 ISSN:2302-8149
e-ISSN:2540-9719

Analisa Performa Penggunaan Feature Selection untuk Mendeteksi Intrusion Detection Systems dengan Algoritma Random Forest Classifier

Setiawan Budiman*, Andi Sunyoto, Asro Nasiri
Magister Teknik Informatika, Universitas AMIKOM Yogyakarta
Jl. Ring Road Utara. Sleman, Yogyakarta, Indonesia
*e-mail: setiawan.1267@students.amikom.ac.id

(***received*:** 30 Juli 2021, ***revised*:** 5 September 2021, ***accepted*:** 28 September 2021)

Abstrak

Semakin penting koneksi data melalui Internet membuat kebutuhan akan keamanan jaringan data semakin meningkat. Salah satu tools yang penting adalah *Intrusion detection systems* (IDS). Salah satu hal yang menjadi masalah dari penggunaan IDS adalah performan kecepatan untuk mendeteksi data yang semakin banyak dalam waktu yang singkat. Dalam penelitian ini kami akan melakukan analisa perbandingan performa IDS menggunakan *features selection* dengan algoritma *Random Forest Classifier* yang disimulasikan pada *dataset* UNSW-NB15, yaitu *dataset* simulasi serangan pada jaringan *network* yang dikembangan oleh Nour Moustafa & Jill Slay dari *University of New South Wales* pada *Australian Defence Force Academy*. Tujuan dari penelitian ini adalah mempercepat waktu proses *Intrusion detection systems* dengan *machile learning*. Penelitian dilakukan dengan 2 tahap, yaitu tahap pertama tanpa *features selection* dan tahap kedua dengan *features selection ExtraTreesClassifier*. Masing-masing tahap dilakukan dengan beberapa kali pengujian dengan persentasi *testing* dan *training* data yang berbeda. Hasil penelitian menunjukan bahwa penggunaan *features selection* dapat mempercepat waktu proses pendeteksian dengan menggunakan *Random Forest Classifier*, walaupun ada sedikit penurun akurasi dibawah 1%.

Kata kunci: *feature selection, random forest, ids, machine learning*

Abstract

Internet data connection is very important, therefore it will increasing the security issues. One of the important tools is Intrusion detection systems (IDS). The main problems of using IDS is the speed performance to detect more and more data in a short time. In this study, we will perform a comparative analysis of IDS performance using features selection with the Random Forest Classifier algorithm which is simulated on the UNSW-NB15 dataset, which is work as the attack simulation dataset on the network developed by Nour Moustafa & Jill Slay from the University of New South Wales at the Australian Defense Force Academy. The purpose of this research is to speed up the processing time of Intrusion detection systems with machile learning. The research was conducted in 2 stages, the first stage without features selection and the second stage with features selection. Each stage is carried out with several study using different percentages of testing and training data. The results showed that by using features selection, it can speed up the detection process time using the Random Forest Classifier, although there is a slight decrease in accuracy below 1%.

Keywords: *feature selection, random forest, ids, machine learning*

1 Pendahuluan

Dengan semakin berkembangnya koneksi Internet di seluruh dunia, maka saat ini semakin banyak juga serangan yang terjadi pada system jaringan Internet. Serangan pada Internet dilakukan untuk mendapatkan keuntungan baik secara finansial maupun keuntungan lainnya seperti permasalahan sosial dan politik. Solusi untuk permasalahan ini adalah dengan dikembangkannya teknologi *intrusion detection systems (IDS), firewall* dan beberapa *tools* lainnya. Penelitian ini akan

berfokus pada penggunaan *intrusion detection systems (IDS)* untuk mendeteksi apakah ada komunikasi data yang anomali atau mencurigakan. Dengan adanya informasi serangan dari IDS, *system firewall* dapat segera melakukan tindakan preventif dengan cara melakukan *blocking* pada aktivitas *network* tersebut. Saat ini penggunaan IDS semakin diperlukan terutama untuk jaringan komunikasi sibuk yang terhubung ke seluruh dunia. Dengan semakin banyaknya data yang harus di pelajari oleh IDS, maka salah satu kunci keberhasilannya adalah akurasi dan kecepatan waktu prosesnya[1].

