Jurnal Informatika dan Rekayasa Perangkat Lunak ISSN 2656-2855
Vol. 6, No. 2, September 2024, Hal. 448-455 e-ISSN 2685-5518

Model Hybrid Random Forest dan Information Gain untuk meningkatkan Performa Algoritma Machine Learning pada Deteksi Malicious Software

Fauzi Adi Rafrastara[1], **Wildanil Ghozi**[2*], **Ramadhan Rakhmat Sani**[3], **L. Budi Handoko**[4]
[1,2,4] Jurusan Teknik Informatika , Fakultas Ilmu Komputer, Unviersitas Dian Nuswantoro
[3] Jurusan Sistem Informasi , Fakultas Ilmu Komputer, Unviersitas Dian Nuswantoro
[*]Email: wildanil.ghozi@dsn.dinus.ac.id

Abstrak

Evolusi malware atau perangkat lunak berbahaya semakin meningkatkan kekhawatiran, menyerang tidak hanya komputer tetapi juga perangkat lain seperti smartphone. Malware kini tidak hanya berbentuk monomorfik, tetapi telah berkembang menjadi bentuk polimorfik, metamorfik, hingga oligomorfik. Dengan perkembangan massif ini, perangkat lunak antivirus konvensional tidak akan mampu mengatasinya dengan baik. Hal ini disebabkan oleh kemampuan malware untuk menyebarkan dirinya dengan pola sidik jari dan perilaku yang berbeda. Oleh karena itu, diperlukan antivirus cerdas berbasis machine learning yang mampu mendeteksi malware berdasarkan perilaku bukan sidik jari. Penelitian ini berfokus pada implementasi model machine learning dalam deteksi malware dengan menggunakan algoritma ensemble dan seleksi fitur untuk mencapai kinerja yang baik. Algoritma ensemble yang digunakan adalah Random Forest, dievaluasi dan dibandingkan dengan k-Nearest Neighbor dan Decision Tree sebagai state-of-the-art. Untuk meningkatkan kinerja klasifikasi dalam hal kecepatan proses, metode seleksi fitur yang diterapkan adalah Information Gain dengan 22 fitur. Hasil tertinggi dicapai dengan menggunakan algoritma Random Forest dan metode seleksi fitur Information Gain, mencapai skor 99.0% untuk akurasi dan F1-Score. Dengan mengurangi jumlah fitur, kecepatan pemrosesan dapat ditingkatkan hingga hampir 5 kali lipat.

***Kata kunci**: deteksi malware, random forest, information gain, machine learning*

Abstract

The evolution of malware, or malicious software, has raised increasing concerns, targeting not only computers but also other devices like smartphones. Malware is no longer just monomorphic but has evolved into polymorphic, metamorphic, and oligomorphic forms. With this massive development, conventional antivirus software is becoming less effective at countering it. This is due to malware's ability to propagate itself using different fingerprint and behavioral patterns. Therefore, an intelligent machine learning-based antivirus is needed, capable of detecting malware based on behavior rather than fingerprints. This research focuses on the implementation of a machine learning model for malware detection using ensemble algorithms and feature selection to achieve optimal performance. The ensemble algorithm used is Random Forest, evaluated and compared with k-Nearest Neighbor and Decision Tree as state-of-the-art methods. To enhance classification performance in terms of processing speed, the feature selection method applied is Information Gain, with 22 features. The highest results were achieved using the Random Forest algorithm and Information Gain feature selection method, reaching a score of 99.0% for accuracy and F1-Score. By reducing the number of features, processing speed can be increased by almost fivefold.

Keywords: malware datection, random forest, information gain, machine learning

PENDAHULUAN

Malware, singkatan dari “malicious software”, merujuk pada berbagai jenis perangkat lunak berbahaya yang dirancang untuk merusak, mengganggu, atau mendapatkan akses tidak sah ke sistem komputer. Sejak awal kemunculannya pada era 1970-an dan 1980-an, perkembangan malware telah mengalami evolusi yang signifikan, sejalan dengan perkembangan teknologi informasi dan komunikasi. Awalnya, malware seperti virus komputer pertama, “Creeper”, yang dibuat sebagai eksperimen, hingga worm “Morris” yang tidak disengaja menyebabkan kerusakan besar, adalah contoh awal dari ancaman ini (Alenezi et al., 2022).

