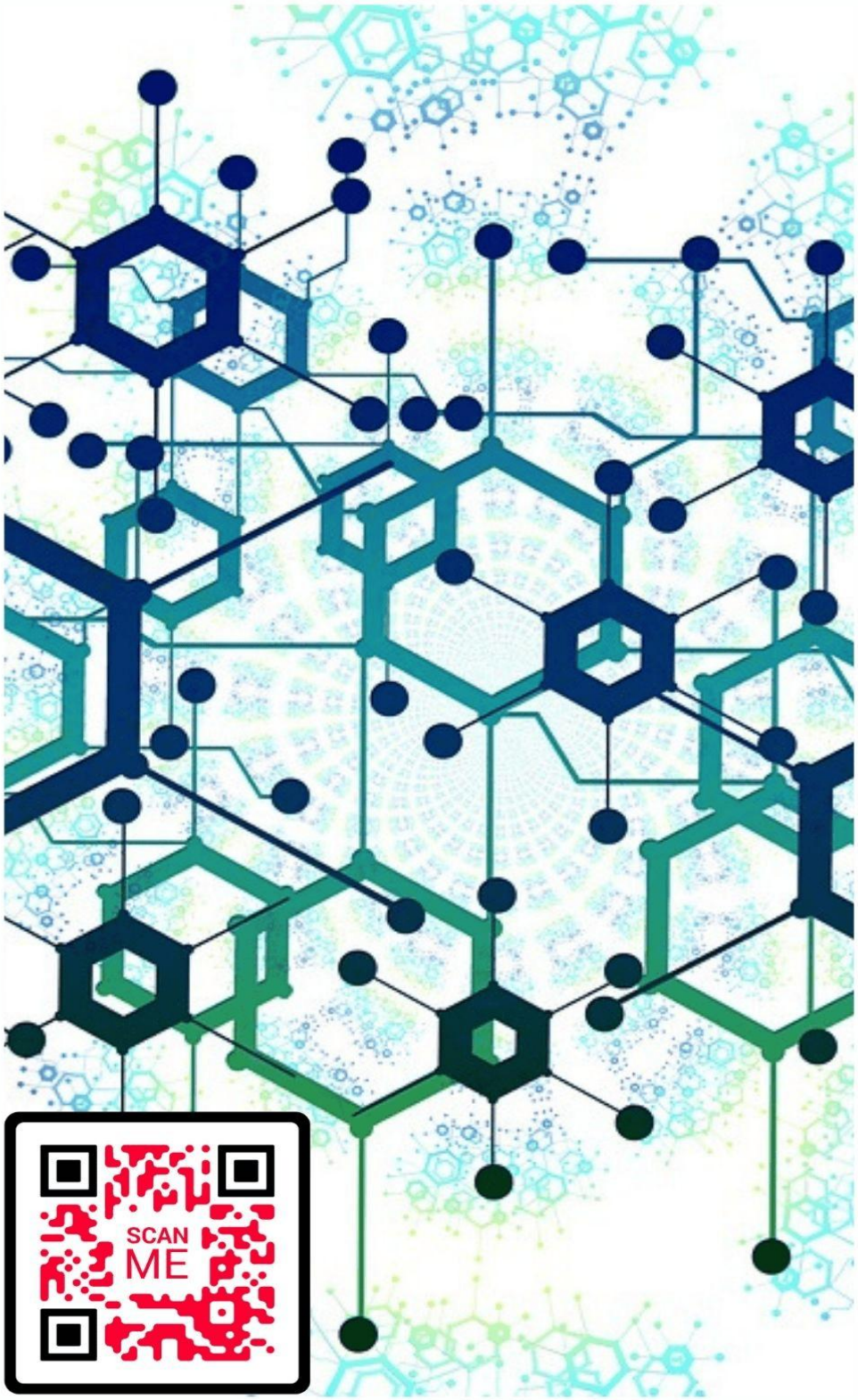
Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

ISSN (ONLINE) 2598-9936



INDONESIAN JOURNAL OF INNOVATION STUDIES

PUBLISHED BY

UNIVERSITAS MUHAMMADIYAH SIDOARJO

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

Table Of Contents

Journal Cover 1
Author[s] Statement 3
Editorial Team 4
Article information 5
Check this article update (crossmark) 5
Check this article impact 5
Cite this article 5
Title page 6
Article Title 6
Author information 6
Abstract 6
Article content 7

Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

Originality Statement

The author[s] declare that this article is their own work and to the best of their knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the published of any other published materials, except where due acknowledgement is made in the article. Any contribution made to the research by others, with whom author[s] have work, is explicitly acknowledged in the article.

Conflict of Interest Statement

The author[s] declare that this article was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright Statement

Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

EDITORIAL TEAM

Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

Article information

Check this article update (crossmark)

Check this article impact (*)

Save this article to Mendeley

(*) Time for indexing process is various, depends on indexing database platform

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

Comparison Of Random Forest and Neural Network for Portable Executable Malware Classification

Perbandingan Random Forest dan Neural Network pada Klasifikasi Malware Portable Executable

Agung Nugroho, 111202214125@mhs.dinus.ac.id, (1)
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Indonesia

Chaerul Umam, chaerul@dsn.dinus.ac.id, ()
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Indonesia

(1) Corresponding author

Abstract

General Background: The rapid growth of information technology has increased the complexity of cyber threats, with malware attacks posing significant risks to computer systems, particularly those based on the Windows operating system. **Specific Background:** Portable Executable files contain structured statistical attributes that can be utilized to distinguish malware from benign software using machine learning techniques. **Knowledge Gap:** Despite extensive use of machine learning in malware detection, comparative evidence using identical Portable Executable statistical features and consistent evaluation settings remains limited. **Aims:** This study aims to compare the classification performance of Random Forest and Neural Network models in malware detection based on Portable Executable statistical features. **Results:** Using the ClaMP Integrated Dataset comprising 5,184 samples and 70 static features, Random Forest achieved an accuracy, precision, recall, and F1-score of 99.14%, while the Neural Network obtained consistent scores of 98.18% across all evaluation metrics. **Novelty:** This research presents a direct and controlled comparison of ensemble and neural-based classifiers using identical preprocessing pipelines, default model configurations, and balanced Portable Executable datasets. **Implications:** The findings demonstrate that ensemble-based approaches provide stable and reliable performance for Portable Executable malware classification and offer a practical foundation for automated machine learning–based cybersecurity systems.