Saat ini sudah tersedia banyak *dataset* dengan kualitas yang baik untuk mewakili hasil kerja dari sebuah IDS. Pada study ini digunakan *dataset* UNSW-NB15 yang dikembangan oleh Nour Moustafa & Jill Slay dari *University of New South Wales* pada *Australian Defence Force Academy*[2]. *Dataset* ini memiliki sekitar 2 juta data yang mewakili 9 jenis serangan dengan total 47 *features.*

Penelitian ini akan melakukan sebuah analisa untuk membandingkan kecepatan waktu proses pendeteksian data dari IDS antara menggunakan *features selection* dibandingkan dengan tidak menggunakan *feature selection*. Basis dari penelitian ini adalah menggunakan algoritma *Random Forest Classifier*, termasuk untuk *feature selection* dan klasifikasinya. Dari beberapa penelitian disebutkan bawah penggunaan *Random Forest Classifier* memberikan hasil yang lebih baik untuk permodelan pada *Intrusion Detection Systems* seperti penelitian yang dilakukan dengan judul *A Survey of Random Forest Based Methods for Intrusion Detection Systems*[3]. *Random forest* akan membuat permodelan dengan beberapa *decision tree*. *Feature selection* yang digunakan adalah dengan menggunakan *library sklearn ExtraTreesClassifier* yang berbasis pada *ensemble learning* dengan menggunakan *decision tree*.

2 Tinjauan Literatur

2.1 Pengembangan Intrusion Detection Systems

Beberapa penelitian sejenis yang telah dilakukan menggunakan data *intrusion detection systems* (IDS) memiliki tujuan utama yang sama, yaitu dengan menganalisa serta meningkatkan performa machine learning tersebut. Penelitian dengan judul *Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset[4]* dilakukan untuk memberikan peningkatan hasil dengan menggunakan XGBoost-based *feature selection* sebelum melakukan deteksi dengan *machine learning Support Vector Machine* (SVM), *k-Nearest-Neighbour* (kNN), *Logistic Regression* (LR), *Artificial Neural Network* (ANN) dan *Decision Tree* (DT). Hasil penelitian tersebut adalah *features selection* XGBoost-based dapat memberikan hasil yang lebih baik pada *machine learning Decession Tree* (DT). Penelitian lainnya dengan judul *Network Intrusion Detection: A Comparative Study Using State-of-the-art Machine Learning Methods*[5] telah dilakukan untuk mengetahui hasil penggunaan *dataset* IDS yang dideteksi dengan beberapa algoritma *machine learning* yaitu *Logistic regression*, SGD, Light GBM, XG-Boost, DNN, *Stacked classifier*. Dan hasil penelitian tersebut adalah bahwa *Gradient Boosting Decision Tree* akan memberikan hasil yang lebih baik. Penggunaan beberapa algoritma *machine learning* dengan metode *ensemble methods* dapat memberikan hasil akurasi yang lebih baik dibandingkan menggunakan single klasifikasi. Hal ini dapat dilihat dari hasil penelitian dengan judul *Use The Ensemble Methods When Detecting DoS Attacks in Network Intrusion Detection*[6].

2.2 Dataset Penelitian

Ada beberapa jenis *dataset* yang dapat digunakan pada penelitian yang berhubungan dengan *intrusion detection systems* (IDS). Salah satu yang memiliki kelengkapan data adalah UNSW-NB15. *Dataset* ini dapat didownlaod dari www.kaggle.com. Penelitian dengan judul *Analysis of UNSW-NB15 Dataset Using Machine Learning Classifiers*[7] telah dilakukan yang tujuannya untuk mendeteksi hasil *machine learning* dengan beberapa algoritma, yaitu *Naive Bayes*, *Logistic Regression*, SMO, J48 dan *Random Forest*. Dan *Random Forest* dapat memberikan hasil akurasi yang paling baik dibanding lainnya. Pada beberapa penelitian lain, juga dilakukan dengan menggunakan *dataset* lain seperti dataset KDDCUP99 tetapi *dataset* ini sudah mulai banyak ditinggalkan karena *out of date*, karena sudah tidak dapat mewakili perkembangan simulasi IDS dengan berbagai macam kasus dan teknik serangan yang lebih *modern*[8]. *Dataset* lain yang bisa digunakan pada sebuah

penelitian berbasis IDS adalah NSL-KDD. Ketiga *dataset* ini sudah dilakukan perbandingan menggunakan machine learning dengan algoritma *Deep Neural Network* (DNN) dan hasilnya sama-sama memberikan akurasi kurang lebih mencapai 90%[9].

