Deteksi Botnet IoT Menggunakan Autoencoder dan Decision Tree

Susanto[1]*, M. Agus Syamsul Arifin[2], Harma Oktafia Lingga Wijaya[3]
Program Studi Informatika : Fakultas Ilmu Teknik[1]
Program Studi Rekayasa Sistem Komputer : Fakultas Ilmu Teknik[2]
Program Studi Sistem Informasi : Fakultas Ilmu Teknik[3]
Universitas Bina Insan
Lubuklinggau, Indonesia
susanto@univbinainsan.ac.id[1], mas.arifin@univbinainsan.ac.id [2], harmaoktafialingga@univbinainsan.ac.id [3]

***Abstract*— The use of IoT devices has grown rapidly, leading to an increase in cyber attacks that pose greater security and privacy threats than ever before. One such threat is botnet attacks on IoT devices. An IoT botnet is a group of Internet-connected IoT devices infected with malware and remotely controlled by an attacker. Machine learning techniques can be employed to detect botnet attacks. The use of machine learning-based detection methods has been shown to be effective in identifying cyber attacks. The performance of the detection system in machine learning can be improved by utilizing data reduction methods. The data reduction process in classification is used to overcome the problem of scalability and computation resources in the IoT. This paper proposes a detection system using the Autoencoder reduction method and the Decision tree classification method. The test results demonstrate that the Deep Autoencoder algorithm can reduce data and memory usage from 1.62 GB to 75.9 MB, while also improving the performance of decision tree classification, resulting in a high level of accuracy up to 100%. The Autoencoder approach in conjunction with the Decision Tree exhibits superior capabilities compared to previous studies.**

***Keywords*— Botnet IoT, Dimensionality reduction, Autoencoder, Decision Tree**

***Abstrak*— Seiring dengan pesatnya pertumbuhan perangkat IoT, serangan dunia maya juga semakin meningkat dan menimbulkan ancaman keamanan dan privasi yang lebih serius daripada sebelumnya. Salah satunya adalah serangan botnet pada perangkat IoT. Botnet IoT adalah kumpulan perangkat IoT yang terhubung ke Internet yang telah terinfeksi malware dan dikelola dari jarak jauh oleh penyerang. Salah satu teknik yang dapat digunakan dalam mendeteksi serangan botnet adalah teknik pembelajaran mesin. Penerapan metode deteksi berbasis pembelajaran mesin telah terbukti efisien dalam mendeteksi serangan cyber. Performa kinerja sistem deteksi pada pembelajaran mesin dapat ditingkatkan dengan menggunakan metode reduksi data. Proses reduksi data pada klasifikasi digunakan untuk mengatasi masalah skalabilitas dan sumber daya komputasi di IoT. Pada makalah ini, kami mengusulkan Sistem deteksi menggunakan metode reduksi Autoencoder dan metode klasifikasi Decision tree. Hasil pengujian menunjukkan bahwa algortima Autoencoder dalam mereduksi data sehingga dapat mengurangi penggunaan memori data dari 1.62 GB menjadi 75.9 MB, selain itu juga meningkatkan kinerja klasifikasi decision tree. Hal ini terlihat dari tingkat akurasi yang tinggi hingga mencapai 100%. Pendekatan Autoencoder dengan Decision Tree memiliki kemampuan yang lebih unggul dibandingkan penelitian sebelumnya.**

***Kata Kunci*— Botnet IoT, Pengurangan Dimensi, Autoencoder, Decision Tree**

I. PENDAHULUAN

Pesatnya perkembangan dan penerapan teknologi berbasis *Internet of Things* (IoT) telah memungkinkan berbagai kemungkinan kemajuan teknologi untuk berbagai aspek kehidupan [1]. Peningkatan penggunaan perangkat perangkat IoT, selaras dengan serangan *cyber* juga yang semakin meningkat, sehingga dapat mengganggu kinerja dari perangkat IoT bahkan menjadi ancaman serius pada keamanan dan privasi [2]. Serangan *cyber* pada perangkat IoT yang paling serius salah satunya adalah serangan botnet [3], [4]. Serangan botnet IoT, yang bertujuan untuk melakukan kejahatan dunia maya yang nyata, efisien, dan menguntungkan, sehingga menjadi salah satu ancaman IoT yang paling serius [5].

