Accredited SINTA 5
72/E/KPT/2024

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 7 No. 3, Juni 2024
P-ISSN 2620-8342
E-ISSN 2621-3052

Optimalisasi Deteksi Malware pada Platform Android dengan Pendekatan *Ensemble Machine Learning*

Karfindo[1*], Rifa Turaina[2], Rusli Saputra[3]
[1-3]Universitas Metamedia
[1-3]Jl. Khatib Sulaiman Dalam No.1, Padang
E-mail: [1]karfindo@metamedia.ac.id, [2]rifaturaina@metamedia.ac.id, [3]ruslisaputra@metamedia.ac.id

Submitted Date: 21 Juni 2024 ***Accepted Date: 25 Juni 2024***

Abstrak - Keamanan perangkat Android telah menjadi perhatian utama di era digital, mengingat dominasi sistem operasi ini dan meningkatnya serangan siber terhadap perangkat mobile. Penelitian ini bertujuan untuk meningkatkan akurasi deteksi malware pada platform Android melalui penggunaan teknik ensemble machine learning, khususnya metode soft voting. Teknik ini menggabungkan prediksi dari beberapa model pembelajaran mesin, seperti Random Forest, Gradient Boosting, dan XGBoost. Dataset yang digunakan dalam penelitian ini adalah KronoDroid. Pendekatan penelitian ini dimulai dengan preprocessing data dan pemilihan fitur izin yang sering digunakan, diikuti oleh pembagian data menjadi set pelatihan dan pengujian. Model-model individual dilatih dan dievaluasi, di mana Random Forest, Gradient Boosting, dan XGBoost masing-masing menunjukkan akurasi 85%, 84%, dan 85%. Hasil ini ditingkatkan melalui teknik soft voting dalam ensemble model, yang mencapai akurasi 90%. Teknik cross-validation lima kali lipat menunjukkan akurasi rata-rata 89.99% dengan deviasi standar 0.19%, menandakan konsistensi dan keandalan model. Confusion matrix yang dihasilkan menunjukkan bahwa model ensemble berhasil mengidentifikasi 7.496 dari 8.314 kasus malware (True Positives) dan 6.551 dari 7.314 kasus non-malware (True Negatives), dengan recall dan precision masing-masing sebesar 90% dan 91%. Meskipun terdapat false negatives dan false positives, model ini menunjukkan keseimbangan yang baik antara precision dan recall dengan F1-score mencapai 0.90. Penelitian ini membuktikan bahwa teknik ensemble, dengan menggabungkan kelebihan dari berbagai model individual, dapat secara signifikan meningkatkan deteksi malware pada perangkat Android.
Kata kunci: Keamanan Android, Deteksi Malware, Pembelajaran Ensemble, Pembelajaran Mesin

Abstract - *Android device security has become a major concern in the digital era, given the dominance of this operating system and the increasing cyber-attacks on mobile devices. This research aims to improve malware detection accuracy on the Android platform by using ensemble machine learning techniques, specifically the soft voting method. This technique combines predictions from several machine learning models, such as Random Forest, Gradient Boosting, and XGBoost. The dataset used in this research is KronoDroid. The research approach begins with data preprocessing and the selection of commonly used permission features, followed by splitting the data into training and testing sets. Individual models are trained and evaluated, where Random Forest, Gradient Boosting, and XGBoost each showed accuracies of 85%, 84%, and 85%, respectively. These results were enhanced through the soft voting technique in the ensemble model, achieving an accuracy of 90%. Five-fold cross-validation showed an average accuracy of 89.99% with a standard deviation of 0.19%, indicating the model's consistency and reliability. The generated confusion matrix shows that the ensemble model successfully identified 7,496 out of 8,314 malware cases (True Positives) and 6,551 out of 7,314 non-malware cases (True Negatives), with recall and precision of 90% and 91%, respectively. Although there are false negatives and false positives, this model demonstrates a good balance between precision and recall with an F1-score of 0.90. This research proves that ensemble techniques, by combining the strengths of various individual models, can significantly improve malware detection on Android devices.*
Keywords: LSB Method, Information Insertion, Image Files, Steganography

1. Pendahuluan

Dalam era digital yang terus berkembang, keamanan perangkat bergerak, khususnya telepon pintar berbasis Android, menjadi fokus utama bagi akademisi dan praktisi industri [1]. Laporan keamanan terbaru menunjukkan bahwa hampir semua organisasi menghadapi ancaman seluler, dengan insiden yang melibatkan unduhan tidak sadar aplikasi berbahaya oleh karyawan [2]. Dengan lebih dari 3 miliar pengguna aktif global, dominasi Android di pasar menjadikannya target utama serangan siber. Tahun 2021 saja, terdapat lebih dari 1,3 juta aplikasi berbahaya baru di perangkat Android [3], mencerminkan tren pertumbuhan malware yang mengkhawatirkan. Ancaman ini tidak hanya mencakup aplikasi berbahaya yang tersedia secara luas di Google Play tetapi juga melibatkan teknik pengaburan kode yang canggih, memperumit upaya deteksi [4]. Menyikapi

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 7 No. 3, Juni 2024
P-ISSN 2620-8342
E-ISSN 2621-3052

hal ini, pentingnya penerapan mekanisme keamanan yang lebih kuat dan pengembangan teknologi deteksi malware yang lebih efektif menjadi sangat krusial. Platform Android, yang telah menjadi salah satu sistem operasi mobile paling populer di dunia, digunakan oleh miliaran pengguna di berbagai perangkat, mulai dari smartphone hingga tablet. Dengan pertumbuhan yang pesat ini, Android telah menjadi target utama bagi pengembang aplikasi dan penjahat siber yang berusaha mengeksploitasi celah keamanan untuk keuntungan mereka. Dalam beberapa tahun terakhir, jumlah serangan malware pada perangkat Android telah meningkat secara signifikan, mengakibatkan kerugian yang besar bagi individu dan organisasi [5]. Sifat terbuka dari ekosistem Android mempermudah penyebaran aplikasi berbahaya melalui berbagai saluran distribusi, termasuk toko aplikasi pihak ketiga, memperkuat kebutuhan akan pendekatan yang canggih dan efektif dalam mendeteksi dan mencegah malware [6]. Studi ini mengeksplorasi pendekatan ensemble dalam machine learning sebagai metode yang berpotensi meningkatkan deteksi malware, dengan tujuan untuk mengurangi risiko keamanan yang meningkat ini dan melindungi integritas serta privasi pengguna telepon pintar.

