Optimalisasi Deteksi Intrusi Menggunakan Feature Selection Lasso Regression dan Random Forest dengan Hyperparameter Tuning Berbasis Bayesian Optimization

Trenady Bagas Pramudya
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
trenadybagas@student.telkomuniversity.ac.id

Aji Gautama Putrada
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
ajigps.telkomuniversity.ac.id

Utami Kusuma Dewi
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
utamikusdew@telkomuniversity.ac.id

ABSTRAK

Perkembangan teknologi informasi yang cepat telah meningkatkan risiko keamanan siber. Intrusion detection system (IDS) berbasis machine learning adalah salah satu metode yang paling umum digunakan untuk mendeteksi aktivitas intrusi dalam jaringan. Namun, tantangan muncul ketika berhadapan dengan dataset besar dan berdimensi tinggi yang dapat mengganggu kinerja dan performansi model. Penelitian ini bertujuan untuk mengoptimalkan kinerja algoritma Random Forest dalam mendeteksi intrusi jaringan melalui penerapan seleksi fitur berbasis Lasso Regression untuk Feature Selection dan Bayesian Optimization sebagai optimasi hyperparameter. Metode penelitian dimulai melalui tahap pre-processing, data splitting atau pembagian data menjadi data latih, data uji dan data validasi. Selanjutnya, dilakukan pengujian terhadap tiga skenario model, yaitu model Random Forest sebagai model baseline, Random Forest dengan seleksi fitur Lasso Regression, serta model Random Forest dengan seleksi fitur Lasso Reggression yang diptimasi menggunakan Bayesian Optimization. Hasil penelitian menunjukkan bahwa penerapan seleksi fitur menggunakan Lasso Regression mampu mengurangi kompleksitas data dengan tetap mempertahankan performa model, sementara Bayseian Optimization memberikan peningkatan akurasi dibandingkan dengan model baseline dan model tanpa optimasi. Model yang diusulkan menunjukkan performa yang stabil dan efisien dalam mendeteksi berbagai jenis serangan pada jaringan. Berdasarkan hasil tersebut, dapat disimpulkan bahwa model kombinasi Lasso Regression dan Bayesian Optimization efektif dalam meningkatkan kinerja dan efisiensi algoritma Random Forest untuk sistem deteksi intrusi jaringan.

Kata Kunci: Deteksi Intrusi, Random Forest, Lasso Regression, Bayesian Optimization, Seleksi Fitur, Hyperparameter Tuning

I. PENDAHULUAN

Perkembangan teknologi informasi, khususnya dalam era modern ini telah memfasilitasi integrasi perangkat dan pertukaran data sehingga aktivitas tersebut menjadi lebih mudah. Perkembangan ini juga diiringi dengan peningkatan risiko terhadap keamanan jaringan. Ketergantungan yang semakin besar terhadap teknologi digital membuat serangan siber menjadi masalah signifikan yang membutuhkan perhatian serius [1]. Oleh karena itu, perlindungan terhadap jaringan menjadi suatu kebutuhan mendesak yang memerlukan pendekatan efektif.

Intrusion Detection System (IDS) dikembangkan sebagai solusi untuk mendeteksi dan mencegah serangan siber dengan cara mengidentifikasi aktivitas mencurigakan dalam jaringan komputer [2]. Dalam pengembangannya, IDS umumnya dilatih dan dievaluasi menggunakan dataset lalu lintas jaringan yang merepresentasikan berbagai jenis serangan dan aktivitas normal [3], [4]. Namun demikian, dataset IDS memiliki karakteristik khusus berupa dimensi fitur yang tinggi serta distribusi kelas yang tidak seimbang. Kondisi tersebut berpotensi menimbulkan permasalahan seperti overfitting dan underfitting, yang pada akhirnya dapat menurunkan performa sistem deteksi intrusi. Oleh sebab itu, tahapan pre-processing data dan pemilihan metode pemodelan yang tepat menjadi aspek penting dalam pengembangan IDS.

Berbagai penelitian sebelumnya menunjukkan bahwa algoritma *ensemble* seperti *random forest* banyak digunakan dalam sistem deteksi intrusi karena ketahanannya terhadap noise serta kemampuannya dalam menangani dataset berukuran besar dan tidak seimbang [5]. Selain itu, performa random forest diketahui dapat ditingkatkan secara signifikan melalui proses *hyperparameter tuning* [6], [7]. Pendekatan berbasis *bayesian optimization* tercatat lebih efisien dalam mengeksplorasi ruang parameter dibandingkan metode pencarian konvensional, karena mampu menemukan konfigurasi optimal dengan jumlah evaluasi yang lebih sedikit [4], [8]. Di sisi lain, teknik seleksi fitur berbasis regularisasi seperti *lasso regression* juga terbukti efektif

dalam mengurangi kompleksitas data berdimensi tinggi dengan tetap mempertahankan fitur-fitur yang relevan [9], [10]. Kombinasi ini mampu meningkatkan performa sistem deteksi intrusi secara signifikan [11], serta relevan untuk lingkungan jaringan yang dinamis dan kompleks [12].

Berdasarkan permasalahan dan temuan penelitian terdahulu tersebut, penelitian ini mengusulkan pendekatan yang mengintegrasikan *lasso regression* sebagai metode seleksi fitur, *random forest* sebagai algoritma klasifikasi utama, serta *bayesian optimization* sebagai metode optimasi *hyperparameter*. Pendekatan ini diharapkan mampu mengurangi kompleksitas data berdimensi tinggi, meningkatkan generalisasi model, serta menghasilkan sistem deteksi intrusi yang akurat, efisien, dan responsif terhadap karakteristik dataset IDS yang kompleks.

II. KAJIAN TEORI

A. Deteksi Intrusi

Intrusion Detection System (IDS) merupakan komponen penting dalam sistem keamanan jaringan yang mengidentifikasi perilaku mencurigakan atau ketidaknormalan yang menandakan adanya upaya intrusi atau serangan. Sistem deteksi intrusi secara umum diklasifikasikan menjadi dua metode utama: berbasis tanda (signature-based) dan berbasis anomali (anomaly-based). Metode berbasis tanda mengidentifikasi serangan menggunakan pola yang telah ditetapkan, sedangkan metode berbasis anomali mengenali perilaku yang menyimpang dari pola normal [1]. Seiring dengan semakin kompleksnya jaringan, terutama dalam sistem IoT, IDS harus dirancang secara efektif untuk beroperasi dalam kondisi sumber daya terbatas. Salah satu strategi yang berkembang adalah penggunaan teknik machine learning untuk mengembangkan sistem deteksi intrusi yang adaptif dan ringan [3]. Solusi deteksi intrusi berbasis machine learning memungkinkan sistem untuk mengidentifikasi ancaman baru menggunakan pola data.

B. Random Forest

Random Forest adalah teknik algoritma ensemble yang berbasis pada decision tree dan digunakan secara luas dalam klasifikasi dan regresi. Pendekatan ini menawarkan keunggulan dalam mengelola dataset yang besar dan tidak seimbang, serta menunjukkan ketahanan terhadap overfitting dalam kasus deteksi intrusi [2]. Cara kerja random forest adalah dengan membuat beberapa pohon keputusan dari subset data dan menghasilkan prediksi melalui pemungutan suara mayoritas.

