Perbangingan Metode *Extreme Learning Machine* dan *Backpropagation* untuk Mengklasifikasi *Phising Websites*

Okky Putra Barus[1], Ronaldo[2]

[2]Fakultas Ilmu Komputer, Universitas Pelita Harapan
E-mail: ronaldo_n09@outlook.com[2)]

Abstract – *Phishing website is one types of electronic crimes that has grown rapidly and also one of the most dangerous, which may have tremendous impact on e-banking and various online businesses such as e-commerce and Software-as-a-Service (SaaS). Generally, this type of crime begins with an e-mail that resembles the official one in which contains information notifying the victims of a necessity to update their information and a link to their site. This site is a phishing website that was built by the criminal to gain access to victims' sensitive and personal information. Therefore, it is necessary for there to be a method which can effectively detect a phishing website. This paper presents a methodology for detecting phishing websites based on machine learning classifier using Extreme Learning Machine (ELM) and neural network with backpropagation. Extreme Learning Machine is a learning algorithm for single layer feedforward neural networks (SLFNs) which randomly choose input weights and analytically determines the output weights. Backpropagation is a learning algorithm for neural networks which helps adjust the weights to get the best possible results. The results show that the backpropagation method provides better classification accuracy of 91.85% compare to ELM with the classification accuracy of 84.07%.*

Keywords:*phishing websites, classification, machine learning, extreme learning machine, backpropagation*

Abstrak – *Situs web phishing merupakan salah satu jenis kejahatan elektronik yang terus berkembang secara pesat dan juga salah satu yang paling berbahaya, yang dapat memberikan dampak negatif yang sangat besar terhadap e-banking dan berbagai jenis bisnis online seperti e-commerce dan penyedia Software-as-a-Service (SaaS). Umumnya, jenis kejahatan ini dimulai dari pengiriman e-mail yang menyerupai e-mail yang dikirim oleh instansi resmi yang berisi informasi pemberitahuan kepada korban bahwa perlu dilakukan suatu pembaharuan atau verifikasi data yang disertai dengan link untuk menuju situs tersebut. Situs tersebut merupakan situs web palsu yang dibangun oleh pelaku kejahatan yang dimana menyamari dan menyerupai situs aslinya untuk mendapatkan informasi sensitif dan pribadi korban. Oleh karena itu, diperlukan suatu metode yang efektif untuk mendeteksi apakah suatu website termasuk kategori phishing atau tidak. Penelitian ini menyajikan metodologi pendeteksian phishing websites berbasis machine learning dengan menggunakan dan membandingkan metode Extreme Learning Machine (ELM) dan neural network dengan algoritma backpropagation. Extreme Learning Machine merupakan algoritma pembelajaran untuk feedforward neural networks dengan 1 lapisan tersembuyi yang secara acak menentukan bobot input dan output. Backpropagation merupakan algoritma pembelajaran neural*

networks yang melakukan penyesuaian bobot untuk mendapatkan hasil terbaik. Hasil penelitian menunjukkan metode backpropagation dalam mengklasifikasi phishing websites memberikan ketepatan klasifikasi sebesar 91.85% lebih besar dibanding dengan metode ELM dengan ketepatan klasifikasi 84.07%.

Kata Kunci:*situs web phishing, klasifikasi, pembelajaran mesin, extreme learning machine, backpropagation*

PENDAHULUAN

Pada era informasi teknologi ini yang dimana pertukaran informasi dapat dilakukan dengan cepat dari seluruh belahan dunia dalam hitungan detik. Ini semua karena adanya teknologi internet yang memudahkan individu maupun organisasi dalam melakukan aktivitas pertukaran informasi dan komunikasi jarak jauh. Namun tidak sampai begitu saja kapabilitas dari internet. Dengan adanya media *website* dan *web apps*, pengguna internet dapat melakukan aktivitas lain seperti transaksi jual-beli, transaksi perbankan (*e-banking*), penyimpanan data secara *online* dengan *cloud storage*, dan lain sebagainya.

Dengan adanya teknologi-teknologi tersebut, banyak individu maupun organisasi/instansi memanfaatkan kemudahan yang diberikan untuk mempermudah kehidupan mereka juga meningkatkan efisien dan efektifitas kerja organisasi. Ini dapat ditunjukkan dengan menggunakan data yang diperoleh dari *InternetWorldStats.com (IWS)*, dimana jumlah pengguna internet tercatat Maret 2019 adalah 4,383,810,342 pengguna, yang berarti 56.8% dari jumlah populasi di seluruh dunia menggunakan internet. Data bersumber dari IWS memiliki kelebihan dimana data dikumpulkan mereka dari berbagai perusahaan riset, konsultan, pemerintah, dan universitas di dunia dan sering mengalami pembaharuan.

