

OPTIMASI FITUR BERBASIS ANOVA UNTUK DETEKSI SERANGAN PADA PROTOKOL MQTT MENGGUNAKAN RANDOM FOREST DI LINGKUNGAN INTERNET OF THINGS

Aditiya Fajrin Maulana ¹, Heri Setiawan ²

¹ Rekayasa Keamanan Siber, Politeknik Angkatan Darat

² Elektronika Sistem Senjata, Politeknik Angkatan Darat

Kesatrian Poltekad, Jl. Raya Anggrek, Pendem, Kec. Junrejo Kota Batu, Jawa Timur, Indonesia

mhs.cyber8.aditiyafm@poltekad.ac.id

ABSTRAK

Protokol MQTT (Message Queuing Telemetry Transport) banyak digunakan pada sistem Internet of Things (IoT) karena arsitektur publish-subscribe yang ringan dan hemat bandwidth, namun memiliki kerentanan keamanan tinggi terhadap serangan Denial of Service (DoS) dan anomali jaringan. Permasalahan utama dalam penerapan Intrusion Detection System (IDS) pada edge IoT adalah tingginya dimensi fitur jaringan yang membebani memori dan waktu komputasi perangkat berdaya terbatas, sehingga menimbulkan trade-off kritis antara akurasi deteksi dan efisiensi latensi. Penelitian ini bertujuan mengevaluasi dampak optimasi fitur berbasis Analysis of Variance (ANOVA) pada model Random Forest guna mereduksi beban komputasi tanpa mengorbankan performa deteksi serangan MQTT. Metode yang digunakan adalah klasifikasi Random Forest dengan partisi data 70% pelatihan dan 30% pengujian pada 1.000 sampel data trafik jaringan seimbang (50% normal dan 50% serangan), membandingkan Skenario 1 (10 fitur penuh) dan Skenario 2 (7 fitur hasil eliminasi metrik transport layer TCP: TcpRtt, AckDat, dan SynAck berbasis ANOVA). Hasil pengujian menunjukkan bahwa Skenario 1 menghasilkan akurasi 100,00%, sedangkan Skenario 2 mempertahankan akurasi sebesar 99,67% (hanya terkoreksi 0,33%). Selain itu, Skenario 2 berhasil memangkas waktu pelatihan model sebesar 34,6% (dari 0,7854 detik menjadi 0,5135 detik). Hasil ini membuktikan bahwa optimasi fitur berbasis ANOVA mampu meningkatkan efisiensi komputasi secara signifikan dengan tetap menjaga akurasi deteksi yang optimal untuk lingkungan edge computing IoT.

Kata kunci : ANOVA, Deteksi Serangan, Edge Computing, Internet of Things, MQTT, Random Forest.

1. PENDAHULUAN

Masifnya pertumbuhan perangkat Internet of Things (IoT) pada berbagai sektor mendorong adopsi protokol komunikasi yang ringan dan efisien. MQTT (Message Queuing Telemetry Transport) hadir sebagai standar de facto karena memanfaatkan arsitektur publish-subscribe yang hemat konsumsi bandwidth [1]. Kendati demikian, efisiensi pertukaran data ini mengorbankan aspek keamanan karena spesifikasi MQTT pada awalnya tidak menyertakan enkripsi bawaan atau verifikasi payload yang ketat, sehingga menjadikannya sasaran empuk terhadap serangan Denial of Service (DoS), flooding broker, dan manipulasi paket data [2].

Pemanfaatan machine learning untuk membangun sistem deteksi intrusi (Intrusion Detection System/IDS) pada ekosistem IoT telah dieksplorasi secara luas oleh para peneliti dalam beberapa tahun terakhir [3], [4]. Mayoritas literatur berupaya mencapai akurasi klasifikasi tertinggi dengan mengekstraksi seluruh fitur statistik dari paket aliran jaringan [5]. Namun, pendekatan pemanfaatan fitur penuh mengabaikan kenyataan fisik bahwa perangkat edge pada jaringan IoT memiliki kapasitas komputasi, memori RAM, dan pasokan daya baterai yang sangat terbatas [6]. Model machine learning yang memproses dimensi fitur terlalu besar membutuhkan waktu pelatihan dan inferensi yang lama, sehingga tidak

realistis jika diterapkan langsung di tingkat perangkat lapangan (edge node) [7].