Highlights

- Random Forest Achieved The Highest Classification Scores Across All Metrics
- Portable Executable Statistical Features Provided Clear Malware Separation
- Ensemble Learning Demonstrated Strong Stability On Structured PE Data

Keywords

Malware Detection; Portable Executable; Random Forest; Neural Network; Machine Learning

Published date: 2026-01-11

Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

I. Pendahuluan

Perkembangan teknologi informasi di era digital membawa dampak besar terhadap kehidupan manusia, baik dalam aspek ekonomi, sosial, maupun pertahanan keamanan. Namun, kemajuan teknologi tersebut juga memunculkan berbagai ancaman keamanan siber yang semakin kompleks dan sulit dideteksi [1]. Salah satu ancaman yang paling sering terjadi adalah serangan malware (malicious software), yaitu perangkat lunak berbahaya yang dirancang untuk mengganggu, mencuri, atau merusak sistem komputer [2]. Ancaman siber yang dialami oleh Indonesia mengalamai peninggkatan pada setiap tahunnya. Contohnya pada Juli 2024 terjadi serangan Ransomware yang menjelaskan sistem peringatan dini yang masih sangat lemah terhadap ancaman siber. Hal ini kemudian mengakibatkan, pengurangan penggunaan platform diital yang dilakukan oleh pemerintah maupun masyarakat dalam beberapa hari seperti yang disampaikan dalam CNN Indonesia (2024). Kaspersky (2018) menjelaskan bahwa serangan WannaCry pada tahun 2017 menunjukkan bagaimana malware dapat menyebar dengan cepat, mengenkripsi data pengguna, dan meminta tebusan digital di seluruh dunia. Kedua peristiwa tersebut menunjukkan pentingnya pengembangan sistem deteksi malware yang cepat, akurat, dan adaptif.

Sebagian besar malware modern menargetkan sistem operasi Windows, karena dominasi penggunaannya di seluruh dunia. File eksekusi pada Windows menggunakan format Portable Executable (PE), yaitu format standar yang berisi struktur data penting seperti headers, sections, dan entry points yang digunakan oleh sistem operasi untuk menjalankan program [3]. Terdapat bagian yang dapat membedakan antara file yang tidak berbahaya dan file malware dengan melihat struktur setiap file PE dan bagaimana bagian-bagiannya tersusun. Ada dua cara utama untuk menemukan malware: analisis dinamis dan analisis statistik. Ketika melakukan analisis, Anda menjalankan malware di lingkungan terkontrol (sandbox) untuk melihat bagaimana perilakunya. Ketika melakukan analisis statistik, Anda tidak menjalankan file tersebut; melainkan, Anda melihat struktur internal dan nilai atributnya [4]. Studi Breiman [5] menunjukkan bahwa Random Forest berfungsi dengan baik dalam metode bagging dan pemilihan fitur acak untuk mengurangi overfitting. Hal ini menghasilkan hasil yang lebih stabil. Sebaliknya, jaringan saraf (NN) memiliki kemampuan untuk mengidentifikasi hubungan non-linier yang kompleks antara fitur data. Menurut Hukubun [6], teknik backpropagation dan mekanisme propagasi feedforward membuat klasifikasi lebih akurat. Menurut studi Efriyani dan Panjaitan [2], model pembelajaran mendalam—khususnya Jaringan Neural Berulang (RNN) mampu membedakan malware dengan akurasi hingga 86%.

Namun demikian, masing-masing algoritma memiliki kelebihan dan kelemahannya. Random Forest dikenal cepat, stabil, dan mudah diinterpretasikan, sementara Neural Network unggul dalam mempelajari pola non-linear namun rawan overfitting jika tidak dilakukan regularisasi yang baik. Tujuan penelitian ini adalah untuk menilai efisiensi dua pendekatan untuk mendeteksi malware menggunakan sifat statistik dari Portable Executables (PE). Kami menggunakan dataset ClaMP Integrated, yang mencakup lebih dari 5.000 sampel file PE yang diklasifikasikan sebagai malware atau bukan. Tujuan penelitian ini adalah untuk menemukan strategi terbaik yang berhasil menggunakan pengujian seperti F1-score, akurasi, presisi, dan memori. Temuan penelitian ini diharapkan dapat berkontribusi pada pengembangan sistem deteksi malware berbasis pembelajaran mesin di Indonesia. Pendekatan ini bermanfaat baik untuk penelitian maupun untuk membuat seluruh negara lebih aman secara online.

II. Metode

Penelitian ini menggunakan pendekatan kuantitatif eksperimental. Sesuai dengan namanya, desain penelitian ini berfokus pada melakukan uji coba (eksperimen) secara langsung untuk mengukur dan membandingkan dampak dari suatu perlakuan terhadap variabel hasil [7]. Dalam konteks ini, "perlakuan" yang diberikan adalah penerapan dua algoritma machine learning yang berbeda, yaitu Random Forest (RF) dan Neural Network (Multi-Layer Perceptron, MLP). Seberapa baik setiap program dapat menemukan file berbahaya didasarkan pada variabel hasil. Tujuan utama proyek ini adalah untuk melihat beberapa statistik tentang file Portable Executable (PE). Tanpa menjalankan program, analisis statistik melihat tata letak file biner, seperti header, bagian, dan impor pustaka. Hal ini mengurangi kemungkinan bug masuk ke dalam analisis mesin. Berdasarkan ciri-ciri statistik ini saja, model akan dilatih untuk mengenali tren yang membedakan file malware dari file yang baik.