2.3 Feature Selection

Pada era IoT dan aplikasi berbasis *web-based*, produksi berbagai macam data penjadi sangat cepat dan ini mengakibatkan terbentuknya *big data* yang terus bertambah secara otomatis. Hal ini menyebabkan akan ditemukan banyak data *noise* yang tidak berguna, *redundancy* dan membutuhkan teknologi komputasi yang besar untuk memproses data. Oleh karena itu diperlukan metode *feature selection* untuk mengurangi jumlah *features* yang *noisy, redundant* dan *irrelevant data*[10]. *Feature selection* juga digunakan pada penelitian dengan judul *Intrusion Detection Model Using Fusion Of Chi-Square Feature Selection And Multi Class SVM*[11]. *Dataset* IDS yang digunakan adalah NSL-KDD. Hasil dari penelitian ini terlihat bahwa dengan menggunakan *Chi-Square feature selection* dapat memberikan percepatan waktu proses dalam algoritma *machine learning Multi Class SVM*. Selain itu juga terdapat *Feature Selection* dengan menggunakan *Chi-Square* yang berfokus pada pendeteksian serangan DDoS dengan judul Peningkatan Akurasi Pendeteksian Serangan DDoS Menggunakan *Multiclassifier Ensemble Learning* dan *Chi-Square*[12] dengan mengunakan *dataset* NSL-KDD. *Feature selection* semakin banyak digunakan untuk melakukan optimalisasi sebelum dilakukan analisa akhir, salah satunya pada penelitian menggunakan *Naive Bayes Feature Selection* yang kemudian dilanjutkan dengan klasifikasi *machine learnin*g menggunakan algoritma SVM telah dilakukan dengan judul *An effective intrusion detection approach using SVM with na¨ive Bayes feature embedding*[13].

2.4 Random Forest

Random forest adalah salah satu teknik klasifikasi *(supervised classification)* yang banyak digunakan diberbagai macam penelitian dan model kasus. Ini adalah sebuah proses pengambilan keputusan berdasarkan pencabangan pohon atau *decision tree*. Tiap cabang berisi tentang pertanyaan untuk memecahkan sebuah keputusan dari jumlah pencabangan ideal[14]. Terdapat beberapa penelitian yang menyebutkan bahwa *Random Forest* dapat memberikan hasil akurasi yang baik pada penelitian dengan judul *A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification* [15]. Penelitian tersebut dilakukan untuk mengetahui bagaimana akurasi jika dibandingkan dengan KNN dan *Logistic Regression*. Hasilnya didapatkan bahwa *Random Forest* masih lebih baik dibandingkan dengan KNN, tetapi pada penelitian tersebut *Logistic Regression* masih memiliki akurasi terbiak, oleh karena itu perlu dibuat penelitian lebih lanjut yang mencari parameter terbaik agar *Random Forest* menjadi lebih akurat. Sedangkan penelitian yang dilakukan dengan judul *Use of Random Forest in The Identification of Important Variables*[16] berfokus pada pencarian *variable* yang lebih penting dengan algoritma *Random Forest*, pada penentuan *data crude oil* yang dilihat berdasarkan *variable* yang paling sering digunakan dari pencabangan pohon tersebut. *Random Forest* juga berguna untuk mengolah klasifikasi *big data* dan IoT yang telah dilakukan pada penelitian dengan judul *Random Forest for Big Data Classification in The Internet of Things Using Optimal Features*[17] yang dilakukan pada bidang kesehatan yang menggunakan data kesehatan pasien. Pengunaan IoT untuk mengumpulkan data dapat mengakibatkan *feedback data* yang besar, itu sebabnya penggunaan *machine learning* dengan alrogitma *Random Forest* dapat membantu klasifikasi *data* secara akurat. Penelitian lain yang berhubungan dengan IoT pada bagian *wireless sensor network* dilakukan dengan judul *Fault Detection in Wireless Sensor Networks through the Random Forest Classifie*[18], penelitian ini dilakukan untuk mendapatkan hasil klasifikasi terbaik antara beberapa jenis algoritma seperti SVM, RF, SGD, MLP, CNN, dan PNN untuk memberikan *data* kegagalan dan kerentanan yang dibutuhkan untuk memperbaiki dengan *system update* dan peningkatan kinerja. Hasil yang didapatkan bahwa *Random Forest* (RF) memberikan hasil yang terbaik dibanding algoritma lainnya.

