Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 5, No 3, June 2026 | Page 128−137
DOI: 10.47065/bulletinds.v5i3.10079
https://ejurnal.seminar-id.com/index.php/bulletinds

Klasifikasi Serangan Uniform Resource Locator Phishing Menerapkan Metode Backpropagation Neural Network

M. Rival Kurniawan, Novi Yanti*, Rahmad Abdillah, Benny Sukma Negara

Fakultas Sains dan Teknologi, Prodi Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia
Email: [1]12050116070@students.uin-suska.ac.id, [2]novi_yanti@uin-suska.ac.id, [3]rahmad.abdillah@uin-suska.ac.id, [4]bsnegara@uin-suska.ac.id
Email Penulis Korespondensi: novi_yanti@uin-suska.ac.id

Abstrak−Perkembangan teknologi internet yang masif turut memicu lonjakan ancaman keamanan siber, salah satu yang paling krusial adalah serangan *phishing***.** *Phishing* merupakan bentuk kejahatan siber yang bertujuan mencuri informasi sensitif pengguna melalui metode penipuan, seperti manipulasi tautan link *Uniform Resource Locator* (URL) pada situs web. Melihat serangan siber yang makin pesat, deteksi dini terhadap URL berbahaya menjadi sangat krusial. Meskipun berbagai penelitian terdahulu telah dilakukan, model klasifikasi konvensional masih memiliki kelemahan mendasar terkait keterbatasan generalisasi dataset dan tingginya beban efisiensi komputasi saat memproses fitur dalam skala besar. Penelitian ini bertujuan untuk melakukan klasifikasi dengan menerapkan algoritma Backpropagation Neural Network (BPNN) yang terintegrasi dengan proses *Knowledge Discovery in Database* (KDD). Integrasi BPNN dan KDD dapat menyelesaikan masalah efisiensi komputasi melalui seleksi dan transformasi fitur secara matematis, sehingga beban komputasi jaringan saraf menjadi jauh lebih ringan dan akurat. Kontribusi utama dari penelitian ini adalah menghasilkan arsitektur klasifikasi yang paling optimal untuk dataset berukuran besar. Data yang digunakan bersumber dari dataset publik Kaggle (PhiUSIIL-2024) berjumlah 235.795 data URL dengan 56 fitur. Tahapan KDD dimulai dari seleksi data dengan mereduksi 5 atribut tidak relevan, hasil pembersihan data menjadi 234.611 data valid, dan transformasi menggunakan *Min-Max Scaler*. Hasil penelitian menunjukkan bahwa model BPNN memiliki kinerja paling optimal menggunakan arsitektur 50 neuron *input*, 75 neuron *hidden*, dan 1 neuron *output* (50-75-1). Hasil pengujian dengan skenario pembagian rasio data latih dan uji 90:10, *learning rate* 0.1, nilai 100 *epoch*, model memberikan nilai akurasi sebesar 99.982%, presisi 99.970%, *recall* 100%, dan *F1-Score* 99.985%. Pendekatan ini menghasilkan sistem deteksi URL *phishing* yang sangat adaptif, stabil, dan berakurasi tinggi dalam menghadapi variasi serangan siber modern.

Kata Kunci: Backpropagation Neural Network; Keamanan Siber; Klasifikasi; Phishing; URL

Abstract−The massive development of internet technology has also triggered a surge in cybersecurity threats, one of the most crucial being phishing attacks. Phishing is a form of cybercrime that aims to steal sensitive user information through fraudulent methods, such as manipulating Uniform Resource Locator (URL) links on websites. Given the increasing pace of cyberattacks, early detection of malicious URLs has become crucial. Despite various previous studies, conventional classification models still have fundamental weaknesses related to limited dataset generalization and the high computational efficiency burden when processing features on a large scale. This study aims to perform classification by implementing the Backpropagation Neural Network (BPNN) algorithm integrated with the Knowledge Discovery in Database (KDD) process. The integration of BPNN and KDD can solve the computational efficiency problem through mathematical feature selection and transformation, thus reducing the computational burden of the neural network and achieving greater accuracy. The main contribution of this study is producing the most optimal classification architecture for large datasets. The data used is sourced from the public Kaggle dataset (PhiUSIIL-2024) totaling 235,795 URLs with 56 features. The KDD process begins with data selection by reducing 5 irrelevant attributes, data cleaning results to 234,611 valid data, and transformation using Min-Max Scaler. The results show that the BPNN model has the most optimal performance using an architecture of 50 input neurons, 75 hidden neurons, and 1 output neuron (50-75-1). Test results with a training and test data ratio of 90:10, a learning rate of 0.1, and 100 epochs, the model provides an accuracy value of 99.982%, a precision of 99.970%, a recall of 100%, and an F1-Score of 99.985%. This approach produces a highly adaptive, stable, and highly accurate phishing URL detection system in the face of a variety of modern cyberattacks.

Keywords: Backpropagation Neural Network; Cyber Security; Classification; Phishing; URL

1. PENDAHULUAN

Perkembangan teknologi internet yang pesat memberikan banyak kemudahan dalam berbagai aspek kehidupan, mulai dari komunikasi, pendidikan, perdagangan, hingga transaksi keuangan. Namun, dibalik kemudahan tersebut, muncul pula berbagai ancaman keamanan siber yang dapat merugikan pengguna, salah satunya adalah serangan *phishing* [1]. *Phishing* merupakan bentuk penipuan *online* yang bertujuan mencuri informasi sensitif pengguna, seperti data pribadi, informasi keuangan, dan kata sandi. Pelaku *phishing* biasanya meniru situs web atau *email* resmi dari perusahaan, organisasi, atau bahkan individu untuk meyakinkan korban agar memberikan informasi pribadi mereka. Serangan ini dapat dilakukan melalui berbagai media, seperti *email*, SMS, situs web palsu, dan media sosial [2].

Berdasarkan laporan dari organisasi internasional *Anti-Phishing Working Group* (APWG), pada tahun 2022 terdapat lebih dari 4.7 juta kasus serangan *phishing*. Angka ini meningkat sekitar 150% setiap tahun sejak awal 2019 [3]. Banyak korban *phishing* mengalami kerugian besar, terutama terkait dengan hal-hal seperti pencurian data, penggunaan informasi secara tidak sah, dan kerugian finansial yang cukup signifikan. Oleh karena itu, melindungi diri dari serangan *phishing* adalah langkah penting untuk mencegah tindakan kejahatan siber [4]. Tingginya angka serangan ini sebagian besar disebabkan oleh rendahnya kesadaran pengguna terhadap keamanan siber serta meningkatnya penggunaan teknologi digital seperti media sosial, *email*, dan platform pembayaran *online* [5][6].

Lebih lanjut, *phishing* URL merupakan alamat sumber yang dirancang khusus sebagai alat utama dalam kejahatan siber [7]. Berbeda dari URL sah yang dapat menunjukkan lokasi atau sumber data pada web, sedangkan *phishing* URL

Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 5, No 3, June 2026 | Page 128−137
DOI: 10.47065/bulletinds.v5i3.10079
https://ejurnal.seminar-id.com/index.php/bulletinds

sengaja dimanipulasi oleh penyerang [8]. Manipulasi *phishing* URL dapat berupa pemalsuan tampilan halaman *login*, sertifikat SSL palsu, serta eksploitasi layanan pemendek alamat URL [9]. Situs ancaman ini dirancang untuk meniru situs asli dengan sangat meyakinkan, memodifikasi secara tipografi, misalnya dengan mengganti huruf 'o' dengan '0'. Identifikasi *phishing* URL bisa dilakukan dengan mengecek ciri-ciri struktur URL yang biasa digunakan dalam serangan tersebut [10]. Beberapa karakteristik umum dari *phishing* URL meliputi penggunaan alamat IP numerik sebagai pengganti nama domain untuk menutupi lokasi *server* sebenarnya dari pengguna [11]. Selain itu dapat berupa URL yang panjang lebih dari 54 karakter untuk menyembunyikan domain berbahaya. Menggunakan tanda hubung ('-') dalam domain juga sering dilakukan untuk meniru merek yang sudah terkenal [12].

