JURNAL ILMU KOMPUTER
Volume 2 Nomor 2, Desember, 2024
ISSN: 3031-125X (ONLINE)
Penerbit: Program Studi Teknik Informatika S-2 Universitas Pamulang

Analisis Kinerja Sistem Deteksi Intrusi Jaringan Internet Of Things Berbasis Metode Ensemble

Eko Kristianto[1], A.A. Waskita [2,], Thoyyibah.T[3]

[1,2,3)] Teknik Informatika, Program Pascasarjana, Universitas Pamulang, Kota Tangerang Selatan, Banten

Email: [1] ekotian23@gmail.com, [2] aawaskita@gmail.com, [3] dosen01116@unpam.ac.id

ABSTRACT

Network intrusion has rapidly evolved, posing significant risks to IT infrastructure. To address this, ensemble learning, known for its robust classification capabilities, is applied to IoT network traffic using the public RT_IOT2022 dataset. Models such as CatBoost, Extreme Gradient Boost (XGBoost), and LightGBM were developed and evaluated. The dataset was normalized using the Normalizer and MinMaxScaler functions from the scikit-learn framework. Model training was conducted with an 80:20 fixed data split for training and testing, along with 5-fold cross-validation. Testing revealed that XGBoost with MinMaxScaler and the 80:20 split achieved the highest accuracy of 99.89%. However, accuracy decreased to 94.04% when using 5-fold cross-validation. Nevertheless, XGBoost with MinMaxScaler consistently demonstrated the fastest computation time across all schemes. For instance, it required only 15 seconds for the fixed split scheme compared to 59 seconds for 5-fold cross-validation. These findings highlight the efficiency and accuracy of XGBoost when combined with MinMaxScaler under specific validation schemes.

Keywords: DDoS Attack Analysis, Ensemble, Decision Tree, RT_IOT2022

ABSTRAK

Intrusi jaringan telah berkembang pesat, menimbulkan risiko signifikan terhadap infrastruktur TI. Untuk mengatasinya, *ensemble learning*, yang dikenal dengan kemampuannya dalam klasifikasi yang kuat, diterapkan pada lalu lintas jaringan IoT menggunakan dataset publik RT_IOT2022. Model seperti CatBoost, Extreme Gradient Boost (XGBoost), dan LightGBM dikembangkan dan dievaluasi. Dataset dinormalisasi menggunakan fungsi Normalizer dan MinMaxScaler dari framework scikit-learn. Pelatihan model dilakukan dengan pembagian data tetap 80:20 untuk *training* dan *testing*, serta validasi silang 5-lipat (*5-fold cross-validation*). Pengujian menunjukkan bahwa XGBoost dengan MinMaxScaler dan pembagian tetap 80:20 menghasilkan akurasi tertinggi sebesar 99,89%. Namun, akurasi menurun menjadi 94,04% ketika menggunakan validasi silang 5-lipat. Meskipun demikian, XGBoost dengan MinMaxScaler secara konsisten menunjukkan waktu komputasi tercepat di antara semua skema. Sebagai contoh, hanya memerlukan waktu 15 detik untuk skema pembagian tetap dibandingkan 59 detik untuk validasi silang 5-lipat. Temuan ini menunjukkan efisiensi dan akurasi XGBoost ketika dikombinasikan dengan MinMaxScaler pada skema validasi tertentu**.**

Kata kunci: Analisis Serangan *DDoS, Ensemble, Decision Tree, RT_IOT2022*

1. Pendahuluan

Indonesia merupakan salah satu negara dengan pengguna internet terbesar di dunia. Berdasarkan laporan *We Are Social*, hingga Januari 2023, jumlah pengguna internet di Indonesia mencapai 212,9 juta. Artinya, sekitar 77% penduduk Indonesia telah menggunakan internet. Jumlah pengguna internet pada Januari 2023 meningkat sebesar

3,85% *year-on-year*. Pada Januari 2022, jumlah pengguna internet di Indonesia mencapai 205 juta jiwa.

