Analisis Perbandingan Metode Seleksi Fitur pada Model Klasifikasi Decission Tree untuk Deteksi Serangan di Jaringan Komputer

Eva Hariyanti[1], Dandy Pramana Hostiadi[2], Anggreni[3], Yohanes Priyo Atmojo[4], I Made Darma Susila[5], Irene Tangkawarow[6]
[1]Universitas Airlangga, [2,4,5]Institut Teknologi dan Bisnis STIKOM Bali Indonesia, [3]Universitas Muhammadiyah Mataram, [6]Universitas Negeri Manado
e-mail: [1]eva.hariyanti@fst.unair.ac.id, [2]dandy@stikom-bali.c.id, [3]muchalianggreni@gmail.com, [4]yohanes@stikom-bali.ac.id, [5]darma_s@stikom-bali.ac.id, [6]irene.tangkawarow@unima.ac.id

Diajukan: 22 April 2024; Direvisi: 28 Mei 2024; Diterima: 29 Mei 2024

Abstrak

Perkembangan informasi dan teknologi memerlukan teknik pengamanan yang tepat. Potensi terjadinya kebocoran data dan informasi di era digital sangat tinggi apabila tidak ditangani dengan serius. Beberapa serangan berbahaya yang terjadi adalah spam, Denial of Service Attack, ARP Poisoning, SQL Injection, U2L, R2L dan Probing. Penelitian sebelumnya telah mengenalkan pendekatan deteksi serangan berbahaya seperti menggunakan klasifikasi, klusterisasi dan analisis statistik. Namun analisis penggunaan fitur terbaik perlu dilakukan untuk mendapatkan hasil model klasifikasi yang optimal. Pada penelitian ini, menganalisis dan mencari metode seleksi fitur terbaik yang dapat diimplementasikan pada model klasifikasi berbasis machine learning untuk mendeteksi serangan di jaringan. Dataset yang digunakan adalah UNSW-NB15, dan dilakukan beberapa proses seperti data transformasi, Data normalisasi, seleksi Fitur dan Klasifikasi. Perbandingan teknik seleksi fitur yang digunakan antara lain ANOVA, UNIVARIATE dan ChiSquare. Tujuan penelitian ini adalah untuk meningkatkan akurasi, precision dan recall pada model klasifikasi Decision Tree. Hasil penelitian pengujian menunjukkan bahwa metode seleksi fitur terbaik dalam model klasifikasi adalah metode ANOVA dengan hasil nilai Area Under Curve sebesar 0.989, nilai F1-score adalah 0.999, akurasi deteksi adalah 0.999, nilai precission adalah 0.999 dan recall adalah 0.999. Hasil penelitian ini dapat digunakan untuk menyempurnakan model Intrusi Detection System berbasis machine learning.

Kata kunci: *Decission tree, Intrusion detection system, Keamanan jaringan, Infrastruktur jaringan.*

Abstract

The development of information and technology requires appropriate security techniques. The potential for data and information leaks in the digital era is very high and needs to be mitigated. Some dangerous attacks are spam, Denial of Service Attack, ARP Poisoning, SQL Injection, U2L, R2L and Probing. Previous studies introduced malicious attack detection using classification, clustering, and statistical analysis. However, an analysis of the best features must be carried out to increase classification model results. This paper analyzed and looked for the best feature selection method to optimize the machine learning-based classification model to detect attacks on the network. The model used the UNSW-NB15 dataset and was divided into several processes: data transformation, data normalization, feature selection and classification. The feature selection techniques used in the comparison are ANOVA, UNIVARIATE, and Chi-Square. This research aims to improve the accuracy, precision, and recall of the Decision Tree classification model. The testing research results show that the best feature selection method in the classification model is the ANOVA method with an Area Under Curve value of 0.989, an F1-score value of 0.999, a detection accuracy of 0.999, a precision value of 0.999, a recall of 0.999. The results of this research can be used to improve machine learning-based Intrusion Detection System models.

Keywords*: Decission tree, Intrusion detecsion system, Network security, Network infrastructure.*

1. Pendahuluan

Perkembangan teknologi di era digital memerlukan sebuah pengawasan terhadap adanya potensi keamanan siber [1–3]. Penanganan yang tidak serius, dapat menyebabkan adanya permasalahan yang serius, misalnya kerugian finansial pada sebuah institusi. Serangan berbahaya dalam jaringan komputer dapat terjadi secara masif tanpa memperhatikan jenis instansi. Serangan dapat terjadi di instasi pendidikan, pemerintahan, industri swasta dan bahkan pada instansi penegak hukum [4, 5]. Serangan yang terjadi kerap menggunakan perangkat lunak ilegal yang sering dikenal dengan Malicious Software (Malware)[6–10].

