Perbandingan Kinerja Algoritma Dalam Klasifikasi Serangan DDoS Berdasarkan Data CIC IoMT Dataset

Fikri Azhari[1*], Bayu Hananto[2], Iin Ernawati[3]
Program Studi S1 Informatika / Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, Daerah Khusus Ibukota *Jakarta 12450*
fikria@upnvj.ac.id [1], bayuhananto@upnvj.ac.id [2] , iinernawati@upnvj.ac.id [3]

Abstrak. Dengan semakin luasnya penerapan *Internet of Things* (IoT) di berbagai sektor, termasuk sektor medis dengan teknologi Internet of Medical Things (IoMT), serangan *Distributed Denial of Service* (DDoS) menjadi ancaman serius bagi keberlangsungan sistem. Penelitian ini membandingkan empat algoritma *machine learning Random Forest, LightGBM, Naïve Bayes*, dan *K-Nearest Neighbors* (KNN) untuk mendeteksi serangan DDoS pada IoMT. Evaluasi dilakukan berdasarkan akurasi dan waktu komputasi yang berjalan secara paralel menggunakan pendekatan *Weighted Sum Method*. Hasil menunjukkan bahwa Random Forest memiliki performa terbaik dengan skor 0.971578, diikuti oleh *Naïve Bayes* dengan skor 0.961235. Meskipun KNN memiliki akurasi tinggi, algoritma ini kurang efisien secara waktu, sedangkan LightGBM menunjukkan performa terendah dalam hal akurasi dan efisiensi. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem deteksi ancaman siber yang cepat dan akurat pada lingkungan IoMT.

Kata Kunci: *Internet_of_Things*, *Internet_of_Medical_Things*, *Distributed_Denial_of_Service*, *Machine Learning*.

1 Pendahuluan

Di era digital dengan konektivitas yang meluas, keamanan jaringan merupakan pertimbangan penting yang harus diperlakukan dengan hati-hati. *Internet of Things* (IoT) telah mempercepat kemajuan teknis dan menghasilkan peluang-peluang baru, tetapi juga menghadirkan masalah keamanan yang substansial. Serangan *Distributed Denial of Service* (DDoS) muncul sebagai salah satu bahaya paling serius bagi Internet of Things. Serangan ini mencoba membanjiri jaringan dengan *bandwidth* yang berlebihan, yang mengakibatkan gangguan layanan yang substansial, kerugian finansial, dan menurunkan kepercayaan konsumen.

Serangan DDoS menjadi lebih sering dan canggih seiring dengan kemajuan teknologi, membuat tindakan cepat menjadi prioritas utama bagi banyak perusahaan. Serangan ini berpotensi mengganggu kelangsungan bisnis, terutama untuk layanan yang sangat bergantung pada ketersediaan jaringan yang berkelanjutan. Pada kuartal kedua tahun 2024, jumlah total serangan mencapai 445.000, mencerminkan meningkat secara dramatis sebesar 46% dibandingkan periode yang sama tahun sebelumnya. Angka ini juga menunjukkan peningkatan sebesar 34% dibandingkan dengan total serangan pada kuartal ketiga dan keempat tahun 2023 [1]. Oleh karena itu, sangat penting bagi semua pihak terkait untuk terus mengembangkan langkah-langkah deteksi dan mitigasi yang lebih efektif.

Menurut Lionel Sujay Vailshery, seorang pakar riset elektronik konsumen, penggunaan gadget yang terhubung dengan IoT oleh konsumen diproyeksikan akan semakin pesat. Pada tahun 2030, jumlah perangkat IoT yang digunakan di seluruh dunia diperkirakan mencapai 25 miliar [2]. Dengan demikian, timbul kekhawatiran bahwa dengan meningkatnya jumlah perangkat yang terhubung maka risiko siber juga akan meningkat. Kenaikan ini menunjukkan betapa pesatnya penggunaan IoT di berbagai sektor, baik industri, rumah tangga, maupun kota pintar. IoT memainkan peran penting dalam transformasi digital, memungkinkan pengawasan, kontrol, dan otomatisasi jarak jauh dari berbagai perangkat yang saling terhubung. Dengan perkembangan teknologi dan infrastruktur yang mendukung, IoT berpotensi merevolusi cara manusia berinteraksi dengan dunia di sekitar mereka.

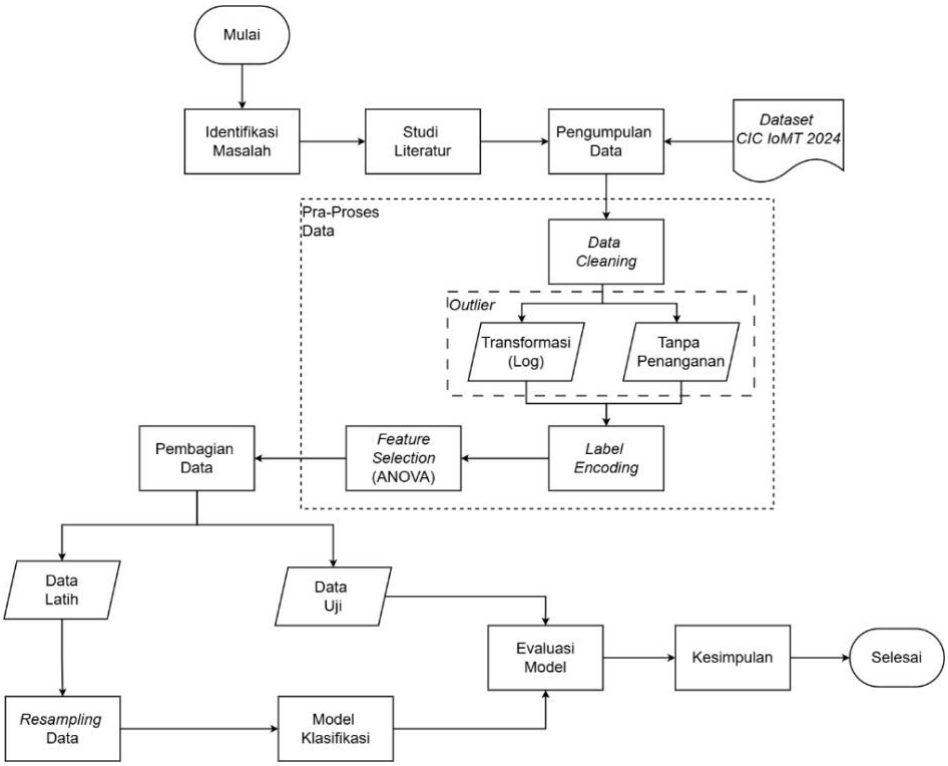
Internet of Medical Things (IoMT) adalah perluasan IoT di bidang kesehatan yang memungkinkan pemantauan terus-menerus data vital seperti tekanan darah dan detak jantung, sehingga meningkatkan kualitas hidup terutama

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

bagi lansia [3]. Seiring perannya yang semakin penting dalam pertukaran data kesehatan dan pengambilan keputusan, keamanan jaringan IoMT menjadi krusial untuk mencegah gangguan yang dapat membahayakan pasien. Dadkhah [4] merilis dataset CICIoMT2024 yang memuat informasi serangan terhadap perangkat IoMT sebagai kontribusi untuk riset keamanan.

