

KLASIFIKASI SERANGAN DDOS MENGGUNAKAN RECURSIVE FEATURE ELIMINATION DAN GRADIENT BOOSTING

Candra Adi Lesmana¹, Lukman Hakim²

¹²Program Studi Teknik Informatika, Universitas Yudharta Pasuruan

Jl. Yudharta No.7, Kembangkuning, Sengonangun, Kec. Purwosari, Pasuruan, Jawa Timur 67162

¹candraadi251@gmail.com, ² lukman@yudharta.ac.id

Abstract

Internet security is becoming an important challenge along with the rapid growth of information technology services. Distributed Denial of Service (DDoS) attacks are one of the serious threats that can crash servers and network systems by flooding the network with excessive packets or requests. In this study, the Recursive Feature Elimination (RFE) feature selection method is applied to the DDoS attack classification process using the Gradient Boosting algorithm. The purpose of this study is to apply the recursive feature elimination (RFE) feature selection algorithm to reduce the number of features in the data. The test results show that the use of Gradient Boosting with the best parameters produces very good performance, with an accuracy of 99.9%, a precision of 99.8%, a recall of 99.9%, and an F1 value of 99.9%. The combination of Recursive Feature Elimination (RFE) method with Gradient Boosting, where the 10 best features are selected, does not significantly affect the model performance results, but the application of this feature selection managed to reduce the computation time significantly from 5463.24 seconds to 1023.19 seconds without sacrificing model performance. These results indicate that the right feature selection method is very important in improving computational efficiency without sacrificing model performance in DDoS attack detection.

Keywords : Feature selection, Recursive Feature Elimination, Gradient boostig, Distrubuted Denial of Service (DDoS).

Abstrak

Keamanan internet menjadi tantangan penting seiring dengan pertumbuhan layanan teknologi informasi yang semakin pesat. Serangan *Distributed Denial of Service (DDoS)* merupakan salah satu ancaman serius yang dapat menyebabkan *crash* pada server dan sistem jaringan dengan cara membanjiri jaringan dengan paket atau permintaan yang berlebihan. Dalam penelitian ini, diterapkan metode seleksi fitur *Recursive Feature Elimination (RFE)* pada proses klasifikasi serangan DDoS menggunakan algoritma *Gradient Boosting*. Tujuan Penelitian ini untuk melakukan penerapan algoritma seleksi fitur *Recursive Feature Elimination (RFE)* untuk mengurangi jumlah fitur yang ada dalam data. Hasil pengujian menunjukkan bahwa penggunaan *Gradient Boosting* dengan parameter terbaik menghasilkan kinerja yang sangat baik, dengan akurasi sebesar 99,9%, presisi 99,8%, *recall* 99,9%, dan nilai F1 sebesar 99,9%. Kombinasi metode *Recursive Feature Elimination (RFE)* dengan *Gradient Boosting*, di mana 10 fitur terbaik dipilih, tidak mempengaruhi hasil kinerja model secara signifikan, tetapi penerapan seleksi fitur ini berhasil mengurangi waktu komputasi secara signifikan dari 5463,24 detik menjadi 1023,19 detik tanpa mengorbankan kinerja model. Hasil ini menunjukkan bahwa metode seleksi fitur yang tepat sangat penting dalam meningkatkan efisiensi komputasi tanpa mengorbankan performa model dalam deteksi serangan DDoS.

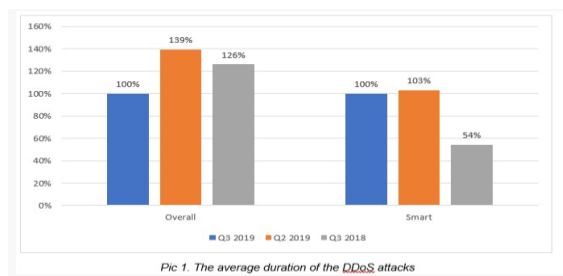
Kata kunci : Seleksi fitur, Recursive Feature elimination, Gradient boostig, Distrubuted Denial of Service (DDoS).

1. PENDAHULUAN

Perkembangan teknologi yang semakin maju membuat individu semakin bergantung pada penggunaan komputer. Sistem komputer digunakan dalam berbagai aspek kehidupan, termasuk industri, perbankan, dan kesehatan. Umumnya, beberapa jenis jaringan digunakan untuk menghubungkan sistem komputer ini satu sama lain. Hal ini tidak diragukan lagi memungkinkan timbulnya masalah keamanan dalam sistem jaringan komputer. Ada banyak insiden jaringan yang mempengaruhi seluruh dunia dalam beberapa tahun terakhir. Untuk meningkatkan keamanan jaringan komputer, beberapa kejadian jaringan ini ditangani dengan alat yang dapat mengidentifikasi serangan jaringan[1].

Keamanan internet adalah masalah yang signifikan, terutama di dunia modern di mana kebutuhan akan layanan teknologi informasi meningkat setiap hari. Serangan *Distributed Denial of Services* (DDoS) adalah salah satu dari beberapa masalah keamanan [2]. Serangan yang dikenal sebagai *Distributed Denial-of-Service* (DDoS) membanjiri jaringan dengan paket atau permintaan, yang membuat crash pada server, dan system di jaringan. Sistem jaringan bertambah banyak penggunaannya sebagai hasil dari perkembangannya. Oleh karena itu, membedakan antara pengguna yang sah dan peretas menjadi sangat sulit [3].

