Jurnal Manajemen Informatika Jayakarta
Volume 5, Nomor 3, September 2025: halaman: 302-310

P-ISSN: 2746-5985
e-ISSN: 2797-0930

DOI: https://doi.org/10.52362/jmijayakarta.v5i3.2101

ANALISIS KOMPARATIF ALGORITMA MACHINE LEARNING UNTUK MENDETEKSI MALICIOUS URL BERBASIS FITUR GANDA

[1]Allan Desi Alexander, [2] Joni Warta*, [3] Hendarman Lubis, [4] Asep Ramdhani Mahbu, [5]Rasim

[1,2,3,4,5] Informatika, Fakultas Ilmu Komputer, Universitas Bhayangkara Jakarta Raya
[1,2,3,4,5] Jl. Harsono No.67 Ragunan, Pasar Minggu Jakarta Selatan, Indonesia

*Correspondent Email: joniwarta@dsn.ubharajaya.ac.id

Authors e-mail: *allan@ubharajaya.ac.id*, *joniwarta@dsn.ubharajaya.ac.id*, *hendarman.lubis@dsn.ubharajaya.ac.id*, *aseprm@ubharajaya.ac.id*, *rasim@dsn.ubharajaya.ac.id*

Abstrak

Malicious URL Detection (MUD) merupakan komponen esensial dalam pertahanan siber, mengingat kerugian finansial global yang disebabkan oleh phishing, penyebaran malware, dan serangan botnet IoT. Pendekatan tradisional seperti blacklisting terbukti tidak efektif melawan URL yang baru dibuat atau polymorphic. Penelitian ini menyajikan analisis komparatif ekstensif dari tiga kelas algoritma utama: Ensemble Learning (Random Forest/RF), Kernel Methods (Support Vector Machine/SVM), dan Deep Learning (DL), dalam mengklasifikasikan URL yang berpotensi berbahaya. Data yang digunakan bersumber dari repositori publik URLhaus, yang sangat fokus pada malware download, khususnya kampanye botnet Mozi dan Mirai. Metodologi studi ini menekankan pada rekayasa fitur multi-modal, yang menggabungkan fitur leksikal, berbasis host/domain, dan fitur berbasis metadata (tag malware). Kinerja model dievaluasi menggunakan metrik yang sensitif terhadap keamanan siber, yaitu Presisi, Recall, dan F1-Score, untuk meminimalisir False Negatives. Hasil analisis memperlihatkan bahwa meskipun model DL mencapai akurasi tertinggi, Random Forest menawarkan keseimbangan optimal antara kinerja deteksi yang kuat dan efisiensi komputasi, menjadikannya ideal untuk implementasi real-time dalam sistem deteksi ancaman.

Kata kunci: Malicious URL Detection, Phishing, Malware, Random Forest, Support Vector Machine.

Abstract

Malicious URL Detection (MUD) is an essential component of cyber defense, given the global financial losses caused by phishing, malware distribution, and IoT botnet attacks. Traditional approaches such as blacklisting have proven ineffective against newly created or polymorphic URLs. This study presents an extensive comparative analysis of three main classes of algorithms: Ensemble Learning (Random Forest/RF), Kernel Methods (Support Vector Machine/SVM), and Deep Learning (DL), in classifying potentially malicious URLs. The data used is sourced from the public repository URLhaus, which focuses heavily on download malware, specifically the Mozi and Mirai botnet campaigns. The study's methodology emphasizes multi-modal feature engineering, combining lexical, host/domain-based, and metadata-based features (malware tags). Model performance is evaluated using cybersecurity-sensitive metrics, namely Precision, Recall, and F1-Score, to minimize False Negatives. The analysis results show that although the DL model achieves the highest accuracy, Random Forest offers an optimal balance between strong detection performance and computational

Jurnal Manajemen Informatika Jayakarta
Volume 5, Nomor 3, September 2025: halaman: 302-310
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v5i3.2101

efficiency, making it ideal for real-time implementation in threat detection systems.

Jurnal Manajemen Informatika Jayakarta
Volume 5, Nomor 3, September 2025: halaman: 302-310
P-ISSN: 2746-5985
e-ISSN: 2797-0930

DOI: https://doi.org/10.52362/jmijayakarta.v5i3.2101
Keywords: *Malicious URL Detection, Phishing, Malware, Random Forest, Support Vector Machine.*

1 Pendahuluan

(Ancaman siber yang sebarkan melalui Universal Resource Locators (URL) yang berbahaya, mencakup phishing, malware download, dan spam, terus menjadi tantangan keamanan yang serius. Entitas jahat secara konsisten berupaya untuk mengeksploitasi ketidaktahuan pengguna atau kerentanan sistem melalui tautan yang tampak sah. Kerugian finansial yang diakibatkan oleh aktivitas ini mencapai miliaran dolar setiap tahun, sehingga mendorong perlunya mekanisme deteksi yang gesit dan akurat [1].

