OPTIMASI KLASIFIKASI MULTIKELAS JENIS SERANGAN JARINGAN IOT

Sunu Ilham Pradika[1], Hidayat Ramadhani[2], Imelda Imelda[3]
Program Studi Ilmu Komputer, Fakultas Teknologi Informasi
Universitas Budi Luhur[1,2,3]
Jl. Ciledug Raya, Petukangan Utara, Kec. Pesanggrahan, Kota Jakarta Selatan
sunuilhamp@gmail.com[1], hidayat.ramadhani1@gmail.com[2], imelda@budiluhur.ac.id[3]

Abstrak

Perkembangan *Internet of Things* (IoT) telah mendorong pemanfaatan perangkat terhubung pada berbagai sektor, mulai dari kesehatan, energi, transportasi, hingga lingkungan rumah tangga. Namun, karakteristik jaringan IoT yang heterogen, dinamis, dan banyak bergantung pada komunikasi nirkabel membuatnya rentan terhadap berbagai bentuk ancaman siber. Penelitian ini bertujuan membangun dan mengevaluasi model deteksi intrusi untuk klasifikasi multikelas pada jaringan IoT menggunakan dataset ToN_IoT. Model yang digunakan adalah Random Forest dan XGBoost dengan beberapa skenario pengujian, yaitu model dasar, model dengan *hyperparameter tuning*, serta penerapan SMOTENC pada XGBoost untuk menangani ketidakseimbangan kelas. Tahapan penelitian meliputi *exploratory data analysis*, *data cleaning, feature selection*, data splitting, *feature encoding, data balancing*, pemodelan, serta evaluasi menggunakan *accuracy, precision, recall, F1-score, training time*, dan *confusion matrix.* Hasil pengujian menunjukkan bahwa seluruh model menghasilkan performa tinggi dengan *accuracy* dan F1-score di atas 99%. Model terbaik diperoleh pada skenario XGBoost dengan SMOTENC, yaitu *accuracy* sebesar 99,59% dan F1-score sebesar 99,59%.

Kata Kunci: Internet of Things, XGBoost, SMOTENC, klasifikasi multikelas, ToN_IoT.

Abstract

The development of the Internet of Things has encouraged the use of connected devices across various sectors, including healthcare, energy, transportation, and household environments. However, the heterogeneous and dynamic nature of IoT networks, along with their reliance on wireless communication, makes them vulnerable to various cyber threats. This study aims to develop and evaluate an intrusion detection model for multiclass classification in IoT networks using the ToN_IoT dataset. The models used in this study are Random Forest and XGBoost, tested under several scenarios, including baseline models, models with hyperparameter tuning, and the application of SMOTENC to XGBoost to address class imbalance. The research stages include exploratory data analysis, data cleaning, feature selection, data splitting, feature encoding, data balancing, modeling, and evaluation using accuracy, precision, recall, F1-score, training time, and confusion matrix. The results show that all models achieved high performance, with accuracy and F1-score values above 99%. The best performance was obtained by XGBoost with SMOTENC, achieving an accuracy of 99.59% and an F1-score of 99.59%.

Keywords*: Internet of Things, XGBoost, SMOTENC, multiclass classification, ToN_IoT.*

PENDAHULUAN

Dalam beberapa tahun terakhir, *Internet of Things* (IoT) telah berkembang sebagai paradigma baru dan mendapatkan perhatian (Choudhary, 2024). IoT membuka ruang untuk berbagai inovasi teknologi berikutnya (Čolaković & Hadžialić, 2018). IoT dianggap sebagai bagian dari internet masa depan dan akan terdiri dari miliaran atau triliunan 'benda' cerdas yang saling berkomunikasi (Chanal & Kakkasageri, 2020; Li et al., 2015). IoT memungkinkan perangkat saling berinteraksi tanpa keterlibatan manusia secara langsung (Al-Fuqaha et al., 2015). Selain itu, IoT juga memberikan pendekatan yang lebih baik untuk menghubungkan dunia digital dengan dunia fisik (Laghari et al., 2024). Sehingga, saat ini IoT telah digunakan pada berbagai sektor seperti pertanian, peternakan, kesehatan, pendidikan, energi, dan transportasi bahkan di rumah-rumah. Salah satu karakteristik IoT adalah memanfaatkan teknologi nirkabel untuk berkomunikasi antar perangkat (Greengard, 2021; Hathaliya & Tanwar, 2020). Namun, di sisi yang lain risiko keamanan siber semakin meningkat (Jardine, 2018). Sifatnya yang heterogen dan dinamis, membuat perangkat IoT rentan terhadap berbagai jenis ancaman dan serangan keamanan (Gugueoth et al.,

2023). Berbeda dengan sistem teknologi informasi tradisional, ekosistem IoT terdiri atas perangkat yang beragam, mulai dari server yang memiliki spesifikasi tinggi hingga perangkat sensor berukuran kecil dengan sumber daya yang terbatas. Perangkat IoT seringkali memperoleh data pribadi pengguna dan menyimpannya secara *online,* hal ini membuat perangkat-perangkat yang saling terhubung tersebut sangat rentan terhadap ancaman yang dapat membahayakan keamanan dan privasi (Deep et al., 2022). Gangguan yang disebabkan karena adanya kegagalan dalam melakukan deteksi intrusi pada perangkat IoT dapat membahayakan ketika terjadi pada sektor strategis seperti kesehatan, energi, dan transportasi. Oleh karena itu diperlukan sebuah sistem deteksi intrusi yang dapat mencegah serangan pada jaringan IoT.