Tantangan fundamental dalam perancangan IDS untuk lingkungan IoT adalah adanya celah riset (research gap) berupa trade-off antara tingkat akurasi deteksi dan beban komputasi [8]. Algoritma ensemble seperti Random Forest terbukti unggul dan stabil dalam menangani data tabular trafik jaringan [9]. Akan tetapi, kompleksitas komputasional Random Forest meningkat secara linear terhadap jumlah dimensi fitur yang digunakan [10]. Apabila seluruh fitur diproses tanpa seleksi, pemantauan status koneksi lapisan transport (TCP) akan menimbulkan penundaan latensi (latency overhead) yang dapat melumpuhkan respons mitigasi serangan.

Oleh karena itu, diperlukan strategi seleksi fitur yang efektif untuk memangkas dimensi data tanpa mendegradasi sensitivitas deteksi serangan. Penelitian ini bertujuan mengatasi permasalahan tersebut dengan mengevaluasi optimasi fitur berbasis Analysis of Variance (ANOVA) pada model Random Forest. Kontribusi utama yang diberikan adalah pembuktian empiris bahwa eliminasi fitur berbasis TCP (TcpRtt, AckDat, SynAck) mampu memangkas waktu komputasi pelatihan secara signifikan dengan tetap mempertahankan akurasi klasifikasi serangan MQTT di atas 99,6%.

2. TINJAUAN PUSTAKA

2.1. MQTT pada IoT

MQTT merupakan protokol komunikasi data mesin-ke-mesin (M2M) yang beroperasi pada lapisan aplikasi di atas protokol transport TCP/IP [1]. Mekanisme kerjanya tersusun atas tiga komponen: publisher yang mengirimkan data sensor, broker sebagai pusat perantara dan penyaring topik, serta subscriber yang menerima data. Walaupun format header MQTT sangat ringkas (minimal 2 byte), sifat koneksi TCP yang persisten rentan dieksploitasi oleh penyerang melalui banjir permintaan koneksi (connection exhaustion) yang dapat melumpuhkan fungsi broker [2].

2.2. Intrusion Detection System (IDS)

IDS berbasis anomali menggunakan model machine learning menawarkan kemampuan adaptif yang lebih baik dibanding metode konvensional berbasis signature atau rule-based [11]. Model dilatih menggunakan representasi fitur statistik trafik normal dan anomali, sehingga mampu mengenali varian serangan baru yang belum pernah teridentifikasi sebelumnya melalui generalisasi pola [3].

2.3. Algoritma Random Forest

Random Forest merupakan algoritma ensemble berbasis bagging yang membangun sekumpulan pohon keputusan (decision trees) secara acak selama fase pelatihan [12]. Setiap pohon membuat prediksi independen, dan keputusan akhir ditentukan melalui skema majority voting. Metode ini memiliki ketahanan tinggi terhadap overfitting dan mampu menangani hubungan non-linear antar fitur [9]. Persamaan klasifikasi akhir Random Forest dirumuskan sebagai:

$$\hat{y} = \text{mode} \{T_1(x), T_2(x), \dots, T_n(x)\} \quad (1)$$

Keterangan: $\hat{y}$ adalah prediksi kelas akhir; $T_i(x)$ menyatakan luaran pohon ke- i ; n adalah jumlah pohon keputusan.

2.4. Seleksi Fitur ANOVA

Analysis of Variance (ANOVA) adalah metode uji statistik parametrik yang mengukur signifikansi rasio varians nilai fitur antar kelompok target dibandingkan varians di dalam kelompok [13]. Dalam konteks seleksi fitur, ANOVA menghitung nilai F-score untuk mengevaluasi daya pemisah suatu fitur kontinu terhadap kelas biner (normal versus serangan). Fitur dengan nilai F-score tinggi menunjukkan bahwa fitur tersebut memiliki kontribusi diskriminatif yang dominan [10].