Proses pembentukan pohon keputusan pada *random forest* sama dengan proses pembentukan pada metode Classification and Regrression Tree (CART) hanya saja tidak dilakukan pemangkasan pohon atau *prunning*. Pada tahap awal, algoritma *random forest* akan menghitung *Gini Impurity* untuk memilih fitur. Nilai indeks *gini* dihitung dengan menggunakan perhitungan sebagai berikut.

$$Gini = 1 - \sum_{i=1}^{C} p_i^2$$

Dimana p_i mewakili probabilitas kemunculan suatu kelas tertentu dan c merupakan jumlah kelas yang terdapat pada node.

Penelitian oleh [1] menunjukkan bahwa *random forest* dapat *outperform* atau mengungguli algoritma lain, termasuk *naïve bayes* dan *svm*, dalam tugas klasifikasi, terutama pada data yang kompleks. Pendekatan ini juga memberikan fleksibilitas yang signifikan untuk diintegrasikan dengan metode lain, termasuk *feature selection* dan *hyperparameter tuning.*

C. Lasso Regression

Lasso Regression atau Least Absolute Shrinkage and Selection Operator adalah teknik regresi linier yang menggunakan penalti L1 untuk memfasilitasi seleksi fitur secara otomatis. Metode ini bekerja dengan mengurangi koefisien regresi menjadi nol, sehingga hanya aspek-aspek yang paling relevan yang dipertahankan dalam model [3]. Secara matematis, lasso regression dinyatakan melalui persamaan sebagai berikut.

$$\hat{\beta} = argmin \sum_{i=1}^{n} \left(\mathrm{y_i} - \beta_0 - \Sigma_{\mathrm{j=1}}^{\mathrm{k}} = \beta_{\mathrm{j}} \mathrm{x_{ij}}\right)^2$$

Dimana y_i merupakan nilai variabel ke-i dan β_j merupakan koefisien regresi variabel ke-j. Dengan menambahkan penalti koefisien L1, maka didapat:

$$\hat{\beta} = argmin \sum_{i=1}^{n} \left(\mathrm{y_i} - \beta_0 - \Sigma_{\mathrm{j=1}}^{\mathrm{k}} = \beta_{\mathrm{j}} \mathrm{x_{ij}}\right)^2 + \lambda \sum_{j=1}^{P} |\beta_j|$$

Dengan λ (lambda) sebagai regularisasi yang memberikan penalti terhadap nilai absolut koefisien regresi, sehingga menekan variabel menjadi nol. Sementara |β_j | merupakan L1-norm yang merepresentasikan nilai absolut dari koefisien regresi [4].

Lasso Regression dalam konteks deteksi intrusi, bertujuan untuk mengurangi kompleksitas data berdimensi tinggi dan meningkatkan efisiensi model [5]. Menurut [6], menggunakan pendekatan reduksi dimensi dan mitigasi multikolinearitas dapat dilakukan untuk mempercepat pelatihan model dalam sistem deteksi intrusi berbasis IoT. Teknik seperti Lasso Regression efektif dalam menciptakan sistem efisien, ringan dan tetap dapat mempertahankan akurasi.

D. Bayesian Optimization

Bayesian Optimization adalah metode optimisasi berbasis model probabilistik yang unggul dalam mengelola data kompleks. Strategi ini sering digunakan dalam pembelajaran machine learning untuk mengidentifikasi kombinasi hyperparameter yang ideal dengan memaksimalkan kinerja model, terutama dalam algoritma yang memiliki banyak parameter dan durasi pelatihan yang panjang.

Dalam studi sistematisnya [8] menyoroti pentingnya Bayesian Optimization dalam deteksi intrusi berbasis IoT, khususnya dalam pengelolaan kompleksitas model, akurasi, dan efisiensi waktu pelatihan. Metode ini dianggap mampu mencapai keseimbangan optimal antara persyaratan sistem minimal dan akurasi tinggi, menjadikannya cocok untuk lingkungan dengan sumber daya terbatas seperti perangkat IoT.

Bayesian Optimization juga dapat mencegah terjadinya *overfitting* yang disebabkan oleh set parameter yang tidak memadai. Hal ini secara signifikan mempercepat proses penyesuaian, yang sebelumnya memerlukan waktu dan sumber daya komputasi yang signifikan menjadi lebih efisien. Akibatnya, Bayesian Optimization tidak hanya

efisien tetapi juga dapat secara signifikan meningkatkan potensi kinerja model.

E. Feature Selection

Feature Selection merupakan prosedur penting dan paling awal dalam *pre-processing* data, terutama saat menangani dataset berdimensi tinggi. Feature Selection bertujuan untuk mengidentifikasi subset fitur yang paling informatif dan relevan, serta mampu meningkatkan generalisasi model. Dengan menghilangkan fitur yang tidak relevan, sistem dapat mencegah terjadinya overfitting, mempercepat training, dan menghemat memori serta sumber daya komputasi.

Pemilihan fitur dapat dilakukan melalui *filter methods*, *wrapper methods*, dan *embedded methods*. Salah satu contoh dari embedded methods adalah Lasso Regression, teknik yang luas digunakan dan efektif untuk data berdimensi tinggi. Lasso menerapkan penalti L1 pada koefisien regresi, sehingga memaksa sebagian koefisien menjadi nol, menandakan bahwa fitur-fitur tersebut dianggap tidak relevan dengan tujuan [3], [5].

Bercontoh dari [6] menggunakan Feature Selection dalam pembuatan sistem *IDS* dalam mengatasi multikolinearitas dan menghasilkan model yang cepat dan efisien sumber daya dengan tetap mempertahankan akurasi. Penghapusan elemen yang tidak relevan terbukti meningkatkan efisiensi pelatihan, terutama pada perangkat IoT. Hal selaras juga ditunjukkan pada integrasi seleksi fitur dengan model klasifikasi yang secara signifikan meningkatkan akurasi pada data stunting berdimensi tinggi [9]. Dalam konteks IDS, pengurangan jumlah fitur dapat mempercepat durasi training data dan mengurangi dampak *noise* yang disebabkan oleh informasi yang tidak relevan atau berulang.

F. Hyperparameter Tuning

Hyperparameter Tuning adalah proses yang bertujuan dalam menyesuaikan nilai parameter eksternal dalam model Machine Learning guna meningkatkan kinerja daripada model. Berbeda dengan parameter internal model yang diperoleh dari data, hyperparameter, seperti jumlah estimator dalam Random Forest, kedalaman maksimum pohon keputusan, dan jumlah fitur yang digunakan dalam setiap pembagian atau split harus ditentukan sebelum proses pelatihan model dimulai.

Penelitian menunjukkan bahwa konfigurasi default dari banyak algoritma, seperti Random Forest, seringkali tidak menghasilkan kinerja model optimal. Langkah penyesuaian diperlukan untuk menyesuaikan hyperparameter dengan karakteristik unik yang terdapat pada dataset [10], [11].