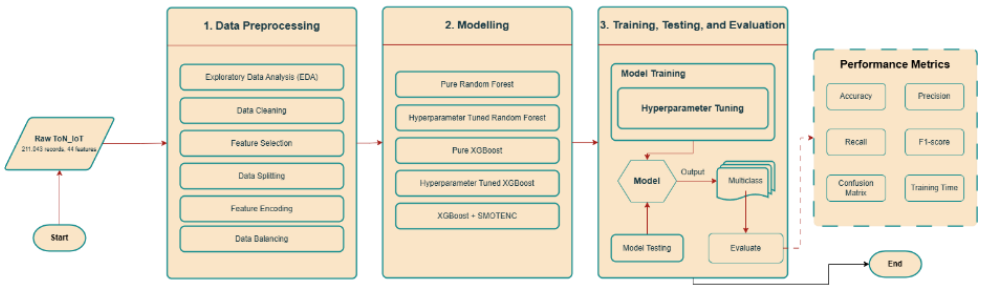
Berdasarkan uraian tersebut, perlu dibuat sistem deteksi intrusi untuk mengamankan jaringan IoT dengan memanfaatkan *machine learning*. Penelitian pada topik ini bukanlah hal baru, sehingga disini dilakukan pengembangan model untuk mengukur performa deteksi intrusi menggunakan XGBoost pada klasifikasi multikelas untuk jaringan IoT khususnya *dataset* ToN_IoT dibandingkan dengan Random Forest, serta bagaimana pengaruh *tuning hyperparameter* dan *oversampling Synthetic Minority Oversampling Technique for Nominal and Continuous features* (SMOTENC) terhadap peningkatan kinerja model. Penelitian ini bertujuan untuk membandingkan kedua algoritma, mengoptimalkan parameter model, serta mengevaluasi efektivitas SMOTENC yang diterapkan secara khusus pada XGBoost dalam menangani ketidakseimbangan kelas. Keterbaruan penelitian ini terletak pada pengembangan model dengan melakukan komparasi algoritma, optimasi *hyperparameter*, dan *oversampling* berbasis SMOTENC dalam konteks deteksi intrusi IoT multikelas, sehingga diharapkan dapat menghasilkan pendekatan yang lebih efektif dan konsisten dalam mengidentifikasi berbagai jenis serangan pada jaringan IoT.

PENELITIAN RELEVAN

Beberapa penelitian sebelumnya telah berupaya meningkatkan deteksi ancaman keamanan siber IoT menggunakan dataset ToN-IoT. Soni et al. membandingkan XGBoost dan LightGBM, dengan hasil terbaik diperoleh LightGBM sebesar 78% untuk akurasi dan F1-score (Soni et al., 2024). Gad et al. mengembangkan IDS terdistribusi berbasis *machine learning* dengan SMOTE serta seleksi fitur Chi2 dan *correlation matrix*, dan menunjukkan bahwa XGBoost mencapai performa terbaik dengan akurasi 98,2% dan F1-score 94,3% pada klasifikasi multikelas (Gad et al., 2022). Félix et al. menggunakan LSTM-Autoencoder untuk deteksi anomali dan *zero-day attack* pada lingkungan Edge-IoT, dengan akurasi 97,2% dan F1-score 96% (Félix et al., 2025). Sementara itu, Dharini et al. mengusulkan *hybrid feature selection* menghasilkan akurasi hingga 98,6% pada klasifikasi biner dan 97,2% pada klasifikasi multikelas menggunakan Decision Tree dan KNN (Dharini et al., 2026).

METODE PENELITIAN

Metodologi penelitian ini sebagaimana ditunjukkan pada Gambar 1 terdiri atas empat tahapan utama, yaitu *data preprocessing, modelling,* dan *training-testing-evaluation*. Penelitian menggunakan dataset ToN_IoT. Pada tahap *preprocessing* dilakukan EDA, pembersihan data, seleksi fitur, pembagian data, *encoding*, dan penyeimbangan kelas. Tahap *modelling* membandingkan Random Forest dan XGBoost dalam kondisi dasar maupun setelah *tuning*, serta skenario XGBoost dengan SMOTENC. Selanjutnya, model dievaluasi untuk klasifikasi multikelas menggunakan metrik *accuracy, precision, recall, F1-score*, *confusion matrix*, dan *training time*.



Gambar 1. Metodologi Penelitian

Dataset

Dataset ToN_IoT merupakan *dataset* keamanan siber yang dikembangkan untuk mendukung evaluasi sistem keamanan berbasis AI, yang menggambarkan jaringan IoT pada dunia nyata dan skenario ancaman siber yang kompleks yang berasal darai *Telemetry data, Operating systems logs,* dan *Network traffic* (Moustafa, 2021). *Dataset* ToN_IoT merupakan *dataset open source* IoT/IIoT yang dirancang untuk evaluasi sistem deteksi intrusi (Alsaedi et al., 2020; Booij et al., 2022). Terdapat sepuluh kelas, yaitu sembilan kelas serangan dan satu kelas normal, 9 kelas serangan yaitu *scanning, backdoor, ransomware, Denial of Service* (DoS), *Distributed Denial of Service* (DDoS), *Man-in-the-Middle* (MITM), *data injection*, *Cross-Site Scripting* (XSS), dan *password violations* (Latif et al., 2022). Selain itu, terdapat 44 (empat puluh empat) fitur dengan 211.043 baris data.

