Optimalisasi Keamanan IoT dan Edge Computing Menggunakan Model Machine Learning

Florentina Tatrin Kurniati[1], Lenny Margaretta Huizen[2]
[1]Institut Teknologi dan Bisnis STIKOM Bali, [2]Universitas Semarang
e-mail: [1]florent@stikom-bali.ac.id, [2]lenny@usm.ac.id

Diajukan: 14 Desember 2022; Direvisi: 16 Januari 2023; Diterima: 19 Januari 2023

Abstrak

Things (IoT) telah meningkat pesat berkat revolusi digital dan membawa tantangan keamanan yang signifikan. Pengoptimalan keamanan IoT pada edge computing dengan menerapkan model berbasis machine learning, untuk deteksi dan identifikasi. Metodologi yang digunakan meliputi pengumpulan data dari sensor IoT dan log aktivitas sebagai data, pra-pemrosesan data, serta pelatihan dan validasi model machine learning. Pada penelitian ini, deteksi dan identifikasi serangan menggunakan empat algoritma, yaitu K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forest (RF), dan Decision Trees (DT). Hasil penelitian menunjukkan bahwa model Random Forest (RF) dan Decision Tree (DT) memiliki kinerja terbaik dalam mendeteksi serangan siber, dengan nilai True Positive (TP) yang tinggi dan tingkat kesalahan yang rendah. Evaluasi kinerja berdasarkan metrik Akurasi, Presisi, Recall, dan F1-Score mengonfirmasi bahwa RF dan DT mampu memberikan hasil yang akurat dan andal dalam mendeteksi ancaman. Model Random Forest menunjukkan Akurasi 98,4%, Presisi 98,4%, Recall 83,9%, dan F1-Score 90,5%, sedangkan Decision Tree menunjukkan Akurasi 98,1%, Presisi 90,5%, Recall 83,9%, dan F1-Score 87,1%. Implementasi model machine learning dalam sistem keamanan IoT dan edge computing terbukti tidak hanya meningkatkan keamanan data dan perangkat, tetapi juga memaksimalkan efisiensi operasional dengan kemampuan untuk mempelajari dan beradaptasi dengan pola serangan baru.

Kata kunci: *IoT, Edge computing, Machine learning, Deteksi serangan siber.*

Abstract

The use of Internet of Things (IoT)-based technologies has rapidly increased due to the digital revolution, bringing significant security challenges. This study focuses on optimizing IoT security on edge computing by applying machine learning-based models to detect and identify cyber-attacks. The methodology includes data collection from IoT sensors and security incident logs, data pre-processing, and machine learning model training and validation. This research employs four algorithms for attack detection and identification: K Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forest (RF), and Decision Trees (DT). The results show that the Random Forest (RF) and Decision Tree (DT) models exhibit the best performance in detecting cyber attacks, with high True Positive (TP) values and low error rates. Performance evaluation based on Accuracy, Precision, Recall, and F1-Score metrics confirms that RF and DT can provide accurate and reliable results in threat detection. The Random Forest model demonstrates 98.4% Accuracy, 98.4% Precision, 83.9% Recall, and 90.5% F1-Score, while the Decision Tree model shows 98.1% Accuracy, 90.5% Precision, 83.9% Recall, and 87.1% F1-Score. The implementation of machine learning models in IoT and edge computing security systems has proven to not only enhance data and device security but also maximize operational efficiency, with the ability to learn and adapt to new attack patterns.