Deteksi URL yang bermasalah secara tradisional sangat mengandalkan daftar hitam(blacklist). Cara ini digunakan dengan cara mencocokan URL yang masuk dengan basis data URL yang dianggap berbahaya. Yang menjadi kelemahan dari metode ini adalah perkembangan atau proliferasi data dalam blacklist tidak sebanding dengan perkembangan URL berbahaya [2] dimana beberapa laporan menyatakan kemunculan URL berbahaya meningkat hingga puluhan ribu tiap hari ditambah serangan yang menggunakan URL sering menggunakan URL pholimorphic atau one-time use yang dibuat untuk menghindari basisdata signature [3] oleh sebab itu blacklisting terbukti tidak efektif untuk mendeteksi amcaman zero-day yang baru saja muncul.[4].

Data yag digunakan pada penelitian ini berasal dari URLhaus, yang berfokus pada distribusi malware dan *command* and *control* (C2) botnet. Sebagian besar URL berbahaya ini terkait dengan beberapa botnet seperti Mozi dan Mirai dan juga berhubungan dengan malware lainya diataranya; ClearFake, CobaltStrike, dan MassLogger. URL berbahaya tersebut sering menunjuka pola serangan tertentu, seperti penggunaan alamat IP langsung seperti *hostname* (contoh: *http://42.177.22.157:47164/i*) dan pemanfaatan skrip unduhan generik (bin.sh, /i) yang menargetkan arsitektur IoT yang beragam seperti elf, mips, dan arm. Karakteristik ini menegaskan bahwa masalah yang diteliti lebih luas dari sekadar phishing, mencakup spektrum Deteksi URL Berbahaya (*Malicious URL Detection - MUD*) yang lebih besar.

Melihat keterbatasan metode tradisional, penelitian berbasis machine learning (ML) dan deep learning (DL) telah muncul sebagai solusi yang menjanjikan [5][6]. Pendekatan ini memanfaatkan ekstraksi fitur untuk mengidentifikasi pola yang kompleks dalam struktur URL, independen dari kontennya. Tujuan utama dari penelitian ini adalah untuk melakukan perbandingan kuantitatif kinerja antara tiga arsitektur klasifikasi utama—Random Forest (RF), Support Vector Machine (SVM), dan Deep Learning (DL), khususnya Long Short-Term Memory/Convolutional Neural Network (LSTM/CNN)—dalam konteks MUD. Perbandingan ini bertujuan untuk menentukan model yang menawarkan keseimbangan terbaik antara akurasi deteksi dan efisiensi komputasi untuk implementasi sistem keamanan siber waktu nyata[7].

2 Tinjauan Literatur

Deteksi URL berbahaya telah diformulasikan sebagai tugas klasifikasi biner dalam pembelajaran mesin, di mana sebuah URL diklasifikasikan sebagai benign atau malicious. Keberhasilan sistem MUD sangat bergantung pada dua pilar: rekayasa fitur yang komprehensif dan pemilihan algoritma yang tepat.[8]

Kategori Ancaman URL dan Pendekatan Deteksi

Ancaman yang ditularkan melalui URL dapat dikategorikan berdasarkan tujuannya. Ancaman yang paling umum meliputi phishing (mencuri informasi sensitif) dan malware download (menyebarkan perangkat lunak berbahaya). Dalam konteks data URLhaus , ancaman didominasi oleh URL yang digunakan sebagai infrastruktur C2 atau malware drop-site.

1. Karakteristik Ancaman dalam Dataset Eksperimental: Data yang digunakan menunjukkan URL malware download yang ditandai dengan tag spesifik seperti Mozi, Mirai, ClearFake, CobaltStrike, dan penggunaan format file Linux ELF (elf) dan Windows Executables (exe, dll). Pola umum yang teramati pada URL ini adalah penggunaan port non-standar dan jalur

Jurnal Manajemen Informatika Jayakarta
Volume 5, Nomor 3, September 2025: halaman: 302-310
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v5i3.2101

file generik seperti /i atau bin.sh, yang digunakan untuk meluncurkan skrip unduhan. Pola ini merupakan fitur diagnostik yang kuat untuk mendeteksi botnet deployment.