Salah satu jenis serangan yang paling berisiko adalah DDoS, seperti yang ditunjukkan oleh sebuah insiden yang terjadi di Hong Kong pada minggu kedua bulan Agustus 2011, seperti yang dilaporkan oleh *cyber threat.id*. Serangan DDoS membuat *server* HKExnews.hk tidak dapat beroperasi. Insiden tersebut terjadi pada tanggal 12-13 Agustus 2011. Dengan total nilai pasar sebesar HK\$ 1,5 triliun, HSBC, *Cathay Pacific Airways*, dan HKEx sendiri termasuk di antara perdagangan signifikan yang terhenti di pasar keuangan terbesar ketiga di Asia ini [4].



Gambar 1. Durasi rata-rata serangan DDoS

Pada gambar 1 di atas menunjukkan serangan DDoS pada tahun 2019, serangan DDoS meningkat sebesar 30% pada triwulan tiga tahun 2019 dibandingkan dengan triwulan sebelumnya dan sebesar 32% jika dibandingkan dengan triwulan ketiga tahun 2018, menurut statistik yang dikumpulkan oleh *Kaspersky DDoS Protection* pada tanggal 11 November 2019. Peningkatan tajam dalam aktivitas DDoS di awal tahun ajaran baru menjadi penyebab perubahan dramatis ini. Menurut Kaspersky, mayoritas serangan DDoS (53%) ditemukan pada bulan September 2019, dan 60% serangan yang berhasil dihentikan selama periode ini menargetkan institusi pendidikan dan situs web jurnal elektronik [4].

Membangun sistem deteksi *Intrusion Detection System* (IDS) untuk memonitor lalu lintas jaringan atau aliran data dari ancaman *intruder* (penyerang) merupakan salah satu upaya yang dilakukan untuk menjaga jaringan dari ancaman serangan. *Intrusion detection* merupakan proses mengawasi aktivitas pada sistem komputer atau jaringan dan mengevaluasinya untuk mengidentifikasi potensi insiden yang mungkin merupakan pelanggaran atau hampir melanggar pedoman penggunaan yang diterima, kebijakan keamanan komputer, atau prosedur keamanan konvensional. Untuk menilai apakah sebuah lalu lintas aman, mencurigakan, atau bahkan mengindikasikan adanya serangan, maka IDS digunakan untuk mengidentifikasi traffic data pada jaringan komputer [5].

Penggunaan *machine learning* dengan model prediksi sering menjadi cara efektif dalam mengklasifikasi berbagai jenis kasus [6]. Klasifikasi serangan DDoS adalah salah satu cara yang sangat penting dalam upaya mencegah serangan siber yang mencoba untuk merusak keamanan jaringan. Tujuan dari mengklasifikasi serangan ddos yaitu memahami berbagai jenis ancaman yang mungkin terjadi serta memahami metode yang digunakan oleh penyerang. Dengan memahami jenis serangan yang terjadi sebuah perusahaan bisa melakukan klasifikasi serangan untuk bisa menilai dampak yang mungkin bisa saja terjadi, misalnya kerugian dalam reputasi maupun kerugian Secara finansial, sehingga memungkinkan untuk sebuah perusahaan dalam mengalokasikan sumber daya untuk membangun dan meningkatkan keamanan jaringan yang lebih kokoh dari pada sebelumnya.

Dalam Penelitian ini penulis menggunakan algoritma *gradient boosting* dan seleksi fitur yaitu

algoritma *recursive feature elimination* untuk meningkatkan nilai akurasi dan performa.

Algoritma *gradient boosting* memiliki kelebihan dalam akurasi yang tinggi, karena metode *ensemble* yang satu ini menggabungkan beberapa *weak learner* untuk membangun sebuah model yang *strong learner*. Sehingga algoritma *gradient boosting* bisa menghasilkan sebuah model yang mempunyai nilai akurasi yang tinggi.

Namun, *gradient boosting* tidak efektif untuk memproses data dengan jumlah fitur (kolom) yang banyak. Menambahkan lebih banyak fitur akan membuat sistem menjadi lebih kompleks dan membutuhkan lebih banyak waktu komputasi karena setiap fitur akan digunakan dalam proses pelatihan [7].

Maka dari itu penulis mencoba menerapkan algoritma seleksi fitur *recursive feature elimination* untuk mengatasi kekurangan tersebut. Keunggulan dari penerapan algoritma *recursive feature elimination* tidak hanya untuk meningkatkan nilai akurasi dan performa, tetapi juga bisa mempermudah model lebih fokus terhadap subset fitur. Algoritma *recursive feature elimination* juga bisa digunakan untuk memilih fitur yang penting dalam data dan menghilangkan fitur yang kurang berkontribusi dalam klasifikasi.

Penelitian terdahulu yang dilakukan oleh fauzi dan teman-teman melakukan klasifikasi serangan DDoS menunjukkan 99% akurasi, 99% presisi, dan 99% gain. Dengan waktu komputasi 84,2 detik, model ini juga membutuhkan waktu paling lama untuk dilatih. Sebaliknya, model *adaboost* dapat dilatih dalam waktu kurang dari sepuluh detik [4]. Sedangkan Penelitian yang dilakukan oleh Zidane memperoleh hasil cukup akurat untuk mengklasifikasikan serangan DDoS, dengan nilai akurasi sebesar 95% dengan sistem metode *naïve bayes* [3].

Selain itu penelitian yang dilakukan oleh nasution dan basuki menggunakan algoritman C5.0 menghasilkan nilai akurasi rata-rata terbesar, yaitu 96.41%, yang dipegang oleh perbandingan 90% data latih dan 10% data uji. Jumlah trial. Semakin banyak uji coba yang dilakukan, semakin baik dampaknya terhadap akurasi. Dengan 100 trial, rata-rata akurasi paling tinggi adalah 98,78% [2].

