KLIK: Kajian Ilmiah Informatika dan Komputer
ISSN 2723-3898 (Media Online)
Vol 4, No 6, Juni 2024, Hal 3073-3079
DOI 10.30865/klik.v4i6.1903
https://djournals.com/klik

Deteksi Aktifitas Malware pada Internet of Things menggunakan Algoritma Decision Tree dan Random Forest

M. Agus Syamsul Arifin[1,*], Andri Anto Tri Susilo[2], Susanto[2], A. Taqwa Martadinata[2], Budi Santoso[2]

[1] Fakultas Teknik, Program Studi Rekayasa Sistem Komputer, Universitas Bina Insan, Lubuklinggau, Indonesia
[2] Fakultas Teknik, Program Studi Informatika, Universitas Bina Insan, Lubuklinggau, Indonesia
Email: [1,*]mas.arifin@univbinainsan.ac.id, [2]andrianto@univbinainsan.ac.id, [3]susanto@univbinainsan.ac.id, [4]taqwa@univbinainsan.ac.id, [5]budisantoso@univbinainsan.ac.id
Email Penulis Korespondensi: mas.arifin@univbinainsan.ac.id

Abstrak−Internet of Things (IoT) telah menjadi bagian integral dari kehidupan modern, menghubungkan perangkat-perangkat pintar untuk meningkatkan efisiensi dan kenyamanan. Namun, dengan peningkatan adopsi IoT, ancaman keamanan siber, khususnya malware, juga semakin meningkat. Penelitian ini fokus pada deteksi serangan malware pada jaringan IoT menggunakan algoritma machine learning Decision Tree dan Random Forest. Dataset yang digunakan adalah CICIoT2023, yang mencakup berbagai jenis lalu lintas jaringan IoT termasuk BenignTraffic, Mirai-greeth_flood, Mirai-greip_flood, dan Backdoor_Malware. Dalam penelitian ini, kedua algoritma menunjukkan akurasi yang sangat tinggi pada data latih, yaitu mencapai 100%, dan pada data uji, akurasi yang dicapai pada kedua algoritma ini adalah 99,94% untuk algoritma random forest dan 99,90 untuk algoritma decision tree. Meskipun performa kedua algoritma pada data latih hampir sama, Random Forest menunjukkan performa yang lebih baik pada kelas Backdoor_Malware dibandingkan Decision Tree ketika menggunakan data uji. Random Forest mencapai precision sebesar 99%, recall 64%, dan F1-Score 78%, sementara Decision Tree mencapai precision sebesar 71%, recall 72%, dan F1-Score 72%. Hasil dari cross-validation 10-fold menunjukkan bahwa model yang dihasilkan tidak mengalami overfitting, mengindikasikan keandalan dan generalisasi model yang baik. Penelitian ini memberikan wawasan bahwa algoritma Random Forest lebih efektif dalam mendeteksi serangan malware pada jaringan IoT dibandingkan Decision Tree, terutama dalam mengidentifikasi kelas Backdoor_Malware. Temuan ini diharapkan dapat berkontribusi pada pengembangan sistem deteksi malware yang lebih efisien dan andal untuk jaringan IoT.

Kata Kunci: Internet of Things (IoT); Malware; Machine Learning; Decision Tree; Random Forest

Abstract−The Internet of Things (IoT) has become an integral part of modern life, connecting smart devices to enhance efficiency and convenience. However, with the increased adoption of IoT, cybersecurity threats, particularly malware, have also risen. This research focuses on detecting malware attacks in IoT networks using machine learning algorithms, specifically Decision Tree and Random Forest. The dataset used is CICIoT2023, which includes various types of IoT network traffic such as BenignTraffic, Mirai-greeth_flood, Mirai-greip_flood, and Backdoor_Malware. In this study, both algorithms demonstrated exceptionally high accuracy on the training data, reaching 100%, and on the test data, achieving 99.94% accuracy for the Random Forest algorithm and 99.90% for the Decision Tree algorithm. Although the performance of both algorithms on the training data was almost identical, Random Forest showed better performance in detecting the Backdoor_Malware class compared to Decision Tree when using test data. Random Forest achieved a precision of 99%, recall of 64%, and F1-Score of 78%, while Decision Tree achieved a precision of 71%, recall of 72%, and F1-Score of 72%. Results from 10-fold cross-validation indicate that the models did not experience overfitting, suggesting reliable and well-generalized models. This research provides insights that the Random Forest algorithm is more effective in detecting malware attacks in IoT networks compared to Decision Tree, particularly in identifying the Backdoor_Malware class. These findings are expected to contribute to the development of more efficient and reliable malware detection systems for IoT networks.

Keywords: Internet of Things (IoT); Malware; Machine Learning; Decision Tree; Random Forest

1. PENDAHULUAN

Internet of Things (IoT) telah menjadi bagian integral dari kehidupan modern, menghubungkan berbagai perangkat pintar untuk meningkatkan efisiensi dan kenyamanan. Dengan IoT, berbagai perangkat seperti lampu, termostat, kunci pintu, dan bahkan peralatan rumah tangga dapat dikendalikan dan dipantau dari jarak jauh. Inovasi ini tidak hanya memberikan kemudahan bagi pengguna individu, tetapi juga memiliki potensi besar dalam meningkatkan produktivitas di sektor industri, kesehatan, dan kota pintar. Namun, di balik berbagai manfaatnya, adopsi IoT juga [1], [2]membawa tantangan baru dalam bidang keamanan siber.