Mengingat Meskipun beberapa penelitian terdahulu telah menunjukkan performa yang baik, masih terdapat kelemahan dan tantangan yang menjadi celah kesenjangan (GAP) penelitian. Keterbatasan ukuran dan variasi dataset pada model deteksi phishing sering menyebabkan model konvensional berperforma buruk saat dihadapkan dengan data baru. Selain itu, waktu komputasi yang tinggi dan kompleksitas pemilihan parameter pada algoritma seperti SVM atau KNN menjadi tantangan tersendiri bagi efisiensi sistem. Model konvensional ini kerap menunjukkan ketidakstabilan akurasi dan kinerja komputasional ketika beban dimensi data membesar. Mengatasi GAP tersebut, penelitian ini mempertegas solusi dengan mengusulkan metode Backpropagation Neural Network (BPNN). Berbeda dengan konvensional, BPNN memiliki kelebihan utama dalam mempelajari pola non-linear yang sangat kompleks secara lebih mendalam melalui propagasi galat [13]. Penelitian lain oleh Vebriani & Yustanti (2024) menerapkan metode *Support Vector Machine* (SVM) untuk klasifikasi deteksi *link phishing* khusus kategori "DANA Kaget" berbasis *website*, menunjukkan performa stabil dengan akurasi *train* 90% dan *test* 88% menggunakan pengujian *fold* 10 [14]. Selain itu, penelitian Adipa dkk. (2023) yang menggunakan algoritma *K-Nearest Neighbor* untuk klasifikasi *email phishing* menghasilkan tingkat akurasi sebesar 84% [15]. Penelitian serupa yang sangat relevan juga dilakukan oleh Hasibuan dkk. (2025) dengan mengevaluasi algoritma *Random Forest* untuk deteksi *phishing* URL menggunakan pendekatan KDD pada *dataset* modern berskala besar PhiUSIIL-2024, menganalisis dampak seleksi fitur berdasarkan *feature importance*. Hasilnya menunjukkan bahwa model yang disederhanakan menggunakan *RF-Top 30 configurations* terbukti efektif dalam menjaga efisiensi sistem tanpa mengorbankan performa, menghasilkan metrik akurasi, presisi, *recall*, dan *F1-score* yang stabil dan mendekati 100% pada berbagai rasio pembagian data [16].

Meskipun beberapa penelitian terdahulu telah menunjukkan performa yang baik, masih terdapat kelemahan dan tantangan yang menjadi celah kesenjangan penelitian. Keterbatasan ukuran dan variasi dataset pada model deteksi phishing sering menyebabkan model konvensional berperforma buruk saat dihadapkan dengan data baru. Selain itu, waktu komputasi yang tinggi dan kompleksitas pemilihan parameter pada algoritma seperti SVM atau KNN menjadi tantangan tersendiri bagi efisiensi sistem. Model konvensional ini kerap menunjukkan ketidakstabilan akurasi dan kinerja komputasional ketika beban dimensi data membesar [17]. Untuk memastikan BPNN ini tidak memakan waktu komputasi yang berat (efisien), penelitian ini mengintegrasikannya dengan kerangka pra-pemrosesan *Knowledge Discovery in Database* (KDD). KDD berperan vital mereduksi dimensi data mentah, sehingga BPNN hanya melatih fitur yang murni relevan, menjaga stabilitas model secara utuh dan mempercepat konvergensi pelatihan [18].

Oleh karena itu, untuk mengatasi tantangan dan celah dari batasan penelitian sebelumnya, penelitian ini mengusulkan penerapan metode *Backpropagation Neural Network* (BPNN) untuk melakukan klasifikasi serangan *phishing* URL. BPNN juga merupakan salah satu algoritma pembelajaran terawasi yang terdiri dari komponen *input layer*, *hidden layer*, dan *output layer* yang secara terus menerus bekerja dengan memperbarui nilai bobot untuk mengurangi kesalahan hasil propagasi balik [19][20]. Jaringan saraf tiruan dipilih karena memiliki kelebihan utama, yaitu kemampuannya dalam mempelajari pola *non-linear* yang kompleks secara lebih mendalam dibandingkan dengan algoritma konvensional standar [21]. Setiap neuron arsitektur BPNN menyesuaikan nilai bobot berdasarkan tingkat kesalahan yang diterima, secara sistematis mengurangi perbedaan antara *output* yang dihasilkan dan meningkatkan target nilai akurasi. Sehingga sistem klasifikasi berakurasi sangat tinggi dan mampu menghadapi berbagai macam variasi trik manipulasi *phishing* URL yang terus berkembang. Selain itu, tujuan dari penelitian ini adalah merancang, mengimplementasikan, dan mengevaluasi sistem pendeteksi serangan URL phishing menggunakan metode BPNN terintegrasi KDD, guna menemukan konfigurasi arsitektur jaringan saraf tiruan terbaik yang mampu mengenali pola tautan berbahaya secara efisien. Kontribusi utama dari hasil penelitian ini adalah penyediaan model deteksi ancaman siber berbasis artificial intelligence yang terbukti kuat, stabil secara matematis, memiliki akurasi tinggi mendekati sempurna, serta waktu komputasi yang efisien dalam memvalidasi keabsahan sebuah tautan pada dataset berskala besar, sehingga dapat diimplementasikan untuk pelindungan sistem informasi digital modern.

2. METODOLOGI PENELITIAN

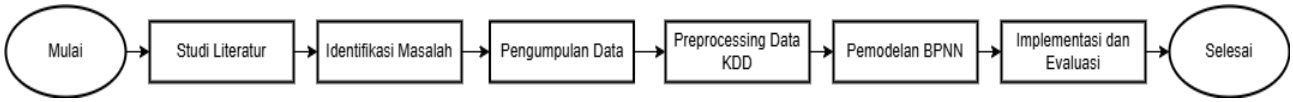
2.1 Tahapan Penelitian

Pada penelitian ini melalui beberapa tahapan mencakup studi literatur, identifikasi masalah, pengumpulan data, tahap pemrosesan awal data berbasis *Knowledge Discovery in Database* (KDD), pemodelan menggunakan *Backpropagation Neural Network* (BPNN), tahapan implementasi dan evaluasi performa model seperti terlihat Gambar 1.

Gambar 1 mengilustrasikan alur kerja penelitian yang terstruktur menjadi tujuh tahapan utama. Proses ini diawali dengan studi literatur untuk memahami konsep serangan siber, dilanjutkan dengan perumusan masalah, pengumpulan

Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 5, No 3, June 2026 | Page 128−137
DOI: 10.47065/bulletinds.v5i3.10079
https://ejurnal.seminar-id.com/index.php/bulletinds

dataset, hingga tahapan pra-pemrosesan data menggunakan standar KDD. Setelah data siap, dilakukan pemodelan menggunakan jaringan saraf tiruan yang diakhiri dengan fase evaluasi performa.