Penelitian terkait *gradient boosting* juga pernah dilakukan oleh sri diantika dan teman-teman dengan judul "Implementasi Multi Class Gradient Boosting Untuk Mengklasifikasikan Jenis Hewan Pada Kebun Binatang". Klasifikasi jenis hewan pada kebun binatang yang memiliki *multi-class* menggunakan model *gradient boosting*

classifier, klasifikasi terbaik didapatkan dengan menerapkan parameter *boosting* dengan *n_estimators* sebanyak 50, *max_depth* 3, *sub sample* sebanyak 1.0, *learning rate* sebesar 0.1, dan menggunakan *criterion friedman Mse*. Selanjutnya dengan menggunakan *split validation*, yang membagi data ke dalam training dan testing set masing-masing sebesar 80% dan 20%. Untuk mengevaluasi model yang dibangun, penulis menggunakan *performance metrics* termasuk akurasi, *recall*, *F1-score*, *ROC curva*, dan *Mse* untuk menilai model yang dibuat. Hasil temuan menyatakan model yang disarankan mengungguli sejumlah model lain yang diuji, dengan akurasi 93,75%, *recall* 94%, presisi 96%, dan *mean square error* (MSE) yang mengukur kesalahan rata-rata pada rentang klasifikasi 12,5%; semakin rendah nilai MSE, semakin baik kinerja klasifikasi model [9].

Suryana dan teman-teman juga melakukan Penelitian dengan *gradient boosting* yang berjudul "PENERAPAN GRADIENT BOOSTING DENGAN HYPEROPT UNTUK MEMPREDIKSI KEBERHASILAN TELEMARTETING BANK". Terbukti dari diskusi dan temuan bahwa *hyperopt* dapat digunakan untuk tuning *hyperparameter* pada ruang pencarian yang besar. Dengan menggunakan penyetelan *hyperparameter*, algoritma *gradient boosting* memprediksi kesuksesan telemarketing dengan akurasi 90,39%, presisi 94,91%, dan AUC 0,939 pada data uji. Mengingat pencapaian ini relatif sangat baik, dapat dikatakan bahwa model ini melakukan pekerjaan yang layak untuk mengklasifikasikan dan memprediksi kedua kelas [10].

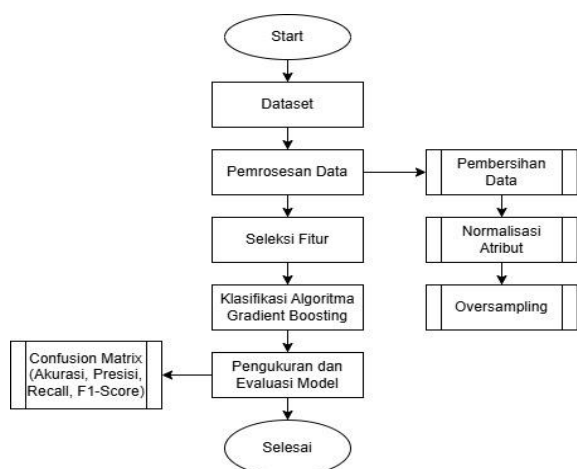
Algoritma *Gradient Boosting* dan *Recursive Feature Elimination* juga pernah digunakan oleh theerthagiri pada tahun 2022 yang berjudul "Predictive analysis of cardiovascular disease using gradient boosting based learning and recursive feature elimination technique". Mengintegrasikan algoritme pembelajaran mesin untuk prediksi dan diagnosis penyakit medis sangat diperlukan untuk pertumbuhan yang menjanjikan di bidang kesehatan. Pekerjaan ini memilih fitur paling signifikan dari kumpulan data penyakit *kardiovaskular* menggunakan pendekatan peningkatan gradien berbasis eliminasi fitur rekursif. Dari 12 karakteristik tersebut, algoritma RFE-GB memilih tiga fitur dengan jumlah ideal, seperti tekanan darah, kolesterol, dan aktivitas fisik. Teknik *unsambel* peningkatan gradien telah dikembangkan untuk meramalkan kasus penyakit kardiovaskular menggunakan tiga karakteristik ini. Metode RFE-GB yang diusulkan telah diuji menggunakan berbagai pengukuran, dan hasilnya telah

dibandingkan dengan algoritma pembelajaran mesin lainnya. Dibandingkan dengan algoritma LR, RF, MLP, SVM, dan NB, metode RFE-GB yang dihadirkan meningkatkan akurasi sebesar 14,12 persen hingga 30,12 persen. Selain itu, ia memiliki konsistensi yang lebih besar (skor Kappa) yaitu 0,56 dan tingkat kesalahan *MSE* yang lebih rendah yaitu 0,1924 untuk memprediksi penyakit jantung secara akurat. Dengan skor *AUROC* sebesar 85 persen, algoritma RFE-GB yang dikembangkan berhasil memperkirakan 88 persen benar positif dan 85 persen benar [11].

Dari uraian latar belakang di atas, Penelitian ini bertujuan untuk melakukan penerapan algoritma seleksi fitur *recursive feature elimination* untuk mengurangi jumlah fitur yang ada dalam data, sehingga dapat membantu dan mempermudah algoritma *gradient boosting* dalam melakukan klasifikasi serangan DDoS dengan menggunakan fitur-fitur yang penting atau berkontribusi.