Data Preprocessing

Pada tahap *data preprocessing*, penelitian diawali dengan *Exploratory Data Analysis* (EDA) untuk memahami pola dan anomali, serta mengungkap struktur dan hubungan dalam data (Simangunsong et al., 2024). Setelah itu dilakukan *data cleaning* untuk menangani *missing value, outlier,* dan ketidaklengkapan data agar kualitas data menjadi lebih layak untuk proses pemodelan (Van den Broeck et al., 2005).

Tahap berikutnya adalah *feature selection*, yaitu memilih fitur yang paling relevan dan menghindari redundansi data (Dhanvijay et al., 2025). Pada tahap *feature selection,* penelitian ini mengevaluasi dan menyeleksi kolom kategorikal dengan kardinalitas tinggi karena fitur-fitur tersebut cenderung menghasilkan representasi tidak berurutan dan bernilai unik seperti hasil *hash* ID (Pargent et al., 2022), terutama setelah proses *encoding* (M et al., 2025), serta dapat meningkatkan kompleksitas model dan risiko *overfitting* (Richman & Wuthrich, 2023).

Selanjutnya, dataset dibagi ke dalam data latih dan data uji sejak awal alur pemrosesan untuk mencegah *data leakage*, yaitu kondisi ketika informasi dari data uji ikut terbawa ke proses pelatihan dan menyebabkan hasil evaluasi menjadi terlalu baik (Bouke & Abdullah, 2023). Umumnya, pembagian data dilakukan dengan proporsi 80:20, artinya 80% untuk data latih dan 20% untuk data uji (Joseph, 2022; Sivakumar et al., 2024).

Pada atribut kategorikal, dilakukan *feature encoding* agar data non-numerik dapat direpresentasikan dalam bentuk yang dapat diproses oleh algoritma *machine learning* (Anitha et al., 2025).

Terakhir, *data balancing* diterapkan untuk mengatasi ketimpangan distribusi kelas, karena ketidakseimbangan kelas dapat menurunkan kemampuan model menjadikannya *overfitting* (Mujahid et al., 2024). Karena data penelitian ini memuat fitur numerik dan kategorikal, pendekatan *oversampling* yang digunakan adalah SMOTENC yang merupakan adaptasi dari SMOTE yang memang dirancang untuk menangani dataset campuran nominal dan kontinu (Fonseca & Bacao, 2023). Sehingga, penggunaan SMOTENC bukan tanpa alasan melainkan karena kondisi *dataset* yang masih tercampur antara data numerik dan kategorikal.

Modelling

Breiman menjelaskan Random Forest merupakan metode ensemble yang membentuk banyak decision tree dan menentukan hasil klasifikasi berdasarkan mekanisme voting dari seluruh pohon. Secara teoretis, Breiman mendefinisikan Random Forest sebagai kumpulan classifier berbentuk pohon, dengan setiap pohon dibangun menggunakan unsur acak yang terdistribusi identik dan saling independen (Breiman, 2001). Untuk menjelaskan performanya, Breiman memperkenalkan margin function sebagai selisih antara rata-rata voting untuk kelas yang benar dan rata-rata suara tertinggi untuk kelas lain dengan formula:

$$mg(X,Y) = 1/K \sum\_(k = 1)\text{^}K\ I(h\_k\ (X) = Y) - max/(j \neq Y)\ \ 1/K \sum\_(k = 1)\text{^}K\ i(h\_k\ (X) = j) \quad (1)$$

Margin pada Persamaan (1) menunjukkan selisih antara proporsi suara yang diberikan oleh seluruh pohon untuk kelas yang benar dengan proporsi suara terbesar yang diberikan untuk kelas lain. Secara matematis, jika terdapat K pohon dalam forest, maka setiap pohon $h\_k\ (X)$ akan menghasilkan prediksi kelas untuk data X. Fungsi indikator $I(h\_k\ (X) = Y)$ bernilai 1 apabila

pohon ke-k memprediksi kelas yang benar, yaitu Y, dan bernilai 0 apabila prediksinya salah. Dengan demikian, suku pertama dalam persamaan tersebut merepresentasikan rata-rata suara pohon yang memilih kelas benar.

$$PE\hat{} *= P\_(X,Y)\ (mg(X,Y) < 0) \tag{2}$$

Generalization error pada Random Forest didefinisikan sebagai peluang margin bernilai negatif, yaitu kondisi ketika model memberi suara mayoritas pada kelas yang salah untuk data baru. Semakin kecil peluang tersebut, semakin baik kemampuan generalisasi model. Menurut Breiman, performa Random Forest dipengaruhi oleh *strength* dan *correlation* di mana model akan bekerja baik jika setiap pohon memiliki kemampuan prediksi tinggi, tetapi korelasi antar pohon rendah.
XGBoost merupakan pengembangan dari *gradient tree boosting* yang memodelkan prediksi sebagai penjumlahan aditif dari sejumlah *regression tree* (Chen & Guestrin, 2016). Chen dan Guestrin merumuskan prediksi model sebagai berikut:

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F}, \tag{3}$$

Persamaan (3) menunjukkan bahwa prediksi untuk data ke-i, yaitu $(y\_i\ )\hat{}$, diperoleh dengan menjumlahkan keluaran dari K pohon, yaitu $f\_k\ [\![(x)]\!]\_i)$. Setiap fungsi f_k merupakan sebuah pohon yang termasuk dalam ruang fungsi $\mathcal{F}$. Dengan kata lain, XGBoost tidak membangun satu pohon besar, tetapi membangun banyak pohon secara bertahap, di mana setiap pohon baru ditambahkan untuk memperbaiki kesalahan prediksi dari model sebelumnya.

$$\begin{aligned} \mathcal{L}(\phi) &= \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \\ \text{where } \Omega(f) &= \gamma T + \frac{1}{2}\lambda \|w\|^2 \end{aligned} \tag{4}$$

Proses pembelajaran pada XGBoost dilakukan dengan meminimalkan fungsi objektif yang ditunjukkan pada Persamaan (4). Fungsi objektif tersebut terdiri dari dua komponen utama, yaitu *loss function* dan *regularization term*. Komponen pertama, yaitu $\sum\_i\ l((y\_i\ )\hat{}, y\_i)$, mengukur besarnya kesalahan antara nilai prediksi $(y\_i\ )\hat{}$ dan nilai sebenarnya y_i. Semakin kecil nilai loss, semakin baik prediksi model terhadap data pelatihan. Komponen kedua, yaitu $\sum\_k\ \Omega(f\_k)$, berfungsi sebagai penalti terhadap kompleksitas model. Regularisasi ini penting agar model tidak terlalu kompleks dan tidak hanya menghafal data pelatihan. Bentuk regularisasi $\Omega(f)$ terdiri atas dua bagian, yaitu γT dan $1/2\ \lambda|(|w|)|$. Parameter T menyatakan jumlah *leaf* dalam sebuah pohon, sedangkan w menyatakan bobot pada *leaf* tersebut. Nilai γ digunakan untuk memberikan penalti terhadap penambahan jumlah *leaf*, sehingga pohon yang terlalu kompleks akan dihindari. Sementara itu, parameter λ berfungsi mengontrol besar kecilnya bobot *leaf*. Semakin besar nilai λ, semakin kuat pembatasan terhadap bobot *leaf*, sehingga model menjadi lebih stabil dan risiko *overfitting* dapat dikurangi.

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}, \tag{5}$$

Dalam proses optimasinya, XGBoost menggunakan pendekatan orde kedua dengan memanfaatkan informasi gradien dan hessian dari fungsi *loss*. Gradien g_i menunjukkan arah dan besar kesalahan prediksi yang perlu diperbaiki, sedangkan hessian h_i memberikan informasi mengenai kelengkungan fungsi loss sehingga proses pembaruan model menjadi lebih akurat. Bobot optimal untuk leaf ke-j, sebagaimana ditunjukkan pada persamaan bobot *leaf*, dihitung berdasarkan jumlah gradien dan jumlah hessian dari seluruh data yang masuk ke *leaf* tersebut.

Training, Testing, and Evaluation

Pada tahap *training, testing,* and *evaluation*, seluruh model yang telah dirancang pada tahap sebelumnya dilatih menggunakan data latih untuk menyelesaikan tugas klasifikasi multikelas. Proses pelatihan dilakukan pada setiap skenario model, baik model dasar maupun model yang telah dioptimasi melalui hyperparameter tuning, untuk memperoleh konfigurasi terbaik yang mampu meningkatkan performa klasifikasi. Pada tahap tuning, beberapa parameter penting pada algoritma Random Forest dan XGBoost dioptimasi untuk mendapatkan kombinasi parameter yang paling efektif.

Hyperparameter merupakan parameter yang mengendalikan proses pembelajaran algoritma, berbeda dari parameter model yang dipelajari langsung dari data. *Hyperparameter* yang dipilih bisa berbeda antar *dataset* (Bergstra & Bengio, 2012).

Setelah proses pelatihan selesai, model diuji menggunakan data uji yang tidak terlibat dalam proses *training*, sehingga performa yang diperoleh dapat merepresentasikan kemampuan generalisasi model terhadap data baru.

Performance Metrics

Hasil prediksi model kemudian dievaluasi menggunakan beberapa metrik, yaitu *accuracy, precision, recall, dan F1-score* untuk mengukur kualitas model (Sokolova & Lapalme, 2009). Khusus untuk multikelas dengan karakteristik data yang tidak seimbang menggunakan formula berikut (Mortaz, 2020):

Akurasi

$$c\_(i.) = \sum\_(j = 1)\^k c\_ij \ \ \forall \in \{1, \ldots, k\} \quad (6)$$

$$c\_(.i) = \sum\_(j = 1)\^k [\![c\_ji \ \ \forall \in \{1, \ldots, k\}]\!]\_ \quad (7)$$

$$\mathrm{ACC} = \frac{\sum_i c_{ii}}{\sum_{ij} c_{ij}} \quad (8)$$

Pada Persamaan 6, 7, dan 8 Evaluasi model dilakukan menggunakan confusion matrix multikelas, dengan c_ij menyatakan jumlah data yang kelas sebenarnya adalah i, tetapi diprediksi sebagai kelas j. Nilai $c\_(i.)$ menunjukkan jumlah seluruh data pada kelas sebenarnya ke-i, sedangkan $c\_(.i)$ menunjukkan jumlah seluruh data yang diprediksi sebagai kelas ke-i. Berdasarkan komponen tersebut, *accuracy* dihitung sebagai perbandingan antara jumlah prediksi benar, yaitu elemen diagonal $\_ii$, terhadap seluruh data yang diuji.