Keywords*: IoT, Edge computing, Machine learning, Cyber attack detection.*

1. Pendahuluan

Teknologi berbasis Internet of Things (IoT) telah mengalami perkembangan yang signifikan dan kini menjadi komponen penting bagi revolusi teknologi digital [1]-[4]. Perangkat IoT, yang terintegrasi dengan sensor, mikrokontroler, aktuator, dan sistem jaringan, telah diadopsi secara luas di berbagai sektor, termasuk keamanan, pertanian, dan sistem transaksi. Pada perangkat IoT, data yang dikumpulkan oleh sensor diproses secara lokal menggunakan teknologi edge computing, yang bertujuan untuk mengurangi latensi, meningkatkan responsivitas, dan meminimalkan beban pada infrastruktur cloud[5], [6]. Integrasi

antara IoT dan edge computing ini terbukti meningkatkan kecepatan dan efisiensi dalam pengumpulan, pemrosesan, dan analisis data[7], [8]. Namun, perkembangan teknologi ini sering kali tidak diikuti oleh peningkatan yang setara dalam aspek keamanan data, yang pada gilirannya memunculkan risiko serangan, virus, dan ancaman lainnya. Untuk mengatasi tantangan ini, model pembelajaran mesin telah diterapkan untuk mengenali pola serangan dan perilaku abnormal pada data yang dihasilkan oleh perangkat IoT, sehingga memungkinkan deteksi ancaman dan identifikasi risiko secara dini. Model keamanan yang berbasis machine learning memanfaatkan data trafik jaringan, log aktifitas, yang dimodelkan menggunakan algoritma klasifikasi K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forest (RF), dan Decision Trees [9], [10].

Penggunaan model klasifikasi K-Nearest Neighbors (KNN) berbasis machine learning mendeteksi serangan dengan membandingkan data trafik baru terhadap sampel data yang sudah dikenali sebagai serangan atau normal. Algoritma ini efektif untuk mendeteksi serangan yang mirip dengan pola dalam database, namun dapat mengalami kesulitan ketika menghadapi serangan baru yang belum pernah terjadi sebelumnya. Untuk model Support Vector Machines (SVM) beroperasi dengan cara yang berbeda, SVM mencari hyperplane yang paling efektif untuk memisahkan data antara dua kategori, serangan dan tidak serangan. SVM sangat berguna dalam menangani data dengan dimensi tinggi.

Sedangkan untuk model klasifikasi pada Random Forest (RF) mengambil pendekatan ensemble, menggunakan banyak decision trees untuk membuat keputusan. Dengan memanfaatkan kekuatan agregasi dari berbagai trees, RF cenderung lebih robust terhadap noise dan overfitting. Ini sangat efektif dalam mendeteksi berbagai pola serangan dengan meminimalkan risiko kesalahan karena variasi data. Namun demikian penggunaan model klasifikasi dengan Decision Trees, sebagai komponen dasar dari Random Forest, juga dapat digunakan secara mandiri. Metode ini mengklasifikasikan data dengan membuat serangkaian keputusan berdasarkan nilai atribut, membuatnya mudah untuk diinterpretasi [11].

2. Metode Penelitian

Model deteksi serangan siber pada jaringan IoT dan edge computing dikembangkan menggunakan sistem berbasis machine learning. Dengan metode klasifikasi K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forest (RF), dan Decision Trees. Untuk klasifikasi KNN bergantung pada pengukuran jarak pada proses mengklasifikasikan data baru[12]. Menghitung kedekatan berdasarkan K, untuk menghitung jarak antar titik menggunakan Euclidean ditunjukkan pada persamaan (1).

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (xi - yi)^2} \tag{1}$$

di mana x dan y adalah dua titik dalam ruang n-dimensi dan $d(x,y)$ adalah jarak Euclidean. Klasifikasi dengan SVM untuk menemukan hyperplane margin antara dua kelas. Dengan persamaan (2).

$$w \cdot x + b = 0 \tag{2}$$

di mana w adalah vektor bobot, x adalah vektor input, dan b adalah bias. SVM mencari w dan b yang minimal dengan mengatur vektor bobot ditunjukkan pada persamaan (3).