Namun, sejalan dengan berkembangnya teknologi, malware menjadi lebih canggih dan berbahaya. Pada era 2000-an, dunia menyaksikan munculnya berbagai jenis malware baru seperti trojan, spyware, ransomware, dan adware (Yadav and Gupta, 2022). Perkembangan ini dipicu oleh peningkatan konektivitas internet dan digitalisasi berbagai aspek kehidupan manusia. Malware kini bukan hanya merusak, tetapi juga mencuri data, memata-matai aktivitas pengguna, dan bahkan mengendalikan sistem komputer korban (Feng et al., 2018).

Selain itu, malware juga mengalami perkembangan dalam kemampuan distribusi dan berkembang biak. Malware konvensional hanya mampu mereplikasi diri sendiri dengan menghasilkan identitas sidik jari yang sama. Malware jenis ini dikenal dengan nama monomorphic. Malware monomorphic mudah diatasi dengan software antivirus atau antimalware yang rutin diupdate databasenya. Berapapun banyakan keturunan atau hasil replikasi dari suatu malware, mereka hanya memiliki satu sidik jari saja. Berbeda halnya jika yang dihadapi adalah malware jenis modern, seperti polymorphic, metamorphic dan oligomorphic (Aslan and Samet, 2020). Malware- malware tersebut mampu mereplikasi dirinya sendiri dengan menghasilkan sidik jari yang berbeda. Apabila mereka mereplikasi sebanyak 1000 kali, maka jumlah sidik jari baru yang terbentuk pun juga sebanyak 1000 sidik jari yang semuanya bersifat unik. Hal ini tentu menyulitkan pengembang antivirus dalam merekam seluruh sidik jari malware jenis modern ini, karena deteksi berbasis sidik jari menjadi tidak efektif. Penyimpanan sidik jari malware jenis ini akan menghabiskan banyak memori, hingga menyebabkan proses deteksi menjadi lambat. Malware pun belum tentu berhasil dideteksi karena sample sidik jari yang diambil dari komputer satu dengan komputer lain tentu berbeda. Oleh karena itu dibutuhkan metode deteksi yang lebih advanced yang tidak lagi berbasis sidik jari, melainkan berbasis perilaku. Metode deteksi modern tersebut diantaranya dapat dikembangkan dengan memanfaatkan machine learning.

TINJAUAN PUSTAKA

(Abujazoh et al., 2023) menguji beberapa algoritma klasifikasi, seperti SVM, k-Nearest Neighbor (kNN) dan Decision Tree (DT) dalam kasus deteksi malware. Peneliti membagi dataset ke dalam 8 bagian untuk mengatasi ketidakseimbangan, dengan menerapkan metode under-sampling. Masing-masing bagian dataset tersebut ditambah dengan 595 file non-malware (goodware) untuk menghasilkan kelas yang seimbang antara kelas malware dan goodware. Selanjutnya peneliti menguji algoritma SVM, kNN, dan DT pada masing-masing dataset tersebut. Hasilnya, DT memiliki performa terbaik dan paling konsisten dibandingkan dua algoritma klasifikasi lain. Skor terbaik DT diperoleh pada bagian dataset nomor 8 dengan metode seleksi fitur Chi Square (30 fitur terbaik), dengan skor akurasi 98.53%.

Pada penelitian lain, (Supriyanto et al., 2024) menguji performa algoritma kNN dengan beragam nilai k dan metode seleksi fitur pada kasus deteksi malware. Skor terbaik diperoleh saat menggunakan kNN dengan nilai k = 3 dan metode seleksi fitur Information Gain (32 fitur), dengan skor akurasi dan F1-Score yaitu 96.9%.

