

Peningkatan Performa Pendeteksian Anomali Menggunakan Ensemble Learning dan Feature Selection

Anomaly Detection Performance Improvement Using Ensemble Learning and Feature Selection

Ripto Sudyarno¹, Arief Setyanto², Emha Taufiq Luthfi³

^{1,2,3}Universitas Amikom Yogyakarta

E-mail: *¹ripto.s@students.amikom.ac.id, ²arief_s@amikom.ac.id,
³emhataufiqluthfi@amikom.ac.id

Abstrak

Intrusion detection systems (IDS) atau Sistem pendeteksian intrusi dikenal sebagai teknik yang sangat menonjol dan terkemuka untuk menemukan malicious activities pada jaringan komputer, tidak seperti firewall konvensional, IDS berbeda dalam hal pengidentifikasian serangan secara cerdas dengan pendekatan analitik seperti data mining dan teknik machine learning. Dalam beberapa dekade terakhir, ensemble learning sangat memajukan penelitian pada machine learning dan klasifikasi pola, serta menunjukkan peningkatan hasil kinerja dibandingkan single classifier. Pada Penelitian ini dilakukan percobaan peningkatan nilai akurasi terhadap sistem pendeteksian anomali, pertama dilakukan klasifikasi menggunakan single classifier untuk didapati hasil nilai akurasi yang nantinya dibandingkan dengan hasil dari ensemble learning dan feature selection. Penggunaan ensemble learning bertujuan untuk mendapatkan nilai akurasi yang terbaik dari single classifier. Hasil didapatkan dari nilai confusion matrix dan akan dilakukan pengujian dengan cara membandingkan nilai kedua metode diatas. Penelitian berhasil mendapatkan nilai akurasi single classifier (naïve bayes) yaitu 77,4% dan nilai ensemble learning 96,8%.

Kata Kunci— ensemble learning, nsl-kdd, naïve bayes, anomali, feature selection

Abstract

Intrusion detection systems (IDS) are known as very prominent and leading techniques for finding malicious activities on computer networks, unlike conventional firewalls, IDS differs in terms of identifying attacks intelligently with analytic approaches such as machine learning techniques. In the last few decades, ensemble learning has greatly advanced research in machine learning and pattern classification it has shown an improve in performance results compared to a single classifier. In this study an attempt was made to increase the accuracy of anomalous detection systems, first by classification using a single classifier to find the results of accuracy which will be compared with the results of ensemble learning and feature selection. The use of ensemble learning aims to get the best accuracy value from a single classifier. The results are obtained from the value of the confusion matrix and will be tested by comparing the values of the two methods above. The research succeeded in getting a single classifier accuracy value of 77,4% and ensemble learning 96,8%.

Keywords— ensemble learning, nsl-kdd, naïve bayes, anomali, feature selection

1. PENDAHULUAN

Intrusion detection systems (IDS) atau sistem pendeteksian intrusi dikenal sebagai teknik yang sangat menonjol dan terkemuka untuk menemukan *malicious activities* pada jaringan komputer. IDS berbeda dalam hal pengidentifikasian serangan secara cerdas dengan pendekatan analitik seperti data mining dan teknik machine learning, tidak seperti firewall konvensional [1].

Pada penelitian ini digunakan *ensemble learning* dan *feature selection* untuk meningkatkan *accuracy score* dan *process time*. Alasan kuat penggunaan *ensemble learning* ini adalah penggunaan *single algorithm* mungkin tidak cukup kuat untuk menghasilkan IDS yang lebih kuat dari *ensemble learning* [2][3]. *Ensemble learning* adalah algoritma yang menggunakan beberapa klasifikasi kemudian membuat titik data baru dengan cara mengambil bobot prediksi [4]. Tujuan utama dari *ensemble learning* adalah untuk menentukan atau menemukan set terbaik berdasarkan bobot datanya untuk dipisah menjadi 1 set data baru[5].

