Optimizing Deep Neural Networks Using ANOVA for Web Phishing Detection

Wulan Sri Lestari[1*], Mustika Ulina[2]

[1,2]Information Technology Department, Mikroskil University, Medan, Sumatera Utara, Indonesia
Email: [1*]wulan.lestari@mikroskil.ac.id, [2]mustika.ulina@mikroskil.ac.id

(Received: 4 Jan 2024, revised: 25 Jan 2024, accepted: 26 Jan 2024)

Abstract

Phishing attacks are crimes committed by sending spoofed Web URLs that appear to come from a legitimate organization in order to obtain another party's sensitive information, such as usernames, passwords, and other confidential data. The stolen information is then used to commit fraud, such as identity theft and financial fraud, and can cause reputational damage to the party that is the victim of the phishing attack. This can cause great harm to the victimized individual or organization. To overcome these problems, this research uses feature selection using ANOVA and Deep Neural Networks (DNN) to detect web phishing attacks. Feature selection is used to optimize the performance of the DNN model to achieve more accurate results. Based on the results of feature selection using ANOVA, there are 52 attributes that have a significant impact on web phishing attack detection. The next step is to implement DNN to build a web phishing attack detection model. The results of testing the web phishing detection model show that in the training phase, the accuracy value increased by 17.51% for the 80:20 dataset and 18.39% for the 70:30 dataset. During the testing phase, the accuracy value increased by 17.8% for the 80:20 dataset and 18.58% for the 70:30 dataset. The resulting recognition model shows consistent and reliable results not only during training, but also during testing in situations closer to real-world conditions. Conclusively, the use of ANOVA proves effective in mitigating less relevant features and contributing to the optimization of web phishing detection models.

Keywords: Web Phishing Detection, ANOVA, Deep Neural Networks, Feature Selection, Optimizing.

I. INTRODUCTION

Phishing attacks are malicious cybercrimes that use social engineering to steal users' personal information by sending them fake links [1], [2]. Attackers trick users into providing their confidential information, such as usernames, passwords, credit card numbers, and other personal data, resulting in data breaches [3], [4]. The stolen data is used to commit fraud, such as identity and financial fraud, and can cause reputational damage to the attacked user. There are two types of phishing attacks: active and passive. While passive attacks are concealed and extremely challenging to detect, active attacks are readily identifiable [5]. Phishing attacks aim to inject malware into the user's personal system and obtain sensitive data from the system without the user's knowledge. The malware can be a virus, Trojan horse, worm, or spyware.

Phishing attacks are extremely harmful since they steal users' sensitive information from e-commerce sites, social networking sites, and the financial sector, among other places [6]. They are also growing over time. According to the Anti-Phishing Working Group's (APWG) most recent study covering the previous five years, the number of phishing assaults is rising annually, especially web phishing attacks [7], [8], [9], [10], [11], as shown in Figure 1.

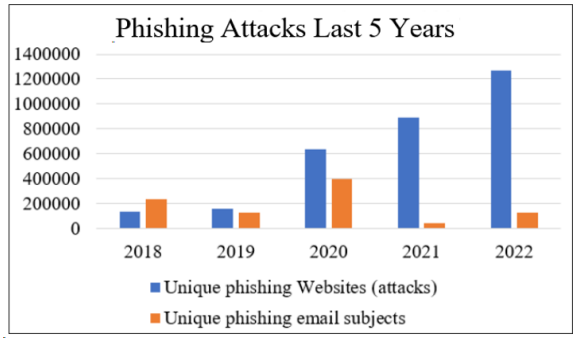

Figure 1. Phishing Attack Reports for the Last 5 Years

In Indonesia alone, there have been 42,442 phishing attacks in the last 5 years, with the number of web phishing attacks being higher than email phishing, with the most targeted industry being financial institutions [12]. In today's Internet era, most financial transactions are conducted online, so it is very important to protect users from phishing attacks that can

cause huge losses. Phishing detection is generally a classification problem that can be solved using machine learning or deep learning algorithms [13]. However, there are several key aspects that need to be considered when using such algorithms, such as the selection of an efficient classification type, the use of features/attributes in model building, and the collection of representative sample datasets for the training process [14]. Machine learning systems designed to detect phishing can be divided into two main categories: systems that actively or passively examine the content of visited web pages, and systems that only examine the URLs of visited web pages.