Dalam mendeteksi serangan DDoS dengan pendekatan machine learning, evaluasi kinerja algoritma klasifikasi sangat penting karena efektivitasnya bervariasi, Berbagai algoritma *machine learning* telah digunakan untuk mendeteksi serangan DDoS. Menurut Dasari & Devarakonda [5], algoritma seperti Random Forest dan KNN menunjukkan hasil akurasi tinggi dalam klasifikasi lalu lintas jaringan. Ghourabi [6] juga mengembangkan model LightGBM untuk sistem kesehatan yang efektif dalam mendeteksi anomali jaringan. Salles & Domingos [7] juga menyoroti pentingnya pengembangan teknik machine learning dalam domain klasifikasi kompleks, termasuk keamanan siber. Penelitian ini bertujuan menilai kinerja beberapa metode klasifikasi dalam mendeteksi serangan DDoS menggunakan dataset CICIoMT2024, dengan fokus pada akurasi dan kecepatan komputasi, guna menemukan algoritma terbaik untuk diterapkan dalam lingkungan IoT khususnya IoMT. Temuan ini diharapkan memberikan kontribusi praktis dalam pembangunan sistem keamanan yang lebih efektif sekaligus memperkaya literatur keamanan jaringan.

2 Metode Penelitian



Gambar. 1. Tahapan Metode Penelitian

2.1 Identifikasi Masalah

Pada tahap identifikasi masalah, ditemukan bahwa serangan *Distributed Denial of Service* (DDoS) menjadi salah satu ancaman keamanan jaringan yang serius di era digital, terutama dengan semakin meluasnya penggunaan perangkat *Internet of Things* (IoT). Serangan ini bertujuan untuk membanjiri jaringan dengan lalu lintas berlebihan, sehingga mengakibatkan gangguan layanan yang signifikan. Meskipun berbagai metode telah dikembangkan untuk mendeteksi serangan DDoS, tidak semua algoritma klasifikasi memiliki efektivitas yang sama. Beberapa algoritma mungkin unggul dalam hal akurasi, sementara yang lain mungkin lebih efisien dalam penggunaan sumber daya atau lebih cepat dalam melakukan deteksi. Oleh karena itu, penelitian ini akan membandingkan kinerja beberapa algoritma klasifikasi dalam mendeteksi serangan DDoS berdasarkan data dari

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

CIC IoMT dataset. Penelitian ini bertujuan untuk menemukan algoritma yang paling efektif dan efisien dalam mengidentifikasi serangan DDoS, sehingga dapat memberikan pandangan bagi pengembangan sistem keamanan yang lebih baik di lingkungan IoT. Pada tahap ini, penulis dapat merumuskan masalah dan tujuan dari penelitian yang akan dilakukan.

2.2 Studi Literatur

Setelah menentukan rumusan masalah dan tujuan penelitian pada tahap identifikasi masalah, penulis melakukan tinjauan pustaka untuk membantu penulis menelaah tema-tema yang berkaitan dengan masalah yang akan diteliti. Studi literatur dilakukan dengan melihat berbagai sumber, termasuk jurnal, buku, dan materi lain yang relevan dengan topik penelitian yang sedang dibahas. Penelitian ini mengkaji literatur mengenai beberapa topik, antara lain serangan DDoS, IoT, IoMT, *machine learning*, *Random Forest*, *LightGBM, Naïve Bayes, K-Nearest Neighbors, Confusion Matrix,* dan materi-materi lain yang relevan.

2.3 Pengumpulan Data

Data yang digunakan pada penelitian ini merupakan hasil penelitian dari yang dilakukan Dadkhah et al. [4]. Data diambil berdasarkan pengambilan sampel menggunakan pengujian serangan pada perangkat IoT yang terhubung dalam jaringan dalam lingkup kesehatan, seperti alat infus, spirometer, fall detector, mouth air flow sensor, dan lain sebagainya. *Dataset* disebarkan ke publik pada situs CIC's, yang dimana ditujukan untuk pengembangan penelitian lainnya.

2.4 Pra-proses

Tahapan pra-proses data yang dilakukan untuk mempersiapkan data sebelum analisis lebih lanjut. Pada tahap ini dilakukan data cleaning yang bertujuan untuk memastikan tidak ada missing value dalam dataset yang akan digunakan. Kemudian untuk mendeteksi outlier dalam dataset dilakukan deteksi outlier dengan menggunakan pendekatan isolation Random Forest. Pada skenario yang akan dilakukan nantinya, terdapat dua metode yang akan diterapkan, yaitu penanganan data outlier menggunakan transformasi logaritma untuk mengurangi dampak nilai ekstrem, serta pendekatan tanpa penanganan outlier untuk melihat bagaimana data tersebut berperilaku tanpa intervensi. Suryanegara et al. [8] menunjukkan bahwa normalisasi dan transformasi data dapat meningkatkan kinerja klasifikasi. Kedua metode ini diharapkan dapat memberikan perbandingan yang jelas dalam analisis data dan menentukan dampaknya terhadap hasil yang diperoleh.

Kemudian, dilakukan tahapan encoding dengan menggunakan metode label encoding untuk mengubah kelas yang sebelumnya kategorikal menjadi bentuk numerik. Selanjutnya, dilakukan *feature selection* menggunakan pendekatan Analysis of variance (ANOVA), yaitu teknik ini membandingkan variasi antara rata-rata kelompok dengan varians dalam setiap kelompok, guna menentukan apakah perbedaan yang diamati melebihi apa yang mungkin terjadi secara kebetulan [9]. Pendekatan ini dipilih untuk memilih fitur-fitur yang paling relevan, dengan harapan dapat menghasilkan data yang berkualitas tinggi.

2.5 Pembagian Data

Kemudian data akan dipisahkan menjadi dua set yaitu data latih dan data uji, dengan tujuan untuk mengevaluasi kinerja model secara objektif dan adil. Tujuan dari pembagian ini adalah untuk memverifikasi bahwa model dapat diuji secara memadai dengan menggunakan data yang tidak disertakan dalam fase pelatihan. Prosedur pembagian data menggunakan 70% data latih untuk melatih model dan 30% data uji untuk menilai akurasi dan kemampuan generalisasi model terhadap data yang sebelumnya tidak terlihat.