3 Metode Penelitian

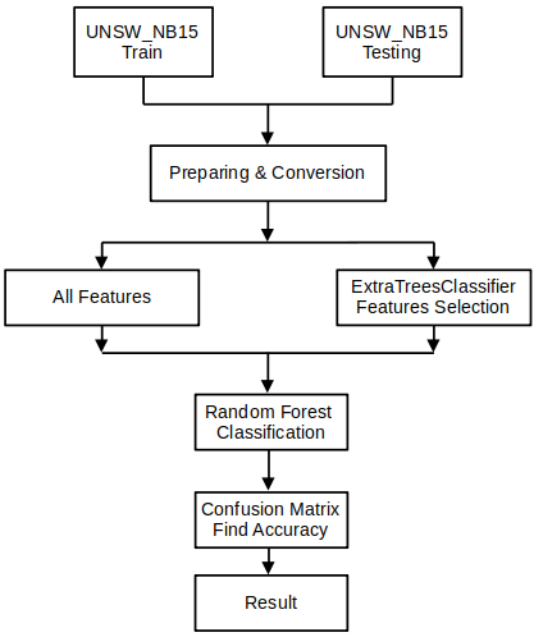
Penelitian ini menggunakan metode yang bersifat kuantitatif dan komparatif. Hal ini dilakukan karena akan membandingkan beberapa data dengan bahasa program *Python* yang dijalan dengan

 ISSN:2302-8149
e-ISSN:2540-9719

menggunakan beberapa *library* seperti sklearn dan *library* pendukung lainnya. Penelitian ini akan melakukan analisis pada suatu jaringan apakah kondisnya normal atau anomali (bermasalah).

Dataset yang digunakan adalah UNSW-NB15 yang tersedia di *Kaggel* oleh Nour Moustafa dan Jill Slay di *University of New South Wales* yang dikembangkan bersama *Australian Centre for Cyber Security (ACCS)* untuk membuat simulasi serangan meggunakan *software IXIA PerfectStorm tool*. *Dataset* ini memiliki 47 *features* dengan 9 jenis serangan. Dari seluruh serangan tersebut, dapat di kategorikan dalam 2 besar yaitu kondisi jaringan normal atau anomali[2].

Langkah yang dilakukan dalam penelitian ini adalah dengan menguplaod data csv UNSW-NB15 agar dapat dibaca oleh *Python. Dataset* yang digunakan adalah gabungan dari UNSW-NB15 *training* dan *testing* dengan total 257.673 baris. Hal ini dilakukan agar data penelitian ini memiliki data yang cukup besar. Proses berikutnya adalah konversi beberapa *features* dari *object* menjadi *numeric* agar dapat dilakukan proses *classfication*. Setelah itu penelitian ini akan dilakukan menjadi 2 bagian, yaitu pertama dengan *full features* dan yang kedua menggunakan *feature selection*. Tiap bagian akan disajikan dalam tabel perbandingan hasil akhir dari *confussion matrix*. Pada proses pemilihan *feature* menggunakan *Feature Selection Extra Trees Classifier* pada *library sklearn.ensemble ExtraTreesClassifier*, yaitu menentukan *feature selection* yang berbasis pada *RandomForestClassifier*, untuk mendapatkan *features* dengan kontribusi sedikit terhadap hasil label yang dapat memberikan akibat *overfit* serta mempengaruhi lamanya waktu proses akibat melakukan analisa pada *feature* yang tidak berpengaruh banyak pada hasil *classification*. Selanjutnya dilakukan pembagian data untuk *testing* dan *training* yaitu dengan komposisi 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%, sehingga total akan dilakukan dengan 5 set data yang dibagi secara acak. *Feature* yang ada pada data *traning* dan *testing* akan dipilih sesuai hasil *output* dari proses *Feature Selection Extra Trees Classifier*. Langkah berikutnya adalah melakukan prediksi dengan menggunakan *Random Forest Classifier* untuk menentukan hasil *accuracy* dan *precission* dengan menggunakan *Confusion Matrix*.