2.5. Penelitian Terdahulu (Related Works)

Sejumlah penelitian mutakhir telah mengkaji deteksi anomali pada protokol MQTT dengan beragam algoritma dan teknik reduksi fitur. Imran dkk. [7] menerapkan analisis korelasi untuk mereduksi dimensi dataset MQTTset, berhasil meningkatkan akurasi namun belum menganalisis penghematan waktu pelatihan pada node edge. Al Hanif dan Ilyas [5] merancang kerangka rekayasa fitur khusus MQTT dengan Decision Tree dan Random Forest, tetapi membutuhkan pemantauan aliran bidirectional yang kompleks. Khan dkk. [3] memanfaatkan arsitektur Deep Neural Network (DNN) pada dataset MQTT-IoT-IDS2020 dengan akurasi melampaui 99%, namun model memerlukan komputasi intensif yang sulit dijalankan pada mikrokontroler. Alassaf dkk. [4] membandingkan model XGBoost dan Random Forest pada anomali MQTT-SN, menegaskan keandalan Random Forest untuk data tabular. Sementara itu, Fauzi dan Syarif [10] membuktikan bahwa seleksi fitur ANOVA F-test mampu memangkas lebih dari 50% fitur IDS tanpa penurunan akurasi yang berarti pada dataset umum. Tabel 1 merangkum perbandingan penelitian terdahulu dengan penelitian ini.

Tabel 1. Perbandingan Penelitian Terdahulu Terkait Deteksi Serangan IoT/MQTT

Peneliti & Tahun	Metode	Dataset	Fokus Fitur	Temuan & Limitasi
Imran dkk. (2023) [7]	Random Forest, k-NN	MQTTset	Analisis Korelasi (Filter)	Akurasi 98,2%; belum mengukur efisiensi komputasi eliminasi fitur transport.
Al Hanif & Ilyas (2024) [5]	Random Forest, DT	MQTT-IoT-IDS2020	Rekayasa Fitur Aliran	Akurasi 99,4%; ekstraksi fitur bidirectional membebani resource gateway edge.
Khan dkk. (2021) [3]	Deep Neural Network	MQTT-IoT-IDS2020	Fitur Paket Lengkap	Akurasi 99,9%; kebutuhan memori dan latensi inferensi tinggi untuk node edge.
Fauzi & Syarif (2022) [10]	ANOVA F-test + RF	UNSW-NB15	Univariate Filter	Mereduksi 50% fitur IDS; belum dievaluasi pada lalu lintas spesifik protokol MQTT.
Penelitian Ini (2026)	ANOVA + Random Forest	BoT-IoT (MQTT)	Eliminasi Metrik TCP (7 Fitur)	Akurasi 99,67%; menghemat waktu pelatihan 34,6% dengan trade-off akurasi hanya 0,33%.

Sumber: Sintesis literatur penelitian terdahulu.

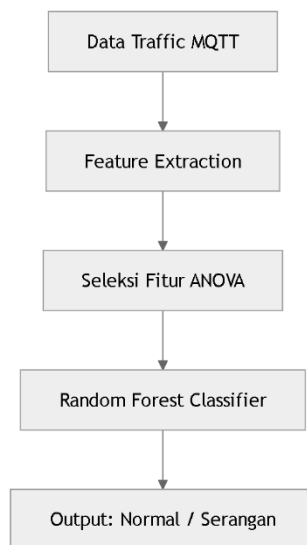
3. METODE PENELITIAN

3.1. Dataset

Eksperimen menggunakan subset data trafik jaringan IoT yang bersumber dari dataset benchmark

BoT-IoT [14]. Dataset terdiri dari 1.000 baris sampel data yang dirancang seimbang secara proporsional, yaitu 500 sampel trafik normal (50%) dan 500 sampel trafik serangan DoS/DDoS (50%) untuk mencegah

terjadinya bias kelas pada model machine learning. Terdapat 10 fitur awal yang merepresentasikan karakteristik statistik aliran data jaringan, yaitu: State_CON (koneksi transaksi), State_FIN (penutupan sesi transaksi), DstRate (laju paket ke tujuan), SrcRate (laju paket dari sumber), Dur (total durasi aliran), Rate (laju transmisi keseluruhan), TcpRtt (round trip time TCP), AckDat (waktu paket TCP ACK ke DATA), SynAck (waktu TCP SYN ke SYN-ACK), dan Load (beban bit data per detik). Dataset dipartisi menggunakan metode train-test split dengan rasio 70% data pelatihan (700 sampel) dan 30% data pengujian (300 sampel) dengan parameter random_state=42.