Jain et.al [15] used a Support Vector Machine (SVM) trained on URLs-based and third-party feature sets to detect phishing websites. The test results of Jain et.al. achieve an accuracy value of more than 90% for the detection of phishing websites. In 2020, Zaini et. al [16] used Random Forest (RF), J48, Multi-Layer Perceptron (MLP) and K-Nearest Neighbors (K-NN) algorithms by selecting 15 features/attributes in the training process. The results of Zaini et. al's research showed that Random Forest has an accuracy value of 94.79% and is better than the other three algorithms. Moorthya & Pabithab [1] used Since Cosine Algorithm (SCA) with K-Nearest Neighbor (K-NN) to optimize the detection of phishing attacks with 30 features/attributes. The test results show that SCA and K-NN achieve an accuracy value of 97.18%. However, Moorthya & Pabithab's research did not perform the most relevant feature selection process to eliminate overfitting/underfitting in the resulting detection model.

In this research, Deep Neural Networks (DNN) are used to detect web phishing attacks. DNN is one of the methods that can be used to solve classification problems. DNN is widely used in various applications such as image classification, object detection, semantic segmentation, face recognition, and other areas [17]. DNN combines the advantages of deep learning and neural networks to solve nonlinear problems better than traditional machine learning algorithms [18]. Faisal & Subekti [19] used DNN for stroke prediction and test results with an accuracy value of 96%. In 2019, Feng at.al [20] used DNN for material defect prediction with an accuracy value of 93%. DNN has the ability to learn patterns and trends in data, make more accurate predictions, and improve the efficiency of attack detection, as well as the ability to process data on a large scale and make real-time detection. However, the use of DNN can also face the problem of overfitting (where the model overfits the training data and loses the ability to generalize to new data) or underfitting (where the model is too simple to understand the complexity of the data).

To solved the problem of overfitting or underfitting in DNN and limitations of Moorthya & Pabithab's research [1], this research used Analysis of Variance (ANOVA). ANOVA is a method used to assess the statistical significance of a set of independent variables in predicting a dependent variable [21]. ANOVA allows the selection of the most significant features in distinguishing between phishing and non-phishing classes. This helps to reduce the data dimensions and improve model performance. The main objective of this research is to test whether ANOVA feature selection can optimize the performance of DNN in detecting phishing attacks using a web page phishing detection dataset, ensuring that the resulting model is not overfitting/underfitting and achieves optimal accuracy.

II. RESEARCH METHODOLOGY

The research method carried out consists of the stages as shown in Figure 2.

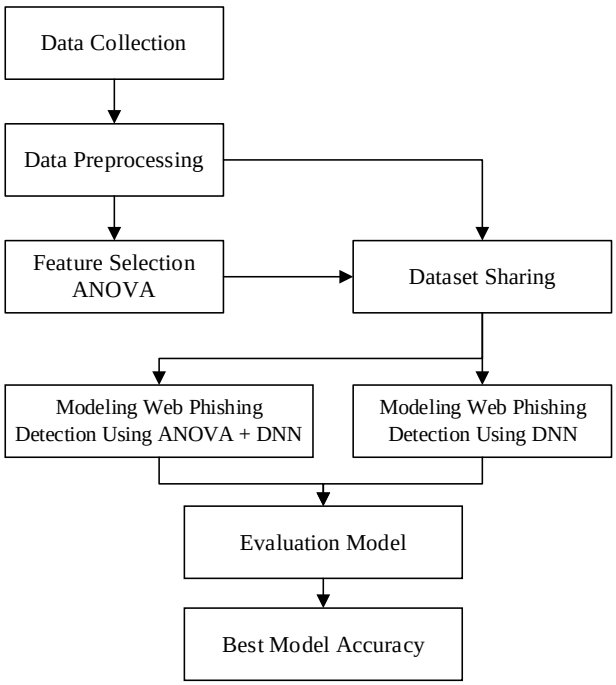

Figure 2. Research Methodology

A. Data Collection

At this stage, the data collection process is carried out, which will be used for the process of creating a web phishing detection model. In this research, secondary data is used, which is taken from the Kaggle website https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset with the title web page phishing detection dataset.

B. Data Preprocessing

Data preprocessing is the process of checking missing values and normalizing the data set. Missing value checking is done to ensure that the dataset being used has missing values or not. This allows the model building process to be optimized. The StandartScaler function is used to adjust the feature/attribute value distribution to have a mean of zero and a standard deviation of one in order to normalize the data collection. Normalization is important to help the algorithm work more efficiently and accurately, especially when the features/attributes in the dataset have different scales. The calculation of the StandartScaler can be done with the following formula 1.

$$x' = (x - \mu) / \sigma \quad (1)$$

where:
x' is the normalized value.
x is the original value of the variable.
μ is the mean of the variables.
σ is the standard deviation of the variable.