2. METODOLOGI PENELITIAN

Metodologi Penelitian yang digunakan pada Penelitian ini dimulai dengan menggunakan dataset yang publik serangan DDoS yang diambil dari data mendeley. selanjutnya dilakukan pemrosesan pada dataset yang telah diambil. Kemudian tahap selanjutnya yaitu dilakukan seleksi fitur dengan menggunakan algoritma *recursive feature elimination* untuk dipilih fitur-fitur yang penting. Setelah seleksi fitur selesai selanjutnya dilakukan klasifikasi menggunakan algoritma *gradient boosting* untuk mengevaluasi kinerja dari model tersebut dengan menggunakan *confusion matrix* untuk mengetahui nilai akurasi, *precision*, *recall* dan *F1-score*. Alur metode Penelitian dapat dilihat pada gambar berikut:



Gambar 2. Alur Penelitian

3.1 Dataset

Pada penelitian ini menggunakan dataset serangan DDoS tahun 2019. Dataset ini memiliki jumlah data yaitu 431.371 data, memiliki 80 jumlah fitur dan 2 *Class* yaitu *attack* dan *benign*. Dataset ini diambil dari situs <https://data.mendeley.com/dataset/ssnc74xm6r/1>.

3.2 Pemrosesan Data

Pemrosesan data adalah langkah penting yang bertujuan untuk mengubah data ke dalam format yang dapat dipahami oleh sistem. Sebagian besar data yang akan dianalisis biasanya tidak lengkap dan tidak konsisten. Oleh karena itu, praproses data diperlukan untuk meningkatkan kualitas data, yang pada gilirannya akan membantu meningkatkan akurasi dan efisiensi hasil analisis. Pemrosesan data sangat krusial dalam analisis lalu lintas jaringan karena pola-pola data tersebut memiliki jenis dan dimensi yang bervariasi. Berikut tahap pemrosesan data yang dilakukan pada Penelitian ini:

a. Pembersihan data

Pembersihan data adalah tahap pemrosesan pertama. Karena data seringkali hilang atau rusak, pembersihan data harus dilakukan. Pahami sumber data yang hilang atau rusak. Kesalahan dapat terjadi selama transfer data, kesalahan sistem, atau masalah lain. Hal ini akan mengganggu proses analisis jika ada pada data yang digunakan. Oleh karena itu, data harus dihilangkan dengan menggantinya dengan nilai konstan, nilai tengah, atau nilai rata-rata dari data yang sekolom dengannya. Pada proses ini dilakukan penghapusan 2 fitur yaitu fitur yang isinya hanya berupa nomer urut dan fitur yang redundan atau fitur yang mungkin mengandung informasi yang berlebihan atau tidak diperlukan. Pada dataset ini, terdapat dua kolom yang berpotensi redundan, yaitu kolom Label dan Class. Kedua kolom ini tampaknya memiliki informasi yang serupa tentang kategori atau kelas dari data. Dengan menghapus fitur yang redundan, dataset menjadi lebih bersih dan siap untuk digunakan dalam proses pelatihan model. Penghapusan fitur yang tidak diperlukan membantu meningkatkan efisiensi dan performa model. Pada klasifikasi serangan ddos ini peneliti memilih fitur class yang digunakan sebagai target dari klasifikasi.

b. Normalisasi atribut *categorical*

Atribut kategorik pada dataset CIC-DDoS2019 terdapat kolom kelas. Pada kolom tersebut terdapat dua kategori yang ada yaitu kelas jinak dan kelas serangan. Dengan dilakukannya normalisasi, yang awalnya merupakan kelas serangan akan berubah menjadi angka 0.

c. Oversampling

Oversampling adalah teknik yang dipakai untuk menambah jumlah kelas kelas minoritas sehingga mencapai kelas mayoritas [12]. Dalam tahap ini, penulis akan menambahkan lebih banyak jumlah sampel dari kelas minoritas untuk mencapai keseimbangan antara kelas mayoritas dan minoritas. Dengan melakukan oversampling, kami bertujuan untuk mengurangi bias yang mungkin terjadi selama proses pelatihan model, sehingga model dapat lebih akurat dalam mengklasifikasikan data dari semua kelas. Teknik oversampling yang digunakan pada Penelitian ini adalah SMOTE (*Synthetic Minority Over-sampling Technique*).

SMOTE adalah metode *oversampling* di mana sampel sintetis dari kelas minoritas dihasilkan dengan menginterpolasi antara sampel-sampel yang ada [13]. Proses ini dilakukan dengan memilih dua atau lebih sampel minoritas yang berada dalam jarak tertentu satu sama lain, kemudian membuat sampel baru yang berada di antara titik-titik tersebut dalam ruang fitur. Dengan cara ini, SMOTE membantu menciptakan distribusi data yang lebih seimbang tanpa menduplikasi data asli, sehingga meningkatkan kemampuan model untuk belajar dari kelas minoritas. Berikut diagram tabel class sebelum dan sesudah smote:

TABEL 1. KELAS SEBELUM DAN SESUDAH SMOTE

Class	
0	1
333.540	97.831



Class	
0	1
266.849	266.849

Tabel 1 menunjukkan distribusi kelas dalam dataset sebelum dan sesudah menerapkan SMOTE. Sebelum SMOTE, terdapat 333.540 sampel untuk kelas 0 dan 97.831 sampel untuk kelas 1, menunjukkan ketidakseimbangan kelas. Setelah SMOTE diterapkan, distribusi kelas menjadi seimbang dengan masing-masing kelas, 0 dan 1, memiliki 266.849 sampel. SMOTE digunakan untuk menyeimbangkan jumlah sampel di setiap kelas dengan menghasilkan sampel sintetis untuk kelas minoritas.