Di era konektivitas yang tak terpisahkan dari kehidupan sehari-hari, ponsel pintar Android telah menjadi pusat perencanaan, pertukaran data, komunikasi, interaksi sosial, bisnis, dan bahkan transaksi keuangan bagi sebagian besar pengguna [7]. Ponsel cerdas tidak hanya memfasilitasi berbagai aspek kehidupan tetapi juga, dengan peningkatan ketergantungan manusia pada teknologi ini, telah menjadi sasaran serangan siber yang meningkat secara eksponensial. Sistem keamanan Android yang bergantung pada izin yang dideklarasikan oleh pengembang dalam AndroidManifest.xml, membatasi akses ke informasi sensitif seperti GPS dan kontak melalui pembatasan yang ditegakkan selama instalasi aplikasi. Izin ini diklasifikasikan menjadi normal dan berbahaya, di mana izin normal dianggap tidak kritis dan diberikan secara otomatis, sedangkan izin berbahaya, yang mengakses data pribadi dan API penting, memerlukan persetujuan pengguna [8]. Karena aplikasi dapat meminta izin untuk mengakses berbagai fungsi perangkat, ini membuka peluang bagi peretas untuk menggunakan izin tersebut untuk melancarkan serangan atau menyuntikkan malware. Risiko ini diperparah oleh sifat terbuka sistem operasi Android yang meningkatkan kerentanan terhadap serangan yang semakin canggih dan beragam. Di tengah tantangan ini, upaya keamanan yang lebih kuat dan pendekatan deteksi yang progresif menjadi imperatif untuk melindungi privasi dan data pengguna dari ancaman yang terus berkembang ini [9].

Dalam upaya untuk meningkatkan keamanan perangkat Android, dua pendekatan utama digunakan untuk mendeteksi malware: analisis statis dan analisis dinamis [10]. Analisis statis melibatkan berbagai metode yang bertujuan untuk membedakan perilaku runtime dari perangkat lunak sebelum dieksekusi. Tujuan utamanya adalah untuk mengidentifikasi dan menyingkirkan aplikasi yang berpotensi berbahaya sebelum diinstal dan dijalankan oleh pengguna. Meskipun dianggap sebagai metode yang kasar karena kecenderungannya untuk menandai aplikasi sebagai berbahaya berdasarkan perkiraan berlebihan terhadap kemungkinan perilaku runtime, analisis statis tetap penting karena dapat mendeteksi ancaman potensial secara proaktif. Tantangannya adalah memaksimalkan efektivitas deteksi sembari meminimalkan risiko kesalahan positif. Di sisi lain, analisis dinamis menawarkan pendekatan alternatif dengan menjalankan program untuk mempelajari perilakunya dan dampaknya pada lingkungan operasionalnya [11].

Berbeda dengan analisis statis, analisis dinamis mendeteksi pelanggaran keamanan tepat pada saat pelanggaran tersebut terjadi. Namun, pendekatan ini juga memiliki keterbatasan, karena hanya dapat menganalisis satu jalur eksekusi pada suatu waktu, sehingga tidak dapat menangkap semua kemungkinan perilaku program. Selain itu, analisis dinamis umumnya lebih terlambat dalam mendeteksi ancaman karena memerlukan eksekusi sebenarnya dari perangkat lunak yang diuji. Pendekatan keamanan yang komprehensif perlu mempertimbangkan kekuatan dan kelemahan dari kedua metode ini untuk memastikan deteksi malware yang efektif dan efisien, serta melindungi privasi dan data pengguna dari ancaman yang terus berkembang [12].

Penelitian ini bertujuan untuk meningkatkan akurasi deteksi malware pada platform Android dengan menggunakan teknik ensemble machine learning. Teknik ini menggabungkan beberapa model pembelajaran mesin yang berbeda untuk bekerja bersama-sama, dengan harapan dapat menghasilkan performa yang lebih baik dibandingkan dengan model individual. Secara khusus, penelitian ini berfokus pada penggunaan metode ensemble soft voting, di mana prediksi dari berbagai model seperti Random Forest, Gradient Boosting, dan XGBoost digabungkan berdasarkan probabilitas masing-masing. Dengan mengimplementasikan pendekatan ini, penelitian ini bertujuan untuk menunjukkan bahwa metode ensemble dapat mengatasi kelemahan model individual dan memberikan hasil yang lebih akurat dan andal dalam mendeteksi malware. Melalui eksperimen dan analisis komprehensif, penelitian ini berusaha memberikan kontribusi yang signifikan dalam bidang keamanan siber, khususnya dalam deteksi malware pada perangkat Android.

Dalam penelitian [13] memperkenalkan framework pendeteksi malware yang baru untuk Android yang dinamakan NATICUSdroid. Framework ini memanfaatkan izin android sebagai fitur utama untuk mengklasifikasikan aplikasi. Penelitian [13] mengevaluasi delapan algoritma *machine learning* di mana model

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 7 No. 3, Juni 2024
P-ISSN 2620-8342
E-ISSN 2621-3052

berbasis Random Forest menunjukkan kinerja terbaik. Dalam penelitian [14] mengadopsi XGBoost karena kemampuannya yang terbukti dalam menangani dataset besar dengan efisiensi tinggi. Peningkatan dilakukan dengan pemilihan fitur, selain itu dioptimalkan lebih lanjut dengan penggunaan Algoritma Genetik Adaptif (AGA) dan Particle Swarm Optimization (PSO) berhasil mencapai akruasi yang sangat baik.

Dalam penelitian [15] pendekatan Gradient Boosting (GB) menunjukkan potensi signifikan sebagai bagian dari kelompok beragam algoritma *machine learning*. Gradient Boosting (GB) adalah teknik yang kuat yang secara efektif mengoptimalkan model prediktif melalui pembentukan ansambel berurutan dari pohon keputusan yang lemah, setiap pohon berusaha memperbaiki kesalahan dari pohon sebelumnya. Penggunaan GB dalam mendeteksi malware di aplikasi Android menawarkan pendekatan yang bagus, khususnya dalam kombinasi dengan teknik lain untuk meningkatkan *robustness* dan keandalan.