Merunut pada studi [8] menegaskan bahwa penyesuaian *hyperparameter* merupakan salah satu dari empat komponen penting yang perlu dioptimalkan dalam sistem IDS modern, selain *feature selection*, pemilihan algoritma, dan keseimbangan data. Konfigurasi hyperparameter yang optimal dapat meningkatkan akurasi dan secara signifikan mempengaruhi efisiensi keseluruhan sistem, termasuk dalam waktu pelatihan model dan kemampuan deteksi

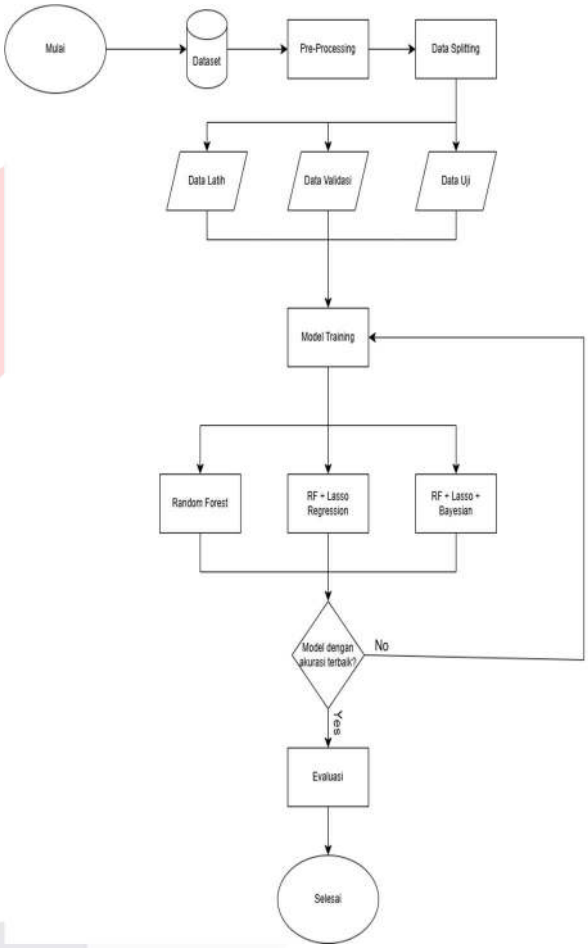
III. METODE

A. Perancangan Sistem

Desain perancangan sistem ini menggambarkan tahapan pembangunan model pendeteksian intrusi jaringan menggunakan Lasso Regression, Random Forest, dan Bayesian Optimization. Proses dimulai dari pembersihan dan prapemrosesan data, dilanjutkan dengan seleksi fitur menggunakan Lasso Regression. Model kemudian dilatih menggunakan Random Forest yang dioptimalkan melalui Bayesian Optimization, dan hasilnya diuji untuk mengukur akurasi serta performa dalam mengklasifikasikan trafik normal dan serangan.

B. Diagram Alur Model

Berikut merupakan diagram alur model yang menguraikan alur proses pengembangan dari evaluasi model deteksi intrusi yang digunakan dalam penelitian.



GAMBAR 1
Diagram Alur Model

Diagram alur model pada gambar 1 menggambarkan tahapan utama dalam proses pengembangan dan evaluasi model deteksi intrusi yang digunakan dalam penelitian ini. Tahapan yang dilakukan adalah sebagai berikut:

1. Dataset

Tahap awal penelitian dimulai dengan pengambilan dataset CSE-CIC-IDS2018 sebagai sumber data utama yang merepresentasikan lalu lintas jaringan normal serta berbagai jenis serangan intrusi. Dataset ini disusun dalam bentuk *flow-based network traffic* yang mana tiap *record* merepresentasikan satu *network flow*. Setiap *flow* dilengkapi dengan fitur statistik yang menggambarkan karakteristik temporal, ukuran paket dan lain sebagainya. Jenis serangan pada dataset ini meliputi diantaranya, Dos, Ddos, Brute Force, Infiltration, serta beberapa jenis serangan lain. Dataset ini memiliki total jumlah data sebelum melewati tahap *pre-processing* sebanyak 16.233.002 baris data dengan jumlah 84 kolom. Fitur-fitur tersebut dikemas dalam bentuk kategori.

2. Pre-Processing

Tahap *pre-processing* dilakukan untuk memastikan dataset berada dalam kondisi yang sesuai sebelum dilakukan dan digunakan dalam proses pemodelan. Dataset yang digunakan pada penelitian ini memiliki jumlah data sebesar 16.233.002 baris data dan 84 kolom, serta memiliki karakteristik data yang kompleks dan distribusi kelas yang tidak seimbang. Kondisi tersebut berpotensi menurunkan kinerja model dan meningkatkan kinerja komputasi serta dapat menyebabkan bias terhadap kelas mayoritas.

Secara umum, tahap *pre-processing* dalam penelitian ini meliputi pembersihan data, penanganan *class imbalance*, normalisasi fitur numerik dan *encoding* fitur kategorikal. Selanjutnya, dilakukan pembatasan jumlah data per-kelas menjadi maksimal 800 data per-kelas, langkah ini dilakukan untuk mengurangi dominasi kelas mayoritas dan menghasilkan distribusi kelas yang lebih seimbang. Pendekatan ini selaras dengan penelitian[12] yang membahas penangan *class imbalance* pada data IDS, dimana teknik *undersampling* digunakan untuk merepresentasi kelas minoritas dan memperbaiki kinerja model klasifikasi. Sehingga didapatkan jumlah total data setelah *pre-processing* sebesar 8249 baris data dan 79 kolom.

Tahapan-tahapan ini bertujuan untuk mengurangi *noise,* menyetarakan skala data, dan memastikan seluruh fitur dapat berjalan optimal pada proses *machine learning* yang digunakan [11].

3. Data Splitting

Dataset yang telah melalui tahap *preprocessing* kemudian dibagi menjadi tiga bagian, yaitu data latih, data validasi dan data uji, yang digunakan untuk proses pelatihan serta evaluasi model. Pembagian dilakukan menggunakan metode *train_test_split* dengan proporsi 70% sebagai data latih, 15% sebagai validitas dan 15% sebagai data uji. Pada proses ini, digunakan opsi *stratify* untuk memastikan bahwa distribusi kelas pada kedua subset tetap seimbang dan merepresentasikan kondisi dataset secara keseluruhan. Dengan demikian, model dapat dilatih dan diuji pada data yang memiliki komposisi label yang konsisten.

4. Feature Selection

Proses setelahnya adalah seleksi fitur yang dilakukan menggunakan metode lasso regression sebelum tahap pelatihan model random forest dengan bayesian. Metode ini dipilih karena mampu menangani dataset berdimensi tinggi serta secara otomatis melakukan pemilihan fitur melalui mekanisme regularisasi.