Ancaman keamanan siber pada jaringan IoT semakin meningkat seiring dengan semakin banyaknya perangkat yang terhubung ke internet. Perangkat IoT sering kali memiliki keterbatasan dalam hal keamanan, seperti sumber daya komputasi yang terbatas, pembaruan perangkat lunak yang jarang, dan mekanisme keamanan yang lemah. Hal ini menjadikan jaringan IoT sebagai target empuk bagi serangan siber, terutama malware [3]. *Malware* yang menyusup ke jaringan IoT dapat menyebabkan berbagai kerusakan, mulai dari pencurian data pribadi, gangguan layanan yang kritis, hingga pelanggaran privasi yang serius . Oleh karena itu, deteksi dini dan akurat terhadap aktivitas malware sangat penting untuk menjaga integritas dan keamanan jaringan IoT [4].

Dalam konteks ini, penelitian kami berfokus pada deteksi serangan malware pada jaringan IoT menggunakan algoritma machine learning *Decision Tree* dan *Random Forest*. Algoritma ini dipilih karena mampu menangani data yang kompleks dan besar dengan efektif, serta memberikan interpretasi yang mudah dipahami terhadap hasil prediksi. *Decision Tree* merupakan algoritma yang membangun model klasifikasi dalam bentuk pohon keputusan, di mana setiap cabang mewakili keputusan berdasarkan fitur tertentu, sedangkan *Random Forest* adalah pengembangan dari *Decision Tree* yang membangun beberapa pohon keputusan dan menggabungkan hasilnya untuk meningkatkan akurasi dan mengurangi *overfitting* yang pada penelitian ini akan diukur berdasarkan hasil cross-validation dengan 10 fold.

Dataset yang digunakan dalam penelitian ini adalah CICIoT2023 [5], yang berisi berbagai jenis lalu lintas jaringan IoT. Dataset ini mencakup beberapa kelas, yaitu BenignTraffic (lalu lintas normal), Mirai-greeth_flood, Mirai-greip_flood, dan Backdoor_Malware. Kelas-kelas ini mewakili berbagai jenis serangan malware yang umum terjadi pada jaringan IoT [6] yang dapat menyebabkan kerusakan signifikan jika tidak terdeteksi dan ditangani dengan cepat. Dengan menggunakan dataset ini.

Penelitian bertujuan untuk mengembangkan model yang dapat mendeteksi dan mengklasifikasikan serangan malware dengan akurat. Penelitian juga membandingkan performa algoritma Decision Tree (DT) dan Random Forest (RF) dalam mendeteksi serangan malware pada jaringan IoT. Performa model akan diukur menggunakan beberapa metrik evaluasi yang umum digunakan dalam machine learning, yaitu Akurasi, Precision, Recall, dan F1-Score. Akurasi mengukur persentase prediksi yang benar dari total prediksi yang dilakukan oleh model. Precision adalah rasio antara prediksi positif yang benar dengan total prediksi positif yang dilakukan oleh model, sedangkan Recall mengukur kemampuan model dalam mendeteksi semua instance positif dari data yang sebenarnya positif. F1-Score adalah harmonisasi antara Precision dan Recall, memberikan gambaran keseimbangan antara kedua metrik tersebut [7].

Selain metrik-metrik tersebut, kurva Receiver Operating Characteristic (ROC) dan nilai Area Under Curve (AUC) juga akan digunakan untuk menilai kemampuan model dalam melakukan klasifikasi yang akurat terhadap jenis-jenis serangan yang ada dalam dataset. Kurva ROC adalah plot yang menggambarkan kinerja model klasifikasi pada berbagai threshold klasifikasi, sedangkan nilai AUC adalah area di bawah kurva ROC yang memberikan indikasi keseluruhan performa model . Nilai AUC mendekati 1 menunjukkan bahwa model memiliki kemampuan klasifikasi yang sangat baik, sedangkan nilai mendekati 0,5 menunjukkan performa yang sebanding dengan prediksi acak [8].

Untuk memastikan kehandalan dan generalisasi model, validasi dilakukan menggunakan teknik cross-validation 10-fold. Dalam teknik ini, dataset dibagi menjadi 10 subset (fold), dan model dilatih sebanyak 10 kali, setiap kali menggunakan 9 subset sebagai data latih dan 1 subset sebagai data uji. Proses ini berulang sebanyak 10 kali dengan setiap subset berperan sekali sebagai data uji. Teknik ini membantu dalam mengurangi bias yang mungkin muncul dari pembagian data secara acak dan memberikan gambaran yang lebih akurat tentang performa model pada data yang tidak terlihat sebelumnya.

Penelitian yang dilakukan oleh Xiao F, et.al [9] menggunakan deep learning untuk mendeteksi malware berdasarkan grafik yang dihasilkan oleh malware tersebut pada jaringan IoT. Penelitian ini mendapatkan perfoma 98,6% untuk algoritma SAE-DT dalam mendeteksi serangan malware. Kemudian penelitian yang dilakukan oleh Takase H, et.al [10] melakukan ekstraksi malware pada prosessor untuk mengenali pola aktifitas malware pada perangkat komputer. Penelitian untuk melakukan deteksi dini aktifitas malware pada perangkat jaringan IoT menggunakan machine learning dilakukan oleh Kumar A, et.al [3] mendapatkan hasil yang baik pada algoritma k nearest neighbors dengan akurasi mencapai 94,4%.