$$|W|^2 \tag{3}$$

dengan kendala bahwa semua sampel data harus berada di sisi yang benar dari margin, untuk setiap sampel data ($\mathbf{x}i$,yi), ditunjukkan pada persamaan (4)

$$yi(\mathbf{w} \cdot \mathbf{x}i + b) \geq 1 \tag{4}$$

Untuk klasifikasi Random Forest menggabungkan setiap tree dirandomisasi dan dibuat berdasarkan subset acak dari data atau fitur, ditunjukkan pada persamaan (5).

$$RF(x)=\text{modus}(\{Tree1(x),Tree2(x),\ldots,Tree n(x)\}) \quad (5)$$

di mana Tree (x) adalah prediksi dari pohon ke-i pada input x.

Klasifikasi Decision Trees membuat keputusan dengan menghasilkan serangkaian pertanyaan biner yang dibuat dari fitur-fitur data. Setiap node dalam pohon memutuskan fitur mana yang akan dibagi berdasarkan metrik seperti entropi atau indeks Gini. Misalnya, untuk indeks Gini, persamaan (6) digunakan untuk memilih fitur.

$$G = 1 - \sum_j p_j^2 \quad (6)$$

di mana p_j adalah proporsi sampel untuk kelas j di node tersebut. Pohon memilih pemisahan yang mengurangi nilai Gini yang tertimbang di cabang-cabangnya.

Matriks konfusi (confusion matrix) ditunjukkan pada Tabel 1 menggambarkan kinerja model. Matriks konfusi biasanya memiliki dua dimensi, Aktual dan Prediksi, dengan setiap dimensi memiliki Positif dan Negatif sebagai kategori.

Tabel 1. Confusion Matrix.

Prediksi Positif	Prediksi Negatif	Prediksi Negatif
Aktual Positif	False Negative (FN)	False Negative (FN)
Aktual Negatif	True Negative (TN)	True Negative (TN)

Dimana untuk True Positive (TP) model memprediksi positif dan aktualnya juga positif. False Positive (FP) model memprediksi positif tetapi aktualnya negatif. True Negative (TN) model memprediksi negatif dan aktualnya juga negatif. Sedangkan False Negative (FN)cmodel memprediksi negatif tetapi aktualnya positif. Dari matriks konfusi, dapat menghitung beberapa metrik penting yang membantu mengevaluasi kinerja model yaitu akurasi, presisi, recall dan F1-Score. Untuk mengetahui nilai akurasi model yang ditunjukkan pada Persamaan (7) dengan mengukur proporsi total prediksi yang benar (baik positif maupun negatif), mengevaluasi model klasifikasi, mengukur seberapa sering model benar dalam semua prediksi.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

Untuk presisi atau mengukur nilai prediksi positif seperti ditunjukkan pada Persamaan (8), mengukur yang benar-benar positif.

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (8)$$

Recall seperti ditunjukkan pada Persmaan (9) mengukur proporsi positif aktual yang berhasil diidentifikasi oleh model.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (9)$$

F1-Score ditunjukkan pada Persamaan (10) menggabungkan presisi dan recall dalam bentuk harmonic mean untuk memberikan keseimbangan antara keduanya.

$$F1 - Score = 2 * \frac{Presisi * Recall}{Presisi + Recall} \quad (10)$$

Data trafik yang diperoleh terdapat aktifitas normal dan serangan, dengan menggunakan klasifikasi berbasis machine learning model disusun untuk mendeteksi serangan siber. Sebagian data digunakan sebagai data latih dan sebagian digunakan sebagai data uji peseudocode model deteksi serangan pada IoT dan Edge Computing ditunjukkan Tabel 2.