Lasso Regression bekerja dengan menambahkan penalti L1 terhadap koefisien regresi sehingga mendorong nilai koefisien fitur yang kurang relevan mendekati nol. Fitur-fitur yang memiliki koefisien bernilai nol dianggap tidak memberikan kontribusi signifikan terhadap proses prediksi dan akan dieliminasi dari dataset. Sebaliknya, fitur dengan koefisien lebih dari nol akan dipertahankan sebagai fitur terpilih dan digunakan pada tahap pemodelan selanjutnya. Dalam konteks dataset yang digunakan, metode lasso regression diaplikasikan pada data hasil pre-processing yang telah dinormalisasi. Proses ini sangat krusial dalam konteks data berdimensi tinggi karena dapat mengurangi mengurangi jumlah fitur, menekan kompleksitas model, serta meminimalkan pengaruh noise dan redundansi antar fitur guna meningkatkan efisiensi dan akurasi model.

5. Implementasi Random Forest

Tahap selanjutnya adalah mengimplementasikan algoritma Random forest pada data latih. Random forest dipilih karena memiliki performa yang baik pada dataset berukuran besar, toleran terhadap ketidakseimbangan kelas, serta relatif tahan terhadap overfitting. Selain itu, algoritma ini menyediakan informasi feature importance yang membantu dalam memahami kontribusi setiap fitur terhadap proses klasifikasi.

Random Forest bekerja dengan membangun sejumlah decision tree menggunakan subset data dan subset fitur yang dipilih secara acak atau random. Setiap decision tree akan menghasilkan prediksi dan keputusan akhir akan ditentukan melalui mekanisme majority voting. Pada tahap awal, model Random Forest dibangun menggunakan konfigurasi parameter default sebagai baseline melalui fungsi RandomForestClassifier.

6. Hyperparameter Tuning dengan Bayesian Optimization

Pada tahap ini, dilakukan hyperparameter tuning untuk meningkatkan model deteksi menggunakan bayesian optimization.Metode ini dipilih karena mampu untuk mencari kombinasi yang optimal secara efisien dengan memanfaatkan informasi hasil dari evaluasi model sebelumnya, sehingga akan lebih efektif dibandingkan metode pencarian lain, seperti grid search atau random search.

Metode ini bekerja dengan membangun model probabalistik kemudian menentukan kombinasi berikutnya yang paling sesuai berdasarkan keseimbangan antara eksplorasi dan eksploitasi. Proses tuning dilakukan dengan menggunakan BayesSearchCV dengan skema cross-validation pada data latih.

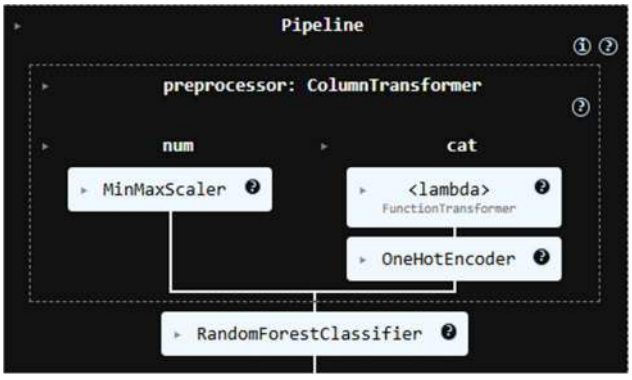
7. Evaluasi

Pada tahap ini dilakukan evaluasi dan analisis daripada model sistem deteksi yang telah dibuat. Evaluasi dilakukan dengan menggunakan confusion matrix dan sejumlah metrik statistik seperti akurasi, precision, recall, dan F1-score.

IV. HASIL DAN PEMBAHASAN

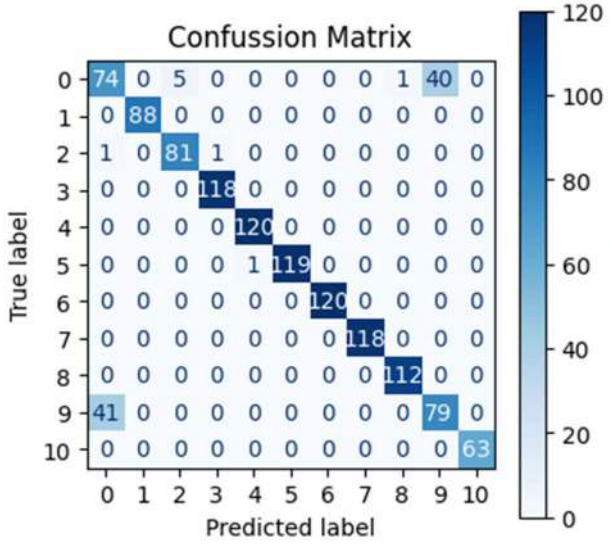
A. Pengujian Model Baseline

Model pertama yang akan diuji adalah model dasar yang hanya menggunakan Random Forest Classifier tanpa melakukan *feature selection* dan optimasi dengan *bayesian optimization*. Pemodelan dilakukan dengan menggunakan Pipeline dimana data telah dipisah sebelumnya antara data yang bersifat numerikal dan kategorikal. Data yang bersifat numerikal sudah dilakukan dan melewati tahap *pre-processing* dengan normalisasi data menggunakan Min Max Scaler. Sementara data yang bersifat kategorikal sudah dilakukan *pre-processing* berupa One Hot Encoder. Pipeline pemodelan yang digunakan pada pengujian ini dapat dilihat pada Gambar 2.



GAMBAR 2
Pipeline Baseline Model

Kemudian dilakukan analisis performa model dilakukan dengan membandingkan hasil prediksi dari data testing tersebut dengan niai yang sebenarnya dari label data testing dengan menggunakan *confusion matrix*. Confusion Matrix yang dihasilkan pada pengujian ini dapat dilihat pada Gambar 3.



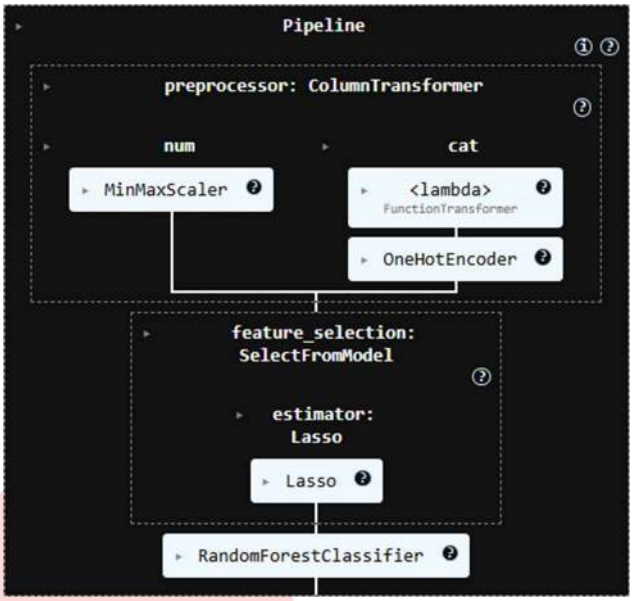
GAMBAR 3
Confusion Matrix Baseline Model

Berdasarkan hasil *confusion matrix*, performa model klasifikasi terhadap berbagai jenis serangan jaringan menunjukkan tingkat akurasi yang relatif tinggi dengan distribusi prediksi yang sebagian besar terkonsentrasi pada diagonal utama.