Distributed Denial of Service (*DDoS*) adalah serangan sederhana namun efektif menghabiskan sumber daya *server* [1]. Penyerang mengalirkan lalu lintas internet ke *server*, layanan, atau jaringan untuk mencegah pengguna sah mengaksesnya. Serangan ini dirancang untuk menghancurkan target dengan banyak permintaan dalam jumlah besar, menghabiskan sumber dayanya, dan membuatnya tidak dapat diakses. Serangan *Distributed Denial of Service* (*DDoS*) adalah subkategori serangan *Denial of Service* (*DoS*) yang terjadi dengan membanjiri *hos*t atau jaringan target dengan lalu lintas hingga target menjadi tidak responsif atau bahkan *crash*, sehingga mencegah akses pengguna.

Serangan *Distributed Denial of Service* (*DDoS*) kini dilakukan secara terorganisir, dikendalikan dari jarak jauh, dan didistribusi ke berbagai komputer *zombie* atau *botnet* yang mengirimkan lalu lintas dalam jumlah besar ke target baik secara terus menerus maupun bersamaan. Hal ini dapat menyebabkan server menjadi lambat, tidak responsif, atau tidak berfungsi sama sekali [2]

Pengembangan *Intrusion Detection Systems (IDS)* yang dilakukan oleh [3] untuk mengatasi serangan *DDoS* pada *Software Defined Network* (SDN) dengan memanfaatkan *Neural Network* berhasil mendeteksi koneksi yang mencurigakan dan berbahaya. Selanjutnya [4] mengembangkan mode IDS yang andal dan efisien menggunakan *Artificial Neural Network* (ANN) dan mengujinya pada dataset *UNB-CIC Tor Network Traffic*, hasil penelitian ini menunjukkan akurasi yang sangat baik dan dapat diandalkan.

Perbedaan penelitian yang sudah dilakukan sebelumnya terhadap penelitian ini adalah pada penelitian ini melakukan analisis deteksi intrusi dengan menggunakan 4 metode Algoritma *Machine Learning* yaitu *eXtreme Gradient Boost, CatBoost, LightGBM* dan *Decision Tree*. Secara keseluruhan berbeda dengan penelitian lain nya. Dan kesamaan penelitian pada sebelumnya dengan penelitian ini adalah ada kesamaan dalam menggunakan *dataset* RT-IoT2022.

Dalam penelitian ini, bertujuan untuk membandingkan beberapa metode akurasi terhadap analisis *Intrusion Detection System (IDS)*. Dataset yang digunakan dalam analisis ini adalah *RT-IoT2022* yang dapat diunduh dari situs https://archive.ics.uci.edu/dataset/942/rt-iot2022. Dan metode yang digunakan adalah *eXtreme Gradient Boost, Catboost, LightGBM dan Decision Tree* sehingga

memungkinkan klasifikasi antara lalu lintas normal dan lalu lintas berbahaya yang merupakan serangan *DdoS*

2. METODE

Deteksi intrusi pernah dilakukan oleh Gregorius Airlangga menggunakan dataset RT-IOT2022, dengan menggunakan *metode Gradient Boosting, Random Forest, Logistic Regression,* dan *Multi-Layer Perception.* Hasil dari *Gradient Boosting* dan *Random Forest* mencapai skor sempurna denga akurasi, presisi, recall dan F1 sebesar 1. Sedangkan model *Multi-Layer Perception* dengan akurasi, presisi, recall dan F1 sebesar 0,99. Dan Regresi Logistik dengan skor 0,96 di seluruh metrik [5].

Pada penelitian BS Sharmila, Rohini Nagapadma. Penelitian ini mengusulkan penggunaan *quantized autoencoder* (*QAE*) sebagai solusi untuk mendeteksi anomaly jaringan pada perangkat *edge IoT* dengan sumber daya terbatas. Model *QAE*, yang menggabungkan teknik pemangkasan, pengelompokkan, dan kuantisasi bilangan bulat, terbukti lebih efisien dibandingkan model *autoencoder* tradisional dalam hal penggunaan memori dan *CPU*. Dalam uji eksperimental, varian *QAE-u8* menunjukkan kinerja unggul dengan pengurangan penggunaan memori rata-rata sebesar 70,01%, kompresi ukuran memori sebesar 92,23%, dan penggunaan *CPU* sebesar 27,94%. Hasil ini menunjukkan bahwa *QAE-u8* sangat cocok untuk diterapkan pada perangkat *edge IoT* yang memiliki keterbatasan sumber daya, menjadikannya solusi efektif untuk deteksi anomali jaringan [6].