Algoritma Random Forest merupakan teknik ensemble learning yang menggabungkan beberapa random tree dalam melakukan klasifikasi (Lymin et al., 2023). Random Forest diharapkan dapat menghasilkan model yang lebih baik dalam hal akurasi. Deteksi malware yang akurat akan mencegah dari munculnya bahaya yang disebabkan oleh terjadinya kesalahan pendeteksian. Penggunaan seleksi fitur khususnya metode Information Gain, berperan untuk menyederhanakan pemrosesan dengan mengurangi jumlah dimensi pada data, sehingga berdampak pada kecepatan proses yang meningkat (Zebari et al., 2020). Model deteksi yang akurat dan cepat dapat diterapkan pada aplikasi smart antivirus yang tidak lagi mendeteksi malware berdasarkan sidik jari, melainkan perilaku, sehingga dapat digunakan untuk mengatasi malware-malware modern dengan jenis polymorphic, metamorphic, dan oligomorphic (Wu et al., 2011).

Penelitian ini mengusulkan teknik ensemble learning yaitu algoritma Random Forest yang dikombinasikan dengan seleksi fitur Information Gain untuk meningkatkan akurasi dan kecepatan deteksi Malicious Software.

METODE PENELITIAN

Deteksi malware dapat menghasilkan resiko yang tinggi, maka skor-skor yang

dihasilkan pada *state-of-the-art* di atas masih perlu ditingkatkan untuk mencapai zero tolerance terhadap kesalahan deteksi malware, baik itu false positive maupun false negative. Oleh karena itu, peneliti mengusulkan model hybrid Random Forest dan Information Gain untuk meningkatkan performa algoritma klasifikasi machine learning pada kasus deteksi malware.

Pelaksanaan penelitian ini dibagi menjadi 5 tahap seperti terlihat pada gambar 1, yaitu: Persiapan Dataset, Pra-Pemrosesan, Pemodelan, Validasi, dan Evaluasi. Tahap pertama yaitu persiapan dataset. Langkah yang pertama kali dilakukan pada tahap ini yaitu mengunduh dataset dari website UCI Machine Learning Repository (https://archive.ics.uci.ed). Dataset yang digunakan pada penelitian ini adalah dataset publik dengan detail seperti terlihat pada tabel 1.

Gambar 1. Metode Penelitian

Tabel 1. Informasi dataset yang digunakan

Nama Dataset	**Malware static and dynamic VxHeaven and VirtusTotal Dataset**
Jumlah File	3 (goodware, malware dari VirusTotal, dan malware dari VxHeaven)
Jumlah Record	Goodware: 595; Malware VirusTotal: 2955; Malware VxHeaven: 2698
Jumlah Fitur	Goodware: 1085; Malware VirusTotal: 1087; Malware VxHeaven: 1087
Missing Values	Tidak ada

Selanjutnya penyeragaman fitur, dari 3 file dataset yang diguankan memiliki jumlah fitur yang tidak sama. Beberapa fitur perlu dihapus seperti: filename, vbaVarIndexLoad, dan SafeArrayPtrOfIndex sehingga menyisakan 1084 fitur untuk ketiga file dataset. Selanjutnya, file-file dataset tersebut digabung menjadi sebuah file dataset yang lengkap. Pada tahap ini, ditambahkan atribut kelas, dimana pada data-data malware diberi nilai 1 dan pada goodware diberi nilai 0. Dengan demikian, kelas 0 merepresentasikan file goodware, dan kelas 1 merepresentasikan file malware.

Tahapan yang ke-dua, yaitu Pra-Pemrosesan. Pada tahap ini, dataset akan diolah terlebih dahulu sebelum siap digunakan untuk pemodelan. Pengolahan data yang pertama dilakukan di tahap ini, yaitu Penyeimbangan Kelas. Setelah ketiga file dataset digabungkan, maka terjadi ketidak-seimbangan kelas, dimana kelas 0 berisi 595 data dan kelas 1 berisi 5653 data. Rasio perbandingan antar kelas mencapai 1:9.5 dan masuk kategori medium imbalanced dan perlu diseimbangkan (Rafrastara et al., 2023). Metode penyeimbangan kelas yang digunakan adalah Random Under Sampling (RUS), yaitu dengan memotong jumlah data pada kelas mayoritas sehingga sama dengan jumlah data pada kelas minoritas. Setelah metode RUS diterapkan, maka dataset yang digunakan berisi data-data dengan jumlah yang sama pada masing-masing kelas, yaitu 595 data.