Berbagai penelitian telah mengeksplorasi penggunaan model pembelajaran mesin seperti Random Forest, Gradient Boosting, dan XGBoost dalam deteksi malware di platform Android. Random Forest terkenal dengan kemampuannya untuk mengelola dataset yang besar dan menangani fitur-fitur yang kurang relevan, sementara Gradient Boosting dikenal unggul dalam meningkatkan akurasi melalui proses pembelajaran yang iteratif. XGBoost, varian yang ditingkatkan dari Gradient Boosting, menawarkan kecepatan dan efisiensi yang lebih tinggi, yang telah membuatnya menjadi pilihan populer dalam berbagai kompetisi pembelajaran mesin. Namun, tiap metode memiliki keterbatasan. Random Forest bisa mengalami overfitting jika jumlah pohon dalam model terlalu banyak. Di sisi lain, Gradient Boosting memerlukan waktu komputasi yang lebih lama, yang dapat mempengaruhi efisiensi dalam pemrosesan data skala besar. Sementara itu, XGBoost, meskipun cepat dan efisien, seringkali menawarkan kompleksitas model yang tinggi yang membuat interpretasi hasilnya menjadi lebih menantang.

Teknik ensemble adalah pendekatan yang menggabungkan beberapa model pembelajaran mesin untuk meningkatkan kinerja prediksi [16]. Metode pembelajaran ansambel menggabungkan hasil dari beberapa algoritma pembelajaran mesin untuk menghasilkan prediksi yang lebih akurat daripada yang mungkin dicapai oleh algoritma tunggal. Salah satu strategi kombinasi yang umum dalam pembelajaran ansambel adalah voting, yang terdiri dari dua jenis: hard voting, yang menghitung suara berdasarkan label kelas yang paling sering muncul dari semua algoritma, dan soft voting, yang menggunakan rata-rata tertimbang dari probabilitas kelas untuk menentukan kelas terpilih. Soft voting mengutamakan kelas dengan nilai agregat probabilitas tertinggi, yang ditentukan melalui fungsi Argmax dari probabilitas yang diprediksi. Dalam penelitian ini, kami mengembangkan metode soft voting berbobot dinamis yang memperbaiki pendekatan tradisional dengan mengatur bobot secara dinamis selama proses pelatihan menggunakan prinsip penurunan gradien. Metode ini memungkinkan bobot yang berbeda untuk diberikan kepada model dasar berdasarkan efektivitas relatif mereka dalam konteks fitur yang diekstrak, memastikan bahwa pembelajaran ansambel memanfaatkan kekuatan individu dari setiap model terlibat, dengan fungsi kerugian baru yang dirancang untuk mengoptimalkan kinerja voting berbobot ini [17].

Dengan menggabungkan berbagai model pembelajaran mesin melalui metode ensemble, penelitian ini bertujuan untuk memberikan solusi yang lebih robust dalam menghadapi ancaman malware yang terus berkembang. Pendekatan ini tidak hanya diharapkan dapat meningkatkan akurasi deteksi tetapi juga memberikan model yang lebih tahan terhadap variasi dan kompleksitas serangan malware. Melalui eksperimen dan evaluasi yang komprehensif, penelitian ini berupaya untuk membuktikan keunggulan teknik ensemble dalam konteks keamanan siber, khususnya untuk perangkat Android.

2. Tinjauan Pustaka

2.1. Malware Android

Malware Android adalah perangkat lunak berbahaya yang dirancang untuk mengganggu fungsi perangkat Android dengan melakukan tindakan ilegal. Malware ini dapat dibagi ke dalam beberapa kategori utama, masing-masing dengan karakteristik yang membedakannya dari yang lain. Kategori yang paling umum termasuk adware, backdoor, file infector, aplikasi yang mungkin tidak diinginkan (Potentially Unwanted Applications, PUA), ransomware, riskware, scareware, dan Trojan. Setiap kategori malware memiliki keluarga tersendiri yang berbagi karakteristik dan metode operasi yang serupa. Identifikasi kategori dan keluarga malware sangat penting bagi peneliti keamanan siber dan perusahaan anti-malware untuk mengambil tindakan pencegahan yang cepat dan tepat, sehingga mengurangi kerugian yang ditimbulkan oleh malware tersebut [18].

2.2. Machine Learning

Pembelajaran mesin secara luas mengacu pada proses penyesuaian model prediktif terhadap data atau mengidentifikasi pola informatif dalam kumpulan data. Melalui komputasi, pembelajaran mesin berupaya untuk meniru kemampuan manusia dalam mengenali pola secara objektif. Misalnya, seorang anak yang belum pernah melihat seekor anjing atau kucing mungkin tidak bisa membedakannya hanya berdasarkan deskripsi verbal. Namun, dengan melihat gambar kedua hewan tersebut berulang kali dan diberi tahu hewan mana yang

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 7 No. 3, Juni 2024
P-ISSN 2620-8342
E-ISSN 2621-3052

mana, anak tersebut dapat belajar membedakan anjing dari kucing. Serupa dengan itu, komputer dapat diajarkan untuk melakukan tugas klasifikasi melalui pembelajaran mesin dengan diberi data yang besar dan algoritma analitis. Pembelajaran mesin adalah subbidang dari AI yang berfokus pada cara komputer belajar dari kumpulan data besar. Bidang ini muncul di persimpangan statistik dan ilmu komputer untuk mempelajari hubungan dalam data dengan menggunakan algoritma komputasi yang efisien. Jaringan saraf, salah satu algoritma pembelajaran mesin, menggunakan model yang meniru neuron di otak manusia. Pembelajaran mendalam, atau deep learning, adalah jenis algoritma yang menggunakan beberapa lapisan jaringan saraf (disebut jaringan saraf dalam) untuk memungkinkan pembelajaran dan analisis yang lebih mendalam dan kompleks. Ini memungkinkan komputer untuk menentukan pola secara otomatis tanpa definisi manual dari manusia dan saat ini merupakan teknik utama yang digunakan dalam jaringan saraf [19].

2.3. Random Forest

Random Forest (RF) adalah salah satu algoritma metode ensemble yang paling populer dalam pembelajaran mesin. Algoritma ini menggunakan pohon keputusan sebagai komponen dasar dan menggabungkan mereka menggunakan teknik suara mayoritas untuk membuat prediksi akhir. Keberhasilan RF terletak pada keragaman dan akurasi pohon-pohonnya, yang dicapai melalui dua prinsip utama: Random Subspaces dan Bagging. Prinsip Random Subspaces melibatkan pemilihan subset fitur secara acak di setiap node selama konstruksi pohon sedangkan *Bagging* memilih subset sampel secara acak dengan penggantian untuk digunakan di setiap pohon. Namun, kelemahan utama RF adalah bahwa menghasilkan banyak pohon yang membutuhkan ruang penyimpanan yang besar dan waktu eksekusi yang lama. Selain itu, pada titik tertentu, menambah jumlah pohon tidak lagi meningkatkan kinerja, melainkan dapat mengurangi kemampuan generalisasi model. Untuk mengatasi masalah ini, beberapa peneliti telah mengembangkan teknik pemangkasan RF untuk menemukan subset pohon yang optimal. Pendekatan pemangkasan ini dapat dilakukan secara statis atau dinamis selama pembangunan hutan [20].

