Laporan Objective Quest 2024

Mohamad Maulana Firdaus Ramadhan[a], Erwan Poltak Halomoan[b], Viktor Arsindiantoro Siringoringo[c]

[a](*18222140*) *Sistem dan Teknologi Informasi, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Bandung*
[b](*18222028*) *Sistem dan Teknologi Informasi, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Bandung*
[c](*18222083*) *Sistem dan Teknologi Informasi, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Bandung*

Abstrak
Keamanan siber menjadi prioritas utama bagi perusahaan dengan lalu lintas jaringan tinggi, mengingat ancaman serangan siber seperti Bruteforce, Probing, dan aktivitas berbahaya lainnya yang semakin kompleks. Dalam penelitian ini, kami mengembangkan model prediksi menggunakan machine learning untuk mendeteksi serangan pada traffic jaringan. Dataset yang digunakan mencakup traffic normal dan beberapa jenis serangan. Teknik feature engineering diterapkan untuk meningkatkan akurasi, terutama pada fitur host, port, flag, IAT, dan packet counts. Model CatBoost dengan class weight yang disesuaikan menghasilkan scoring 0.872 setelah dilakukan threshold tuning. Evaluasi menunjukkan model dapat mendeteksi sebagian besar serangan dengan baik.

Kata Kunci: Machine learning, deteksi serangan jaringan, traffic jaringan, threshold tuning, class weight

1. Pendahuluan

Dalam era digital saat ini, keamanan siber menjadi salah satu prioritas utama bagi perusahaan yang mengelola jaringan dengan lalu lintas tinggi, di mana volume data yang diproses di seluruh dunia mencapai lebih dari 100 zettabytes per tahun. Ancaman serangan siber seperti bruteforce, probing, dan aktivitas berbahaya lainnya semakin kompleks, seiring dengan bertambahnya jumlah pengguna dan perangkat yang terhubung ke internet. Untuk menjaga integritas sistem, .

Tantangan utama dalam analisis lalu lintas jaringan ini adalah keberagaman pola traffic, mulai dari traffic yang tidak berbahaya (benign) hingga traffic berbahaya seperti bruteforce dan probes. Fitur penting seperti origin host, response host, origin port, dan response port memberikan informasi tentang asal dan tujuan dari lalu lintas jaringan, membantu mengidentifikasi sumber serta target serangan. Selain itu, fitur lain seperti flow duration (durasi aliran jaringan), forward packets per second (laju paket dari klien ke server), backward packets per second (laju paket dari server ke klien), serta flow flags (seperti SYN, ACK, FIN) berperan penting dalam mengkarakterisasi pola komunikasi dan potensi anomali di jaringan. Kompleksitas data dan volume lalu lintas yang sangat besar membuat klasifikasi dan deteksi ancaman menjadi tantangan yang lebih besar, terutama dalam menangani data yang tidak seimbang antar kelas traffic.

Penelitian ini bertujuan untuk mengembangkan model prediktif yang dapat mengklasifikasikan traffic dalam jaringan secara akurat, dengan memanfaatkan fitur utama seperti origin host, response host, origin port, response port dan fitur lainnya seperti flow duration, forward packets per second, backward packets per second, dan flow flags. Model ini diharapkan dapat mendeteksi berbagai jenis traffic dan potensi serangan secara efektif, memungkinkan perusahaan untuk merespons ancaman secara proaktif dan meningkatkan keamanan jaringan secara keseluruhan.

2. Analisis dan Pembahasan

Bagian ini akan membahas secara detail proses analisis yang dilakukan untuk mengklasifikasikan traffic jaringan, mulai dari pengenalan data yang digunakan hingga hasil akhir dari model prediktif yang dikembangkan. Tahapan-tahapan yang dilalui meliputi deskripsi data awal, analisis eksplorasi (exploratory data analysis), pra-pemrosesan data (preprocessing), rekayasa fitur (feature engineering), serta pengembangan dan evaluasi model.

2.1.*Deskripsi Data*

Dalam upaya menyelesaikan masalah dan mencapai tujuan yang telah ditetapkan, kami telah memperoleh sebuah dataset yang berfokus pada analisis lalu lintas jaringan. Dataset ini dikumpulkan untuk memahami pola traffic jaringan, mengidentifikasi aktivitas mencurigakan, dan meningkatkan strategi keamanan siber. Dataset ini memungkinkan kami untuk menganalisis jenis-jenis traffic yang berbeda serta mengenali pola-pola tertentu yang dapat membantu dalam deteksi dini aktivitas mencurigakan di dalam jaringan. Dataset ini terdiri dari berbagai jenis traffic, yang mencakup traffic normal (benign), traffic latar belakang (background), dan traffic berbahaya (attacker).

Dataset yang diperoleh mencakup 416.473 baris data dan 43 kolom, termasuk kolom target yang terdiri dari beberapa **jenis traffic seperti Background, Benign, Probing, Bruteforce, XMRIGCC CryptoMiner, dan Bruteforce-XML.** Selain itu, dataset ini mengandung informasi penting yang berkaitan dengan komunikasi jaringan, seperti **alamat IP asal dan tujuan, nomor port, durasi aliran, serta sejumlah fitur teknis** yang berhubungan dengan analisis traffic. Berikut merupakan tabel yang berisi deskripsi seluruh fitur yang ada dalam dataset

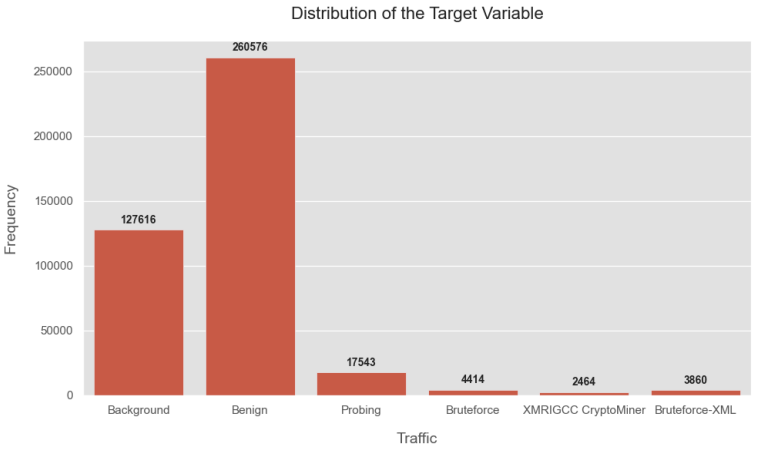
Tabel 1. Deskripsi Fitur Fitur dalam Dataset

Fitur	**Deskripsi**	**Fitur**	**Deskripsi**
id	ID untuk setiap aliran jaringan	flow_pkts_payload	Ukuran rata-rata payload dalam seluruh aliran, mencakup paket dari client dan server
origin_host	Alamat IP client	forward_iat	Rata-rata waktu antar kedatangan (Inter Arrival Time, IAT) antara paket dari client ke server
origin_port	Nomor port yang digunakan oleh client	backward_iat	Rata-rata waktu antar kedatangan (IAT) antara paket dari server ke client
response_host	Alamat IP server	flow_iat	Rata-rata waktu antar kedatangan (IAT) dalam seluruh aliran, mencakup paket dari client dan server
response_port	Nomor port yang digunakan oleh server	payload_bytes_per_sec	Laju transfer byte payload per detik dalam aliran jaringan
flow_duration	Durasi aliran jaringan	forward_subflow_packets	Jumlah paket dalam subflow dari client ke server
forward_packets_per_sec	Laju pengiriman paket dari client ke server per detik	backward_subflow_packets	Jumlah paket dalam subflow dari server ke client
backward_packets_per_sec	Laju pengiriman paket dari server kembali ke client per detik	forward_subflow_bytes	Jumlah byte dalam subflow dari client ke server
flow_packets_per_sec	Laju keseluruhan paket yang dikirim per detik dalam seluruh aliran	backward_subflow_bytes	Jumlah byte dalam subflow dari server ke client
down_up_ratio	Rasio antara jumlah paket atau byte yang diunduh dan diunggah	forward_bulk_bytes	Jumlah byte bulk dalam aliran dari client ke server
flow_FIN_flags	Jumlah bendera FIN yang menandakan akhir dari koneksi TCP.	backward_bulk_bytes	Jumlah byte bulk dalam aliran dari server ke client
flow_SYN_flags	Jumlah bendera SYN yang digunakan untuk memulai koneksi TCP.	forward_bulk_packets	Jumlah paket bulk dalam aliran dari client ke server
flow_RST_flags	Jumlah bendera RST yang menandakan reset koneksi TCP.	backward_bulk_packets	Jumlah paket bulk dalam aliran dari server ke client
forward_PSH_flags	Jumlah bendera PSH yang menandakan data harus segera diproses oleh server.	forward_bulk_rate	Laju lalu lintas bulk dalam aliran dari client ke server