2.6 Resampling Data

Dikarenakan terdapat ketidakseimbangan yang cukup signifikan antara data mayoritas dan data minoritas dalam *dataset*, dilakukan resampling data untuk menyeimbangkan distribusi data sebelum analisis lebih lanjut. Pada tahap ini terdapat empat skenario berbeda akan diterapkan untuk mengatasi masalah tersebut, yaitu *undersampling* dengan menggunakan *tomek link*, *oversampling* dengan menggunakan SMOTE (*Synthetic Minority Over-sampling Technique*), *hybrid sampling* dengan metode gabungan tomek link dan SMOTE, lalu tanpa resampling untuk memahami bagaimana model bekerja tanpa adanya intervensi dalam distribusi data. Pendekatan-pendekatan

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

ini bertujuan untuk menyeimbangkan distribusi kelas sehingga diharapkan model dapat belajar dari data dengan lebih efektif dan menghasilkan performa yang lebih baik pada tiap kelas [10].

2.7 Model Klasifikasi

Dalam penelitian ini, algoritma pemodelan yang digunakan untuk mengklasifikasikan jenis serangan DDoS meliputi *LightGBM, Random Forest, Naïve Bayes, dan K-Nearest Neighbors*. Deng [11] menyatakan bahwa LightGBM unggul dalam efisiensi waktu pada sistem deteksi intrusi di lingkungan IoT industri. Sementara itu, Haseeb-ur-Rehman et al. [12] menekankan pentingnya penggunaan algoritma yang cepat dan akurat untuk mendeteksi serangan DDoS pada jaringan berkecepatan tinggi. Pemilihan algoritma tersebut didasarkan pada karakteristik unik masing-masing dalam menangani berbagai jenis data dan permasalahan klasifikasi. Data latih dari setiap skenario yang telah disiapkan pada tahap sebelumnya akan digunakan untuk melatih masing-masing model, sehingga diharapkan dapat mengidentifikasi pola dan hubungan dalam data secara lebih efektif.

2.8 Evaluasi Model

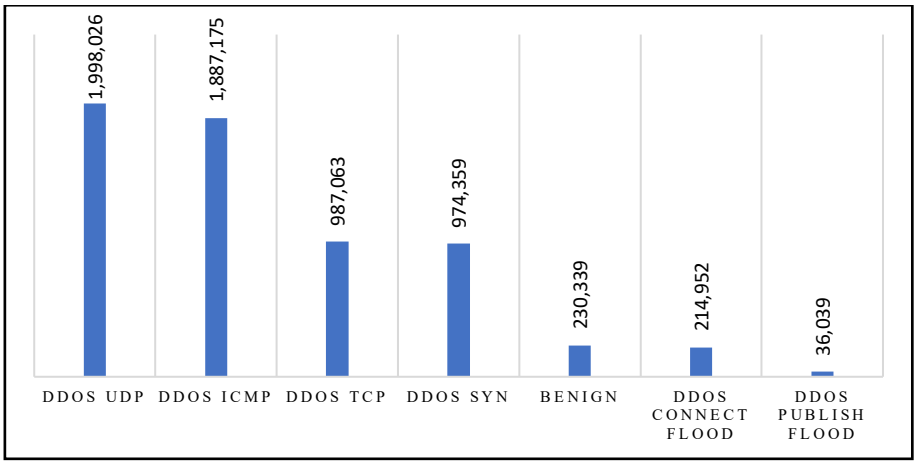
Pada tahap ini dilakukan proses evaluasi dengan tujuan untuk menilai seberapa akurat model yang telah dibangun dalam melakukan prediksi klasifikasi. Tran [13] menyarankan kombinasi metrik performa dan waktu sebagai pendekatan evaluasi yang adil dalam sistem klasifikasi. Dengan begitu model yang telah dikembangkan akan diuji menggunakan data uji untuk mengevaluasi baik tingkat akurasi prediksinya maupun efisiensi waktu komputasi yang diperlukan. Untuk memperoleh gambaran yang jelas tentang kinerja model, *confusion matrix* akan digunakan sebagai alat evaluasi utama. *Confusion matrix* memungkinkan untuk menghitung dengan rinci jumlah prediksi yang benar dan salah, serta mengidentifikasi pola kesalahan dalam prediksi model. Adapun faktor pendukung yang dihasilkan dari *confusion matrix* yaitu seperti nilai *accuracy*, *precision, recall, f1-score*.

2.9 Kesimpulan

Tahap ini bertujuan untuk menjawab rumusan masalah dengan menyimpulkan hasil evaluasi model berdasarkan metrik performa seperti *accuracy, precision, recall, F1-score*, dan waktu komputasi. Evaluasi ini memberikan gambaran menyeluruh tentang efektivitas prediksi dan efisiensi pemrosesan data. Untuk menentukan algoritma terbaik, digunakan metode weighted sum yang menggabungkan beberapa kriteria dengan bobot berbeda, di mana *accuracy* diberi bobot lebih besar (60%) dan waktu komputasi 40%, karena kesalahan deteksi serangan memiliki dampak lebih besar dibanding keterlambatan respons dalam konteks sistem *real-time* [14]. Semua temuan dirangkum secara komprehensif untuk memberikan pemahaman mendalam terhadap pencapaian dan implikasi penelitian.

3 Hasil dan Pembahasan

Dataset yang digunakan pada penelitian ini didapat dari penelitian yang berjudul "CICIoMT2024: *Attack Vectors in Healthcare devices-A Multi-Protocol Dataset for Assessing IoMT Device Security*" [4]. Setelah dilakukan eksplorasi dataset dan ditemukan total keseluruhan data yang digunakan mencapai 6.327.953 records pada gambar 2 merupakan jumlah jenis serangan yang ada dalam *dataset*.



ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

Gambar. 2. Visualisasi Persebaran Jumlah Data

Gambar 2 menampilkan jumlah data di setiap jenis serangan DDoS berbasis IoT dalam dataset. Kelas UDP DDoS memiliki *record* terbanyak, 1.998.026, diikuti oleh ICMP DDoS, TCP DDoS, dan SYN DDoS. Sebagai perbandingan, kelas seperti DDoS Publish Flood hanya memiliki 36.039 records. Perbedaan jumlah ini menunjukkan bahwa *dataset* tidak seimbang, dengan DDoS UDP mencakup 31,6% dari total data dan DDoS Publish Flood kurang dari 1%, sehingga membutuhkan pemrosesan khusus untuk memastikan bahwa model tidak bias terhadap kelas mayoritas.

Pada Tabel 1, dapat dilihat tampilan dari sejumlah data yang terdapat dalam dataset CICIoMT2024, yang memberikan gambaran umum mengenai informasi yang terkandung di dalamnya. *Dataset* ini juga terdiri dari 44 atribut yang saling terkait, yang masing-masing memiliki peran penting dalam analisis.