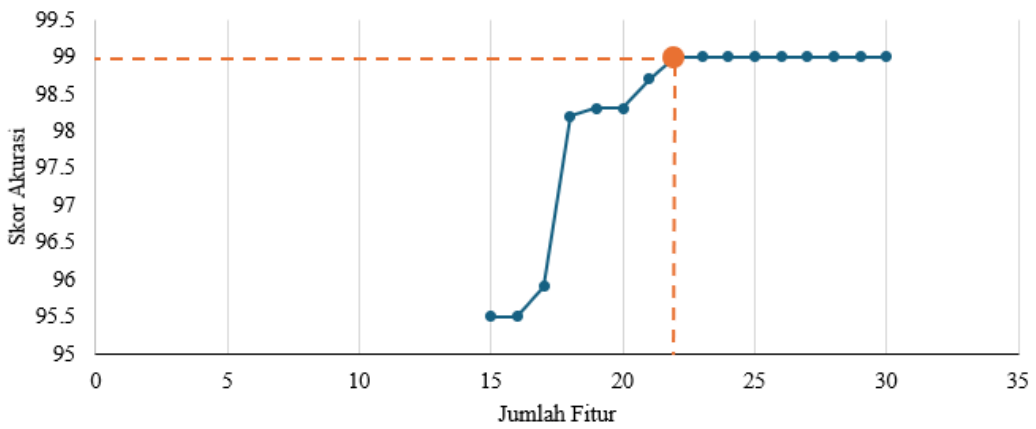
Langkah berikutnya yaitu melakukan penskalaan fitur, baik pada fitur numerik maupun kategorikal. Penskalaan fitur merupakan upaya menyeragamkan rentang fitur untuk mengurangi bias (Singh and Dwivedi, 2020). Metode yang digunakan dalam menyeimbangkan fitur di penelitian ini adalah MinMax Scaler atau MinMax Normalization

dengan rentang 0 – 1. Fitur numerik akan langsung diproses menggunakan MinMax Normalization (0-1). Sedangkan pada fitur kategorikal, fitur-fitur tersebut akan diperlakukan layaknya fitur ordinal yang dinormalisasi. Dengan demikian, baik fitur numerik maupun kategorikal akan memiliki rentang yang sama, yaitu antara 0 dan 1.

Metode seleksi fitur yang digunakan pada penelitian ini adalah perangkingan berbasis Information Gain. Tabel 2 menunjukkan 22 fitur dengan nilai information gain terbaik. Jumlah 22 pada fitur dipilih karena berhasil mempertahankan performa algoritma Random Forest pada skor akurasi 99% pada Gambar 2. Semakin sedikit fitur yang terlibat, maka proses komputasi akan berlangsung lebih cepat. Oleh karena itu, 22 fitur lebih dipilih dibandingkan dengan jumlah fitur lain seperti 30, 50, atau bahkan 1084 yang sama-sama menghasilkan skor akurasi 99%.

Tabel 2. 22 Fitur dengan skor IG tertinggi

No.	Fitur	Skor IG
1.	Minor_image_version	0.761
2.	Minor_operating_system_version	0.716
3.	Major_operating_system_version	0.639
4.	Size_of_stack_reverse	0.626
5.	Compile_date	0.593
6.	Minor_linker_version	0.566
7.	Major_image_version	0.555
8.	Major_subsystem_version	0.523
9.	Dll_characteristics	0.485
10.	Minor_subsystem_version	0.477
11.	CheckSum	0.408
12.	Major_linker_version	0.395
13.	Characteristics	0.389
14.	Number_of_IAT_entires	0.257
15.	Number_of_IAT_entires.1	0.257
16.	Pushf	0.243
17.	Size_of_stack_commit	0.239
18.	Files_operations	0.221
19.	.text:	0.205
20.	Count_dll_loaded	0.202
21.	SizeOfHeaders	0.195
22.	Size_of_headers	0.195