backward_PSH_flags	Jumlah bendera PSH yang menandakan data harus segera diproses oleh client.	backward_bulk_rate	Laju lalu lintas bulk dalam aliran dari server ke client
flow_ACK_flags	Jumlah bendera ACK yang digunakan untuk mengonfirmasi penerimaan paket.	active	Rata-rata waktu aktif dalam aliran, menggambarkan periode ketika data sedang ditransmisikan
forward_URG_flags	Jumlah bendera URG yang menandakan data mendesak dari client.	idle	Rata-rata waktu idle dalam aliran, menggambarkan periode ketika tidak ada data yang ditransmisikan
backward_URG_flags	Jumlah bendera URG yang menandakan data mendesak dari server.	forward_initial_window_size	Ukuran Window Flow Control yang tersedia dari client ke server pada awal koneksi, menentukan jumlah data yang dapat dikirim tanpa memerlukan konfirmasi.
flow_CWR_flags	Jumlah bendera CWR yang menandakan pengurangan ukuran jendela kongesti.	backward_initial_window_size	Ukuran Window Flow Control yang tersedia dari server ke client pada awal koneksi, menentukan jumlah data yang dapat dikirim tanpa memerlukan konfirmasi.
flow_ECE_flags	Jumlah bendera ECE yang menunjukkan adanya kemacetan jaringan.	forward_last_window_size	Ukuran Window Flow Control dari client ke server pada akhir koneksi, menentukan jumlah data terakhir yang dapat dikirim sebelum koneksi ditutup.
forward_pkts_payload	Ukuran rata-rata payload dalam paket dari client ke server	traffic	Jenis traffic yang harus diprediksi: Background - Lalu lintas rutin atau latar belakang yang biasanya tidak menimbulkan ancaman. Benign - Traffic yang tidak menunjukkan tanda-tanda aktivitas berbahaya atau anomali. Probing - Aktivitas pemindaian atau penjelajahan untuk menemukan kerentanan atau target. Bruteforce - Upaya penyerangan dengan mencoba berbagai kombinasi kata sandi untuk mendapatkan akses. XMRIGCC Crypto Miner - Aktivitas terkait dengan penambangan cryptocurrency secara tersembunyi. Bruteforce-XML - Serangan brute force yang menargetkan aplikasi berbasis XML
backward_pkts_payload	Ukuran rata-rata payload dalam paket dari server ke client		

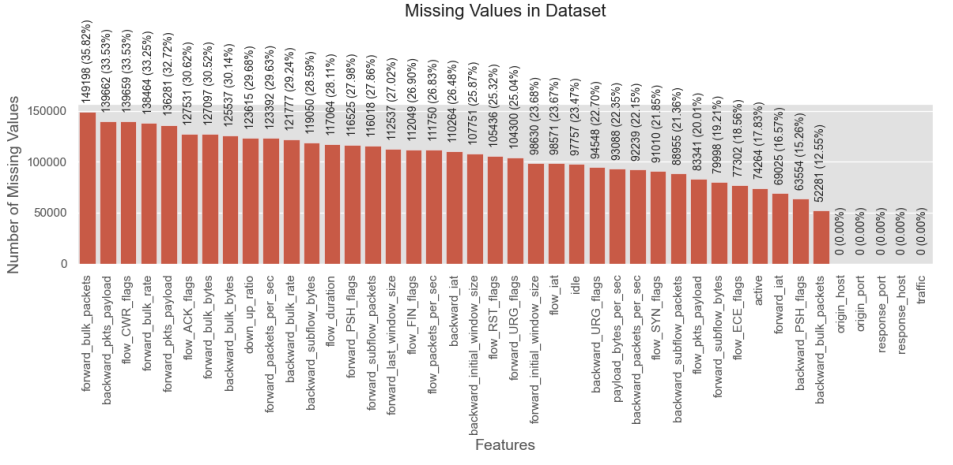
2.2.Exploratory Data Analysis

Pada tahap ini, dilakukan Exploratory Data Analysis (EDA) untuk memahami karakteristik dataset secara mendalam. EDA membantu **mengidentifikasi pola, potensi bias, serta kondisi data** yang dapat mempengaruhi kinerja model prediksi.



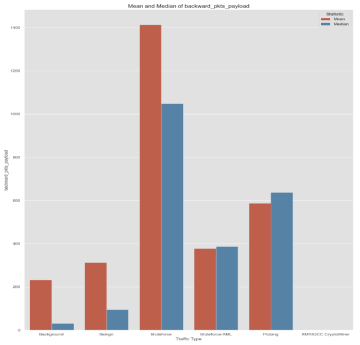
Gambar 1. Distribusi Data Traffic

Distribusi kategori traffic pada dataset menunjukkan ketidakseimbangan yang mencolok, di mana traffic berbahaya seperti **Probing**, **Bruteforce**, dan **XMRIGCC CryptoMiner** memiliki jumlah *instance* yang sangat sedikit dibandingkan dengan kategori **Benign** dan **Background**. Ketidakseimbangan ini berpotensi mempengaruhi kemampuan model dalam mendeteksi traffic berbahaya yang lebih jarang terjadi, karena model mungkin lebih fokus pada pengenalan pola dari kategori yang mendominasi. Hal ini dapat mengarah pada rendahnya sensitivitas model terhadap anomali atau serangan siber yang lebih jarang, tetapi berisiko tinggi. Upaya untuk menyeimbangkan data atau menerapkan teknik penanganan ketidakseimbangan akan sangat penting untuk meningkatkan akurasi dan generalisasi model.

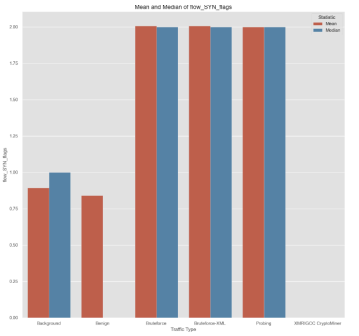


Gambar 2. Distribusi Data NaN

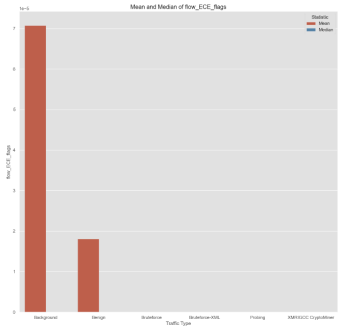
Dataset ini memiliki sejumlah fitur dengan missing values yang cukup signifikan, seperti **forward_bulk_packets** dengan 149.198 nilai hilang dan **backward_bulk_bytes** dengan 139.662 nilai hilang. Selain itu, **flow_duration** dan **forward_packets_per_sec** juga mengalami *missing values* dalam jumlah besar, yang berpotensi mempengaruhi analisis. Penanganan *missing values* ini perlu dilakukan untuk mencegah bias dan menjaga performa model, baik melalui teknik imputasi maupun penghapusan data. Jika tidak diatasi, *missing values* dapat mengurangi akurasi dan reliabilitas model prediksi.