Gambar 1. Arsitektur Sistem Deteksi Serangan MQTT Berbasis ANOVA dan Random Forest

3.2. Skenario Eksperimen

Seleksi fitur ANOVA dilakukan dengan mengevaluasi variansi nilai masing-masing fitur kontinu antar kelas target. Secara matematis, rasio F (F-score) dihitung melalui perbandingan kuadrat rata-rata antar kelompok (Mean Square Between/MSB) terhadap kuadrat rata-rata dalam kelompok (Mean Square Within/MSW) sesuai persamaan (2):

$$F = \frac{MSB}{MSW} = \frac{\sum n_j (\bar{X}_j - \bar{X})^2 / (k - 1)}{\sum \sum (X_{ij} - \bar{X}_j)^2 / (N - k)} \quad (2)$$

Keterangan: n_j adalah ukuran sampel kelompok j ; $\bar{X}_j$ adalah rata-rata kelompok j ; $\bar{X}$ adalah rata-rata global; k adalah jumlah kelas target ($k = 2$); N adalah jumlah total sampel ($N = 1.000$).

Berdasarkan analisis variansi dan pertimbangan kompleksitas komputasi edge, tiga fitur metrik lapisan transport (TcpRtt, AckDat, dan SynAck) dieliminasi. Ketiga fitur ini memerlukan pemantauan stateful TCP handshake yang intensif terhadap memori buffer. Eliminasi fitur menghasilkan ruang fitur baru X' yang dirumuskan pada persamaan (3):

$$X' = X - F_{TCP} \quad (3)$$

Keterangan: X' adalah matriks 7 fitur terpilih (State_CON, State_FIN, DstRate, SrcRate, Dur, Rate, Load); X menyatakan matriks 10 fitur awal; F_{TCP} adalah himpunan fitur metrik transport layer TCP.

3.3. Konfigurasi Model Random Forest

Model klasifikasi dibangun menggunakan algoritma Random Forest Classifier dari pustaka Scikit-learn. Parameter yang dikonfigurasi meliputi jumlah pohon keputusan $n_estimators=100$, kriteria kemurnian Gini Impurity, serta alokasi seluruh thread pemrosesan CPU ($n_jobs=-1$) untuk menjamin kecepatan komputasi. Inisialisasi pengacakan dikontrol menggunakan random_state=42 agar eksperimen dapat direplikasi secara presisi.

3.4. Skenario Pengujian dan Metrik Evaluasi

Pengujian dirancang dalam dua skenario komparatif: Skenario 1 menerapkan seluruh 10 fitur awal sebagai baseline, sedangkan Skenario 2 menerapkan 7 fitur hasil seleksi ANOVA. Evaluasi model diukur melalui metrik Accuracy, Precision, Recall, dan F1-Score berdasarkan nilai True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN) sesuai persamaan (4) hingga (7):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \quad (6)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

Efisiensi komputasi dihitung melalui persentase reduksi waktu pelatihan model (ΔT) dengan persamaan (8):

$$\Delta T = \left(\frac{T_1 - T_2}{T_1} \right) \times 100\% \quad (8)$$

Keterangan: T_1 adalah waktu pelatihan Skenario 1 (detik); T_2 adalah waktu pelatihan Skenario 2 (detik).

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengujian Model

Hasil eksperimen klasifikasi deteksi serangan MQTT pada data pengujian (300 sampel) dirangkum secara komparatif pada Tabel 2.