2.4. Gradient Boosting

Gradient boosting adalah algoritma pembelajaran mesin yang memanfaatkan teknik boosting untuk membentuk ensemble yang kuat dan akurat. Algoritma ini biasanya menggunakan pohon keputusan sebagai pembelajar dasar, sehingga sering disebut sebagai pohon keputusan dengan peningkatan gradien (*Gradient Boosted Decision Trees, GBDT)*. Teknik gradient boosting pertama kali diperkenalkan oleh Breiman, yang mencatat bahwa boosting dapat dilihat sebagai teknik optimasi pada fungsi kerugian yang sesuai. Kemudian, versi yang lebih maju dari algoritma gradient boosting dikembangkan oleh Friedman. Proses pembelajaran dalam algoritma ini melibatkan pelatihan model baru secara berurutan untuk menghasilkan pengklasifikasi yang lebih baik. Meskipun mirip dengan teknik boosting lainnya, inti dari gradient boosting adalah mengembangkan pembelajar dasar yang sangat berkorelasi dengan gradien negatif dari fungsi kerugian yang terkait dengan keseluruhan ensemble. Algoritma gradient boosting bekerja dengan iterasi yang dimulai dari model dasar sederhana dan secara bertahap menambahkan model baru yang mengoreksi kesalahan dari model sebelumnya. Setiap model baru dilatih untuk memperbaiki sisa (residual) dari prediksi yang dihasilkan oleh model-model sebelumnya. Dengan cara ini, setiap model baru berkontribusi untuk mengurangi kesalahan prediksi secara keseluruhan, sehingga meningkatkan kinerja ensemble secara bertahap. Prinsip dasar dari gradient boosting adalah sequential learning, di mana model baru ditambahkan satu per satu dan setiap model baru dilatih untuk memperbaiki kesalahan dari model sebelumnya; gradient descent optimization, di mana setiap model baru dikembangkan untuk mengurangi gradien negatif dari fungsi kerugian; dan additive model, di mana model akhir adalah kombinasi dari semua model yang telah dilatih. Gradient boosting dikenal karena kemampuannya dalam menangani data yang kompleks dan menghasilkan model yang sangat akurat, menjadikannya salah satu teknik pembelajaran mesin yang paling kuat dan serbaguna [21].

2.5. XGBoost

XGBoost adalah algoritma ensemble berbasis pohon keputusan yang menggunakan kerangka kerja gradient boosting. Algoritma ini dikenal karena skalabilitas dan akurasinya yang tinggi, serta digunakan secara luas dalam aplikasi klasifikasi dan regresi. XGBoost telah mendominasi bidang pembelajaran mesin terapan dan memenangkan banyak kompetisi Kaggle. Dikembangkan oleh Chen dan Guestrin pada tahun 2016, XGBoost memiliki beberapa keunggulan dibandingkan dengan algoritma gradient boosting konvensional. Salah satu perbedaan utamanya adalah fungsi kerugian XGBoost yang mencakup istilah regularisasi, yang bertujuan untuk mencegah overfitting. XGBoost bekerja dengan cara yang mirip dengan algoritma gradient boosting lainnya, yaitu membangun model secara berurutan untuk memperbaiki kesalahan dari model sebelumnya.

XGBoost memperkenalkan sejumlah optimasi dan peningkatan performa. Algoritma ini menggunakan pendekatan regularisasi yang lebih kuat, baik dari segi L1 (lasso) maupun L2 (ridge), untuk mengendalikan kompleksitas model dan mencegah overfitting. Selain itu, XGBoost menerapkan teknik komputasi yang efisien, seperti paralelisasi untuk mempercepat proses pelatihan dan pengoptimalan penggunaan memori [21].

2.6. Ensemble Learning

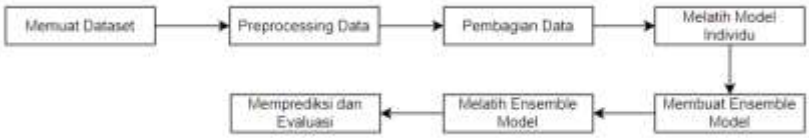
Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 7 No. 3, Juni 2024
P-ISSN 2620-8342
E-ISSN 2621-3052

Ensemble learning adalah metode pembelajaran mesin yang melibatkan pelatihan beberapa model dasar dan menggabungkan prediksi mereka untuk meningkatkan kinerja dan kemampuan generalisasi dibandingkan dengan model tunggal. Konsep utama di balik ensemble learning adalah pengakuan bahwa setiap model pembelajaran mesin memiliki keterbatasan dan rentan terhadap kesalahan. Dengan menggabungkan kekuatan dari berbagai model dasar, ensemble learning bertujuan untuk meningkatkan akurasi klasifikasi. Beberapa kelemahan dari model pembelajaran mesin tunggal termasuk varians tinggi, bias tinggi, dan akurasi rendah.

Penelitian telah menunjukkan bahwa model ensemble sering mencapai akurasi yang lebih tinggi daripada model tunggal. Metode ensemble dapat mengurangi varians dan kesalahan bias yang terkait dengan model tunggal; misalnya, teknik bagging mengurangi varians tanpa meningkatkan bias, sementara boosting mengurangi bias. Secara keseluruhan, pengklasifikasi ensemble lebih tangguh dan berkinerja lebih baik dibandingkan dengan model individual, menjadikannya pilihan yang kuat dalam pembelajaran mesin [21].

3. Metode Penelitian

Pendekatan yang digunakan dalam penelitian ini melibatkan serangkaian langkah sistematis yang bertujuan untuk meningkatkan akurasi deteksi malware pada platform Android. Proses ini dimulai dengan pemilihan dan preprocessing data, dilanjutkan dengan pelatihan model individual, pembentukan model ensemble, dan evaluasi kinerja model. Dengan menggunakan berbagai teknik *machine learning* dan metodel *ensemble*, penelitian ini berupaya untuk mengidentifikasi pendekatan terbaik dalam mendeteksi malware secara efektif, seperti terlihat pada gambar 1.