Secara keseluruhan, distribusi hasil prediksi pada Gambar 3 menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap sebagian besar jenis serangan maupun lalu lintas normal. Meski demikian, peningkatan masih diperlukan pada kelas-kelas dengan tingkat misklasifikasi yang tinggi, misalnya melalui penambahan fitur pembeda, seleksi fitur yang lebih relevan, atau metode optimasi tambahan yang sesuai dengan karakteristik data. Tahapan perbaikan tersebut akan dilakukan pada pengujian selanjutnya.

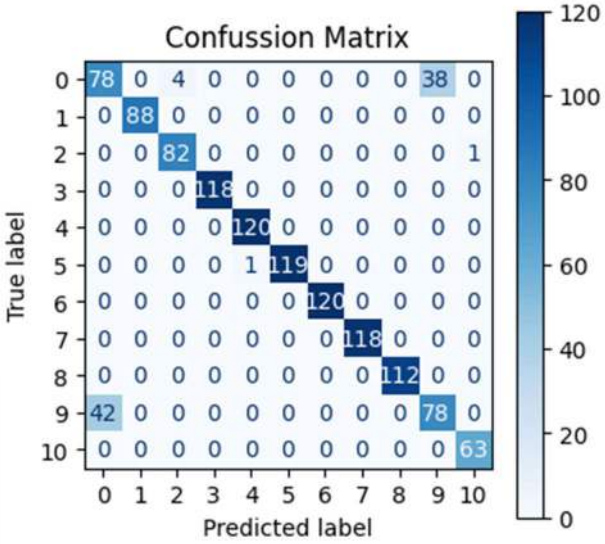
B. Pengujian Model Baseline Dengan Feature Selection

Model kedua yang akan diuji adalah model dasar Random Forest Classifier dengan menambahkan *feature selection* menggunaka *lasso regression* untuk menjawab tantangan pada pengujian sebelumnya. Pemodelan dilakukan dengan menggunakan Pipeline dimana data telah dipisah sbelumnya antara data yang bersifat numerikal dan kategorikal. Data yang bersifat numerikal sudah dilakukan preprocessing dengan normalisasi data menggunakan Min Max Scaler. Sementara data yang bersifat kategorikal sudah dilakukan *pre-processing* berupa One Hot Encoder. Pipeline pemodelan yangdigunakan pada pengujian ini dapat dilihat pada Gambar 4.



GAMBAR 4
Pipeline Baseline Model dengan Feature Selection

Kemudian dilakukan analisis performa model dilakukan dengan membandingkan hasil prediksi dari data testing tersebut dengan niai yang sebenarnya dari label data testing dengan menggunakan *confusion matrix*. Confusion Matrix yang dihasilkan pada pengujian ini dapat dilihat pada Gambar 5.



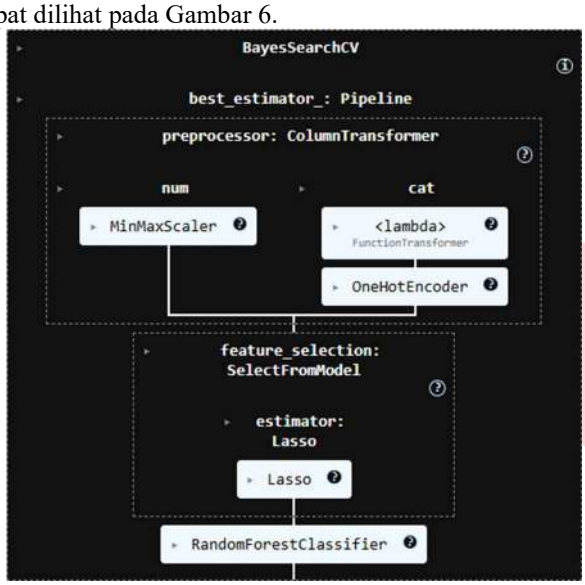
GAMBAR 5
Confusion Matrix Pengujian Model Kedua

Berdasarkan hasil *confusion matrix* model kedua yang telah ditambahkan *lasso regression*, terlihat adanya peningkatan konsistensi klasifikasi pada sebagian besar kelas dibandingkan dengan model sebelumnya (baseline model). Hal ini menunjukkan bahwa penerapan *lasso regression* tidak menurunkan kemampuan model dalam mengidentifikasi pola yang sudah kuat pada kelas-kelas tersebut.

Secara keseluruhan, model kedua menunjukkan hasil yang stabil dengan peningkatan ketepatan prediksi pada beberapa kelas minor dan pengurangan kesalahan klasifikasi acak, menandakan bahwa *lasso regression* berhasil meningkatkan kemampuan model dalam mengontrol kompleksitas dan memperkuat generalisasi.

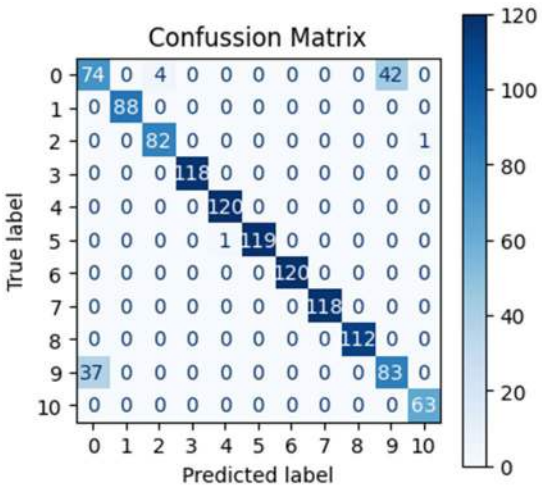
C. Pengujian Model Baseline Dengan Feature Selection & Bayesian Optimization

Model ketiga yang akan diuji adalah model dasar Random Forest Classifier dengan menambahkan *feature selection* menggunakan *lasso regression* dan optimasi dengan bayesian optimization. Pemodelan dilakukan dengan menggunakan pipeline dengan perlakuan yang sama seperti pada dua model sebelumnya. Pipeline yang digunakan pada pemodelan ini dapat dilihat pada Gambar 6.



GAMBAR 6
Pipeline Baseline Model dengan Feature Selection & Bayesian Optimization

Kemudian dilakukan analisis performa model dilakukan dengan membandingkan hasil prediksi dari data testing tersebut dengan niai yang sebenarnya dari label data testing dengan menggunakan *confusion matrix*. Confusion Matrix yang dihasilkan pada pengujian ini dapat dilihat pada Gambar 7.



GAMBAR 7
Confusion Matrix Model Ketiga

Berdasarkan hasil *confusion matrix* model ketiga pada Gambar 4.6 yang telah ditingkatkan menggunakan kombinasi *Iasso regression* dan *bayesian optimization*, terlihat adanya peningkatan stabilitas dan konsistensi hasil klasifikasi dibandingkan dua model sebelumnya. Pola distribusi pada diagonal utama masih menunjukkan dominasi prediksi yang benar pada hampir semua kelas, yang menandakan bahwa model memiliki kemampuan klasifikasi yang kuat dan generalisasi yang baik.