Serangan berbahaya di jaringan komputer yang melibatkan penggunaan malware dapat menyerang secara personal atau secara distribusi. Aktivitas berbahaya dapat berupa serangan spam, probing, DoS, DDoS, U2L, R2L, ARP Poisoning, SQL Injection dan lain-lain [11–13]. Tujuan dari penyerangan umumnya adalah untuk mendapatkan akses secara ilegal, pencurian akun atau identitas, pengambilan data penting hingga perusakan terhadap sistem[14–17]. Sehingga penanganan terhadap serangan yang terjadi memerlukan fokus dan teknik yang tepat.

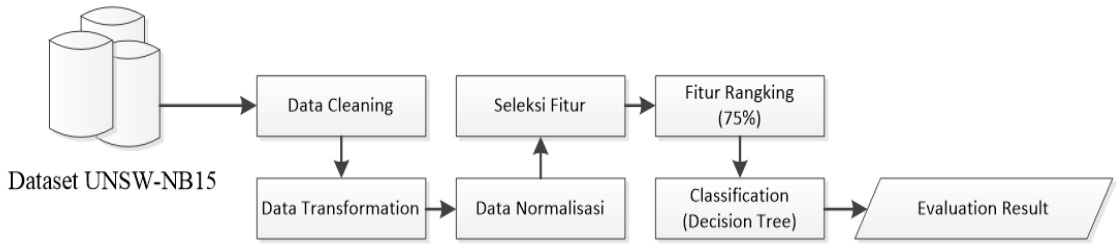
Beberapa penelitian deteksi serangan dalam jaringan, dikenalkan dalam penelitian sebelumnya, seperti dengan pendekatan klasifikasi, klusterisasi dan pendekatan statistik. Pendekatan klasifikasi yang umum digunakan adalah denga menggunakan metode k-NN, Naive Bayes, Decission Tree, Support Vector Machine dan Random Forest [5, 18–20]. Sedangkan metode klasifikasi yang sering digunakan adalah K-means dan DBSCAN. Pendekatan stastikal umumnya digunakan dalam mencari korelasi atau membangun basis pengetahuan dengan pendekatan kemiripan, seperti metode cosinus dan pearson correlation[21]. Namun tidak semua metode dapat mendeteksi secara akurat.

Salah satu teknik optimasi yang sering digunakan pada metode klasifikasi untuk deteksi serangan di jaringan komputer adalah seleksi fitur. Tidak semua metode seleksi fitur dapat bekerja optimal pada metode klasifikasi, sehingga perlu dilakukan analisis perbandingan di antara metode seleksi fitur seperti ANOVA, UNIVARIATE dan Chi-Square [11, 19, 20, 22, 23]. Analisis perbandingan seleksi fitur pada model klasifikasi perlu dilakukan untuk mengetahui seberapa besar pengaruh dan mendapatkan mana metode seleksi terbaik yang dapat meningkatkan akurasi deteksi serangan.

Penelitian ini mengusulkan metode baru dalam deteksi serangan di jaringan komputer dengan mengoptimasi model klasifikasi decision tree menggunakan teknik seleksi fitur. Metode seleksi fitur yang digunakan adalah metode terbaik dari yang dibandingkan, yaitu ANOVA, Univariate dan Chis-square. Tujuannya adalah untuk meningkatkan akurasi, precision dan recall deteksi pada model decision tree melalui pemilihan metode seleksi fitur. Manfaat dari analisis perbandingan adalah menyempurnakan model Intrusion detection system berbasis klasifikasi machine learning sehingga menghasilkan model deteksi yang optimal.

2. Metode Penelitian

Penelitian ini melakukan analisis perbandingan teknik seleksi fitur pada model deteksi serangan yang menggunakan pendekatan klasifikasi berbasis machine learning. Alur penelitian dapat ditunjukkan pada Gambar 1.



Gambar 1. Alur Model.

Dalam penelitian ini menggunakan dataset UNSW-NB15 [24–26], yang terdiri dari 2.540.044 data raw, memiliki 48 fitur dan satu label kelas. Kelas label terdiri dari dua nilai yaitu aktivitas normal dan aktivitas serangan. Dataset awal adalah dalam bentuk file .pcap dan export ke dalam file .csv.

2.1. Data *Cleaning*

Pada tahap ini, dilakukan penghapusan data yang memiliki nilai kosong. Penghapusan dilakukan terhadap baris data atau raw data yang memiliki nilai kosong pada salah satu fitur. Proses ini bertujuan juga untuk mereduksi data sehingga memudahkan dalam pengolahan data.

2.2. Data Transformation

Pada tahap ini dilakukan tranformasi data terhadap jenis data pada fitur yang bersifat kategori menjadi numerik. perubahan ini menggunakan teknik *one-hot-encode* sehingga berdampak pada peningkatan jumlah fitur. Tujuan dari data transformasi adalah menstandarisasi seluruh nilai di setiap fitur menjadi nilai numerik, sehingga memudahkan dalam proses penghitungan pada tahap selanjutnya.