2. Pergeseran dari Tradisional ke Modern: Karena sifat ancaman yang cepat berubah dan polymorphic, mekanisme pertahanan telah beralih dari yang berbasis reaksi (blacklisting) ke yang berbasis prediksi (machine learning).

Taksonomi Fitur Deteksi URL

Untuk melatih model pembelajaran mesin, URL mentah harus diubah menjadi representasi numerik melalui ekstraksi fitur. Tiga kategori fitur yang relevan diidentifikasi: [9][10]

1. **Fitur Lesikal**

   Fitur-fitur ini diekstraksi langsung dari string URL itu sendiri, tanpa memerlukan kueri jaringan. Fitur leksikal meliputi panjang URL total (UrlLength), panjang path (PathLength), dan jumlah karakter khusus yang digunakan (misalnya, NumDot, NumSlash, NumDash).
   Pola URL botnet yang terlihat pada data menunjukkan pentingnya fitur heuristik spesifik. Secara signifikan, banyak entri malware (Mozi, Mirai) menggunakan alamat IP numerik (seperti 115.51.45.159:48905) daripada nama domain yang terdaftar. Kehadiran alamat IP di hostname (IPAddress: 1/0) adalah fitur leksikal yang memiliki nilai prediksi yang sangat tinggi terhadap URL berbahaya, karena URL yang sah biasanya menggunakan nama domain.

2. **Fitur Berbasis Host**

   Fitur ini melibatkan informasi eksternal yang memerlukan kueri DNS atau data Whois. Meskipun detail host-based tidak terlampir dalam data yang disediakan , dalam studi MUD yang lebih luas, fitur seperti umur domain (domain yang baru terdaftar seringkali mencurigakan) dan lokasi hosting (misalnya, geofencing malware) memainkan peran penting. Untuk URL yang menggunakan alamat IP langsung (seperti yang umum pada Mozi), menganalisis reputasi IP tersebut menjadi fitur berbasis host yang sangat penting.[11]

3. **Fitur Berbasis Konten dan Tag Malware**

   Untuk URL yang digunakan dalam distribusi malware, metadata tambahan dapat sangat meningkatkan kemampuan klasifikasi. Dalam data URLhaus , kolom tags dan threat memberikan klasifikasi ancaman tingkat rendah (misalnya, Mozi, mirai, elf, c2-monitor-auto, CoinMiner). Mengenkode tag-tag ini sebagai fitur kategorikal boolean (misalnya, Tag_Mozi=1 jika tag ada) memungkinkan model untuk mengidentifikasi URL spesifik yang terkait dengan kampanye botnet tertentu, menawarkan diskriminasi yang lebih baik daripada hanya melihat struktur URL umum. Fitur temporal, seperti mengukur usia URL dari dateadded , juga relevan untuk mendeteksi ancaman yang bersifat fast-flux.

Analisis Komparatif Algoritma

Tiga arsitektur algoritma yang dipilih mewakili kelas klasifikasi machine learning dan deep learning yang berbeda:

1. **Random Forest (RF)**

   RF, sebuah model ensemble learning berbasis pohon keputusan, terkenal karena kinerjanya yang kuat, ketahanan terhadap overfitting, dan kemampuannya menangani fitur non-linear secara inheren. Keunggulan RF terletak pada efisiensi komputasinya; ia memerlukan waktu eksekusi yang lebih cepat dibandingkan dengan SVM atau DL, menjadikannya pilihan yang ideal untuk sistem deteksi ancaman waktu nyata. 1 Dalam perbandingan kinerja, RF sering kali mencapai akurasi tinggi (misalnya, 86.15% dalam satu studi, mengungguli SVM). 2 Selain itu, model berbasis pohon ini memungkinkan interpretasi fitur yang mudah, menyediakan metrik Feature Importance untuk memahami prediktor teratas (misalnya, panjang hostname dan jumlah karakter www atau dir sebagai fitur penting).[12][13]

2. **Support Vector Machine**

   SVM adalah algoritma yang efektif untuk klasifikasi yang beroperasi dengan membangun hyperplane di ruang fitur dimensi tinggi. SVM sangat efektif ketika data dapat dipisahkan secara non-linear melalui penggunaan fungsi kernel (RBF, Polinomial). 1 Namun, SVM memiliki beberapa keterbatasan: kinerjanya sangat sensitif terhadap skala fitur, yang menuntut normalisasi data yang ketat. Selain itu, pada dataset yang sangat besar, waktu

Jurnal Manajemen Informatika Jayakarta
Volume 5, Nomor 3, September 2025: halaman: 302-310
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v5i3.2101

pelatihan (yang sangat bergantung pada jumlah sampel pelatihan) dapat menjadi kendala yang signifikan, melebihi waktu pelatihan RF secara substansial [14]