Gambar 1. Metodologi Penelitian

Proses metodologi penelitian ini terdiri dari beberapa langkah seperti terlihat pada gambar 1. Berikut adalah penjelasan dari setiap langkah yang diambil :

a. Memuat Dataset

Kumpulan data yang tersedia untuk umum ini, yang dikenal sebagai KronoDroid [22], merupakan dataset Android berfitur hibrid terbesar dan satu-satunya yang menyediakan data dengan stempel waktu, mencakup lebih dari 209 keluarga malware Android. Kumpulan data emulator terdiri dari 28.745 aplikasi berbahaya dari 209 kelompok malware dan 35.246 sampel jinak, sementara kumpulan data perangkat nyata berisi 41.382 malware yang termasuk dalam 240 keluarga malware, serta 36.755 aplikasi jinak. Namun, dalam penelitian ini, fokus diberikan pada fitur izin yang sering digunakan oleh aplikasi, karena izin-izin ini merupakan indikator penting dalam deteksi malware. Dengan memfokuskan pada fitur izin yang sering digunakan, penelitian ini bertujuan untuk meningkatkan efektivitas dan efisiensi sistem deteksi malware. Fitur izin ini mencakup berbagai izin yang diminta oleh aplikasi untuk mengakses fungsi perangkat yang sensitif, dan dengan demikian dapat menjadi petunjuk penting dalam mengidentifikasi aplikasi berbahaya. Melalui pendekatan ini, diharapkan dapat mengembangkan model deteksi malware yang lebih kuat dan akurat, yang mampu beradaptasi dengan evolusi ancaman siber yang terus berkembang.

b. Preprocessing Data

Setelah dataset dimuat, langkah selanjutnya adalah preprocessing data, yang merupakan tahap penting dalam memastikan kualitas data sebelum melatih model. Proses ini mencakup identifikasi dan penanganan nilai yang hilang, yang dapat menyebabkan bias dalam hasil model jika tidak ditangani dengan benar. Teknik imputasi yang sesuai diterapkan untuk menggantikan nilai-nilai yang hilang dengan estimasi berdasarkan data yang ada, memastikan integritas dataset tetap terjaga. Selanjutnya, normalisasi data dilakukan untuk memastikan bahwa semua fitur berada pada skala yang seragam, yang penting untuk mencegah dominasi fitur dengan skala besar terhadap fitur lainnya selama pelatihan model. Normalisasi dilakukan dengan menskalakan fitur ke rentang tertentu atau dengan menggunakan metode standar seperti z-score. Tahap preprocessing ini memastikan bahwa dataset yang digunakan untuk pelatihan dan pengujian model bebas dari masalah yang dapat mempengaruhi performa model, sehingga meningkatkan akurasi dan keandalan deteksi malware pada platform Android.

c. Pembagian Data

Data yang telah diprepreses kemudian dibagi menjadi set pelatihan dan pengujian menggunakan metode train-test split dengan proporsi 80:20, di mana 80% data digunakan untuk pelatihan dan 20% sisanya digunakan untuk pengujian. Pembagian ini penting untuk memungkinkan evaluasi model yang tidak bias, di mana model

dilatih pada subset data yang lebih besar dan diuji pada subset data yang terpisah. Dengan membagi data dalam proporsi ini, kita dapat menilai kinerja model secara objektif pada data yang belum pernah dilihat selama proses pelatihan. Set pelatihan digunakan untuk melatih model, memungkinkan algoritma pembelajaran mesin untuk mempelajari pola dan hubungan dalam data. Sementara itu, set pengujian digunakan untuk mengevaluasi performa model, memberikan indikasi seberapa baik model dapat menggeneralisasi dari data pelatihan ke data yang baru. Pendekatan ini memastikan bahwa hasil evaluasi memberikan gambaran yang akurat tentang kemampuan model dalam mendeteksi malware pada data yang belum pernah dilihat sebelumnya, sehingga membantu dalam menilai keandalan dan efektivitas model yang dikembangkan.

d. Melatih Model Individu

Setelah proses preprocessing data selesai dan data telah dibagi menjadi set pelatihan dan pengujian, langkah berikutnya adalah melatih model-model individu. Dalam penelitian ini, tiga model pembelajaran mesin digunakan: Random Forest, Gradient Boosting, dan XGBoost. Setiap model dilatih secara terpisah menggunakan data pelatihan untuk mempelajari pola dan hubungan dalam data yang dapat membantu dalam deteksi malware. Random Forest menggunakan banyak pohon keputusan yang digabungkan untuk menghasilkan prediksi akhir, sementara Gradient Boosting membangun pohon keputusan secara berurutan untuk memperbaiki kesalahan dari pohon sebelumnya, dan XGBoost menawarkan kecepatan dan efisiensi yang lebih tinggi dengan menggunakan teknik regularisasi untuk mencegah overfitting. Setelah pelatihan, model-model ini dievaluasi menggunakan set pengujian untuk mengukur performa mereka berdasarkan metrik seperti akurasi, precision, recall, dan F1-score, yang memberikan wawasan mengenai kekuatan dan kelemahan masing-masing model sebelum menggabungkannya menggunakan teknik ensemble untuk mencapai hasil prediksi yang lebih akurat dan andal.

e. Membuat Ensemble Model

Ensemble model yang telah dibuat kemudian dilatih menggunakan data pelatihan untuk mengintegrasikan kekuatan dari setiap model individu dan mengurangi kelemahan masing-masing. Dalam penelitian ini, teknik soft voting digunakan untuk menggabungkan hasil prediksi dari model Random Forest, Gradient Boosting, dan XGBoost. Soft voting bekerja dengan menggabungkan probabilitas prediksi dari setiap model individu dan menghasilkan prediksi akhir berdasarkan rata-rata tertimbang dari probabilitas tersebut. Dengan cara ini, ensemble model mampu memanfaatkan keunggulan masing-masing model individu dan mengurangi dampak kelemahan mereka, sehingga meningkatkan akurasi dan keandalan prediksi secara keseluruhan. Pendekatan ini memastikan bahwa hasil prediksi lebih stabil dan robust, memberikan model yang lebih efektif dalam mendeteksi malware pada platform Android.

f. Memprediksi dan Evaluasi

Langkah terakhir adalah memprediksi kelas malware pada data pengujian menggunakan ensemble model yang telah dilatih. Kinerja model dievaluasi dengan menghitung akurasi dan membuat laporan klasifikasi yang mencakup metrik seperti precision, recall, dan f1-score. Selain itu, teknik cross-validation lima kali lipat digunakan untuk menilai keefektifan model secara lebih mendalam. Cross-validation membagi data menjadi lima subset, melatih model pada empat subset dan mengujinya pada subset yang tersisa, proses ini diulang lima kali sehingga setiap subset digunakan sekali sebagai data pengujian. Evaluasi ini bertujuan untuk menilai efektivitas model dalam mendeteksi malware secara akurat dan memastikan bahwa model tidak overfit atau underfit. Akurasi mengukur seberapa sering model membuat prediksi yang benar, sementara precision, recall, dan f1-score memberikan wawasan yang lebih mendalam tentang kinerja model dalam konteks klasifikasi biner, menunjukkan keseimbangan antara mendeteksi malware (true positives) dan menghindari alarm palsu (false positives). Hasil cross-validation menunjukkan konsistensi dan keandalan model, memberikan gambaran yang lebih akurat tentang kemampuan model dalam situasi dunia nyata. Melalui evaluasi komprehensif ini, kita dapat memastikan bahwa ensemble model yang dikembangkan tidak hanya akurat tetapi juga andal dan konsisten dalam mendeteksi malware pada platform Android.