Gambar 1. Tahapan Penelitian

2.2 Studi Literatur dan Identifikasi Masalah

Tahap awal penelitian adalah melakukan studi literatur untuk memahami konsep dasar dan penelitian terkait serangan *phishing*, pengolahan data *machine learning*, dan mekanisme algoritma BPNN. Berdasarkan studi literatur dilakukan identifikasi masalah dengan rumusan dan fokus pada tiga tahap utama. Pertama, pencarian dan penyiapan *dataset* berskala besar yang relevan untuk klasifikasi *phishing* URL karena validitas model sangat bergantung pada kualitas data masukan. Kedua, penentuan alur klasifikasi yang optimal, mencakup tahapan pra-pemrosesan data, pembagian proporsi data latih dan uji menggunakan metode *data split validation* dengan membagi data menjadi data latih dan data uji dengan rasio yang ditentukan. Menemukan nilai metrik evaluasi untuk akurasi, presisi, *recall*, dan *F1-score*. Ketiga, konfigurasi implementasi algoritma BPNN secara optimal untuk mendeteksi pola data URL, meliputi penyesuaian parameter jaringan seperti jumlah *neuron*, *learning rate*, jumlah *hidden layer*, dan iterasi *epoch.*

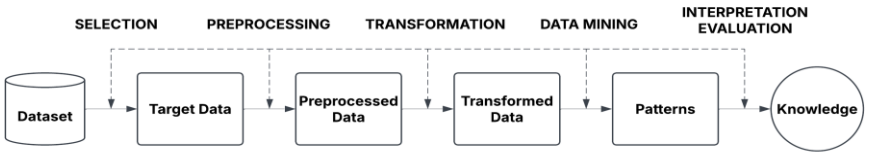
2.3 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder publik yang diunduh dari platform repositori data *machine learning* Kaggle PhiUSIIL-2024 dengan volume data yang sangat besar, karakteristik mutakhir, dan sangat relevan dengan tujuan penelitian terkait klasifikasi ancaman siber berbasis manipulasi tautan. *Dataset* dengan total 235.795 sampel data URL, yang didistribusikan ke dalam dua kelas utama dengan 134.850 URL kategori aman yang dilabel dengan kelas 1 (*non-phishing*) dan 100.945 URL kategori berbahaya yang dilabel dengan kelas 0 (*phishing*). Data tersebut dibentuk oleh 56 fitur (atribut) yang merepresentasikan berbagai karakteristik teknis dan leksikal dari setiap struktur URL, seperti perhitungan panjang URL, ketersediaan tanda baca atau karakter khusus (seperti karakter ?, =, %, $), penghitungan jumlah *subdomain*, verifikasi protokol keamanan HTTPS, rasio huruf dan angka, hingga pengenalan pola penyamaran (*obfuscated characters*) yang umum digunakan *hacker* dalam melancarkan serangan penipuan digital.

Sementara itu, distribusi kelas target memiliki *imbalance ratio* yang termasuk *slightly imbalanced*, yaitu 57% kelas aman (134.850 data non-phishing) dan 43% kelas berbahaya (100.945 data phishing). Rasio ketidakseimbangan yang tidak terlampau ekstrem ini memastikan bahwa model masih sangat reliabel untuk dilatih tanpa harus menggunakan metode *oversampling* tambahan, serta terhindar dari potensi *overfitting* pada salah satu kelas dominan.

2.4 *Preprocessing Knowledge Discovery in Database* (KDD)

Untuk menghasilkan klasifikasi yang berakurasi tinggi, *dataset* mentah tidak dapat langsung digunakan ke dalam algoritma pelatihan. Data tersebut harus melalui serangkaian proses, dimulai dari pra-pemrosesan data (*preprocessing*) dengan merujuk pada tahapan standar KDD [22]. Tahapan dalam proses KDD dapat dilihat melalui Gambar 2.



Gambar 2. Tahapan Knowledge Discovery in Database

Gambar 2 diatas menjabarkan lima tahapan dari *Knowledge Discovery in Database* (KDD) yang krusial untuk mengolah data URL mentah menjadi pola yang dapat dianalisis. Tahapan ini mencakup seleksi fitur target, pembersihan data *(preprocessing)*, transformasi data ke dalam format yang dapat dihitung secara matematis, penggalian pola (data mining), hingga interpretasi akhir menjadi sebuah pengetahuan *(knowledge)*. Proses KDD adalah sebagai berikut:

a. *Data Selection*
   Pada proses seleksi, dilakukan penyaringan awal pada *dataset* PhiUSIIL-2024. Seleksi bertujuan untuk menyaring informasi yang berdampak signifikan pada proses penggalian *knowledge* dan memastikan ketersediaan label target biner yaitu kelas 1 untuk *non-phishing* dan kelas 0 untuk *phishing*.
b. *Preprocessing*/ *Data Cleaning*
   Proses eksekusi pembersihan data, seluruh sampel yang mengandung *missing values*, data duplikat ditangani dan dibersihkan dari himpunan data uji. Proses ini menyebabkan penyusutan baris data dari 235.795 data menjadi 234.611 data (terdiri atas 134.849 *non-phishing* dan 99.762 *phishing*). Pada tahapan ini 5 atribut (FILENAME, URL, DOMAIN, TLD, dan TITLE) dieksklusi dan dihapus dari himpunan fitur karena keberadaannya hanya

Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 5, No 3, June 2026 | Page 128−137
DOI: 10.47065/bulletinds.v5i3.10079
https://ejurnal.seminar-id.com/index.php/bulletinds

merepresentasikan indeks atau label penamaan tanpa memiliki korelasi matematis terhadap sifat *phishing*. Hal ini mereduksi total fitur algoritma dari 56 menjadi 51 atribut fungsional (50 atribut kontinu dan 1 atribut diskrit).

c. Transformasi Data/ Normalisasi Data
Untuk memastikan seluruh parameter dapat diolah secara efisien oleh BPNN, dilakukan transformasi nilai dari teks ke bentuk numerik melalui pendekatan *Label Encoding*. Variabel kategorikal *"phishing"* nilai 0 dan *"non-phishing"* nilai 1. Rentang nilai 50 variabel kontinu memiliki disparitas nilai yang tinggi, teknik Normalisasi Data *Min-Max Scaler* diimplementasikan untuk melakukan penempatan rasio ke dalam skala seimbang, yaitu pada batas bawah 0 hingga batas atas 1. Pendekatan skala logaritmik ini sangat vital untuk mencegah dominasi fitur berangka besar (seperti *"URLLength"*) terhadap fitur lain yang memiliki skala kecil dalam kalkulasi bobot BPNN [23].

2.5 Pemodelan *Backpropagation Neural Network* (BPNN)

Setelah tahap transformasi selesai, kumpulan data yang telah dinormalisasi diterapkan ke dalam model pembelajaran terawasi menggunakan algoritma BPNN. Jaringan dirancang dengan arsitektur input layer 50 *node*, *hidden layer*, dan *output layer* 1 *node* kelas prediksi. Untuk menguji reliabilitas, arsitektur divalidasi menggunakan mekanisme *split validation* dengan membagi proses pelatihan dan pengujian menjadi 3 rasio yaitu 70:30, 80:20 dan 90:10. Konfigurasi jaringan bermula dari penempatan nilai acak pada bobot matriks (W) dan intersep bias (b). Pelatihan terbagi dalam dua siklus berkesinambungan. Siklus maju (*forward propagation*) untuk menghitung masukan melalui agregasi linier ke seluruh *node* di *hidden layer*, kemudian diproses oleh fungsi aktivasi *sigmoid* biner untuk membentuk nilai estimasi *output*. Sedangkan siklus mundur (*backward propagation*) mengevaluasi gradien perbedaan (selisih tingkat kesalahan/ *error*) antara nilai estimasi dengan label aktual. Proyeksi gradien galat (*error gradient*) tersebut kemudian dialirkan secara terbalik dari *output layer* ke *input layer* guna mengkalkulasi selisih penyesuaian bobot delta (ΔW). Modifikasi adaptif bobot sinaptik ini terus beriterasi sampai konvergensi nilai laju kesalahan berhasil menyentuh ambang batas toleransi minimum atau menyentuh nilai *epoch* maksimal.