Penelitian yang dilakukan G Ranjith Kumar, Navnath Sopan Govekar, A Karthik, Ginni Nijhawan, Ahmed Hussien Alawadi, Asha V. mengevaluasi efektivitas empat algritma *QAE, WOA, PSO-DL,* dan *DL-AD* dalam meningkatkan keamanan jaringan rumah sakit *IoT* yang memiliki sumber daya terbatas. Menggunakan kumpulan data *RT-IoT2022* untuk mensimulasikan serangan siber, hasilnya menunjukkan bahwa *QAE* memiliki *presisi* (0,92) dan *recall* (0,88) yang baik, sementara *WOA* menunjukkan kinerja kompetitif dengan *presisi* (0,85) dan *recall* (0,78). Metode *PSO-DL* menunjukkan kinerja terbaik dengan akurasi (0,94), *recall* (0,91), dan skor *F1* (0,92). *DL-AD* berfungsi sebagai tolak ukur dengan *presisi* dan *recall* masing-masing rata-rata 0,98 dan 0,86. Algoritma yang diusulkan, terutama *QAE* dan *PSO-DL*, memberikan hasil kinerja yang lebih baik dalam kebanyakan kasus dibandingkan karya terkait lainnya, menunjukkan bahwa

pendekatan yang disesuaikan efektif untuk jaringan *IoT* Kesehatan dengan sumber daya terbatas [7].

Penelitian yang ditulis oleh Farid Muhammad, Ida Wahidah, dan Arif Idra Irawan menganalisis pendeteksian serangan *Denial of Service* (*DoS*) menggunakan algoritma Logika *Fuzzy* metode *Mamdani* pada jaringan *Internet of Things* (IoT). Penelitian ini menggunakan perangkat lunak *MATLAB* dan melakukan perbandingan *Quality of Service* (*QoS)* dengan *Cooja Simulator*. Hasil perbandingan *QoS* antara *MATLAB* dan *Cooja* menunjukkan bahwa *Cooja* mencapai 98,62% pada pengujian 20 *node* [8].

Penelitian oleh M. Alfine Ridho dan Molavi Arman, menggunakan Jaringan Saraf untuk mendeteksi serangan *DDoS* dengan metode *Intrusion Detection System* (IDS). Skema serangan *DDoS* menggunakan topologi jaringan yang dirancang berdasarkan pemantauan lalu lintas. Data terdiri dari 27 log lalu lintas untuk *DDoS* dan normal, dengan total 54 dataset, dan 10 data uji untuk masing-masing. Pengumpulan data dilakukan dengan *LOIC, HOIC,* dan *DoSHTTP* selama 300 detik. Hasil pemrosesan *Fixed Moving Window* adalah nilai ektraksi dengan 6 input, satu lapisan tersembunyi (300 *neuron*), dan 2 output (normal dan *DDoS*). Penelitian menunjukkan bahwa Jaringan Saraf dapat mendeteksi *DDoS* dan Normal dengan akurasi sebesar 95% [9].

Pada makalah yang diteliti oleh Adam Zukhruf, Bagus Fatkhurrozi, Andriyana Agung Kurniawan. bertujuan untuk membandingkan kinerja aplikasi *Snort*, *Suricata*, dan *Wireshark* dalam mendeteksi serangan *Distributed Denial of Service* (*DDoS*), yang bertujuan untuk menghabiskan sumber daya server sehingga tidak dapat digunakan. Parameter perbandingan melibatkan total serangan yang terdeteksi dan penggunaan memori, dengan jenis serangan termasuk *syn flood* dan *ping of death*. Hasil penelitian menunjukkan bahwa *Suricata* menjadi aplikasi paling efektif, unggul dalam penggunaan memori pada kedua jenis serangan 0.1891 GB atau 4,974% untuk *syn flood*, dan 0,00114 GB atau 0.03% untuk *ping of death*). *Suricata* juga mencapai tingkat deteksi tertinggi untuk serangan *ping of death*, yaitu sebesar 86.472% [10].

3. Dalam penelitian Hartanto Tantriawan, Rajif Agung Yunmar, Andhika Setiawan, Meiji Suryadi, menggunakan algoritma *fuzzy logic* untuk mendeteksi serangan *DDoS* dengan menganalisis lalu lintas jaringan menggunakan *Wireshark*. Pengolahan data lalu lintas dengan *Fuzzy Logic Sugeno* memungkinkan deteksi

serangan *DDoS*, dengan hasil penelitian menunjukkan tingkat deteksi sebesar 70% [11] **. Metode**

3.1 Analisis Kebutuhan

Tahapan pertama yang dilakukan yaitu mencari dan menganalisis kebutuhan yang diperlukan untuk mendukung penelitian ini. Analisis dan pendefinisian kebutuhan meliputi, kebutuhan perangkat keras (*hardware*), kebutuhan perangkat lunak (*software*), dan data yang diperlukan.