Tingginya jumlah pengguna internet juga sebanding dengan kuantitas *website* yang dibuat, yang berarti adanya dampak terhadap kualitas website yang beredar di internet. Berikut grafik yang menunjukkan total jumlah situs *phishing* yang terdeteksi oleh *Anti-Phishing Working Group (APWG)* untuk periode ke-4 tahun 2018 sampai dengan periode ke-1 tahun 2019 dengan total angka pada periode ke-1 tahun 2019 yaitu sebanyak 180,768.

Gambar 1. Grafik Jumlah Phishing Sites pada periode ke-4 2018 sampai dengan period ke-1 2019.

Ini dikarenakan adanya para pelaku kejahatan yang menemukan kesempatan untuk memanfaatkan popularitas suatu website untuk melakukan tindakan kejahatan yang dikenal dengan *phishing website*. *Phishing website* merupakan suatu jenis atau metode penipuan yang dimana pelaku kejahatan membuat website yang menyerupai website asli secara visual tanpa persetujuan dari pihak yang bersangkutan untuk menipu pengguna website aslinya, biasanya untuk mencuri data *credential* pengguna yang dapat berisi informasi pribadi pengguna sampai dengan informasi pembayaran yang digunakan pengguna pada website aslinya. Oleh karena itu, perlu dilakukan pendeteksian terhadap *websites* yang berada di internet guna meminimalisir jumlah korban penipuan dan pengguna internet dapat mengakses *website* yang diinginkan dengan aman.

Data mining merupakan satu cara untuk melakukan hal tersebut, dimana pada prosesnya dilakukan analisis sekumpulan data yang sangat banyak (*dataset*) sehingga lebih mudah untuk dimengerti. Salah satu yaitu *machine learning*.

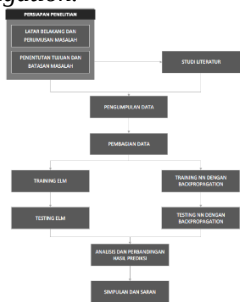
Sebelumnya terdapat penelitian yang dilakukan menggunakan algoritma *SMO (Sequential Minimal Optimization)* dengan tingkat akurasi sebesar 95.70 % untuk melakukan pendeteksian *phishing* terhadap situs-situs yang berfokus pada *e-Business websites* di China untuk memitigasi kerugian finansial yang terjadi (Jiang, 2013). Namun penelitian tersebut bersifat spesifik karena penelitian tersebut menyangkutkan atribut yang khusus terdapat pada *e-Business websites* yang berada di China.

Penelitian ini akan menggunakan metode ELM (*Extreme Learning Machine)* dan *neural network backpropagation* untuk mendeteksi phishing websites dan dibandingkan hasilnya karena metode ini merupakan salah satu yang lazim digunakan dalam kasus pengklasifikasian seperti yang akan dilakukan pada penelitian ini.

METODE PENELITIAN

Desain Penelitian

Langkah-langkah proses yang dilakukan untuk membandingkan model untuk klasifikasi *phishing website* yaitu: pengumpulan data, pembagian data ke dalam *training set* dan *testing set*, dan perbandingan hasil prediksi metode *ELM* dan *Neural Network* dengan *backpropagation.*



Gambar 2. Desain penelitian

Pada tahap awal penelitian, latar belakang ditentukan dan dilanjutkan dengan mendefinisikan rumusan masalah, penentuan tujuan dan batasan masalah yang menjadi bahasan penelitian. Studi literatur bertujuan untuk mendalami pemahaman terhadap konsep, teori, dan cara kerja algoritma *ELM* dan *backpropagation* serta untuk memahami fitur di dalam dataset.

Tahap berikutnya dari penelitian ini yaitu tahap pengumpulan data yang kemudian dinormalisasi dan dibagi ke dalam *training set* dan t*esting set* yang kemudian diimplementasikan ke dalam model dan dibandingkan tingkat akurasi hasil output prediksi dari kedua model tersebut.