Salah satu teknik yang dapat digunakan dalam mendeteksi serangan botnet adalah teknik pembelajaran mesin [6]. Penerapan metode deteksi berbasis pembelajaran mesin telah terbukti efisien dalam mendeteksi serangan *cyber* [7], meskipun bukan tanpa keterbatasan [8]. Oleh karena itu performa kinerja sistem deteksi pada pembelajaran mesin dapat ditingkatkan dengan menggunakan metode reduksi data [9]. Tujuan reduksi data bukan untuk menghilangkan informasi yang dapat diekstraksi tetapi untuk meningkatkan efektivitas pembelajaran mesin ketika kumpulan data yang tersedia besar [10]. Data yang direduksi menjadikan penggunaan sumber daya penyimpanan yang lebih sedikit sehingga manajemen penyimpanan lebih efisien [11].

Hal ini terlihat telah banyak penelitian diantaranya Bahsi et al. [12] melakukan reduksi data dengan menggunakan metode fisher score yang kemudian diklasifikasikan dengan menggunakan k-NN dan decision tree. Kemudian Nomm et al. [13] menggunakan metode entropi, variance, dan Hopkins dalam mereduksi data yang kemudian diklasifikasikan dengan one class SVM and isolation forest. Selanjutnya Susanto et al. [14] melakukan proses reduksi data dengan menggunakan

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i3.1693, Copyright ©2023
Submitted : May 3, 2023, Revised : May 8, 2023, Accepted : June 15. 2023, Published : November 03, 2023

metode fastICA, setelah itu diklasifikasikan dengan metode k-NN, Decision Tree, Random Forest, and Gradient Boosting. Selain itu Alqahtani et al. [15] mereduksi data mengunakan metode fisher score, yang selanjutnya diklasifikasikan dengan metode XGBoost.

Pekerjaan yang dilakukan ini berkontribusi terhadap pengembangan model untuk sistem deteksi serangan botnet pada jaringan IoT. Sistem deteksi yang diusulkan dengan melakukan proses reduksi data menggunakan Autoencoder, kemudian dilanjutkan dengan proses klasifikasi menggunakan Decision tree. Metode reduksi Autoencoder dipilih karena memiliki perbedaan dengan metode reduksi dimensi lainnya. Hal ini terlihat dalam beberapa kasus, pembuat enkode otomatis tidak hanya mengurangi dimensi, tetapi juga dapat mendeteksi struktur berulang [16]. Disisi lain metode klasifikasi decision tree memiliki kelebihan antara lain, Pertama, model pohon keputusan sampai taraf tertentu mudah diikuti dan diterapkan. Kedua, model pembelajaran berbasis pohon keputusan kurang memperhatikan proses penyiapan sampel karena pada tingkat tertentu tidak berguna dan memakan waktu. Ketiga, proses verifikasi efektivitas dan ketahanan model pohon keputusan dapat dengan mudah diukur [17].

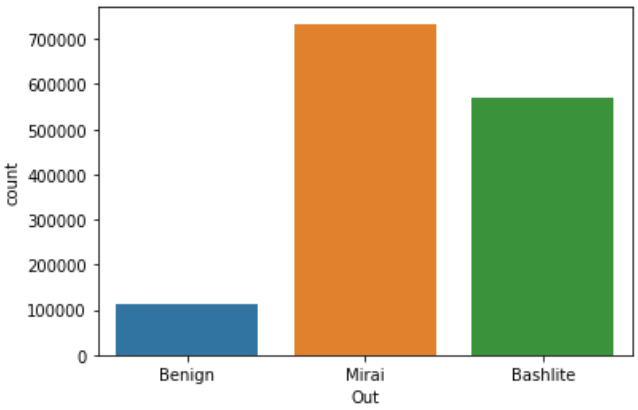
II. METODE PENELITIAN

A. *Dataset*

Penelitian ini menggunakan menggunakan jenis *dataset* comma-separated value (CSV) yaitu dataset N-BaIoT [18]. Dataset N-BaIoT diekstraksi menggunakan metode incremental statistic [19], yang menghasilkan total 115 fitur. Dataset N-BaIoT terdiri dari sembilan perangkat IoT yang mencakup webcam, monitor bayi, empat kamera keamanan, termostat, dan dua bel pintu. *Dataset* ini memiliki tiga jenis data lalu lintas (yaitu, satu jenis lalu lintas Benign dan dua jenis serangan lalu lintas yaitu Mirai dan Bashlite), yang berjumlah total 7.062.606 data. Dalam pengujian ini menggunakan sekitar 20% dari kumpulan data N-BaIoT, dengan total 1.415.912 data. Proses pengambilan data dilakukan dengan cara mengambil data sebanyak 20% dari setiap file dataset dimulai dari baris pertama sehingga dapat mewakili seluruh datanya. Selain itu pada dataset pelabelan data terletak pada penamaan file maka diperlukan penambahan label baru pada fiturnya untuk proses klasifikasi sehingga total fitur menjadi 116 fitur. Sebaran data tersaji pada tabel I dan gambar 1.