Macro Average Precision

$$\mathrm{MAP} = 1/k \sum_{i=1}^{k} \frac{c_{ii}}{c_{.i}} \quad (9)$$

Macro average precision menghitung rata-rata precision dari seluruh kelas sebagaimana ditunjukkan pada Persamaan (9). *Precision* pada suatu kelas menunjukkan seberapa banyak data yang diprediksi sebagai kelas tersebut benar-benar berasal dari kelas yang sesuai. Dengan kata lain, *precision* mengukur ketepatan model ketika memberikan prediksi pada suatu kelas.

Macro Average Recall

$$\mathrm{MAR} = 1/k \sum_{i=1}^{k} \frac{c_{ii}}{c_{i.}} \quad (10)$$

Macro average recall menghitung rata-rata *recall* dari seluruh kelas, dengan *recall* menunjukkan kemampuan model dalam mengenali seluruh data yang sebenarnya termasuk dalam suatu kelas. *Recall* yang tinggi berarti model mampu menemukan sebagian besar data pada kelas tersebut.

F1-*Score*

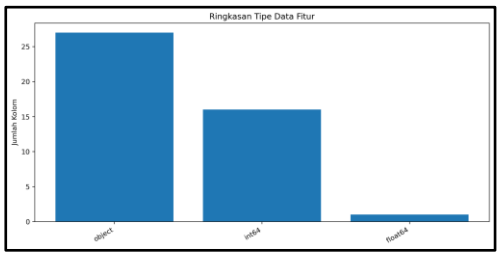
$$\mathrm{F} = 1/k \sum_{i=1}^{k} \frac{2 \frac{c_{ii}}{c_{.i}} \frac{c_{ii}}{c_{i.}}}{\frac{c_{ii}}{c_{.i}} + \frac{c_{ii}}{c_{i.}}} \quad (11)$$

Selanjutnya, F1-score digunakan untuk menggabungkan *precision* dan *recall* ke dalam satu nilai evaluasi. F1-score merupakan rata-rata harmonik antara *precision* dan *recall*, sehingga metrik ini memberikan gambaran keseimbangan antara ketepatan prediksi dan kemampuan model dalam

mengenali data pada setiap kelas. Penggunaan *macro average* pada F1-score bertujuan agar performa model terhadap seluruh kelas, termasuk kelas dengan jumlah data yang lebih sedikit, tetap diperhitungkan secara adil.

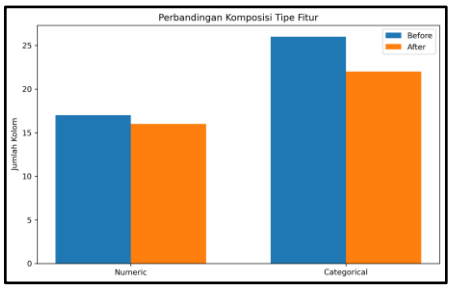
HASIL DAN PEMBAHASAN

Penelitian ini diawali dengan penggunaan *dataset* ToN_IoT sebagai sumber data, kemudian dilanjutkan dengan tahap data preprocessing yang meliputi *Exploratory Data Analysis* (EDA), *data cleaning, feature selection, data splitting, feature encoding,* dan *data balancing* untuk menyiapkan data sebelum pemodelan.



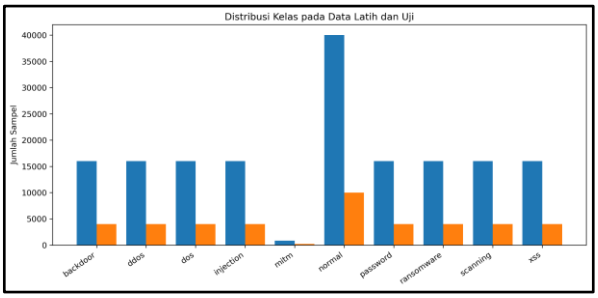
Gambar 1. Tipe data tiap fitur.

Pada Gambar 1 menunjukkan EDA bahwa tipe data yang ada pada *dataset* didominasi oleh data bertipe *object*, diikuti *integer*, dan terakhir *float*.



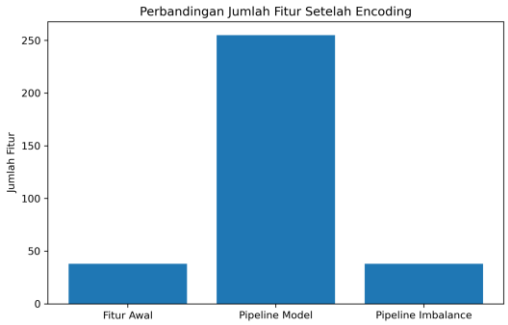
Gambar 2. Perbandingan data *numeric* dan *categoric* sebelum dan setelah *data cleaning* dan *feature selection*.

