Jurnal JEETech: Vol. 7 No. 1 Tahun 2026
Journal Of Electrical Engineering And Technology
https://ejournal.ft-undar.ac.id/index.php/jeetech
DOI: https://doi.org/10.32492/jeetech.v7i1.7112
e-ISSN: 2722-5321
p-ISSN: 2964-7320

Prototipe AI-IoT Edge Berbasis Raspberry Pi dan TinyML untuk Pemantauan Jaringan Kampus secara Real-Time

[1] Bima Aulia Firmandani, [2*] F Yudi Limpraptono, [3] Michael Ardhita, [4]Machrus Ali
[1,2] Magister Teknik Elektro, Institut Teknologi Nasional, Malang,
[3] Teknik Elektro S1, Institut Teknologi Nasional Malang, Malang
[4]Teknik Elektro, Universitas Darul Ulum, Jombang
[1] bima@pustik.itn.ac.id, [2] fyudil@lecturer.itn.ac.id, [3] michael.ardita@lecturer.itn.ac.id, [4] machrus7@gmail.com

Article Info

Article history:

Received March 27, 2026
Revised April 19, 2026
Accepted May 09, 2026

Keyword:

NetFlow
TinyML
edge computing
autoencoder
Raspberry Pi

ABSTRACT

Complex campus networks, characterized by expanding server-based services and Internet of Things (IoT) features, require near real-time monitoring systems that do not generate significant overhead. This study proposes a lightweight AI-IoT network monitoring prototype deployed on an edge computing platform, utilizing an unsupervised autoencoder for anomaly detection. The prototype is implemented out-of-band on a Raspberry Pi 4 Model B, serving as both the NetFlow data collection and inference node. The deep learning model built on TensorFlow Lite is compressed via TinyML INT8 quantization for resource-constrained devices. Experiments were conducted on the CIC-IDS2017 benchmark dataset comprising 600,000 labeled network flows (70% normal, 30% anomaly), split 80:20 for training and testing with Min-Max normalization. P70 denotes the 70th-percentile reconstruction-error threshold (aggressive, threat-hunting mode), while P95 denotes the 95th-percentile threshold (conservative, low-alert-fatigue mode). At P70, the model achieved F1-score 0.60 (precision 0.96, recall 0.43); at P95, all false positives were eliminated (precision 1.00, recall 0.07, F1-score 0.14). Edge infrastructure demonstrated computational efficiency: average batch latency 74 ms, throughput >300 flows/s, RAM utilization 2.8%. These results confirm that the prototype is computationally viable on commodity hardware, though detection sensitivity at both thresholds remains limited and warrants further improvement.

Corresponding Author:
F Yudi Limpraptono,
Magister Teknik Elektro, Institut Teknologi Nasional Malang,
Email: fyudil@lecturer.itn.ac.id

Abastraks: Jaringan kampus yang kompleks dengan layanan berbasis server dan fitur Internet of Things (IoT) yang terus berkembang memerlukan sistem pemantauan mendekati real-time tanpa menimbulkan overhead yang signifikan. Penelitian ini mengusulkan prototipe pemantauan jaringan ringan berbasis AI-IoT pada platform edge computing dengan memanfaatkan unsupervised autoencoder untuk deteksi anomali (anomaly detection). Prototipe diimplementasikan secara out-of-band pada Raspberry Pi 4 Model B sebagai node pengumpul data NetFlow sekaligus node inferensi. Model deep learning berbasis TensorFlow Lite dikompres menggunakan kuantisasi TinyML INT8. Eksperimen dilakukan pada dataset benchmark CIC-IDS2017 yang terdiri atas 600.000 aliran (flow) berlabel (70% normal, 30% anomali), dengan pembagian latih-uji 80:20 dan normalisasi Min-Max. P70 merujuk pada ambang batas reconstruction error persentil ke-70 (mode agresif untuk threat hunting), sedangkan P95 merujuk pada persentil ke-95 (mode konservatif, meminimalkan alert fatigue). Pada P70: F1-score 0,60 (presisi 0,96, recall 0,43); pada P95: seluruh false positive tereliminasi (presisi 1,00, recall 0,07, F1-score 0,14). Infrastruktur edge efisien secara komputasi: latensi batch 74 ms,

throughput >300 flows/detik, utilisasi RAM 2,8%. Prototipe ini terbukti layak secara komputasi pada perangkat keras komoditas, meskipun sensitivitas deteksi pada kedua ambang batas masih terbatas dan perlu ditingkatkan lebih lanjut.