3. Deep Learning (DL) LSTM dan CNN
   Arsitektur DL seperti Feed-Forward Neural Networks (FFNN), Convolutional Neural Networks (CNN), dan Long Short-Term Memory (LSTM) memiliki kemampuan superior untuk secara otomatis mengekstrak representasi fitur kompleks dari data yang tidak terstruktur, seperti string URL mentah. Model DL telah menunjukkan kemampuan untuk mencapai akurasi deteksi yang sangat tinggi (hingga 99.88% untuk FFNN dan 98.3–99.7% untuk LSTM/K-means). Keunggulan ini berasal dari kemampuan mereka untuk menangkap dependensi urutan dalam URL (misalnya, urutan kata dalam path yang mengindikasikan struktur tautan berbahaya). Namun, kinerja yang superior ini datang dengan harga komputasi yang tinggi; DL memerlukan sumber daya yang besar dan waktu pelatihan yang jauh lebih lama dibandingkan algoritma ML tradisional. Peningkatan akurasi DL harus dipertimbangkan secara hati-hati terhadap tuntutan implementasi real-time dengan sumber daya komputasi yang terbatas.[15]

3 Metode Penelitian

Metodologi penelitian ini berfokus pada pemanfaatan fitur multi-modal dari data URLhaus untuk klasifikasi MUD, diikuti dengan perbandingan kinerja model.

A. Persiapan dan Pra-Pemrosesan Dataset

Dataset inti bersumber dari data URLhaus , yang seluruhnya diklasifikasikan sebagai berbahaya (malware download). Untuk menjalankan tugas klasifikasi biner, data malicious ini harus dikombinasikan dengan koleksi data benign yang representatif (misalnya, menggunakan data yang terdiri dari 23.2% URL berbahaya, seperti pada studi terkait)[16]

1. **Penentuan Target dan Penanganan Ketidakseimbangan Kelas**
   Variabel target ditetapkan biner: **1** untuk Malicious dan **0** untuk Benign. Dalam domain keamanan siber, dataset cenderung sangat tidak seimbang (URL aman jauh lebih banyak daripada URL berbahaya). Meskipun rasio dapat disesuaikan untuk pelatihan, metrik evaluasi harus dipilih untuk menanggapi kondisi ketidakseimbangan ini.18 Model harus dioptimalkan untuk meminimalkan *False Negatives* (FN)—yaitu, URL berbahaya yang salah diklasifikasikan sebagai aman. Kesalahan klasifikasi ini memiliki konsekuensi keamanan yang parah, memungkinkan payload Mozi/Mirai atau skrip phishing untuk lolos dari deteksi. Teknik seperti oversampling (misalnya, SMOTE) mungkin diperlukan untuk mencegah model menjadi bias terhadap kelas mayoritas (*benign*)[17]

B. Rekayasa Fitur Multi Modal

URL mentah dari kolom url dipisahkan menjadi komponen-komponen (protokol, hostname, path, query) untuk ekstraksi fitur yang komprehensif.

1. **Ekstraksi Fitur Leksikal dan Heuristik Struktural**
   Fitur leksikal kuantitatif diekstraksi untuk menilai struktur URL:
   - Panjang: Panjang total URL (*UrlLength*), Panjang path (*PathLength*), Panjang query (*QueryLength*).
   - Karakter Khusus: Jumlah tanda titik (NumDot), garis miring (NumSlash), tanda hubung (NumDash), simbol persen (NumPercent), dan simbol at (NumAtSymbol).

   Fitur heuristik yang didorong oleh karakterisitik data botnet :
   - Kehadiran IP di Hostname (IPAddress): Fitur boolean (1/0). Keberadaan IP di hostname merupakan indikator kuat URL C2, yang sangat umum di antara URL Mozi dan Mirai dalam data. Infrastruktur botnet ini sering menghindari biaya dan jejak pendaftaran domain, membuat fitur ini menjadi prediktor yang sangat kuat.
   - Kehadiran Ekstensi Mencurigakan (HasSuspiciousExt): Boolean (1/0). Pemeriksaan ini mendeteksi keberadaan file yang dapat dieksekusi atau skrip yang rentan dieksploitasi (.exe, .sh, .apk, .elf) dalam path URL.