Tabel 2. Pseudocode Model Keamanan IoT dan Edge Computing.

```
Algorithm:  Enhancing IoT and Edge Computing Security with Machine Learning
Inputs:     - IoT sensors data from various sectors
            - Security incident logs
Outputs:    - Optimized machine learning model for threat detection and prevention
Process :   Begin
              Function CollectData()
                return gather IoT data and security logs
              EndFunction
              Function PreprocessData(data)
                return normalize and reduce dimensions of data
              EndFunction
              Function TrainAndOptimizeModels(data)
                models = [KNN(), SVM(),Random Forest(), DecisionTrees()]
                foreach model in models
                  train and validate model with data
                  optimize model based on validation results
                endfor
                return models
              EndFunction
              Function DeployModels(models)
                deploy to IoT infrastructure
                continuously monitor and update models
              EndFunction
              data = CollectData()
              processed_data = PreprocessData(data)
              models = TrainAndOptimizeModels(processed_data)
              DeployModels(models)
            End
```

Model deteksi serangan pada Pseudocode tersebut menggunakan model klasifikasi K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forest (RF), dan Decision Trees (DT). data dilatih dan divalidasi. Setiap model dioptimalkan berdasarkan hasil validasi untuk meningkatkan efektivitas deteksi, model diukur menggunakan matriks konfusi (confusion matrix) sehingga masing-masing diketahui nilai akurasi, presisi, recall dan F1-Score.

3. Hasil dan Pembahasan

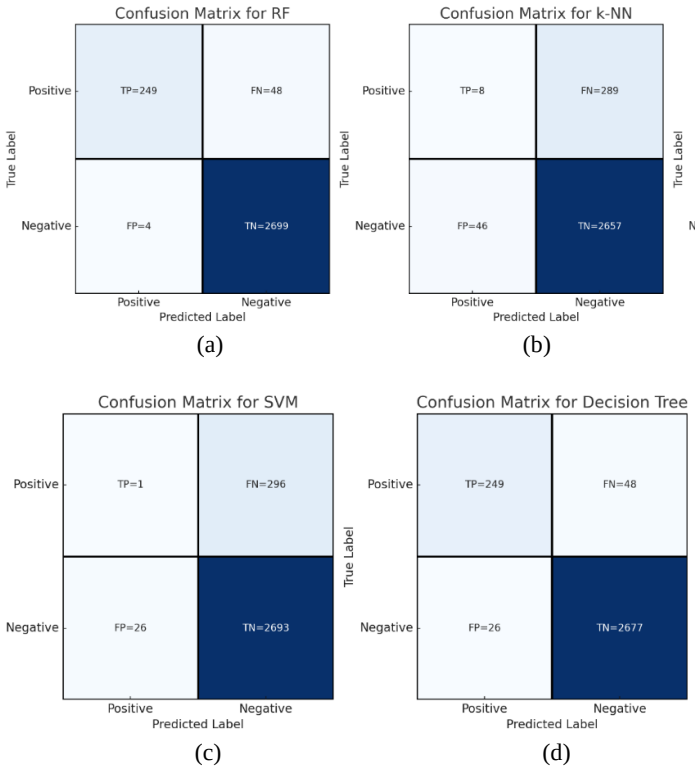
Model diuji menggunkan dataset trafik sebanyak 9.874 transaksi, aktifitas di jaringan berupa aktifitas transaksi normal dan aktifitas serangan DDoS serta Brute Force. Hasil pengujian untuk mendeteksi serangan pada Tabel 3.

Tabel 3. Confusion Matrik Model Klasifikasi.

Variable	RF	k-NN	SVM	DT
TP	249	8	1	249
FN	48	289	296	48
TN	2699	2657	2693	2677
FP	4	46	26	26