Tabel 2. Perbandingan Performa Deteksi Serangan MQTT Antar Skenario

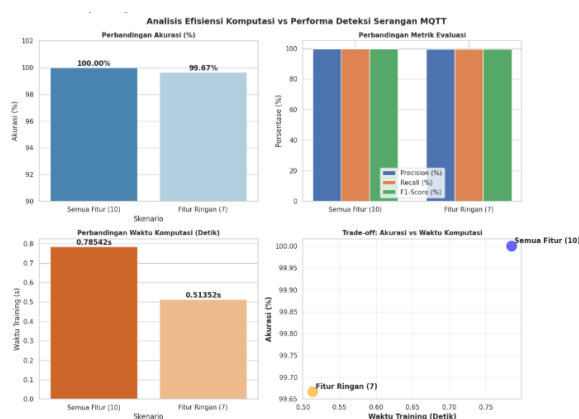
Metrik Evaluasi	Skenario 1 (10 Fitur)	Skenario 2 (7 Fitur ANOVA)
Jumlah Fitur	10	7
Akurasi (%)	100,00	99,67
Precision (%)	100,00	99,67
Recall (%)	100,00	99,67
F1-score (%)	100,00	99,67
Waktu Pelatihan (detik)	0,785420	0,513518

Sumber: Hasil eksperimen pengujian Random Forest.

4.2. Analisis Performa dan Efisiensi Komputasi

Berdasarkan Tabel 2, model Skenario 1 yang menggunakan seluruh 10 fitur mencapai performa sempurna dengan akurasi, precision, recall, dan F1-score sebesar 100,00%. Ketika model dioptimasi pada Skenario 2 dengan mengeliminasi fitur TcpRtt, AckDat, dan SynAck, performa model hanya mengalami penurunan sangat kecil sebesar 0,33%, mempertahankan akurasi sebesar 99,67%. Dari aspek efisiensi komputasi, Skenario 2 mencatatkan waktu pelatihan 0,513518 detik dibanding 0,785420 detik pada Skenario 1. Melalui persamaan (8), diperoleh persentase penurunan waktu pelatihan sebesar:

$$\Delta T = \left(\frac{0,785420 - 0,513518}{0,785420} \right) \times 100\% = 34,6\% \quad (9)$$



Gambar 2. Analisis Trade-off Akurasi vs Waktu Komputasi Antara Skenario 1 dan Skenario 2

4.3. Pembahasan dan Implikasi pada Edge Computing

Temuan eksperimen ini menjawab celah riset yang diidentifikasi pada bagian pendahuluan mengenai trade-off antara akurasi dan kompleksitas komputasi pada perangkat IoT. Pemotongan sepertiga (34,6%) durasi pelatihan model dengan kompensasi kehilangan akurasi yang hanya sebesar 0,33% merupakan capaian yang sangat menguntungkan. Dalam arsitektur edge computing sesungguhnya, fitur statistik berbasis transport layer TCP menuntut perangkat edge untuk terus menyimpan tabel status sesi (session state table) yang rentan terhadap serangan SYN flood dan memory exhaustion. Dengan hanya mengandalkan 7 fitur statistik aliran dasar (State_CON, State_FIN, DstRate, SrcRate, Dur, Rate, dan Load), gerbang edge IoT dapat melakukan deteksi serangan secara instan, menghemat siklus komputasi CPU, memperpanjang daya tahan baterai, dan mengurangi ketergantungan transmisi data ke server cloud [6], [7].

$$\Delta T = \left(\frac{T_1 - T_2}{T_1} \right) \times 100\% \quad (10)$$

Melalui substitusi nilai $T_1 = 0,785420$ detik dan $T_2 = 0,513518$ detik ke dalam persamaan, diperoleh hasil persentase efisiensi sebesar 34,6%. Model Skenario 2 terbukti memberikan peningkatan efisiensi komputasi yang masif.

5. KESIMPULAN DAN SARAN

Penelitian ini membuktikan bahwa penerapan optimasi fitur berbasis Analysis of Variance (ANOVA) mampu memberikan peningkatan efisiensi komputasi yang substansial pada sistem deteksi serangan protokol MQTT berbasis Random Forest. Mereduksi jumlah fitur dari 10 menjadi 7 fitur berhasil memangkas waktu pelatihan model sebesar 34,6% dengan tingkat akurasi yang tetap sangat tinggi, yaitu 99,67% (penurunan akurasi hanya 0,33%). Hasil ini menegaskan bahwa metrik spesifik transport TCP dapat dieliminasi guna mewujudkan IDS yang ringan dan responsif untuk lingkungan edge IoT. Saran untuk pengembangan penelitian selanjutnya adalah menguji ketahanan model terhadap variasi serangan multi-kelas MQTT yang lebih kompleks (seperti malformed packet, unauthorized publish, dan slow DoS) serta mengimplementasikan model secara langsung pada perangkat keras edge nyata seperti Raspberry Pi atau mikrokontroler ESP32.