Tahapan penelitian ini dirancang secara sistematis mengikuti prinsip machine learning pipeline (alur kerja machine learning) yang standar. Pendekatan ini memastikan bahwa proses, mulai dari akuisisi data mentah hingga evaluasi model akhir, dapat direplikasi dan divaDalidasi [8]. Setiap langkah dalam alur ini memiliki tujuan spesifik yang berkontribusi pada hasil akhir penelitian. Secara umum, alur penelitian yang akan dilaksanakan dapat dilihat pada Gambar 1 berikut.

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

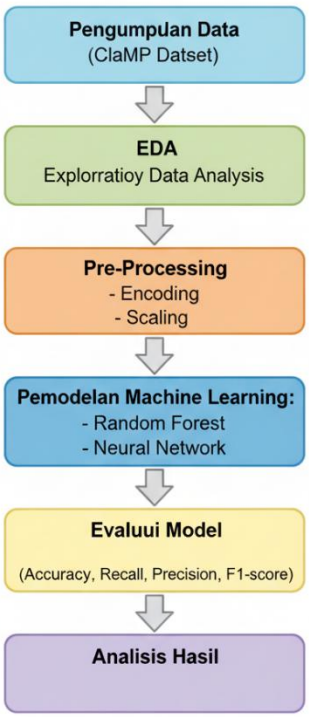


Figure 1. *Alur Penelitian*

Dataset yang digunakan adalah ClaMP Integrated Dataset, sebuah dataset publik yang banyak digunakan dalam penelitian deteksi malware berbasis file PE (Portable Executable). Dataset ini berisi 5.184 entri data, yang terbagi menjadi dua kelas utama:

1. 2.501 file benign (tidak berbahaya)
2. 2.683 file malware (berbahaya)

Setiap entri data memiliki sekitar 70 atribut fitur statis, yang diekstraksi dari struktur PE, atribut fitur tersebut dapat dilihat pada table 3.1 meliputi:

Kategori Fitur	Contoh Atribut	Keterangan
Header	SizeOfImage, CheckSum, NumberOfSections, AddressOfEntryPoint	Metadata utama struktur PE
Section	EntropySection1, RawDataSize, VirtualSize	Menunjukkan distribusi dan ukuran data pada setiap bagian file
Optional Header	Subsystem, DllCharacteristics, SizeOfCode	Parameter eksekusi file
Import Table	NumberOfImports, DllImportEntropy	Jumlah pustaka eksternal yang digunakan file

Table 1. *Dataset Penelitian*

Dataset ClaMP ini dipilih karena representatif untuk menganalisis malware berbasis Windows, memiliki format CSV, dan telah dibersihkan dari duplikasi.

Adapun algoritma yang digunakan ialah sebagai berikut:

1. Random Forest

Random Forest merupakan algoritma ensemble learning yang menggabungkan beberapa decision tree untuk membentuk model yang lebih kuat [9]. Setiap pohon dilatih menggunakan subset data dan subset fitur acak [4]. Kemampuan Random

Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

Forest untuk menghasilkan prediksi yang konsisten tanpa overfitting adalah manfaat utamanya. Berikut adalah rumus prediksi RF fundamental:

$$\boldsymbol{f(x) = mode\,\{h_1(x), h_2(x), \ldots, h_k(x)\}}$$

Model RF dalam penelitian ini dibangun menggunakan fungsi Random Forest Classifier() dari pustaka Scikit-learn [10]. Keseimbangan antara akurasi dan risiko overfitting, serta pengulangan penelitian, semuanya dipengaruhi oleh pemilihan hyperparameter. Parameter yang digunakan dalam penelitian ini dapat dilihat pada Tabel 2:

Parameter	Nilai	Deskripsi
n_estimators	200	Jumlah pohon dalam hutan
max_depth	20	Kedalaman maksimum setiap pohon
min_samples_split	2	Jumlah minimum sampel untuk pemisahan node
random_state	42	Reproducibility

Table 2. *Parameter Random Forest*

2. Neural Network

Model kedua yang dievaluasi adalah Neural Network dengan arsitektur Multi-Layer Perceptron (MLP). Model ini dipilih karena kemampuannya untuk menjelaskan interrelasi non-linier yang rumit di antara fitur-fitur [11]. Arsitektur MLP memiliki satu lapisan input dengan 70 neuron, yang didasarkan pada jumlah fitur PE. Terdapat juga tiga lapisan tersembunyi yang meruncing dengan 128, 64, dan 32 neuron, dan satu lapisan output dengan satu neuron untuk klasifikasi biner malware–benign. Fungsi aktivasi ReLU digunakan di setiap lapisan tersembunyi karena stabil dan bekerja dengan baik selama pelatihan. Lapisan output menggunakan Sigmoid untuk mengetahui peluang klasifikasi [12]. Proses optimasi bobot dilakukan dengan Adam sebagai optimizer yang dikenal cepat dan stabil, sementara loss function yang digunakan adalah binary cross-entropy sebagai standar untuk klasifikasi biner. Arsitektur ini dirancang untuk memungkinkan pembelajaran pola secara hierarkis dan meningkatkan akurasi deteksi.

Parameter	Nilai	Deskripsi
hidden_layer_sizes	(128, 64, 32)	Jumlah neuron per lapisan tersembunyi
activation	'relu'	Fungsi aktivasi antar neuron
solver	'adam'	Optimizer adaptif berbasis momentum
learning_rate_init	0.001	Laju pembelajaran awal
max_iter	500	Iterasi maksimum pelatihan

Table 3. *Parameter Neural Network*

Proses pelatihan dilakukan dengan fungsi MLPClassifier() dari pustaka Scikit-learn. Model divalidasi menggunakan data pengujian yang sama dengan Random Forest.