HASIL DAN PEMBAHASAN

Hasil Pengumpulan Data

Dengan melakukan pencarian pada situs *UCI Repository*, diperoleh data set *phishing websites* yang bersifat klasifikasi. Data set ini memiliki 1353 sampel data. Data *phishing websites* didapatkan dari arsip data *Phishtank* dan *website* yang legit bersumber dari *Yahoo*. Format *dataset* dapat dilihat pada gambar dibawah.

SFH	popUpWindow	SSLfinal_State	Request_URL	URL_of_Anchor	web_traffic	URL_length	age_of_domain	having_IP_address	Result
1	-1	1	-1	-1	1	1	1	0	0
-1	-1	-1	-1	-1	0	1	1	1	1
1	-1	0	0	-1	0	-1	1	0	1
1	0	1	-1	-1	0	1	1	0	0
-1	-1	1	-1	0	0	-1	1	0	1
-1	-1	1	-1	-1	1	0	-1	0	1
1	-1	0	1	-1	0	0	1	0	-1
1	0	1	1	0	0	0	1	1	-1
-1	-1	0	-1	-1	-1	-1	1	0	0
-1	0	-1	-1	1	1	0	-1	0	1
-1	-1	0	-1	-1	1	-1	-1	0	1
1	0	1	1	1	-1	1	1	0	-1
1	-1	0	-1	1	0	1	1	0	1
1	0	1	0	-1	1	0	1	0	-1
-1	-1	-1	1	-1	0	-1	1	0	1
0	0	-1	0	0	1	1	-1	1	1
-1	0	0	0	-1	1	-1	-1	0	1
1	1	1	-1	1	-1	1	1	0	-1
0	-1	-1	0	1	1	0	-1	0	1
1	-1	0	1	0	0	-1	1	0	1
0	-1	1	-1	1	-1	1	-1	0	1
1	0	-1	-1	1	-1	1	-1	0	-1
1	-1	0	-1	1	-1	0	1	0	-1
1	0	1	0	-1	0	0	1	0	-1
1	0	-1	0	-1	1	-1	-1	0	-1
-1	-1	0	-1	1	-1	0	1	0	-1
0	0	1	1	1	0	0	1	0	-1
1	0	-1	1	1	0	0	-1	0	-1
1	0	1	0	-1	0	0	1	0	-1

Tabel 1. Dataset

Prediksi dengan *ELM*

Penelitian ini menggunakan 9 variabel input yang telah dijelaskan pada sub bab sebelumnya. Kolom ke-10 bukan termasuk variabel input karena merupakan indikator prediksi apakah suatu *website* adalah *phishing*.

1. Training *ELM*
   Sebelum data training dapat diolah oleh *ELM*, perlu dilakukan

normalisasi terhadap data tersebut sehingga nilainya konsisten dan berada di dalam rentang nilai -1 sampai 1. Normalisasi dapat dilakukan dengan menggunakan rumus berikut.

$$x = 2 \times \frac{x_p - \min x_p}{\max x_p - \min x_p} - 1$$

Keterangan:
x = nilai hasil normalisasi
x_p = nilai actual dari dataset
$Min\ x_p$ = nilai minimum dari suatu atribut
$Max\ x_p$ = nilai maksimum dari suatu atribut

X1	X2	X3	X4	X5	X6	X7	X8	X9
1	-1	1	-1	-1	1	1	1	-1
-1	-1	-1	-1	-1	0	1	1	1
1	-1	0	0	-1	0	-1	1	-1
1	0	1	-1	-1	0	1	1	-1
-1	-1	1	-1	0	0	-1	1	-1
-1	-1	1	-1	-1	1	0	-1	-1
1	-1	0	1	-1	0	0	1	-1
1	0	1	1	0	0	0	1	1
-1	-1	0	-1	-1	-1	-1	1	-1
-1	0	-1	-1	1	1	0	-1	-1
-1	-1	0	-1	-1	1	-1	-1	-1
1	0	1	1	1	-1	1	1	-1
1	-1	0	-1	1	0	1	1	-1
1	0	1	0	-1	1	0	-1	-1
-1	-1	-1	1	-1	0	-1	1	-1

Tabel 1. Data Input Setelah Normalisasi.