Gambar 2 menunjukkan hasil dari *data cleaning* dan *feature selection* yakni penghapusan kolom kategorikal dengan kardinalitas tinggi yaitu *'http_uri', 'http_user_agent', 'label', 'ssl_issuer', 'ssl_subject'*. Sehingga, dari 44 (empat puluh empat) kolom awal setelah dihapus tersisa 39 (tiga puluh sembilan) kolom saja.



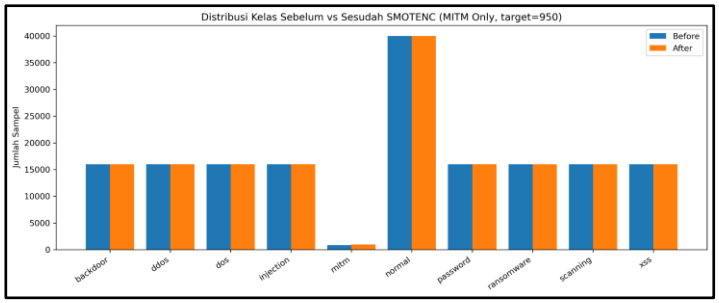
Gambar 3. Data splitting.

Gambar 3 menunjukkan hasil dari *data splitting* untuk seluruh kelas yang ada yaitu *backdoor, ddos, dos, injection, mitm, normal, password, ransomware, scanning,* dan *xss* di mana data dibagi dengan rasio 80:20, yaitu 80% digunakan sebagai data latih dan 20% data uji.



Gambar 4. Data encoding.

Gambar 4 menunjukkan hasil dari *feature encoding*, hasilnya menunjukkan bahwa jumlah fitur meningkat secara signifikan pada *pipeline* model dibandingkan fitur awal. Hal ini disebabkan oleh penggunaan OneHotEncoder pada fitur kategorikal, yang mengubah setiap kategori menjadi fitur biner terpisah. Sebaliknya, pada *pipeline* untuk penanganan ketidakseimbangan kelas menggunakan SMOTENC, fitur kategorikal dikodekan menggunakan OrdinalEncoder sehingga jumlah fitur tetap sama seperti fitur awal. Perbedaan ini menunjukkan adanya *trade-off* antara representasi fitur yang lebih informatif dengan kompleksitas dimensi data. Gambar 5 menunjukkan hasil di mana perbandingan terhadap fitur minoritas yaitu mitm, dilakukan *oversampling* menggunakan SMOTENC dengan target yang telah ditentukan yaitu 950.



Gambar 5. Oversampling SMOTENC.

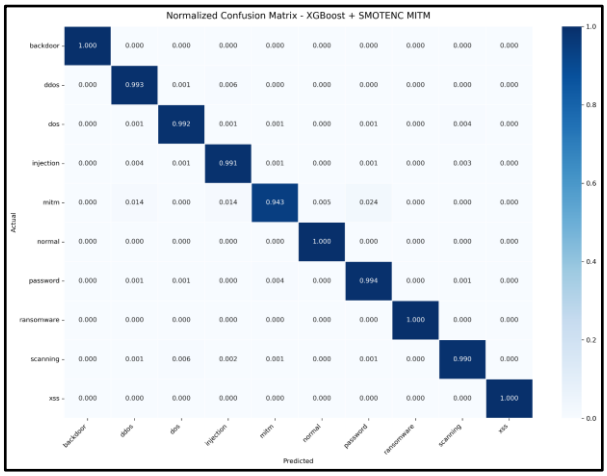
Selanjutnya, pada tahap modelling, penelitian membangun lima skenario model, yaitu Pure Random Forest, Hyperparameter Tuned Random Forest, Pure XGBoost, Hyperparameter Tuned XGBoost, dan XGBoost + SMOTENC. Pada tahap *training, testing, and evaluation*, setiap model dilatih pada data latih, dioptimasi melalui *hyperparameter tuning*, lalu diuji pada data uji dalam skenario klasifikasi multikelas. Performa model kemudian dievaluasi menggunakan *accuracy, precision, recall, F1-score, training time, dan confusion matrix.*

Tabel 1. Hasil model pada data uji

No	Fitur	Accuracy	Precision	Recall	F1-score	Training Time
1	Pure Random Forest	99.48%	98.3%	99%	99.48%	68.18s
2	Hyperparameter Tuned RF	99.50%	98.50%	98.98%	99.51%	59.20s
3	Pure XGBoost	99.58%	98.37%	98.87%	99.58%	15.46s
4	Hyperparameter Tuned XGBoost	99.56%	98.35%	98.94%	99.56%	30.49s
5	**XGBoost + SMOTENC**	**99.59%**	**98.39%**	**99%**	**99.59%**	**36.88s**

Tabel 1 menunjukkan bahwa seluruh model memiliki performa yang sangat tinggi, dengan nilai akurasi dan F1-score di atas 99%. Meskipun selisih kinerjanya relatif kecil, tetap terlihat adanya peningkatan pada beberapa pendekatan yang digunakan. Model Pure Random Forest memperoleh akurasi dan F1-score sebesar 99,48%, lalu meningkat menjadi 99,50% dan 99,51% setelah dilakukan tuning hyperparameter. Pada sisi lain, Pure XGBoost menunjukkan performa yang lebih baik dengan akurasi dan F1-score sebesar 99,58%, sedangkan hasil tuning hyperparameter pada XGBoost sedikit menurun menjadi 99,56%. Performa terbaik diperoleh oleh model XGBoost +

SMOTENC dengan akurasi dan F1-score sebesar 99,59%. Hasil ini menunjukkan bahwa XGBoost lebih unggul dibandingkan Random Forest pada dataset yang digunakan, dan penerapan SMOTENC memberikan tambahan peningkatan performa dalam klasifikasi multikelas.