Evaluasi kinerja Random Forest dan Neural Network dilakukan menggunakan empat metrik utama yang dihitung berdasarkan Confusion Matrix, yaitu True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Keempat komponen ini menggambarkan kemampuan model dalam mengklasifikasikan file malware maupun benign, sekaligus menunjukkan jenis kesalahan yang terjadi. Berdasarkan nilai TP, TN, FP, dan FN tersebut, metrik akurasi, presisi, recall, dan F1-score dihitung untuk memberikan gambaran komprehensif mengenai tingkat ketepatan, kemampuan deteksi, serta keseimbangan performa kedua model dalam skenario deteksi malware [13], [14], [15].

a. Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Pada dataset yang tidak seimbang, akurasi dapat sangat menyimpang (menyesatkan) [16]. Misalnya, jika 99% file aman, model yang secara konsisten memprediksi "tidak berbahaya" akan 99% benar tetapi akan gagal total dalam mendeteksi malware.

Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

b. Precision

$$Precicion = \frac{TP}{TP + FP}$$

Untuk mencegah False Positives (FPA), ketelitian yang tinggi sangatlah penting. Di dunia nyata, tingkat false positive yang tinggi menunjukkan bahwa sistem antivirus akan selalu menolak file yang bersih dan asli. Hal ini sangat membuat frustrasi pengguna dan dapat menghentikan aktivitas penting perusahaan.

c. Recall

$$Recall \frac{TP}{TP + FN}$$

Metrik keamanan yang paling krusial disebut dengan recall.

d. F1-Score

$$F1 - Score = 2 \times \frac{Precicion \times Recall}{Precicion + Recall}$$

F1-Score akan bernilai tinggi hanya jika Precision dan Recall keduanya tinggi. Ini mencegah situasi di mana sebuah model "curang" dengan mengorbankan salah satu metrik (misalnya, mencapai Recall 100% dengan Presisi 10% dengan cara melabeli semua file sebagai malware).

Penelitian ini menggunakan kombinasi perangkat lunak dan perangkat keras untuk mendukung proses pengolahan data, pelatihan model, serta analisis hasil. Python 3.10 dan lingkungan komputasi utama Google Colab digunakan. Memiliki akses ke GPU dan TPU memudahkan pelatihan model, terutama jaringan saraf. Inilah alasan mengapa pengaturan ini digunakan. Scikit-learn menjalankan algoritma dan mengontrol model. Pandas dan NumPy memproses data, dan Matplotlib serta Seaborn menampilkannya. Dalam hal ini, Anda bisa mendapatkan laptop Windows 11 dengan CPU AMD Ryzen 7, RAM 16GB, dan SSD 512GB. Perangkat ini digunakan untuk mempersiapkan data, mengkodekannya, dan menghasilkan output. Dengan menggunakan metode yang disebut stratified sampling, dataset dibagi menjadi 80% data pelatihan dan 20% data uji. Hal ini memastikan bahwa setiap kelompok memiliki jumlah kelas berbahaya dan aman yang sama. Karena terdapat banyak kelompok dan informasinya sangat besar, metode pelatihan hanya dijalankan sekali. Seberapa baik kinerja Anda dapat dinilai dari F1-Score, Akurasi, Presisi, dan Recall. Ketiganya menunjukkan seberapa baik model dapat melakukan hal-hal seperti menemukan dan mengurutkannya. Untuk menjamin kinerja yang konsisten, Anda dapat melakukan validasi tambahan dengan mengeksekusi percobaan acak dengan nilai random_state yang bervariasi dan menggunakan teknik K-Fold Cross Validation. Pendekatan ini memastikan model memiliki tingkat generalisasi dan konsistensi yang baik pada skenario deteksi malware di dunia nyata.

III. Hasil dan Pembahasan

Seluruh eksperimen, dari pelatihan dan evaluasi model pembelajaran mesin hingga pra-pemrosesan data, dilakukan menggunakan Google Colaboratory Cloud. Platform ini membutuhkan akses ke sumber daya komputasi berkinerja tinggi, seperti GPU. Sumber daya ini memungkinkan penyelesaian tugas komputasi kompleks dengan cepat dan tanpa memperhatikan keterbatasan teknologi saat ini. Analisis data pertama, manajemen notebook Colab, dan penulisan skrip semuanya dilakukan pada laptop lokal dengan CPU AMD Ryzen 7 dan RAM 16GB. Konfigurasi ini memudahkan pengorganisasian eksperimen menggunakan Google Colab. Studi ini menggunakan Python dan pustaka penting seperti Scikit-learn untuk pemodelan pembelajaran mesin, penskalaan fitur, pengumpulan data partisipan, dan evaluasi kinerja; Pandas untuk pemuatan dan manipulasi data; NumPy untuk perhitungan numerik; dan Matplotlib dan Seaborn untuk visualisasi, yang mencakup analisis distribusi fitur dan matriks noise. Dataset ClaMP gabungan berisi 5.184 file Portable Executable (PE) biner yang diklasifikasikan sebagai aman atau berbahaya. Masing-masing file memiliki 70 fitur numerik yang diambil dari karakteristik statistik file seperti SizeOfImage, NumberOfSections, dan EntropySection. Pengambilan sampel bertingkat membagi dataset menjadi dua puluh persen data pelatihan dan dua puluh persen data uji. Ini menghasilkan model evaluasi yang dapat diandalkan dan menjaga keseimbangan antara kategori malware dan kategori yang tidak berbahaya.

Tampilan pertama pada Dataset Terintegrasi ClaMP menunjukkan bahwa distribusi label kelas sangat seimbang. Sebanyak 5.184 sampel dianalisis, terdiri dari 2.683 sampel malware (sekitar 52%) dan 2.501 sampel aman (48%). Distribusi yang hampir merata ini optimal untuk klasifikasi biner karena memberikan setiap kelas kesempatan yang adil untuk representasi, mengurangi bias model terhadap satu kelas, dan menjamin bahwa metrik penilaian seperti akurasi akurat dan adil. Distribusi yang konsisten seperti yang digambarkan pada Gambar 2 menjamin proses pelatihan model yang lancar, menghilangkan kebutuhan akan teknik penyeimbangan data tambahan seperti oversampling atau undersampling.