1. Menentukan Fungsi Aktivasi dan Jumlah *Hidden Neurons*
   Fungsi aktivasi yang digunakan adalah *sigmoid*. Jumlah *hidden neurons* yang digunakan adalah 30 setelah dilakukan percobaan untuk mendapatkan hasil paling optimal.

```
Function
[TrainingTime,TrainingAccuracy]
= elm_train(TrainingData_File,
Elm_Type, NumberofHiddenNeurons,
ActivationFunction)
```

Gambar 3. Fungsi *ELM training*

2. Perhitungan Bobot dan Bias
   Output dari pelatihan yang dilakukan yaitu model *ELM* yang kemudian dapat digunakan untuk melakukan *testing* atau prediksi. Pada model berisi bobot *input*, bobot *output* dan bias dari *hidden neurons* yang kalkulasinya dapat dilihat dibawah ini.

```
% training
dimulaistart_time_train=c
putime;

% penentuanbobot input dan bias
InputWeight=rand(NumberofHiddenNeuro
ns,NumberofInputNeurons)*2-1;
BiasofHiddenNeurons=rand(NumberofHid
denNeurons, 1);

temp=InputWeight*P; clear P;

ind=ones(1,NumberofTrainingData);
BiasMatrix=BiasofHiddenNeurons(:,ind);
tempH=tempH+BiasMatrix;

H = 1 ./ (1 + exp(-tempH)); clear
tempH;

% penentuanbobot output
OutputWeight=pinv(H') * T';

% training
selesaiend_time_train=cpu
```

Gambar 4. Kalkulasi bobot *input*, bobot *output*, dan bias.

1	0.022054	16	0.221564
2	0.849524	17	0.371421
3	0.807625	18	0.441232
4	0.542783	19	0.06069
5	0.433783	20	0.364042
6	0.602437	21	0.376869
7	0.335105	22	0.30124
8	0.778107	23	0.918389
9	0.09672	24	0.477351
10	0.451955	25	0.180392
11	0.761949	26	0.297038
12	0.625833	27	0.829299
13	0.001492	28	0.618124
14	0.017172	29	0.6636
15	0.456854	30	0.616203

Tabel 2. Bias dari *hidden neurons*

3. Denormalisasi *Output*
   Setelah didapatkannya bobot *input*, bias dari *hidden neurons* dan bobot *output*, maka dapat dilakukan perhitungan nilai *output* data training.Denormalisasi dapat dilakukan dengan menggunakan rumus berikut.

$$x = \frac{(x_p + 1) \times (max\ x_p - \min x_p) + \min x_p}{2}$$

Keterangan:
x = nilai hasil normalisasi

x_p = nilai actual dari dataset
$Min\ x_p$ = nilai minimum dari suatu atribut
$Max\ x_p$ = nilai maksimum dari suatu atribut

Y1	-0.32635	**Y1**	-0.435842
Y2	-0.455154	**Y2**	-0.656684
Y3	-0.189638	**Y3**	-0.201442
Y4	-1.23363	**Y4**	-1.991423
Y5	-0.775452	**Y5**	-1.205852
Y6	1.031677	**Y6**	1.89257
Y7	-0.188825	**Y7**	-0.200048
Y8	-0.816781	**Y8**	-1.276713
Y9	0.004273	**Y9**	0.1310293

Tabel 3. *Output* Data *Training* Sebelum (Kiri)dan Sesudah (Kanan) Denormalisasi

2. Training *ELM*
   Setelah *training ELM* berhasil dan selesai dilakukan, maka tahap selanjutnya adalah *testing ELM* dengan menggunakan model hasil keluaran *training ELM*. Seperti yang telah dijelaskan pada sub bab 3.4, jumlah *testing set* yaitu 20% dari total keseluruhan data di dalam *dataset* yaitu berjumlah 270 sampel data.

```
function [TestingTime,
TestingAccuracy] =
elm_predict(TestingData_File)
```

Gambar 5. Fungsi *ELM predict*

Setelah prediksi selesai dilakukan, sama dengan pada tahap *training ELM*, selanjutnya dilakukan denormalisasi terhadap output testing. Berikut ditunjukkan tabel berisi 9 sampel output data testing sebelum denormalisasi dan sesudah denormalisasi.