Gambar 2. Perbandingan jumlah fitur terhadap akurasi pada algoritma Random Forest

Tahap ke-3 yaitu Pemodelan. Terdapat 3 algoritma yang dibandingkan pada tahap ini, yaitu Decision Tree, kNN, dan Random Forest. Dengan menggunakan dataset yang telah disiapkan sebelumnya, maka ke-3 algoritma tersebut pun diterapkan satu per satu. Decision Tree dipilih karena merupakan algoritma terbaik dalam berdasarkan penelitian (Abujazoh et al., 2023). Sementa itu, kNN dipilih karena merupakan algoritma yang direkomendasikan oleh (Supriyanto et al., 2024).

Sedangkan Random Forest merupakan usulan dari peneliti, karena merupakan algoritma ensemble yang mampu menghasilkan performa lebih baik dibandingkan dengan algoritma Decision Tree dan kNN pada umumnya.

Untuk menguji keberhasilan suatu algoritma, terlebih dahulu perlu dilakukan validasi sebelum ke tahap evaluasi. Metode validasi yang digunakan pada penelitian ini adalah Cross Validation dengan jumlah k=10 (10-fold cross validation). Metode ini dipilih karena dapat mencegah terjadinya overfitting pada model (Battineni et al., 2019; Orrù et al., 2020).

Setelah divalidasi, berikutnya model dievaluasi menggunakan Confusion Matrix. Confusion Matrix berguna untuk mengetahui gambaran hasil prediksi yang dilakukan oleh model dengan menghasilkan nilai True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Ke-4 nilai tersebut kemudian dapat dimanfaatkan untuk mencari skor akurasi, recall, presisi, dan F1-Score. Pada penelitian ini, 2 metric evaluasi yang akan digunakan, yaitu akurasi dan F1-Score. Skor akurasi menunjukkan seberapa sering suatu model memprediksi dengan benar (Dev et al., 2022). Rumus perhitungan nilai akurasi dapat dilihat pada Formula 1.

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

Metrik evaluasi berikutnya yang akan digunakan adalah F1-Score. Metrik ini memiliki peran untuk mendapatkan keseimbangan antara presisi (Formula 2) dan recall (Formula 3) (Gupta et al., 2021) Formula yang digunakan untuk mendapaatkan nilai F1-Score dapat dilihat pada Formula 4.

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

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

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

HASIL DAN PEMBAHASAN

Penelitian ini menggunakan dataset yang sama dengan yang digunakan oleh (Abujazoh et al., 2023) dan (Supriyanto et al., 2024), yaitu "Malware static and dynamic features VxHeaven and VirusTotal Data Set" yang didownload dari UCI Machine Learning Repository. Setelah melalui tahap persiapan dan pra -pemrosesan, algoritma Random Forest pun diterapkan dan diuji performanya. Hasil pengujian performa algoritma Random Forest dibandingkan dengan 2 algoritma lain yang digunakan oleh Abujazoh et al. dan Supriyanto et al. dapat dilihat pada Tabel 3, serta diilustrasikan melalui grafik di gambar 3.