C. Feature Selection

The feature selection process is performed to select features/attributes from the dataset that have a significant impact on building web phishing detection models. The feature selection process is performed using Analysis of Variance (ANOVA). This is because the input data used is numerical and the target variable is categorical according to the dataset used.

D. Dataset Sharing

At this stage, the dataset distribution process is randomized. There are two experimental scenarios 80% training data and 20% testing data and 70% training data and 30% testing data.

E. Modeling and Evalution

At this stage, a web phishing detection model is built using Deep Neural Networks (DNN). The training data used in modeling is the data from ANOVA feature selection as much as 9,144 data. Determination of DNN parameters is based on experiments and analysis results during web phishing detection modeling. The following are the parameters used to build a web phishing detection model.

a. The input layer is 52 variables according to the number of ANOVA feature selection results with 128 neurons
b. Hidden layer of 3 layers with 128 neurons
c. Output layer with 1 neuron
d. Dropout = 0.02 and L2 regularization = 0.01 to prevent overfitting/underfitting of the model
e. The Relu activation function is used in the input and hidden layers. The sigmoid activation function is used in the output layer. The above two functions are used because this model is specifically designed for binary classification
f. The optimization algorithm used is Adam, which combines the advantages of AdaGrad and RMSProp algorithms. Adam is effective in dealing with sparse gradients in noisy problems
g. The number of iterations (epochs) used to train the model is 300
h. The confusion matrix used is accuracy, precision, and loss
i. The loss function used is binary_crossentropy. This is consistent with the nature of the output data, which has only values of 0 (legitimate) and 1 (phishing). The sigmoid activation function in the output layer produces values of 0 and 1, so binary_crossentropy is the right choice to measure the accuracy of the model in binary classification

The developed web phishing detection model is then evaluated using the Confusion Matrix to calculate the accuracy, precision and loss levels according to Table 1. The model evaluation process is based on 2,286 test data.

Table 1. Confusion Matrix

Actual Class	Class	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Accuracy and precision can be calculated using the following formula 2-3:

$$\text{Accuracy} = (TP + TN) / (TP+TN+FP+FN) \quad (2)$$
$$\text{Precision} = (TP + TN) / (TP + FP) \quad (3)$$

III. RESULT AND DISCUSSION

In this research, data preprocessing, feature selection, and dataset sharing of feature selection results were performed before implementing DNN for modeling. The following are the results obtained.

A. Dataset

The data used are secondary data from the Kaggle website https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset with the title web page phishing detection dataset. The dataset is designed as a benchmark for machine learning based web phishing detection systems. 11,430 records make up this online phishing dataset, which has 89 features/attributes generated from three separate classes: the target page's content provides 24 features/attributes, the URL structure and syntax provides 56 features/attributes, and external service queries provide 7 features/attributes. Table 2 shows the number of records for phishing and legitimate.

Table 2. Number of Dataset Details

Dataset	Legitimate	Phishing
11,430	5,715	5,715

B. Data Preprocessing Result

The first step in data preprocessing is to check for missing values in the data set. Figure 3, is the result of checking for missing values using the isnull() function in Python.

```
data.isnull().sum()

url                0
length_url         0
length_hostname    0
ip                 0
nb_dots            0
                  ..
web_traffic        0
dns_record         0
google_index       0
page_rank          0
status             0
Length: 89, dtype: int64
```

Figure 3. Missing Value Check Process

From Figure 3, we can see that the data set used has no missing values. The second step is to perform the data normalization process using StandartScaller. The normalization process is necessary to ensure that all the data has a mean of 0 and a standard deviation of 1, so that it can produce a better model accuracy. Table 3 is the result of the data normalization using StandartScaller.