Tabel 1. Sampel Data

No.	Header_ Length	Protocol Type	Duration	Rate	…	Covariance	Variance	Weight	Class
0	13268.50	16.68	67.82	24531.591658	…	0.398892	0.02	141.55	DDoS UDP
1	54.00	6.00	64.00	60787.014493	…	0.000000	0.00	141.55	DDoS TCP
2	36757.00	17.00	64.00	14846.789926	…	0.000000	0.00	141.55	DDoS UDP
3	54	6.00	64.00	54.736391	…	0.000000	0.00	141.55	DDoS TCP
4	23229.00	17.00	64.00	18068.366797	…	0.000000	0.00	141.55	DDoS UDP
…	…	…	…	…	…	…	…	…	…
6327948	54.00	6.00	64.00	0.730193	…	0.000000	0.00	141.55	DDoS SYN
6327949	428.51	6.00	64.00	11.649699	…	32.817959	0.81	141.55	DDoS Connect Flood
6327950	0.00	1.00	64.00	350.284283	…	0.000000	0.00	141.55	DDoS ICMP
6327951	6570.50	17.00	64.00	26511.535478	…	0.000000	0.00	141.55	DDoS UDP
6327952	0.00	1.00	64.00	3.992558	…	0.000000	0.00	141.55	DDoS ICMP

3.1 Skenario Percobaan

Dalam penelitian ini dilakukan skema atau skenario percobaan yang dilakukan yaitu *dataset* utuh / awal, untuk *outlier* terdapat dua skenario yaitu dengan penanganan dan tanpa penanganan. Kemudian pada tahap *resampling* dilakukan empat skenario tambahan yaitu tanpa *resampling, undersampling, oversampling, hybrid sampling*. Dengan begitu total percobaan yang akan dilakukan menjadi delapan percobaan. Berikut gambaran skenario yang akan dilakukan.

Tabel 2. Skenario Percobaan

Scenarios	Outlier	Resampling
1	tanpa penanganan	tanpa resampling
2	tanpa penanganan	undersampling
3	tanpa penanganan	oversampling
4	tanpa penanganan	hybrid sampling
5	tanpa penanganan	tanpa resampling
6	tanpa penanganan	undersampling

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

7	tanpa penanganan	oversampling
8	tanpa penanganan	hybrid sampling

3.2 Pra-proses

Pra-proses adalah tahap awal penting dalam analisis data yang menyiapkan data agar siap digunakan dalam machine learning. Tahapan ini meliputi pembersihan data untuk menghilangkan nilai hilang (*missing value*), transformasi data, dan label encoding guna memastikan data berada pada skala yang tepat. Proses pra-pemrosesan yang baik dapat meningkatkan kualitas dan akurasi model. Pemeriksaan missing value menunjukkan bahwa dataset tidak mengandung nilai kosong, sehingga kualitas data sudah baik dan siap untuk analisis lebih lanjut.

Selanjutnya, dilakukan penanganan outlier dengan metode *log transform* pada fitur-fitur yang memiliki nilai besar dan distribusi eksponensial, sementara fitur kategorikal seperti protokol tidak diubah karena memiliki konteks khusus. Proses label *encoding* juga diterapkan pada data kategorikal dari variabel nominal atau ordinal untuk mengubahnya menjadi format numerik yang memudahkan pemodelan. Hasil dari pra-proses ini menghasilkan dataset yang siap digunakan untuk tahap analisis dan pemodelan berikutnya.

Tabel 3. Sampel Data Setelah Pra-Proses

No.	Header_ Length	Protocol Type	Duration	Rate	…	Covariance	Variance	Weight	Class
0	9.493223	16.68	4.231494	10.10776	…	0.33568	0.019803	4.95969	6
1	4.007333	6.0	4.174387	11.01515	…	0.0	0.0	4.95969	5
2	10.512111	17.0	4.174387	9.60561	…	0.0	0.0	4.95969	6
3	4.007333	6.0	4.174387	4.02063	…	0.0	0.0	4.95969	5
4	10.0532	17.0	4.174387	9.80197	…	0.0	0.0	4.95969	6
…	…	…	…	…	…	…	…	…	…
6327948	4.007333	6.0	4.174387	0.54823	…	0.0	0.0	4.95969	4
6327949	6.026645	6.0	4.174387	2.53763	…	3.520992	0.593327	4.95969	1
6327950	0.0	1.0	4.174387	5.8616	…	0.0	0.0	4.95969	2
6327951	8.790497	17.0	4.174387	10.18537	…	0.0	0.0	4.95969	6
6327952	0.0	1.0	4.174387	1.60795	…	0.0	0.0	4.95969	2

Dikarenakan terdapat banyak atribut atau fitur yang berfungsi sebagai indikator dalam penilaian klasifikasi, dilakukan pemilihan fitur yang relevan dan berdampak pada pemodelan klasifikasi menggunakan ANOVA. Dengan pendekatan ANOVA dengan memilih nilai f-statistic tertinggi dan nilai signifikan yang kurang dari 0.05 serta menggunakan nilai median untuk threshold dari f-statistic antar fitur yang bergunakan sebagai batasan nilai yang dapat diterima untuk dipilih sebagai fitur atau atribut yang digunakan dalam analisis klasifikasi nantinya.

3.3 Pembagian Data

Dilakukan pembagian data dengan membagi data menjadi dua bagian, yaitu data latih dan data uji dengan perbandingan rasio 70:30. Dalam pembagian ini, 70% dari total data sebanyak 6.327.953 baris digunakan untuk data latih, sementara 30% sisanya digunakan untuk data uji. Sehingga jumlah data latih sebanyak 4.429.567 dan jumlah data uji sebanyak 1.898.386. Pembagian ini bertujuan untuk memastikan bahwa model yang dibangun dapat terlatih dengan baik dan diuji secara efektif.

3.4 Resampling Data

Dikarenakan dataset yang digunakan bersifat *imbalance* atau tidak seimbang antar kelas nya. Oleh karena itu, diperlukan penangan data *imbalance* dengan menerapkan resample data. Pada penelitian ini, akan dilakukan skenario penelitian dengan data latih dari dua dataset sebelumnya menjadi delapan dataset, dengan melakukan skenario pada resampling data yaitu tanpa *resampling, undersampling* menggunakan *tomek link, oversampling* menggunakan SMOTE, dan *hybrid sampling* menggunakan *tomek link* dan SMOTE.

Proses tanpa *resampling* bertujuan untuk memahami bagaimana model bekerja tanpa adanya intervensi dalam distribusi data. Kemudian pada data latih dari kedua dataset yaitu tanpa penanganan *outlier* dan dengan penanganan *outlier* didapat informasi mengenai jumlah data yang sama, informasi lebih rinci dapat dilihat pada Tabel 4.