Pada penelitian deteksi intrusi ini menggunakan komputer dengan spesifikasi *hardware* sebagai berikut:

Tabel I. Spesifikasi Hardware Komputer

No	***Hardware***	**Spesifikasi**
1.	*Processor*	Intel(R) Core(TM) CPU i5-9300 @ 2.40GHz (8 CPUs), ~2,4GHz
2.	Memori (*RAM*)	24GB
3.	*Harddisk*	512GB SSD
4.	Grafis	NVIDIA GeForce GTX

Dan spesifikasi komputer ini menggunakan *software* sebagai berikut:

Tabel II Spesifikasi *Software* Komputer

No	***Software***	**Spesifikasi**
1.	Sistem Operasi	Windows 11 Home Single Language 64-bit (10.0, Build 22621)
2.	Aplikasi	*Visual Code Studio*
3.	Pemrograman	*Python* 3.12.0

3.2 Dataset

Penelitian dengan pendekatan kuantitatif memerlukan sumber daya yang besar, baik dari segi waktu maupun biaya, yang sering kali sulit dipenuhi oleh seorang mahasiswa. Oleh karena itu, menganalisis data sekunder menjadi strategi yang lebih efektif. Dalam penelitian ini, digunakan data sekunder.

Untuk membuat model deteksi serangan intrusi, diperlukan *dataset* untuk proses pelatihan dan pengujian. *Dataset* ini diperoleh dari situs

https://archive.ics.uci.edu/dataset/94/rt-iot2022, yang menyediakan *dataset* terkait keamanan siber, termasuk *malware* pada *android*, serangan *DDoS*, serangan *botnet* dan lainnya

3.3 Teknik Analisis

Tahapan-tahapan analisis yang akan dilakukan peneliti dalam penelitian ini adalah sebagai berikut:

a. Pembersihan data, beberapa operasi dilakukan untuk mendeteksi baris kosong, fitur yang redundan, dan nilai non numeric pada kolom numeric. Nilai seperti '*NaN*' atau '*infinity*' akan dihapus.
b. Pembuatan *dataset training* dan *testing* dengan metode *train-test split*, metode evaluasi ini membagi *dataset* menjadi dua bagian dengan proporsi persentase ke dalam *training* data dan *testing* data. *Training* data digunakan untuk fit model, sedangkan *testing* data digunakan untuk mengevaluasi hasil fit model tersebut.
c. Dengan menggunakan metode *Ensemble* dan *Decision Tree* sehingga didapat fitur-fitur yang relevan dari *dataset.* Pemilihan metode yang relevan akan meningkatkan akurasi model.
d. Melatih *Ensemble* dan *Decision Tree* dengan *dataset training*.
e. Melakukan pengujian terhadap model Ensemble dan Decision Tree hasil pelatihan terhadap *dataset test*, sehingga dapat ditemukan nilai akurasi dan waktu komputasi dari proses pelatihan dan pengujian.

3.4 Metode Catboost

CatBoost (*Categorical Boosting*) adalah algoritma yang berbasis pada *gradient boosting decision tree* yang efektif dalam menangani fitur kategori. Penanganan fitur kategori dilakukan selama proses pelatihan, bukan pada tahap pemrosesan. Algoritma ini memperkenalkan skema baru untuk menghitung nilai daun saat memilih struktur pohon yang membantu mengurangi risiko *overfitting* [12]

3.5 Metode *Extreme Gradient Boost*

Extreme Gradient Boosting (*XGBoost*) adalah teknik *machine learning* untuk analisis regresi dan klasifikasi yang didasarkan pada *Gradient Boosting Decision Tree* (*GBDT*). *XGBoost* pertama kali diperkenalkan oleh [13]. menggabungkan konsep *boosting* dan optimasi dalam pengembangan *Gradient Boosting Machine* (*GBM*). Dalam *boosting*, model baru dibangun untuk memprediksi kesalahan dari model sebelumnya,

dan proses ini berlanjut hingga tidak ada perbaikan lebih lanjut. Dengan menggunakan *gradient descent* untuk meminimalkan kesalahan, algoritma ini dikenal sebagai *gradient boosting* [14].