Model klasifikasi RF memiliki jumlah True Positive (TP) yang tinggi (249), yang menunjukkan bahwa model mampu dengan baik dalam mengidentifikasi serangan positif yang sebenarnya sebagai positif. Selain itu, jumlah False Negative (FN) dan False Positive (FP) yang rendah menunjukkan bahwa RF juga memiliki tingkat kesalahan yang rendah. Pada model k-NN memiliki jumlah TP yang relatif rendah (8) dibandingkan dengan model lainnya, dan jumlah FN yang tinggi (289). Ini menunjukkan bahwa k-NN mungkin memiliki masalah dalam mengidentifikasi kasus positif yang sebenarnya. Untuk model SVM memiliki jumlah TP yang rendah (1) dan jumlah FN yang tinggi (296). Meskipun SVM memiliki jumlah FP yang rendah, kemampuannya untuk mengidentifikasi serangan positif yang sebenarnya rendah. Sedangkan Decision Tree memiliki jumlah TP yang tinggi (249) dan jumlah FN yang rendah, menunjukkan

kemampuannya dalam mengidentifikasi serangan positif yang sebenarnya. Dalam bentuk confusion matrix ditunjukkan pada Gambar 1.



Gambar 1 (a). Confusion Matrik RF (b). Confusion Matrik KNN
(c). Confusion Matrik SVM (d). Confusion Matrik DT

Pada Tabel 4 analisis menunjukkan bahwa Random Forest (RF) memiliki Presisi yang sangat tinggi (98.4%), yang menunjukkan bahwa hampir semua prediksinya yang positif adalah benar. Recall RF juga tinggi (83.9%), menunjukkan kemampuannya dalam mengidentifikasi sebagian besar kasus positif yang ada. F1-Score RF mencapai 90.5%, yang menunjukkan kinerja keseluruhan yang sangat baik.

KNN memiliki Presisi yang sangat rendah (14.8%), yang mengindikasikan bahwa sebagian besar prediksi positifnya tidak akurat. Recall k-NN sangat rendah (2.7%), yang berarti model ini melewatkan sebagian besar kasus positif. F1-Score k-NN hanya mencapai 4.5%, menunjukkan kinerja yang kurang memadai secara keseluruhan.

Support Vector Machine (SVM) memiliki Presisi yang sangat rendah (3.7%) dan Recall yang hampir tidak ada (0.3%), menunjukkan bahwa model ini tidak efektif dalam mengidentifikasi atau mempertahankan kasus positif. F1-Score SVM hanya mencapai 0.6%, menunjukkan kinerja yang sangat buruk.

Sedangkan Decision Tree menunjukkan kinerja yang lebih baik dengan Presisi yang tinggi (90.5%) dan Recall yang tinggi (83.9%), menghasilkan F1-Score yang tinggi (87.1%). Secara keseluruhan, tabel evaluasi kinerja menunjukkan bahwa Random Forest dan Decision Tree memiliki kinerja yang sangat baik di semua metrik, sedangkan k-NN dan SVM menunjukkan kekurangan signifikan dalam semua aspek kinerja.

Tabel 4. Confusion Matrik Model Klasifikasi.

Variable	Prediksi	Presisi	Recall	F1-Score
RF	98.4%	98.4%	83.9%	90.5%
k-NN	89.1%	14.8%	2.7%	4.5%
SVM	89.7%	3.7%	0.3%	0.6%
Decision Tree	98.1%	90.5%	83.9%	87.1%

4. Kesimpulan

Berdasarkan hasil pengujian dan analisis matriks konfusi untuk keempat model klasifikasi, dapat disimpulkan bahwa Random Forest dan Decision Tree menampilkan kinerja yang luar biasa dengan presisi dan recall yang tinggi, mencapai F1-Score sebesar 90.5% dan 87.1% secara berturut-turut. Hal ini menunjukkan efektivitas kedua model ini dalam mengklasifikasikan dengan akurat dan efisien. Sedangkan KNN akurasi sangat rendah dengan F1-Score sebesar 4.5%, menunjukkan presisi dan kemampuan deteksi yang sangat rendah, yang mengindikasikan kurang cocoknya model ini untuk dataset ini. Untuk Support Vector Machine (SVM) juga menunjukkan kinerja yang kurang, hasil F1-Score sebesar 0.6%, menandakan bahwa model ini memerlukan optimisasi parameter yang signifikan untuk meningkatkan kinerjanya. Pemilihan model yang tepat berdasarkan karakteristik data menunjukkan model Random Forest dan Decision Tree menjadi pilihan yang sangat andal dalam studi ini.