Gambar 1. Alur Penelitian

4 Hasil dan Pembahasan

Hasil dari penelitian ini kami tuliskan dapat bentuk beberapa tabel dan grafik agar memudahkan proses membaca hasil. Proses penelitian ini dibagi menjadi 2 bagian, yaitu proses awal sampai akhir dengan menggunakan semua *feature* sebanyak 41 yang ada pada *dataset* UNSW-NB15 dengan jumlah 257.673 baris. Setelah dilakukan konversi dengan sklearn.preprocessing LabelEncoder, maka

 ISSN:2302-8149
e-ISSN:2540-9719

hasilnya seluruh *type object* telah diubah menjadi *type integer*. Selanjutnya dilakukan penghapusan *features* bertype *object*, sehingga hasilnya dapat dilihat pada tabel 1.

Tabel 1. Hasil Konversi Features

No.	Feature	Type	No.	Feature	Type
0	id	float64	22	ackdat	float64
1	spkts	int64	23	smean	int64
2	dpkts	int64	24	dmean	int64
3	sbytes	int64	25	trans_depth	int64
4	dbytes	int64	26	response_body_len	int64
5	rate	float64	27	ct_srv_src	int64
6	sttl	int64	28	ct_state_ttl	int64
7	dttl	int64	29	ct_dst_ltm	int64
8	sload	float64	30	ct_src_dport_ltm	int64
9	dload	float64	31	ct_dst_sport_ltm	int64
10	sloss	int64	32	ct_dst_src_ltm	int64
11	dloss	int64	33	is_ftp_login	int64
12	sinpkt	float64	34	ct_ftp_cmd	int64
13	dinpkt	float64	35	ct_flw_http_mthd	int64
14	sjit	float64	36	ct_src_ltm	int64
15	djit	float64	37	ct_srv_dst	int64
16	swin	int64int64	38	is_sm_ips_ports	int64
17	stcpb	int64int64	39	proto_n	int64
18	dtcpb	int64int64	40	service_n	int64
19	dwin	int64	41	state_n	int64
20	tcprtt	float64			
21	synack	float64			

Pada bagian pertama, penulis tidak melakukan proses *feature selection*, sehingga semua 41 *features* pada tabel 1 akan digunakan. *Data* UNSW-NB15 dibagi menjadi menjadi *training* dan *testing*. Proses ini akan dilakukan sebanyak 5 pembagian yaitu dengan komposisi 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%. Langkah berikutnya adalah melakukan klasifikasi dengan *Random Forest*, percobaan dilakukan masing masing sebanyak 4 kali dan didapatkan hasil akurasi terbaik sebagai yang pada dilihat pada tabel 2.

Pada bagian kedua, agar data yang digunakan tetap konsisten, penelitian ini dilanjutkan dengan menggunakan komposisi *dataset* yang sudah dibagi sebelumnya untuk dimasukkan pada pemilihan *feature selection*. Hasil *output* dari proses *Feature Selection Extra Trees Classifier* didapatkan pengurangan sebanyak 13 *features*. Selanjutnya program akan menghilangkan ke 13 *feature* yang tidak mempunyai pengaruh dengan nilai dibawah 0.01, seperti yang terlihat pada tabel 3. Kemudian

 ISSN:2302-8149
e-ISSN:2540-9719

dilanjutkan lagi untuk proses klasifikasi dengan *Random Forest* yang dilakukan sebanyak 4 kali. Hasil terbaiknya dituliskan pada tabel 2. Penulis menentukan batas nilai kurang 0.01 untuk dihapus karena hasil output dari *Feature Selection Extra Trees Classifier* rata-rata berkisar antara dari 0.1 sampai 0.01, dimana semakin kecil jarak selisihnya, maka semakin tidak berguna *feature* tersebut. Contoh *feature* yang dihapus memiliki nilai *output* 0.00804461, 0.00770632 dan 0.00630018.