2.6 Implementasi, Skenario Pengujian, dan Evaluasi

Tahapan implementasi menggunakan bahasa pemrograman *Python* dengan antarmuka *Jupyter Notebook* menggunakan *Google Chrome* dengan spesifikasi processor intel core i7 gen 7, CPU 2.80 GHz, RAM 8 GB dan SSD 1TB. Skenario pengujian dirancang berjalan pada kombinasi tiga variasi utama dengan fungsi neuron pada struktur *hidden layer* (51, 75, dan 99 *neuron*) Rasio pembagian *data split validation* untuk himpunan *Training: Testing* adalah 70:30, 80:20, dan 90:10). Indeks *learning rate* konstan menggunakan nilai 0.1, 0.01 dan 0.001, serta batasan perulangan *epoch* pembelajaran jaringan adalah 50 dan 100 iterasi. Tahap pengujian menggunakan *Confusion Matrix*, dan evaluasi keakuratan detektor model analitis menggunakan empat variabel penilaian komputasi utama, yaitu *Accuracy*, *Precision*, *Recalol*, dan *F1-Score*. Berikut rumus hasil evaluasi menggunakan *confusion matrix* [24]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1-Score = 2 * \frac{(Precision * Recall)}{(Precision+Recall)} \quad (4)$$

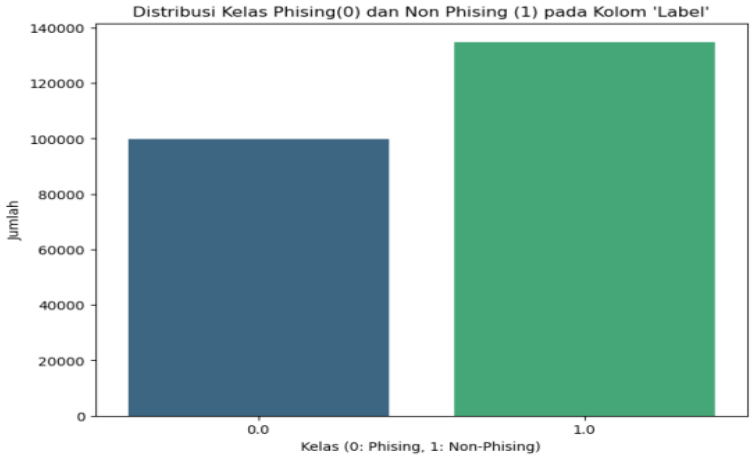
Pada persamaan (1) hingga (4), variabel TP *(True Positive)* mewakili jumlah data yang diprediksi positif dan terdeteksi dengan benar, sedangkan TN *(True Negative)* menunjukkan jumlah data yang diprediksi negatif dan terdeteksi dengan benar. Sementara itu, kesalahan dalam klasifikasi direpresentasikan oleh variabel FP *(False Positive)*, yakni jumlah data yang diprediksi positif namun terdeteksi dengan salah, serta FN *(False Negative)*, yang merupakan jumlah data yang diprediksi negatif namun terdeteksi dengan salah.

3. HASIL DAN PEMBAHASAN

3.1 Analisa Pengumpulan Data

Analisa pengumpulan data merupakan langkah awal dalam tahapan KDD. Data yang digunakan dalam penelitian ini bersumber dari repositori publik Kaggle PhiUSIIL-2024. Dataset ini dikembangkan secara khusus untuk studi keamanan siber dan berisi kumpulan data historis dari ribuan URL, baik yang bersifat aman (*legitimate*) maupun yang teridentifikasi sebagai ancaman *phishing*. Pada tahap awal sebelum dilakukan manipulasi, dataset ini memiliki dimensi yang sangat besar, yaitu terdiri dari 235.795 baris data (URL) dengan 56 fitur (kolom). Fitur-fitur ini merepresentasikan berbagai karakteristik teknis, struktural, leksikal, dan jaringan dari sebuah URL, seperti atribut panjang karakter URL (*URLLength*), probabilitas legitimasi *Top-Level Domain* (*TLDLegitimateProb*), jumlah karakter khusus, hingga keberadaan informasi hak cipta (*HasCopyrightInfo*). Berdasarkan analisis eksplorasi data awal (EDA) pada bahasa pemrograman *Python* menggunakan pustaka *Pandas* dan *Seaborn*, distribusi kelas target seperti pada Gambar 3.

Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 5, No 3, June 2026 | Page 128−137
DOI: 10.47065/bulletinds.v5i3.10079
https://ejurnal.seminar-id.com/index.php/bulletinds



Gambar 3. Diagram Jumlah Kelas Data Phising dan Non-Phising dalam Dataset

Pada Gambar 3 memperlihatkan visualisasi distribusi kelas pada dataset publik Kaggle PhiUSIIL-2024. Hasil eksplorasi data menunjukkan bahwa kelas 0 *(phishing)* memiliki 100.945 data, sedangkan kelas 1 *(non-phising)* memiliki 134.849 data. Distribusi ini menunjukkan bahwa dataset berada dalam kondisi yang relatif seimbang (*slightly imbalanced*), namun masih dalam batas toleransi yang sangat baik untuk pelatihan. Tabel 1 dibawah ini merupakan dataset awal.

Tabel 1. Dataset Awal

Data Ke-	FileName	URL	URLLength	…	NoOfSelfRef	NoOfEmptyRef	NoOfExternalRef	Label
1.	521848.txt	https://www.southbankmosaics.com	31	…	39	0	217	1
2.	31372.txt	https://www.uni-mainz.de	23	…	42	2	5	1
3.	597387.txt	https://www.voicefmradio.co.uk	29	…	22	1	31	1
…	…	…	…	…	…	…	…	…
235.793	622132.txt	https://www.nononsensedesign.be	30	…	58	2	67	1
235.794	7503962.txt	https://patient-cell-40f5.updatedlogmylogin.workers.dev/	55	…	0	0	0	0
235.795	384822.txt	https://www.alternativefinland.com	33	…	256	0	261	1

Tabel 1 menyajikan struktur dataset mentah PhiUSIIL-2024 yang memiliki dimensi 235.795 baris dan 56 kolom. Pada tahap ini, dataset masih memuat fitur-fitur identifier berbasis teks seperti “FileName” dan “URL” utuh, yang belum siap untuk digunakan dalam perhitungan propagasi linier.

3.2 Analisa Proses KDD

Analisa proses KDD menggunakan pemrograman *python* untuk menghasilkan dataset yang sesuai dan relevan, sehingga dapat dilakukan proses klasifikasi menggunakan metode BPNN. Analisa proses KDD yang dilakukan sebagai berikut:

3.2.1 Analisa Hasil *Data Selection*

Analisa pada tahap *data selection* dilakukan dengan mengeliminasi fitur-fitur yang tidak memberikan nilai informatif atau tidak relevan terhadap proses pembelajaran mesin. BPNN bekerja dengan melakukan komputasi matematis terhadap bobot numerik. Atribut yang bernilai *string* unik atau identifier berupa teks yang tidak memiliki relasi pola repetitif terhadap probabilitas *phishing* dibuang. Dilakukan penghapusan terhadap 5 atribut utama, yaitu: FILENAME, URL, Domain, TLD (Top-Level Domain), dan Title. Atribut yang dihapus bersifat sebagai metadata identifier yang unik untuk setiap baris, bukan merupakan pola matriks yang dapat dipelajari secara matematis oleh jaringan saraf. Setelah proses *data selection*, jumlah kolom (fitur) semula 56 atribut menjadi 51 atribut.

3.2.2 Analisa Hasil *Preprocessing*

Setelah atribut yang tidak relevan dieliminasi, langkah selanjutnya adalah pembersihan data dari *missing values* atau nilai kosong (direpresentasikan sebagai NaN atau *Not a Number* dalam *dataframe* Pandas). Nilai yang kosong dapat merusak proses kalkulasi *gradient descent* dan *forward propagation* pada BPNN, sehingga menghasilkan *error* komputasi. Pengecekan *missing value* dilakukan pada 51 atribut dengan metode pembersihan *Listwise Deletion* (menghapus seluruh baris data yang mengandung minimal satu nilai NaN). Setelah proses pembersihan dilakukan, jumlah baris data semula 235.795 data menjadi 234.611 dataset.

Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 5, No 3, June 2026 | Page 128−137
DOI: 10.47065/bulletinds.v5i3.10079
https://ejurnal.seminar-id.com/index.php/bulletinds

Tabel 2. Dataset Hasil Proses *Preprocessing*

Data Ke-	URLLength	DomainLength	IsDomainLength	…	NoOfSelfRef	NoOfEmptyRef	NoOfExternalRef	label
1.	23	16	0	…	39	0	217	1
2.	29	22	0	…	42	2	5	1
3.	26	19	0	…	22	1	31	1
…	…	…	…	…	…	…	…	…
234.609	30	23	0	…	58	2	67	1
234.610	55	47	0	…	0	0	0	0
234.611	33	26	0	…	256	0	261	1

Tabel 2 menampilkan hasil dataset pasca pembersihan menggunakan metode *Listwise Deletion*, di mana baris dengan missing values telah dihapus, mengurangi jumlah keseluruhan menjadi 234.611 data. Lima atribut identifier yang tidak memiliki korelasi matematis (FILENAME, URL, Domain, TLD, TITLE) juga telah dieliminasi, sehingga hanya tersisa 51 atribut fungsional.

3.2.3 Analisa Hasil *Data Transformation* / *Data Normalization*

Tahap *Data Transformation* tidak dilakukan karena seluruh atribut bertipe string yang tidak relevan tidak digunakan, sehingga 50 fitur prediktor berformat numerik yang kompatibel dengan algoritma BPNN. Sebaliknya, proses *data normalization* dilakukan karena fitur-fitur numerik dalam dataset *Kaggle* PhiUSIIL-2024 memiliki rentang skala yang tidak simbang. Normalisasi skala diterapkan untuk menyeragamkan rentang data agar variabel bernilai besar tidak mendominasi proses pembaruan bobot (*weight updates*) selama pelatihan untuk mencegah bias pada model dan mempercepat BPNN mencapai titik konvergensi optimal dalam mengklasifikasikan *phishing* URL.

Tabel 3. Hasil Normalisasi Data

Data Ke-	URLLength	DomainLength	IsDomainLength	…	NoOfSelfRef	NoOfEmptyRef	NoOfExternalRef	label
1.	0.001643655	0.113207547	0	…	0.001423514	0	0.007886321	1
2.	0.002629849	0.169811321	0	…	0.001533015	0.000409249	0.000181712	1
3.	0.002136752	0.141509434	0	…	0.000803008	0.000204625	0.001126617	1
…	…	…	…	…	…	…	…	…
234.609	0.002794214	0.179245283	0	…	0.00211702	0.000409249	0.002434947	1
234.610	0.006903353	0.405660377	0	…	0	0	0	0
234.611	0.003287311	0.20754717	0	…	0.009344089	0	0.00948539	1

Tabel 3 menampilkan himpunan data yang telah ditransformasi menggunakan teknik Normalisasi Data *Min-Max Scaler*. Normalisasi ini memampatkan rentang disparitas yang tinggi pada atribut kontinu menjadi batas bawah 0 dan batas atas 1, memastikan bahwa algoritma BPNN dapat mencapai titik konvergensi secara lebih efisien dan menghindari pembiasan.

3.3 Analisa Proses Pembagian Data

Analisa proses pembagian data dilakukan untuk mengetahui jumlah data latih dan data uji. Metode pembagian data menggunakan *split validation* yang berfungsi membagi dataset menjadi dua bagian, yaitu data latih (*data training*) dan data uji (*data testing*) secara acak berdasarkan rasio. Pada Tabel 4 berikut ini menampilkan hasil kuantitas pembagian rasio data latih dan data uji, untuk menemukan skema performa model yang paling adaptif.

Tabel 4. Pembagian Data Latih dan Data Uji

Data	Rasio Pembagi dan Jumlah Data		
	70 : 30	80 : 20	90 : 10
Latih	164.228	187.689	211.150
Uji	70.383	46.922	23.461

3.4 Analisa Akhir dan Evaluasi Penelitian

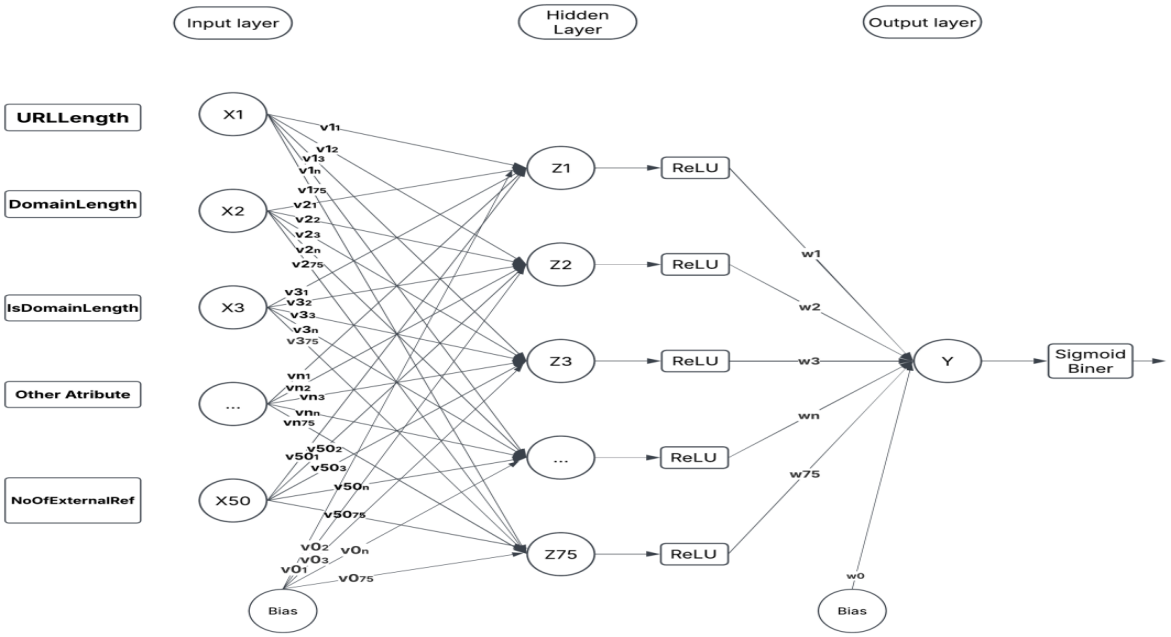
Analisa akhir penelitian dilakukan untuk mendapatkan hasil terbaik. Terdapat beberapa skenario yang diterapkan, salah satunya seperti yang terlihat pada Gambar 4 Arsitektur BPNN dan Tabel 5 skenario penelitian berikut ini.

Pada Gambar 4 menampilkan spesifikasi visual struktur arsitektur BPNN yang berfokus pada variasi neuron 50-75-1. Desain ini menggunakan 50 node pada layer masukan yang mewakili setiap fitur data, dialirkan melalui satu layer tersembunyi berisikan 75 neuron pemrosesan dengan fungsi aktivasi ReLU, dan diproyeksikan pada 1 node lapisan keluaran menggunakan aktivasi *Sigmoid Biner* untuk menentukan hasil klasifikasi.

Tabel 5 menunjukkan matriks parameter untuk eksperimen pelatihan BPNN. Skenario ini melibatkan manipulasi nilai iterasi epoch (50 dan 100) dan modifikasi konstan *learning rate* (0.1, 0.01, 0.001) terhadap tiga perancangan konfigurasi hidden layer (51, 75, dan 99 neuron) serta penggunaan fungsi aktivasi ReLU pada jalur dari *hidden layer* menuju *output layer* dan fungsi aktivasi pada *output layer* untuk menentukan komposisi yang menghasilkan tingkat kesalahan terkecil. Setelah beberapa skenario telah dilakukan, selanjutnya adalah membandingkan hasil pengujian

Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 5, No 3, June 2026 | Page 128−137
DOI: 10.47065/bulletinds.v5i3.10079
https://ejurnal.seminar-id.com/index.php/bulletinds

berdasarkan nilai *accuracy*, *precision*, *recall*, dan *f1-score* dengan menggunakan beberapa parameter seperti terlihat pada Tabel 6 Hasil Pengujian.