2.3. Data Normalisasi

Pada tahap ini dilakukan proses normalisasi data, di mana data distandarisasi dalam nilai skala 0 hingga 1. Metode normalisasi menggunakan linear scalling, yang ditunjukkan pada persamaan (1)

$$x' = \frac{(x - x_{min})}{(x_{max} - x_{min})}, \quad (1)$$

di mana, x'adalah hasil normalisasi, x adalah nilai data pada setiap fitur, x_{min} adalah nilai minimum dalam seluruh nilai fitur dan x_{max} adalah nilai maksimum pada seluruh nilai fitur.

2.4. Seleksi Fitur

Pada tahap ini dilakukan pemilihan fitur, di mana metode seleksi yang digunakan adalah ANOVA, Univariate dan Chi-square. Pengukuran seleksi fitur ANOVA ditunjukkan pada algoritma 1, seleksi fitur Univariate ditunjukkan pada algoritma 2 dan Chi-square ditunjukkan pada algoritma 3.

Algorithm 1. *ALGORITMA ANOVA*

```
FUNCTION perform_anova_selection(data, target_variable, significance_level):
    fitur_terpilih = []
    FOR each setiap fitur IN data:
        statistik_anova, nilai_p = perform_anova_test(data[fitur], target_variable)
        IF nilai_p <= significance_level:
            fitur_terpilih.append(fitur)
    RETURN fitur_terpilih

FUNCTION perform_anova_test(data_fitur, target_variable):
    # Lakukan uji ANOVA (misalnya, menggunakan scipy.stats.f_oneway)
    statistik_anova, nilai_p = perform_f_oneway_test(data_fitur, target_variable)
    RETURN statistik_anova, nilai_p

FUNCTION perform_f_oneway_test(data_fitur, target_variable):
    # Lakukan uji F (misalnya, menggunakan scipy.stats.f_oneway)
    statistik_f, nilai_p = scipy.stats.f_oneway(data_fitur, target_variable)
    RETURN statistik_f, nilai_p
```

Algorithm 2. *ALGORITMA Univariate*

```
FUNCTION perform_univariate_selection(data, target_variable, num_features):
    skor_fitur = hitung_skor_univariate(data, target_variable)
    fitur_terpilih = pilih_fitur_top_k(skor_fitur, num_features)
    RETURN fitur_terpilih

FUNCTION hitung_skor_univariate(data, target_variable):
    skor_fitur = {}
    FOR each fitur IN data:
        skor = hitung_skor_fitur(data[fitur], target_variable)
        skor_fitur[fitur] = skor
    RETURN skor_fitur

FUNCTION hitung_skor_fitur(fitur_data, target_variable):
    # Hitung skor fitur (misalnya, menggunakan mutual information, chi-square, dll.)
    skor = hitung_skor_misalnya(fitur_data, target_variable)
    RETURNN skor

FUNCTION pilih_fitur_top_k(skor_fitur, k):
    fitur_terpilih = ambil_k_fitur_tertinggi(skor_fitur, k)
    RETURNN fitur_terpilih
```

```
FUNCTION ambil_k_fitur_tertinggi(skor_fitur, k):
    fitur_tertinggi = SORT(skor_fitur, BY=skor, DESCENDING)
    fitur_terpilih = fitur_tertinggi[:k]
    RETURN fitur_terpilih
```

Algorithm 3. *ALGORITMA Chi-square*

```
FUNCTION chi_square_feature_selection(data, target_variable, num_features):
    skor_fitur = hitung_skor_chi_square(data, target_variable)
    fitur_terpilih = pilih_fitur_top_k(skor_fitur, num_features)
    RETURN fitur_terpilih

FUNCTION hitung_skor_chi_square(data, target_variable):
    skor_fitur = {}
    FOR each fitur IN data:
        skor = hitung_skor_chi_square_per_fitur(data[fitur], target_variable)
        skor_fitur[fitur] = skor
    RETURN skor_fitur

FUNCTION hitung_skor_chi_square_per_fitur(fitur_data, target_variable):
    tabel_kontingensi = buat_tabel_kontingensi(fitur_data, target_variable)
    chi2, nilai_p, _, _ = hitung_uji_chi_square(tabel_kontingensi)
    RETURN chi2

FUNCTION buat_tabel_kontingensi(fitur_data, target_variable):
    # Membuat tabel kontingensi untuk uji chi-square
    tabel_kontingensi = COUNT_OCCURRENCES(fitur_data, target_variable)
    RETURN tabel_kontingensi

FUNCTION hitung_uji_chi_square(tabel_kontingensi):
    # Melakukan uji chi-square
    chi2, nilai_p, _, _ = scipy.stats.chi2_contingency(tabel_kontingensi)
    RETURN chi2, nilai_p

FUNCTION pilih_fitur_top_k(skor_fitur, k):
    fitur_terurut = SORT (skor_fitur, BY=skor, DECENDING)
    fitur_terpilih = AMBIL_K_TERATAS(fitur_terurut, k)
    RETURN fitur_terpilih
```

2.5. Fitur *Ranking*

Pada tahap ini, setiap fitur yang diseleksi adalah sebanyak 25%. Sehingga jumlah yang digunakan dalam proses selanjutnya adalah sebanyak 75%. Kemudian fitur terbaik di-*rangking* dan digunakan dalam model klasifikasi decision tree.