TABEL I. SEBARAN DATASET N-BAIOT

Label Baru	Label File	Jumlah Data
Benign	Benign	111179
Bashlite	Combo	103030
	Junk	52158
	Scan	51022
	TCP	171969
	UDP	192873
Mirai	Ack	128764
	Scan	107596
	Syn	146660
	UDP	246001
	UDPplain	104660
Total		1415912



Gambar 1. Sebaran label pada 20% dataset N-BaIoT

B. *Autoencoder*

Proses reduksi dimensi data berguna untuk mengurangi jumlah fitur dan memperkecil ukuran dataset sehingga dapat meningkatkan performa deteksi. Penelitian ini menggunakan algoritma Autoencoder sebagai metode reduksi data. Rumus Autoencoder dinyatakan dalam (1) dan (2) [20].

$$Y = f(X) = s(WX + bx) \quad (1)$$

$$X' = g(Y) = s(W'Y + by) \quad (2)$$

Dimana,

$Y = f(X)$ = Fungsi Encoded
W = Fungsi Weighted encoded
$X' = g(Y)$ = Fungsi Decode
bx = Bias encoded
S = Fungsi Activation
W' = Weighted decoded
by = Bias decoded

Arsitektur dan parameter Autoencoder disajikan pada tabel II. Nilai pada parameter tersebut diisi dengan cara manual yang digunakan selama pengujian trial error sampai menemukan hasil yang maksimal.

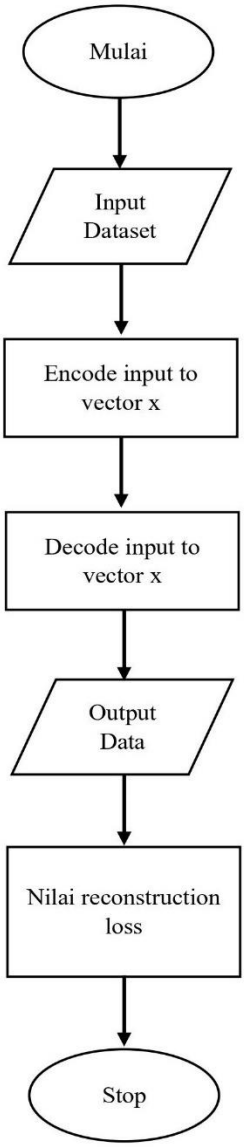
TABEL II. PARAMETER AUTOENCODER

Parameter	Nilai
Node Input layer	74
Node Ouput layer	74
Node on Hidden layer 1	50
Node on Hidden layer 2	30
Node on Hidden layer 3	20

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i3.1693, Copyright ©2023
Submitted : May 3, 2023, Revised : May 8, 2023, Accepted : June 15. 2023, Published : November 03, 2023

Node on Hidden layer 4	10
Node on Hidden layer 5	5
Activation function of hidden layer	Relu
Activation function of Ouput layer	Sigmoid and Softmax
Learning rate	0.00001
Fungsi Loss	Categorical crossentropy
Fungsi optimasi Optimization	Adam

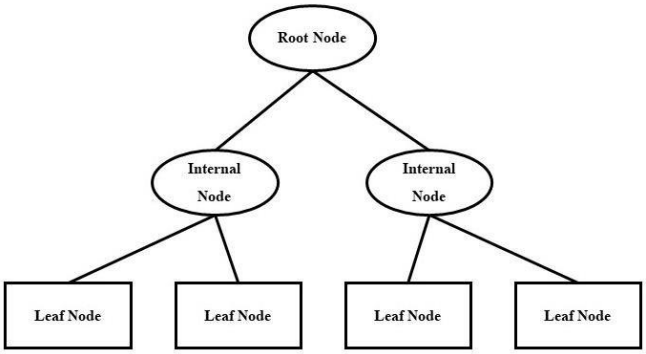
Proses pengurangan dimensi data menggunakan Autoencoder disajikan pada gambar 2.