Gambar 4. Arsitektur BPNN dengan Output Layer 50-75-1

Berikut pada Tabel 5 merupakan Skenario Pengujian.

Tabel 5. Skenario Pengujian

Kriteria	Pembagian
Fungsi Aktivasi	*ReLU* pada *Hidden Layer* dan *Sigmoid Biner* pada *Output Layer*
Pembagian Dataset	70:30, 80:20 dan 90:10
Jumlah *Hidden Layer*	1
Learning Rate	0.1, 0.01dan 0.001
Epoch	50 dan 100
Jumlah *Neuron*	50 Neuron pada *Input Layer*, 51,75 dan 99 Neuron pada *Hidden Layer* dan 1 Neuron pada *Output Layer* (50-51-1), (50-75-1) dan (50-99-1)

Pada Tabel 6 menjelaskan hasil pengujian klasifikasi BPNN dengan rancangan arsitektur *input layer* sebanyak 50 *neuron*, *output layer* 1 *neuron*, serta variasi *hidden layer* 50-51-1, 50-75-1, dan 50-99-1, model BPNN mampu membentuk pola dari *dataset phishing* URL secara optimal. Arsitektur dapat secara konsisten menerapkan fungsi aktivasi *ReLU* pada *hidden layer* dan fungsi aktivasi *sigmoid biner* pada *output layer*. Evaluasi dilakukan dengan menggunakan beberapa skenario pembagian data yang terdiri atas 3 rasio yaitu 70:30, 80:20, dan 90:10. Pengujian pada *dataset* menggunakan kombinasi variasi *epoch* sebesar 50 dan 100 *epoch*, nilai *learning rate* 0.1, 0.01 dan 0.001, dan jumlah *neuron* pada *hidden layer* 51, 75, dan 99.

Tabel 6. Hasil Pengujian

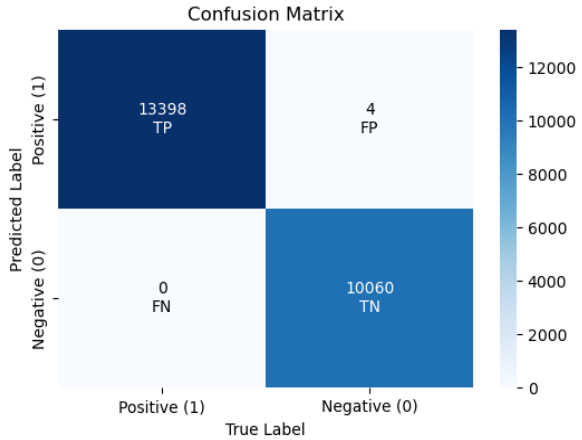
Jumlah *Neuron Hidden*	*Ratio*	*Learning Rate*	*Epoch*	*Accuracy*	*Precision*	*Recall*	*F1-Score*
51	70:30	0.1	50	99.965%	99.962%	99.977%	99.970%
		0.1	100	99.964%	99.967%	99.970%	99.968%
		0.01	50	99.937%	99.920%	99.970%	99.945%
		0.01	100	99.951%	99.927%	99.987%	99.957%
		0.001	50	99.738%	99.659%	99.883%	99.771%
		0.001	100	99.853%	99.783%	99.960%	99.871%
	80:20	0.1	50	99.968%	99.955%	99.988%	99.972%
		0.1	100	99.976%	99.962%	99.996%	99.979%
		0.01	50	99.933%	99.921%	99.962%	99.942%
		0.01	100	99.948%	99.940%	99.970%	99.995%
		0.001	50	99.786%	99.727%	99.899%	99.813%
		0.001	100	99.850%	99.794%	99.944%	99.869%
	90:10	0.1	50	99.978%	99.962%	100.000%	99.981%
		0.1	100	99.978%	99.970%	99.992%	99.981%
		0.01	50	99.948%	99.940%	99.970%	99.955%
		0.01	100	99.961%	99.947%	99.985%	99.966%
		0.001	50	99.778%	99.739%	99.873%	99.806%

Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 5, No 3, June 2026 | Page 128−137
DOI: 10.47065/bulletinds.v5i3.10079
https://ejurnal.seminar-id.com/index.php/bulletinds

Jumlah *Neuron Hidden*	*Ratio*	*Learning Rate*	*Epoch*	*Accuracy*	*Precision*	*Recall*	*F1-Score*
		0.001	100	99.897%	99.858%	99.962%	99.910%
75	70:30	0.1	50	99.965%	99.955%	99.985%	99.970%
		0.1	100	99.976%	99.962%	99.996%	99.997%
		0.01	50	99.940%	99.915%	99.980%	99.947%
		0.01	100	99.955%	99.940%	99.982%	99.996%
		0.001	50	99.741%	99.664%	99.883%	99.773%
		0.001	100	99.843%	99.773%	99.952%	99.863%
	80:20	0.1	50	99.965%	99.947%	99.992%	99.970%
		0.1	100	99.976%	99.966%	99.992%	99.979%
		0.01	50	99.936%	99.917%	99.970%	99.944%
		0.01	100	99.953%	99.936%	99.981%	99.958%
		0.001	50	99.767%	99.713%	99.880%	99.796%
		0.001	100	99.848%	99.794%	99.940%	99.867%
	90:10	0.1	50	99.970%	99.970%	99.977%	99.973%
		0.1	**100**	**99.982%**	**99.970%**	**100%**	**99.985%**
		0.01	50	99.944%	99.932%	99.970%	99.995%
		0.01	100	99.961%	99.947%	99.985%	99.966%
		0.001	50	99.791%	99.731%	99.902%	99.817%
		0.001	100	99.876%	99.835%	99.947%	99.891%
99	70:30	0.1	50	99.968%	99.950%	99.995%	99.972%
		0.1	100	99.971%	99.957%	99.992%	99.975%
		0.01	50	99.970%	99.970%	99.977%	99.973%
		0.01	100	99.948%	99.932%	99.977%	99.955%
		0.001	50	99.744%	99.662%	99.890%	99.776%
		0.001	100	99.846%	99.776%	99.955%	99.865%
	80:20	0.1	50	99.974%	99.966%	99.988%	99.977%
		0.1	100	99.974%	99.966%	99.988%	99.977%
		0.01	50	99.936%	99.921%	99.966%	99.944%
		0.01	100	99.955%	99.936%	99.985%	99.960%
		0.001	50	99.774%	99.716%	99.888%	99.802%
		0.001	100	99.848%	99.798%	99.936%	99.867%
	90:10	0.1	50	99.974%	99.970%	99.985%	99.977%
		0.1	100	99.978%	99.970%	99.992%	99.998%
		0.01	50	99.948%	99.925%	99.985%	99.955%
		0.01	100	99.961%	99.947%	99.985%	99.966%
		0.001	50	99.803%	99.739%	99.917 %	99.828%
		0.001	100	99.893%	99.843%	99.970%	99.906%

Sedangkan untuk hasil evaluasi performa tahap akhir dipetakan menggunakan pengukuran *confusion matrix* yang diambil dari skenario pengujian terbaik. Evaluasi *confusion matrix* ini dikalkulasikan secara analitis untuk melihat ketepatan prediksi model dalam mengklasifikasi secara benar berdasarkan tautan URL yang merupakan anomali *phishing* dan tautan URL yang aman (*non-phishing*) pada data uji.

Gambar 5 merupakan hasil pengujian *confusion matrix* dengan performa terbaik yaitu nilai *accuracy*, *precision*, *recall*, dan *f1-score* yang sangat stabil dan sangat tinggi di seluruh konfigurasi. Hasil performa tertinggi didapat pada skenario pembagian data dengan rasio 90:10 menggunakan *learning rate* 0.1 dan 100 *epoch*, baik pada model dengan 51 *neuron* maupun 75 *neuron hidden layer*. Sebagai contoh, pada arsitektur 75 *neuron* (50-75-1) nilai *accuracy* yang dihasilkan mencapai 99.982%, dengan nilai *precision* 99.970%, *recall* mencapai angka sempurna 100%, dan *f1-score* sebesar 99.985%. Sehingga, performa terbaik terlihat pada Gambar 5 merupakan hasil *confusion matrix* dengan nilai hasil data uji *True Positive* (TP) sebanyak 13.398 data dan *True Negative* (TN) sebanyak 10.060 data, Sedangkan nilai *False Negative* (FN) sebesar 0 data dan *False Positive* (FP) sebanyak 4 data.