2.6. Proses Pemodelan Klasifikasi

Pada proses klasifikasi, data dibagi menjadi dua bagian, yaitu data latih dan data uji. Dalam penelitian ini, komposisi pembagian data menggunakan persentase 80% sebagai data latih dan 20% sebagai data uji. Persamaan decision tree ditunjukkan pada persamaan (2).

$$E(S) = \sum_i^N -P_i * \log_2(P_i), \tag{2}$$

Di mana P_i adalah rasio dari class C_i didalam set data sampel

$$S = \{x_1, x_2, \dots, x_k\} \tag{3}$$

$$P_i = \frac{\sum x_k \in C_i}{S} \tag{4}$$

2.7. Proses Evaluasi Model

Dalam tahap ini dilakukan evaluasi pengukuran model deteksi. Di mana dilakukan penelusuran nilai dari confusion matrix yaitu True Negative yaitu nilai aktivitas normal yang terdeteksi sebagai normal, False Positive yaitu aktivitas normal yang terdeteksi sebagai aktivitas serangan, False Positif adalah aktivitas serangan yang terdeteksi sebagai aktivitas normal dan True Positive adalah aktivitas serangan yang

benar terdeteksi sebagai aktivitas serangan. Kemudian dari confusion matrix, dilakukan perhitungan akurasi, precision dan recall.

3. Hasil dan Pembahasan

Di penelitian ini, model diolah dengan spesifikasi komputer dengan processor core i7, RAM 16 GB dan storage SSD 512GB. Bahasa pemrograman yang digunakan adalah Python versi 3. Dataset yang digunakan dalam penelitian ini ditunjukkan pada Tabel 1.

Tabel 1. Deskripsi Dataset UNSW-NB15.

Dataset	Jumlah Fitur	Jumlah Data	Jenis Class Label
UNSW-NB15	49	2.540.044	Serangan (1); Normal (0)

Proses data *cleaning* adalah proses penghapusan data yang memiliki nilai kosong di salah satu fitur. Di penelitian ini, dataset UNSW-NB15 terdapat 1.452.842 data raw yang memiliki data kosong. Sehingga nilai reduksi data cukup besar yaitu 57%. Selain itu dari sisi fitur, terdapat satu fitur yaitu attack_cat yang dihapus. Penghapusan fitur dilakukan karena dalam penelitian ini hanya berfokus untuk mendeteksi eksistensi serangan dan bukan mendeteksi jenis serangan. Fitur attack_cat merupakan fitur yang digunakan untuk melakukan deteksi jenis serangan. Sehingga dari 49 fitur dasar menjadi 48 fitur.

Setelah proses data *cleaning* dilakukan proses data transformasi, yaitu proses perubahan fitur yang bersifat kategori ke numerik. Di penelitian ini ditemukan 4 fitur bersifat kategori yaitu is_sm_ips_port, service, proto dan state. Sehingga keempat fitur ini ditransformasi menjadi numerik dengan metode *one-hot-encode*. Transformasi ini menyebabkan penambahan fitur sebesar 326.5%. di mana dari 48 fitur menjadi 209 fitur. Perubahan data dari dua proses data *cleaning* dan data transformasi ditunjukkan pada Tabel 2.

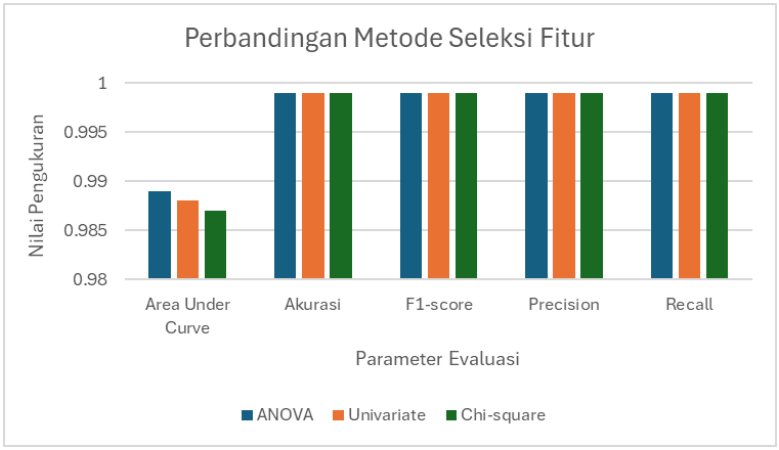
Tabel 2. Deskripsi Dataset UNSW-NB15.