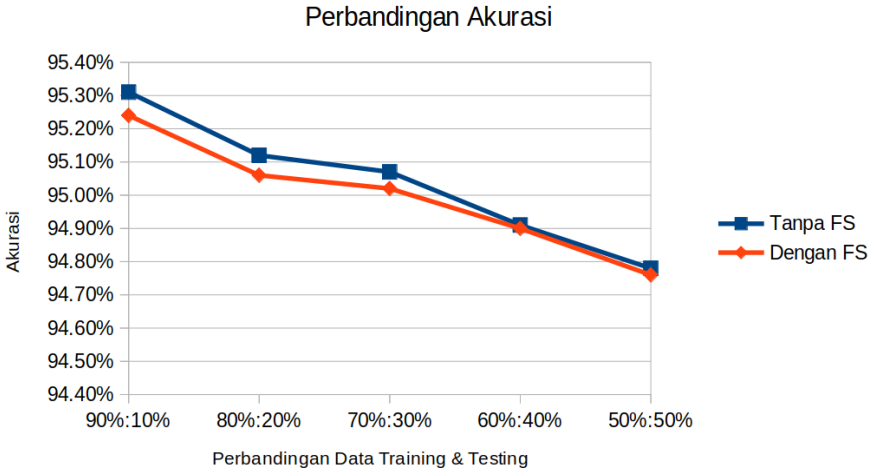
Tabel 2. Hasil Klasifikasi Random Forest Dengan dan Tanpa Feature Selection (FS)

No.	Komposisi	Akurasi		Waktu Proses	
		Tanpa FS	Dengan FS	Tanpa FS	Dengan FS
1.	90%:10%	95.31%	95.24%	50.08s	45.13s
2.	80%:20%	95.12%	95.06%	45.52s	40.96s
3.	70%:30%	95.07%	95.02%	39.94s	36.17s
4.	60%:40%	94.91%	94.90%	39.35s	33.00s
5.	50%:50%	94.78%	94.76%	27.70s	24.86s

Tabel 3. Feature Dengan Kontribusi Paling Sedikit

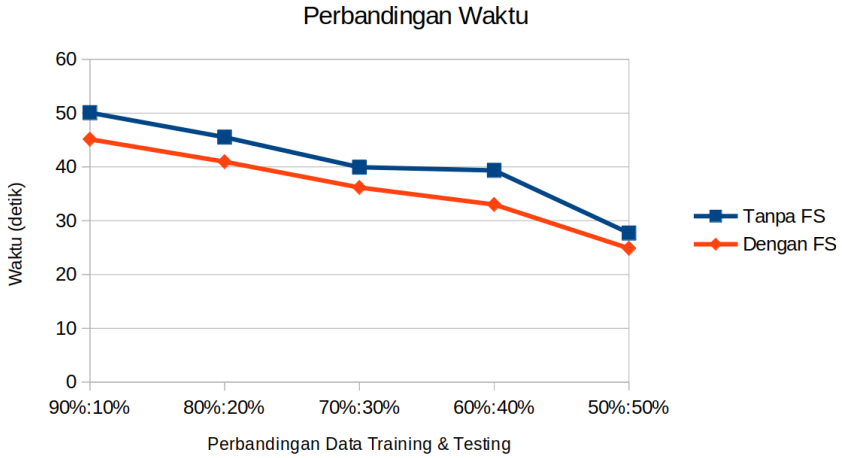
No.	Feature	Type	No.	Feature	Type
0	ct_ftp_cmd	int64	8	dbytes	int64
1	is_ftp_login	int64	9	dinpkt	int64
2	response_body_len	int64	10	djit	float64
3	ct_flw_http_mthd	int64	11	sjit	float64
4	trans_depth	int64	12	dpkts	int64
5	spkts	int64			
6	dloss	int64			
7	sloss	int64			