Gambar 5. Hasil Pengujian Menggunakan Confusion Matrix

Bulletin of Data Science
ISSN 2745-7966 (Media Online)
Vol 5, No 3, June 2026 | Page 128−137
DOI: 10.47065/bulletinds.v5i3.10079
https://ejurnal.seminar-id.com/index.php/bulletinds

3.5 Pembahasan

Berdasarkan pengujian yang dilakukan, tingginya performa model klasifikasi sangat bergantung pada keberhasilan tahapan pra-pemrosesan data menggunakan KDD. Reduksi dimensi data dari 56 atribut menjadi 51 atribut fungsional dapat meringankan beban komputasi. Proses normalisasi data menggunakan *Min-Max Scaler* memastikan rentang nilai pada 50 atribut prediktor berada pada skala yang seimbang, yakni 0 hingga 1. Penyeimbangan ini berperan penting saat data masuk ke dalam algoritma BPNN, karena mencegah variabel dengan nilai besar mendominasi dan mengacaukan perhitungan pembaruan bobot selama proses pelatihan jaringan.

Setelah data siap dan dinormalisasi, proses klasifikasi masuk ke tahap propagasi maju. Pengujian membuktikan bahwa konfigurasi 50 neuron pada *input layer*, 75 neuron pada *hidden layer*, dan 1 neuron pada *output layer* (50-75-1) adalah arsitektur yang paling optimal dalam mengenali pola. Pada *hidden layer*, penerapan fungsi aktivasi ReLU secara nyata mempercepat komputasi. Fungsi ini bekerja secara efisien dengan mengabaikan sinyal bernilai negatif menjadi nol, sehingga jaringan saraf bisa lebih fokus memproses fitur-fitur yang memiliki indikasi matematis kuat terhadap skema *phishing*. Sinyal kemudian diteruskan ke *output layer* menggunakan fungsi aktivasi *sigmoid biner*, yang menekan estimasi akhir ke dalam rentang probabilitas untuk menentukan kelas klasifikasi akhir.

Pada proses pembelajaran, model diawal menghasilkan klasifikasi yang berbeda dari target aktual. Di sinilah metode klasifikasi bekerja yakni pada tahap propagasi mundur. Sistem secara otomatis menghitung selisih tingkat kesalahan (*error gradient*) antara keluaran yang dihasilkan dengan label data asli. Perbedaan hasil dialirkan secara terbalik dari lapisan *output* hingga lapisan *input* untuk memperbaiki atau memodifikasi bobot matematis dan nilai bias. Berdasarkan hasil pengujian dengan *learning rate* sebesar 0.1 selama 100 *epoch*, memberikan titik konvergensi yang stabil. Angka *learning rate* sangat pas dan tidak terlalu besar yang berisiko membuat model melompati akurasi optimal, dan tidak terlalu kecil untuk menghambat kecepatan pemrosesan data uji.

Efektivitas dari penyesuaian bobot melalui propagasi terlihat sangat jelas pada hasil evaluasi *confusion matrix* pada data latih dan data uji 90:10. Model memvalidasi ketepatannya dengan mendeteksi 13.398 data *True Positive* dan 10.060 data *True Negative*. Indikator yang paling mengesankan dari evaluasi ini adalah nilai *False Negative* yang berada di angka nol, yang dapat mendorong nilai *recall* hingga menyentuh angka 100%. Kondisi ini membuktikan bahwa tidak ada satupun tautan *phishing* yang berhasil masuk ke sistem dan lolos sebagai tautan yang aman.

Secara keseluruhan, integrasi pemrosesan fitur KDD ke dalam algoritma BPNN berhasil memecahkan tantangan klasik dari model konvensional, yakni ketidakstabilan saat dihadapkan pada dataset berskala besar. Pendekatan ini tidak sekadar menebak kelas data berdasarkan statistik dasar, melainkan membedah pola *non-linear* pada struktur sintaksis setiap URL secara mendalam melalui penyesuaian *error* yang berulang-ulang. Hasilnya, klasifikasi serangan *phishing URL* ini mampu bekerja dengan performa presisi yang sangat tinggi, stabil terhadap variasi manipulasi tautan baru, namun tetap memiliki beban komputasi yang tergolong ringan dan efisien.

4. KESIMPULAN

Penelitian ini berhasil menerapkan metode BPNN untuk mengklasifikasikan serangan *phishing URL*, yakni kejahatan siber yang bertujuan mencuri informasi sensitif pengguna melalui manipulasi tautan. Pendekatan ini diusulkan untuk mengatasi kelemahan model terdahulu, khususnya terkait batasan ukuran dataset dan efisiensi komputasi, dengan memanfaatkan dataset berskala besar PhiUSIIL-2024 dari Kaggle agar model mampu mempelajari pola *non-linear* secara mendalam. Tahapan KDD diterapkan secara terstruktur, mencakup seleksi fitur yang mereduksi 56 menjadi 51 atribut, pembersihan data menghasilkan 234.611 baris data valid, dan proses normalisasi menggunakan *Min-Max Scaler* guna menyeimbangkan skala matematis. Model BPNN divalidasi melalui metode *split validation* dengan rasio 70:30, 80:20, dan 90:10, serta penyesuaian jumlah neuron *hidden layer*, *learning rate*, dan batasan *epoch*. Hasil pengujian komputasi menunjukkan hasil performa terbaik secara keseluruhan dicapai oleh arsitektur 50 *input neuron*, 75 *hidden neuron*, dan 1 *output neuron* (50-75-1) dengan rasio 90:10, *learning rate* 0,1, dan 100 *epoch*. Skenario ini berhasil mencatatkan akurasi 99.982%, dengan tingkat presisi 99.970%, *recall* 100%, serta nilai *F1-Score* sebesar 99.985%. Sebagai bahan perbandingan yang relevan, penelitian serupa juga dilakukan oleh Hasibuan dkk. yang mengevaluasi algoritma *Random Forest (RF)* berbasis *feature importance* menggunakan dataset yang sama, menunjukkan metrik performa yang mendekati atau bahkan mencapai 100% pada *accuracy*, *precision*, *recall*, dan *F1-score*. Penelitian tersebut membuktikan bahwa algoritma RF yang dioptimalkan dengan memangkas atribut menggunakan *RF-Top 30 configurations* yang mampu mempertahankan klasifikasi yang sangat stabil dan setara dengan penggunaan fitur penuh. Dengan demikian, dapat disimpulkan bahwa kedua pendekatan komputasi ini terbukti sama-sama sangat adaptif dan memiliki akurasi tinggi, metode BPNN sangat direkomendasikan untuk menelusuri kedalaman pola *non-linear* pada keseluruhan atribut URL secara utuh, sementara algoritma RF menjadi alternatif yang sangat efisien secara komputasi untuk menyeleksi fitur-fitur yang paling krusial.

REFERENCES

[1] M. W. A. Prastya, M. Tahir, A. A. Ningrum, and A. P. Zaibintoro, "Analisis Ancaman Pishing melalui Aplikasi WhatsApp : Review Metode Studi Literatur," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 7, no. 3, pp. 190–197, 2024, doi: 10.32672/jnkti.v7i3.7551.

[2] R. K. Sujiwana, A. Fahmi, A. Ridho, D. C. Aryanti, and N. A. Rakhmawati, "Analisis Bibliometrik Mengenai Serangan Phishing dan Whatsapp menggunakan Vosviewer," *Jurnal Infokom.*, vol. 8, no. 1, pp. 101–105, 2024, doi: 10.55886/infokom.v8i1.880.