Data Cleanning			Data Transformasi		
Jumlah Data Awal	Setelah Proses Data Cleansing	Persentase Reduksi Data	Jumlah Fitur Awal	Setelah Cleanning	Setelah Transformasi
2.540.044	1.087.202	57%	49	48	209

Setelah melakukan data transformasi, data diubah ke dalam bentuk nilai 0 dan 1. Perubahan data bertujuan untuk menstandarisasi skala nilai dari data di setiap fitur. Setelah seluruh data terstandarisasi, dilakukan seleksi fitur. Di mana fitur yang diseleksi adalah sebanyak 25%, dan yang digunakan adalah sebanyak 75% atau sebanyak 157 fitur yang digunakan dalam proses model klasifikasi. Kemudian fitur diurutkan dengan nilai urutan tertinggi. Pada penelitian ini, seleksi fitur yang menggunakan metode ANOVA memiliki perubahan kelas label menjadi fitur kategori, sedangkan pada metode seleksi fitur univariate dan chi-square mengubah jenis kelas label menjadi numerik. setelah didapatkan 75% fitur terbaik, maka nilai label di seragamkan kembali menjadi jenis kategori. Kemudian dilakukan tahap klasifikasi. Di penelitian ini, proses pengujian menggunakan cross validation dengan k yang digunakan adalah sebanyak 20. Hasil klasifikasi ditunjukkan pada Tabel 3. Perbandingan hasil klasifikasi dari nilai akurasi, AUC, precision, recall dan F1-score ditunjukkan pada Gambar 2.

Tabel 3. Hasil Klasifikasi.

Parameter	Metode		
	ANOVA	Univariate	Chi-square
Area Under Curve	0.989	0.988	0.987
Akurasi	0.999	0.999	0.999
F1-score	0.999	0.999	0.999
Precision	0.999	0.999	0.999
Recall	0.999	0.999	0.999



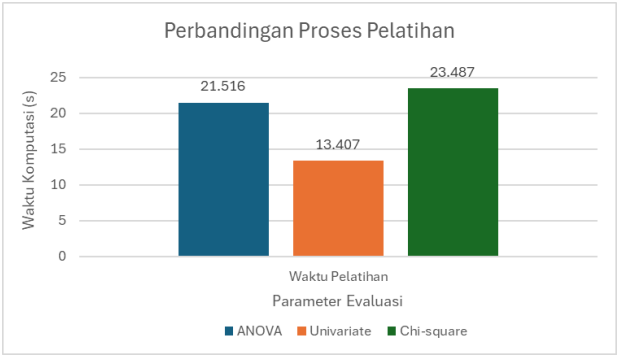
Gambar 2. Hasil Klasifikasi Perbandingan Metode Seleksi Fitur.

Dari hasil klasifikasi, didapatkan bahwa metode seleksi fitur terbaik adalah ANOVA. Di mana parameter terbaik menunjukkan hasil ANOVA pada Are Under Curve (AUC) sebesar 0.989. Selain nilai AUC, nilai precision, recall, F1-score dan akurasi memiliki nilai yang sama dengan metode seleksi fitur univariate dan chi-square yaitu sebesar 0.999. Di penelitian ini, perbandingan analisis waktu komputasi pada proses pelatihan model klasifikasi dan proses pelatihan juga dilakukan, ditunjukkan pada Tabel 4.

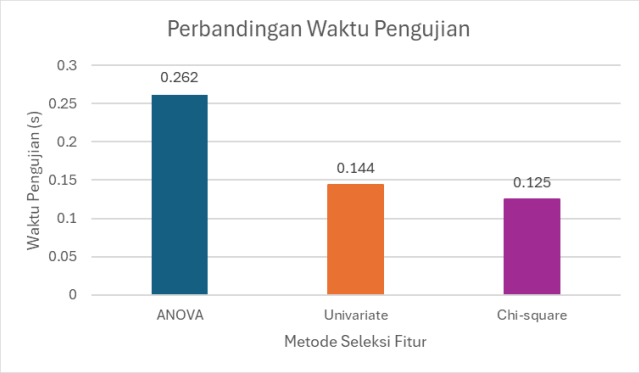
Tabel 4. Analsisi Waktu Komputasi.