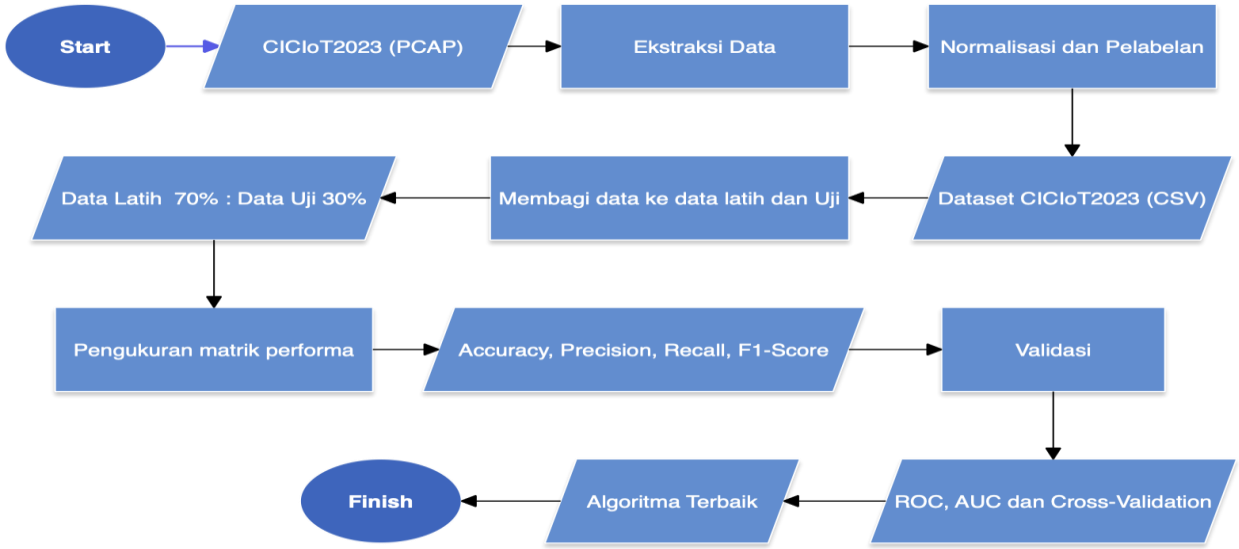
Hasil dari penelitian ini diharapkan dapat memberikan wawasan yang mendalam mengenai efektivitas penggunaan algoritma *Decision Tree* dan *Random Forest* dalam mendeteksi aktivitas malware pada jaringan IoT. Dengan membandingkan performa kedua algoritma ini, penelitian ini dapat membantu dalam pengembangan sistem deteksi malware yang lebih efektif dan efisien, yang dapat diterapkan dalam berbagai skenario penggunaan IoT. Selain itu, temuan penelitian ini juga dapat memberikan panduan bagi para pengembang dan peneliti dalam memilih dan mengoptimalkan algoritma machine learning untuk deteksi ancaman keamanan siber pada jaringan IoT.

Dengan semakin berkembangnya teknologi IoT dan meningkatnya ancaman keamanan yang menyertainya, penelitian ini menjadi sangat relevan. Penerapan algoritma machine learning seperti Decision Tree dan Random Forest dalam mendeteksi malware dapat menjadi solusi yang potensial untuk meningkatkan keamanan jaringan IoT. Diharapkan bahwa hasil dari penelitian ini tidak hanya berkontribusi pada literatur akademik tetapi juga dapat diimplementasikan dalam sistem keamanan dunia nyata untuk melindungi infrastruktur IoT dari ancaman malware.