Gambar 6. *Confusion matrix model* XGBoost + SMOTENC

Gambar 6 menunjukkan hasil confusion matrix yang telah dinormalisasi agar lebih sederhana. Berdasarkan confusion matrix yang telah dinormalisasi, model XGBoost + SMOTENC menunjukkan performa klasifikasi yang sangat baik pada hampir seluruh kelas, yang terlihat dari dominasi nilai diagonal mendekati 1.000. Kelas *backdoor, normal, ransomware,* dan *xss* terklasifikasi sempurna, sedangkan kelas lain seperti *ddos, dos, injection, password,* dan *scanning* juga memiliki tingkat klasifikasi yang sangat tinggi. Kesalahan prediksi relatif kecil dan tersebar tipis pada beberapa kelas. Kelas mitm menjadi kelas yang paling menantang dengan nilai diagonal 0.943 akibat kurang beragamnya data latih meski telah dilakukan *oversampling*, yang menunjukkan masih adanya kemiripan pola dengan beberapa kelas lain. Secara keseluruhan, hasil ini menegaskan bahwa model mampu melakukan klasifikasi multikelas secara sangat akurat dan stabil meski terdapat catatan pada kelas mitm.

Berdasarkan Tabel 2, model yang diusulkan pada penelitian ini menunjukkan performa yang sangat kompetitif dibandingkan penelitian-penelitian sebelumnya pada dataset ToN_IoT. Nilai *accuracy* dan F1-score yang diperoleh masing-masing sebesar 99,59%, sehingga menjadi hasil tertinggi di antara studi yang dibandingkan. Jika dibandingkan dengan penelitian lain, sebagian besar studi sebelumnya masih berada pada rentang performa 74% hingga 99,17% untuk *accuracy* maupun F1-score. Hasil penelitian ini melampaui Latif et al. yang sebelumnya memperoleh performa tinggi dengan *accuracy* 99,05% dan F1-score 99,17%, serta lebih baik dibandingkan Gad et al. dengan accuracy 98,3% dan F1-score 94,9%. Dengan demikian, hasil ini menunjukkan bahwa pendekatan yang digunakan dalam penelitian ini mampu memberikan peningkatan kinerja klasifikasi yang lebih baik pada skenario deteksi intrusi multikelas berbasis ToN_IoT.

Tabel 2. Perbandingan dengan penelitian sebelumnya pada ToN IoT

No	Researcher	Year	Accuracy	F1-score
1	Soni et al.	2024	74%	74%
2	(Kasina et al., 2026)	2026	83.82%	85.18%
3	Dharini et al.	2026	97.2%	91.8%
4	Félix et al.	2025	97.2%	96%
5	Gad et al.	2022	98.3%	94.9%
6	**Kami**	**2026**	**99.59%**	**99.59%**

SIMPULAN

Penelitian ini menunjukkan bahwa pendekatan berbasis machine learning mampu memberikan kinerja yang sangat kuat untuk deteksi intrusi multikelas pada dataset ToN_IoT. Dari lima skenario yang diuji, model XGBoost + SMOTENC menghasilkan performa terbaik dengan accuracy dan F1-score sebesar 99,59%, mengungguli Random Forest maupun XGBoost pada konfigurasi lainnya.

Hasil ini menegaskan bahwa XGBoost memiliki kemampuan representasi yang lebih baik terhadap pola data jaringan IoT, sementara SMOTENC efektif dalam menangani ketidakseimbangan kelas sehingga meningkatkan kemampuan model dalam mengenali kelas minoritas.

Sebagai arah penelitian selanjutnya, pengembangan dapat difokuskan pada validasi lintas dataset untuk menguji *robustnes model*, eliminasi atau generalisasi fitur yang berpotensi bias seperti alamat IP dan *port*, serta eksplorasi model *lightweight* berbasis *edge computing* untuk deteksi intrusi secara real-time. Selain itu, integrasi deep learning atau hybrid model dapat diteliti untuk meningkatkan deteksi kelas kompleks seperti mitm, sementara pendekatan explainable AI dapat digunakan untuk memperjelas perilaku model.

DAFTAR PUSTAKA

Choudhary, A. (2024). Internet of Things: a comprehensive overview, architectures, applications, simulation tools, challenges and future directions. *Discover Internet of Things*, *4*(1), 31.

Čolaković, A., & Hadžialić, M. (2018). Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. *Computer Networks*, *144*, 17–39.

Chanal, P. M., & Kakkasageri, M. S. (2020). Security and Privacy in IoT: A Survey. *Wireless Personal Communications*, *115*(2), 1667–1693.

Li, S., Xu, L. Da, & Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, *17*(2), 243–259.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, *17*(4), 2347–2376.

Laghari, A. A., Li, H., Khan, A. A., Shoulin, Y., Karim, S., & Khani, M. A. K. (2024). Internet of Things (IoT) applications security trends and challenges. *Discover Internet of Things*, *4*(1), 36.