Tabel 4. Rincian Data Latih

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

No.	Class Name	Class Encoding	Number of Records	Percentage of Records
1	Benign	0	160850	4%
2	DDoS Connect Flood	1	150703	3%
3	DDoS ICMP	2	1320249	30%
4	DDoS Publish Flood	3	25132	1%
5	DDoS SYN	4	681951	15%
6	DDoS TCP	5	690774	16%
7	DDoS UDP	6	1399908	32%
	Total		**4429567**	**100%**

Pada proses *undersampling* digunakan pendekatan *tomek link*. *Tomek link* dipilih agar data yang tidak relevan atau *noise* dapat dihilangkan, sehingga meningkatkan kualitas *dataset* yang digunakan. Pada tabel 5 merupakan rincian jumlah data pada dataset tanpa penanganan *outlier* yang telah di *resampling* menggunakan *tomek link.*

Tabel 5. Rincian Data Latih Undersampling

No.	Class Name	Class Encoding	Records (Before)	% (Before)	Records (After)	% (After)
1	Benign	0	160850	4%	159668	4%
2	DDoS Connect Flood	1	150703	3%	150383	3%
3	DDoS ICMP	2	1320249	30%	1317450	30%
4	DDoS Publish Flood	3	25132	1%	25132	1%
5	DDoS SYN	4	681951	15%	677548	15%
6	DDoS TCP	5	690774	16%	686039	16%
7	DDoS UDP	6	1399908	32%	1397357	32%
	Total		**4429567**	**100%**	**4413577**	**100%**

SMOTE dipilih pada proses *oversampling* pada penelitian ini guna untuk menyeimbangkan data minoritas dengan data sintetis terhadap data mayoritas. Pada tabel 6 merupakan rincian jumlah data pada dataset tanpa penanganan outlier dan dengan penanganan outlier yang telah di *resampling* menggunakan SMOTE. Kedua *dataset* menghasilkan jumlah data yang sama setelah resampling.

Tabel 6. Rincian Data Latih Oversampling

No.	Class Name	Class Encoding	Records (Before)	% (Before)	Records (After)	% (After)
1	Benign	0	160850	4%	1399908	14%
2	DDoS Connect Flood	1	150703	3%	1399908	14%
3	DDoS ICMP	2	1320249	30%	1399908	14%
4	DDoS Publish Flood	3	25132	1%	1399908	14%
5	DDoS SYN	4	681951	15%	1399908	14%
6	DDoS TCP	5	690774	16%	1399908	14%
7	DDoS UDP	6	1399908	32%	1399908	14%
	Total		**4429567**	**100%**	**9799356**	**100%**

Pada proses *hybrid sampling* yaitu menggabungkan dua proses resampling yaitu *oversampling* dan *undersampling.* Pada *oversampling* digunakan pendekatan SMOTE dan pada *undersampling* digunakan pendekatan *tomek link.* Kombinasi kedua metode ini, diharapkan menghasilkan data latih yang dapat membuat model yang dibangun dapat lebih akurat dalam mengklasifikasikan data dari kelas yang kurang terwakili, serta

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

mengurangi risiko *overfitting* yang sering terjadi pada *dataset* yang tidak seimbang. Pada tabel 7 merupakan rincian jumlah data pada dataset tanpa penanganan *outlier* yang telah di *resampling* menggunakan SMOTE dan *tomek link.*

Tabel 7. Rincian Data Latih Hybrid Sampling

No.	Class Name	Class Encoding	Records (Before)	% (Before)	Records (After)	% (After)
1	Benign	0	160850	4%	1399812	14%
2	DDoS Connect Flood	1	150703	3%	1399386	14%
3	DDoS ICMP	2	1320249	30%	1398404	14%
4	DDoS Publish Flood	3	25132	1%	1399574	14%
5	DDoS SYN	4	681951	15%	1397446	14%
6	DDoS TCP	5	690774	16%	1397289	14%
7	DDoS UDP	6	1399908	32%	1397289	14%
	Total		**4429567**	**100%**	**9789200**	**100%**

Dengan demikian proses resampling ini menghasilkan delapan dataset untuk dilatih dalam pembuatan model. Untuk kedepannya penamaan *dataset* yang digunakan akan mengacu pada tabel 2 skenario percobaan dengan nama Dataset 1 sampai dengan Dataset 8 secara berurutan.

3.5 Pemodelan Klasifikasi

Setelah melalui proses sebelumnya, seperti pra-proses, pembagian data yang menjadi 70% data latih dan 30% data uji, dan resampling data dengan pendekatan *undersampling, oversampling* dan *hybrid sampling*, serta tanpa *resampling* sehingga menghasilkan delapan dataset latih yang akan menjadi bahan pemodelan menggunakan algoritma *machine learning* untuk mengklasifikasikan serangan DDoS pada IoMT.
Pemodelan dan evaluasi dilakukan dalam komputasi cloud menggunakan *Google Colabs*, yang menyediakan dukungan pemrosesan data secara paralel dengan GPU (*Graphics Processing Unit*) untuk mempercepat proses data baik pembuatan model maupun evaluasi model menggunakan data latih. Pada penelitian ini, jenis GPU yang digunakan adalah T4 GPU, yang tersedia secara default di platform *Google Colabs*.

3.5.1 Model *Random Forest*

Dalam pembuatan model klasifikasi menggunakan *Random Forest* dilakukan dengan pemrosesan GPU. *Random Forest* memiliki beberapa kriteria pemisah dalam pohon keputusan salah satunya gini. Karena dinilai lebih cepat dalam perhitungan serta efisiensi dalam komputasi sehingga gini sering digunakan dalam pemodelan pohon keputusan salah satunya *Random Forest* dan menjadi parameter bawaan dari pustaka yang akan digunakan. Pustaka yang digunakan yaitu "*RandomForestClassifier*" yang berasal dari "cuml.ensemble" yang dimana memungkinkan pemrosesan data menggunakan algoritma Random Forest yang berjalan pada GPU. Kemudian parameter yang digunakan dalam model terdiri dari max_features, max_depth, min_samples_leaf, min_samples_split, n_estimators, serta random_state.

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

Tabel 8. Parameter Model Klasifikasi Algoritma Random Forest

Parameter	Value
max_features	'sqrt'
max_depth	16
min_samples_leaf	1
min_samples_split	2
n_estimator	100
random_state	42

3.5.2 Model *LightGBM*

LightGBM mendukung beberapa jenis boosting antara lain, *Gradient Boosting Decision Tree* (gbdt), *Random Forest* (rf), dan *Dropouts meet Multiple Additive Regression Trees* (dart). Pada parameter bawaan pustaka LightGBM "gdbt" menjadi pilihan pada jenis boosting. Digunakan pustaka dari "lightgbm.LGBMClassifier", untuk parameter yang digunakan dalam model terdiri dari device_type, boosting, num_iterations, dan num_leaves.