4. Hasil dan Pembahasan

Pada bagian ini, mempresentasikan hasil dari berbagai eksperimen yang dilakukan menggunakan model *machine learning* individual dan *ensemble* model. Hasil eksperimen ini mencakup metrik kinerja seperti akurasi, precision, recall, dan f1-score. Kami juga membandingkan performa model individual dengan *ensemble* model untuk menilai peningkatan akurasi yang diperoleh melalui teknik soft voting. Selain itu, kami melakukan analisis mendalam mengenai mengapa metode ensemble memberikan hasil yang lebih baik dibandingkan model individual.

4.1 Individual Models

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 7 No. 3, Juni 2024
P-ISSN 2620-8342
E-ISSN 2621-3052

Penelitian ini dimulai dengan melatih tiga model individual yaitu Random Forest, Gradient Boosting, dan XGBoost. Setiap model dievaluasi berdasarkan metrik akurasi, precision, recall, dan f1-score untuk menilai performa mereka dalam mendeteksi malware pada plafform Android.

Tabel 1. Eksperimen Individual Models

Model	Akurasi	Precision	Recall	F1-Score
Random Forest	85 %	0.86	0.84	0.85
Gradient Boosting	84 %	0.85	0.83	0.84
XGBoost	85 %	0.86	0.84	0.85

Tabel 1 menunjukkan bahwa model Random Forest dan XGBoost memberikan hasil yang serupa dengan akurasi masing-masing 85%, sementara Gradient Boosting sedikit lebih rendah dengan akurasi 84%. Meskipun akurasi dan F1-Score dari Random Forest dan XGBoost adalah sama, setiap model memiliki kekuatan unik yang dapat dimanfaatkan melalui pendekatan ensemble.

4.2 Ensemble Model

Untuk meningkatkan performa deteksi malware, model-model individual tersebut digabungkan menggunakan teknik soft voting dalam *ensemble* model. Hasil dari *ensemble* model dibandingkan dengan hasil model individual seperti ditunjukkan dalam tabel berikut.

Tabel 2. Eksperimen Models

Model	Akurasi	Precision	Recall	F1-Score
Random Forest	85 %	0.86	0.84	0.85
Gradient Boosting	84 %	0.85	0.83	0.84
XGBoost	85 %	0.86	0.84	0.85
Ensemble (Soft Voting)	90 %	0.91	0.89	0.90

Dari tabel tersebut, terlihat bahwa *ensemble* model memberikan peningkatan akurasi yang signifikan hingga 90%, serta peningkatan metrik precision, recall, dan F1-Score dibandingkan dengan model individual.

4.3 Cross Validation

Dalam penelitian ini, teknik cross-validation lima kali lipat digunakan untuk menilai *ensemble* model, yang terdiri dari Random Forest, Gradient Boosting, dan XGBoost. Hasil cross-validation menunjukkan bahwa model *ensemble* ini memiliki akurasi rata-rata 89.99 dengan deviasi standar 0.00.

Tabel 3. Hasil Cross-Validation

Fold	Akurasi	Deviation Standard
1	89.65 %	0.00%
2	89.95%	0.00%
3	90.07%	0.00%
4	90.04%	0.00%
5	90.22%	0.00%
Average	89.99%	0.19%

Dalam penelitian ini, teknik cross-validation lima kali lipat digunakan untuk menilai keefektifan model *ensemble* yang terdiri dari Random Forest, Gradient Boosting, dan XGBoost dalam mendeteksi malware pada platform Android. Hasil dari cross-validation menunjukkan bahwa model tersebut secara konsisten mencapai akurasi tinggi di semua fold, dan skor individu masing-masing fold berkisar antara 89.65% hingga 90.22%, dan rata-rata akurasi keseluruhan mencapai 89.99%. menariknya, deviasi standar di setiap fold adalah 0.00%, menandakan bahwa hasil model sangat stabil dan hommogen dalam setiap fold terpisa. Namun, ketika melihat keseluruhan hasil dari semua fold, ditemukan deviasi standar keseluruhan sebesar 0.19%, yang mengindikasi adanya variasi minor dalam performa model antar fold yang berbeda.

Deviasi standar yang sangat rendah di setiap fold individual mengimplikasi bahwa model memiliki kemampuan prediksi yang sangat konsisten dalam kondisi data yang serupa. Ini menunjukkan bahwa model ini sangat efektif dalam mengatasi fluktuasi kecil dalam data pengujian dan dapat diandalkan dalam menghasilkan prediksi yang akurat dalam lingkungan yang terkontrol dengan baik. Sementara itu, deviasi standar yang sedikit lebih tinggi saat dilihat dari keseluruhan fold menunjukkan bahwa ada sedikit perbedaan dalam cara model merespons kepada variasi data yang lebih luas antar fold. Hal ini bisa diinterpretasikan sebagai indikasi bahwa walaupun model cukup robust, masih ada ruang untuk peningkatan dalam hal menggeneralisasi lebih baik lagi terhadap data yang lebih bervariasi.