Table 3. Data Normalization Result

length_url	length_hostname	ip	...	page_rank
-0.43632748	-0.19396372	-0.42102044	...	0.32097385
0.28706655	0.17720743	2.37518157	...	-0.46740717
1.17322424	2.6826127	2.37518157	...	-1.25578819
...	...	...	...	...
0.79344237	-0.47234208	2.37518157	...	2.68611691
-0.41824263	0.82675695	-0.42102044	...	0.32097385
7.52100688	-0.65792766	2.37518157	...	-1.25578819

C. Feature Selection Result

At this stage, feature selection is performed using Python's built-in ANOVA function, sklearn.feature_selection with the SelectPercentile function. The SelectPercentile function uses the parameters score_func=f_classif, percentile=60. These parameters are determined based on the results of the experiments performed to obtain the best parameters. Figure 4, is the ANOVA feature selection function that has been done.

```
SP = SelectPercentile(score_func=f_classif , percentile=60)

X = SP.fit_transform(X , Y)
```

Figure 4. ANOVA Feature Selection Process

Of the 89 features in the dataset, 52 characteristics/attributes had a significant effect in identifying web phishing assaults with numeric data types, according to the results of ANOVA feature selection. The 52 features/attributes are length_url, length_hostname, ip, nb_dots, nb_hyphens, nb_at, nb_qm, nb_and, nb_eq, nb_slash, nb_colon, nb_semicolumn, nb_www, nb_com, https_token, ratio_digits_url, ratio_digits_host, tld_in_subdomain, abnormal_subdomain, nb_subdomains, prefix_suffix, shortening_service, length_words_raw, shortest_word_host, longest_words_raw, longest_word_host, longest_word_path, avg_words_raw, avg_word_host, avg_word_path, phish_hints, domain_in_brand, suspecious_tld, statistical_report, nb_hyperlinks, ratio_intHyperlinks, ratio_extHyperlinks, nb_extCSS, ratio_extRedirection, external_favicon, links_in_tags, ratio_intMedia, ratio_extMedia, safe_anchor, empty_title, domain_in_title, domain_with_copyright, domain_registration_length, domain_age, dns_record, google_index, page_rank.

D. Dataset Sharing Result

In this research, dataset is split using 80:20 and 70:30 scenarios. Tables 4 and 5 show the details of how the datasets were split.

Table 4. Details of 80:20 Dataset Split

Dataset	Legitimate	Phishing	Total
Training	4,563	4,581	9,144
Testing	1,152	1,134	2,286
Total	5,715	5,715	11,430

Table 5. Details of the 70:30 Dataset Split

Dataset	Legitimate	Phishing	Total
Training	4,019	3,982	8,001
Testing	1,696	1,733	3,429
Total	5,715	5,715	11,430

E. Modeling and Evaluation

The next step is to build and evaluate a web phishing detection model using the following scenarios to achieve the best accuracy results.

a. Modeling Web Phishing Detection Using Deep Neural Networks (DNN)
b. Modeling Web Phishing Detection Using Deep Neural Networks (DNN) with ANOVA Feature Selection

The results of implementing the training phase using the 80:20 dataset are shown in Table 6.

Table 6. Web Phishing Detection Modeling Results 80:20 Training Phase

Methods	Features/ Attributes	Accuracy	Precision
DNN	87	77.78%	71.89%
Anova + DNN	52	95.29%	95.60%

Based on Table 6, the results show that the DNN model optimized with ANOVA has increased accuracy by 17.51%, precision by 23.71%. Figure 5, is a graph of the loss value of making the web phishing detection model in the training stage with the 80:20 dataset.

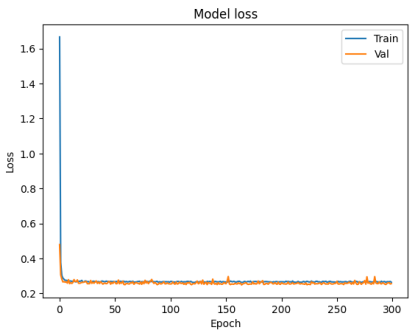

Figure 5. Loss Value of the Web Phishing Detection Model Using Anova and DNN with the 80:20 Dataset

The result of implementing the training phase using the 70:30 dataset is shown in Table 7.

Table 7. Modeling Results of Phishing Web Detection 70:30 Training Stage

Methods	Features/ Attributes	Accuracy	Precision
DNN	87	76.80%	71.18%
Anova + DNN	52	95.19%	94.55%

Based on Table 7, the results show that the DNN model optimized with ANOVA has increased accuracy by 18.39% and precision by 23.37%. Figure 6, is a graph of the loss value of the web phishing detection model in the training stage with a 70:30 dataset.