1. Pendahuluan

Transformasi digital telah mengubah paradigma dalam mengelola infrastruktur teknologi informasi di pendidikan tinggi. Kampus yang memiliki jaringan informasi dituntut untuk melakukan modernisasi atau upgrade yang mengakibatkan semakin banyaknya penggabungan jenis *endpoint* Internet of Things (IoT) baru dan aliran data yang semakin luas, sehingga secara drastis meningkatkan volume dan kompleksitas kepadatan lalu lintas jaringan [1] [2]. Sistem pemantauan jaringan kampus yang efektif harus mampu mendeteksi *packet loss*, lonjakan latensi, dan intrusi tidak sah secara lokal (*in situ*) dengan segera dalam waktu mendekati *real-time* (NRT). Seperti pada penelitian [3] membahas implementasi real-time network traffic monitoring untuk jaringan kampus yang memungkinkan deteksi pola lalu lintas, masalah keamanan, secara langsung.

Model yang menggunakan kondisi pemantauan jaringan yang masih didominasi oleh pendekatan terpusat memang menawarkan kemudahan pengelolaan, namun tidak terukur pasti dan menciptakan hambatan *bandwidth* karena harus mengirimkan data terus-menerus dari sisi klien ke server pusat [4], [5]. Penelitian ini juga menyimpulkan arsitektur yang memerlukan pemrosesan dan penyimpanan terdistribusi secara signifikan meningkatkan efisiensi dibandingkan pendekatan terpusat. Kondisi yang terbaca menunjukkan *edge computing* menyediakan paradigma terdistribusi dengan memindahkan pemrosesan data ke tepi jaringan. Pendekatan ini secara signifikan mengurangi latency dan konsumsi bandwidth sekaligus meningkatkan privasi data karena meminimalkan pengiriman informasi sensitif ke server pusat [6], [7], [8]. Bersamaan dengan itu, AI-IoT dan *Tiny Machine Learning* (TinyML) memberikan peluang untuk menjalankan model *deep learning* pada komputer papan tunggal (*single-board computer*) yang memiliki keterbatasan sumber daya. TinyML mengubah arsitektur AI tradisional dengan menjalankan model machine learning pada microcontroller units dengan sumber daya sangat terbatas (kurang dari 1 MB flash memory) . Penelitian ini mengidentifikasi bahwa TinyML menawarkan latensi mendekati nol (0–5 ms) untuk tugas inferensi, mengurangi ketergantungan pada cloud, serta meningkatkan privasi dan keamanan data [9]. Banyak penelitian terkait Tiny Machine Learning (TinyML) telah muncul sebagai subbidang spesialisasi machine learning yang fokus pada deployment model pada perangkat dengan sumber daya sangat terbatas, seperti mikrokontroler dengan kapasitas memori kurang dari 1 MB [10], [11], [12]. Model pembelajaran tidak terawasi (*unsupervised learning*), khususnya *autoencoder*, cukup kuat untuk meningkatkan jalur pemodelan dan jaringan saraf dapat mengindeks perilaku normal jaringan secara efektif. Beberapa penelitian yang telah dilakukan seperti deteksi anomali lalu lintas jaringan menggunakan pendekatan unsupervised machine learning, mempelajari pola normal dari data secara tidak terawasi, dan memanfaatkan kemampuan unsupervised learning dan feature extraction dari Denoising Autoencoder (DAE) untuk membangun sistem deteksi intrusi real-time pada jaringan IoT [13], [14], [15], [16].

Beberapa penelitian seperti [17], [18], [19] telah menyelidiki arsitektur *edge* atau *deep learning* untuk IoT secara terpisah, namun sangat sedikit yang membahas lanjut tentang teori perangkat keras yang bekerja dengan sumber daya terbatas. Berawal dari hal tersebut, penelitian ini memperkenalkan kebaruan melalui tiga faktor: (1) *autoencoder* TinyML yang sangat terkuantisasi, (2) ambang batas numerik berbasis persentil dinamis (P95/P70), dan (3) penanganan *out-of-band* pada transit NetFlow institusi. Penelitian ini juga menjelaskan pendekatan tiga faktor yang dilakukan serta evaluasi akurasi yang dilakukan untuk melakukan deteksi *end-to-end* serta kemampuan memberikah hasil komputasi pada node *edge* Raspberry Pi fisik melalui prototipe AI-IoT *edge computing* yang ringan.