2. METODOLOGI PENELITIAN

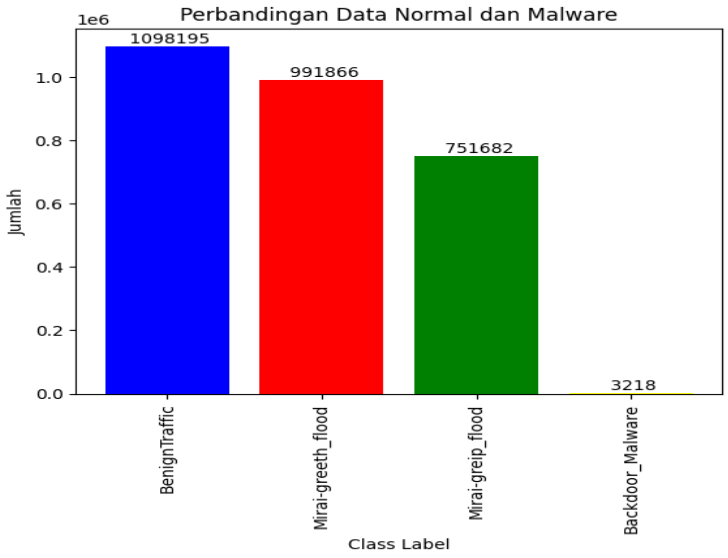
2.1 Tahapan Penelitian

Gambar 1 berikut menunjukkan alur yang digunakan dalam penelitian ini untuk menentukan model terbaik dalam mendeteksi serangan malware. Dataset CICIoT2023 yang digunakan merupakan data mentah dengan format pcap, kemudian dilakukan ekstraksi data ke dalam format comma sparated value (CSV) selanjutnya dilakukan proses normalisasi dengan menghapus data-data yang tidak digunakan dan melakukan proses pelabelan berdasarkan serangan. Dataset dibagi menjadi data latih dan data uji dengan komposisi 70% data latih dan 30% data uji kemudian dilakukan pengukuran performa dengan mengukur accuracy, precision, recall dan f1-score. Setelah melakukan pengukuran perfoma dilakukan validasi untuk melihat kemampuan model dalam mengklasifikasikan serangan dalam dataset dengan menggunakan ROC dan nilai AUC, selanjutnya dilakukan prosess cross validation dengan 10-fold untuk melihat apakah terdapat indikasi overfitting atau underfitting dari model IDS yang dibuat, dari hasil pengukuran dan validasi baru dapat ditentukan algoritma terbaik yang akan digunakan dalam membuat model IDS dalam mendeteksi serangan pada dataset CICIoT2023.



Gambar 1. Metode yang digunakan untuk menentukan algoritma terbaik dalam mendeteksi serangan malware pada dataset CICIoT2023

Penelitian ini menggunakan dataset CICIoT2023 memiliki format data pcap kemudian melakukan ekstraksi data dan pelabelan untuk data normal (*Benign_Traffic*) dan data serangan (*Backdoor_Malware, Mirai-greeth_flood dan Mirai-greip_flood*) kemudian disimpan dalam format csv (*Comma Separated Values*). Gambar 2 berikut menunjukkan jumlah data dan kelas dalam dataset CICIoT2023.



Gambar 2. Perbandingan data setiap kelas dalam dataset CICIoT2023 untuk serangan *malware*

Setelah proses ekstraksi kami membagi dataset menjadi data latih (70%) dan data uji (30%), kemudian mengukur performa model yang diuji berdasarkan akurasi, presisi, *recall* dan nilai F1-score. Metrik untuk mengukur performa kinerja model yang dibuat memiliki persamaan (1) – (4).

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

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

$$\text{Recall} = \frac{\text{TP}}{(TN+FP)} \quad (3)$$

$$\text{F1 Measure} = 2\frac{(\text{Precision x Recall})}{(Precision+Recall)} \quad (4)$$

Pada proses validasi penelitian ini menggunakan kurva ROC (*Receiver Operating Characteristic*) dan nilai AUC (*Area Under the Curve*) untuk melihat kemampuan model dalam mendeteksi serangan dalam dataset [11], [12] selanjutnya *cross-validation* (CV) juga digunakan dalam penelitian ini untuk melihat apakah model yang dibuat terdapat indikasi *overfitting* [13] dengan menggunakan 10-fold, CV akan mengacak setiap *sample* untuk setiap repetisi dengan

data yang relatif sama [14], kemudian dari hasil pengujian, pengukuran dan validasi tersebut baru dapat ditentukan algoritma terbaik dalam mendeteksi serangan pada dataset CICIoT2023.

2.2 Decision Tree

Penelitian ini menggunkan algoritma *Decision tree* yang merupakan algoritma dalam machine learning bersifat supervised. *Decision tree* adalah teknik dengan basis pohon dimana setiap jalur dimulai dari akar yang dijelaskan oleh urutan pemisahan data hingga hasil pada simpul daun tercapai dimana relasi digunakan untuk klasifikasi [15]. Beberapa penelitian tentang keamanan yanng menggunakan algoritma *Decision Tree* adalah penelitian yang dilakukan oleh Hilda M, et.al [16] menggunakan *Decision tree* untuk mendeteksi anomaly pada jaringan. Penelitian yang dilakukan oleh Ghani A, et.al [17]menggunakan decision tree untuk meningkatkan kemampuan sebuah sistem pendeteksi serangan dalam mendeteksi dan menanggulangi serangan yang memanfaatkan paket data dalam membanjiri jaringan. Kemudian penelitian yang dilakukan oleh Arifin M, et.al [7] membandingkan algoritma decision tree dengan algoritma gaussian naive bayes dan support vector machine dalam mendeteksi serangan DoS pada sistem jaringan SCADA dan algoritma decision tree mendapatkan hasil terbaik dari pengujian yang dilakukan.

2.3 Random Forest

Random Forest adalah salah satu algoritma klasifikasi berbasis pohon yang banyak digunakan dalam *ensemble*. Banyak aspek dari pembangunan *tree ensemble* diperkenalkan untuk mengurangi korelasi di antara *Decision tree* dalam *forest*. *Bootstrap* digunakan dalam Random Forest untuk mengurangi bias pada *Decision tree* dan untuk menentukan pembagian (*splits*) di setiap *decision tree* [18]. Dalam keamanan data algoritma *Random forest* digunakan oleh Faarnaz N, et.al [19] yang menggunakan *random forest* untuk membangun sebuah model IDS yang digunakan dalam mendeteksi serangan pada jaringan. Kemudian penelitian yang dilakukan oleh Wu T, et.al [20] menggunakan random forest dan smote untuk membuat model machine learning dalam mendeteksi aktifitas jahat pada dataset NSL-KDD. Dalam penelitian yang dilakukan oleh Rafanjani M, et.al [21] menggunakan random untuk mendeteksi aktifitas malware botnet dalam jaringan IoT dalam penelitian ini menggunakan dataset dari UNSW Canberra yaitu Bot-IoT UNSW-2018.