3.3 Seleksi Fitur

Seleksi fitur adalah proses untuk menyeleksi fitur-fitur penting yang memiliki dampak paling penting dalam proses klasifikasi [14]. Selain itu, seleksi fitur ini juga bertujuan mengurangi banyak dimensi data dan meningkatkan kinerja model dengan hanya memakai fitur-fitur penting dalam dataset. Pada Penelitian ini Algoritma *reqursive feature Elimination* dipilih untuk seleksi fitur.

Reqursive feature elimination merupakan metode yang sering digunakan untuk mengatasi dataset yang mempunyai banyak fitur. Pada dasarnya algoritma ini bekerja dengan cara memilih kombinasi fitur yang terbaik sekaligus melakukan eliminasi berdasarkan peringkat fitur untuk dinilai kepentingan dalam proses klasifikasi yang dilakukan secara rekursif dengan jumlah fitur yang diperlukan.

3.4 Gradient Boosting

Gradient Boosting adalah teknik pembelajaran mesin yang biasa digunakan untuk regresi dan klasifikasi yang bertujuan untuk menghasilkan nilai model pada algoritma. *Gradient Boosting* digunakan Setelah data selesai diseleksi fitur menggunakan *Reqursive Feature Elimination* [15]. Berikut cara kerja dari algoritma gradient boosting:

3.5 Evaluasi model

Evaluasi model ini bertujuan untuk mengukur nilai pada kinerja algoritma yang digunakan. Berikut beberapa pengukuran yang digunakan dalam Penelitian ini:

1. Confusion Matrix

Untuk menguji hasil klasifikasi pada sistem yang dibangun, diperlukan suatu metode perhitungan untuk evaluasi kinerja. Tahap yang di perlukan untuk evaluasi kinerja yaitu dengan menghitung nilai presisi, *recall*,

dan nilai-nilai ukuran f1. Akurasi adalah cara sistem mengklasifikasikan benar dengan data benar dan salah, dan ukuran-f1 adalah untuk menilai kinerja keseluruhan sistem dengan menghitung nilai akurasi dan *recall*. Perhitungannya dapat dilihat pada matriks konfusi [3]. Berikut tabel confusion matrix yang digunakan:

TABEL II. CONFUSION MATRIX

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

Pada tabel 2 dijelaskan TP (*True Positive*) merupakan prediksi positif dan nilai sebenarnya positif, TN (*True Negative*) merupakan prediksi negatif dan nilai sebenarnya *negative*, FP (*False Positive*) merupakan prediksi positif dan nilai sebenarnya negatif dan FN (*False Negative*) merupakan prediksi negatif dan nilai sebenarnya positif.

2. Accuracy

Accuracy bisa diartikan dengan rasio jumlah total prediksi *true* dan *false* dari semua data. Untuk menghitung akurasi bisa menggunakan rumus sebagai Berikut:

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn} \quad (1)$$

3. Precision

Precision bisa didefinisikan yaitu rasio jumlah *true* positif terhadap total jumlah *true* positif dan *false* positif. Berikut rumus yang digunakan untuk mengukur nilai *precision*:

$$Precision = \frac{tp}{tp + fp} \quad (2)$$

4. Recall

Recall merupakan rasio jumlah *true* positif yang dibagi dengan jumlah *true* positif dan *false* negatif. Formula yang umum digunakan dalam menghitung *recall*:

$$Recall = \frac{tp}{tp + fn} \quad (3)$$

5. F1-score

F1 score adalah ukuran dari jumlah yang didapat dari *precision* dan *recall*, *F1 score* juga bisa untuk menambah keseimbangan pada kedua *precision* dan *recall*. *F1 score* bisa dihitung dengan rumus:

$$F1\ score = \frac{2 \times precision \times recall}{precision + recall} \quad (4)$$

3. HASIL DAN PEMBAHASAN

3.1 Hasil pengujian tanpa parameter

Hasil kinerja dari algoritma *Gradient Boosting* tanpa pengaturan parameter khusus. Dalam konfigurasi default ini, model mencapai akurasi sebesar 99,8%, presisi sebesar 99,4%, *recall* sebesar 99,9%, dan nilai F1 sebesar 99,7%. Hasil ini menunjukkan bahwa bahkan tanpa penyesuaian parameter, *Gradient Boosting* dapat memberikan performa yang sangat baik dengan tingkat akurasi yang hampir sempurna, presisi yang tinggi, *recall* yang sangat tinggi, dan nilai F1 yang hampir optimal.

3.2 Hasil pengujian menggunakan parameter *n_estimator*

Pada tahap ini dilakukan uji coba hasil kinerja dari algoritma *Gradient Boosting* dengan variasi pada parameter, yaitu *n_estimator*, *learning_rate*, dan *max_depth*. Parameter *n_estimator* bervariasi dari 100 hingga 400, sementara *learning_rate* dan *max_depth* tetap konstan pada 0.1 dan 3, masing-masing. Berikut tabel hasil pengujian menggunakan parameter *n_estimator*:

TABEL III. HASIL PARAMETER *N_ESTIMATOR*

Gradient Boosting				
<i>n_estimator</i>	acc	pr	rc	F1
100	99,8%	99,1%	100%	99,5%
200	99,9%	99,5%	100%	99,7%
300	99,9%	99,7%	100%	99,8%
400	99,9%	99,7%	99,9%	99,8%

Pada tabel 3 diatas menunjukkan hasil untuk model dengan 100 *estimator*, akurasi mencapai 99,8%, presisi sebesar 99,1%, *recall* 100%, dan

nilai F1 adalah 99,5%. Ketika jumlah *estimator* ditingkatkan menjadi 200, akurasi meningkat menjadi 99,9%, presisi naik menjadi 99,5%, *recall* tetap 100%, dan nilai F1 meningkat menjadi 99,7%. Pada percobaan dengan 300 *estimator*, akurasi tetap di 99,9%, presisi meningkat sedikit menjadi 99,7%, *recall* tetap 100%, dan nilai F1 meningkat menjadi 99,8%. Sedangkan dengan 400 *estimator*, akurasi tetap 99,9%, presisi dan *recall* masing-masing adalah 99,7% dan 99,9%, dan nilai F1 tetap di 99,8%.