1. Tinjauan Pustaka

Perkembangan penelitian terbaru terkait pemantauan jaringan terdistribusi dan sistem deteksi intrusi berbasis kecerdasan buatan telah menunjukkan pergeseran fokus menuju arsitektur komputasi *edge*. Implementasi arsitektur *edge* [20]. Penelitian ini terbukti secara empiris mampu mereduksi latensi transmisi secara signifikan dibandingkan jaringan terpusat. Dalam domain keamanan siber, pemanfaatan model *unsupervised*, khususnya *autoencoder*, menunjukkan efektivitas yang tinggi untuk deteksi intrusi [21]. Penelitian dengan melakukan pendekatan yang memformulasikan batas keputusan matematis yang tangguh untuk membedakan lalu lintas normal dan anomali intrusi siber telah dilakukan [22]

Dalam beberapa tahun terakhir, literatur menyajikan beragam pemodelan mutakhir untuk analisis data berbasis aliran. Solusi yang diusulkan mencakup implementasi *Graph Neural Networks* (GNN) dan *Adaptive Robust Autoencoders* [23], [24], [25]. Selain itu, kerangka kerja spesifik seperti EcoDefender dan SAE-FF diklaim mampu mencapai metrik akurasi melampaui 93% untuk jaringan IoT [21], [26]. Pada ranah infrastruktur fisik, kelayakan komputasi perangkat berbiaya rendah seperti Raspberry Pi telah divalidasi kemampuannya dalam mendukung sistem *real-time* [27]. Terkait kompresi jaringan saraf, pada [9] berhasil mengimplementasikan inferensi TinyML terkuantisasi secara laten pada *edge*, sementara penelitian lain [10], [11] mengeksplorasi pembelajaran mesin daring yang agnostik terhadap inspeksi *pay-*

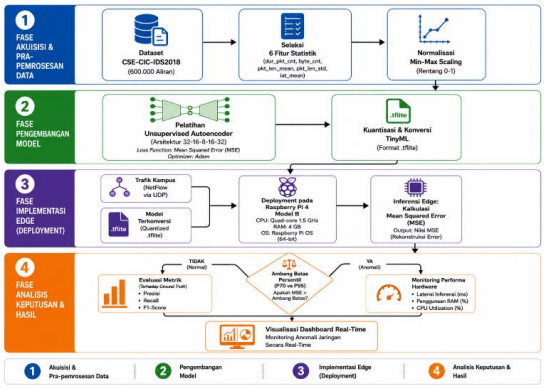
load. Survei komprehensif mengenai AI pada jaringan terbatas juga telah banyak dilakukan. Meskipun demikian, tinjauan analitis terhadap literatur mengungkap sebuah celah fundamental: arsitektur berkinerja tinggi (seperti GNN dan EcoDefender) umumnya masih mensyaratkan ketergantungan komputasi pada klaster *edge-server* yang berat, atau secara implisit mengeksploitasi fitur dari *payload*. Penelitian ini mengusulkan resolusi terhadap celah tersebut melalui implementasi *autoencoder* TinyML berbasis NetFlow murni yang dioptimalkan menggunakan pendekatan fungsi kuantil matematika.

Kerangka Konseptual dan Definisi Variabel Aliran (NetFlow) Guna mencapai efisiensi inferensi secara ekstrem pada perangkat *edge*, sistem deteksi anomali yang diusulkan beroperasi sepenuhnya melalui analisis perilaku (*behavioral analysis*) terhadap dinamika metadata aliran jaringan (NetFlow), tanpa melakukan inspeksi terhadap *payload*. Secara operasional, alur konseptual pemrosesan anomali dibangun melalui lima tahapan utama:

1. Ekstraksi Pasif: *Gateway* mentransformasi lalu lintas jaringan mentah menjadi instrumen metrik aliran secara pasif, yang selanjutnya dikirimkan secara *out-of-band* menuju *node edge* (Raspberry Pi).
2. Rekayasa Fitur (*Feature Engineering*): Enam atribut statistik aliran diekstrak dan dinormalisasi menggunakan *Min-Max Scaler* untuk memastikan distribusi data fitur konvergen pada rentang skala (0,1).
3. Inferensi Terkuantisasi: *Autoencoder* berbasis TinyML pada kerangka kerja TensorFlow Lite memproses matriks fitur ke dalam ruang laten untuk melakukan rekonstruksi data.
4. Kalkulasi Kerugian (*Loss Calculation*): Diferensiasi kuantitatif antara tensor data *input* asli dan data rekonstruksi dihitung secara matematis menggunakan metrik *Mean Squared Error* (MSE).
5. Keputusan Deterministik: Klasifikasi anomali dieksekusi dengan mengevaluasi nilai MSE terhadap ambang batas (*threshold*) persentil dinamis (P70/P95). Deviasi rekonstruksi yang melampaui batas kuantil tersebut memicu eskalasi klasifikasi aliran sebagai anomali intrusi.