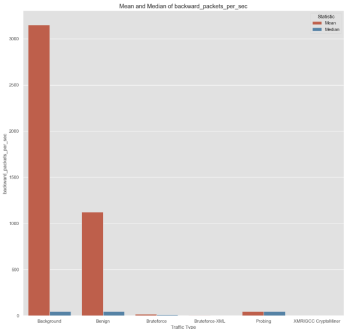
Gambar 3. Distribusi Mean/Median Normal



Gambar 4. Distribusi Mean/Median Sama



Gambar 5. Mean/Median Fitur Jomplang

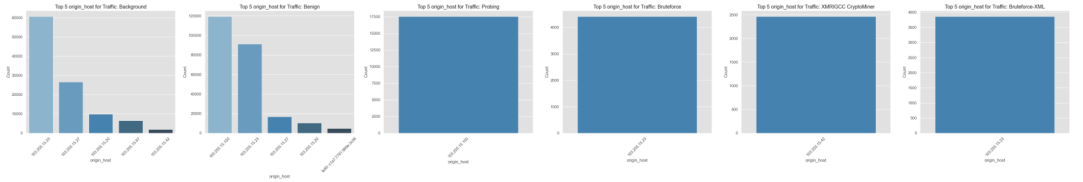


Gambar 6. Mean/Median Fitur 0

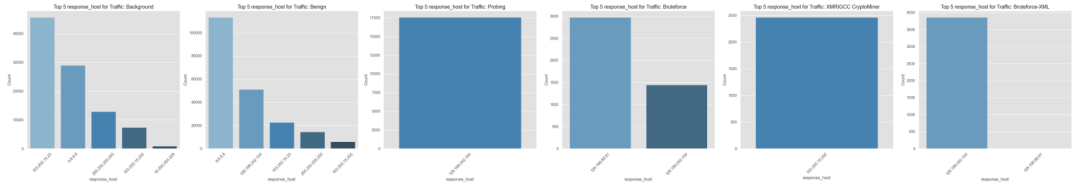
Berdasarkan visualisasi distribusi mean dan median dari berbagai fitur terhadap label traffic, terlihat bahwa beberapa kategori traffic memiliki nilai mean dan median yang mirip. Hal ini bisa menjadi indikasi bahwa model prediksi berisiko mengalami bias ketika membedakan antara kategori-kategori traffic tertentu. Ketika mean dan median antar kategori mirip, model cenderung tidak dapat mengidentifikasi karakteristik unik dari masing-masing kategori. Ini dapat menyebabkan model memprediksi traffic berbahaya sebagai traffic normal karena kurangnya perbedaan yang signifikan dalam distribusi data di beberapa fitur. Situasi ini meningkatkan potensi *false negatives*, di mana traffic berbahaya tidak terdeteksi.

Selain mean dan median yang mirip antar kategori, beberapa fitur menunjukkan nilai mean dan median yang sama-sama **0** di berbagai kategori traffic. Hal ini mengindikasikan bahwa fitur tersebut memiliki banyak nilai nol, yang dapat mengurangi kebermanfaatannya dalam membedakan traffic. Ketika sebagian besar data dalam sebuah fitur bernilai nol, kontribusi fitur tersebut dalam proses prediksi menjadi terbatas, karena tidak menyediakan cukup variasi untuk mengidentifikasi pola-pola yang berbeda di antara jenis traffic. Kondisi ini dapat menyebabkan model tidak mampu memanfaatkan fitur tersebut secara optimal dan, dalam kasus ekstrem, fitur ini mungkin diabaikan oleh model dalam pemrosesan lebih lanjut.

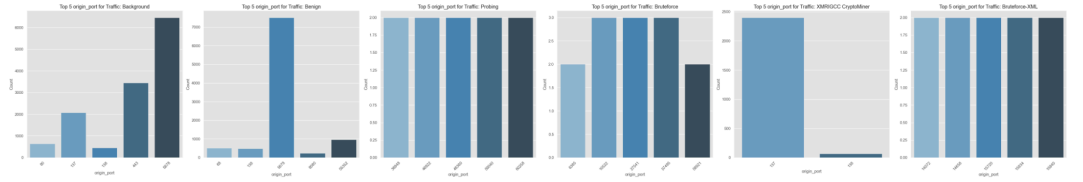
Selain itu, beberapa fitur juga menunjukkan adanya **kejomplangan** yang signifikan antar kategori. Misalnya, fitur tertentu mungkin memiliki nilai yang sangat besar pada satu kategori traffic seperti **Background**, tetapi bernilai nol atau mendekati nol pada kategori lain seperti **Benign** atau **Bruteforce**. Kejomplangan ini menandakan perbedaan yang ekstrem dalam distribusi data antar kategori, yang bisa memperkuat kemampuan model dalam mengidentifikasi traffic tertentu, tetapi juga meningkatkan risiko bias. Jika tidak ditangani dengan benar, fitur ini dapat menyebabkan model terlalu fokus pada kategori dengan nilai yang besar, sementara kategori lainnya tidak mendapatkan representasi yang memadai, sehingga menimbulkan potensi ketidakseimbangan dalam performa prediksi.



Gambar 7. Distribusi Origin Host Setiap Target



Gambar 8. Distribusi Response Host Setiap Target

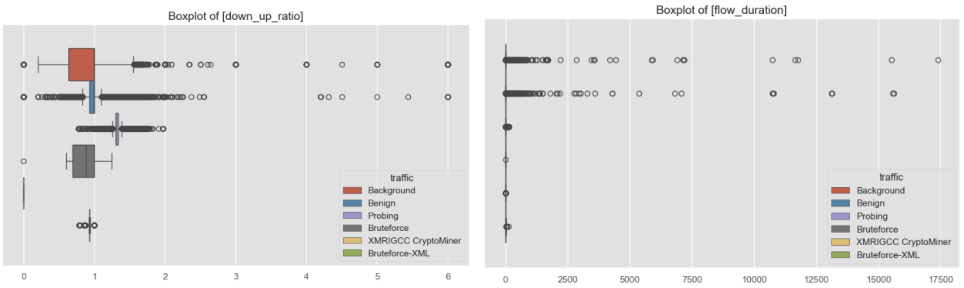


Gambar 9. Distribusi Origin Port Setiap Target

Gambar 10. Distribusi Response Port Setiap Target

Pada traffic *non-attack*, seperti **Background** dan **Benign**, pola penggunaan **origin host** dan **response host** serta **origin port** dan **response port** cenderung lebih bervariasi dibandingkan dengan traffic serangan. **Background traffic** menunjukkan penggunaan host dan port yang lebih beragam, mencerminkan aktivitas jaringan normal yang melibatkan berbagai sumber dan tujuan. Traffic **Benign** juga memiliki distribusi yang lebih merata, meskipun terdapat pola konsistensi pada beberapa host dan port yang sering digunakan, seperti port 443 untuk HTTPS.

Sebaliknya, pada traffic **attack** seperti **Bruteforce** dan **XMRIGCC CryptoMiner**, terlihat pola serangan terfokus, di mana serangan berulang kali menggunakan host dan port yang sama, menargetkan alamat IP tertentu secara intensif. Pola ini mencerminkan serangan yang terstruktur dan menyasar celah keamanan tertentu secara sistematis. Perbedaan ini sangat penting untuk membantu model prediksi membedakan antara traffic normal dan traffic berbahaya, karena variasi yang lebih rendah pada traffic attack memudahkan identifikasi pola serangan, sementara traffic non-attack lebih bervariasi dalam aktivitasnya.

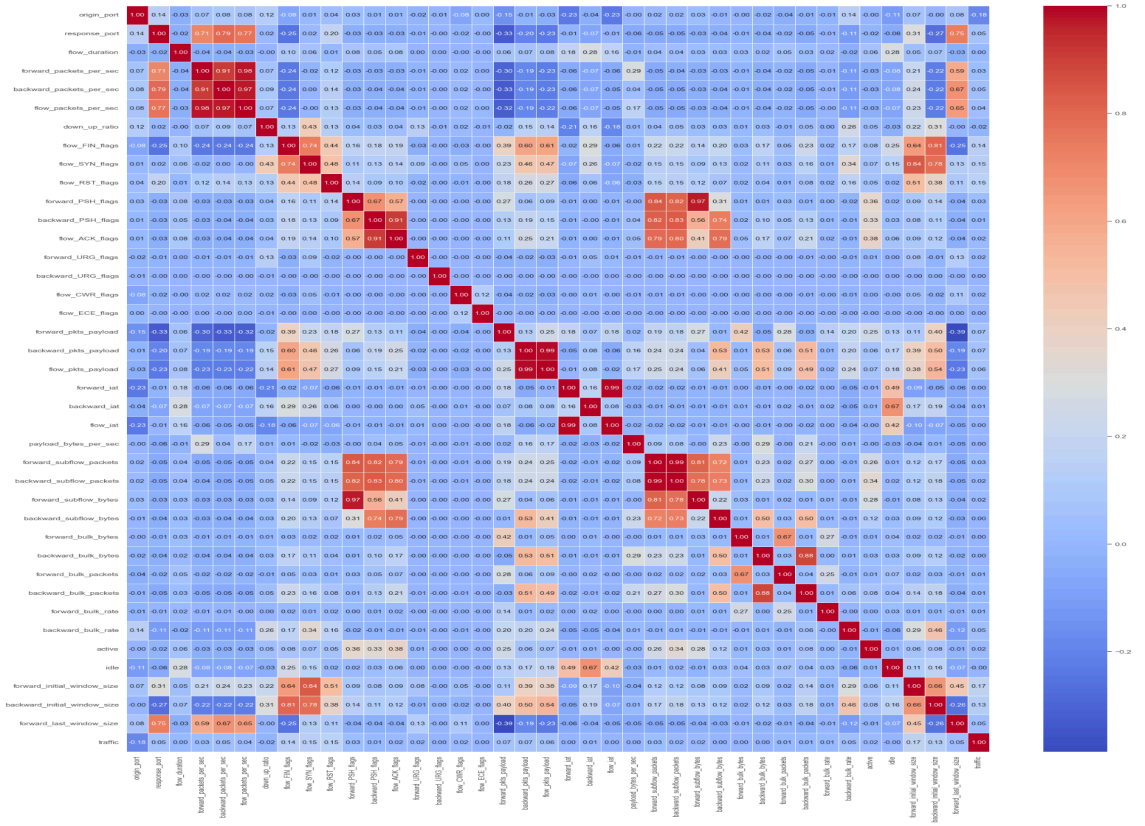


Gambar 11. Distribusi Boxplot down_up_ratio

Secara keseluruhan, boxplot menunjukkan adanya outlier di berbagai fitur dalam dataset. Meskipun outlier umumnya dianggap sebagai anomali yang bisa mengganggu performa model, dalam konteks ini, outlier **tidak dihapus** karena dianggap **merefleksikan karakteristik penting** yang dapat membantu membedakan antara **kelas traffic normal dan berbahaya**.