[3] B. Gumay, A. H. Hendrawan, F. Satrya, F. Kusumah, T. Informatika, and U. I. Khaldun, "Analisis Dampak Ancaman Cybercrime Terhadap Data Mahasiswa Pada Serangan Web Phising Siak UIKA," *Jurnal INFOTECH.*, vol. 10, no. 2, pp. 297–305, 2024, doi: https://doi.org/10.31949/infotech.v10i2.11463.

[4] M. Arif, B. Dewanto, M. Fathurrahman, D. R. Firdaus, and A. Setiawan, "Penipuan Penambah Followers Instagram : Analisis Serangan Phising dan Dampaknya pada Keamanan Data," *Pubmedia Journal of Internet and Software Engineering (PJISE).*, vol. 1, no. 4, pp. 1–11, 2024, doi: https://doi.org/10.47134/pjise.v1i4.2672.

[5] E. Haryadi, D. Wijayanti, E. C. Ramdhani, I. Widyastuti, P. S. Informasi, and P. S. Akuntansi, "Identifikasi Ancaman Keamanan Siber Dari Penyalahgunaan Sumber Daya Tik : Studi Kasus," *Technologia: Jurnal Ilmiah (TJI).*, vol. 14, no. 4, pp. 886–896, 2024, doi: http://dx.doi.org/10.31602/tji.v15i4.16429.

[6] K. Astianingrum, B. N. Yulisasih, and S. A. Putri, "Perilaku Remaja dalam Menggunakan Internet untuk Mengenali dan Mengindari Phishing pada SMA Muhammadiyah Pacitan," *Jurnal Ahsana.*, vol. 2, no. 3, pp. 70–78, 2024, doi: 10.59395/ahsana.v2i3.367.

[7] Y. Wang, W. Ma, H. Xu, Y. Liu, and P. Yin, "applied sciences A Lightweight Multi-View Learning Approach for Phishing Attack Detection Using Transformer with Mixture of Experts," *Applied Sciences.*, vol. 13, no. 13, pp. 1–17, 2023, doi: https://doi.org/10.3390/app13137429.

[8] T. Hidayatullah and U. M. Sukabumi, "Implementasi Algoritma Base-64 Dalam Mengamankan URL (Uniform Resource Locator) Website Layanan Pengaduan," *J. Media Infotama.*, vol. 18, no. 2, pp. 337–343, 2022, doi: https://doi.org/10.37676/jmi.v18i2.2937.

[9] A. D. Sulistyo, B. D. Wicaksono, R. N. Saputra, and R. Ramadhani, "Strategi Penanggulangan Serangan Phishing di Media Sosial," *Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB).*, vol. 55, no. 1, pp. 385–396, 2024.

[10] D. B. Suwarno, M. Hardjianto, F. T. Informasi, U. B. Luhur, and K. J. Selatan, "Deteksi Website Phishing Dari Analisis Url Phishing Website Detection From Url Analysis Using Random Forest Algorithm," *Bit (Fakultas Teknol. Inf. Univ. Budi Luhur).*, vol. 21, no. 2, pp. 145–152, 2024, doi: 10.36080/bit.v21i2.3603.

[11] N. Reza, F. Rozi, S. Z. Fajriyah, R. Maulida, N. Rahmania, and A. Zahra, "DNS Server Berbasis Ubuntu-22.04.1-Eve- NG-5.0.3.105," *J. Informatics Commun. Technol.*, vol. 4, no. 2, pp. 1–8, 2022, doi: 10.52661/j_ict.v4i2.140.

[12] T. Sistem and M. Fahri, "Penerapan Algoritma Random Forest untuk Deteksi Phishing pada Website," *JITSI (Jurnal Ilmiah Teknologi Sistem Informasi).*, vol. 6, no. 2, pp. 186–194, 2025, doi: 10.62527/jitsi.6.2.472.

[13] R. Fauzan, A. V. Vitianingsih, and D. Cahyono, "Penerapan Algoritma Klasifikasi pada Machine Learning untuk Deteksi Phishing," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 5, no.2 April, pp. 531–540, 2025, doi:https://doi.org/10.57152/malcom.v5i2.1968.

[14] M. Vebriani and W. Yustanti, "Klasifikasi Deteksi Link Phising DANA Kaget Menggunakan Metode Support Vector Machine Berbasis Website," *J. Informatics Comput. Sci.*, vol. 06, no. 02, pp. 408–416, 2024, doi: 10.26740/jinacs.v6n02.p408-416.

[15] M. Adipa, A. T. Zy, and M. M. Effendi, "Klasifikasi Email Phishing Menggunakan Algoritma K-Nearest Neighbor," *J. RESTIKOM Ris. Tek. Inform. dan Komput.*, vol. 5, no. 2, pp. 148–157, 2023, doi: 10.52005/restikom.v5i2.152.

[16] M. Hasibuan, R. Abdillah, S. Agustian, and R. M. Candra, "Classification of Phishing URL Attacks Using Random Forest Algorithm Based on Feature Importance," *Building of Informatics, Technology and Science (BITS).*, vol. 8, no. 2, pp. 1–10, 2025, doi: 10.32877/bt.v8i2.3511.

[17] K. Subashini and V. Narmatha, "Detecting Phishing Websites using recent Techniques : A Systematic Literature Review," *ITM Web of Conferences.*, vol. 01008, pp. 1–13, 2023, doi: https://doi.org/10.1051/itmconf/20235701008.

[18] W. Wang, "Using Machine Learning BT - Pro iPhone Development with Swift 5: Design and Manage Top Quality Apps," W. Wang, Ed., Berkeley, CA: Apress, 2019, pp. 399–432. doi: 10.1007/978-1-4842-4944-4_16.

[19] N. T. Khair, I. Afrianty, F. Syafria, E. Budianita, and S. K. Gusti, "Penerapan Information Gain Untuk Seleksi Fitur Pada Klasifikasi Jenis Kelamin Tulang Tengkorak Menggunakan Backpropagation," *Bulletin of Computer Science Research.*, vol. 5, no. 4, pp. 666–678, 2025, doi: 10.47065/bulletincsr.v5i4.637.

[20] M. Waail *et al.*, "Klasifikasi Jenis Kelengkeng Berdasarkan Morfologi Daun Dengan Ekstraksi Ciri RGB , GLCM , dan Bentuk Menggunakan Metode," *J. Apl. Teknol. Inf. dan Manaj.*, vol. 4, no. 2, pp. 183–193, 2023, doi: 10.31102/jatim.v4i2.2341.

[21] M. Kadarman, I. Afrianty, E. Budianita, dan F. Syafria, "Classification of Human Skull Bones on Gender Using Backpropagation in Forensic Anthropology," *Computer Science and Information Technology (CoSciTech).*, vol. 5, no. 3, hal. 619–625, 2025, doi:10.37859/coscitech.v5i3.8235.

[22] S. Widaningsih, S. Yusuf, J. T. Informatika, F. Teknik, and U. Suryakancana, "Penerapan Data Mining Untuk Memprediksi Siswa Berprestasi Dengan Menggunakan Algoritma K Nearest Neighbor," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi).*, vol. 9, no. 3, pp. 2598–2611, 2022, doi: 10.35957/jatisi.v9i3.859.

[23] F. N. S. Inggih Permana, "Pengaruh Normalisasi Data Terhadap Performa Hasil Klasifikasi Algoritma Backpropagation," *Indones. J. Inform. Res. Softw. Eng.*, vol. 2, no. 1, pp. 67–72, 2022, doi:10.57152/ijirse.v2i1.311.

[24] M. Azhima, I. Afrianty, E. Budianita, and S. K. Gusti, "Penerapan Metode Backpropagation Neural Network untuk Klasifikasi Penyakit Stroke," *KLIK: Kajian Ilmiah Informatika dan Komputer.*, vol. 4, no. 6, pp. 3013–3021, 2024, doi: 10.30865/klik.v4i6.1956.