2. Metodologi

Penelitian ini menggunakan desain eksperimen kuasi-kuantitatif melalui pengembangan dan pengujian prototipe pemantauan *edge* yang disebut *netmon*. Alur eksperimen secara komprehensif mencakup ekstraksi data, pra-pemrosesan, pelatihan model komputasi, kuantisasi, hingga inferensi secara *real-time*. Tahapan alur eksperimen dari ekstraksi data hingga analisis keputusan anomali diilustrasikan pada Gambar1.



Gambar 1. Diagram Alir Prototipe AI IoT Deteksi Anomali Jaringan

Konfigurasi Perangkat Keras dan Perangkat Lunak Infrastruktur pengujian dibangun menggunakan perangkat *gateway* yang dikonfigurasi untuk mengekspor lalu lintas jaringan mentah menjadi aliran NetFlow/IP-FIX secara pasif. Data aliran tersebut kemudian ditransmisikan menuju *node edge* inferensi, yakni Raspberry Pi 4 Model B (RAM 8 GB), melalui protokol UDP. Konfigurasi perangkat lunak pada tahap pengembangan model menggunakan bahasa pemrograman Python yang terintegrasi dengan pustaka TensorFlow dan Keras. Untuk mengekstraksi aliran jaringan secara agnostik terhadap *payload*, instrumen perangkat lunak *CICFlowMeter* digunakan. Model algoritma yang telah dilatih kemudian dikonversi dan dieksekusi pada Raspberry Pi menggunakan *interpreter* TensorFlow Lite (TFLite) untuk menyimulasikan arsitektur eksekusi TinyML. Seluruh hasil komputasi anomali secara *real-time* diekspor dan divisualisasikan pada *dashboard* situasional berbasis web.

Sumber Dataset dan Pra-pemrosesan Evaluasi prototipe menggunakan dataset publik deteksi intrusi komprehensif, yaitu CSE-CIC-IDS2018. Dataset ini secara *native* berformat data aliran (*flow-based*) sehingga sejalan dengan batasan operasional agnostik terhadap *payload*. Dilakukan *sampling* terarah guna mengekstraksi subsampel berjumlah 600.000 aliran lalu lintas. Rincian distribusi kelas dataset untuk merepresentasikan kondisi ketidakseimbangan kelas (*imbalanced data*) diuraikan pada Tabel 1.

Tabel 1. Metrik Evaluasi untuk Ambang Batas P95 dan P70

Kelas Lalu Lintas	Jumlah Aliran (Flows)	Presentase	Keterangan
Normal (Kelas 0)	200,000	33,3%	Mewakili lalu lintas institusi yang aman
Anomali (Kelas 1)	400,000	66,7%	Mewakili kondisi intrusi/serangan siber
Total Dataset Uji	600,000	100%	Digunakan untuk pelatihan (80%) dan pengujian (20%)

Dataset dibagi menjadi himpunan pelatihan dan pengujian dengan rasio 80:20, direproduksi secara deterministik menggunakan *random seed* 42. Sebelum pelatihan, matriks enam fitur direkayasa menggunakan *Min-Max Scaler* agar rentang nilai absolut terdistribusi (0, 1), selaras dengan fungsi aktivasi Sigmoid pada lapisan *output*.

Arsitektur Autoencoder dan Validasi Kuantisasi TinyML Optimasi arsitektur jaringan saraf dibentuk menggunakan model *symmetric bottleneck* dengan lima lapisan padat (32-16-8-16-32 neuron), di mana lapisan tengah bertindak sebagai penyandi ruang laten (*latent space*). Fungsi aktivasi *Rectified Linear Unit* (ReLU) digunakan untuk memitigasi *vanishing gradient*. Di sisi lain, lapisan terluar menggunakan aktivasi Sigmoid untuk menstabilkan konvergensi fungsi kerugian *Mean Squared Error* (MSE). Bobot parametrik dari model *baseline* dikompresi menjadi format .tflite. Kualitas instrumen divalidasi dengan membandingkan galat rekonstruksi antara model mentah dan model terkuantisasi. Hasil observasi mengonfirmasi bahwa degradasi performa kompresi berada pada level marjinal (selisih peningkatan MSE < 0,001), membuktikan proses kuantisasi TFLite reliabel untuk mendeteksi anomali.