Secara keseluruhan, model ketiga menunjukkan performa paling seimbang di antara ketiga model yang diuji. Penggunaan *bayesian optimization* setelah penerapan *lasso reggression* terbukti efektif dalam menemukan kombinasi parameter yang meningkatkan kemampuan model dalam mengenali pola penting sekaligus menekan kompleksitas yang berlebihan.

D. Studi Perbandingan

Setelah pengujian dilakukan secara menyeluruh, selanjutnya akan dibandingkan model terbaik dari ketiga model tersebut. Sekaligus menjawab apakah penerapan *lasso regression* dan *bayesian optimization* mampu meningkatkan performasi model secara keseluruhan. Hasil perbandingannya dapat dilihat pada Tabel 1.

TABEL 1
Tabel Perbandingan Hasil Akurasi Ketiga Model

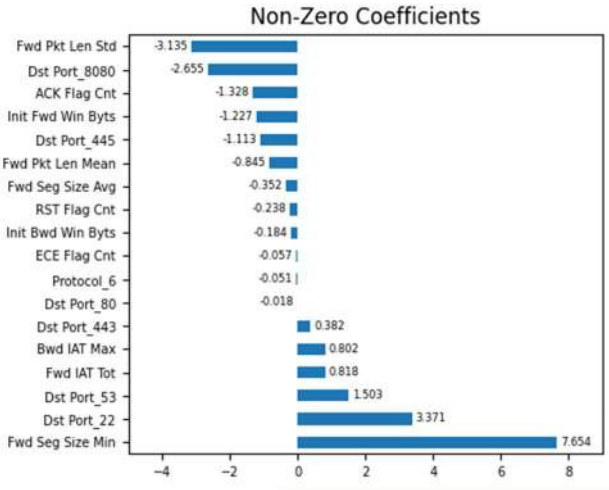
Model	Akurasi Keseluruhan	Rata-rata Precision	Rata-rata Recall	Rata-rata F1-score
Model 1 (Baseline)	**0. 9239**	**0. 9289**	**0. 9311**	**0. 9300**
Model 2 (Dengan Regularisasi)	**0. 9272**	**0.9320**	**0.9345**	**0.9332**
Model 3 (Lasso + Bayesian Optimization)	**0. 9281**	**0. 9328**	**0. 9353**	**0. 9339**

Dari hasil studi perbandingan pada Tabel 1 diatas dapat disimpulkan bahwa penggunaan *lasso regression* dan *bayesian optimization* telah memberikan dampak positif yang cukup signifikan terhadap performa model klasifikasi. *Lasso regression* berperan dalam mengeliminasi fitur-fitur yang tidak relevan, sehingga model menjadi lebih sederhana namun tetap akurat. Sementara itu, *bayesian optimization* secara sistematis mencari kombinasi yang paling optimal dengan efisiensi komputasi yang lebih tinggi. Kedua pendekatan ini secara simultan telah meningkatkan keseimbangan antara bias dan variansi, yang tercermin dari kenaikan akurasi dan *f1-score* pada hampir seluruh kelas. Dengan demikian, model ketiga dapat dianggap sebagai versi paling optimal dalam penelitian ini, baik dari segi performa prediksi maupun stabilitas antar kelas.

E. Analisis Koefisien Lasso Regression

Setelah model terbaik berhasil ditemukan (yaitu pada model ketiga dengan penambahan lasso regression dan bayesian optimization), selanjutnya akan dilakukan analisis-analisis terkait dengan model tersebut. Analisis pertama dilakukan dengan melihat apa yang sebenarnya terjadi pada lasso regression setelah melakukan seleksi fitur. Variabel yang tidak memiliki koefisien *lasso regression* dapat dikatakan memiliki signifikansi yang sangat-sangat kecil sehingga fitur tersebut akan dihilangkan pada tahapan

berikutnya. Semebntara fitur yang bersifat *non-zero* akan tetap dimasukan kedalam analisis. Hasilnya dapat dilihat pada Gambar 8.

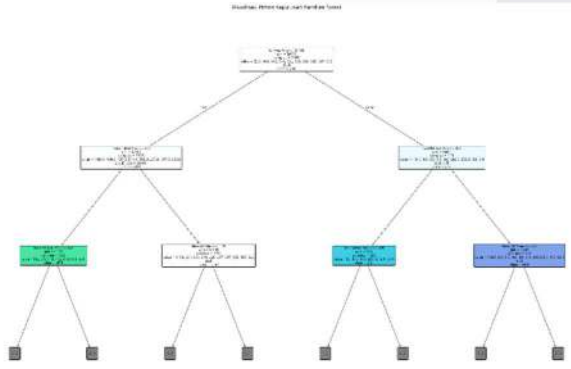


GAMBAR 8
Non-Zero Coefficient Lasso

Berdasarkan hasil visualisasi pada gambar 8, fitur Fwd Seg Size Min memiliki koefisien positif paling besar, menandakan bahwa peningkatan nilai minimum ukuran segmen pada arah forward sangat berpengaruh positif terhadap peningkatan prediksi kelas target. Fitur *Dst Port_22 dan Dst Port_53* juga menunjukkan pengaruh positif yang signifikan, yang mengindikasikan bahwa lalu lintas pada port tertentu dapat menjadi indikator penting dalam membedakan kategori data jaringan. Sebaliknya, fitur seperti *Fwd Pkt Len Std, Dst Port_8080, dan ACK Flag Cnt* memiliki koefisien negatif terbesar. Nilai negatif ini menunjukkan bahwa peningkatan nilai fitur-fitur tersebut justru menurunkan probabilitas terhadap kelas target tertentu.

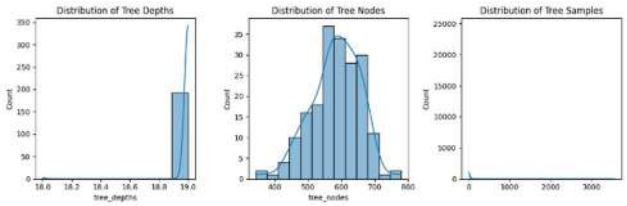
F. Tree Analysis

Secara umum, model *Random Forest* terdiri dari banyak sekali model *decision tree* (jumlah defaultnya adalah 100 buah). Dari kumpulan *decision tree* ini kemudian dilakukan mekanisme voting terhadap keputusan yang hendak dibuat. Contoh *decision tree* yang telah dibuat dapat dilihat pada gambar 9.



GAMBAR 9
Output Pohon Keputusan

Setelah melalui *Bayesian Optimization*, didapatkan jumlah *decision tree* (pohon keputusan) terbaik untuk model adalah sebanyak 322 pohon keputusan. 5 sampel pohon keputusan teratas dan 5 sampel pohon keputusan terbawah, Dari analisis pohon keputusan diatas secara rata-rata, tingkat kedalaman pohon yang dihasilkan adalah 23 baris dan rata-rata jumlah nodes yang terbentuk pada setiap pohon adalah 353 *nodes*. Kedalaman minimum adalah 16 baris dan kedalaman maksimum adalah 30 baris. Sementara jumlah *nodes* paling sedikit adalah 309 *nodes* dan yang paling banyak adalah 401 *nodes*. Grafik distribusinya dapat dilihat pada Gambar 10.