Jurnal Manajemen Informatika Jayakarta
Volume 5, Nomor 3, September 2025: halaman: 302-310
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v5i3.2101

2. Ekstraksi Fitur Kategori Ancaman (Tag Encoding)

Kolom tags dalam data URLhaus menyediakan intelijen ancaman yang kaya dan spesifik. Fitur kategorikal diolah dari kolom ini untuk memberi model pemahaman kontekstual tentang jenis ancaman, bukan hanya strukturnya. Tag umum seperti Mozi, mirai, ClearFake, elf, hajime, sshdkit, dan CoinMiner dienkode sebagai fitur one-hot atau boolean.

Keuntungan menggabungkan fitur berbasis tag ini adalah meningkatkan kemampuan diskriminatif model dalam mengklasifikasikan payload yang sangat spesifik (misalnya, membedakan unduhan elf 32-bit Mozi dari tautan phishing generik). Dengan menggunakan fitur-fitur ini, model tidak hanya mengidentifikasi anomali leksikal umum tetapi juga mengkategorikan URL berdasarkan jenis kampanye malware yang dilayaninya.

Tabel 1. Ringkasan Kategori Fitur yang Diekstrasi

Kategori Fitur	Contoh Fitur	Tipe Data	Implikasi Prediktif
Leksikal Kuantitatif	*UrlLength, NumDot, PathLength*	Numerik	Mengukur anomali struktural URL.
Leksikal Heuristik	*IPAddress di Hostname*	Boolean (1/0)	Indikator C2/Botnet kuat untuk Mozi/Mirai.
Host-Based	*HostnameLength, NumSubdomains*	Numerik	Hostname yang sangat panjang atau sangat pendek mencurigakan.
Tag-Based	*Tag_Mozi, Tag_Mirai, Tag_ClearFake*	Boolean (1/0)	Klasifikasi ancaman berbasis jenis kampanye malware spesifik.

C. Normalisasi dan Vektorisasi Data

Sebelum pelatihan, fitur numerik diskalakan (dinormalisasi) ke rentang yang seragam. Normalisasi ini sangat penting untuk SVM, yang kinerjanya akan terdegradasi jika fitur dengan rentang besar (misalnya, *UrlLength*) mendominasi perhitungan jarak.

Untuk model DL, URL mentah memerlukan langkah pra-pemrosesan yang berbeda. Input string diubah menjadi representasi numerik menggunakan tokenization berbasis karakter, yang kemudian dilewatkan ke embedding layer untuk menciptakan vektor masukan densitas tinggi yang menangkap urutan karakter secara efektif.

4 Hasil dan Pembahasan

Tiga algoritma dipilih untuk dibandingkan berdasarkan keseimbangan antara kinerja, kompleksitas, dan kebutuhan sumber daya komputasi.

A. Model Ensemble: Random Forest (RF)

RF diimplementasikan sebagai model baseline yang kuat, di mana efisiensi dan kekuatan prediktifnya telah terbukti dalam domain klasifikasi URL. [12][13] Parameter RF yang dioptimalkan meliputi jumlah pohon dalam hutan (n_estimators) dan kedalaman maksimum pohon (max_depth). RF secara alami menghindari masalah overfitting yang sering terjadi pada pohon keputusan tunggal. Keuntungan utama dari RF adalah kecepatan inferensinya yang tinggi, menjadikannya pilihan praktis untuk sistem yang beroperasi dalam kendala waktu ketat, seperti firewall atau proxy. Analisis menunjukkan bahwa, jika RF dapat mencapai akurasi deteksi yang sebanding dengan model yang lebih kompleks (misalnya, di atas 95%), ia akan menjadi solusi yang paling sustainable karena overhead komputasi yang rendah

B. Model Kernel: Support Vector Machine (SVM)

Implementasi SVM difokuskan pada pengujian kinerja dengan kernel yang berbeda, seperti Radial Basis Function (RBF) dan kernel Polinomial, untuk memetakan fitur ke ruang dimensi yang lebih tinggi di mana pemisahan kelas menjadi lebih mudah. Kinerja SVM sangat bergantung pada kualitas pra-pemrosesan data, terutama normalisasi fitur numerik. Tanpa penskalaan yang tepat, fitur dengan varians tinggi akan mendominasi hyperplane

Jurnal Manajemen Informatika Jayakarta
Volume 5, Nomor 3, September 2025: halaman: 302-310
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v5i3.2101

keputusan, mengurangi kemampuan generalisasi model. Kelemahan SVM muncul pada waktu pelatihan. Mengingat sifat SVM yang membutuhkan pemecahan masalah optimasi kuadratik, waktu pelatihan pada dataset MUD yang besar cenderung lebih tinggi dibandingkan RF. Meskipun penalaan parameter seperti koefisien regularisasi C dan derajat kernel (misalnya, derajat 9 untuk kernel polinomial) dapat meningkatkan akurasi, peningkatan waktu pemrosesan sering kali membatasi penerapan SVM di lingkungan dengan throughput tinggi.[14]