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

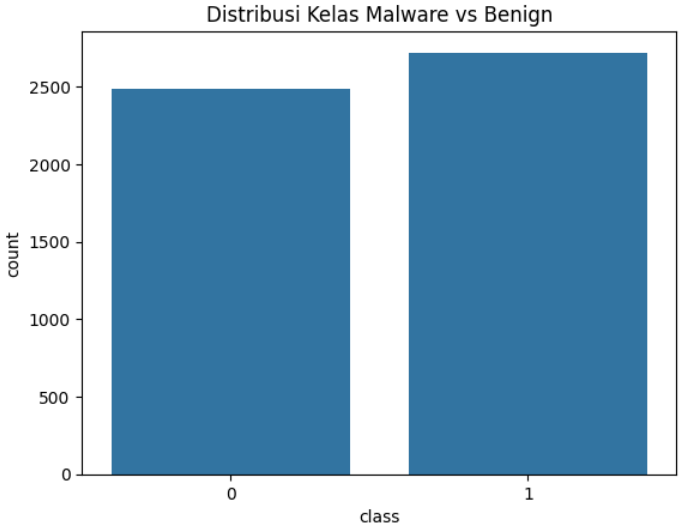


Figure 2. *Distribusi Kelas Malware vs Benign*

Analisis korelasi Pearson digunakan untuk menilai hubungan setiap fitur dengan label kelas, sehingga fitur yang paling relevan dapat diidentifikasi sebelum pemodelan. Hasilnya, sebagaimana ditampilkan pada Gambar 2, menunjukkan variasi koefisien korelasi positif maupun negatif terhadap kelas malware dan benign. Selain itu, matriks korelasi pada Gambar 3 memperlihatkan tingkat kedekatan antar fitur, dengan warna merah menunjukkan korelasi yang lebih tinggi.

Berdasarkan grafik di bawah ini, terlihat bahwa beberapa fitur memiliki korelasi positif yang kuat terhadap kelas malware terutama atribut File Header seperti FH_char0, FH_char3, e_file, serta fitur struktural seperti ImageBase dan SizeOfStackReserve yang menunjukkan bahwa nilai-nilai tertentu pada header dan alokasi memori cenderung menjadi karakteristik file berbahaya. Sebaliknya, sejumlah fitur menunjukkan korelasi negatif yang kuat, terutama flag DLL pada Optional Header (misalnya OH_DLLChar10, OH_DLLChar6, OH_DLLChar5) serta SectionAlignment, yang mengindikasikan pola khas file benign yang lebih stabil dan standar. Temuan ini selaras dengan literatur malware analysis dan menegaskan bahwa variasi pada header PE merupakan indikator penting. Walaupun sebagian fitur menunjukkan korelasi lemah, seluruh 70 fitur tetap digunakan karena baik Random Forest maupun Neural Network mampu menangkap hubungan non-linear dan interaksi antarfitur, sehingga fitur yang lemah secara individu masih berpotensi menjadi prediktif ketika dikombinasikan dalam model.

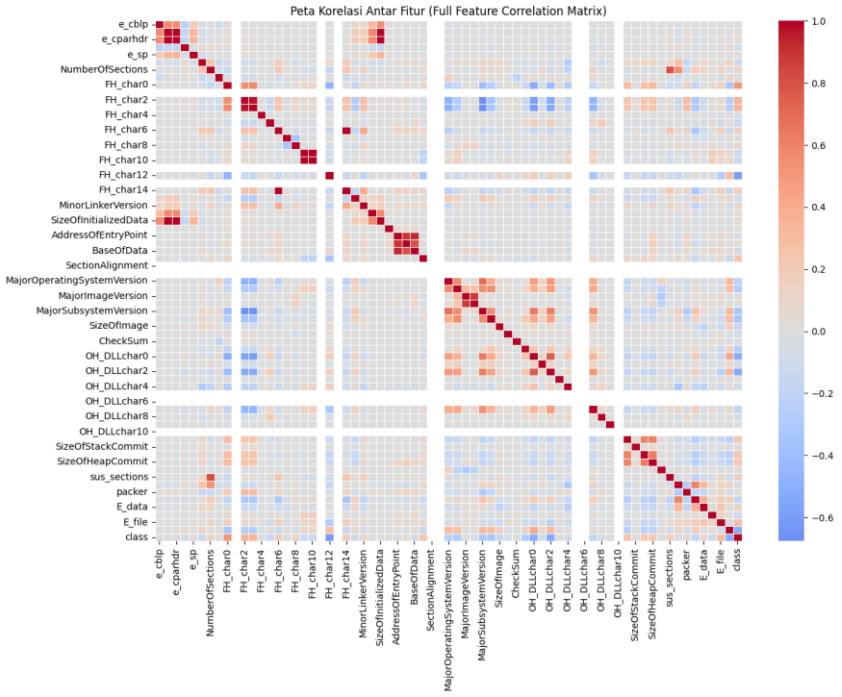


Figure 3. *Peta Korelasi Antar Fitur*

Tahap preprocessing dilakukan untuk memastikan data dalam kondisi seragam dan siap diproses oleh model machine

Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

learning. ClaMP Integrated Dataset yang berisi 5.184 sampel dengan 70 fitur statis hasil ekstraksi struktur Portable Executable (PE) diperiksa terlebih dahulu untuk menilai kelengkapan datanya. Audit terhadap seluruh atribut meliputi header, section properties, entropy, dan import functions menunjukkan bahwa tidak terdapat nilai kosong pada satu pun kolom, sehingga seluruh fitur memiliki tingkat kelengkapan 100%. Hasil ini, yang diringkas pada Tabel 4, memastikan bahwa proses pemodelan dapat dilakukan tanpa perlu penanganan nilai hilang seperti imputasi.