Parameter	Metode		
	ANOVA	Univariate	Chi-square
Waktu Pelatihan	21.516	13.407	23.487
Waktu Pengujian	0.262	0.144	0.125

Dari hasil analisis perbandingan waktu komputasi, menunjukkan bahwa dalam proses pelatihan, metode univariate memiliki waktu komputasi yang lebih cepat yaitu sebesar 13.407 detik. Metode tercepat selanjutnya adalah ANOVA sebesar 21.516 detik dan terakhir adalah chi-square sebesar 23.487 detik. Terhadap proses pengujian data tes, waktu tercepat adalah pada metode chi-suare yaitu selama 0.125 detik. Tercepat kedua dalam proses pelatihan adalah metode Univariate yaitu selama 0.144 detik dan terlama dalam pengujian adalah metode ANOVA. Dari ketiga metode seleksi fitur, rata rata tercepat dalam proses pelatihan dan pengujian adalah ditunjukkan oleh metode univariate yaitu selama 6.7755 detik. Sedangkan tercepat kedua adalah ANOVA yaitu selama 10.889 dan tercepat ketiga adalah Chi-square selama 11.806 detik. Hasil pengujian dari sisi waktu pelatihan ditunjukkan pada Gambar 3 dan waktu pengujian ditunjukkan pada Gambar 4.



Gambar 3. Perbandingan Proses Pelatihan.



Gambar 4. Perbandingan Proses Pengujian.

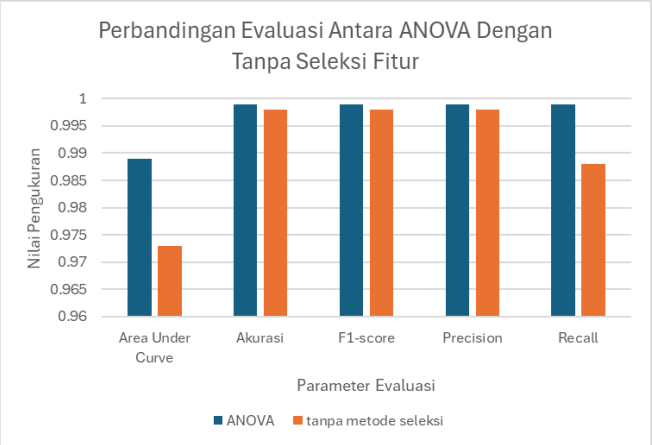
Di penelitian ini, proses seleksi fitur memiliki pengaruh terhadap akurasi deteksi dan waktu komputasi. Sehingga dilakukan analisis perbandingan terhadap klasifikasi antara model yang menggunakan teknik seleksi fitur dan tanpa seleksi fitur. Metode seleksi yang dibandingkan adalah metode terbaik yaitu metode ANOVA. Sehingga hasil perbandingan ditunjukkan pada Tabel 5.

Tabel 5. Hasil Klasifikasi.

Parameter	Metode	
	ANOVA	Tanpa Metode Seleksi Fitur
Area Under Curve	0.989	0.973
Akurasi	0.999	0.998
F1-score	0.999	0.998
Precision	0.999	0.998
Recall	0.999	0.988

Dari hasil perbandingan, penggunaan teknik seleksi fitur dapat mempengaruhi akurasi deteksi, AUC, F1-score, precision dan recall, di mana hasil AUC lebih tinggi sebesar 0.989 dibandingkan dengan tanpa seleksi fitur yang hanya mencapai 0.973 atau mampu meningkat sebesar 1.62%. Sedangkan nilai dari Akurasi, F1-score dan precision mampu ditingkatkan sebesar 0.1 %, di mana tanpa metode seleksi fitur hanya mencapai 0.998 menjadi 0.999. Untuk nilai recall, teknik seleksi fitur mampu meningkatkan sebesar

1.1 % atau ANOVA lebih tinggi sebesar 0.999 dibandingkan tanpa seleksi fitur yang hanya mencapai 0.988. Hasil perbandingan ditunjukkan pada Gambar 5.

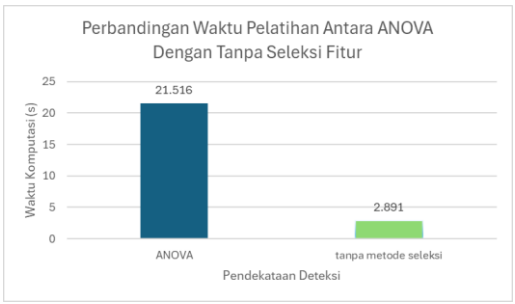


Gambar 5. Perbandingan Proses Pengujian.