Gambar 2. Flowchart pengurangan dimensi menggunakan Autoencoder

C. Decision Tree

Decision Tree adalah struktur mirip pohon yang memiliki daun, yang merepresentasikan klasifikasi dan cabang, yang pada gilirannya merepresentasikan konjungsi fitur yang mengarah ke klasifikasi tersebut. Keuntungan dari klasifikasi Decision Tree adalah ekspresi pengetahuan intuitif, akurasi klasifikasi tinggi, dan implementasi sederhana. Kerugian utamanya adalah untuk data, termasuk variabel kategori dengan beberapa level yang berbeda, nilai perolehan informasi cenderung mendukung fitur dengan lebih banyak level [21]. Struktur decision tree [22] disajikan pada gambar 3.



Gambar 3. Struktur decision tree

D. Evaluasi Performa

Proses evaluasi performa klasifikasi decision tree menggunakan *confusion matrix* seperti yang disajikan pada table III [23].

TABEL III. CONFUSION MATRIX

	Predicted Class		
		Possitive (P)	***Negative (N)***
Actual Class	*True* (T)	TP	FP
	False (F)	FN	TN

Dimana, TP= *True Positive*, TN= *True Negative*, FP= *False Positive*, and FN= *False Negative*. Sehingga didapat kalkulasi *classification matrix* untuk evaluasi performa peningkatan klasifikasi (3) – (7),

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (5)$$

$$Specificity = \frac{TP}{TP+FN} \quad (6)$$

$$False\text{-}positive\ rate = \frac{FP}{TN+FP} \quad (7)$$

E. Validasi

Proses validasi dilakukan dengan menggunakan berbagai perbandingan data *testing* dan data *training*, seperti yang disajikan pada tabel IV.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i3.1693, Copyright ©2023
Submitted : May 3, 2023, Revised : May 8, 2023, Accepted : June 15. 2023, Published : November 03, 2023

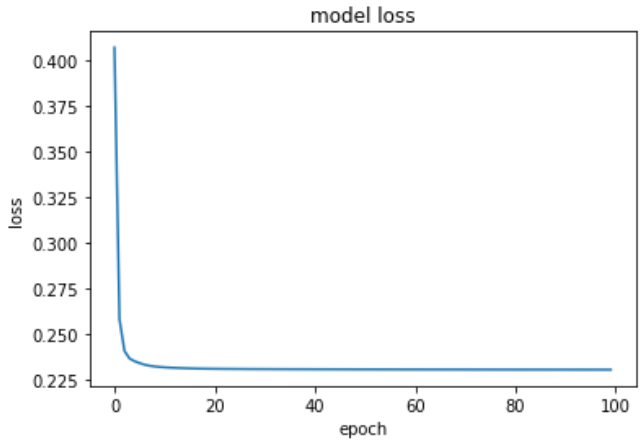
TABEL IV. VALIDASI PEMBAGIAN DATA

Validasi	Data
Validasi pertama	*Training* Data 50% and *Testing* Data 50 %
Validasi kedua	*Training* Data 60% and *Testing* Data 40 %
Validasi ketiga	*Training* Data 70% and *Testing* Data 30 %
Validasi keempat	*Training* Data 80% and *Testing* Data 20 %
Validasi kelima	*Training* Data 90% and *Testing* Data 10 %

III. HASIL PENGUJIAN DAN KOMPARASI

A. *Reduksi Data menggunakan Autoencoder*

Hasil pengujian dalam mereduksi data menggunakan arsitektur *autoencoder* dengan dengan konfigurasi 3 layer, optimasi Adam, nilai *epoch* sebesar 100, ukuran *batch size* sebesar 64, nilai *learning rate* sebesar 0.00001 dan menggunakan fungsi *loss categorial_crossentropy*. Berikut plot grafik untuk model *loss* dengan menggunakan arsitektur *autoencoder* 3 layer dapat dilihat pada gambar 4.



Gambar 4. Grafik loss data multiclass terhadap autoencoder 3 layer

Pada gambar 4 menunjukkan grafik *loss* dari proses reduksi dimensi dataset menggunakan *autoencoder* 3 layer. Nilai loss yang didapatkan pada arsitektur *autoencoder* 3 layer ini sebesar 23.09. Hasil reduksi data yang pada awalnya memiliki ukuran 115 kolom menjadi 5 kolom yang dapat dilihat pada tabel V. *Dataset* awal memiliki ukuran file sebesar 1.62 GB setelah melewati proses reduksi data menjadi sebesar 75.9 MB.