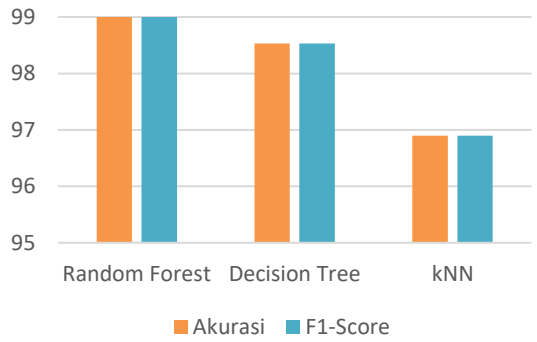
Tabel 3. Performa algoritma pada lingkungan metode masing-masing

Algoritma	Metode	Akurasi	Pustaka
Random Forest	Usulan	99.00	99.00
Decision Tree	(Abujazoh et al., 2023)	98.53	98.53
k-Nearest Neighbor	(Supriyanto et al., 2024)	96.90	96.90

Algoritma Random Forest unggu hampir 0.5 poin dibandingkan Decision Tree pada metrik akurasi dan F1-Score. Sementara itu, algoritma kNN memiliki skor terendah di antara ke-3 algoritma yang diuji. Perbedaan skor ini tidak dapat dikatakan fair mengingat adanya perbedaan pada perlakuan dataset. Algoritma Decision Tree pada penelitian (Abujazoh et al., 2023) memperoleh skor 98.53 setelah dilakukan pre- processing dengan melakukan balancing dataset dengan cara membagi dataset malware ke dalam 8 bagian, dan masing-masingnya diberikan tambahan berupa data-data file goodware sebanyak 595 data. Selanjutnya, pada fase data splitting, metode yang digunakan adalah Monte Carlo Cross-Validation.

Sementara itu, pada penelitian (Supriyanto et al., 2024), penggunaan algoritma

kNN menghasilkan skor akurasi dan F1-Score sebesar 96.9%. Pada tahap pra- pemrosesan, menggunakan metode Random Under Sampling (RUS) untuk menyeimbangkan data, dan metode seleksi fitur Information Gain untuk memilih 32 fitur terbaik. Sedangkan untuk metode validasinya, peneliti menggunakan 10-fold Cross Validation. Nilai k pada algoritma kNN yaitu 3 karena menghasilkan performa yang lebih baik dibandingkan k 5 dan 7.



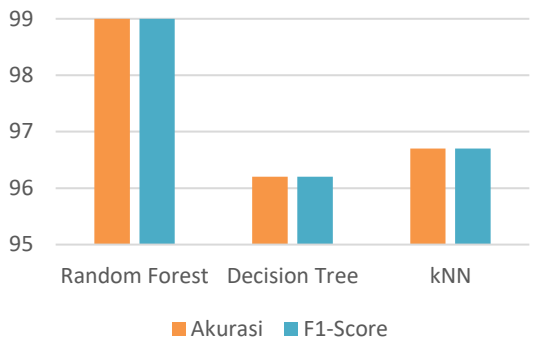
Gambar 3. Perbandingan performa algoritma pada lingkungan metode masing-masing

Untuk mendapatkan perbandingan yang lebih adil, maka algoritma Decision Tree dan algoritma kNN akan diuji pada lingkungan metode yang diusulkan pada penelitian ini. Hasilnya, algoritma Random Forest tetap mendapatkan nilai tertinggi dibandingkan dengan Decision Tree dan kNN. Hasil dari pengujian performa tersebut dapat dilihat di tabel 4 serta diilustrasikan pada grafik di gambar 4.

Pada pengujian di lingkungan metode yang diusulkan oleh penulis, terlihat Random Forest masih unggul dari 2 algoritma lain, dengan nilai akurasi dan F1-Score sebesar 99.0%. Sementara itu, kNN menggeser posisi Decision Tree sebagai runner-up dengan skor 96.7%, baik untuk akurasi dan F1-Score. Skor tereendah dimiliki oleh Decision dengan 96.2% untuk akurasi dan F1-Score. Dengan hasil pengujian ini, terbukti Random Forest lebih efektif dibandingkan kNN dan Decision Tree dalam mendeteksi malware.