3. HASIL DAN PEMBAHASAN

Jumlah data pada dataset setelah proses normalisasi adalah 2.844.961 dengan data normal (Benign_Traffic) berjumlah 1.098.195 data, serangan *Mirai-greeth_flood* 991.866 data, *Mirai-greip_flood* 751.682 data, dan *Malware_Backdoor* 3.218. Sub bab 3.1 akan menjabarkan hasil dari performa model IDS (Intrusion Detection System) menggunakan machine learning dengan Decision tree dan Random Forest yang menggunakan dataset CICIoT2023 sekaligus hasil validasi dari model machine learning yang dibuat dan sub bab 3.2 akan menjabarkan pembahasan hasil dari performa model IDS pada dataset CICIoT2023.

3.1 Hasil

Dari pengujian yang dilakukan performa model IDS yang dibangun menggunakan algoritma random forest mendapatkan hasil yang lebih baik pada data uji, namun pada data latih performa kedua model machine learning yang digunakan mendapatkan hasil yang sama. Tabel 1 berikut menunjukkan perbandingan performa model IDS yang menggunakan algoritma DT dan GB pada data latih dan tabel 2 menggunakan data uji.

Tabel 1. Perbandingan performa model IDS dengan algoritma Decision tree dan Random Forest pada data latih

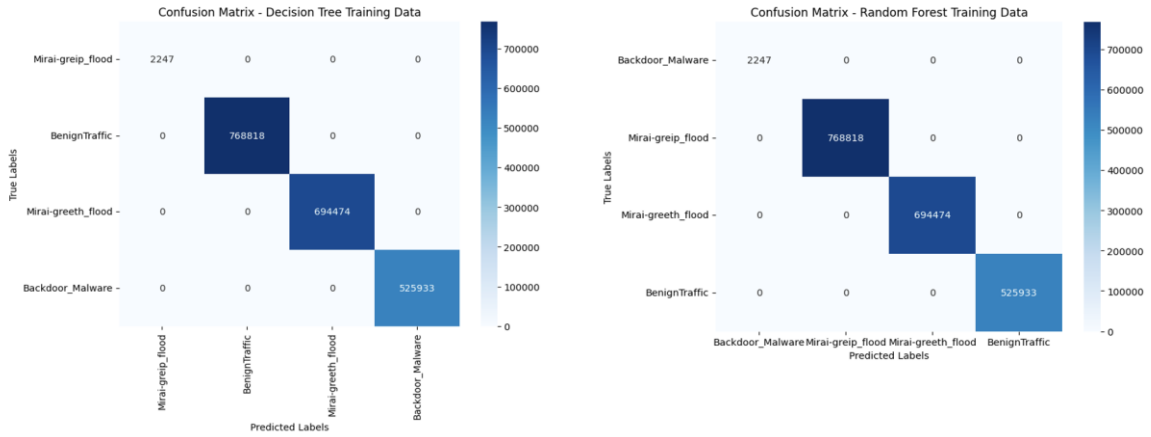
Model	Evaluation	Perfoma			
		Akurasi	Precision	Recall	F1-Score
Decision Tree	BenignTraffic	100%	1,00	1,00	1,00
	Backdoor_Malware		1,00	1,00	1,00
	Mirai-greeth_flood		1,00	1,00	1,00
	Mirai-greip_flood		1,00	1,00	1,00
Gradient Boosting	BenignTraffic	100%	1,00	1,00	1,00
	Backdoor_Malware		1,00	1,00	1,00
	Mirai-greeth_flood		1,00	1,00	1,00
	Mirai-greip_flood		1,00	1,00	1,00

Tabel 2. Perbandingan performa model IDS dengan algoritma Decision tree dan Random Forest pada data uji

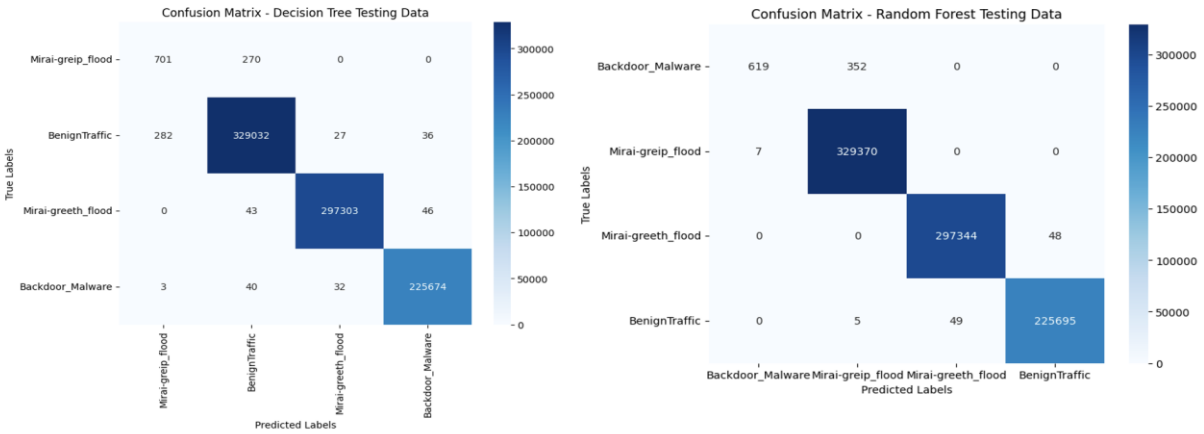
Model	Kelas	Performa			
		Akurasi	Precision	Recall	F1-Score
Decision Tree	BenignTraffic	99,90%	0,71	0,72	0,72
	Backdoor_Malware		1,00	1,00	1,00
	Mirai-greeth_flood		1,00	1,00	1,00
	Mirai-greip_flood		1,00	1,00	1,00