Sebagai contoh, pada fitur down_up_ratio, terlihat bahwa kelas Benign dan Background memiliki rentang nilai yang lebih kecil dibandingkan dengan kelas serangan seperti Probing dan Bruteforce. Outlier yang muncul justru menunjukkan perbedaan signifikan dalam rasio download dan upload, yang penting untuk mengidentifikasi serangan.

Demikian pula pada fitur flow_duration, outlier ditemukan terutama pada kelas Probing dan Bruteforce, yang menunjukkan durasi aliran yang lebih lama dibandingkan traffic normal. Durasi yang ekstrem ini penting dalam mengidentifikasi pola serangan yang mencoba mengeksploitasi koneksi dalam jangka waktu lebih panjang, sehingga outlier dalam hal ini memberikan kontribusi yang signifikan untuk klasifikasi.



Gambar 12. Heatmap Korelasi

Berdasarkan matriks korelasi, beberapa fitur menunjukkan korelasi yang sangat tinggi, yang dapat dijadikan dasar untuk aturan **imputasi berbasis hubungan antar fitur**.. Di sisi lain, meskipun ada fitur dengan korelasi rendah, hal ini tidak selalu berarti tidak ada hubungan. Kemungkinan adanya hubungan non-linear antara fitur-fitur ini mengindikasikan bahwa penggunaan model non-linear lebih tepat daripada model linear untuk menangkap pola yang kompleks dalam data.

2.3.Data Preprocessing

Pada tahap ini, dilakukan data preprocessing untuk menangani missing values dengan metode imputasi berbasis aturan (rule-based imputation). Pendekatan ini didasarkan pada hubungan antar fitur yang memiliki korelasi tinggi, seperti fitur yang berhubungan dengan paket, payload, dan durasi aliran. Pendekatan ini memastikan bahwa nilai yang hilang dapat diisi secara konsisten berdasarkan nilai yang tersedia dari fitur lain yang berkaitan, sehingga tidak mengganggu kualitas prediksi. Berikut adalah beberapa aturan imputasi yang diterapkan:

*2.3.1.*Imputasi Forward, Backward, dan Flow Packets per Second:

	forward_packets_per_sec	backward_packets_per_sec	flow_packets_per_sec
151776	0.000000	0.000000	0.000000
192474	13706.875817	13706.875817	27413.751634
244469	44.461235	57.799606	102.260842
372768	21290.883249	21290.883249	42581.766497

Gambar 13. Kolom Forward, Backward, dan Flow Packets per Second

- Jika flow_packets_per_sec hilang:

$$\text{flow\_packets\_per\_sec} = \begin{cases} 2 \times \text{forward\_packets\_per\_sec} & \text{jika backward\_packets\_per\_sec hilang} \\ 2 \times \text{backward\_packets\_per\_sec} & \text{jika forward\_packets\_per\_sec hilang} \\ \text{forward\_packets\_per\_sec} + \text{backward\_packets\_per\_sec} & \text{jika keduanya ada} \end{cases}$$

- Jika forward_packets_per_sec hilang:

$$\text{forward\_packets\_per\_sec} = \begin{cases} \text{flow\_packets\_per\_sec} - \text{backward\_packets\_per\_sec} & \text{jika backward\_packets\_per\_sec ada} \\ \text{backward\_packets\_per\_sec} & \text{jika flow\_packets\_per\_sec hilang} \end{cases}$$

- Jika backward_packets_per_sec hilang:

$$\text{backward\_packets\_per\_sec} = \begin{cases} \text{flow\_packets\_per\_sec} - \text{forward\_packets\_per\_sec} & \text{jika forward\_packets\_per\_sec ada} \\ \text{forward\_packets\_per\_sec} & \text{jika flow\_packets\_per\_sec hilang} \end{cases}$$

Gambar 14. Imputasi Forward, Backward, dan Flow Packets per Second

*2.3.2.*Imputasi Payload Packets:

	forward_pkts_payload	backward_pkts_payload	flow_pkts_payload
151776	201.0	0.0	201.000000
192474	0.0	0.0	0.000000
244469	83.9	637.0	396.521739
372768	0.0	0.0	0.000000

Gambar 15. Kolom Payload Packets

- Jika forward_pkts_payload atau backward_pkts_payload hilang:

$$\text{forward\_pkts\_payload} = 2 \times \text{flow\_pkts\_payload} - \text{backward\_pkts\_payload} \quad \text{jika backward\_pkts\_payload ada}$$

$$\text{backward\_pkts\_payload} = 2 \times \text{flow\_pkts\_payload} - \text{forward\_pkts\_payload} \quad \text{jika forward\_pkts\_payload ada}$$

$$\text{flow\_pkts\_payload} = \frac{\text{forward\_pkts\_payload} + \text{backward\_pkts\_payload}}{2} \quad \text{jika keduanya ada}$$

Gambar 16. Imputasi Payload Packets

*2.3.3.*Imputasi Inter Arrival Time (IAT):

	forward_iat	backward_iat	flow_iat
151776	0.000000	0.000000	0.000000
192474	0.000000	0.000000	72.956085
244469	24990.558624	16342.163086	10223.410346
372768	0.000000	0.000000	46.968460

Gambar 17. Kolom Inter Arrival Time (IAT)

- Jika forward_iat atau backward_iat hilang:

$$\begin{aligned} \text{forward\_iat} &= 4 \times \text{flow\_iat} - \text{backward\_iat} && \text{jika backward\_iat ada} \\ \text{backward\_iat} &= 4 \times \text{flow\_iat} - \text{forward\_iat} && \text{jika forward\_iat ada} \\ \text{flow\_iat} &= \frac{\text{forward\_iat} + \text{backward\_iat}}{4} && \text{jika keduanya ada} \end{aligned}$$

Gambar 18. Imputasi Inter Arrival Time (IAT)

*2.3.4.*Imputasi Subflow Packets:

	forward_subflow_packets	backward_subflow_packets	down_up_ratio
151776	1.0	0.0	0.0
192474	1.0	1.0	1.0
244469	10.0	13.0	1.3
372768	1.0	1.0	1.0

Gambar 19. Kolom Subflow Packets

- Jika forward_subflow_packets hilang:

$$\text{forward\_subflow\_packets} = \begin{cases} \text{backward\_subflow\_packets} \times \text{down\_up\_ratio} & \text{jika backward\_subflow\_packets dan down\_up\_ratio ada} \\ 0 & \text{jika down\_up\_ratio} \leq 0 \text{ atau tidak ada data} \end{cases}$$

- Jika backward_subflow_packets hilang:

$$\text{backward\_subflow\_packets} = \begin{cases} \frac{\text{forward\_subflow\_packets}}{\text{down\_up\_ratio}} & \text{jika down\_up\_ratio} > 0 \text{ dan forward\_subflow\_packets ada} \\ 0 & \text{jika down\_up\_ratio} \leq 0 \text{ atau tidak ada data} \end{cases}$$

Gambar 20. Imputasi Subflow Packets

*2.3.5.*Imputasi Flow Duration:

	flow_duration	active	idle
151776	0.000000	0.000000	0.0
192474	0.000073	72.956085	0.0
244469	0.224915	224915.027618	0.0
372768	0.000047	46.968460	0.0

Gambar 21. Kolom Flow Duration

- Konversi nilai idle dan active:

$$\text{idle} = \frac{\text{idle}}{1,000,000}$$

$$\text{active} = \frac{\text{active}}{1,000,000}$$

- Jika flow_duration hilang:

$$\text{flow\_duration} = \text{active} + \text{idle}$$

- Jika active hilang:

$$\text{active} = \text{flow\_duration} - \text{idle}$$

- Jika idle hilang:

$$\text{idle} = \text{flow\_duration} - \text{active}$$

.