Tabel 4. Performa algoritma pada lingkungan metode yang diusulkan

Algoritma	Akurasi	Pustaka
Random Forest	99.00	99.00
Decision Tree	98.53	98.53
k-Nearest Neighbor	96.90	96.90

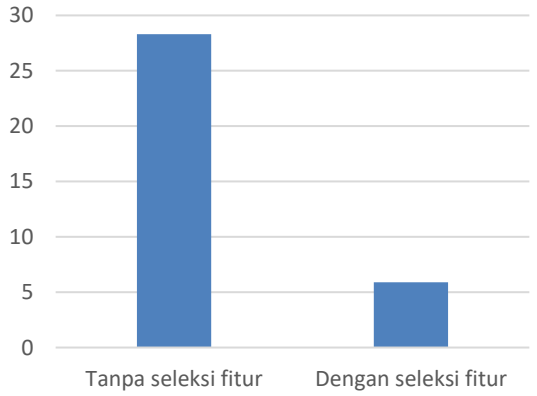


Gambar 4. Perbandingan performa algoritma pada lingkungan metode yang diusulkan

Penggunaan seleksi fitur juga berdampak pada waktu pemrosesan yang berlangsung lebih cepat, tanpa mengurangi performa pada metrik akurasi dan F1-Score. Tabel 5 dan Gambar 5 memuat informasi kecepatan pemrosesan tanpa dan dengan seleksi fitur. Selisih waktu yang dipangkas cukup signifikan, hingga 22.4 detik. Hal ini cukup wajar mengingat jumlah fitur yang digunakan pun berkurang drastis, yaitu 1084 menjadi 22 fitur berkat penggunaan information gain sebagai metode seleksi fiturnya.

Tabel 5. Kecepatan random forest dengan seleksi fitur dan tanpa seleksi fitur

	Kecepatan Pemrosesan (detik)
Tanpa seleksi fitur	28.3
Dengan seleksi fitur	5.9



Gambar 5. Perbandingan kecepatan random forest dengan seleksi fitur dan tanpaseleksi fitur

Hasil pengujian kecepatan eksekusi algoritma Random Forest dengan dan tanpa fitur seleksi menunjukkan betapa pentingnya fitur seleksi di dalam fase pre- processing. Penurunan waktu pemrosesan terlihat cukup

signifikan, dari 28.3 detik menjadi 5.9 detik. Dengan kata lain, kecepatan komputasinya meningkat hampir 5x lipat. Berdasarkan data yang diperoleh tersebut, dapat disimpulkan bahwa penerapan information gain untuk fitur seleksi mampu menghasilkan algoritma Random Forest yang lebih efisien.

SIMPULAN

Model hybrid antara Random Forest dan metode seleksi fitur Information Gain menghasilkan suatu algoritma prediksi dengan performa yang efektif dan efisien. Kombinasi Random Forest dan Information Gain mampu mengungguli 2 model hybrid lain, yaitu Decision Tree + Chi-Square dan kNN + Information Gain. Random Forest + Information Gain mampu menghasilkan nilai akurasi dan F1- Score sebesar 99%, atau 2.3% lebih unggul dibandingkan algoritma kNN yang menjadi runner-up. Selain itu, penerapan information gain sebagai metode seleksi fitur juga mampu memangkas waktu komputasi secara signifikan. Kecepatan komputasi meningkat hampir 5x lipat jika dibandingkan dengan tanpa adanya seleksi fitur.

Untuk penelitian selanjutnya, kami memberikan saran untuk menerapkan metode yang kami usulkan pada dataset malicious software lainnya. Metode ini juga dapat dikembangkan dengan mengganti algoritma ensemble dengan algoritma deep learning untuk meningkatkan akurasi.

UCAPAN TERIMA KASIH

Ucapan terima kasih penulis sampaikan kepada Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi Kementrian Pendidikan, Kebudayaan, Riset, dan Teknologi yang telah mendanai penelitian ini melalui skema Penelitian Fundamental - Reguler.

KONTRIBUSI PENULIS

Penulis pertama berkontribusi sebagai pengusul gagasan, merancang metode penelitian dan menyusun draft asli publikasi. Penulis kedua bertanggung jawab pada persiapan dataset, pra-pemrosesan, visualisasi, dan mereview draft publikasi. Penulis ketiga bertanggung jawab pada eksperimen terkait pemodelan, validasi dan evaluasi. Penulis keempat bertanggung jawab dalam supervisi pada setiap tahap penelitian hingga publikasi penelitian.