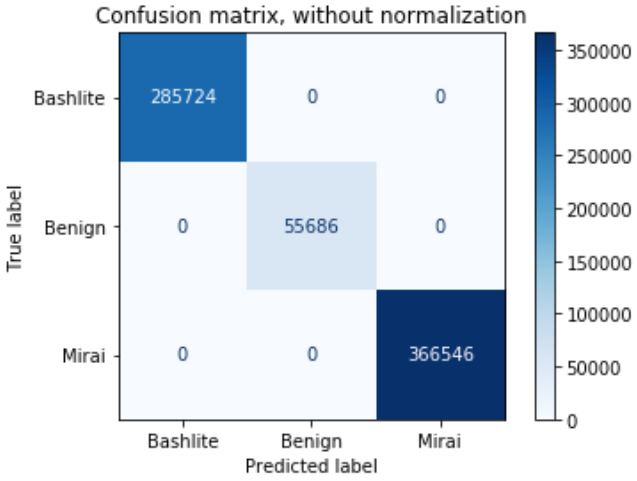
TABEL V. REDUKSI DATA MENGGUNAKAN 3 LAYER AUTOENCODER

Feature_0	Feature_1	Feature_2	Feature_3	Feature_4
0.000000	64.312424	28.513479	27.695612	50.174500
11.541241	5.329647	17.573317	4.191519	19.135214
11.444980	6.075598	36.264290	11.650904	22.337646
0.000000	15.838280	2.586906	39.339481	51.064587
9.027152	25.660934	16.014372	8.803638	8.126516

B. *Hasil Klasifikasi*

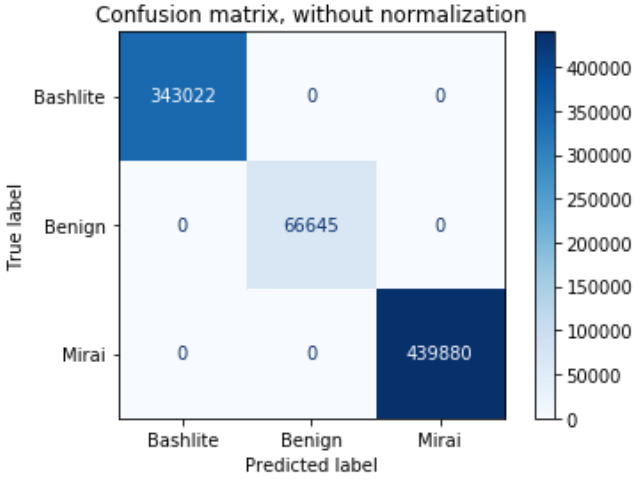
Algoritma klasifikasi decision tree diuji pada dataset kemudian dievaluasi dan dibandingkan. Tujuh metrik kinerja, yaitu: *confusion matrix*, akurasi, *precision*, *sensitifity*, *specificity*, dan *False Positive Rate* (FPR).

Gambar 5 menunjukkan hasil *confusion matrix* dari deteksi botnet dengan perbandingan 50% data *training* dan 50% data *testing*. Hasil pengujian menunjukkan bahwa klasifikasi Decision tree mampu mengklasifikasikan 709956 record data atau 100% dengan benar. Gambar 6 menunjukkan hasil *confusion matrix* dari deteksi botnet dengan perbandingan 60% data *training* dan 40% data *testing*. Hasil pengujian menunjukkan bahwa klasifikasi Decision tree mampu mengklasifikasikan 849547 record data atau 100% dengan benar. Gambar 7 menunjukkan hasil *confusion matrix* dari deteksi botnet dengan perbandingan 70% data *training* dan 30% data *testing*. Hasil pengujian menunjukkan bahwa klasifikasi Decision tree mampu mengklasifikasikan 991138 record data atau 100% dengan benar. Gambar 8 menunjukkan hasil *confusion matrix* dari deteksi botnet dengan perbandingan 80% data *training* dan 20% data *testing*. Hasil pengujian menunjukkan bahwa klasifikasi Decision tree mampu mengklasifikasikan 1132729 record data atau 100% dengan benar. Gambar 9 menunjukkan hasil *confusion matrix* dari deteksi botnet dengan perbandingan 90% data trainig dan 10% data testing. Hasil pengujian menunjukkan bahwa klasifikasi Decision tree mampu mengklasifikasikan 1274320 record data atau 100% dengan benar.