Y1	0.202812	**Y1**	0.3406026
Y2	-0.901047	**Y2**	-1.411583
Y3	-0.297764	**Y3**	-0.453975
Y4	1.061778	**Y4**	1.7040629
Y5	-1.034916	**Y5**	-1.624077
Y6	-1.053033	**Y6**	-1.652835
Y7	-1.514765	**Y7**	-2.385755
Y8	-0.425207	**Y8**	-0.656269
Y9	0.919424	**Y9**	1.4781005

Tabel 4. *Output* Data *Testing*Sebelum (Kiri)dan Sesudah (Kanan) Denormalisasi

Algoritma *Backpropagation*

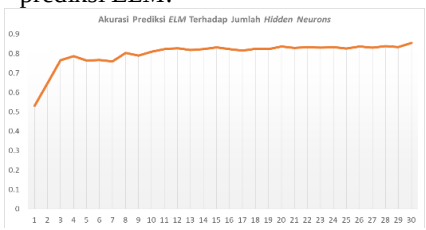
Penelitian ini menggunakan *dataset* dengan 9 fitur atau variabel *input* dan kolom ke-10 merupakan indikator prediksi apakah suatu website diklasifikasi sebagai *phishing websites*. Berikut langkah-langkah implementasi algoritma backpropagation:

1. Inisialisasi
   Pada tahap inisialisasi, data dinormalisasi agar diperoleh data yang konsisten dan dapat diolah. Data yang telah dinormalisasi kemudian disimpan ke dalam satu file terpisah. Bobot awal dihasilkan secara acak dan pemisahan *dataset* menjadi 80% *training set* dan 20% *testing set.*
2. Konfigurasi
   Fungsi aktivasi yang digunakan adalah *logistic sigmoid*. Tidak ada suatu standar untuk seberapa banyak jumlah hidden neurons yang digunakan. Oleh karena itu, agar variabel yang digunakan semirip mungkin dengan yang digunakan pada metode *ELM*, maka jumlah *hidden neurons* yang digunakan adalah 30 *neurons*.
3. Training *Backpropagation*
   Pelatihan *backpropagation* dengan menggunakan *tool* Orange dilakukan dengan *widget Neural Network* sesuai dengan konfigurasi tahap sebelumnya.
4. Testing *Backpropagation*
   Setelah training selesai dilakukan, akan mengeluarkan output model atau *Learner* dalam *tool* Orange yang kemudian digunakan untuk melakukan testing dan menganalisis keakuratan training yang telah dilakukan. Testing dengan *tool* Orange menggunakan *widget Test & Score*.

Analisis Hasil Prediksi

1. Akurasi *ELM* terhadap Jumlah *Hidden Neurons*
   Seperti yang telah dijelaskan sebelumnya bahwa jumlah *hidden neurons* yang digunakan memengaruhi hasil yang dikeluarkan oleh *ELM*. Berikut grafik yang menggambarkan pengaruh jumlah *hidden neurons* yang

digunakan terhadap tingkat akurasi prediksi *ELM*.



Gambar 6. Grafik Akurasi Prediksi *ELM* terhadap Jumlah *Hidden Neurons.*

Grafik menunjukkan dari rentang 1 sampai dengan 30, tingkat akurasi terbaik diperoleh oleh jumlah *hidden neurons* sebanyak 30 dengan angka akurasi prediksi sebesar 0.8541 atau 85.41%.

2. Keakuratan Hasil Prediksi *ELM* dengan *Confusion Matrix*.
   Untuk dapat melihat keakuratan hasil prediksi *ELM* dalam pengklasifikasian *phishing websites*, hasil prediksi *training set* dan *testing set ELM* yang diperoleh kemudian dibandingkan dengan hasil aktual *training set* dan *testing set*. Perbandingan hasil tersebut kemudian menghasilkan *confusion matrix* yang dapat dilihat pada tabel dibawah ini.

		Predicted		
		-1	**0**	**1**
Actual	**-1**	535	0	40
	0	36	9	36
	1	45	1	381

Tabel 5. *Confusion Matrix* Prediksi *ELM* pada *Training Set*

		Predicted		
		-1	**0**	**1**
Actual	**-1**	115	0	12
	0	8	3	11
	1	12	0	109

Tabel 6. *Confusion Matrix* Prediksi *ELM* pada *Testing Set*

3. Keakuratan Hasil Prediksi Algoritma *Backpropagation* dengan *Confusion Matrix*.

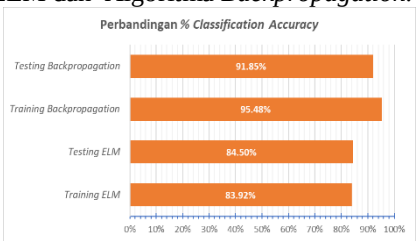
		Predicted		
		-1	**0**	**1**
Actual	**-1**	531	4	24
	0	0	82	1
	1	17	3	421