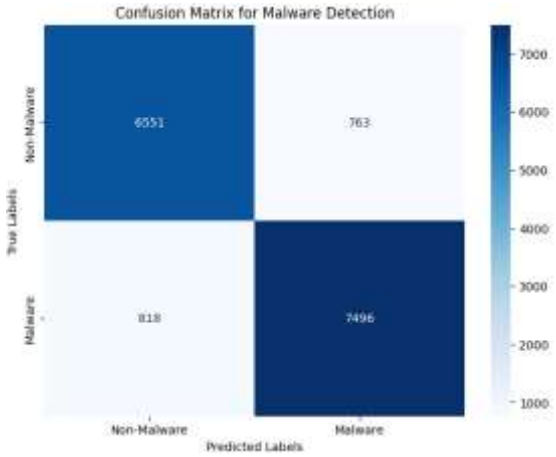
4.4 Analisa Hasil

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 7 No. 3, Juni 2024
P-ISSN 2620-8342
E-ISSN 2621-3052

Peningkatan akurasi dari 85% menjadi 90% dengan menggunakan *ensemble* model menunjukkan bahwa teknik soft voting efektif dalam menggabungkan kekuatan dari setiap model individual dan mengurangi kelemahan masing-masing. Pendekatan ini menghasilkan modle yang lebih robust dan andal dalam mendeteksi malware. Metode *ensemble* memberikan hasil yang lebih baik karena mampu mengkobinasikan berbagai perspektif dari model yang berbeda. Random Forest, dengan kammpuannya dalam menangani data berukuran besar dan fitur-fitur yang kompleks, Gradient Boosting yang unggul dalam menangkap interaksi antara fitur-fitur, serta XGBoost yang dikenal efisien dalam pelatihan dan memiliki regularisasi untuk mencegah overfitting, bersama-sama memberikan hasil yang lebih akurat.

Analisis hasil cross-validation lima kali lipat menunjukkan bahwa model ini tidak hanya mencapai akurasi tinggi tetapi juga menunjukkan kestabilan yang luar biasa dalam performanya di berbagai kondisi pengujian. Dengan deviasi standar rata-rata yang hanya 0.19%, hasil ini menegaskan bahwa model ensemble memiliki konsistensi yang tinggi, menandakan kehandalan dalam kondisi penggunaan yang beragam. Fakta ini memperkuat kepercayaan bahwa pendekatan ensemble dapat efektif dalam skenario dunia nyata, di mana variasi data yang ditemui bisa lebih besar.

Dalam evaluasi model machine learning, terutama dalam aplikasi yang berhubungan dengan keamanan seperti deteksi malware, penting untuk memahami tidak hanya akurasi keseluruhan, tetapi juga bagaimana model melakukan klasifikasi di berbagai kelas. Confusion matrix merupakan alat analisis yang kritikal dalam mengungkapkan kinerja model lebih dalam dengan memvisualisasikan frekuensi kesalahan dan keberhasilan prediksi lintas kategori. Gambar 2 berikut menampilkan confusion matrix yang dihasilkan dari model deteksi malware kami, memberikan wawasan mendalam tentang kemampuan model dalam membedakan antara 'Malware' dan 'Non-Malware'.



Gambar 2. Confusion Matrix Model Deteksi Malware

Dalam evaluasi model deteksi malware, confusion matrix menyediakan wawasan yang berharga tentang performa model dalam mengklasifikasikan malware dan non-malware. Model ini berhasil mengidentifikasi 7.496 dari 8.314 kasus malware yang sebenarnya (*True Positives*), dan 6.551 dari 7.314 kasus non-malware yang sebenarnya (*True Negatives*), menunjukkan efektivitas yang tinggi dalam mendeteksi kedua kategori. Nilai ini menghasilkan tingkat sensitivitas atau recall sebesar 90% untuk malware, yang menunjukkan bahwa model sangat kompeten dalam mengidentifikasi malware yang ada. Secara paralel, spesifisitas model juga mencapai 90%, yang mengindikasikan kemampuan yang baik dalam menghindari alarm palsu, mengurangi potensi gangguan yang tidak perlu pada pengguna atau sistem.

Namun, model masih menghasilkan 818 False Negatives, di mana malware tidak terdeteksi, dan 763 False Positives, di mana non-malware salah diidentifikasi sebagai malware. Tingkat False Negative ini menekankan pentingnya peningkatan lebih lanjut pada model untuk menangkap semua ancaman potensial, mengingat bahwa setiap malware yang tidak terdeteksi dapat menimbulkan risiko serius. Sementara itu, jumlah False Positives menggambarkan area lain yang membutuhkan perbaikan untuk mengoptimalkan pengalaman pengguna dan efisiensi operasional, mengurangi insiden yang bisa mengganggu kegiatan normal pengguna karena alarm yang tidak perlu.

Dengan precision sekitar 91% untuk kategori malware, model ini memperlihatkan keandalan tinggi dalam prediksi yang positif; sebagian besar label positif yang diberikan oleh model memang merepresentasikan malware. F1-score dari model, yang mencapai sekitar 0.90, mengonfirmasi keseimbangan yang baik antara precision dan recall, menandakan bahwa model tersebut cukup kuat dalam berbagai aspek kinerjanya.

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 7 No. 3, Juni 2024
P-ISSN 2620-8342
E-ISSN 2621-3052

Pendekatan *ensemble* yang digunakan dalam penelitian ini memiliki beberapa kelebihan utama seperti meningkatkan akurasi deteksi malware secara signifikan dibandingkan model individual. Kombinasi model membantu dalam mengurangi overfitting yang mungkin terjadi pada model individual. Serta *ensemble* model lebih tahan terhadap variabilitas data dan lebih stabil dalam performa. Namun penelitian ini juga memiliki keterbatasan dalam melatih dan menggabungkan beberapa model memerlukan sumber daya komputasi yang lebih tinggi dibandingkan dengan menggunakan satu model saja. Implementasi *ensemble* model lebih kompleks dan memerlukan pemahaman mendalam tentang setiap model yang digunakan serta bagaimana berinteraksi dalam *ensemble*.

Secara teoritis, metode *ensemble* memberikan hasil yang lebih baik karena mampu menggabungkan kelebihan dari setiap model individual. Setiap model dapat menangkap pola dan anomali yang mungkin tidak terdeteksi oleh model lain. *Soft Voting* memungkinkan penggunaan probabilitas prediksi dari setiap model untuk membuat keputusan akhir yang leibh informatif dan akurat. Implikasi praktif dari hasil penelitian ini adalah bahwa pendekatan *ensemble* dapat digunakan secara efektif dalam sistem deteksi malware untuk memberikan perlindungan yang lebih baik bagi pengguna Android.

5. Kesimpulan

Penelitian ini berhasil menunjukkan bahwa penggunaan teknik *ensemble* melalui soft voting dapat secara signifikan meningkatkan akurasi deteksi malware pada platform Android. Dari eksperimen yang dilakukan, model *ensemble* yang menggabungkan Random Forest, Gradient Boosting, dan XGBoost menunjukkan peningkatan akurasi dari 85% pada model terbaik secara individual menjadi 90% pada *ensemble*. Peningkatan ini mengindikasikan bahwa kombinasi prediksi dari berbagai model dengan teknik soft voting memberikan pendekatan yang lebih robust dan akurat dibandingkan dengan menggunakan model tunggal. Penelitian ini memberikan bukti kuat bahwa teknik *ensemble* dalam *machine learning* merupakan stragetgi yang efektif untuk meningkatkan deteksi malware, memberikan kontribusi penting dalam peningkatan keamanan siber pada perangkat mobile.