3.3 Hasil Pengujian menggunakan parameter *learning_rate*

Pada tahap ini dilakukan uji coba hasil kinerja dari algoritma *gradient boosting* dengan variasi pada parameter *learning_rate* sementara parameter *n_estimator* dan *max_depth* tetap konstan masing-masing pada 100 dan 3. Berikut tabel hasil uji coba yang telah dilakukan:

TABEL IV. HASIL PARAMETER *LEARNING_RATE*

<i>Gradient Boosting</i>				
<i>learning_rate</i>	acc	pr	rc	F1
0.0001	96%	85,2%	99,6%	91,8%
0.001	96,1%	85,5%	99,6%	92%
0.01	97,2%	89%	99,8%	94,1%
0.1	99,8%	99,1%	100%	99,5%

Tabel 4 di atas menunjukkan kinerja model *gradient boosting* dengan berbagai nilai *learning_rate*. Pada *learning_rate* 0.0001, model mencapai akurasi 96%, presisi 85,2%, *recall* 99,6%, dan F1 score 91,8%. Dengan *learning_rate* 0.001, akurasi meningkat menjadi 96,1%, presisi 85,5%, *recall* 99,6%, dan F1 92%. Pada *learning_rate* 0.01, akurasi naik ke 97,2%, presisi 89%, *recall* 99,8%, dan F1 94,1%. *Learning_rate* 0.1 memberikan kinerja terbaik dengan akurasi 99,8%, presisi 99,1%, *recall* 100%, dan F1 99,5%.

3.4 Hasil pengujian menggunakan parameter *max_depth*

Pada tahap ini dilakukan uji coba hasil kinerja dari algoritma *gradient boosting* dengan variasi pada parameter *max_depth* sementara parameter *n_estimator* dan *learning_rate* tetap konstan masing-masing pada 100 dan 0.1. berikut

adalah tabel dari uji coba dengan menggunakan parameter *max_depth*:

TABEL V. HASIL PARAMETER *MAX_DEPTH*

<i>Gradient Boosting</i>				
<i>max_depth</i>	acc	pr	rc	F1
3	99,8%	99,1%	100%	99,5%
5	99,9%	99,6%	100%	99,8%
7	99,9%	99,8%	100%	99,9%
9	99,9%	99,8%	99,9%	99,9%

Pada tabel 5 menunjukkan percobaan dengan *max_depth* sebesar 3, model mencapai akurasi 99,8%, presisi 99,1%, *recall* 100%, dan nilai F1 99,5%. Ketika *max_depth* ditingkatkan menjadi 5, akurasi meningkat menjadi 99,9%, presisi naik menjadi 99,6%, *recall* tetap 100%, dan nilai F1 juga meningkat menjadi 99,8%. Dengan *max_depth* sebesar 7, akurasi tetap 99,9%, presisi meningkat lebih lanjut menjadi 99,8%, *recall* tetap 100%, dan nilai F1 mencapai 99,9%. Pada kedalaman maksimal pohon yang tertinggi yaitu 9, akurasi tetap di 99,9%, presisi tetap 99,8%, *recall* sedikit menurun menjadi 99,9%, dan nilai F1 tetap di 99,9%

3.5 Hasil Seleksi fitur

Pada tahap ini dilakukan Seleksi fitur dengan menggunakan algoritma *recursive feature elimination* untuk memilih fitur terbaik dari dataset yang ada. Prosesnya dimulai dengan menentukan model klasifikasi yang akan digunakan untuk mengevaluasi fitur, pada Penelitian ini digunakan algoritma *gradient boosting* dengan nilai parameter terbaik dari setiap parameter yang telah diuji coba sebelumnya sebagai model untuk klasifikasi. Selanjutnya, model dilatih menggunakan semua fitur yang tersedia, dan pentingnya setiap fitur dihitung. Fitur yang memiliki peringkat terendah (paling tidak penting) kemudian dihapus. Proses ini diulang secara iteratif, di mana model dilatih ulang dengan fitur yang tersisa, fitur dievaluasi kembali, dan fitur dengan peringkat terendah dihapus, hingga mencapai jumlah fitur yang diinginkan. Setelah iterasi selesai, subset fitur yang tersisa adalah fitur terbaik menurut RFE, yang diharapkan dapat meningkatkan akurasi dan efisiensi model prediktif. Berikut adalah tabel hasil seleksi fitur menggunakan RFE