3.6 Metode *LightGBM*

LightGBM (Light Gradient-Bossting Machine) algoritma yang dirancang oleh *Microsoft Research Asia* menggunakan kerangka *Gradient Boosting Decision Tree* (*GBDT*) [15]. Tujuannya untuk meningatkan efisiensi komputasi, sehingga masalah prediksi dengan *big data* dapat diselesaikan dengan efisien [16]. *LightGBM* memiliki beberapa keunggulan dibandingkan metode *GBDT* lainnya, yaitu kecepatan pelatihan lebih cepat, efisiensi lebih tinggi, penggunaan memori lebih rendah, tingkat akurasi lebih baik, kemampuan dalam menangani data dengan skala yang besar dan dukungan pembelajaran paralel dan *GPU* [17]. *LightGBM* adalah kerangka *Gradient Boosting* yang cepat, terdistribusi dan berkinerja tinggi berdasarkan algoritma pohon Keputusan yang dapat digunakan untuk peringkat, klasifikasi, regresi dan banyak tugas pembelajaran mesin lainnya [17].

3.7 Normalisasi

Normalisasi adalah proses penskalaan fitur ke dalam rentang [0,1], yang merupakan bentuk khusus penskalaan min-maks dapat diterapkan pada satu atau beberapa kolom fitur,. Berikut adalah rumus untuk menormalisasi data menggunakan peskalaan min,maks.

$$x_{norm}^{(i)} = \frac{x^{(i)} - x_{min}}{x_{max} - x_{min}}$$

Gambar I. Menormalkan data berdasarkan konsep penskalaan min-maks
Sumber: https://vitalflux.com/minmaxscaler-standardscaler-python-examples/

MinMaxScaler sangat berguna ketika data memiliki rentang terbatas atau distribusi yang tidak mengikuti pola *Gaussian*. Contohnya, dalam pemrosesan gambar, nilai piksel biasanya berada dalam rentang 0-255. Dengan menggunakan *MinMaxScaler*, nilai-nilai ini dapat diperkecil sehingga dalam rentang tetap, biasanya antara 0 dan 1.

1) ***Train-Test Split***

2) **Computational Resources:** This included CPU and memory usage for GNFS in classical systems, and the qubit requirements for running Shor's algorithm in quantum systems.

Train-Test Split adalah metode yang digunakan untuk mengevaluasi performa model *Machine Learning*. Metode ini membagi *dataset* menjadi dua bagian: satu bagian untuk pelatihan (*training)* dan satu bagian untuk pengujian (*testing)*, dengan proporsi yang ditentukan. Data pelatihan digunakan untuk melatih model, sedangkan data pengujian digunakan untuk mengevaluasi hasil dari pelatihan model tersebut.

Metode *train-test split* sangat cocok untuk dataset yang berukuran besar. Dengan metode ini, dataset dibagi menjadi *train* set dan *test* set, atau dengan kata lain, data yang digunakan untuk pelatihan dan pengujian adalah bagian yang berbeda dari dataset. (Sumber: https://ilmudatapy.com/evaluasi-model-machine-learning-dengan-train-test-split/).

Berikut penjelasan tentang kinerja Train-Test Split:

1. Fungsi '*Train-Test Split*'
   - Digunakan untuk membagi *dataset* ('*dfNormalizer*','*dfMinMax*') dan label ('*y1*') menjadi data latih ('*Xtrain*','*Ytrain*') dan data uji ('*Xtest*','*Ytest*').
   - Parameter '*test_size*=0.2' berarti 20% data digunakan untuk pengujian, dan 80% sisanya digunakan untuk pelatihan.
2. Pemanggilan *Train-Test Split Classifier*
   - Fungsi ini menggabungkan proses *train-test split* dengan pelatihan dan evaluasi model menggunakan *classifier* yang berbeda seperti '*clfXGB*','*clfGBM*', dan '*clfCat*'.

Inti dari proses *Train-Test Split* adalah proses penting dalam *Machine Learning* untuk memastikan model tidak *overfitting* dengan menguji model pada data yang belum pernah dilihat selama pelatihan. *Xtrain* dan *Ytrain* digunakan untuk melatih model, *Xtest* dan *Ytest* digunakan untuk menguji performa model tersebut. Secara keseluruhan, *train-test split* untuk membagi dataset dan melakukan pelatihan serta evaluasi pada beberapa model *machine learning* yang berbeda.