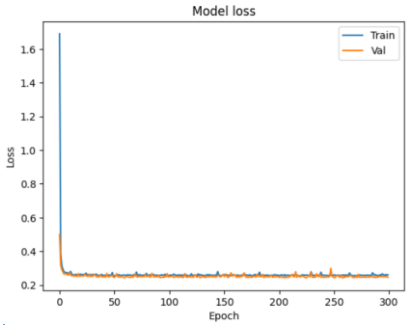

Figure 6. Loss Value of Web Phishing Detection Model Using Anova and DNN with Dataset 70:30

Based on Figure 5 and Figure 6, it can be concluded that the web phishing detection model does not experience underfitting/overfitting for both dataset sharing scenarios. The two web phishing detection models built on the training data are then evaluated on the test data.

Table 8 shows the evaluation result of the web phishing detection model using 80:20 test data.

Table 8. Modeling Results of 80:20 Web Phishing Detection Test Phase

Methods	Features/ Attributes	Accuracy	Precision
DNN	87	77.21%	71.71%
Anova + DNN	52	95.01%	95.05%

A comparison chart of model evaluation results using 80:20 test data is shown in Figure 7.

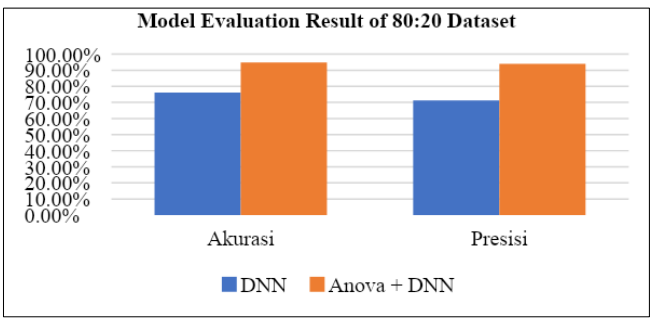

Figure 7. 80:20 Web Phishing Detection Model Dataset Evaluation Results Comparison

Table 8 and Figure 7 show that the evaluation results of the web phishing detection model using DNN optimized by ANOVA have increased the accuracy by 17.8%, the precision by 23.34%. The evaluation results of the Web Phishing detection model using 70:30 test data are shown in Table 9.

Table 9. Modeling Results of Web Phishing Detection Level 70:30 Testing Stage

Methods	Features/ Attributes	Accuracy	Precision
DNN	87	76.17%	71.24%
Anova + DNN	52	94.75%	93.99%

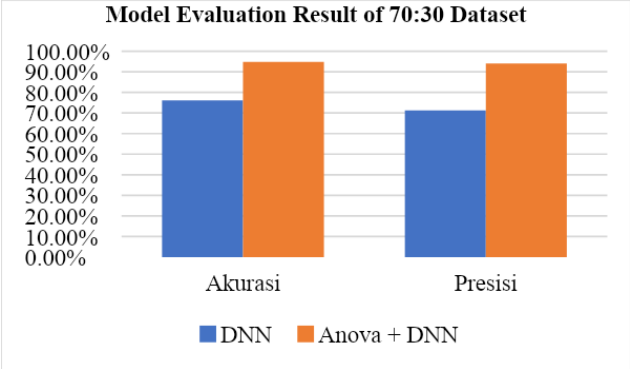

Figure 8. Comparison Chart of Model Evaluation Results using 70:30 Testing Stage

Table 9 and Figure 8, show that the evaluation results of the web phishing detection model using DNN optimized by ANOVA have increased accuracy by 18.58%, precision by 22.75%. Based on the two dataset sharing scenarios performed for the testing process, it can be concluded that the 80:20 and 70:30 dataset sharing can achieve good accuracy for web phishing detection models using ANOVA and DNN.

IV. CONCLUSION

Based on the results of the implementation and tests carried out to detect web phishing using Deep Neural Networks with ANOVA feature selection, it is evident that this approach is able to optimize the performance of Deep Neural Networks with a significant increase in the accuracy and precision values during the training phase. In the training phase using the 80:20 dataset, an increase in accuracy of 17.51% and precision of 23.71% was recorded, and in the 70:30 dataset, an increase in accuracy of 18.39% and precision of 23.27% was recorded. This shows the effectiveness of the ANOVA method in selecting relevant features to improve model performance. Meanwhile, at the testing stage, there was a significant improvement in accuracy of 17.8% and precision of 23.34% for the 80:20 dataset, and accuracy of 18.58% and precision of 22.75% for the 70:30 dataset. This indicates that the developed detection model is capable of providing consistent and reliable results not only in the training phase, but also in the testing process, which is more similar to real-world conditions. The evaluation of the accuracy and precision results concluded that

ANOVA was successful in reducing less relevant features in the formation of web phishing detection models. This success is not only seen from the aspect of improving accuracy and precision, but also from the ability of the model to avoid overfitting and underfitting. The fact that the model does not experience overfitting or underfitting confirms the reliability of the model in generalizing information from training data to test data.