Tabel 7. *Confusion Matrix* Prediksi *ELM* pada *Training Set*

		Predicted		
		-1	**0**	**1**
Actual	**-1**	115	0	12
	0	8	3	11
	1	12	0	109

Tabel 8. *Confusion Matrix* Prediksi *ELM* pada *Testing Set*

4. Perbandingan Tingkat Keakuratan ELM dan Algoritma *Backpropagation.*



Gambar 7. Grafik Perbandingan Tingkat Akurasi Metode *ELM* dan *Backpropagation*

KESIMPULAN

Dari hasil implementasi dalam mengklasifikasi *phishing websites* dengan menggunakan metode *extreme learning machine (ELM)* dan *neural network* dengan algoritma *backpropagation*, dapat disimpulkan bahwa:

1. Jumlah *hidden neurons* yang digunakan memengaruhi hasil keluaran atau output yang dihasilkan oleh kedua metode tersebut. Untuk penelitian ini dari rentang 1 sampai 30, jumlah hidden neurons yang memberikan hasil paling optimal yaitu 30 *hidden neurons*.
2. Perbandingan tingkat akurasi dalam mengklasifikasi phishing websites dengan algoritma backpropagation memberikan tingkat akurasi yang lebih baik dibandingkan ELM dimana hasil prediksi pada testing set yang dihasilkan algoritma backpropagation

yaitu 91.85% yang 1.08 kali lebih baik dibandingkan metode ELM dengan 84.50%.

DAFTAR PUSTAKA

[1] Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006)., Extreme Learning Machine : Theory and Applications. Extreme Learning Machine : Theory and Applications, 490-501

[2] Goel, A., & Sharma, D. (2014). Prevention from hacking attacks: Phishing Detection Using Associative Classification Data Mining.

[3] Lesnussa, Y. A., Sinay, L. J., & Idah, M. R. (2017). Aplikasi Jaringan Saraf Tiruan Backpropagation untuk Penyebaran Penyakit Demam Berdarah Dengue (DBD) di Kota Ambon. Jurnal Matematika Integratif, 13(2), 63-72.

[4] APWG Q1 Report (2019). Phishing Activity Trends Report 1st Quarter 2019.

[5] Jeeva, S. C., & Rajsingh, E. B. (2016). Intelligent phishing url detection using association rule mining. Human-centric Computing and Information Sciences, 6(1), 10.

[6] Jeeva, S. C., & Rajsingh, E. B. (2016). Intelligent phishing url detection using association rule mining. Human-centric Computing and Information Sciences, 6(1), 10.

[7] Tan, P. N. (2018). Introduction to data mining. Pearson Education India.

[8] Vercellis, C. (2009). Business intelligence: data mining and optimization for decision making. New York: Wiley.

[9] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

[10] Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.

[11] Dadkhah, M., & Sutikno, T. (2015). Phishing or hijacking? Forgers hijacked DU journal by copying content of another authenticate journal. Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 3(3), 119-120.

[12] Sharma, A., Singh, P., & Kaur, A. (2015). Phishing Websites Detection Using Back Propagation Algorithm: A Review.

[13] Abdelhamid, N., Ayesh, A., & Thabtah, F. (2014). Phishing detection based associative classification data mining. Expert Systems with Applications, 41(13), 5948-5959.

[14] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[15] Rachmawati, D. (2014). Phising Sebagai Salah Satu Bentuk Ancaman Dalam Dunia Cyber. Jurnal SAINTIKOM Vol, 13(3), 210.

[16] Hansi, J., Dongsong, Z., & Zhijun, Y. (2013). A Classification Model for Detection of Chinese Phishing e-Business Websites. PACIS Proceedings.

[17] Widodo, S. (2017). Klasifikasi Situs Phishing dengan Menggunakan Neural Network dan K-Nearest Neighbor. Information Management for Educators and Professionals, 1(2), 145-154.

[18] Ali, W. (2017). Phishing Website Detection based on Supervised Machine Learning with Wrapper Features Selection. International Journal of Advanced Computer Science and Applications, 8(9), 72-78.

[19] Zareapoor, M., & Seeja, K. R. (2015). Feature extraction or feature selection for text classification: A case study on phishing email detection. International Journal of Information Engineering and Electronic Business,

7(2), 60.

[20] Gurney, K. (2014). An introduction to neural networks. CRC press.