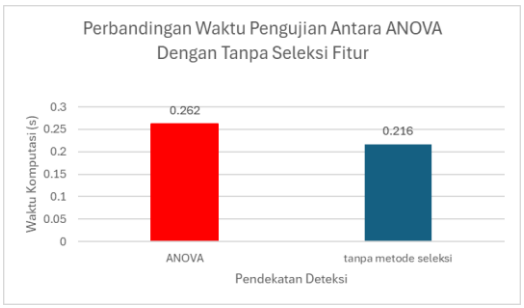
Dari hasil pengukuran waktu, analisis perbandingan menunjukkan bahwa dengan teknik seleksi fitur, model klasifikasi memiliki waktu komputasi pelatihan dan pengujian lebih lama dibandingkan dengan tanpa seleksi fitur. Hal ini terlihat lebih logis karena model melakukan proses pemilihan dan perankingan fitur terbaik. Hasil perbandingan waktu komputasi ditunjukkan pada Tabel 6 dan Gambar 6.

Tabel 6. Hasil Klasifikasi.

Parameter	Metode	
	ANOVA	Tanpa Seleksi Fitur
Waktu Pelatihan	21.516	2.891
Waktu Pengujian	0.262	0.216



(a)



(b)

Gambar 6. (a) Perbandingan Pelatihan ; (b) Perbandingan Proses Pengujian.

Secara keseluruhan, performa model klasifikasi dapat meningkatkan akibat adanya proses seleksi fitur. Namun penggunaan seleksi fitur mempengaruhi waktu komputasi, di mana proses seleksi fitur menambah waktu komputasi.

4. Kesimpulan

Penelitian ini mengusulkan analisis perbandingan dari beberapa metode seleksi fitur terhadap model klasifikasi. Beberapa metode seleksi fitur yang dibandingkan adalah ANOVA, univariate, dan chis -square. Metode seleksi fitur terbaik dalam penelitian ini dihasilkan oleh metode ANOVA di mana nilai AUC yang dicapai adalah sebesar 0.89, akurasi sebesar 0.999, precision sebesar 0.999 F1-score sebesar

0.999 dan recall sebesar 0.999. waktu komputasi terbaik adalah dicapai oleh metode univariate di mana waktu pelatihan dicapai sebesar 13.407 detik dan waktu pengujian selama 0.144 detik. Metode seleksi fitur terbaik yaitu ANOVA mampu meningkatkan performa model klasifikasi dibandingkan dengan tanpa metode seleksi fitur khususnya adalah pada nilai AUC sebesar 1.62 %, precision, F1-score dan akurasi sebesar 0.1% dan recall sebesar 1.1%. dari sisi waktu komputasi, penggunaan seleksi fitur memiliki waktu yang lebih lama dibandingkan dengan tanpa seleksi fitur. Di Penelitian selanjutnya akan dikembangkan model deteksi dengan membandingkan pengaruh terhadap metode klasifikasi dengan metode ANOVA. Dari sisi waktu komputasi, penelitian selanjutnya akan mempertimbangkan keseimbangan antara waktu komputasi dengan kinerja prediksi pada pengolahan data yang bersifat besar dan *realtime*. Selain itu penelitian selanjutnya akan menganalisis perbandingan pengaruh teknik data transformasi terhadap beberapa metode klasifikasi pada model deteksi serangan yang menggunakan analisis trafik jaringan.

Daftar Pustaka

[1] J. Velasco-Mata, V. Gonzalez-Castro, E. F. Fernandez, and E. Alegre, "Efficient Detection of Botnet Traffic by Features Selection and Decision Trees," *IEEE Access*, vol. 9, pp. 120567–120579, 2021, doi: 10.1109/ACCESS.2021.3108222.

[2] F. Hussain, S. G. Abbas, I. M. Pires, S. Tanveer, U. U. Fayyaz, N. M. Garcia, G. A. Shah, and F. Shahzad, "A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks," *IEEE Access*, vol. 9, pp. 163412–163430, 2021, doi: 10.1109/ACCESS.2021.3131014.

[3] A. Muhammad, M. Asad, and A. R. Javed, "Robust Early Stage Botnet Detection using Machine Learning," *1st Annu. Int. Conf. Cyber Warf. Secur. ICCWS 2020 - Proc.*, 2020, doi: 10.1109/ICCWS48432.2020.9292395.

[4] M. Choubisa, "A Simple and Robust Approach of Random Forest for Intrusion Detection System in Cyber Security," pp. 5–9, 2022.

[5] G. Zhu, H. Yuan, Y. Zhuang, Y. Guo, X. Zhang, and S. Qiu, "Research on network intrusion detection method of power system based on random forest algorithm," *Proc. - 2021 13th Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2021*, pp. 374–379, 2021, doi: 10.1109/ICMTMA52658.2021.00087.

[6] A. Kumar and T. J. Lim, "EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques," *IEEE 5th World Forum Internet Things, WF-IoT 2019 - Conf. Proc.*, pp. 289–294, 2019, doi: 10.1109/WF-IoT.2019.8767194.

