishing Techniques Based on Machine Learning," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, IEEE, May 2019, pp. 963–968. doi: 10.1109/ICCS45141.2019.9065639.

[12] B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[13] R. Genuer, J.-M. Poggi, R. Genuer, and J.-M. Poggi, *Random forests*. Springer, 2020.

[14] S. V. M. Vishwanathan and M. Narasimha Murty, "SSVM: a simple SVM algorithm," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, IEEE, pp. 2393–2398. doi: 10.1109/IJCNN.2002.1007516.

[15] M. Praveena and V. Jaiganesh, "A literature review on supervised machine learning algorithms and boosting process," *Int J Comput Appl*, vol. 169, no. 8, pp. 32–35, 2017.

[16] T. Chen *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.

[17] V. Sagar and K. Kumar, "Autoencoder Artificial Neural Network Public Key Cryptography in Unsecure Public channel Communication," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 11, pp. 4023–4032, Sep. 2019, doi: 10.35940/ijitee.K1456.0981119.

[18] H. Ramchoun, Y. Ghanou, M. Ettaouil, and M. A. Janati Idrissi, "Multilayer perceptron: Architecture optimization andtraining," 2016.

[19] H. A. M. Wahsheh and M. S. Al-Zahrani, "Secure Real-Time Computational Intelligence System Against Malicious QR Code Links," *INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL*, vol. 16, no. 3, May 2021, doi: 10.15837/ijccc.2021.3.4186.

[20] P. Hemavathi, N. Hegde, R. Bharti, R. Sur, and S. Priyanka, "Anti-Malware Phishing QR Scanner," *Int J Innov Sci Res Technol*,vol. 3, no. 5, 2018.

[21] S. Khomane, S. Bhosale, D. Tanpure, and S. B. Karpe, "PHISHING WEBSITE DETECTION USING MACHINE LEARNING".

[22] S. Alnemari and M. Alshammari, "Detecting Phishing Domains Using Machine Learning," *Applied Sciences*, vol.13, no. 8, p. 4649, Apr. 2023, doi: 10.3390/app13084649.

[23] M. S. Islam, M. Sajjad, M. M. Hasan, and M. S. I. Mazumder, "Phishing Attack Detecting System Using DNS and IPFiltering".

[24] S. Ismail, M. H. Alkawaz, and A. E. Kumar, "Quick Response Code Validation and Phishing Detection Tool," in *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, IEEE, Apr. 2021, pp. 261–266. doi: 10.1109/ISCAIE51753.2021.9431807.

[25] A. Paullada, I. D. Raji, E. M. Bender, E. Denton, and A. Hanna, "Data and its (dis)contents: A survey of dataset

development and use in machine learning research," *Patterns*, vol. 2, no. 11, p. 100336, Nov. 2021, doi: 10.1016/j.patter.2021.100336.

[26] S. Bagui, D. Nandi, S. Bagui, and R. J. White, "Machine Learning and Deep Learning for Phishing Email Classification using One-Hot Encoding," *Journal of Computer Science*, vol. 17, no. 7, pp. 610–623, Jul. 2021, doi:10.3844/jcssp.2021.610.623.

[27] I. Etikan, "Sampling and Sampling Methods," *Biom Biostat Int J*, vol. 5, no. 6, May 2017, doi: 10.15406/bbij.2017.05.00149.

[28] V. R. Joseph and A. Vakayil, "SPlit: An Optimal Method for Data Splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, Apr. 2022, doi: 10.1080/00401706.2021.1921037.

[29] Z. Halim, " Prediksi Website Pemancing Informasi Penting Phising Menggunakan Support Vector Machine (SVM)," *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS : Journal of Information System*, vol. 2, no.1, pp. 71 – 82, 2017, [Online]. Available: https://ejournal-binainsani.ac.id/index.php/ISBI/article/view/673

[30] J. Moolayil, "An Introduction to Deep Learning and Keras," in *Learn Keras for Deep Neural Networks*, Berkeley, CA: Apress, 2019, pp. 1–16. doi: 10.1007/978-1-4842-4240-7_1.

[31] B. Pang, E. Nijkamp, and Y. N. Wu, "Deep Learning With TensorFlow: A Review," *Journal of Educational and Behavioral Statistics*, vol. 45, no. 2, pp. 227–248, Apr. 2020, doi: 10.3102/1076998619872761.

[32] G. Harinahalli Lokesh and G. BoreGowda, "Phishing website detection based on effective machine learning approach," *Journal of Cyber Security Technology*, vol.5, no. 1, pp. 1–14, Jan. 2021, doi: 10.1080/23742917.2020.1813396.

[33] S. Jain, "Phishing Websites Detection Using Machine Learning," *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4121102.