Gradient Boosting	BenignTraffic	99,94%	0,99	0,64	0,78
	Backdoor_Malware		1,00	1,00	1,00
	Mirai-greeth_flood		1,00	1,00	1,00
	Mirai-greip_flood		1,00	1,00	1,00

Kami menggunakan grafik confusion matrix untuk melihat alarm palsu dan kesalahan deteksi dari model yang dibuat. Gambar 3 berikut menunjukkan confusion matrix model IDS ketika menggunakan data latih sedangkan Gambar 4 menunjukkan confusion matrix model IDS ketika menggunakan data uji.

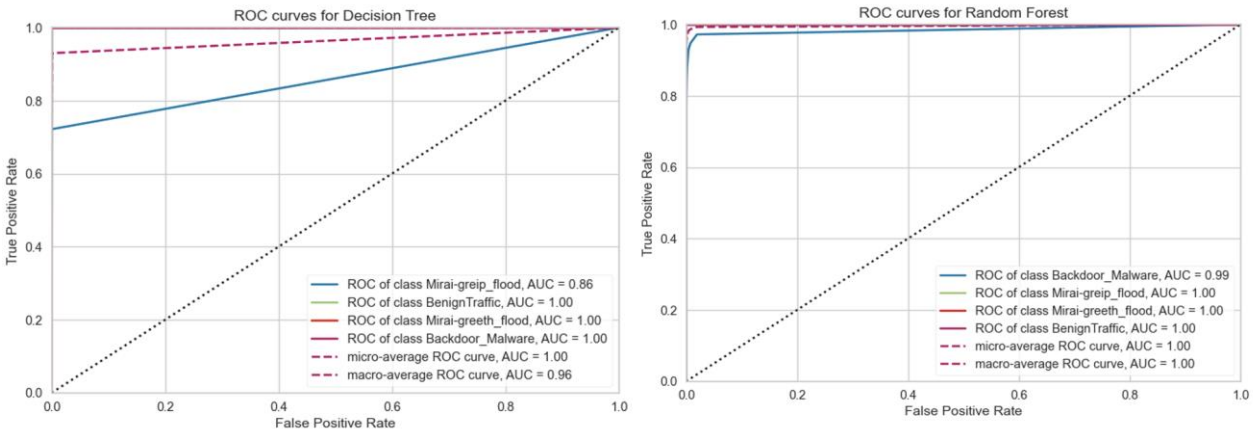


Gambar 3. Confusion matrix untuk model IDS dengan menggunakan data latih

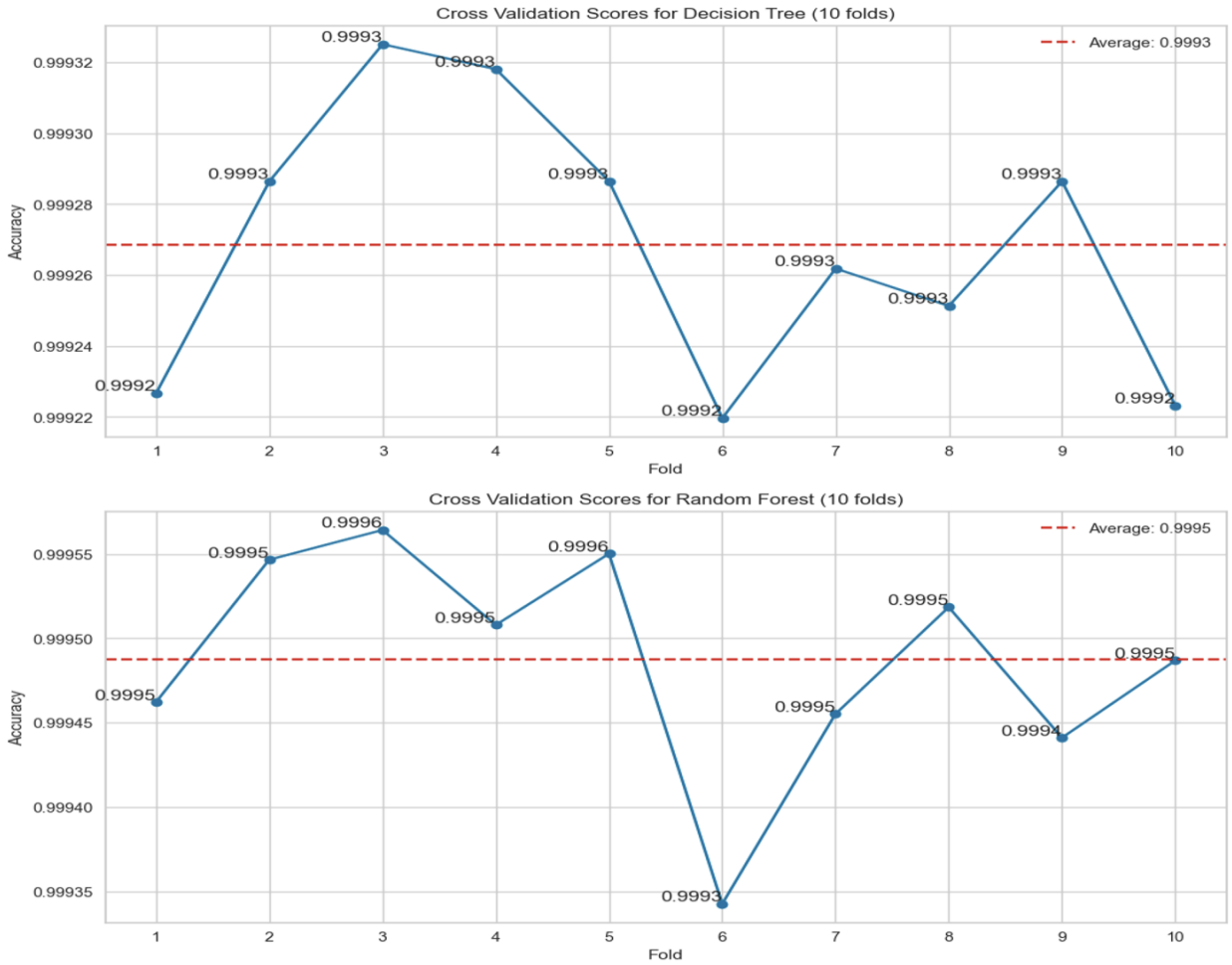


Gambar 4. Confusion matrix untuk model IDS dengan menggunakan data latih

Gambar 5 berikut menunjukkan kurva ROC dan nilai AUC model IDS untuk mengukur kemampuan model yang dibuat dalam melakukan klasifikasi serangan dan Gambar 6 menunjukkan hasil cross-validation dari model IDS untuk melihat model yang dibuat terdapat overfitting [22] atau tidak.



Gambar 5. Perbandingan Kurva ROC dan nilai AUC untuk model IDS machine learning



Gambar 6. Perbandingan hasil cross-validation dari model IDS machine learning yang dibangun.

3.2 Pembahasan

Random forest dan Decision tree dalam pengujian mendapatkan hasil yang sangat tinggi ketika menggunakan data latih dengan akurasi 100%, namun untuk data uji dimana data yang diujikan tidak dimasukkan dalam data latih algoritma Random forest mendapatkan hasil yang lebih baik dengan akurasi 99,94% dari pada algoritma Decision tree yang mendapatkan akurasi sebesar 99,90% hal ini menunjukan algoritma Random forest merupakan model IDS yang lebih baik untuk mendeteksi serangan yang belum dipelajari sebelumnya dibandingkan dengan model IDS dengan algoritma Decision tree. Pada hasil pengukuran alarm palsu dan kesalahan deteksi yang terlihat pada gambar confusion matrix model IDS dengan algoritma random forest juga mendapatkan hasil yang lebih baik jika dibandingkan dengan performa model IDS dengan algoritma decision tree.