Greengard, S. (2021). *The Internet of Things, revised and updated edition*. MIT Press.

Hathaliya, J. J., & Tanwar, S. (2020). An exhaustive survey on security and privacy issues in Healthcare 4.0. *Computer Communications*, *153*, 311–335.

Jardine, E. (2018). Mind the denominator: towards a more effective measurement system for cybersecurity. *Journal of Cyber Policy*, *3*(1), 116–139.

Gugueoth, V., Safavat, S., & Shetty, S. (2023). Security of Internet of Things (IoT) using federated learning and deep learning — Recent advancements, issues and prospects. *ICT Express*, *9*(5),

Deep, S., Zheng, X., Jolfaei, A., Yu, D., Ostovari, P., & Kashif Bashir, A. (2022). A survey of security and privacy issues in the Internet of Things from the layered context. *Transactions on Emerging Telecommunications Technologies*, *33*(6).

Soni, S., Remli, M. A., Daud, K. M., & Al Amien, J. (2024). Performance evaluation of multiclass classification models for ToN-IoT network device datasets. *Indonesian Journal of Electrical Engineering and Computer Science*, *35*(1), 485.

Gad, A. R., Haggag, M., Nashat, A. A., & Barakat, T. M. (2022). A Distributed Intrusion Detection System using Machine Learning for IoT based on ToN-IoT Dataset. *International Journal of Advanced Computer Science and Applications*, *13*(6).

Félix, B. A. H., Victoire, K. E., Placide, N. N. C., Pacôme, B., & Pascal, A. O. (2025). Intrusion Detection for Edge-IoT Using LSTM-Autoencoder. *Open Journal of Applied Sciences*, *15*(09), 2638–2647.

Dharini, N., Janani, V. S., & Katiravan, J. (2026). Efficient detection of intrusions in TON-IoT dataset using hybrid feature selection approach. *Scientific Reports*, *16*(1), 7763.

Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustainable Cities and Society*, *72*, 102994.

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. *IEEE Access*, *8*, 165130–165150.

Booij, T. M., Chiscop, I., Meeuwissen, E., Moustafa, N., & Hartog, F. T. H. den. (2022). ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Data Sets. *IEEE Internet of Things Journal*, *9*(1), 485–496.

Latif, S., Huma, Z. e, Jamal, S. S., Ahmed, F., Ahmad, J., Zahid, A., Dashtipour, K., Aftab, M. U., Ahmad, M., & Abbasi, Q. H. (2022). Intrusion Detection Framework for the Internet of Things Using a Dense Random Neural Network. *IEEE Transactions on Industrial Informatics*, *18*(9), 6435–6444.

Simangunsong, J., Simanjuntak, M. S., & Simanjuntak, N. D. (2024). Mental disorder classification with exploratory data analysis (EDA). *Journal of Intelligent Decision Support System (IDSS)*, *7*(3), 210–217.

Van den Broeck, J., Argeseanu Cunningham, S., Eeckels, R., & Herbst, K. (2005). Data Cleaning: Detecting, Diagnosing, and Editing Data Abnormalities. *PLoS Medicine*, *2*(10), e267.

Dhanvijay, D. M., Dhanvijay, M. M., & Kamble, V. H. (2025). Cyber intrusion detection using ensemble of deep learning with prediction scoring based optimized feature sets for IOT networks. *Cyber Security and Applications*, *3*, 100088.

Pargent, F., Pfisterer, F., Thomas, J., & Bischl, B. (2022). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics*, *37*(5), 2671–2692.

M, A., Savarimuthu, N., & Bhanu, S. M. S. (2025). WoEEE: a hybrid approach for enhancement of categorical data transformation. *International Journal of Data Science and Analytics*, *20*(7), 6635–6663.

Richman, R., & Wuthrich, M. V. (2023). High-Cardinality Categorical Covariates in Network Regressions. *SSRN Electronic Journal*.

Bouke, M. A., & Abdullah, A. (2023). An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability. *Expert Systems with Applications*, *230*, 120715.

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*(4), 531–538.

Sivakumar, M., Parthasarathy, S., & Padmapriya, T. (2024). Trade-off between training and testing ratio in machine learning for medical image processing. *PeerJ Computer Science*, *10*, e2245.

Anitha, M., Savarimuthu, N., & Bhanu, S. M. S. (2025). Chi-Square Target Encoding for Categorical Data Representation: A Real-World Sensor Data Case Study. *SN Computer Science*, *6*(3), 228.

Mujahid, M., Kına, E., Rustam, F., Villar, M. G., Alvarado, E. S., De La Torre Diez, I., & Ashraf, I. (2024). Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering. *Journal of Big Data*, *11*(1), 87.

Fonseca, J., & Bacao, F. (2023). Geometric SMOTE for imbalanced datasets with nominal and continuous features. *Expert Systems with Applications*, *234*, 121053.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32.

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, *13*, 281–305.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437.

Mortaz, E. (2020). Imbalance accuracy metric for model selection in multi-class imbalance classification problems. *Knowledge-Based Systems*, *210*, 106490.

Kasina, D., Reddy, V. S., Rajamanickam, S., Jose, J., & Palani, B. (2026). A novel intrusion detection system for class imbalance datasets using hybrid sampling with deep learning techniques. *Information Sciences*, *741*, 123208.