Tabel 9. Parameter model klasifikasi algoritma LightGBM

Parameter	Value
device_type	'gpu'
boosting	'gdbt'
num_iterations	100
num_leaves	31

3.5.3 Model *Naïve Bayes*

Naïve Bayes terdapat beberapa pendekatan yaitu *Gaussian, Multinomial, Bernoulli.* Pada penelitian ini dipilih pendekatan Gaussian karena sering digunakan untuk fitur atau variabel yang memiliki nilai kontinu. Untuk melakukan pemrosesan secara paralel dengan GPU, digunakan pustaka "CuPy" yang memiliki fungsi yang sama seperti "NumPy". Karena pustaka yang mendukung *Naïve Bayes* yang dapat dijalankan secara GPU belum tersedia, maka dilakukan pembuatan fungsi pemodelan secara manual sesuai dengan prosedur pustaka Naïve Bayes pada sklearn dan parameter yang akan digunakan yaitu var_smooting dengan nilai "1e-9".

3.5.4 Model *K-Nearest Neighbors*

Terdapat beberapa query algoritma dalam algoritma KNN salah satunya "auto" merupakan parameter bawaan dari pustaka "cuML" untuk memilih secara otomatis query brute-force atau random ball berdasarkan bentuk dan metrik data. pustaka "KNeighborsClassifier" yang berasal dari "CuML" dan parameter yang digunakan yaitu n_neighbors dengan nilai "5".

3.6 Evaluasi

3.6.1 Evaluasi *Random Forest*

Hasil penilaian model dengan metode *Random Forest* untuk semua skenario percobaan dapat dilihat pada tabel 10. Tabel ini menampilkan parameter penilaian seperti *accuracy, precision, recall, dan F1-score* serta waktu komputasi untuk menunjukkan keberhasilan model dalam mengklasifikasi serangan DDoS pada IoMT.

Tabel 10. Hasil Evaluasi Klasifikasi Random Forest

Dataset Name	Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1-Score	Train Time (s)	Train Predict Time (s)	Test Predict Time (s)
Dataset 1	0.999651	0.999472	0.999472	0.999472	0.999472	31.68	1.91	1.26
Dataset 2	0.999705	0.999394	0.999396	0.999394	0.999394	28.49	1.47	0.86

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

Dataset 3	0.999517	0.999317	0.999322	0.999317	0.999319	71.12	2.44	0.70
Dataset 4	0.999548	0.999314	0.999319	0.999314	0.999315	70.87	1.94	0.66
Dataset 5	0.999757	0.999572	0.999572	0.999572	0.999572	28.23	1.15	0.61
Dataset 6	0.999810	0.999550	0.999551	0.999550	0.999550	27.78	1.30	0.63
Dataset 7	0.999535	0.999409	0.999414	0.999409	0.999411	70.74	2.07	0.66
Dataset 8	0.999556	0.999403	0.999407	0.999403	0.999404	70.41	1.85	0.61

3.5.2 Evaluasi *LightGBM*

Hasil penilaian model dengan metode *LightGBM* untuk semua skenario percobaan dapat dilihat pada tabel 11. Tabel ini menampilkan parameter penilaian seperti *accuracy, precision, recall, dan F1-score* serta waktu komputasi untuk menunjukkan keberhasilan model dalam mengklasifikasi serangan DDoS pada IoMT.

Tabel 11. Hasil Evaluasi Klasifikasi LightGBM

Dataset Name	Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1-Score	Train Time (s)	Train Predict Time (s)	Test Predict Time (s)
Dataset 1	0.840768	0.840359	0.880080	0.840359	0.789860	4.42	3.91	2.32
Dataset 2	0.463389	0.464250	0.687910	0.464250	0.384031	4.89	4.12	3.22
Dataset 3	0.999962	0.999658	0.999658	0.999658	0.999658	14.31	5.74	2.52
Dataset 4	0.868963	0.855780	0.895764	0.855780	0.812638	15.31	5.95	3.52
Dataset 5	0.996146	0.995947	0.995954	0.995947	0.995938	4.31	3.74	2.20
Dataset 6	0.916593	0.916451	0.930565	0.916451	0.915065	4.91	4.28	3.71
Dataset 7	0.999166	0.998812	0.998821	0.998812	0.998814	14.56	5.82	2.34
Dataset 8	0.998488	0.998211	0.998219	0.998211	0.998213	15.27	6.21	3.12

3.5.3 Evaluasi *Naïve Bayes*

Hasil penilaian model dengan metode *Naïve Bayes* untuk semua skenario percobaan dapat dilihat pada tabel 12. Tabel ini menampilkan parameter penilaian seperti *accuracy, precision, recall, dan F1-score* serta waktu komputasi untuk menunjukkan keberhasilan model dalam mengklasifikasi serangan DDoS pada IoMT.

Tabel 12. Hasil Evaluasi Klasifikasi Naïve Bayes

Dataset Name	Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1-Score	Train Time (s)	Train Predict Time (s)	Test Predict Time (s)
Dataset 1	0.992774	0.992799	0.993395	0.992799	0.992986	0.46	1.57	0.01
Dataset 2	0.993715	0.992071	0.992819	0.992071	0.992303	0.29	1.56	0.01
Dataset 3	0.985282	0.992878	0.993466	0.992878	0.993061	0.49	3.40	0.01
Dataset 4	0.985588	0.992692	0.993319	0.992692	0.992886	1.13	3.66	0.01
Dataset 5	0.991378	0.991437	0.992755	0.991437	0.991842	0.55	1.51	0.01
Dataset 6	0.991369	0.991182	0.992553	0.991182	0.991602	0.27	1.50	0.01
Dataset 7	0.985052	0.991317	0.992638	0.991317	0.991724	0.47	3.33	0.01
Dataset 8	0.985043	0.991234	0.992571	0.991234	0.991645	3.06	3.97	0.01

3.5.4 Evaluasi *K-Nearest Neighbors*

Hasil penilaian model dengan metode *K-Nearest Neighbors* untuk semua skenario percobaan dapat dilihat pada tabel 13. Tabel ini menampilkan parameter penilaian seperti *accuracy, precision, recall, dan F1-score* serta waktu komputasi untuk menunjukkan keberhasilan model dalam mengklasifikasi serangan DDoS pada IoMT.