Penentuan Ambang Batas dan Metrik Evaluasi Batas keputusan (*decision boundary*) diekstraksi secara matematis melalui fungsi kuantil dari distribusi kerugian rekonstruksi (MSE) pada fase pengujian data pelatihan normal. Eksperimen dirancang untuk mengevaluasi dua skenario persentil dinamis:

1. **P95 (Persentil ke-95):** Fungsi pendekatan konservatif. Objektif utamanya adalah mereduksi *False Positive* ke tingkat absolut (nol) guna memitigasi kelelahan akibat peringatan (*alert fatigue*) pada administrator.
2. **P70 (Persentil ke-70):** Fungsi pendekatan agresif guna memfasilitasi kapabilitas perburuan ancaman (*threat hunting*) terhadap anomali tersembunyi.

Mengingat tipologi data siber sangat tidak seimbang, metrik **Presisi (*Precision*)** digunakan untuk memvalidasi keandalan absolut peringatan, sedangkan **Sensitivitas (*Recall*)** mengukur persentase cegatan ancaman nyata. Metrik **F1-Score** diaplikasikan sebagai penentu kualitas utama yang memberikan rata-rata harmonik paling objektif untuk mengukur pertukaran strategis (*trade-off*) operasional prototipe.

3. Hasil dan Diskusi

Prototipe diuji menggunakan subsampel berlabel (600.000 aliran) untuk mengevaluasi kemampuan deteksi anomali dan performa komputasi fisiknya.

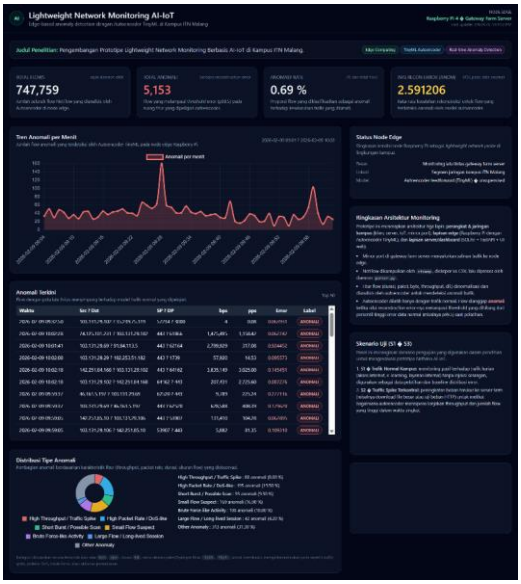
Tabel 2. Metrik Evaluasi untuk Ambang Batas P95 dan P70

No	Parameter	Nilai P95	Nilai P70
1	True Negative (TN)	200,000	192,849
2	False Positive (FP)	0	7,151
3	False Negative (FN)	370,000	227,151
4	True Positive (TP)	30,000	172,849
5	Presisi	1.00 (100%)	0.96 (96%)
6	Recall	0.07 (7%)	0.43 (43%)
7	F1-Score	0.14	0.60

Seperti disajikan pada Tabel 2, ambang batas P95 berhasil menghilangkan alarm palsu sepenuhnya (FP = 0) dengan tingkat presisi 100%. Sebaliknya, skenario P70 beroperasi lebih sensitif, meningkatkan penangkapan ancaman (*recall*) menjadi 43% dan mencetak nilai *F1-Score* sebesar 0,60. Dibandingkan dengan kerangka komputasi *edge* terbaru seperti EcoDefender (~94%) atau SAE-FF (~93%), *F1-score* 0,60 mungkin secara sekilas terlihat inferior. Namun, tinjauan analitis memperlihatkan bahwa EcoDefender sangat bergantung pada inspeksi fitur *payload* (isi paket) yang mendalam, dan lazimnya dieksekusi pada *edge-server* berspesifikasi tinggi. Sebaliknya, model ini dilatih di bawah batasan ekstrem: sepenuhnya agnostik terhadap *payload* (hanya berdasarkan NetFlow), beroperasi *unsupervised*, dan berjalan di atas arsitektur TinyML Raspberry Pi.