TABEL VI. HASIL SELEKSI FITUR

Recursive Feature Elimination	
Jumlah Fitur	Fitur Terpilih
10	'Protocol', 'Flow Duration', 'Fwd Packets/s', 'Bwd Packets/s', 'Packet Length Min', 'ACK Flag Count', 'URG Flag Count', 'Avg Fwd Segment Size', 'Init Fwd Win Bytes', 'Active Min'
15	'Protocol', 'Flow Duration', 'Flow IAT Mean', 'Flow IAT Min', 'Fwd IAT Min', 'Fwd Packets/s', 'Bwd Packets/s', 'Packet Length Min', 'Packet Length Max', 'Packet Length Variance', 'ACK Flag Count', 'URG Flag Count', 'Avg Packet Size', 'Init Fwd Win Bytes', 'Idle Std'
20	'Protocol', 'Flow Duration', 'Fwd Packets Length Total', 'Fwd Packet Length Max', 'Fwd Packet Length Mean', 'Flow IAT Mean', 'Flow IAT Std', 'Flow IAT Min', 'Fwd IAT Min', 'Fwd Packets/s', 'Bwd Packets/s', 'Packet Length Min', 'Packet Length Max', 'Packet Length Variance', 'ACK Flag Count', 'URG Flag Count', 'Avg Packet Size', 'Init Fwd Win Bytes', 'Fwd Seg Size Min', 'Idle Std'

Tabel 6 di atas menunjukkan bagaimana *recursive feature elimination* bekerja dengan memeringkat dan memilih fitur yang dianggap paling relevan pada setiap iterasi. Fitur yang dipilih diharapkan akan meningkatkan kinerja model prediksi. Informasi tentang apakah fitur tersebut terpilih (*selected=True/False*) dan peringkatnya (*rank=n*) membantu dalam memahami kontribusi setiap fitur dalam model yang dibangun.

3.6 Hasil perbandingan akurasi dan waktu komputasi Setelah dan sebelum seleksi fitur

Pada tahap ini menunjukkan hasil perbandingan akurasi kinerja model *gradient boosting* dengan parameter terbaik yang telah diuji coba sebelumnya, sebelum dan sesudah menggunakan *recursive feature elimination* untuk memilih fitur terbaik. Berikut tabel perbandingan akurasi dan waktu komputasi:

TABEL VII. HASIL PERBANDINGAN

Model	Parameter	Waktu komputasi	Nama Fitur	Hasil akurasi
GB	<i>n_estimator</i> = 300 <i>learning_rate</i> = 0.1 <i>max_depth</i> = 7	5463.24 s	Semua fitur	Acc = 99,9% Pr = 99,8% Rc = 99,9% F1 = 99,9%
GB+RFE	<i>n_estimator</i> = 300 <i>learning_rate</i> = 0.1 <i>max_depth</i> = 7 <i>n_features_to_select</i> = 10	1023.19 s	'Protocol', 'Flow Duration', 'Fwd Packets/s', 'Bwd Packets/s', 'Packet Length Min', 'ACK Flag Count', 'URG Flag Count', 'Avg Fwd Segment Size', 'Init Fwd Win Bytes', 'Active Min'	Acc = 99,9% Pr = 99,8% Rc = 99,9% F1 = 99,9%
GB+RFE	<i>n_estimator</i> = 300 <i>learning_rate</i> = 0.1 <i>max_depth</i> = 7 <i>n_features_to_select</i> = 15	1530.92 s	'Protocol', 'Flow Duration', 'Flow IAT Mean', 'Flow IAT Min', 'Fwd IAT Min', 'Fwd Packets/s', 'Bwd Packets/s', 'Packet Length Min', 'Packet Length Max', 'Packet Length Variance', 'ACK Flag Count', 'URG Flag Count', 'Avg	Acc = 99,9% Pr = 99,8% Rc = 99,9% F1 = 99,9%

Model	Parameter	Waktu komputasi	Nama Fitur	Hasil akurasi
			<i>Packet Size', 'Init Fwd Win Bytes', 'Idle Std'</i>	
GB+RFE	<i>n_estimator</i> = 300 <i>learning_rate</i> = 0.1 <i>max_depth</i> = 7 <i>n_features_to_select</i> = 20	2025.35 s	<i>'Protocol', 'Flow Duration', 'Fwd Packets Length Total', 'Fwd Packet Length Max', 'Fwd Packet Length Mean', 'Flow IAT Mean', 'Flow IAT Std', 'Flow IAT Min', 'Fwd IAT Min', 'Fwd Packets/s', 'Bwd Packets/s', 'Packet Length Min', 'Packet Length Max', 'Packet Length Variance', 'ACK Flag Count', 'URG Flag Count', 'Avg Packet Size', 'Init Fwd Win Bytes', 'Fwd Seg Size Min', 'Idle Std'</i>	Acc = 99,9% Pr = 99,8% Rc = 99,9% F1 = 99,9%

Pada tabel 7 ini menunjukkan Dengan menggunakan *recursive feature elimination* dan memilih fitur terbaik, model *Gradient Boosting* berhasil mengurangi waktu komputasi secara signifikan dari 5463.24 detik menjadi 1023.19 detik tanpa mengorbankan kinerja model. Akurasi, presisi, *recall*, dan skor F1 tetap sama tinggi 99.9%, 99.8%, 99.9%, dan 99.9% masing-masing. Ini menunjukkan bahwa menggunakan *recursive feature elimination* untuk mengurangi jumlah fitur dapat meningkatkan efisiensi model tanpa mengorbankan kinerjanya.