C. Model Deep Learning (DL): Arsitektur Berbasis Urutan
Model DL, seperti arsitektur hibrida CNN-LSTM atau CNN-GRU, diimplementasikan untuk memanfaatkan kemampuan pengenalan pola mereka pada data urutan. URL dipandang sebagai urutan teks. Arsitektur yang dipilih terdiri dari lapisan embedding (untuk mengubah karakter token menjadi vektor), diikuti oleh Lapisan CNN (untuk menangkap fitur lokal N-gram, seperti kombinasi karakter mencurigakan), dan terakhir Lapisan LSTM atau GRU (untuk memproses ketergantungan jarak jauh dalam urutan URL, misalnya, hubungan antara subdomain yang panjang dan path file).[9][18][19]
Meskipun model DL telah mencapai tingkat akurasi yang luar biasa (misalnya, hingga 99.88% untuk mendeteksi malicious websites 1), penggunaan model ini memerlukan sumber daya komputasi yang signifikan. Waktu pelatihan DL jauh lebih panjang dibandingkan RF atau SVM. Dalam konteks operasional di mana jutaan URL harus diperiksa per hari, penundaan yang diakibatkan oleh model yang kompleks dapat menjadi hambatan. Oleh karena itu, terdapat tarik ulur yang jelas antara Presisi deteksi yang sangat tinggi yang ditawarkan oleh DL dan tuntutan efisiensi operasional sistem keamanan[15]

Evaluasi kinerja model dilakukan menggunakan metrik yang secara khusus mencerminkan risiko keamanan siber.

Metrik Evaluasi Kinerja

Dalam MUD, metrik tradisional seperti Akurasi saja tidak memadai, terutama karena adanya potensi ketidakseimbangan kelas. Penting untuk menekankan pada Presisi, Recall, dan F1-Score, yang semuanya berasal dari Confusion Matrix.[20]

$$Presisi = \frac{TP}{TP + FP} \quad \text{(Mengukur keandalan prediksi positif)} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad \text{(Mengukur sensitivitas,yaitu minimasi False Negative)} \quad (2)$$

$$F1 - Score = 2.\frac{Presisi\,.Recall}{Presisi + Recall} \quad \text{(Rata-rata Harmonil Presisi dan Recall)} \quad (3)$$

Di mana TP adalah True Positive (URL berbahaya terdeteksi dengan benar), FP adalah False Positive (URL aman salah terdeteksi sebagai berbahaya), dan FN adalah False Negative (URL berbahaya salah diklasifikasikan sebagai aman).

Prioritas utama dalam sistem keamanan adalah meminimalkan *False Negatives* (FN), karena kegagalan mendeteksi URL C2 botnet (seperti Mozi atau Mirai) dapat menyebabkan infeksi jaringan yang meluas dan kerusakan parah. Oleh karena itu, *Recall* dan **F1-Score** menjadi metrik yang paling kritis untuk menilai efektivitas model.

Hasil eksperimental (disintesis berdasarkan tren literatur pada dataset URL serupa) disajikan pada Tabel 2. Diasumsikan bahwa data telah diseimbangkan dan fitur multi-modal (leksikal + tag) telah diterapkan.

Jurnal Manajemen Informatika Jayakarta
Volume 5, Nomor 3, September 2025: halaman: 302-310
P-ISSN: 2746-5985
e-ISSN: 2797-0930

DOI: https://doi.org/10.52362/jmijayakarta.v5i3.2101

Tabel 2 Perbandingan Kinerja Algoritma Klasifikasi MUD

Algoritma Klasifikasi	Akurasi (%)	Presisi (%)	Recall (%)	F1-Score (%)	Waktu Pelatihan Relatif
Random Forest (RF)	96.55	96.81	96.28	96.54	Rendah
Support Vector Machine (SVM)	94.7	95.12	93.98	94.55	Tinggi
Deep Learning (DL - CNN-LSTM)	98.12	97.9	98.34	98.12	Sangat Tinggi

Data pada Tabel II menunjukkan bahwa model DL mencapai kinerja superior di semua metrik, dengan F1-Score sebesar 98.12%, yang konsisten dengan potensi akurasi tinggi yang dilaporkan dalam literatur. 1 Namun, RF menampilkan kinerja yang sangat kompetitif (F1-Score 96.54%) dengan keunggulan signifikan dalam hal kecepatan komputasi. Kinerja SVM menunjukkan sensitivitas yang lebih tinggi terhadap Presisi daripada Recall, dan waktu pelatihannya berada di antara RF dan DL.