Daftar Pustaka

[1] K. Rastogi and D. Lohani, "IoT-based Indoor Occupancy Estimation Using Edge Computing," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 1943–1952. doi: 10.1016/j.procs.2020.04.208.

[2] T. Takenaka, A. Ashima, and N. Nishino, "Strategies for evolving IoT-based Product-Service Systems from Emergent Synthesis Perspective," *Procedia CIRP*, vol. 112, no. March, pp. 1–5, 2022, doi: 10.1016/j.procir.2022.09.014.

[3] Veeramanikandan, S. Sankaranarayanan, J. J. P. C. Rodrigues, V. Sugumaran, and S. Kozlov, "Data Flow and Distributed Deep Neural Network based low latency IoT-Edge computation model for big data environment," *Eng Appl Artif Intell*, vol. 94, Sep. 2020, doi: 10.1016/j.engappai.2020.103785.

[4] Y. Yao, Z. Wang, and P. Zhou, "Privacy-preserving and energy efficient task offloading for collaborative mobile computing in IoT: An ADMM approach," *Comput Secur*, vol. 96, Sep. 2020, doi: 10.1016/j.cose.2020.101886.

[5] W. Huang, K. Ota, M. Dong, T. Wang, S. Zhang, and J. Zhang, "Result return aware offloading scheme in vehicular edge networks for IoT," *Comput Commun*, vol. 164, pp. 201–214, Dec. 2020, doi: 10.1016/j.comcom.2020.10.019.

[6] M. Babar, M. U. Tariq, and M. A. Jan, "Secure and resilient demand side management engine using machine learning for IoT-enabled smart grid," *Sustain Cities Soc*, vol. 62, Nov. 2020, doi: 10.1016/j.scs.2020.102370.

[7] K. Sha, T. A. Yang, W. Wei, and S. Davari, "A survey of edge computing-based designs for IoT security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 195–202, May 2020, doi: 10.1016/j.dcan.2019.08.006.

[8] C. M. Diaz, K. K. R. Choo, and A. Zunino, "Sharpening the edge: Towards improved edge computing environment for mobile and IoT applications," *Future Generation Computer Systems*, vol. 107. Elsevier B.V., pp. 1130–1133, Jun. 01, 2020. doi: 10.1016/j.future.2019.06.017.

[9] A. S. Nasution, A. Alvin, A. T. Siregar, and M. S. Sinaga, "KNN Algorithm for Identification of Tomato Disease Based on Image Segmentation Using Enhanced K-Means Clustering," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 4, no. 3, 2022, doi: 10.22219/kinetik.v7i3.1486.

[10] H. Syahputra and A. Wibowo, "Comparison of Support Vector Machine (SVM) and Random Forest Algorithm for Detection of Negative Content on Websites," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 1, pp. 165–173, 2023, doi: 10.26555/jiteki.v9i1.25861.

[11] D. A. Gustian, N. L. Rohmah, G. F. Shidik, A. Z. Fanani, R. A. Pramunendar, and Pujiono, "Classification of Troso Fabric Using SVM-RBF Multi-class Method with GLCM and PCA Feature Extraction," *Proceedings - 2019 International Seminar on Application for Technology of Information and Communication: Industry 4.0: Retrospect, Prospect, and Challenges, iSemantic 2019*, pp. 7–11, 2019, doi: 10.1109/ISEMANTIC.2019.8884329.

[12] Aditya Gumilar, Sri Suryani Prasetiyowati, and Yuliant Sibaroni, "Performance Analysis of Hybrid Machine Learning Methods on Imbalanced Data (Rainfall Classification)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 3, pp. 481–490, 2022, doi: 10.29207/resti.v6i3.4142.