Kurva ROC dan nilai AUC menunjukkan algoritma random forest mendapatkan hasil yang lebih baik dalam melakukan klasifikasi serangan pada dataset dari pada algortima decision tree ditunjukkan dengan nilai kurva ROC dan nilai AUC yang lebih baik. Dari hasil validasi menggunakan cross-validation menunjukkan tidak terdapat gejala overfitting dari model yang dibangun karena akurasi yang didapat pada setiap fold tidak berbeda jauh dari akurasi model yang dihasilkan sehingga model IDS yang dibangun merupakan model yang valid dalam mendeteksi serangan pada jaringan IoT.

4. KESIMPULAN

Dari hasil pengujian menunjukkan bahwa model Intrusion Detection System (IDS) yang dibangun menggunakan algoritma random forest mendapatkan hasil yang lebih baik dibandingkan dengan model IDS yang dibangun menggunakan algoritma decision tree. Hasil validasi menggunakan metode cross-validation dengan 10-folds menunjukkan bahwa tidak terdapat overfitting pada model yang dibangun, sehingga model yang dihasilkan merupakan model yang valid dan andal. Hal ini mengindikasikan bahwa model random forest memiliki kemampuan generalisasi yang lebih baik dalam mendeteksi intrusi. Selain itu, penelitian ini tidak menggunakan metode untuk menyeimbangkan kelas meskipun terdapat ketidakseimbangan pada data, khususnya pada kelas Backdoor_Malware. Ketidakseimbangan kelas ini sering kali menjadi tantangan dalam pengembangan model machine learning, karena dapat menyebabkan model

lebih cenderung mengabaikan kelas minoritas. Namun, meskipun tanpa penyeimbangan kelas, model yang dibangun menunjukkan performa yang baik dalam hal akurasi, precision, recall, dan F1-Score. Ini menandakan bahwa algoritma random forest mampu menangani ketidakseimbangan data dengan cukup baik dan masih memberikan hasil yang akurat dan dapat diandalkan untuk mendeteksi serangan malware. Keberhasilan model ini dalam mencapai performa yang tinggi tanpa perlu penyeimbangan kelas menunjukkan potensi besar algoritma random forest dalam aplikasi IDS. Dalam konteks keamanan siber, kemampuan untuk mendeteksi intrusi dengan akurasi tinggi sangatlah penting untuk mencegah dan menangani serangan sebelum mereka menyebabkan kerusakan yang signifikan. Dengan demikian, penelitian ini memberikan kontribusi yang berharga dalam pengembangan teknologi IDS yang lebih efektif dan efisien..