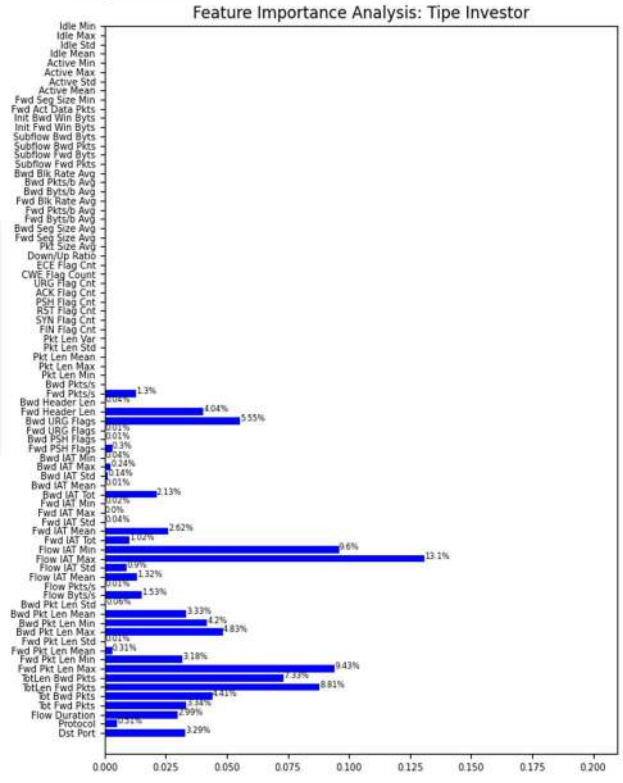


GAMBAR 10
Distribusi Atribut Pohon Keputusan yang Terbentuk

Gambar 10, menunjukkan distribusi dari tiga variabel utama hasil pemodelan pohon keputusan pada Model 3, yaitu *tree_depths*, *tree_nodes*, dan *tree_samples*. Distribusi *tree_depths* menunjukkan bahwa distribusi kedalaman pohon mayoritas memiliki kedalaman yang relatif seragam, yaitu sekitar 18-19. Hal ini menunjukkan bahwa model yang dibangun mempelajari pola yang cukup kompleks dan membutuhkan struktur pohon yang cukup dalam untuk menghitung kompleksitas trafik.

G. Feature Importance Analysis

Feature Importance Analysis merupakan sebuah analisis yang digunakan untuk mengukur kontribusi relatif masing-masing fitur terhadap hasil prediksi model. Nilai ini mengukuir seberapa besar kontribusi masing-masing fitur dalam mempengaruhi keputusan klasifikasi yang hasilnya dapat dilihat pada Gambar 11.



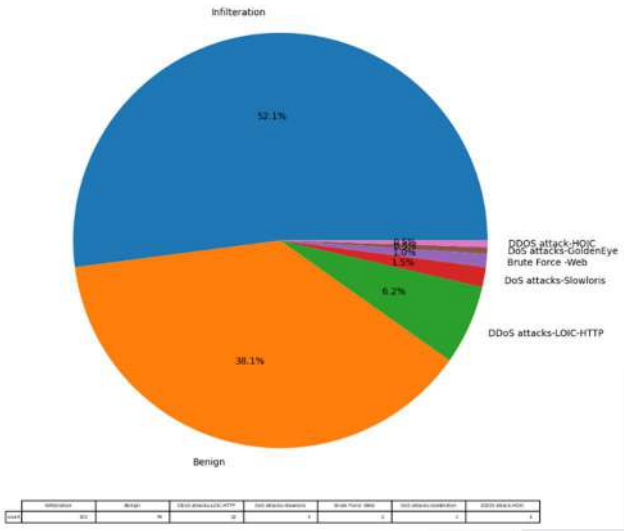
GAMBAR 11
Importance Feature Analysis

Dalam model terbaik (model ketiga) yang menggunakan kombinasi *lasso regression* dan *bayesian optimization*, proses seleksi fitur dilakukan secara dua tahap. Pertama, *lasso regression* yang melakukan regularisasi untuk menekan koefisien fitur yang kontribusinya kecil hingga bernilai nol. Hal ini bertujuan mengeliminasi fitur yang redundan, tidak relevan, atau memiliki korelasi tinggi dengan fitur lain. Kedua, hasil reduksi fitur tersebut kemudian digunakan sebagai masukan untuk model Random Forest, yang selanjutnya menghitung feature importance.

Dari grafik *feature importance* pada Gambar 11, terlihat bahwa beberapa fitur memiliki nilai kontribusi yang sangat tinggi, sementara sebagian besar lainnya bernilai nol menunjukkan bahwa fitur-fitur tersebut telah dieliminasi oleh *lasso regression* sebelum proses pelatihan model Random Forest dilakukan.

H. Voting Mechanism

Voting Mechanism adalah bagian dari explandable AI yang berfungsi untuk men jelaskan mengapa sebuah keputusan dibuat. Pada model yang sudah dilatih dihasilkan 322 buah pohon keputusan, dan Random Forest bekerja dengan melakukan prediksi pada setiap pohon dan mengambil keputusan dengan teknik voting, yaitu mengambil suara atau keputusan terbanyak. Hasil voting pada data pada 322 pohon keputusan tersebut terhadap data pada Tabel 4.6 dapat dilihat pada Gambar 12.



GAMBAR 12
Voting Mechanism

Dari Gambar 12, dapat dilihat bahwa 52.1% pohon keputusan menghasilkan keputusan Infilteration Keputusan ini merupakan keputusan terbanyak (mayoritas). Inilah yang menyebabkan model mengeluarkan keputusan Infilteration.

I. Pembahasan

Ketiga model yang dikembangkan menunjukkan peningkatan performa bertahap seiring penambahan teknik optimasi. Model pertama hanya menggunakan baseline arsitektur tanpa regularisasi tambahan, model kedua menambahkan regularisasi dasar untuk mengatasi overfitting, sedangkan model ketiga mengintegrasikan lasso regression dan bayesian optimization untuk penyetelan parameter yang lebih efisien.