Untuk memberikan gambaran hasil perbandingan yang lebih mudah, maka data output dari penelitian menggunakan *dataset* UNSW-NB15 dengan komposisi yang terdiri dari 5 set pembagian data *training* dan *testing* yaitu 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50% disajikan juga dalam bentuk grafik yang dapat dilihat pada gambar 2 untuk perbandingan akurasi dan pada gambar 3 untuk perbandingan waktu saat tanpa menggunakan *feature selection* dan dengan menggunakan *feature selection*. Untuk hasil akurasi terbaik didapatkan dengan menggunakan data *training* sebanyak 90% dan data *testing* sebanyak 10% yang terlihat pada gambar 4 di bawah ini.

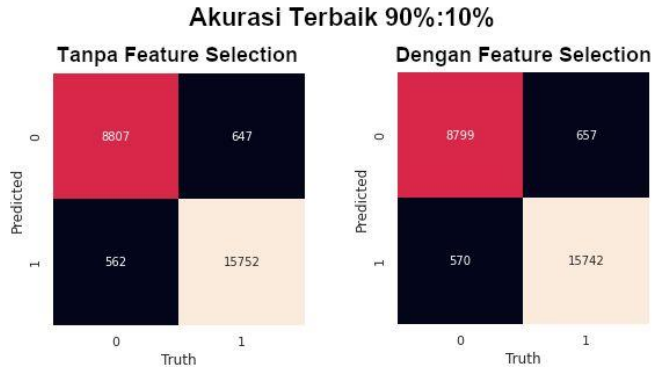


 ISSN:2302-8149
e-ISSN:2540-9719

Gambar 2. Perbandingan Akurasi Antara Tanpa dan Dengan Features Selection



Gambar 3. Perbandingan Waktu Proses Antara Tanpa dan Dengan Features Selection



Gambar 4. Akurasi Terbaik Dengan dan Tanpa Feature Selection

Dari hasil tabel penelitian ini, terlihat bahwa akurasi terbaik didapatkan ada data *training* sebanyak 90% dan data *testing* sebanyak 10% karena semakin banyak data *training* yang diberikan, maka semakin banyak contoh data yang dapat dijadikan acuan oleh *machine learning* untuk menetukan kesalahan pendeteksian. Hasil penelitian didapatkan bahwa waktu proses dengan menggunakan *feature selection* menjadi lebih cepat. Tetapi ada beberapa *features* penting yang tercampur dengan *data noise*, namun jumlahnya sangat sedikit sehingga tidak dapat dibedakan saat diproses dengan *Feature Selection Extra Trees Classifier*. Akibatnya hasil akurasinya menjadi sedikit berkurang namun sangat kecil sekali, hanya kurang dari 1%. Oleh karena itu, penggunaan *feature selection* masih memberikan manfaat yang lebih baik dibandingkan tanpa *feature selection*, yaitu pada kecepatan proses yang lebih baik.

5 Kesimpulan

Penelitian ini dilakukan untuk membandingkan performa dari algoritma *machine learning Random Forest* dengan menggunakan *feature selection* atau tanpa *feature selection*. *Dataset* yang digunakan adalah UNSW-NB15 dengan kategori *label* normal atau anomali. Jumlah *dataset* yang digunakan adalah 257.673 *record*. Proses penelitian dilakukan dengan beberapa kali ujicoba menggunakan perbandingan data *traning* dan *testing* sebanyak 90%:10%, 80%:20%, 70%:30%, 60%:40%, 50%:50%. Hasil yang didapatkan bahwa penggunaan *feature selection* berbasis *decision tree ExtraTreesClassifier* akan memberikan waktu proses yang lebih cepat dibandingkan tanpa *feature selection*, walaupun terdapat penurunan hasil akurasi kurang dari 1% jika dibandingkan tanpa *feature selection*. Peningkatan kecepatan ini memberikan dampak yang lebih penting untuk memberikan efisiensi waktu pada deteksi data IDS dan berkurangnya *load* kinerja *hardware*. Untuk hasil akurasi terbaik didapatkan dengan menggunakan data *training* sebanyak 90% dan data *testing* sebanyak 10%. Untuk penelitian yang akan datang, penelitian dapat menggunakan *feature selection* jenis lainnya untuk mendapatkan hasil kecepatan dan akurasi yang lebih baik.