Meskipun efisien secara komputasi, sensitivitas deteksi sistem ini secara fundamental masih terbatas. Tingginya angka *false negative* (terutama pada P95) disebabkan oleh ketidakmampuan data metadata NetFlow murni untuk mendeteksi indikasi intrusi siber halus yang jejaknya hanya terdapat pada level *payload* paket. Implementasi kedua ambang batas tersebut secara gamblang mendemonstrasikan sebuah *trade-off* operasional: administrator harus mengorbankan keamanan absolut (presisi 100%) apabila ingin memburu lebih banyak ancaman (P70), yang diiringi dengan kemunculan peringatan palsu.

Hasil analisis diteruskan ke *dashboard* web interaktif untuk memberikan kesadaran situasional (*situational awareness*) secara waktu nyata.



Gambar 2. Real-time network monitoring dashboard interface

Kelayakan penggunaan perangkat keras Raspberry Pi 4 Model B telah diuji untuk berbagai interval lalu lintas yang berbeda. Proses monitoring jaringan secara realtime ditunjukkan pada gambar 2. Dari dashboard tersebut kita dapat mengetahui informasi jaringan yang kita perlukan.

Tabel 3. Log Performa Pemrosesan Edge

No	Parameter	Values
1	Average Batch Latency	0.074 s
2	Average Throughput	≈ 300 flows/s
3	Peak Throughput Handling	≈ 521 flows/s
4	Average RAM Usage	2.8%

Kelayakan penggunaan perangkat keras divalidasi pada Tabel 3, di mana inferensi Raspberry Pi 4 Model B mencatat rata-rata latensi pemrosesan sangat rendah sebesar 74 milidetik. Secara operasional, perangkat dengan mudah menangani rata-rata 300 aliran/detik, bahkan sanggup memproses lonjakan puncak 521 aliran/detik tanpa hambatan antrean, ditunjang oleh pemanfaatan RAM yang stabil pada 2,8%.

Keterbatasan Penelitian*:* Validasi implementasi dalam penelitian ini masih terbatas pada penggunaan subsampel dataset statis dan evaluasi arsitektur agnostik *payload*. Sistem memiliki risiko *false negative* yang tinggi pada jenis serangan non-volumetrik, dan belum teruji generalisasinya pada keragaman topologi fisik jaringan institusi di luar lingkungan pengujian.

4. Kesimpulan

Penelitian ini telah berhasil memenuhi tujuan merancang prototipe pemantauan *edge* AI-IoT dan mengevaluasi kinerja infrastrukturnya secara terukur berdasarkan indikator perangkat lunak maupun perangkat keras. Berdasarkan hasil evaluasi, secara teknis, penelitian ini berkontribusi membuktikan bahwa kompresi model *autoencoder* TinyML dapat dieksekusi secara sangat efisien pada Raspberry Pi murni, yang mampu menghilangkan hambatan (*bottleneck*) arsitektur terpusat dengan mencapai latensi inferensi 74 ms dan pemanfaatan RAM yang stabil pada 2,8%. Secara

praktis, penerapan ambang batas dinamis berbasis persentil memberikan administrator jaringan sebuah kendali yang fleksibel terhadap *trade-off* operasional sistem: batas P95 dapat digunakan untuk menjamin presisi absolut (tanpa alarm palsu), sementara batas P70 dapat diaktifkan untuk kapabilitas perburuan ancaman (*threat hunting*). Keterbatasan utama dari pendekatan ini terletak pada sensitivitas deteksi (*recall*) yang secara fundamental masih terbatas akibat tingginya angka *false negative*, yang merupakan konsekuensi logis dari ketergantungan pada metadata NetFlow murni tanpa melakukan inspeksi isi paket (*payload*).

Untuk memitigasi keterbatasan tersebut dan memperluas skala implementasi, agenda penelitian di masa mendatang direkomendasikan mencakup beberapa fokus pengembangan berikut:

1. Pengujian Lingkungan Nyata: Evaluasi prototipe secara langsung pada lalu lintas jaringan institusi/kampus nyata (*in situ*) secara *real-time*.
2. Pengembangan Fitur: Penambahan ekstraksi fitur turunan temporal guna menangkap korelasi deret waktu anomali antar aliran data.
3. Komparasi Model: Pembandingan metrik kinerja prototipe terhadap arsitektur dasar (*baseline*) yang menggunakan pendekatan pembelajaran terawasi (*supervised learning*).
4. Otomatisasi Ambang Batas: Integrasi algoritma adaptif yang mampu menyesuaikan nilai persentil *threshold* (seperti P70/P95) secara otomatis berdasarkan fluktuasi kondisi jaringan.
5. Validasi Ancaman: Evaluasi dan validasi ketahanan deteksi model terhadap ragam ancaman siber spesifik yang lebih mutakhir dan kompleks.