Gambar 22. Imputasi Flow Duration

Namun, perlu dicatat bahwa metode ini belum mencakup keseluruhan missing values di dataset. Oleh karena itu, diperlukan eksperimen tambahan untuk menentukan strategi imputasi yang paling efektif untuk sisa missing values. Ada tiga pendekatan yang diusulkan untuk eksperimen ini:

- Fill Mean: Mengisi missing values dengan rata-rata nilai dari masing-masing fitur.
- Fill Median: Mengisi missing values dengan nilai tengah (median) dari fitur.
- Tanpa Imputasi: Membiarkan missing values tetap ada tanpa diisi, yang kemudian akan diolah oleh model sesuai dengan metode penanganan missing values yang disediakan oleh model tersebut.

2.4.Feature Engineering

Pada tahap feature engineering, dilakukan konversi tipe data untuk memastikan model dapat menginterpretasikan fitur-fitur kategorikal dengan benar, terutama untuk kolom yang berkaitan dengan category dan port. Fitur-fitur yang awalnya bertipe object, seperti origin_host dan response_host, dikonversi menjadi tipe category. Hal ini penting agar model dapat memahami bahwa fitur-fitur tersebut bersifat kategorikal, bukan teks, sehingga bisa mengenali kategori yang berbeda dengan lebih efisien.

Selain itu, kolom origin_port dan response_port yang mungkin sebelumnya dianggap sebagai data numerik, juga diubah menjadi tipe category. Ini karena port bukanlah angka yang bersifat kontinu, melainkan label yang menunjukkan jenis koneksi atau layanan tertentu dalam jaringan. Dengan melakukan konversi ini, model dapat memperlakukan setiap nilai port sebagai kelas kategori yang berbeda, bukan sebagai nilai numerik yang dapat memengaruhi interpretasi model.

2.5.Feature Creation

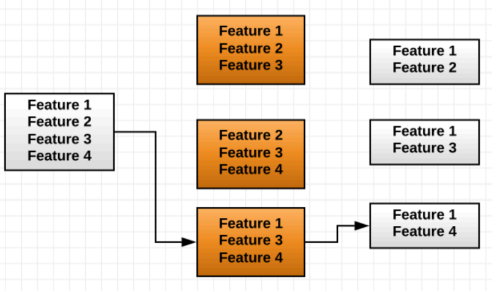
Fitur-fitur yang diciptakan melalui FeatureCreator ini bertujuan untuk meningkatkan daya prediksi model dengan memanfaatkan hubungan yang mungkin tidak terlihat jelas dari fitur-fitur asli. Beberapa alasan utama pembuatan fitur ini adalah sebagai berikut:

- Binary Feature for PSH Flags: Fitur biner is_PSH diciptakan untuk menangkap jumlah PSH flags yang sangat tinggi (>150), yang dapat menjadi indikator penting dalam mendeteksi jenis traffic tertentu, seperti serangan atau pola penggunaan jaringan yang berbeda dari biasanya.
- Kombinasi Host dan Port: Berbagai kombinasi antara origin_host, response_host, dan port (baik origin maupun response) dibuat untuk menangkap interaksi yang lebih kompleks antara sumber dan tujuan komunikasi jaringan. Kombinasi ini memungkinkan model untuk lebih mudah mendeteksi pola-pola yang mungkin tersembunyi dalam hubungan antara host dan port, yang bisa menjadi indikator kuat dalam membedakan jenis traffic, seperti serangan yang terfokus pada host atau port tertentu.
- Host-to-Port Ratio: Rasio ini dibuat untuk mengukur hubungan antara panjang string origin_host sebagai proksi untuk keunikan dan origin_port. Rasio ini bisa mengindikasikan apakah sebuah

host cenderung berinteraksi dengan berbagai port yang berbeda, yang mungkin menunjukkan aktivitas normal, atau jika interaksi terbatas pada port tertentu, yang dapat menjadi sinyal serangan yang terfokus.

- Frequency Encoding: Frequency encoding pada origin_host dan response_host dibuat untuk menangkap popularitas atau frekuensi penggunaan host tertentu. Host yang sering muncul mungkin merupakan bagian dari traffic yang normal, sementara host yang jarang muncul bisa mengindikasikan potensi serangan atau anomali.
- Port Service Type: Fitur port_service_type diciptakan untuk mengklasifikasikan origin_port dan response_port berdasarkan layanan yang umumnya dikaitkan dengan port tersebut, seperti HTTP, HTTPS, FTP, atau SSH. Hal ini membantu model memahami jenis komunikasi jaringan yang terjadi dan bisa menjadi indikator penting dalam memisahkan traffic normal dari traffic berbahaya yang mungkin terjadi pada port layanan tertentu.

2.6.Feature Selection



Gambar 23. Visualisasi Backward Elimination

Pada tahap feature selection, langkah pertama yang dilakukan adalah menerapkan metode backward feature elimination. Metode ini bertujuan untuk mengidentifikasi dan menghapus fitur-fitur yang tidak memberikan kontribusi signifikan terhadap akurasi model. Proses ini dilakukan dengan menghilangkan satu fitur dalam setiap iterasi dan mengukur dampaknya terhadap metrik. Fitur yang jika dihapus justru meningkatkan performa model dianggap sebagai fitur yang tidak penting.

Backward feature elimination dimulai dengan mempertahankan semua fitur, kecuali fitur kategorikal yang tidak akan dihapus. Setelah itu, fitur yang paling tidak berpengaruh pada performa model diidentifikasi dan dihapus dari kumpulan fitur. Proses ini diulang hingga tidak ada lagi fitur yang dapat dihapus tanpa menurunkan performa model. Setelah didapatkan daftar fitur yang tidak penting dari proses backward elimination, fitur-fitur tersebut kemudian akan di-drop. Dengan begitu, dataset yang dihasilkan menjadi lebih ringan dan model dapat lebih fokus pada fitur yang benar-benar penting, meningkatkan efisiensi dan akurasi prediksi.

2.7.One Hot Encoding

Kemudian dilakukan one-hot encoding untuk fitur port service type. Fitur ini awalnya bertipe kategorikal dengan beberapa kategori, seperti HTTP, HTTPS, FTP, dan SSH. Karena data kategorikal ini tidak memiliki urutan atau skala numerik, penggunaan one-hot encoding bertujuan untuk menghindari model menganggap adanya hubungan hierarkis atau urutan di antara kategori tersebut, yang bisa terjadi jika fitur dikodekan sebagai nilai numerik biasa.

Metode one-hot encoding ini mengubah setiap kategori pada kolom origin_port_service_type dan response_port_service_type menjadi kolom biner yang terpisah. Setiap kolom biner mewakili satu kategori, dengan nilai 1 jika data tersebut termasuk dalam kategori tersebut, dan 0 jika tidak. Dengan cara ini, model dapat memproses data secara lebih tepat tanpa membuat asumsi yang salah tentang hubungan antara kategori yang berbeda.

2.8.Model Selection

Dalam proses pemilihan model, beberapa pertimbangan penting diambil berdasarkan karakteristik data dan kebutuhan dalam menangani hubungan antar fitur. Pada tahap ini, beberapa jenis model telah dievaluasi, dan keputusan dibuat untuk menggunakan model non-linear.

Model linear, seperti Logistic Regression atau Linear SVM, tidak dipilih karena dataset menunjukkan adanya hubungan non-linear yang kuat antar fitur. Hubungan-hubungan ini tidak dapat ditangkap dengan baik oleh model linear yang mengasumsikan bahwa semua hubungan antara fitur dan target bersifat linier. Oleh karena itu, model linear dianggap tidak efektif dalam memodelkan kompleksitas data ini.

Model berbasis instance, seperti k-Nearest Neighbors (k-NN), juga tidak dipilih. Alasan utamanya adalah sensitivitas model ini terhadap outlier. Dataset ini memiliki potensi outlier, yang dapat menyebabkan model berbasis instance gagal memprediksi dengan baik karena mereka cenderung memperlakukan setiap instance secara individu. Selain itu, kinerja model instance-based juga lambat pada dataset yang besar seperti ini, karena proses pencarian instance terdekat cukup memakan waktu.