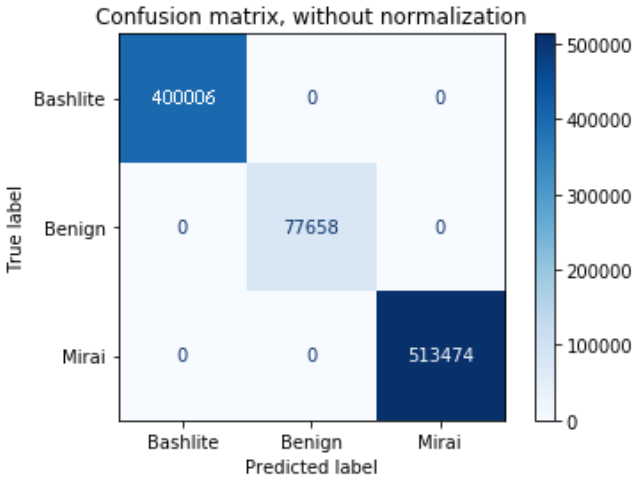


Gambar 5. Confusion matrix decision tree dengan perbandingan 50% data *training* dan 50% data *testing*

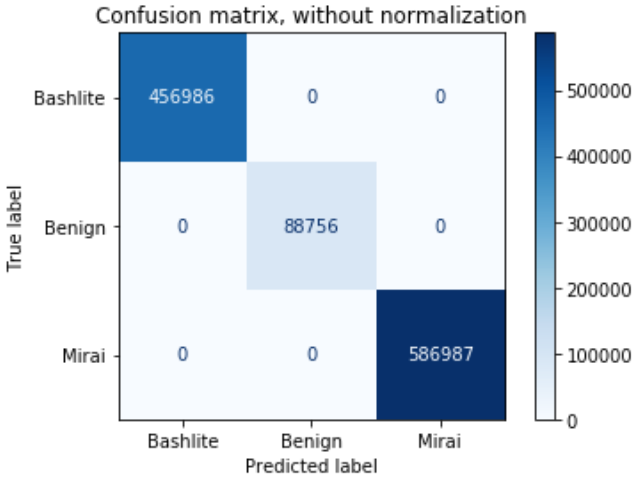
p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i3.1693, Copyright ©2023
Submitted : May 3, 2023, Revised : May 8, 2023, Accepted : June 15. 2023, Published : November 03, 2023



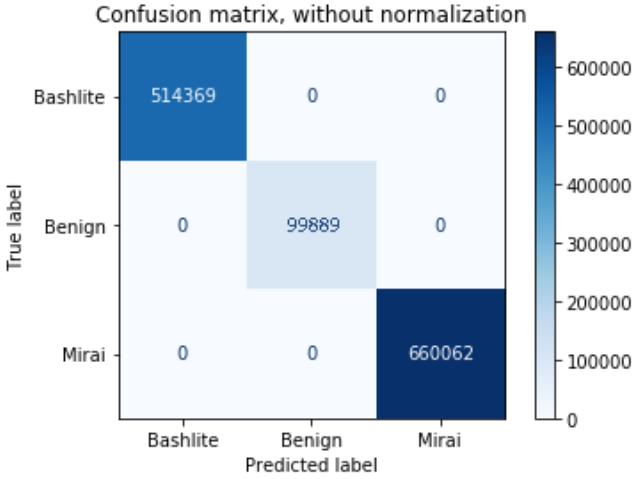
Gambar 6. Confusion matrix decision tree dengan perbandingan 60% data *training* dan 40% data *testing*



Gambar 7. Confusion matrix decision tree dengan perbandingan 70% data training dan 30% data testing



Gambar 8. Confusion matrix decision tree dengan perbandingan 80% data training dan 20% data testing



Gambar 9. Confusion matrix decision tree dengan perbandingan 90% data *training* dan 10% data *testing*

Berdasarkan hasil pengujian, didapatkan performa kinerja deteksi botnet IoT yang direduksi dengan menggunakan autoencoder, kemudian diklasifikasikan menggunakan decision tree. Hasil validasi yang disajikan pada tabel VI dan tabel VII, menunjukkan bahwa tingkat akurasi, *specificity*, *sensitivity*, *precision*, dan *false positive rate* tidak terpengaruh oleh perubahan rasio data training dan data testing. Tingkat akurasi mencapai 100% baik pada data training maupun data testing. Selanjutnya nilai *specificity*, *sensitivity*, dan *precision* tetap dinilai 1.0 baik pada data training maupun data testing. Kemudian nilai false positive rate stabil di angka 0 baik pada data *training* maupun data *testing*.