DAFTAR PUSTAKA

Abujazoh, M., Al-Darras, D., A. Hamad, N., Al-Sharaeh, S., 2023. Feature Selection for High-Dimensional Imbalanced Malware Data Using Filter and Wrapper Selection Methods, in: 2023 International Conference on Information Technology (ICIT). pp. 196–201. https://doi.org/10.1109/ICIT58056.2023.10226049

Alenezi, M.N., Alabdulrazzaq, H.K., Alshaher, A.A., Alkharang, M.M., 2022. Evolution of Malware Threats and Techniques: a Review. Int. j. commun. netw. inf. secur. 12. https://doi.org/10.17762/ijcnis.v12i3.4723

Aslan, Ö.A., Samet, R., 2020. A Comprehensive Review on Malware Detection Approaches. IEEE Access 8, 6249–6271. https://doi.org/10.1109/ACCESS.2019.2963724

Battineni, G., Sagaro, G.G., Nalini, C., Amenta, F., Tayebati, S.K., 2019. Comparative Machine-Learning Approach: A Follow-Up Study on Type 2 Diabetes Predictions by Cross-Validation Methods. Machines 7, 74. https://doi.org/10.3390/machines7040074

Dev, S., Kumar, B., Dobhal, D.C., Singh Negi, H., 2022. Performance Analysis and Prediction of Diabetes using Various Machine Learning Algorithms, in: 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N). pp. 517–521. https://doi.org/10.1109/ICAC3N56670.2022.10074117

Feng, P., Ma, J., Sun, C., Xu, X., Ma, Y., 2018. A Novel Dynamic Android Malware Detection System With Ensemble Learning. IEEE Access 6, 30996–31011. https://doi.org/10.1109/ACCESS.2018.2844349

Gupta, G., Rai, A., Jha, V., 2021. Predicting the Bandwidth Requests in XG-PON System using Ensemble Learning, in: 2021 International Conference on Information and Communication Technology Convergence (ICTC). pp. 936–941.

https://doi.org/10.1109/ICTC52510.2021.9620935

Lymin, Alvin, Lhoardi, B., Siahaan, J., Dharma, A., 2023. Analysis of Classification Models for ICU Mortality Prediction using Random Forest and Neural Network. Jurnal Informatika dan Rekayasa Perangkat Lunak 5, 130–134.

Orrù, G., Monaro, M., Conversano, C., Gemignani, A., Sartori, G., 2020. Machine Learning in Psychometrics and Psychological Research. Front. Psychol. 10, 2970. https://doi.org/10.3389/fpsyg.2019.02970

Rafrastara, F.A., Supriyanto, C., Paramita, C., Astuti, Y.P., Ahmed, F., 2023. Performance Improvement of Random Forest Algorithm for Malware Detection on Imbalanced Dataset using Random Under-Sampling Method. Jurnal Informatika 8, 113–118.

Singh, S.K., Dwivedi, Dr.R.K., 2020. Data Mining: Dirty Data and Data Cleaning. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3610772

Supriyanto, C., Rafrastara, F.A., Amiral, A., Amalia, S.R., Daffa, M., Fahreza, A., 2024. Malware Detection Using K-Nearest Neighbor Algorithm and Feature Selection 8.

Wu, L., Ping, R., Ke, L., Hai-xin, D., 2011. Behavior-based Malware Analysis and Detection, in: 2011 First International Workshop on Complexity and Data Mining. Presented at the 2011 First International Workshop on Complexity and Data Mining (IWCDM 2011), IEEE, Nanjing, Jiangsu, pp. 39–42.

Yadav, C.S., Gupta, S., 2022. A Review on Malware Analysis for IoT and Android System. SN Computer Science 4, 118. https://doi.org/10.1007/s42979-022-01543-w

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., Saeed, J., 2020. A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. JASTT 1, 56–70. https://doi.org/10.38094/jastt1224