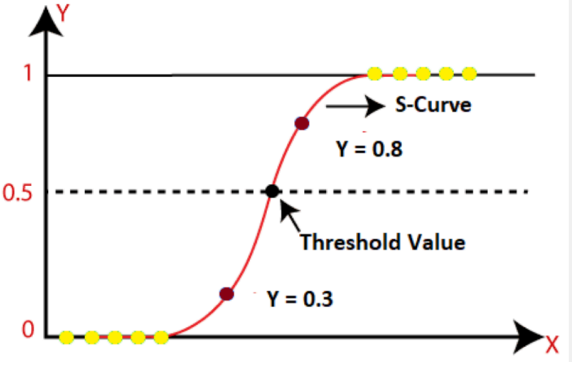
Pada akhirnya, CatBoost dan LightGBM dipilih sebagai model utama. Kedua model ini memiliki kemampuan yang kuat untuk menangani hubungan non-linear antar fitur dan secara otomatis dapat menangani fitur kategorikal serta missing values, yang sangat relevan untuk dataset ini. Selain itu, kedua model ini memiliki performa yang cepat, terutama pada dataset yang besar dan banyak kolom seperti yang sedang digunakan. CatBoost dan LightGBM dirancang untuk pelatihan yang efisien, memungkinkan pelatihan pada data besar tanpa mengorbankan akurasi.

Random Forest tidak dipilih meskipun merupakan model non-linear yang efektif, karena waktu pelatihan yang lambat pada dataset besar dengan banyak kolom. Proses pelatihan yang melibatkan banyak pohon keputusan dalam Random Forest memerlukan waktu yang signifikan, yang tidak ideal untuk skenario ini. Selain itu, XGBoost juga tidak dipilih karena meskipun merupakan model yang kuat, fitur class weight pada XGBoost tidak bisa langsung diterapkan dan harus dimasukkan secara manual saat fitting, yang memakan waktu lebih lama dan menambah kompleksitas proses.

Salah satu alasan utama memilih CatBoost dan LightGBM adalah kemampuan mereka dalam menangani missing values dan fitur kategorikal tanpa banyak preprocessing tambahan. Kedua model ini memungkinkan fitur-fitur seperti origin_host dan response_port untuk diproses secara langsung sebagai kategori, yang sangat penting untuk dataset ini.

CatBoost dan LightGBM juga dipilih karena mendukung penggunaan class weights untuk mengatasi masalah ketidakseimbangan kelas yang mungkin ada pada dataset ini. Class weights memungkinkan model untuk memberikan bobot lebih besar pada kelas minoritas, seperti traffic berbahaya, sehingga model dapat lebih sensitif dalam mendeteksi kelas tersebut tanpa bias terhadap kelas mayoritas.

2.9.Threshold Tuning



Gambar 24. Visualisasi Threshold Model

Threshold tuning digunakan untuk memaksimalkan kinerja model dengan menyesuaikan batas probabilitas (threshold) untuk setiap kelas, sehingga menghasilkan prediksi yang lebih akurat. Biasanya, threshold default 0.5 tidak optimal, terutama pada dataset dengan ketidakseimbangan kelas. Dalam proses ini, berbagai nilai threshold dicoba untuk setiap kelas, dan threshold yang memberikan scoring terbaik—kombinasi accuracy dan balanced accuracy—dipilih sebagai yang optimal. Setelah threshold

optimal ditemukan untuk setiap kelas, model menghasilkan prediksi akhir dengan menggunakan nilai threshold tersebut. Ini meningkatkan sensitivitas model terhadap kelas minoritas yang biasanya terabaikan dengan threshold default.

2.10.Design Experiment

Desain eksperimen yang digunakan berfokus pada dua aspek utama: penanganan missing values dan pemilihan model yang optimal.

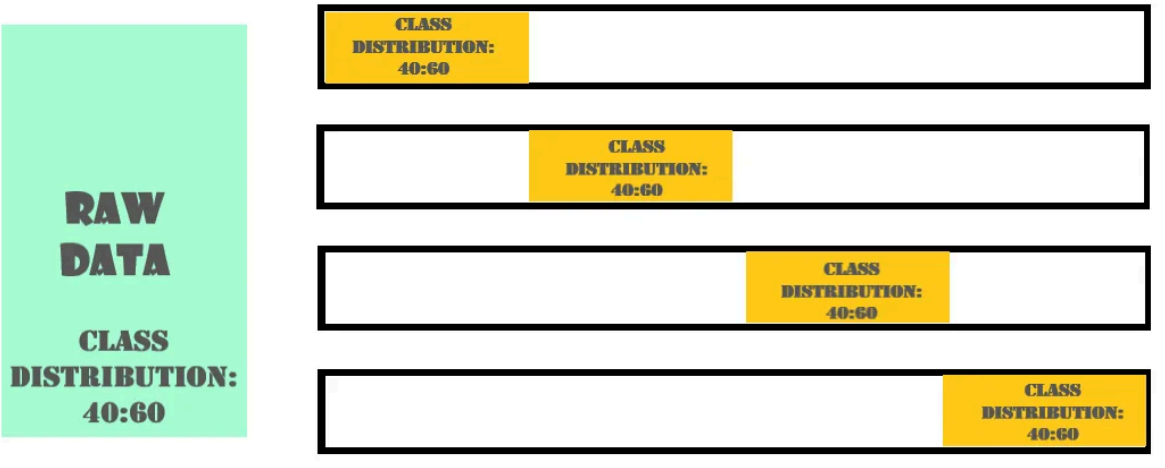
2.10.1. Pada tahap pertama, eksperimen dilakukan untuk menentukan metode terbaik dalam menangani data yang mengandung missing values. Tiga pendekatan yang akan diuji adalah:

- Mengisi missing values dengan mean.
- Mengisi missing values dengan median.
- Membiarkan missing values tanpa diisi (tidak dilakukan imputasi).

2.10.2. Pemilihan Model: Model yang dipilih untuk eksperimen adalah CatBoost dan LightGBM, karena keduanya unggul dalam menangani missing values, fitur kategorikal, serta dapat memproses dataset besar secara efisien. Dua pendekatan yang akan diterapkan pada kedua model adalah:

- Baseline: Model standar tanpa penanganan untuk ketidakseimbangan kelas.
- Class Weight: Model dengan class weights untuk menyeimbangkan bobot antar kelas

2.10.3. Untuk mengevaluasi kinerja model, digunakan metode Stratified K-Fold cross-validation 5-fold menggunakan dengan custom scoring, yang menghubungkan accuracy dan balanced accuracy. Selain itu, classification report dan confusion matrix akan digunakan untuk analisis.



Gambar 25. Visualisasi Strafied K-Fold

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(y_{\text{true},i} = y_{\text{pred},i}) \qquad \text{Balanced Accuracy} = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FN_c} \qquad \text{Score} = \frac{\text{Accuracy} + \text{Balanced Accuracy}}{2}$$