ISSN:2302-8149
e-ISSN:2540-9719

Referensi

[1] I. Sumaiya Thaseen, J. Saira Banu, K. Lavanya, M. Rukunuddin Ghalib, and K. Abhishek, "An integrated intrusion detection system using correlation-based attribute selection and artificial neural network," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 2, pp. 1–15, 2021, doi: 10.1002/ett.4014.

[2] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015 - Proc.*, no. November, 2015, doi: 10.1109/MilCIS.2015.7348942.

[3] P. A. A. Resende and A. C. Drummond, "A survey of random forest based methods for intrusion detection systems," *ACM Comput. Surv.*, vol. 51, no. 3, 2018, doi: 10.1145/3178582.

[4] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00379-6.

[5] M. Rai and H. L. Mandoria, "Network Intrusion Detection: A comparative study using state-of-the-art machine learning methods," *IEEE Int. Conf. Issues Challenges Intell. Comput. Tech. ICICT 2019*, pp. 0–4, 2019, doi: 10.1109/ICICT46931.2019.8977679.

[6] H. Thanh and T. Lang, "Use the ensemble methods when detecting DoS attacks in Network Intrusion Detection Systems," *EAI Endorsed Trans. Context. Syst. Appl.*, vol. 6, no. 19, p. 163484, 2019, doi: 10.4108/eai.29-11-2019.163484.

[7] A. Dickson and C. Thomas, *Analysis of UNSW-NB15 Dataset Using Machine Learning Classifiers*, vol. 1366. Springer Singapore, 2021.

[8] D. G. Mogal, S. R. Ghungrad, and B. B. Bhusare, "NIDS using Machine Learning Classifiers on UNSW-NB15 and KDDCUP99 Datasets," *Ijarcce*, vol. 6, no. 4, pp. 533–537, 2017, doi: 10.17148/ijarcce.2017.64102.

[9] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 1561–1573, 2020, doi: 10.1016/j.procs.2020.03.367.

[10] B. Venkatesh and J. Anuradha, "A review of Feature Selection and its methods," *Cybern. Inf. Technol.*, vol. 19, no. 1, pp. 3–26, 2019, doi: 10.2478/CAIT-2019-0001.

[11] I. Sumaiya Thaseen and C. Aswani Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462–472, 2017, doi: 10.1016/j.jksuci.2015.12.004.

[12] D. B. Satmoko, P. Sukarno, and E. M. Jadied, "Peningkatan Akurasi Pendeteksian Serangan DDoS Menggunakan Multiclassifier Ensemble Learning dan Chi-Square," vol. 5, no. 3, pp. 7977–7985, 2018.

[13] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Comput. Secur.*, vol. 103, p. 102158, 2021, doi: 10.1016/j.cose.2020.102158.

[14] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7376 LNAI, pp. 154–168, 2012, doi: 10.1007/978-3-642-31537-4_13.

[15] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," *Augment. Hum. Res.*, vol. 5, no. 1, 2020, doi: 10.1007/s41133-020-00032-0.

[16] B. P. O. Lovatti, M. H. C. Nascimento, Á. C. Neto, E. V. R. Castro, and P. R. Filgueiras, "Use of Random forest in the identification of important variables," *Microchem. J.*, vol. 145, no. December 2018, pp. 1129–1134, 2019, doi: 10.1016/j.microc.2018.12.028.

[17] S. K. Lakshmanaprabu, K. Shankar, M. Ilayaraja, A. W. Nasir, V. Vijayakumar, and N. Chilamkurti, "Random forest for big data classification in the internet of things using optimal features," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 10, pp. 2609–2618, 2019, doi: 10.1007/s13042-018-00916-z.

[18] Z. Noshad *et al.*, "Fault detection in wireless sensor networks through the random forest classifier," *Sensors (Switzerland)*, vol. 19, no. 7, pp. 1–21, 2019, doi: 10.3390/s19071568.