Tabel 13. Hasil Evaluasi Klasifikasi KNN

Dataset Name	Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1-Score	Train Time (s)	Train Predict Time (s)	Test Predict Time (s)
Dataset 1	0.995250	0.993314	0.993513	0.993314	0.993392	3.83	999.71	428.27
Dataset 2	0.996767	0.993081	0.993523	0.993081	0.993237	0.08	990.72	425.82
Dataset 3	0.996428	0.993760	0.994592	0.993760	0.994018	3.53	5227.54	992.39
Dataset 4	0.996847	0.993634	0.994545	0.993634	0.993917	0.14	5214.84	992.92
Dataset 5	0.999563	0.999353	0.999353	0.999353	0.999353	0.07	1011.96	431.02

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

Dataset 6	0.999661	0.999332	0.999332	0.999332	0.999332	0.07	1003.39	428.36
Dataset 7	0.999813	0.999425	0.999426	0.999425	0.999425	4.01	4949.53	952.80
Dataset 8	0.999827	0.999421	0.999423	0.999421	0.999422	0.14	4974.30	952.54

3.6 Pembahasan

Berdasarkan hasil evaluasi yang dihasilkan oleh model yang diuji seperti *Random Forest, LightGBM, naïve bayes, K-Nearest Neighbors* pada klasifikasi serangan DDoS. Berikut nilai perbandingan rata – rata dari hasil evaluasi yang dihasilkan oleh keempat algoritma yang diuji dapat dilihat pada tabel berikut.

Tabel 14. Rata – Rata Hasil Evaluasi Algoritma Klasifikasi

Algorithm	Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test F1-Score	Train Time (s)	Train Predict Time (s)	Test Predict Time (s)
Random Forest	0.999635	0.999429	0.999432	0.999429	0.999430	49.92	1.77	0.75
LightGBM	0.885434	0.883684	0.923371	0.883684	0.861777	9.75	4.97	2.87
Naïve Bayes	0.988775	0.991951	0.992940	0.991951	0.992256	0.84	2.56	0.01
K-Nearest Neighbors	0.998020	0.996415	0.996713	0.996415	0.996512	1.48	3046.50	700.52

Berdasarkan tabel 14, evaluasi rata-rata empat algoritma *Random Forest, LightGBM, Naïve Bayes,* dan *K-Nearest Neighbors (KNN)* dalam mengklasifikasikan serangan DDoS pada IoMT menunjukkan perbedaan signifikan pada akurasi, waktu komputasi, dan kualitas klasifikasi. *Random Forest* unggul dengan akurasi hampir sempurna (0.999429) dan metrik evaluasi tinggi, meskipun *Train Time* relatif lama (49,92 detik). LightGBM lebih cepat melatih model (9,75 detik) tetapi memiliki akurasi lebih rendah (0.883684). Naïve Bayes sangat efisien dalam Train Time (0,84 detik) dan prediksi (0,01 detik) dengan akurasi yang cukup tinggi (0.991951). Sementara itu, KNN memiliki akurasi tinggi (0.996415) namun *Test Predict Time* sangat lama (700,52 detik), sehingga kurang cocok untuk aplikasi *real-time*. Pemilihan algoritma terbaik tergantung pada kebutuhan antara akurasi dan efisiensi waktu.

Penelitian ini sejalan dengan temuan dari berbagai penelitian sebelumnya. Dadkhah et al. [4] meneliti dataset CICIoMT2024 dan menemukan bahwa Random Forest memiliki performa terbaik, dengan akurasi 99,84% pada klasifikasi biner dan 99,55% pada klasifikasi multiclass, yang sebanding dengan akurasi 99,94% pada data uji penelitian ini. Sebaliknya, LightGBM, yang dinyatakan memiliki kinerja yang sangat tinggi oleh Ghourabi [6], dengan akurasi dan *F1-score* mendekati 99,99% pada dataset ToN-IoT, menunjukkan kinerja terburuk dalam penelitian ini, dengan akurasi 88,36% dan F1 86,17%. Metode KNN juga menunjukkan akurasi yang baik dalam studi Dasari [5] dan Haribalaji [15] yang mengamati akurasi berkisar antara 99,6% hingga 99,97% tetapi membutuhkan waktu pemrosesan yang lama hingga 3390 detik. Demikian pula, dalam penelitian ini, KNN mencapai akurasi 99,64% tetapi memiliki waktu prakiraan hingga 700 detik, sehingga tidak cocok untuk aplikasi waktu nyata.

Dalam penelitian ini, Naïve Bayes berkinerja baik dengan akurasi 99,20% dan waktu prediksi 0,01 detik. Hasil ini serupa dengan hasil studi Dasari [5], yang menemukan akurasi 99,58% dan waktu pemrosesan 0,35 detik. Haseeb-ur-Rehman [12] menekankan pentingnya efisiensi dalam mendeteksi DDoS pada jaringan berkecepatan tinggi. Tran [13] juga menekankan pentingnya pemeriksaan komprehensif terhadap kebenaran dan waktu komputasi, terutama dalam pengaturan data berdimensi tinggi seperti CICIoMT2024. Dengan demikian, kontribusi utama dari penelitian ini adalah untuk memvalidasi kinerja berbagai algoritma dalam konteks IoMT, serta eksplorasi keampuhan komputasi berbasis GPU dalam konteks dataset berskala besar.

Untuk memilih algoritma terbaik yang dapat digunakan secara *real-time* dalam mendeteksi serangan DDoS pada lingkungan IoT dan IoMT, dilakukan evaluasi berdasarkan rata-rata akurasi dan waktu komputasi dengan pembobotan 60% untuk akurasi dan 40% untuk waktu komputasi, guna mencapai keseimbangan antara ketepatan deteksi dan efisiensi proses. Sebelum pembobotan, data dinormalisasi menggunakan min-max scaler agar semua kriteria, baik *benefit* seperti akurasi maupun *cost* seperti waktu komputasi, memiliki skala yang seragam antara 0 dan 1. Normalisasi ini memastikan nilai akurasi yang diinginkan semakin tinggi dan waktu komputasi yang lebih rendah tercermin secara proporsional. Selanjutnya, skor total dihitung menggunakan metode weighted sum, yang

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)

menggabungkan nilai normalisasi dari kriteria tersebut sesuai bobotnya menjadi skor tunggal untuk pengambilan keputusan. Hasil perhitungan bobot ini kemudian digunakan untuk menentukan algoritma dengan performa terbaik, sebagaimana ditampilkan pada tabel evaluasi berikut.

Tabel 15. Hasil Pembobotan

Algorithm	Score
Random Forest	0.971578
Naïve Bayes	0.961235
K-Nearest Neighbors	0.584376
LightGBM	0.393285

4 Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan hasil pengujian klasifikasi serangan DDoS pada IoMT menggunakan *dataset* CICIoMT2024, algoritma *Random Forest* menunjukkan performa terbaik dengan akurasi 0.999635 pada data latih dan 0.999429 pada data uji, serta waktu komputasi 49.92 detik untuk pelatihan dan 0.49 detik untuk pengujian, menghasilkan skor akhir tertinggi sebesar 0.971578. *Naïve Bayes* mencatat akurasi 0.988775 (latih) dan 0.991951 (uji), dengan waktu komputasi sangat efisien yaitu 0.84 detik (latih), 2.56 detik (uji latih), dan hanya 0.01 detik (uji), sehingga memperoleh skor akhir 0.961235. *K-Nearest Neighbors* memiliki akurasi tinggi sebesar 0.998020 (latih) dan 0.996415 (uji), namun memerlukan waktu komputasi yang sangat lama, yakni 1.57 detik (latih), 3046.5 detik (uji latih), dan 700.52 detik (uji), menghasilkan skor akhir lebih rendah yaitu 0.584376. Sementara itu, *LightGBM* menunjukkan performa terendah dengan akurasi 0.885434 (latih) dan 0.883684 (uji), serta waktu komputasi 9.75 detik (latih), 2.73 detik (uji latih), dan 2.75 detik (uji), dengan skor akhir paling rendah sebesar 0.393285.