Pada model pertama, telah dihasilkan performa yang cukup baik, dimana akurasi yang dihasilkan sudah mencapai 92.39%. Hal ini sejalan dengan penelitian sebelumnya yang menyatakan bahwa random forest memiliki kemampuan yang kuat dalam menangani data berdimensi tinggi dan imbalance [2]. Namun, masih terdapat beberapa miss klasifikasi pada beberapa data. Penambahan lasso regression pada model kedua memperlihatkan keseimbangan yang lebih baik antara kompleksitas dan akurasi. Lasso Regression berperan sebagai metode regularisasi yang mengeliminasi fitur tidak relevan dengan menekan koefisien fitur menjadi nol, sehingga hanya fitur yang relevan yang dipertahankan [3]. Hasil pengujian memperlihatkan bahwasanya nilai precision dan recall yang tinggi pada mayoritas kelas menunjukkan kemampuan model dalam melakukan klasifikasi yang konsisten, stabil, dan efisien. Dengan demikian, penambahan lasso regression terbukti efektif dalam meningkatkan generalisasi model tanpa mengorbankan performa pada kelas mayor, meskipun peningkatan tambahan masih dapat diperoleh dengan melakukan optimasi lanjutan pada fitur pembeda antara kelas GoldenEye dan Brute Force - Web. Sementara model ketiga memperlihatkan kinerja paling optimal di antara seluruh eksperimen. Penerapan lasso regression terbukti efektif dalam melakukan regularisasi untuk mengurangi kompleksitas model, sedangkan bayesian optimization berperan dalam menemukan parameter terbaik yang menyeimbangkan antara bias dan variance. Kombinasi keduanya menghasilkan model yang lebih efisien, akurat, dan stabil, sehingga berpotensi besar untuk diterapkan dalam sistem deteksi intrusi (IDS) yang adaptif dan andal di lingkungan jaringan nyata.

Hasil confusion matrix dan classification report memperlihatkan bahwa model ketiga memiliki akurasi keseluruhan sebesar 92,81%, meningkat signifikan dibanding dua model sebelumnya. Peningkatan ini menunjukkan bahwa kombinasi lasso dan bayesian mampu memperkuat generalisasi model, terutama pada kelas dengan distribusi data yang tidak seimbang.

Secara umum, model ketiga menghasilkan nilai precision, recall, dan f1-score yang sangat tinggi (mendekati 1.00) pada sebagian besar kelas, dengan pengecualian pada kelas 0 dan kelas 9 yang masih menunjukkan sedikit kesalahan prediksi akibat kemiripan fitur antar kelas. Namun demikian, peningkatan konsistensi antar metrik menunjukkan stabilitas model dalam mengenali pola pada data uji model uji.

V. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa penerapan mengenai fitur berbasis Lasso Regression dan optimasi Hyperparameter menggunakan Bayesian efektif dalam meningkatkan kinerja algortima Random Forest dalam deteksi intrusi jaringan pada dataset CSE-CIC-IDS2018. Seleksi fitur dengan Lasso Regression mampu mengurangi kompleksitas data berdimensi tinggi dengan tetap mempertahankan fitur-fitur yang penting dan mengeliminasi fitur-fitur yang kurang atau tidak berkontribusi terhadap hasil akhir model. Selain itu, penerapan Bayesian Optimization berhasil mengoptimalkan konfigurasi hyperparameter Random Forest secara sistematis

yang berdampak pada peningkatan akurasi dan konsistensi model dibandingkan dengan model Baseline.

Secara keseluruhan, penelitian ini berhasil mencapai tujuan, yaitu mengoptimalkan performa deteksi intrusi pada dataset CSE-CIC-IDS2018 melalui integrasi seleksi fitur dan optimasi hyperparameter. Model terbaik mampu memberikan performa klasifikasi yang tinggi, stabil, dan efisien, sehingga berpotensi diterapkan pada sistem deteksi intrusi berbasis machine learning di lingkungan jaringan nyata.

REFERENSI

[1] C. Zhang, D. Jia, L. Wang, W. Wang, F. Liu, and A. Yang, "Comparative research on network intrusion detection methods based on machine learning," *Comput. Secur.*, vol. 121, p. 102861, 2022, doi: 10.1016/j.cose.2022.102861.

[2] B. Pes, "Learning from high-dimensional and class-imbalanced datasets using random forests," *Inf.*, vol. 12, no. 8, 2021, doi: 10.3390/info12080286.

[3] Z. Mei and Z. Shi, "On LASSO for high dimensional predictive regression," *J. Econom.*, vol. 242, no. 2, 2024, doi: 10.1016/j.jeconom.2024.105809.

[4] I. Sartika and N. N. Debataraja, "ANALISIS REGRESI DENGAN METODE LEAST ABSOLUTE SHRINKAGE AND SELECTION OPERATOR (LASSO) DALAM MENGATASI MULTIKOLINEARITAS," vol. 09, no. 1, pp. 31–38, 2020, doi: https://doi.org/10.26418/bbimst.v9i1.38029.

[5] C. Yokochi, R. Bispo, F. Ricardo, and R. Calado, "Regularization Methods for High-Dimensional Data as a Tool for Seafood Traceability," *J. Stat. Theory Pract.*, vol. 17, no. 3, pp. 1–21, 2023, doi: 10.1007/s42519-023-00341-8.

[6] S. Roy, J. Li, B. J. Choi, and Y. Bai, "A lightweight supervised intrusion detection mechanism for IoT networks," *Futur. Gener. Comput. Syst.*, vol. 127, pp. 276–285, 2022, doi: 10.1016/j.future.2021.09.027.

[7] D. Conn, T. Ngun, G. Li, and C. M. Ramirez, "Fuzzy forests: Extending random forest feature selection for correlated, high-dimensional data," *J. Stat. Softw.*, vol. 91, no. 9, 2019, doi: 10.18637/jss.v091.i09.

[8] S. Sharma, V. Kumar, and K. Dutta, "Multi-objective optimization algorithms for intrusion detection in IoT networks: A systematic review," *Internet Things Cyber-Physical Syst.*, vol. 4, no. February, pp. 258–267, 2024, doi: 10.1016/j.iotcps.2024.01.003.

[9] B. Satria, "Optimasi random forest dengan ga dan rfe pada high dimensional data stunting skripsi," 2024.

[10] N. Salman and M. Sari, "PENGARUH PENYETELAN HYPERPARAMETER TERHADAP KINERJA PREDIKSI RANDOM FOREST PADA PENDETEKSIAN SPAM," *J. INSTEK (Informatika Sains dan Teknol.*, vol. 5, no. 2, p. 149, Aug. 2020, doi: 10.24252/instek.v5i2.15617.

[11] I. Muhamad Malik Matin, "Hyperparameter Tuning Menggunakan GridsearchCV pada Random Forest untuk Deteksi Malware," *Multinetics*, vol. 9, no. 1, pp. 43–50, 2023, doi: 10.32722/multinetics.v9i1.5578.

[12] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *J. Big Data*, vol. 8, no. 1, p. 6, 2021, doi: 10.1186/s40537-020-00390-x.

[13] Y. R. K and G. S. Lingam, "INTELLIGENT SYSTEMS AND APPLICATIONS IN A Novel Hybrid Deep Learning Framework with Multi-Scale Temporal Convolutions , Attention , and Uncertainty Quantification for Network Intrusion Detection," vol. 12, pp. 1018–1028, 2024, doi: https://doi.org/10.1038/s41598-025-32697-1.

[14] R. Yacouby and D. Axman, "Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models," pp. 79–91, 2020, doi: 10.18653/v1/2020.eval4nlp-1.9.

[15] S. S. Islam, M. S. Haque, M. S. U. Miah, T. Bin Sarwar, and R. Nugraha, "Application of machine learning algorithms to predict the thyroid disease risk: an experimental comparative study," *PeerJ Comput. Sci.*, vol. 8, pp. 1–35, 2022, doi: 10.7717/peerj-cs.898.