4. Kesimpulan dan Saran

Hasil pengujian menunjukkan bahwa *gradient boosting* tanpa pengaturan parameter khusus sudah memberikan kinerja yang sangat baik dengan akurasi 99,8%, presisi 99,4%, *recall* 99,9%, dan nilai F1 99,7%. Setelah dilakukan pengujian dengan 3 parameter berikut diambil 3 hasil terbaik dari setiap parameter yaitu *n_estimator* 300, *learning_rate* 0.1, *max_depth* 7. Untuk hasil pengujian menggunakan parameter terbaik adalah 99,9%, 99,8%, 99,9%, dan 99,9% Ketika dilakukan penerapan *recursive feature elimination* dengan parameter terbaik yaitu (*n_estimator* 300, *learning_rate* 0.1, *max_depth* 7), jumlah fitur yang dipilih tidak mempengaruhi akurasi, presisi, *recall*, dan nilai F1, yang tetap pada 99,9%, 99,8%, 99,9%, dan 99,9%. Namun,

jumlah fitur yang lebih sedikit mengurangi waktu komputasi secara signifikan. Pemilihan 10 fitur terbaik dengan *recursive feature elimination* menurunkan waktu komputasi dari 5463.24 detik menjadi 1023.19 detik tanpa mengorbankan kinerja model. Ini menunjukkan bahwa penggunaan *recursive feature elimination* dapat meningkatkan efisiensi model secara signifikan sambil mempertahankan kinerja yang tinggi.

Dari kesimpulan diatas, mungkin saran berikut dapat digunakan pada penelitian kedepannya. Dari 3 parameter yang digunakan dalam penelitian ini sudah memberikan hasil yang sangat baik, eksplorasi lebih lanjut terhadap berbagai kombinasi parameter lain dapat dilakukan untuk meningkatkan efisiensi atau performa model. Selain itu, disarankan untuk menguji pendekatan yang sama pada dataset yang lebih besar atau berbeda guna memastikan generalisasi hasil penelitian dan memahami bagaimana model serta teknik seleksi fitur bekerja di berbagai kondisi. Selain seleksi fitur *Recursive Feature Elimination* eksplorasi metode seleksi fitur lain, seperti L1 regularization atau metode berbasis mutual information, juga dapat dilakukan untuk membandingkan efektivitasnya dengan metode yang telah digunakan.

Daftar Pustaka:

- [1] I. M. W. Bhaskara, I. P. G. H. Suputra, I. M. Widiartha, I. G. A. G. A. Kadyanan, I. G. N. A. C. Putra, and I. B. G. Dwidasmara,

- "Klasifikasi Serangan Distributed Denial of Service (DDoS) Menggunakan Random Forest Dengan CFS," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 11, no. 2, p. 215, 2022.
- [2] E. O. Nasution and A. Basuki, "Implementasi Algoritme C5.0 Untuk Klasifikasi Serangan DDoS," vol. 5, no. 1, pp. 389–395, 2021.
- [3] M. Zidane, "Klasifikasi Serangan Distributed Denial-of-Service (DDoS) menggunakan Metode Data Mining Naive Bayes," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 1, pp. 172–180, 2022.
- [4] A. Fauzi, E. Utami, and A. D. Hartanto, "DDoS Penerapan Random Forest dan Adaboost untuk Klasifikasi Serangan DDoS," *J. Educ.*, vol. 5, no. 3, pp. 7925–7937, 2023.
- [5] A. Prasetyo, L. Affandi, and D. Arpandi, "Implementasi Metode Naive Bayes Untuk Intrusion Detection System (Ids)," *J. Inform. Polinema*, vol. 4, no. 4, p. 280, 2018.
- [6] A. Afifuddin and L. Hakim, "Deteksi Penyakit Diabetes Mellitus Menggunakan Algoritma Decision Tree Model Arsitektur C4.5," *J. Krisnadana*, vol. 3, no. 1, pp. 25–33, 2023.
- [7] LamanIT, "Gradient Boosting: Fungsi, Tahapan dan Kelebihan," 2023.
- [8] M. K. Harto and A. Basuki, "Deteksi Serangan DDoS Pada Jaringan Berbasis SDN Dengan Klasifikasi Random Forest," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 4, pp. 1329–1333, 2021.
- [9] Sri Diantika, Hiya Nalatissifa, Riki Supriyadi, Nurlaelatul Maulidah, and Ahmad Fauzi, "Implementation of Multi-Class Gradient Boosting To Classify Animal Species in Zoos," *Antivirus J. Ilm. Tek. Inform.*, vol. 17, no. 1, pp. 33–40, 2023.
- [10] S. E. Suryana, B. Warsito, and S. Suparti, "Penerapan Gradient Boosting Dengan Hyperopt Untuk Memprediksi Keberhasilan Telemarketing Bank," *J. Gaussian*, vol. 10, no. 4, pp. 617–623, 2021.
- [11] P. Theerthagiri, "Predictive analysis of cardiovascular disease using gradient boosting based learning and recursive feature elimination technique," *Intell. Syst. with Appl.*, vol. 16, no. August, p. 200121, 2022.
- [12] L. Qadrini, H. Hikmah, and M. Megasari, "Oversampling, Undersampling, Smote SVM dan Random Forest pada Klasifikasi Penerima Bidikmisi Sejava Timur Tahun 2017," *J. Comput. Syst. Informatics*, vol. 3, no. 4, pp. 386–391, 2022.
- [13] E. Sutoyo and M. A. Fadlurrahman, "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network," *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 3, p. 379, 2020.
- [14] A. C. Kurniawan and A. Salam, "Seleksi Fitur Information Gain untuk Optimasi Klasifikasi Penyakit Tuberkulosis," *J. Media Inform. Budidarma*, vol. 8, no. 1, p. 70, 2024.
- [15] P. Theerthagiri, "Predictive analysis of cardiovascular disease using gradient boosting based learning and recursive feature elimination technique," *Intell. Syst. with Appl.*, vol. 16, no. September, p. 200121, 2022.