Studi ini memberikan gambaran awal yang kuat tentang kemampuan teknik *machine learning* dalam mengidentifikasi serangan DDoS pada sistem IoMT. Namun, ruang lingkup eksperimen tetap terbatas pada simulasi, tanpa pengujian langsung pada perangkat IoMT asli. Hasilnya, temuan ini harus dievaluasi dalam pengaturan operasi dunia nyata untuk menentukan ketahanan sistem terhadap perubahan topologi jaringan, perangkat IoMT yang beragam, dan potensi ancaman baru seperti serangan *zero-day*.

4.2 Saran

Berdasarkan hasil penelitian, beberapa saran untuk penelitian selanjutnya dalam meningkatkan performa model klasifikasi serangan DDoS pada IoMT antara lain meliputi penerapan teknik pra-pemrosesan yang lebih beragam, seperti metode *resampling* untuk meningkatkan presisi pada kelas minoritas; penerapan teknik hyperparameter tuning guna mengoptimalkan metrik performa seperti akurasi, presisi, recall, F1-score, dan efisiensi waktu komputasi; serta eksplorasi algoritma *machine learning* lain untuk menemukan model yang lebih unggul dalam klasifikasi.

Selain itu, penelitian lebih lanjut diperlukan untuk membangun model berbasis *deep learning* seperti CNN, LSTM, atau Transformer, yang memiliki kapasitas untuk menemukan pola yang kompleks dalam lalu lintas jaringan. Pengujian dalam konteks dunia nyata juga diperlukan untuk menilai fleksibilitas dan ketergantungan sistem dalam menghadapi ancaman yang realistis dan beragam.

Referensi

[1] E. W. Ramadani, R. D. K. A. Harahap, and R. Fibriani, “Cybercrime Punishment Formulation Using Methods DDoS Attack Regarding Websites from a Positive Legal Perspective,” J. Huk. Prasada, vol. 12, no. 1, pp. 26–35, 2025, doi: 10.22225/jhp.12.1.2025.26-35.

[2] F. Prasetyo Eka Putra, S. Mellyana Dewi, and A. Hamzah, “Jurnal Sistim Informasi dan Teknologi https://jsisfotek.org/index.php Privasi dan Keamanan Penerapan IoT Dalam Kehidupan Sehari-Hari : Tantangan dan Implikasi,” J. Sistim Inf. dan Teknol., vol. 5, no. 2, pp. 26–32, 2023, doi: 10.37034/jsisfotek.v5i1.232.

[3] F. Kamalov, B. Pourghebleh, M. Gheisari, Y. Liu, and S. Moussa, “Internet of Medical Things Privacy and Security: Challenges, Solutions, and Future Trends from a New Perspective,” Sustain., vol. 15, no. 4, 2023, doi:

ISSN: 2655-139X (ONLINE)

ISSN: 0216-4221 (PRINT)

10.3390/su15043317.

[4] S. Dadkhah, E. C. P. Neto, R. Ferreira, R. C. Molokwu, S. Sadeghi, and A. Ghorbani, "CICIoMT2024: Attack Vectors in Healthcare devices-A Multi-Protocol Dataset for Assessing IoMT Device Security," J. Comput. Commun., 2024, doi: 10.20944/preprints202402.0898.v1.

[5] K. B. Dasari and N. Devarakonda, "Detection of DDoS Attacks Using Machine Learning Classification Algorithms," Int. J. Comput. Netw. Inf. Secur., vol. 14, no. 6, pp. 89–97, 2022, doi: 10.5815/ijcnis.2022.06.07.

[6] A. Ghourabi, "A Security Model Based on LightGBM and Transformer to Protect Healthcare Systems From Cyberattacks," IEEE Access, vol. 10, pp. 48890–48903, 2022, doi: 10.1109/ACCESS.2022.3172432.

[7] M. Salles and F. M. C. B. Domingos, "Towards the next generation of species delimitation methods: an overview of Machine Learning applications," pp. 1–17, 2023, doi: https://doi.org/10.32942/X2W313.

[8] G. A. B. Suryanegara, Adiwijaya, and M. D. Purbolaksono, "Peningkatan Hasil Klasifikasi pada Algoritma Random Forest untuk Deteksi Pasien Penderita Diabetes Menggunakan Metode Normalisasi," J. RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 5, no. 1, pp. 114–122, 2021, doi: 10.29207/resti.v5i1.2880.

[9] P. Bruce, A. Bruce, P. Gedeck, and an O. M. C. Safari, Practical Statistics for Data Scientists, 2nd Edition, 2nd ed., vol. 2. Sebastopol: O'Reilly Media, Inc., 2020. [Online]. Available: https://www.oreilly.com/library/view/practical-statistics-for/9781491952955/

[10] E. Erlin, Y. Desnelita, N. Nasution, L. Suryati, and F. Zoromi, "Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak seimbang," MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput., vol. 21, no. 3, pp. 677–690, 2022, doi: 10.30812/matrik.v21i3.1726.

[11] Y. Deng, "Design of Industrial IoT Intrusion Security Detection System Based on LightGBM Feature Algorithm and Multi-layer Perception Network," J. Cyber Secur. Mobil., vol. 13, no. 2, pp. 327–348, 2024, doi: 10.13052/jcsm2245-1439.1327.

[12] R. M. A. Haseeb-ur-rehman et al., "High-Speed Network DDoS Attack Detection: A Survey," Sensors, vol. 23, no. 15, 2023, doi: 10.3390/s23156850.

[13] C. T. Tran, "Ensemble Learning Approaches for Classification With High-Dimensional Data," J. Sci. Tech., vol. 12, no. 01, pp. 83–96, 2023, doi: 10.56651/lqdtu.jst.v12.n1.659.ict.

[14] I. Almomani, A.; Alazab, M.; Alharkan, "A hybrid model for detecting DDoS attacks in cloud computing environments," Comput. Secur., 2020, doi: 10.1016/j.cose.2020.101603.

[15] S. Haribalaji and P. Ranjana, "Distributed Denial of Service (DDOS) Attack Detection Using Classification Algorithm," in 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), IEEE, Apr. 2024, pp. 1–6. doi: 10.1109/ADICS58448.2024.10533510.

ISSN: 2655-139X (ONLINE)
ISSN: 0216-4221 (PRINT)