3.8 K-Folds Cross Validation

K-Folds Cross Validation merupakan metode yang digunakan untuk mengevaluasi model guna memperoleh model terbaik. Validasi model perlu untuk mengukur kinerja model serta mengetahui tingkat kesalahan model dalam hal memprediksi, kemudian dapat mengoptimalkan model sehingga diperoleh model yang lebih baik dan lebih akurat [18].

Pada setiap iterasi, data *test* diganti dengan bagian (*fold*) data lain sehingga setiap bagian data akan digunakan sebagai data *test* sebanyak satu kali [19]. Dalam 5-fold Cross-Validation, dataset dibagi menjadi 5 subset yang kurang lebih sama besar.

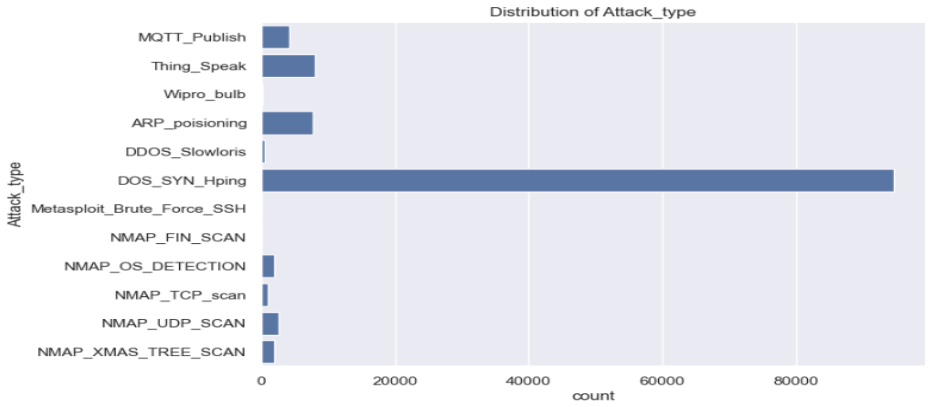
1. Proses CV=5
   - Dataset dibagi menjadi 5 subset (fold).
   - Pada setiap iterasi, 1 subset digunakan sebagai data uji *(testing set*) dan 4 *subset* lainnya digunakan sebagai data latih (*training set*)
   - Proses ini diulang 5 kali sehingga setiap *subset* digunakan sebagai data uji satu kali.
   - Skor dari 5 iterasi tersebut dihitung rata-rata untuk mendapatkan estimasi akurasi model yang stabil dan mengurangi variansi dibandingkan dengan hanya satu *train-test split*.
2. Fungsi ‘cvClassifier’
   - Fungsi ini melakukan *cross-validation* pada model (‘clf’) menggunakan *dataset* (‘X’,’Y’) dengan ‘Boost’, ‘normalizer’, dan ‘fitur’ yang diberikan.
   - Parameter ‘cv=5’ pada fungsi ‘cross_val_score’ (‘cvs’) mengindikasikan bahwa proses *5-fold cross-validation* sedang digunakan.
   - Skor akurasi rata-rata (‘scores.mean()’) dan deviasi standar akurasi (scores.std()’) dari 5-fold cross-validation dicetak sebagai output untuk menilai performa model.

CV=5 menunjukkan bahwa kinerja model dievaluasi menggunakan *5-fold cross-validation,* yang membantu dalam mengukur seberapa baik model dapat menggeneralisasi ke data yang belum dilihat selama pelatihan.

4. Hasil Dan Pembahasan

Data – data pada Tabel I tersebut menjadi kelas sesuai dengan nama-nama trafik. Dari data populasi akan diambil data sampel secara acak dengan memperhatikan proporsi

data yang terdapat pada data populasi. Berikut visualisasi sederhana mengenai label trafik pada Gambar I.