Kategori Pemeriksaan	Hasil
Total Sampel Data	5.184
Total Fitur	70
Total Nilai (Sel) Diperiksa	362.880
Total Nilai Kosong (Missing) Ditemukan	0
Persentase Kelengkapan Data	100%

Table 4. *Pemeriksaan Nilai Kosong*

Setelah memastikan dataset bebas dari nilai kosong, tahap pra-pemrosesan dilanjutkan dengan verifikasi tipe data, di mana hasil pemeriksaan menunjukkan bahwa seluruh 70 fitur dan kolom target telah berformat numerik sehingga tidak memerlukan proses encoding kategorikal. Dengan menggunakan pengambilan sampel bertingkat, dataset dibagi menjadi 20% data uji dan 80% data pelatihan untuk menjaga rasio malware terhadap kelas yang tidak berbahaya. Kami menggunakan random_state sebesar 42 untuk memastikan hasilnya dapat diulang. StandardScaler digunakan untuk menstandarisasi semua fitur agar model bekerja lebih baik, terutama untuk jaringan saraf, yang sangat sensitif terhadap penskalaan fitur. Kami hanya menggunakan scaler ini pada data pelatihan, dan untuk mencegah kebocoran data, kami melakukan hal yang sama pada data uji. X_train_scaled dan X_test_scaled dibuat selama pra-pemrosesan untuk digunakan dalam pelatihan dan pengujian model.

Setelah semua pra-pemrosesan selesai, model Random Forest dan Neural Network dilatih pada data pelatihan reguler (X_train_scaled dan y_train). Kami melakukan ini dengan pengaturan default Scikit-learn sehingga kami dapat melihat seberapa baik kinerjanya tanpa penyetelan apa pun. Kami menggunakan RandomForestClassifier untuk menguji model pertama. Untuk menjamin pengulangan, parameter default adalah 200 pohon (n_estimators=200), kriteria Gini, kedalaman regularisasi 20, dan keadaan acak 42. Manfaat teknik bagging, yang melatih beberapa pohon secara paralel dan mempercepat komputasi, ditunjukkan oleh proses pelatihan yang sangat efektif pada 4.147 sampel, yang membutuhkan waktu sekitar tiga detik untuk diselesaikan di Google Colab. Berdasarkan 70 karakteristik statistik PE, data dasar ini digunakan untuk membandingkan seberapa baik setiap model mendeteksi malware. Setelah dilatih, model RF segera dievaluasi menggunakan 1.037 sampel data pengujian (X_test_scaled) yang belum pernah dilihat sebelumnya. Hasil evaluasi performa baseline dari model Random Forest ditunjukkan pada Tabel 5 di bawah ini.

Kelas	Precision	Recall	F1-Score	Support (Jumlah Sampel)
0 (Benign)	0.9939	0.9880	0.9909	498
1 (Malware)	0.9890	0.9945	0.9918	544
Akurasi Keseluruhan			0.9914	1042
Rata-rata (Macro Avg)	0.9915	0.9912	0.9913	1042
Rata-rata (Weighted Avg)	0.9914	0.9914	0.9914	1042

Table 5. *Hasil Evaluasi Random Forest*

Analisis Confusion Matrix pada Gambar 5 menunjukkan bahwa model Random Forest memberikan kinerja klasifikasi yang sangat akurat, dengan 541 file malware dan 492 file benign berhasil diprediksi dengan benar dari total 1.042 data uji, serta hanya menghasilkan 11 kesalahan klasifikasi. Nilai dominasi pada diagonal utama menunjukkan bahwa sistem menemukan file berbahaya (TP) dan file aman (TN) dengan jumlah false positive (FP) dan false negative (FN) paling sedikit. Hasil ini sejalan dengan metrik evaluasi sebelumnya, yang menunjukkan bahwa skor F1 untuk presisi, recall, dan konsistensi semuanya mencapai 0,99. Ini berarti Random Forest adalah cara yang bagus untuk menemukan malware berdasarkan fitur statistik PE.

ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

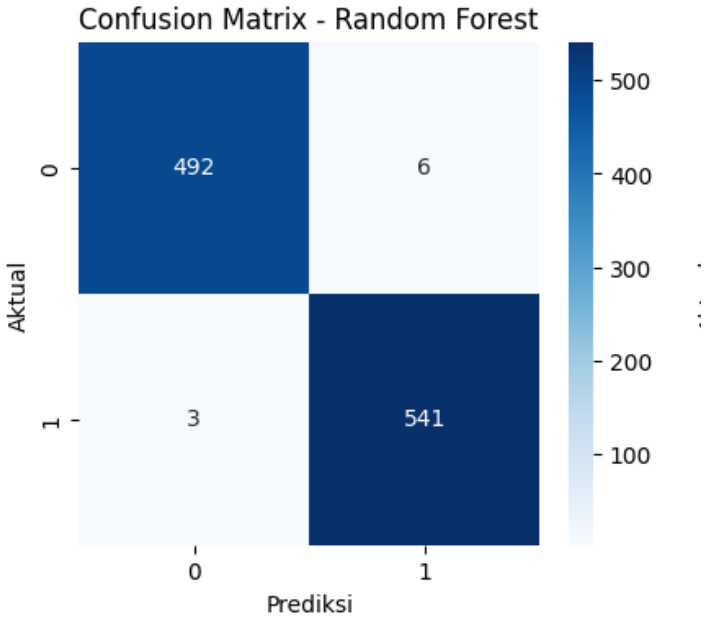


Figure 4. *Confusion Matrix – Random Forest*

Model kedua yang diuji adalah Neural Network berbasis MLPClassifier, menggunakan arsitektur tiga hidden layer berstruktur corong (128–64–32) dengan fungsi aktivasi ReLU, solver Adam, learning rate 0.001, max_iter 500, dan random_state 42 untuk memastikan hasil yang konsisten. Model ini dilatih menggunakan data yang telah distandarisasi, karena Neural Network sangat sensitif terhadap perbedaan skala fitur dan membutuhkan input yang seragam agar proses optimasi berjalan stabil. Proses pelatihan menggunakan X_train_scaled memungkinkan model mempelajari pola non-linear dan representasi fitur yang lebih kompleks, sehingga performanya dapat dibandingkan secara adil dengan Random Forest pada tahap evaluasi berikutnya.Setelah dilatih, performa model dievaluasi pada X_test_scaled. Hasil metrik performa rinci disajikan pada Tabel 6.