DAFTAR PUSTAKA

- [1] OASIS, "MQTT Version 5.0," OASIS Standard, Mar. 2019. [Online]. Tersedia: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>
- [2] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, dan E. Cambiaso, "MQTTset, a New Dataset for Machine Learning Techniques on MQTT," *Sensors*, vol. 20, no. 22, p. 6578, 2020, doi: 10.3390/s20226578.
- [3] M. A. Khan, M. A. Khan, S. U. Jan, J. Ahmad, S. S. Jamal, A. A. Shah, N. Pitropakis, dan W. J. Buchanan, "A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT," *Sensors*, vol. 21, no. 21, p. 7016, 2021, doi: 10.3390/s21217016.
- [4] N. M. Allassaf, S. Manickam, A. Odeh, dan M. Anbar, "Detecting anomalies in MQTT/MQTT-SN traffic using intelligent learning models," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 4, pp. 2404-2414, 2023, doi: 10.11591/eei.v12i4.4988.
- [5] A. Al Hanif dan M. Ilyas, "Effective Feature Engineering Framework for Securing MQTT Protocol in IoT Environments," *Sensors*, vol. 24, no. 6, p. 1782, 2024, doi: 10.3390/s24061782.
- [6] S. Latif, Z. Zou, N. Idrees, dan J. Ahmad, "A Novel Attack Detection Scheme for the Industrial Internet of Things Using Explainable Machine Learning and Feature Selection," *Electronics*, vol. 12, no. 11, p. 2486, 2023, doi: 10.3390/electronics12112486.
- [7] Imran, M. F. A. Zuhairi, S. M. Ali, Z. Shahid, M. M. Alam, dan M. M. Su'ud, "Improving

- Reliability for Detecting Anomalies in the MQTT Network by Applying Correlation Analysis for Feature Selection Using Machine Learning Techniques,” *Applied Sciences*, vol. 13, no. 11, p. 6753, 2023, doi: 10.3390/app13116753.
- [8] A. Fatani, A. Dahou, M. A. Al-qaness, S. Lu, dan M. A. Abd Elaziz, “Advanced Feature Extraction and Selection Approach Using Deep Learning and Metaheuristic for IoT Intrusion Detection,” *IEEE Access*, vol. 10, pp. 104998-105011, 2022, doi: 10.1109/ACCESS.2022.3210459.
- [9] W. D. Nanda dan F. D. S. Sumadi, “DDoS Detection System in SDN Based on Machine Learning Methods,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 2, pp. 221-228, 2022, doi: 10.29207/resti.v6i2.3878.
- [10] M. A. Fauzi dan I. Syarif, “Analyzing ANOVA F-test and Sequential Feature Selection for Intrusion Detection Systems,” *International Journal of Advances in Soft Computing and Its Applications*, vol. 14, no. 2, pp. 175-188, 2022, doi: 10.15849/IJASCA.220720.13.
- [11] A. L. Buczak dan E. Guven, “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-1176, 2016, doi: 10.1109/COMST.2015.2494502.
- [12] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001, doi: 10.1023/A:1010933404324.
- [13] I. Guyon dan A. Elisseeff, “An Introduction to Variable and Feature Selection,” *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.
- [14] N. Koroniotis, N. Moustafa, E. Sitnikova, dan B. Turnbull, “Towards the Development of Realistic Botnet Dataset in the Internet of Things: The BoT-IoT Dataset,” *Future Generation Computer Systems*, vol. 100, pp. 779-796, 2019, doi: 10.1016/j.future.2019.05.041.
- [15] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, dan M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015, doi: 10.1109/COMST.2015.2444095.