Gambar II. Jenis – jenis Trafik Pada Dataset RT_IOT2022

Dari gambar I tersebut menunjukkan distribusi jenis trafik dalam kumpulan data. Jenis trafik terbagi menjadi dua yaitu Pola Serangan dan Pola Normal sebagai berikut:

Tabel III Label Trafik Pola Serangan
Sumber: https://archive.ics.uci.edu/dataset/942/rt-iot2022

No	Nama Variabel	Jumlah Serangan
1	DOS_SYN Hping	94659
2	ARP_poisioning	7750
3	NMAP_UDP_SCAN	2590
4	NMAP_XMAS_TREE_SCAN	2010
5	NMAP_OS_DETECTION	2000
6	NMAP_TCP_scan	1002
7	DDOS_Slowloris	534
8	Metasploit_Brute_Force_SSH	37
9	NMAP_FIN_SCAN	28

Tabel IV Label Trafik Pola Normal
Sumber: https://archive.ics.uci.edu/dataset/942/rt-iot2022

No	Nama Variabel	Jumlah Serangan
1	MQTT	8108
2	Thing_speak	4146
3	Wipro_bulb_Dataset	253
4	Amazon-Alexa	86842

Langkah selanjutnya melatih model *Ensemble* (*Extreme Gradient Boost, LightGBM,* dan *Catboost*) yang dapat dilihat pada Tabel V sebagai berikut:

Tabel V Klasifikasi Kinerja *Boosting*

Object	**Classifier**	**TTS 80:20**				**5 Folds CV**			
		Acc	**Time**			**Acc**	**Time**		
			H	**M**	**S**		**H**	**M**	**S**
Normalizer	XGBoost	0.9982	0	00	22	0.9909	0	01	27
	LightGBM	0.7704	0	00	31	0.8585	0	02	20
	CatBoost	0.9982	0	10	33	0.9916	0	52	41
MinMaxScaler	XGBoost	0.9989	0	00	15	0.9409	0	00	59
	LightGBM	0.5128	0	00	25	0.6580	0	01	54
	CatBoost	0.9988	0	09	51	0.9903	0	52	37

Dari skema pengujian yang dilakukan, XGBoost dengan MinMaxScaler dan Train-Test Split 80:20 menghasilkan akurasi tertinggi sebesar 99,89%. Sebaliknya, akurasi XGBoost dengan 5 Fold Cross-Validation turun menjadi 94,04%. Namun XGBoost dengan MinMaxScaler secara konsisten membutuhkan waktu komputasi terendah di antara skema yang dilakukan. Untuk skema pembagian tetap, XGBoost membutuhkan 15 detik untuk train-test split, sedangkan untuk 5 fold cross-validation 59 detik.

5. Kesimpulan

Berdasarkan hasil penelitian yang telah dijelaskan, dapat disimpulkan model terbaik dengan memperhatikan akurasi dan waktu eksekusi dari berbagai kombinasi model, metode normalisasi, dan skema validasi. Berikut adalah rangkuman dari setiap kombinasi yang diberikan sebagai berikut:

1. *Extreme Gradient Boost (XGBoost)*
   - Cenderung memberikan akurasi tinggi dengan waktu eksekusi yang relatif cepat, terutama saat menggunakan *Normalizer.*
2. *LightGBM*
   - Menunjukkan variasi yang cukup besa dalam akurasi, dengan hasil terbaik pada *Normalizer* dan metode *5 Folds Cross Validation.*
3. *Catboost*

- Memberikan akurasi yang sangat tinggi namun dengan waktu eksekusi yang lebih lama, khususnya dalam metode *5 Folds Cross Validation.*

4. Normalisasi

- Umumnya memberikan hasil akurasi yang lebih tinggi dibandingkan dengan *MinMaxScaler* dalam kebanyakan kombinasi *Classifier*.

5. DAFTAR PUSTAKA

[1] I. Sharafaldin, A. Lashkari, S. Hakak and A. Ghorbani, Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy, International Carnahan Conference on Security Technology (ICCST) 2019., 2019.

[2] K. Prasad, DoS and DDoS Attacks:Defence, Detection and Traceback Mechanism -A Survey., Global Journal of Computer Science, 2014.

[3] N. Meti, D. G. Narayan and V. P. Baligar, Detection of Distributed Denial of Service Attacks Using Machine Learning Algorithms in Software Defined Networks, International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017.

[4] E. Hodo, X. Bellekens, E. Iorkyase, A. Hamilton, C. Tachtatzis and R. Atkinson, Machine Learning Approach for Detection of non Tor Traffic, Proceedings of the 12th International Conference on Availability, Realiability and Security, 2017.

[5] G. Airlangga, Comparative Analysis of Machine Learning Models for Intrusion Detection in Internet of Things Network Using the RT-IoT2022 Dataset, Malcom: Indonesia Journal of Machine Learning and Computer Science, 2024, 2024.