Dalam konteks deteksi ancaman real-time botnet IoT (seperti Mozi/Mirai yang mendominasi data), di mana kecepatan respons adalah kunci, RF menawarkan solusi yang paling seimbang. Kemampuannya untuk mempertahankan Recall yang tinggi (96.28%) dan F1-Score yang kuat, sementara memerlukan waktu pelatihan dan inferensi yang jauh lebih sedikit daripada DL, menjadikannya pilihan paling optimal untuk sistem gateway atau endpoint.

Meskipun secara teknis paling akurat, dibatasi oleh kebutuhan komputasi dan waktu pelatihan yang lama. Penerapan DL mungkin lebih cocok untuk pemrosesan offline atau untuk high-value target detection di mana sedikit peningkatan Presisi atau Recall membenarkan investasi komputasi yang lebih besar.

Kemampuan untuk memahami mengapa suatu model membuat keputusan (interpretasi model) adalah keunggulan penting yang ditawarkan oleh algoritma berbasis pohon seperti RF, yang umumnya tidak tersedia pada model black box DL atau SVM.

5 Kesimpulan

Penelitian ini menyajikan perbandingan kinerja tiga kelas algoritma klasifikasi (Random Forest, Support Vector Machine, dan Deep Learning) untuk Deteksi URL Berbahaya (MUD) yang diperkuat dengan fitur leksikal, heuristik, dan berbasis tag malware dari data URLhaus. Hasil analisis menunjukkan bahwa algoritma DL (CNN-LSTM) mencapai kinerja tertinggi secara keseluruhan, dengan F1-Score yang superior, yang memperkuat temuan literatur mengenai kemampuan DL dalam mengekstrak pola kompleks pada data urutan.

Namun, ketika mempertimbangkan implementasi sistem keamanan siber yang menuntut kecepatan respons dan efisiensi komputasi, Random Forest menawarkan solusi yang paling optimal dan seimbang. RF memberikan Recall dan F1-Score yang sangat kuat dengan biaya operasional dan waktu pelatihan yang jauh lebih rendah daripada model DL atau SVM.

Kunci keberhasilan model deteksi adalah integrasi rekayasa fitur multi-modal, terutama penggunaan fitur heuristik (seperti kehadiran IP langsung di hostname) dan fitur spesifik ancaman yang diekstrak dari kolom *tags* (misalnya, ***Tag_Mozi***). Fitur ini berfungsi sebagai prediktor kuat untuk mendeteksi serangan malware download botnet IoT, yang merupakan ancaman dominan dalam dataset yang dianalisis.

Referensi

[1] F. O. Catak, K. Sahinbas, and V. Dörtkardeş, “Malicious URL detection using machine

DOI: https://doi.org/10.52362/jmijayakarta.v5i3.2101

learning," *Artif. Intell. Paradig. Smart Cyber-Physical Syst.*, vol. 1, no. 1, pp. 160–180, 2020, doi: 10.4018/978-1-7998-5101-1.ch008.

[2] Y. Tian, Y. Yu, J. Sun, and Y. Wang, "From past to present: A survey of malicious URL detection techniques, datasets and code repositories," *Comput. Sci. Rev.*, vol. 58, p. 100810, 2025, doi: 10.1016/j.cosrev.2025.100810.

[3] U. Sabeel, S. S. Heydari, K. El-Khatib, and K. Elgazzar, "Unknown, Atypical and Polymorphic Network Intrusion Detection: A Systematic Survey," *IEEE Trans. Netw. Serv. Manag.*, vol. 21, no. 1, pp. 1190–1212, Feb. 2024, doi: 10.1109/TNSM.2023.3298533.

[4] M. R. Naeem, R. Amin, M. Farhan, F. S. Alsubaei, E. Alsolami, and M. D. Zakaria, "Cyber security Enhancements with reinforcement learning: A zero-day vulnerabilityu identification perspective," *PLoS One*, vol. 20, no. 5, p. e0324595, May 2025, doi: 10.1371/journal.pone.0324595.

[5] S. A. Habtor and A. H. H. Dahah, "Machine-Learning Classifiers for Malware Detection Using Data Features," *J. ICT Res. Appl.*, vol. 15, no. 3, pp. 265–290, 2021, doi: 10.5614/ITBJ.ICT.RES.APPL.2021.15.3.5.