Kelas	Precision	Recall	F1-Score	Support (Jumlah Sampel)
0 (Benign)	0.9838	0.9779	0.9809	498
1 (Malware)	0.9799	0.9853	0.9826	544
Akurasi Keseluruhan			0.9818	1042
Rata-rata (Macro Avg)	0.9819	0.9816	0.9817	1042
Rata-rata (Weighted Avg)	0.9818	0.9818	0.9818	1042

Table 6. *Hasil Metrik Performa Rinci*

Berdasarkan Confusion Matrix pada Gambar 6, model Neural Network mampu mengklasifikasikan 487 file benign dan 536 file malware dengan benar, menghasilkan total 19 kesalahan klasifikasi yang sebagian besar berupa false negative, yaitu malware yang tidak terdeteksi. Meskipun performanya berada sedikit di bawah Random Forest, model ini tetap menunjukkan akurasi yang tinggi, yaitu sekitar 98%, dengan keseimbangan precision dan recall yang kuat. Dominasi nilai pada diagonal utama matriks menunjukkan bahwa MLP mampu mengenali pola non-linear pada fitur PE secara efektif dan tetap menjadi model yang kompetitif dalam skenario deteksi malware.

Indonesian Journal of Innovation Studies
Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

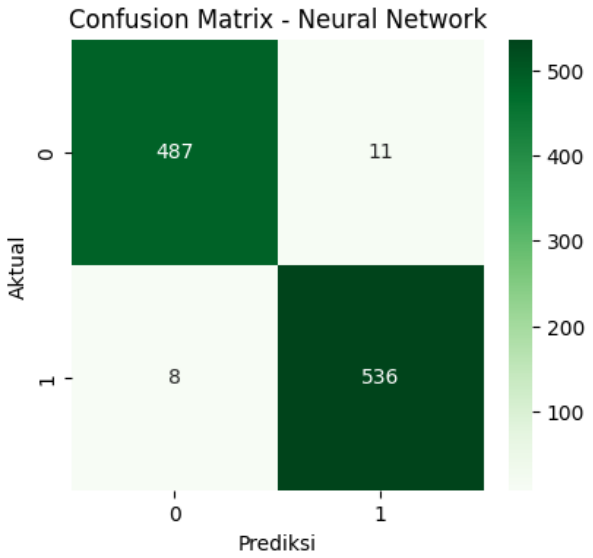


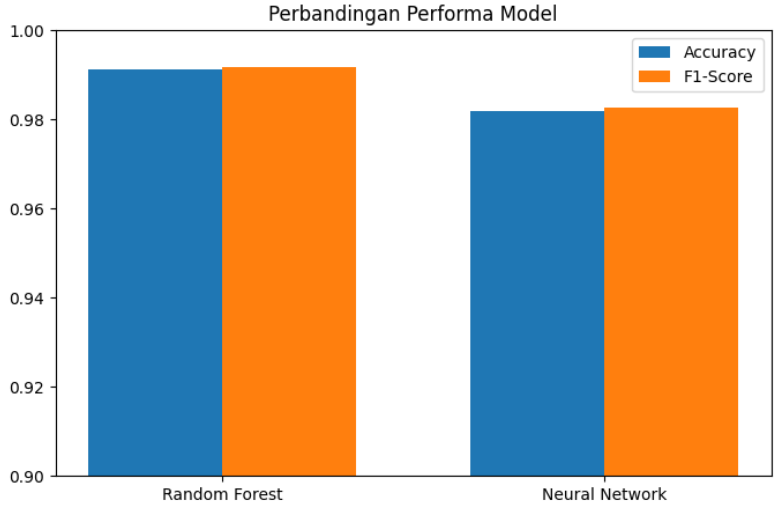
Figure 5. *Confusion Matrix – Neural Network*

Setelah kedua model dievaluasi secara terpisah, dilakukan analisis komparatif untuk menentukan model yang paling unggul dalam mendeteksi malware. Perbandingan performa keduanya disajikan pada Tabel 4.4 menggunakan metrik weighted average dari classification report, sehingga setiap nilai akurasi, presisi, recall, dan F1-score dihitung dengan mempertimbangkan jumlah sampel pada masing-masing kelas. Pendekatan ini memastikan bahwa perbandingan berlangsung adil dan representatif, terutama karena distribusi kelas malware dan benign yang seimbang dalam dataset.

Algoritma	Accuracy	Precision (Weighted Avg)	Recall (Weighted Avg)	F1-Score (Weighted Avg)
Random Forest (RF)	0.9914	0.9914	0.9914	0.9914
Neural Network (MLP)	0.9818	0.9818	0.9818	0.9818

Table 7. *Data Perbandingan*

Berdasarkan hasil perbandingan pada Tabel 7, Random Forest menunjukkan performa paling unggul dengan skor konsisten 99.14% pada seluruh metrik, mengindikasikan tingkat akurasi, presisi, recall, dan F1-score yang sangat tinggi dan stabil. Neural Network juga memberikan hasil yang kuat dengan skor 98.18%, hanya terpaut sekitar 0.96% dari RF, yang menunjukkan bahwa arsitektur MLP tetap efektif dalam mempelajari pola non-linear dari 70 fitur statis. Selisih kecil ini sejalan dengan karakteristik data tabular terstruktur, di mana algoritma ensemble seperti Random Forest cenderung memberikan kinerja optimal secara lebih efisien. Keunggulan tipis namun konsisten dari RF ini divisualisasikan pada Gambar 7 melalui grafik batang komparatif kedua model.