TABEL VI. HASIL PENGUJIAN TRAINING DATA KLASIFIKASI DECISION TREE

Rasio Data	Training data				
	Akurasi	*Specificity*	*Sensitivity*	*Precision*	*False Positive rate*
50:50	100	1.0	1.0	1.0	0
60:40	100	1.0	1.0	1.0	0
70:30	100	1.0	1.0	1.0	0
80:20	100	1.0	1.0	1.0	0
90:10	100	1.0	1.0	1.0	0

TABEL VII. HASIL PENGUJIAN TESTING DATA KLASIFIKASI DECISION TREE

Rasio Data	Testing data				
	Akurasi	*Specificity*	*Sensitivity*	*Precision*	*False Positive rate*
50:50	100	1.0	1.0	1.0	0
60:40	100	1.0	1.0	1.0	0
70:30	100	1.0	1.0	1.0	0
80:20	100	1.0	1.0	1.0	0
90:10	100	1.0	1.0	1.0	0

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i3.1693, Copyright ©2023
Submitted : May 3, 2023, Revised : May 8, 2023, Accepted : June 15. 2023, Published : November 03, 2023

C. Komparasi dengan penelitian lainnya

Untuk mengetahui tingkat ke unggulan metode yang diusulkan, penulis mengkomparasikan dengan penelitian sebelumnya yang menggunakan *dataset* sama yaitu N-BaIoT. Hasil komparasi menunjukkan bahwa metode yang diusulkan lebih akurat dalam mendeteksi botnet IoT jika dibandingkan metode lainnya, seperti yang ditunjukkan pada tabel VIII.

TABEL VIII. KOMPARASI DENGAN PENELITIAN LAINNYA

Referensi & Tahun	Metode	Dataset	Jumlah Data	Jumlah Dimensi	Akurasi
[12] (2018)	Fisher score + k-NN	N-BaIoT	1507815 (21.35%)	3	97.24
[15] (2020)	Fisher score + XGBoost with Genetika Algorithm	N-BaIoT	1667796 (23.61%)	3	99.96
This Work	**Autoencoder + Decision Tree**	**N-BaIoT**	**1415912 (20%)**	**3**	**100%**

IV. KESIMPULAN

Kami telah mengimplementasikan algortima Autoencoder dalam mereduksi data sehingga dapat mengurangi memori data dari 1.62 GB menjadi 75.9 MB, selain itu juga meningkatkan kinerja klasifikasi decision tree. Model yang diusulkan memiliki kinerja yang sangat baik pada deteksi botnet IoT, hal ini terlihat dari tingkat akurasi yang tinggi hingga mencapai 100%.. Pada penelitian selanjutnya, kami akan mencoba untuk meningkatkan kinerja deteksi dan pencegahan serangan *cyber* dari jaringan IoT yang lebih kompleks..

REFERENCES

[1] S. Nižetić, P. Šolić, D. López-de-Ipiña González-de-Artaza, and L. Patrono, "Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future," *J. Clean. Prod.*, vol. 274, 2020, doi: 10.1016/j.jclepro.2020.122877.

[2] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1606–1616, 2019, doi: 10.1109/JIOT.2018.2847733.

[3] Susanto, M. A. Syamsul Arifin, D. Stiawan, M. Y. Idris, and R. Budiarto, "The trend malware source of IoT network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 450–459, 2021, doi: 10.11591/ijeecs.v22.i1.pp450-459.

[4] M. Alshamkhany, W. Alshamkhany, M. Mansour, M. Khan, S. Dhou, and F. Aloul, "Botnet Attack Detection using Machine Learning," in *Proc. 14th International Conference on Innovations in Information Technology, IIT*, 2020, no. November, pp. 203–208.

[5] Z. Shao, S. Yuan, and Y. Wang, "Adaptive online learning for IoT botnet detection," *Inf. Sci. (Ny).*, vol. 574, pp. 84–95, 2021, doi: 10.1016/j.ins.2021.05.076.