ACKNOWLEDGMENTS

We would like to thank the Universitas Mikroskil for supporting this research.

REFERENSI

[1] R. S. Moorthy, and Dr. P. Pabithab, "Optimal Detection of Phising Attack using SCA based K-NN," Procedia Computer Science, vol. 171, pp. 1716-1725, 2020, doi: 10.1016/j.procs.2020.04.184.

[2] T.O. Ojewumi et al. "Performance evaluation of machine learning tools for detection of phishing attacks on web pages." Scientific African. Vol. 16, 2022, doi: 10.1016/j.sciaf.2022.e01165.

[3] B. A. Tama and K. H. Rhee, "A Comparative Study of Phishing Websites ClassificationBased on Classifier Ensembles," Journal of ultimedia Information System, vol. 5, no. 2, pp. 99-104, 2018, doi: 10.9717/JMIS.2018.5.2.99.

[4] I. Saha, D. Sarma, R. J. Chakma, M. N. Alam, A. Sultana and S. Hossain, "Phishing Attacks Detection using Deep Learning Approach," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020, pp. 1180-1185, doi: 10.1109/ICSSIT48917.2020.9214132.

[5] V. Suganya, "A Review on Phishing Attacks and Various Anti Phishing Techniques," International Journal of Computer Applications vol. 139, pp. 20-23, 2016.

[6] M. Bahaghighat, M. Ghasemi, and F. Ozen, "A high-accuracy phishing website detection method based on machine learning," Journal of Information Security and Applications, Elsevier, vol. 77, 2023, doi: 10.1016/j.jisa.2023.103553.

[7] APWG Report 2018, "Phishing Activity Trends Report," pp. 1-13, 2018.

[8] APWG Report 2019, "Phishing Activity Trends Report," pp. 1-13, 2019.

[9] APWG Report 2020, "Phishing Activity Trends Report," pp. 1-14, 2020.

[10] APWG Report 2021, "Phishing Activity Trends Report," pp. 1-15, 2021.

[11] APWG Report 2022, "Phishing Activity Trends Report," pp. 1-11, 2022.

[12] IDADX Report, "Laporan Aktivitas Phishing Domain.ID," 2022.

[13] S.C. Jeeva and E.B. Rajsingh, "Intelligent Phishing URL Detection Using Association Rule Mining," Human-Centric Computing and Information Sciences, Vol. 6, No. 1, pp. 1-19, 2016.

[14] A. Hannousse and S. Yahiouche, "Towards Benchmark Datasets for Machine Learning based Website Phishing Detection: An Experimental Study," arXiv:2010.12847, 2020, doi: 10.48550/arXiv.2010.12847

[15] A. Jain, and B. Gupta, "Phish-safe: Url features-based phishing detection system using machine learning," Cyber Security, pp. 467–474, Singapore: Springer Singapore, 2018, doi: 10.1007/978-981-10-8536-9_44

[16] N. Zaini, et. al, "Phishing detection system using machine learning classifiers," Indonesian Journal of Electrical Engineering and Computer Science, vol. 17, pp. 1165–1171, 2020.

[17] J. Tan and Y. Tian, "Fuzzy Retrieval Algorithm for Film and Television Animation Resource Database based on Deep Neural Network," Elsevier BV, Journal of Radiation Research and Applied Sciences, 2023, doi: 10.1016/j.jrras.2023.100675.

[18] P. Yu, and X. Yan, "Stock Price Prediction based on Deep Neural Networks," Neural Computing and Applications 32, pp. 1609–1628, 2020.

[19] A. Faisal, and A. Subekti, "Deep Neural Network untuk Prediksi Stroke," Jurnal Edukasi dan Penelitian Informatika, vol. 7, No 3, 2021.

[20] Feng et al., "Using Deep Neural Network with Small Dataset to Predict Material Defects", Materials and Design Elsevier, vol. 162, pp. 300-310, 2019.

[21] A. Wenda, "Support Vector Machine untuk Pengenalan Bentuk Manusia Menggunakan Kumpulan Fitur yang Dioptimalkan," Jurnal Sains & Teknologi, vol. 11, no. 1, 2022, doi: 10.23887/jstundiksha.v11i1.44437.