ISSN 2598-9936 (online), https://ijins.umsida.ac.id, published by Universitas Muhammadiyah Sidoarjo

Vol. 27 No. 1 (2026): January
DOI: 10.21070/ijins.v27i1.1883

Figure 6. *Perbandingan Performa Model*

IV. Kesimpulan

Berdasarkan hasil analisis dan evaluasi terhadap dua algoritma klasifikasi Random Forest (RF) dan Neural Network (MLP) pada dataset ClaMP berbasis fitur statis PE, dapat disimpulkan bahwa proses pra-pemrosesan data berjalan optimal dengan kondisi dataset yang lengkap dan proporsional berkat penggunaan stratified sampling. Hasil eksperimen menunjukkan bahwa Random Forest secara konsisten unggul dibandingkan Neural Network, dengan akurasi, presisi, recall, dan F1-score mencapai 99.14%, lebih tinggi dari MLP yang memperoleh 98.18%. Dalam konteks matriks noise, RF menghasilkan kesalahan klasifikasi yang lebih sedikit (9) daripada MLP. Temuan ini menunjukkan bahwa sifat statistik PE sangat berguna, dan pendekatan ensemble seperti Random Forest berkinerja baik pada dataset yang kehilangan parameter kunci. Topik ini dapat dikembangkan lebih lanjut untuk mendapatkan kombinasi parameter optimal, khususnya dengan menggunakan pendekatan model optimasi seperti Grid Search atau Bayesian Optimization. Penelitian di masa mendatang dapat menggunakan pendekatan deep learning, seperti CNN dan autoencoder, untuk meningkatkan representasi fitur dan mengidentifikasi malware polimorfik menggunakan metode statistik dan dinamis. Lebih lanjut, menambahkan mekanisme deteksi real-time dan memperluas dataset untuk mencakup spektrum jenis malware yang lebih luas akan meningkatkan kinerja model di bidang keamanan siber industri. Studi ini meneliti kegunaan Random Forest dan Neural Network dalam mendeteksi malware menggunakan sifat statistik PE. Temuan menunjukkan bahwa Random Forest memperoleh akurasi 99,14% sambil mempertahankan stabilitas prediksi yang luar biasa. Makalah ini menunjukkan bahwa model ensemble dapat berhasil menciptakan sistem deteksi malware yang kuat dan menawarkan dasar akademis yang solid untuk dataset ClaMP. Temuan studi ini akan digunakan sebagai referensi untuk pengembangan sistem keamanan berbasis pembelajaran mesin, sehingga memajukan penelitian keamanan siber di Indonesia.

References

1. A. Mello, V. G. Lemos, F. Barbosa, and E. Simoni, “Malware identification on Portable Executable files using Opcodes Sequence,” in *Anais do XVI Congresso Brasileiro de Inteligência Computacional*, SBIC, Dec. 2023, pp. 1–8. doi: 10.21528/CBIC2023-006.
2. D. Efriyani and F. Panjaitan, “Klasifikasi Malware dengan Menggunakan Recurrent Neural Network,” *jurnalmatrik*, vol. 23, no. 3, pp. 310–316, Dec. 2021, doi: 10.33557/jurnalmatrik.v23i3.1592.
3. M. Pietrek, “An In-Depth Look into the Win32 Portable Executable File Format, Part 2,” *MSDM Megazine Inside Windows*, pp. 1–10, 2002.
4. V. Kumar and A. Agarwal, “Circular Quorum Systems for Write Dominant Data Replication Protocols under Serial Isolation Using Quorum Consensus Approach,” *Procedia Computer Science*, vol. 46, pp. 867–875, 2015, doi: 10.1016/j.procs.2015.02.156.
5. L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
6. C. Hukubun, Abdurrokhim, Y. A. Sendjaja, and W. G. Hukubun, “Litofasies Dan Lingkungan Pengendapan Pada Formasi Elat, Kecamatan Kei Besar, KabupatenM Maluku Tenggara, Maluku,” *JURNAL GEOLOGI KELAUTAN*, vol. 20, no. 1, pp. 1–14, June 2022, doi: 10.32693/jgk.20.1.2022.759.
7. Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta, 2020.
8. M. S. Akhtar and T. Feng, “Evaluation of Machine Learning Algorithms for Malware Detection,” *Sensors*, vol. 23, no. 2, p. 946, Jan. 2023, doi: 10.3390/s23020946.
9. P. T. Noi and M. Kappas, “Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery,” *Sensors*, vol. 18, no. 1, p. 18, Dec. 2017, doi: 10.3390/s18010018.
10. J. N. Makapuan and R. C. Rohmana, “Prediksi Elektrofasies Berdasarkan Data Log Sumur pada Formasi Huggin: Studi Kasus Menggunakan Algoritma Random Forrest dan Multi-Layer Perceptron,” *Jurnal Teknik & Teknologi Terapan*, vol. 3, no. 1, pp. 1–12, 2025, doi: https://doi.org/10.47970/jttt.v3i1.863.
11. H. Babbar, S. Rani, D. K. Sah, S. A. AlQahtani, and A. Kashif Bashir, “Detection of Android Malware in the Internet of Things through the K-Nearest Neighbor Algorithm,” *Sensors*, vol. 23, no. 16, p. 7256, Aug. 2023, doi: 10.3390/s23167256.
12. M. Hamir and P. Nurtantio Andono, “Model Neural Network untuk Memprediksi Tingkat Kemenangan Berdasarkan Draft Pick Mobile Legends,” *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 5, no. 4, pp. 899–908, Apr. 2025, doi: 10.52436/1.jpti.723.
13. D. A. Fitri and D. Damayanti, “Komparasi Algoritma Random Forest Classifier dan Support Vector Machine untuk Sentimen Masyarakat Terhadap Pinjaman Online di Media Sosial,” *jipi. jurnal. ilmiah. penelitian. dan. pembelajaran. informatika.*, vol. 9, no. 4, pp. 2018–2029, Nov. 2024, doi: 10.29100/jipi.v9i4.5608.
14. N. Cahyani, R. Irsyada, and A. Y. Kartini, “Implementasi Machine Learning Model sebagai Sistem Prediksi Penyakit Breast Cancer,” *digitech*, vol. 4, no. 2, pp. 1112–1120, Jan. 2025, doi: 10.47709/digitech.v4i2.5209.
15. K. Anam, A. R. Rinaldi, and F. Fathurrohman, “Komparasi Algoritma Mechine Learning dalam Klasifikasi Loyalitas Nasabah Bank Berbasis Particle Swarm Optimization,” *jati*, vol. 8, no. 4, pp. 8212–8218, Aug. 2024, doi: 10.36040/jati.v8i4.10941.