Daftar Pustaka

[1] J. Li, N. B. Linsangan, and H. Dong, "Campus Network Traffic Prediction and Anomaly Detection Based on Deep Learning," *Int. J. Emerg. Technol. Adv. Appl.*, vol. 1, no. 7, pp. 8–13, Aug. 2024, doi: 10.62677/IJETAA.2407123.

[2] R. Gutierrez, W. Villegas-Ch, and J. Govea, "Modular middleware for IoT: scalability, interoperability and energy efficiency in smart campus," *Front. Commun. Netw.*, vol. 6, p. 1672617, Sep. 2025, doi: 10.3389/frcmn.2025.1672617.

[3] A. Fathima and G. S. Devi, "Enhancing university network management and security: a real-time monitoring, visualization & cyber attack detection approach using Paessler PRTG and Sophos Firewall," *Int. J. Syst. Assur. Eng. Manag.*, Aug. 2024, doi: 10.1007/s13198-024-02448-y.

[4] R. M. Oviedo, F. Ramos, S. Gormus, P. Kulkarni, and M. Sooriyabandara, "A Comparison of Centralized and Distributed Monitoring Architectures in the Smart Grid," *IEEE Syst. J.*, vol. 7, no. 4, pp. 832–844, Dec. 2013, doi: 10.1109/JSYST.2013.2246033.

[5] Xiaojiang Du, "Toward efficient distributed network monitoring," in *IEEE International Conference on Performance, Computing, and Communications, 2004.*, Phoenix, AZ, USA: IEEE, 2004, pp. 87–94. doi: 10.1109/PCCC.2004.1394950.

[6] F. Zhou, M. Yuan, Y. Liu, H. Zhang, M. Gu, and T. Zhou, "Niect: A Model for Intrusion Security Detection Applied to Campus Video Surveillance Edge Networks," in *2024 IEEE 11th International Conference on Cyber Security and Cloud Computing (CSCloud)*, Shanghai, China: IEEE, Jun. 2024, pp. 24–29. doi: 10.1109/CSCloud62866.2024.00012.

[7] S. Hussain *et al.*, "Edge AI-based self-learning technique for mitigating DDoS attacks in WSN," *Comput. Netw.*, vol. 273, p. 111769, Dec. 2025, doi: 10.1016/j.comnet.2025.111769.

[8] P. V. Sithole and T. Justice Lavhengwa, "Exploring theories towards deploying edge computing in South African Higher Education Institutions," in *2025 Conference on Information Communications Technology and Society (ICTAS)*, Durban, South Africa: IEEE, Jul. 2025, pp. 1–5. doi: 10.1109/ICTAS64866.2025.11155306.

[9] S. Heydari and Q. H. Mahmoud, "Tiny Machine Learning and On-Device Inference: A Survey of Applications, Challenges, and Future Directions," *Sensors*, vol. 25, no. 10, p. 3191, May 2025, doi: 10.3390/s25103191.

[10] H.-A. Rashid, U. Kallakuri, and T. Mohsenin, "TinyM^2 Net-V2: A Compact Low-power Software Hardware Architecture for M ulti m odal Deep Neural Networks," *ACM Trans. Embed. Comput. Syst.*, vol. 23, no. 3, pp. 1–23, May 2024, doi: 10.1145/3595633.

[11] J. Leslin, M. Trapp, and M. Andraud, "Hardware-efficient tractable probabilistic inference for TinyML Neurosymbolic AI applications," in *2025 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, Madison, WI, USA: IEEE, Aug. 2025, pp. 1–6. doi: 10.1109/COINS65080.2025.11125733.

[12] J. D. Velasquez, L. Cadavid, and C. J. Franco, "Emerging trends and strategic opportunities in tiny machine learning: A comprehensive thematic analysis," *Neurocomputing*, vol. 648, p. 130746, Oct. 2025, doi: 10.1016/j.neucom.2025.130746.

[13] A. Vikram and Mohana, "Anomaly detection in Network Traffic Using Unsupervised Machine learning Approach," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India: IEEE, Jun. 2020, pp. 476–479. doi: 10.1109/ICCES48766.2020.9137987.