Gambar 26. Metrik yang Digunakan

2.11.Experiment Evaluation

Tabel 2. Hasil Evaluasi Score

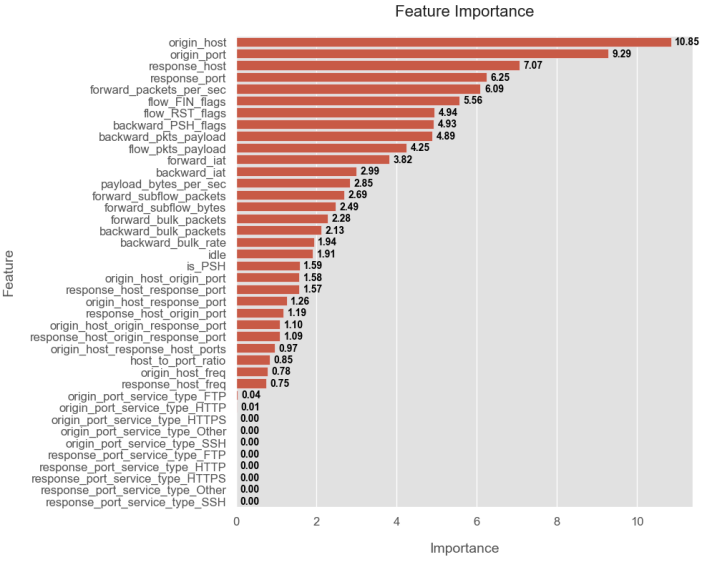
Imputasi	Model	Fold					Mean	Std
		1	2	3	4	5		
Mean	CatBoost Baseline	0.8126	0.8147	0.8116	0.8133	0.8137	0.8132	0.001095
	CatBoost ClassWeight	0.8640	0.8644	0.8629	0.8622	0.8622	0.8631	0.000839
	LightGBM Baseline	0.7217	0.7154	0.7291	0.7255	0.7311	0.7245	0.005712
	LightGBM ClassWeight	0.8267	0.8282	0.8235	0.8254	0.8254	0.8258	0.001775
Median	CatBoost Baseline	0.8035	0.8058	0.8017	0.8050	0.8040	0.8040	0.001542
	CatBoost ClassWeight	0.8575	0.8578	0.8564	0.8562	0.8565	0.8569	0.000576
	LightGBM Baseline	0.7160	0.7182	0.7195	0.7237	0.7148	0.7185	0.003167
	LightGBM ClassWeight	0.8209	0.8214	0.8190	0.8196	0.8180	0.8198	0.001381
Baseline	CatBoost Baseline	0.8107	0.8140	0.8131	0.8123	0.8127	0.8126	0.001157
	CatBoost ClassWeight	0.8645	0.8649	0.8627	0.8629	0.8626	0.8635	0.000853
	LightGBM Baseline	0.7299	0.7331	0.7270	0.7380	0.7301	0.7316	0.004065
	LightGBM ClassWeight	0.8274	0.8281	0.8240	0.8273	0.8260	0.8266	0.001558

Berdasarkan hasil eksperimen, imputasi baseline (tanpa imputasi tambahan) memberikan hasil terbaik dibandingkan dengan imputasi mean dan median. Ini terlihat dari performa cross-validation yang lebih tinggi pada model CatBoost dan LightGBM ketika tidak dilakukan imputasi pada missing values.
Metode terbaik adalah ClassWeight, yang secara konsisten meningkatkan performa model dibandingkan dengan baseline tanpa class weight. Penggunaan class weight sangat efektif dalam menangani ketidakseimbangan kelas pada dataset, terbukti dengan peningkatan skor yang signifikan pada kedua model.

Model terbaik adalah CatBoost, terutama pada imputasi baseline dan menggunakan ClassWeight, yang memberikan skor cross-validation tertinggi sebesar 0.8635. Selain itu, LightGBM with ClassWeight juga memberikan hasil yang baik, meskipun di bawah CatBoost.

Nilai standard deviation yang rendah pada semua model, berkisar antara 0.0005 hingga 0.0057, menunjukkan bahwa model tidak terlalu overfit dan memiliki stabilitas yang baik. Konsistensi performa pada setiap fold cross-validation memastikan bahwa model dapat diandalkan untuk generalisasi pada data baru.

2.12.Model Interpretability



Gambar 27. Feature Importance

Berdasarkan hasil analisis feature importance, fitur yang paling berpengaruh dalam model adalah origin_host dan origin_port, diikuti oleh response_host dan response_port. Keempat fitur ini sangat krusial dalam membedakan jenis traffic pada jaringan, terutama dalam hal mengenali pola serangan dan aktivitas normal. Host dan port memainkan peran penting dalam mendeteksi anomali atau serangan, karena serangan sering kali melibatkan perubahan pola komunikasi dari atau ke host dan port tertentu.

Selain itu, fitur lain seperti forward_packets_per_sec, flow_FIN_flags, dan flow_RST_flags juga memiliki kontribusi yang signifikan. Fitur-fitur ini membantu dalam memahami karakteristik aliran data yang terlibat dalam setiap koneksi, termasuk bagaimana paket dikirimkan dan bagaimana sesi diakhiri. Adanya flag seperti FIN dan RST membantu model dalam mengidentifikasi koneksi abnormal atau yang berpotensi berbahaya.

Fitur tambahan seperti forward_iat dan backward_iat (Inter Arrival Time) juga penting karena memberikan informasi tentang waktu jeda antara paket yang diterima dan dikirim. Perubahan pada jeda waktu ini bisa menjadi indikator adanya serangan atau aktivitas tidak biasa.

Meskipun fitur seperti bulk packets, bulk rate, dan payload bytes per second tidak sebesar host dan port, mereka tetap memberikan kontribusi penting dalam mendeteksi pola traffic yang berbeda. Secara keseluruhan, kombinasi dari informasi tentang host, port, flags, dan karakteristik traffic lainnya memungkinkan model untuk secara efektif membedakan traffic normal dari serangan.

Selain fitur dengan pengaruh tinggi seperti origin_host, origin_port, response_host, dan response_port, ada beberapa fitur yang memiliki importance score 0, seperti origin_port_service_type_HTTP, origin_port_service_type_HTTPS, dan semua variasi response_port_service_type. Fitur-fitur ini tidak memberikan kontribusi dalam model, kemungkinan karena mereka tidak memberikan informasi tambahan yang signifikan dalam membedakan jenis traffic yang ada di dataset.

Hal ini menunjukkan bahwa tipe service yang digunakan pada port tidak relevan untuk prediksi dalam konteks dataset ini. Artinya, model lebih banyak memanfaatkan pola host dan port secara langsung, serta kombinasi paket dan flag, daripada tipe layanan pada port tersebut.

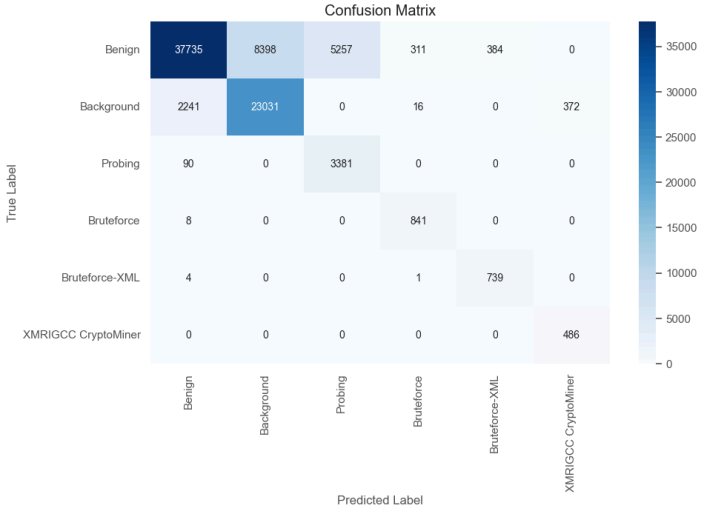
2.13.Error Analysis

```
Score: 0.8624576007844553
Balanaced Accuracy: 0.9299935376034866
Accuracy: 0.7949216639654241
                    precision    recall  f1-score   support

        Background       0.73      0.90      0.81     25660
            Benign       0.94      0.72      0.82     52085
       Bruteforce        0.72      0.99      0.83       849
    Bruteforce-XML       0.66      0.99      0.79       744
           Probing       0.39      0.97      0.56      3471
XMRIGCC CryptoMiner      0.57      1.00      0.72       486

          accuracy                           0.79     83295
         macro avg       0.67      0.93      0.76     83295
      weighted avg       0.85      0.79      0.80     83295
```

Gambar 28 Classification Report dan. Hasil Modeling

Berdasarkan hasil error analysis, terlihat bahwa model memiliki kelemahan dalam menangani kelas yang lebih jarang muncul, seperti Probing dan XMRIGCC CryptoMiner. Precision yang rendah pada kelas-kelas ini menunjukkan bahwa model sering salah dalam memprediksi traffic berbahaya ini, meskipun recall sangat tinggi. Ini menunjukkan bahwa model mampu mengenali sebagian besar traffic berbahaya, tetapi juga sering salah prediksi terhadap kelas-kelas lain, yang menyebabkan penurunan precision.

Ketidakseimbangan data menjadi salah satu penyebab utama masalah ini. Kelas yang lebih besar seperti Benign mendominasi prediksi, sementara kelas minor cenderung lebih sulit dikenali dengan tepat. Meskipun recall tinggi menunjukkan bahwa model tidak banyak melewatkan traffic berbahaya, rendahnya precision menunjukkan bahwa model tidak cukup spesifik dalam membedakan kelas-kelas tersebut.



Gambar 29. Evaluasi Hasil Model

Berdasarkan confusion matrix, salah satu hal yang menonjol adalah adanya kesalahan klasifikasi yang cukup signifikan antara kelas Benign dan Background. Model cenderung sering salah mengklasifikasikan traffic Benign sebagai Background dan sebaliknya. Ini menunjukkan bahwa ada kesamaan karakteristik antara kedua jenis traffic tersebut yang membuat model sulit membedakannya.