Daftar Pustaka

[1] J. Senanayake, H. Kalutarage, and M. O. Al-Kadri, "Android mobile malware detection using machine learning: A systematic review," *Electronics (Switzerland)*, vol. 10, no. 13. MDPI AG, Jul. 01, 2021. doi: 10.3390/electronics10131606.

[2] H. Bakır, "VoteDroid: a new ensemble voting classifier for malware detection based on fine-tuned deep learning models," *Multimed Tools Appl*, May 2024, doi: 10.1007/s11042-024-19390-7.

[3] P. Kumar and S. Singh, "Security Testing of Android Apps Using Malware Analysis and XGboost Optimized by Adaptive Particle Swarm Optimization," *SN Comput Sci*, vol. 5, no. 1, Jan. 2024, doi: 10.1007/s42979-023-02411-x.

[4] S. Bashir, F. Maqbool, F. H. Khan, and A. S. Abid, "Hybrid machine learning model for malware analysis in android apps," *Pervasive Mob Comput*, vol. 97, Jan. 2024, doi: 10.1016/j.pmcj.2023.101859.

[5] L. Meijin *et al.*, "A Systematic Overview of Android Malware Detection," *Applied Artificial Intelligence*, vol. 36, no. 1. Taylor and Francis Ltd., 2022. doi: 10.1080/08839514.2021.2007327.

[6] A. Alzubaidi, "Recent Advances in Android Mobile Malware Detection: A Systematic Literature Review," *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 146318–146349, 2021. doi: 10.1109/ACCESS.2021.3123187.

[7] S. I. Imtiaz, S. ur Rehman, A. R. Javed, Z. Jalil, X. Liu, and W. S. Alnumay, "DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network," *Future Generation Computer Systems*, vol. 115, pp. 844–856, Feb. 2021, doi: 10.1016/j.future.2020.10.008.

[8] A. Ehsan, C. Catal, and A. Mishra, "Detecting Malware by Analyzing App Permissions on Android Platform: A Systematic Literature Review," *Sensors*, vol. 22, no. 20. MDPI, Oct. 01, 2022. doi: 10.3390/s22207928.

[9] T. Sharma and D. Rattan, "Malicious application detection in android - A systematic literature review," *Computer Science Review*, vol. 40. Elsevier Ireland Ltd, May 01, 2021. doi: 10.1016/j.cosrev.2021.100373.

[10] A. Muzaffar, H. Ragab Hassen, M. A. Lones, and H. Zantout, "An in-depth review of machine learning based Android malware detection," *Computers and Security*, vol. 121. Elsevier Ltd, Oct. 01, 2022. doi: 10.1016/j.cose.2022.102833.

[11] A. Razgallah, R. Khoury, S. Hallé, and K. Khanmohammadi, "A survey of malware detection in Android apps: Recommendations and perspectives for future research," *Computer Science Review*, vol. 39. Elsevier Ireland Ltd, Feb. 01, 2021. doi: 10.1016/j.cosrev.2020.100358.

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 7 No. 3, Juni 2024
P-ISSN 2620-8342
E-ISSN 2621-3052

[12] H. Haidros Rahima Manzil and S. Manohar Naik, "Detection approaches for android malware: Taxonomy and review analysis," *Expert Systems with Applications*, vol. 238. Elsevier Ltd, Mar. 15, 2024. doi: 10.1016/j.eswa.2023.122255.
[13] A. Mathur, L. M. Podila, K. Kulkarni, Q. Niyaz, and A. Y. Javaid, "NATICUSdroid: A malware detection framework for Android using native and custom permissions," *Journal of Information Security and Applications*, vol. 58, May 2021, doi: 10.1016/j.jisa.2020.102696.
[14] P. Tarwireyi, A. Terzoli, and M. O. Adigun, "Assessment of The Detection Capacity of Normalized Gammachirp Cepstral Coefficients In Android Malware," *Indian Journal of Computer Science and Engineering*, vol. 13, no. 6, pp. 1809–1821, 2022, doi: 10.21817/indjcse/2022/v13i6/221306102.
[15] L. Hammood, İ. A. Doğru, and K. Kılıç, "Machine Learning-Based Adaptive Genetic Algorithm for Android Malware Detection in Auto-Driving Vehicles," *Applied Sciences (Switzerland)*, vol. 13, no. 9, May 2023, doi: 10.3390/app13095403.
[16] H. Zhu, Y. Li, R. Li, J. Li, Z. You, and H. Song, "SEDMDroid: An Enhanced Stacking Ensemble Framework for Android Malware Detection," *IEEE Trans Netw Sci Eng*, vol. 8, no. 2, pp. 984–994, Apr. 2021, doi: 10.1109/TNSE.2020.2996379.
[17] T. Peng *et al.*, "A Lightweight Multi-Source Fast Android Malware Detection Model," *Applied Sciences (Switzerland)*, vol. 12, no. 11, Jun. 2022, doi: 10.3390/app12115394.
[18] A. H. El Fiky, M. A. Madkour, and A. El Shenawy, "Android Malware Category and Family Identification Using Parallel Machine Learning," *Journal of Information Technology Management*, vol. 14, no. 4, pp. 19–39, 2022, doi: 10.22059/jitm.2022.88133.
[19] J. Komuro, D. Kusumoto, H. Hashimoto, and S. Yuasa, "Machine learning in cardiology: Clinical application and basic research," *Journal of Cardiology*, vol. 82, no. 2. Japanese College of Cardiology (Nippon-Sinzobyo-Gakkai), pp. 128–133, Aug. 01, 2023. doi: 10.1016/j.jjcc.2023.04.020.
[20] Y. Manzali and M. Elfar, "Random Forest Pruning Techniques: A Recent Review," *Operations Research Forum*, vol. 4, no. 2. Springer International Publishing, Jun. 01, 2023. doi: 10.1007/s43069-023-00223-6.
[21] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," *IEEE Access*, vol. 10. Institute of Electrical and Electronics Engineers Inc., pp. 99129–99149, 2022. doi: 10.1109/ACCESS.2022.3207287.
[22] A. Guerra-Manzanares, H. Bahsi, and S. Nõmm, "KronoDroid: Time-based hybrid-featured dataset for effective android malware detection and characterization," *Comput Secur*, vol. 110, Nov. 2021, doi: 10.1016/j.cose.2021.102399.