REFERENCES

[1] H. Alasmary et al., "Analyzing and Detecting Emerging Internet of Things Malware: A Graph-based Approach," IEEE Internet Things J, vol. 4662, no. c, pp. 1–1, 2019, doi: 10.1109/jiot.2019.2925929.

[2] M. Wazid, A. K. Das, J. J. P. C. Rodrigues, S. Shetty, and Y. Park, "IoMT Malware Detection Approaches: Analysis and Research Challenges," IEEE Access, vol. 7, pp. 182459–182476, 2019, doi: 10.1109/ACCESS.2019.2960412.

[3] A. Kumar and T. J. Lim, "EDIMA: Early Detection of IoT Malware Network Activitay Using Machine Learning Techniques," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), pp. 289–294, 2019, doi: 10.1109/wf-iot.2019.8767194.

[4] ec-council, Ec-Council Official Curricula Hacking Essentials Ethical PROFESSIONAL SERIES, Version 1. New Mexico: EC-Council, 2021.

[5] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment," Sensors, vol. 23, no. 13, p. 5941, 2023, doi: 10.3390/s23135941.

[6] A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, and S. Nõmm, "MedBIoT: Generation of an IoT botnet dataset in a medium-sized IoT network," ICISSP 2020 - Proceedings of the 6th International Conference on Information Systems Security and Privacy, no. March, pp. 207–218, 2020, doi: 10.5220/0009187802070218.

[7] M. A. S. Arifin, D. Stiawan, Susanto, J. Rejito, Mohd. Y. Idris, and R. Budiarto, "Denial of Service Attacks Detection on SCADA Network IEC 60870-5-104 using Machine Learning," in International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) 2021, 2021, pp. 228–232. doi: 10.23919/eecsi53397.2021.9624255.

[8] M. A. S. Arifin, D. Stiawan, and B. Y. Suprapto, "Oversampling and undersampling for intrusion detection system in the supervisory control and data acquisition IEC 60870 - 5 - 104," IET Cyber-Physical Systems: Theory & Applications, no. November 2023, 2024, doi: 10.1049/cps2.12085.

[9] F. Xiao, Z. Lin, Y. Sun, and Y. Ma, "Malware Detection Based on Deep Learning of Behavior Graphs," Math Probl Eng, vol. 2019, 2019, doi: 10.1155/2019/8195395.

[10] H. Takase, R. Kobayashi, M. Kato, and R. Ohmura, "A prototype implementation and evaluation of the malware detection mechanism for IoT devices using the processor information," Int J Inf Secur, 2019, doi: 10.1007/s10207-019-00437-y.

[11] N. Gupta, V. Jindal, and P. Bedi, "LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system," Computer Networks, vol. 192, no. March, pp. 1–19, 2021, doi: 10.1016/j.comnet.2021.108076.

[12] P. I. priyadarsini, ABC-BSRF: Artificial Bee Colony and Borderline-SMOTE RF Algorithm for Intrusion Detection System on Data Imbalanced Problem, vol. 56. Springer Singapore, 2021. doi: 10.1007/978-981-15-8767-2_2.

[13] H. Shafique, A. A. Shah, M. A. Qureshi, and M. K. Ehsan, "Machine Learning Empowered Efficient Intrusion Detection Framework," VFAST Transactions on Software Engineering, vol. 10, no. 2, pp. 27–35, 2022, doi: http://dx.doi.org/10.21015/vtse.v10i2.1017.

[14] M. Artur, "Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features," Procedia Comput Sci, vol. 190, no. 2019, pp. 564–570, 2021, doi: 10.1016/j.procs.2021.06.066.

[15] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," Journal of Applied Science and Technology Trends, vol. 2, no. 01, pp. 20–28, 2021, doi: 10.38094/jastt20165.

[16] M. Hilda, L. Louk, and B. Adhi, "Dual-IDS : A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system," Expert Syst Appl, vol. 213, no. PB, p. 119030, 2023, doi: 10.1016/j.eswa.2022.119030.

[17] A. Ghani, M. Rafie, and F. M. Abdalla Ali, "Enchancing Hybrid Intrusion Detection and Prevention System for Flooding Attacks Using Decision Tree," in International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), 2019, pp. 2–5. [Online]. Available: https://www.ptonline.com/articles/how-to-get-better-mfi-results

[18] R. M. Mohana, C. K. K. Reddy, P. R. Anisha, and B. V. R. Murthy, "Random forest algorithms for the classification of tree-based ensemble," Mater Today Proc, 2021, doi: 10.1016/j.matpr.2021.01.788.

[19] N. Farnaaz and M. A. Jabbar, "Random Forest Modeling for Network Intrusion Detection System," Procedia Comput Sci, vol. 89, pp. 213–217, 2016, doi: 10.1016/j.procs.2016.06.047.

[20] T. Wu, H. Fan, H. Zhu, C. You, H. Zhou, and X. Huang, "Intrusion detection system combined enhanced random forest with SMOTE algorithm," EURASIP J Adv Signal Process, vol. 2022, no. 1, 2022, doi: 10.1186/s13634-022-00871-6.

[21] M. S. Rafsanjani, V. Suryani, and R. R. Pahlevi, "Deteksi serangan botnet pada jaringan internet of things menggunakan algoritma random forest," e-Proceeding of Engineering, vol. 9, no. 3, pp. 1862–1871, 2022.

[22] A. C. Müller and S. Guido, "Introduction to Machine Learning with Python A GUIDE FOR DATA SCIENTISTS Introduction to Machine Learning with Python." 2016