Untuk kelas yang lebih jarang seperti Probing dan Bruteforce, model memiliki performa yang sangat baik, dengan sedikit atau bahkan tidak ada kesalahan prediksi antara kelas-kelas tersebut. Namun, ada beberapa kesalahan klasifikasi minor, seperti Bruteforce-XML yang kadang diklasifikasikan sebagai Bruteforce.

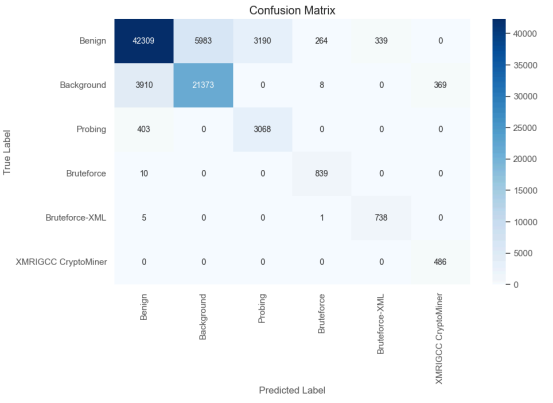
Pada kelas XMRIGCC CryptoMiner, tidak ada kesalahan prediksi, yang menunjukkan bahwa model cukup baik dalam mengenali serangan ini. Namun, untuk kelas Probing, meskipun recall tinggi, ada

kesalahan yang memprediksi traffic ini sebagai Benign, yang bisa mengakibatkan risiko tidak terdeteksinya traffic berbahaya.

2.14.Aplikasi Threshold Tuning dan Evaluasi

```
Best Thresholds: [0.37, 0.18, 0.75, 0.66, 0.66, 0.01]
Score: 0.8721754687399257
Balanced Accuracy: 0.9182149149094689
Accuracy: 0.8261360225703823
                    precision    recall  f1-score   support

        Background       0.78      0.83      0.81     25660
            Benign       0.91      0.81      0.86     52085
        Bruteforce       0.75      0.99      0.86       849
    Bruteforce-XML       0.69      0.99      0.81       744
           Probing       0.49      0.88      0.63      3471
XMRIGCC CryptoMiner       0.57      1.00      0.72       486

          accuracy                           0.83     83295
         macro avg       0.70      0.92      0.78     83295
      weighted avg       0.85      0.83      0.83     83295
```

Gambar 30. Evaluasi Model Tuning



Gambar 31. Evaluasi Hasil Model

Hasil penerapan threshold tuning menunjukkan peningkatan performa model secara signifikan, terutama dalam kemampuan mendeteksi kelas-kelas minoritas yang sebelumnya sulit diidentifikasi. Model menjadi lebih sensitif terhadap variasi jenis traffic berbahaya, mengurangi kesalahan klasifikasi pada serangan yang memiliki karakteristik mirip.

Peningkatan f1-score pada kelas tertentu menunjukkan bahwa model lebih akurat dalam membedakan antara traffic normal dan traffic berbahaya setelah tuning. Meski demikian, ada beberapa kelas yang performanya tidak berubah secara signifikan, menandakan bahwa model sudah cukup optimal untuk mendeteksi traffic tersebut sejak awal.

2.15.Future Improvements

Untuk peningkatan di masa depan, langkah pertama yang perlu dilakukan adalah memastikan kualitas data yang lebih baik, terutama dengan meminimalkan nilai NaN. Dengan data yang lebih bersih, berbagai eksperimen tambahan seperti undersampling, SMOTE, dan Tomek Link dapat dilakukan untuk menangani ketidakseimbangan kelas dengan lebih efektif. Selain itu, pengayaan dataset dengan fitur-fitur baru, seperti timestamp atau informasi spesifik tentang waktu serangan, dapat membantu memperbaiki pemisahan antar kelas.

Saat ini, model banyak bergantung pada host dan port, yang berarti jika ada data baru dengan kombinasi host dan port yang berbeda dari data yang dilatih, kemungkinan besar prediksi akan salah. Oleh karena itu, diperlukan eksplorasi fitur baru yang lebih bersifat generalizable, misalnya, pola trafik temporal, pola paket, atau indikator perilaku anomali lainnya, yang tidak hanya bergantung pada atribut statis seperti host dan port.

3. **Kesimpulan**

3.1. Kesimpulan

Pada penelitian ini, kami mengeksplorasi faktor-faktor yang mempengaruhi deteksi serangan pada traffic jaringan menggunakan teknik feature engineering dan beberapa model klasifikasi. Hasilnya menunjukkan bahwa model yang dibangun dengan CatBoost dan LightGBM mampu membedakan antara traffic normal dan berbahaya dengan performa yang cukup baik. Penggunaan class weight dan imputasi baseline memberikan peningkatan yang signifikan dalam custom scoring dibandingkan metode lainnya, terutama untuk kelas-kelas dengan jumlah data yang tidak seimbang.

Selanjutnya, dengan memanfaatkan threshold tuning, kami berhasil memaksimalkan prediksi untuk setiap kelas dengan akurasi yang lebih baik. Hal ini terlihat dari peningkatan custom scoring. Namun, masih ada kesulitan dalam membedakan kelas Probing dan CryptoMiner yang memerlukan pengembangan lebih lanjut.

Dengan menggunakan fitur-fitur yang terkait dengan host, port, serta dukungan dari fitur lain seperti flag, IAT, dan packets, model ini menunjukkan bahwa fitur tersebut sangat penting dalam memprediksi traffic jaringan. Namun, fitur-fitur yang memiliki importance rendah atau bernilai 0 menunjukkan bahwa mereka kurang berkontribusi pada performa model dan mungkin perlu diabaikan atau diganti di masa mendatang.

3.2. Insight

Beberapa insight yang dapat diambil dari penelitian ini adalah pentingnya targeted host dan port dalam mendeteksi serangan jaringan. Fitur seperti origin_host, origin_port, dan response_host menjadi indikator kuat untuk membedakan traffic berbahaya, terutama pada serangan yang terfokus pada host atau port tertentu. Misalnya, serangan Bruteforce dan CryptoMiner secara konsisten menyasar host dan port yang sama, sehingga fitur ini sangat membantu model dalam mengenali pola serangan yang berulang.

Selain itu, kombinasi fitur seperti flag, IAT, dan packet counts juga memberikan informasi tambahan yang penting, karena serangan sering kali menunjukkan pola komunikasi yang berbeda dibandingkan traffic normal.

Selain itu, penggunaan threshold tuning menunjukkan bahwa pengaturan ambang prediksi yang tepat dapat secara drastis meningkatkan performa, terutama pada kelas dengan jumlah data yang tidak seimbang. Implementasi class weight juga membantu memperbaiki bias pada kelas minoritas, yang sebelumnya sering terabaikan oleh model.

Daftar Pustaka

Brownlee, J. (2018). A Gentle Introduction to Threshold-Moving for Imbalanced Classification. Machine Learning Mastery. Diakses dari https://machinelearningmastery.com/

CatBoost. (2023). CatBoost: Gradient Boosting with Categorical Features Support. Diakses dari https://catboost.ai/

EditVerse. (2024). Big Data dalam Penelitian: Memanfaatkan Kumpulan Data Besar 2024. Diakses dari https://www.editverse.com/id/big-data-in-research-harnessing-the-power-of-large-datasets-in-2024/

Hosseini, M. (2019, August 6). Mastering logistic regression thresholds. Medium. Diakses dari https://morihosseini.medium.com/mastering-logistic-regression-thresholds-b34fe07f09f5

LightGBM. (2023). A fast, distributed, high performance gradient boosting (GBT, GBDT, GBRT, GBM or MART). Diakses dari https://lightgbm.readthedocs.io/

Rahman, M. A. (2018, September 20). How to deal with imbalanced data in classification. Medium. Diakses dari https://medium.com/game-of-bits/how-to-deal-with-imbalanced-data-in-classification-bd03cfc66066

Singh, S. (2020, October 14). Improve class imbalance using class weights. Analytics Vidhya. Diakses dari https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/

Venkat, P. (2021, January 27). Cross-validation techniques. Medium. Diakses dari https://medium.com/geekculture/cross-validation-techniques-33d389897878

Woolf, M. (2018). Mastering machine learning algorithms: Expert techniques for implementing popular machine learning algorithms and fine-tuning your models. O'Reilly Media. Diakses dari https://www.oreilly.com/library/view/mastering-machine-learning/9781788997409/b2ae63ec-1c6c-48be-a98b-508075e782e2.xhtml