[7] H. T. Nguyen, Q. D. Ngo, D. H. Nguyen, and V. H. Le, "PSI-rooted subgraph: A novel feature for IoT botnet detection using classifier algorithms," *ICT Express*, vol. 6, no. 2, pp. 128–138, 2020, doi: 10.1016/j.icte.2019.12.001.

[8] G. Xiao, J. Li, Y. Chen, and K. Li, "MalFCS: An effective malware classification framework with automated feature extraction based on deep convolutional neural networks," *J. Parallel Distrib. Comput.*, vol. 141, pp. 49–58, 2020, doi: 10.1016/j.jpdc.2020.03.012.

[9] H. Darabian, A. Dehghantanha, S. Hashemi, S. Homayoun, and K. K. R. Choo, "An opcode-based technique for polymorphic Internet of Things malware detection," *Concurr. Comput. Pract. Exp.*, vol. 32, no. 6, 2020, doi: 10.1002/cpe.5173.

[10] G. D'Angelo, M. Ficco, and F. Palmieri, "Association rule-based malware classification using common subsequences of API calls," *Appl. Soft Comput.*, vol. 105, p. 107234, 2021, doi: 10.1016/j.asoc.2021.107234.

[11] Z. S. Malek, "User behavior Pattern -Signature based Intrusion Detection," vol. 7, pp. 549–552, 2020.

[12] F. H. Almasoudy, W. L. Al-Yaseen, and A. K. Idrees, "Differential Evolution Wrapper Feature Selection for Intrusion Detection System," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 1230–1239, 2020, doi: 10.1016/j.procs.2020.03.438.

[13] C. M. Ou, "Host-based Intrusion Detection Systems Inspired by Machine Learning of Agent-Based Artificial Immune Systems," *IEEE Int. Symp. Innov. Intell. Syst. Appl. INISTA 2019 - Proc.*, pp. 1–5, 2019, doi: 10.1109/INISTA.2019.8778269.

[14] B. Sergey, "Intrusion Detection System and Intrusion Prevention System with Snort provided by Security Onion .," *Bachelor's Thesis Inf. Technol. MAMK Univ. Appl. Sci.*, no. May, 2016.

[15] H. Alnabulsi, M. R. Islam, and Q. Mamun, "Detecting SQL injection attacks using SNORT IDS," *Asia-Pacific World Congr. Comput. Sci. Eng. APWC CSE 2014*, no. November, 2014, doi: 10.1109/APWCCSE.2014.7053873.

[16] N. Khamphakdee, N. Benjamas, and S. Saiyod, "Improving Intrusion Detection System Based on Snort Rules for Network Probe Attacks Detection with Association Rules Technique of Data Mining," *J. ICT Res. Appl.*, vol. 8, no. 3, pp. 234–250, 2015, doi: 10.5614/itbj.ict.res.appl.2015.8.3.4.

[17] A. A. A, A. Ademola, and A. A. A, “Development Of An SMS Based Alert Systemusing Object Oriented Design Concept,” vol. 3, no. 5, pp. 71–76, 2014.
[18] S. Ouiazzane, M. Addou, and F. Barramou, “A Multi-Agent Model for Network Intrusion Detection,” *ICSSD 2019 - Int. Conf. Smart Syst. Data Sci.*, 2019, doi: 10.1109/ICSSD47982.2019.9003119.
[19] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M. Lahza, “Improving performance of intrusion detection system using ensemble methods and feature selection,” *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3167918.3167951.
[20] M. N. Aziz and T. Ahmad, “Clustering under-sampling data for improving the performance of intrusion detection system,” *J. Eng. Sci. Technol.*, vol. 16, no. 2, pp. 1342–1355, 2021.
[21] S. Anwar, J. M. Zain, M. F. Zolkipli, Z. Inayat, S. Khan, B. Anthony, and V. Chang, “From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions,” *Algorithms*, vol. 10, no. 2, 2017, doi: 10.3390/a10020039.
[22] A. A. Megantara and T. Ahmad, “ANOVA-SVM for Selecting Subset Features in Encrypted Internet Traffic Classification,” *Int. J. Intell. Eng. Syst.*, vol. 14, no. 2, pp. 536–546, 2021, doi: 10.22266/ijies2021.0430.48.
[23] J. Lee, D. Park, and C. Lee, “Feature selection algorithm for intrusions detection system using sequential forward search and random forest classifier,” *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 10, pp. 5132–5148, 2017, doi: 10.3837/tiis.2017.10.024.
[24] N. Moustafa and J. Slay, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” In: *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.
[25] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, “NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems,” In: *Big Data Technologies and Applications*, 2021, pp. 117–135.
[26] N. Moustafa, G. Creech, and J. Slay, “Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models,” in *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*, I. Palomares Carrascosa, H. K. Kalutarage, and Y. Huang, Eds. Cham: Springer International Publishing, 2017, pp. 127–156. doi: 10.1007/978-3-319-59439-2_5.