[14] Dj. K. Nkashama *et al.*, "ANADOE: Auto-Encoder-Based Network Anomaly Detection with Outlier Exposure," Mar. 12, 2025. doi: 10.36227/techrxiv.174181618.89242519/v1.

[15] L. A. K. Mekemte and G. Chalhoub, "On the Use of Autoencoders in Unsupervised Learning for Intrusion Detection Systems," in *Ubiquitous Networking*, vol. 14757, O. Habachi, G. Chalhoub, H. Elbiaze, and E. Sabir, Eds., in Lecture Notes in Computer Science, vol. 14757. , Cham: Springer Nature Switzerland, 2024, pp. 54–69. doi: 10.1007/978-3-031-62488-9_5.

[16] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. Iqbal Khan, and M. Helal, "Intrusion Detection in IoT Systems Using Denoising Autoencoder," *IEEE Access*, vol. 12, pp. 122401–122425, 2024, doi: 10.1109/ACCESS.2024.3451726.

[17] M. Jagarajan and R. Jayaraman, "IoT edge computing and deep learning analytics: A survey," presented at the 4TH INTERNATIONAL CONFERENCE ON INTERNET OF THINGS 2023: ICIoT2023, Kattankalathur, India, 2024, p. 020283. doi: 10.1063/5.0217187.

[18] M. A. Ali and F. Dornaika, "Edge Artificial Intelligence: A Systematic Review of Evolution, Taxonomic Frameworks, and Future Horizons," Oct. 01, 2025, *arXiv*: arXiv:2510.01439. doi: 10.48550/arXiv.2510.01439.

[19] T. K. S. Flores, D. G. Costa, and I. Silva, "TensorFlores: An enhanced Python-based TinyML framework," *SoftwareX*, vol. 31, p. 102224, Sep. 2025, doi: 10.1016/j.softx.2025.102224.

[20] S. Wei, "Edge-Based Real-Time IIoT Anomaly Detection Using Semi-Supervised CNN-Attention Model with Cross-Protocol Capabilities," *Informatica*, vol. 49, no. 4, Dec. 2025, doi: 10.31449/inf.v49i4.10508.

[21] M. Y. Jo and H. J. Kim, "A Comparative Study of Lightweight, Sparse Autoencoder-Based Classifiers for Edge Network Devices: An Efficiency Analysis of Feed-Forward and Deep Neural Networks," *Sensors*, vol. 25, no. 20, p. 6439, Oct. 2025, doi: 10.3390/s25206439.

[22] S. Jeon, C. Park, G. Lee, S. Kim, and B. Gu, "Threshold Determination Method in Anomaly Detection using LSTM Autoencoder," *J. Korean Inst. Inf. Technol.*, vol. 21, no. 4, pp. 21–30, Apr. 2023, doi: 10.14801/jkiit.2023.21.4.21.

[23] H. Zhang and T. Cao, "A Hybrid Approach to Network Intrusion Detection Based On Graph Neural Networks and Transformer Architectures," in *2024 14th International Conference on Information Science and Technology (ICIST)*, Chengdu, China: IEEE, Dec. 2024, pp. 574–582. doi: 10.1109/ICIST63249.2024.10805457.

[24] Azyk Orozonova, "Ai-Driven Anomaly Detection in Iot Networks Using Advanced Machine Learning Techniques," *Int. J. Appl. Math.*, vol. 38, no. 11s, pp. 1657–1671, Nov. 2025, doi: 10.12732/ijam.v38i11s.1277.

[25] L. Van Langendonck, I. Castell-Uroz, and P. Barlet-Ros, "PPT-GNN: A Practical Pretrained Spatio-Temporal Graph Neural Network for Network Security," in *2025 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, Venice, Italy: IEEE, Jun. 2025, pp. 169–175. doi: 10.1109/EuroSPW67616.2025.00026.

[26] S. Jamshidi, F. Erfan, O. Abdul-Wahab, M. Bellaiche, and F. Khomh, "Lightweight Autoencoder-Isolation Forest Anomaly Detection for Green IoT Edge Gateways," 2025, *arXiv*. doi: 10.48550/ARXIV.2511.18235.

[27] M. B. Musthafa, S. Huda, T. T. Nguyen, Y. Kodera, and Y. Nogami, "Optimized Ensemble Deep Learning for Real-Time Intrusion Detection on Resource-Constrained Raspberry Pi Devices," *IEEE Access*, vol. 13, pp. 113544–113556, 2025, doi: 10.1109/ACCESS.2025.3584373.