[6] S. Srinivasan and D. P, "Enhancing the security in cyber-world by detecting the botnets using ensemble classification based machine learning," *Meas. Sensors*, vol. 25, no. December 2022, p. 100624, 2023, doi: 10.1016/j.measen.2022.100624.

[7] C. Maudoux, S. Boumerdassi, A. Barcello, and E. Renault, "Combined Forest: A New Supervised Approach for a Machine-Learning-based Botnets Detection," *2021 IEEE Glob. Commun. Conf. GLOBECOM 2021 - Proc.*, pp. 1–6, 2021, doi: 10.1109/GLOBECOM46510.2021.9685261.

[8] S. Miller and C. Busby-Earle, "The role of machine learning in botnet detection," *2016 11th Int. Conf. Internet Technol. Secur. Trans. ICITST 2016*, no. December, pp. 359–364, 2017.

[9] Susanto, D. Stiawan, M. A. S. Arifin, J. Rejito, M. Y. Idris, and R. Budiarto, "A Dimensionality Reduction Approach for Machine Learning Based IoT Botnet Detection," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, vol. 2021–Octob, no. October, pp. 26–30, 2021, doi: 10.23919/EECSI53397.2021.9624299.

[10] I. Czarnowski and P. Jędrzejowicz, "An approach to data reduction for learning from big datasets: Integrating stacking, rotation, and agent population learning techniques," *Complexity*, vol. 2018, 2018, doi: 10.1155/2018/7404627.

[11] M. H. ur Rehman, C. S. Liew, A. Abbas, P. P. Jayaraman, T. Y. Wah, and S. U. Khan, "Big Data Reduction Methods: A Survey," *Data Sci. Eng.*, vol. 1, no. 4, pp. 265–284, 2016, doi: 10.1007/s41019-016-0022-0.

[12] H. Bahsi, S. Nomm, and F. B. La Torre, "Dimensionality Reduction for Machine Learning Based IoT Botnet Detection," in *Proc. 2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV*, 2018, pp. 1857–1862.

[13] S. Nomm and H. Bahsi, "Unsupervised Anomaly Based Botnet Detection in IoT Networks," in *Proc.- 17th IEEE International Conference on Machine Learning and Applications, ICMLA*, 2019, pp. 1048–1053.

[14] Susanto *et al.*, "Dimensional Reduction With Fast ICA for IoT Botnet Detection," *J. Appl. Secur. Res.*, vol. 0, no. 0, pp. 1–24, 2022, doi: 10.1080/19361610.2022.2079906.

[15] M. Alqahtani, H. Mathkour, and M. M. Ben Ismail, "IoT botnet attack detection based on optimized extreme gradient boosting and feature selection," *Sensors (Switzerland)*, vol. 20, no. 21, pp. 1–21, 2020, doi: 10.3390/s20216336.

[16] Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, pp. 232–242, 2016, doi: 10.1016/j.neucom.2015.08.104.

[17] Y. Liu and S. Yang, "Application of Decision Tree-Based Classification Algorithm on Content Marketing," *J. Math.*, vol. 2022, 2022, doi: 10.1155/2022/6469054.

[18] Y. Meidan *et al.*, "N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Sep. 2018, doi: 10.1109/MPRV.2018.03367731.

[19] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," *arXiv*, no. February, pp. 18–21, 2018.

[20] Y. N. Kunang, S. Nurmaini, D. Stiawan, A. Zarkasi, and F. Jasmir, "Automatic Features Extraction Using Autoencoder in Intrusion Detection System," in *Proceedings of 2018 International Conference on Electrical Engineering and Computer Science, ICECOS 2018*, 2019, vol. 17, pp. 219–224, doi: 10.1109/ICECOS.2018.8605181.

[21] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.

[22] A. Banjongkan, W. Pongsena, N. Kerdprasop, and K. Kerdprasop, "A study of job failure prediction at job submit-state and job start-state in high-performance computing system: Using decision tree algorithms," *J. Adv. Inf. Technol.*, vol. 12, no. 2, pp. 84–92, 2021, doi: 10.12720/jait.12.2.84-92.

[23] A. Tharwat, "Classification assessment methods," *Appl. Comput. Informatics*, vol. 17, no. 1, pp. 168–192, 2021, doi: 10.1016/j.aci.2018.08.003.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v12i3.1693, Copyright ©2023
Submitted : May 3, 2023, Revised : May 8, 2023, Accepted : June 15. 2023, Published : November 03, 2023