[6] B. Sharmila and R. Nagapadma, Sistem Deteksi Intrusi Quantized Autoencoder (QAE) Untuk Deteksi Anomali Pada Perangkat IoT Dengan SUmber Daya Terbatas Menggunakan Kumpulan Data RT-IoT2022, Keamanan Siber Sharmila dan Magapadma (2023), 2023.

[7] G. R. Kumar, N. S. Govekar, A. Karthik, G. Nijhawan, A. H. Alawadi and A. V, Real-TIme Monitoring and Anomaly Detection in Hospital IoT Networks Using Machine Learning, 2023 International Conference on Artificial Intelligence for Innovation in Healtcare Industries, 2023.

[8] F. Muhammad, I. Wahidah and A. I. Irawan, Analisis Pendeteksian Serangan Denial of Service (DoS) Menggunakan Logika FUzzy Metode Mamdani Pada Jaringan Internet of Things, e-Prociding of Engineering, 2021, 2021.

[9] M. R. Alfine and M. Arman, Analisis Serangan DDoS Menggunakan Metode Jaringan Saraf Tiruan, Jurnal SISFOKOM (Sistem Informasi dan Komputer), 2020., 2020.

[10] A. Zukhruf, B. Fatkkurrozi and A. A. Kurniawan, Comparative Study of Distributed Denial of Service (DDoS) Attack Detection in Computer Networks, Jurnal Teknik Informatika (JUTIF), 2023., 2023.

[11] H. Tantriawan, R. A. Yunmar, A. Setiawan and M. Suryadi, Deteksi Distributed Denial of Service (DDoS) Menggunakan Fuzzy Logic Sugeno, Malcom: Indonesia Journal of Machine Learning and Computer Science, 2021., 2021.

[12] L. Prokhorenkova, L. Guseb, A. Vorobev, A. Dorogush and A. Gulin, Catboost: unbiased boosting with categorial features [online], https://github.com/catboost/catboost, 2019.

[13] J. Friedman, Greedy Function Aproximation A Gradient Boosting Machine, Annals of Statistics, 29(5), 1189-1232, 2001.

[14] M. Hao, S. Hejiang, L. Junjie and W. Shen, Developing window behavior models for residential buildings using XGBoost algorithm., Energy and Buildings, 205, 109564. https://doi.org/10.1016/j.enbuild.2019.109564, 2019.

[15] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, M. Weidong, Q. Ye and L. Tie-Yan, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, Advances in neural information processing systems, 30, 3146-3154, 2017.

[16] W. Liang, S. Luo, G. Zhao and H. Wu, Predicting Hard Rock Pillar Stability Using GBDT, XGBoost, and LighGBM, Mathematics 8(5), 765, https://doi.org/10.3390/math8050765, 2020.

[17] D. D. Rufo, T. G. Debelee, A. Ibenthal and W. G. Negera, Diagnosis of Diabetes Mellitus Using Gradient Boosting Machine (LightGBM), Diagnostics 11(9), 1714, https://doi.org/10.3390/diagnostics11091714, 2021.

[18] T. T. Wong, Performance evaluation of classification algorithm by k-fold and leave-one-out cross validation, Pattern Recognition, vol. 48 no.9, pp. 2839-2846 sep. 2015, doi:10.1016/j.patcog.2015.03.009, 2015.

[19] L. Prokhorenkova, G. Guseb, A. Vorobev, A. V. Dorogush and A. Gulin, Catboost: unbiased boosting with categorial features [online], https://github.com/catboost/catboost, 2019.

[20] I. Sharafaldin, A. H. Lashkari, S. Hakak and A. A. Ghorbani, Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy, 2019 International Carnahan Conference on Security Technology (ICCST), 2019.

[21] S. Artificial Intelligence: Searching, Reasoning, Planning and Learning(2nd ed)., Informatika, Bandung, Indonesia., 2014.

[22] I. Mengenal Decision Tree dan Manfaatnya, https://medium.com/iykra/mengenal-decision-tree-dan-manfaatnya-b98cf3cf6a8d, 2018.

[23] L. Prokhorenkova, G. Guseb, A. Vorobev, A. Dorogush and A. Gulin, Catboost: unbiased boosting with categorial features [online]., 2019: https://github.com/catboost/catboost.