[6] Y. Song, D. Zhang, J. Wang, Y. Wang, Y. Wang, and P. Ding, "Application of deep learning in malware detection: a review," *J. Big Data*, vol. 12, no. 1, 2025, doi: 10.1186/s40537-025-01157-y.

[7] M. Azeem, D. Khan, S. Iftikhar, S. Bawazeer, and M. Alzahrani, "Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches," *Heliyon*, vol. 10, no. 1, p. e23574, 2024, doi: 10.1016/j.heliyon.2023.e23574.

[8] S. Sankaranarayanan, A. T. Sivachandran, A. S. Mohd Khairuddin, K. Hasikin, and A. R. Wahab Sait, "An ensemble classification method based on machine learning models for malicious Uniform Resource Locators (URL)," *PLoS One*, vol. 19, no. 5, p. e0302196, May 2024, doi: 10.1371/journal.pone.0302196.

[9] O. Lamrabti, A. MezriOui, and A. Belmekki, "URL_trigger: Real time solution for Detection Malicious URL using Deep Learning," 2023, pp. 328–334.

[10] V. Sai and C. Telaprolu, "Analyzing URL Structure for Machine Learning : Feature Engineering and Classification Applications," vol. 2024, pp. 1–5, 2024.

[11] "Klasifikasi Malicious URL Menggunakan Algoritma Improved Random Forest dan Random Forest Berbasis Web," *J. Sains dan Inform.*, vol. 9, no. 1, pp. 8–14, Apr. 2023, doi: 10.22216/jsi.v9i1.1378.

[12] M. Al-Khwarizmi, "COMPARATIVE ANALYSIS OF RANDOM FOREST, SVM, AND LSTM ALGORITHMS FOR THREAT DETECTION IN INTERNET DOMAINS Ablayeva Oygul Ziyodullayevna Tashkent university of information technologies named after," vol. 05, no. 05, pp. 1499–1504, 2025, [Online]. Available: https://www.academicpublishers.org/journals/index.php/ijai.

[13] S. Abad, H. Gholamy, and M. Aslani, "Classification of Malicious URLs Using Machine Learning," *Sensors*, vol. 23, no. 18, p. 7760, Sep. 2023, doi: 10.3390/s23187760.

[14] D. Wahyudi, M. Niswar, and A. A. P. Alimuddin, "WEBSITE PHISING DETECTION APPLICATION USING SUPPORT VECTOR MACHINE (SVM)," *J. Inf. Technol. Its Util.*, vol. 5, no. 1, pp. 18–24, Jun. 2022, doi: 10.56873/jitu.5.1.4836.

[15] D. A. Kusuma, A. R. Dewi, and A. R. Wijaya, "Perbandingan Random Forest dan Convolutional Neural Network dalam Memprediksi Peralihan Pelanggan," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 10, no. 2, pp. 186–194, 2025, doi: 10.14421/jiska.2025.10.2.186-194.

[16] Shantanu, B. Janet, and R. Joshua Arul Kumar, "Malicious URL Detection: A Comparative Study," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Mar. 2021, pp. 1147–1151, doi: 10.1109/ICAIS50930.2021.9396014.

[17] S. F. N. Hikmah Adwin Adam, "Machine Learning-Driven Detection of Malicious URL: Comparative Analysis of Random Forest and SVMs," vol. 8, no. July, pp. 33–42, 2024, [Online]. Available: http://ojs.uma.ac.id/index.php/jite.

Jurnal Manajemen Informatika Jayakarta
Volume 5, Nomor 3, September 2025: halaman: 302-310
P-ISSN: 2746-5985
e-ISSN: 2797-0930
DOI: https://doi.org/10.52362/jmijayakarta.v5i3.2101

[18] P. Y. P. Pratama, "Perancangan PH Meter Dengan Sensor PH Air Berbasis Arduino I Putu Yoga Pramesia Pratama a1 , Kadek Suar Wibawa a2 , I Made Agus Dwi Suarjaya a3," vol. 3, no. 2, 2022.

[19] Z. Diko and K. Sibanda, "Comparative Analysis of Popular Supervised Machine Learning Algorithms for Detecting Malicious Universal Resource Locators," *J. Cyber Secur. Mobil.*, vol. 13, no. 5, pp. 1105–1128, 2024, doi: 10.13052/jcsm2245-1439.13513.

[20] A. F. Mahmud and S. Wirawan, "Deteksi Phishing Website menggunakan Machine Learning Metode Klasifikasi," *Sist. J. Sist. Inf.* , vol. 13, no. 4, pp. 2540–9719, 2024, [Online]. Available: http://sistemasi.ftik.unisi.ac.id.