Dalam beberapa dekade terakhir, *ensemble learning* sangat memajukan penelitian pada *machine learning* dan klasifikasi pola, serta menunjukkan peningkatan hasil kinerja dibandingkan *single classifier* [6]. Ada faktor lain yang perlu dipertimbangkan juga seperti *feature selection*. *Feature selection* sangat penting pada *preprocessing* karna memungkinkan untuk menghilangkan fitur yang tidak relevan pada *dataset* untuk mengurangi *process time* dan juga meningkatkan *accuracy rate* [7]. (Yinhui Li 2012) mengusulkan untuk teknik untuk menaikkan kualitas *feature selection* yang disebut Metode GFR, memilih 19 dari 41 fitur pada NSL-KDD *dataset* [8].

Pada penelitian ini akan melakukan peningkatan terhadap anomali deteksi menggunakan *feature selection* dimana dapat mengurangi redundant data dan mengurangi waktu proses klafikasi serta peningkatan nilai akurasi. Dan juga penerapan *ensemble learning* untuk dibandingkan dengan hasil dari *single classifier*, pada penelitian ini digunakan algoritma *Naïve bayes*, algoritma SVM dan *Random forest* sebagai ensemble.

Hasil akhir dari penelitian ini berhasil meningkatkan nilai akurasi dan mengurasi proses time. Nilai akurasi berhasil dinaikan untuk setiap nilai *attack class* diantara ~1-12%. Terlihat jelas perbedaan nilai akurasi antara *single classifier* dan *ensemble learning*.

2. METODE PENELITIAN

2.1. Tinjauan Pustaka

Pada fitur hibrida yang terdiri dari 3 metode yaitu algortima PSO, algortima genetic, dan algoritma colony, digunakan untuk mengurangi ukuran dari fitur dataset training pada NSL-KDD dan UNSW-NB15.

Fitur yang dipilih berdasarkan kinerja dari klasifikasi reduced error pruning tree (REPT) classifier. Pada NSL-KDD didapati hasil 85.8% untuk *accuracy score* [9].

Didapati hasil klasifikasi pada SVM linear, SVM RBF, Random forest, dan ELM pada pembagian dataset 80:20, ELM menghasilkan nilai yang bagus diantara semua classifier, sementara SVM RBF terdapat peningkatan akurasi diantara RF dan ELM pada pembagian setengah sampel. Sementara diketahui ELM sangat cocok digunakan untuk IDS dengan jumlah data yang besar tetapi belum didapati hasil yang sama untuk fitur yang diseleksi dan transformasi fitur [10].

Dengan menggunakan *feature selection* untuk mengurangi jumlah dataset yang digunakan serta meningkatkan akurasi dari model. Penelitian ini menggunakan metode *feature selection* yang menghasilkan nilai akurasi yang tinggi dimana fitur dihasilkan dari informasi yang didapat dan pemeringkatan terhadap fitur. Sebagai contoh nilai akurasi dari R2L berhasil ditingkatkan dari 97.03% menjadi 99.88% [11].

Algoritma pemilihan fitur berdasarkan informasi yang sama dapat menangani lienar atau non-linear data. Dikarenakan fitur yang terlalu banyak memperlambat prooses klasifikasi dan juga

dapat mengurangi nilai akurasi dikarenakan data berulang atau fitur yang tidak relevan. Dan terbukti feature selection dapat menaikkan accuracy dan mengurangi waktu proses [12].

2.2. NSL-KDD Dataset

Pada penelitian ini digunakan NSL-KDD Dataset yang merupakan versi terbaik dari KDD '99 dimana terdapat banyak duplikasi data pada data train dan test nya. Oleh karena itu digunakannya NSL-KDD untuk membangun model yang memiliki akurasi yang kuat [13]. Terdapat 41 Feature dimana diantaranya ada 3 nominal data, 6 binary data, dan 32 lainnya data kontinu, dan setiap data memiliki label sebagai jenis serangan atau normal.

Table 1. Tipe dan nama fitur NSL-KDD

Type	Features
Caregorical	Protocol_type(2), Service(3), Flag(4)
Binary	Land(7), logged_in(12),root_shell(14), su_attempted(15),is_host_login(21),, is_guest_login(22)
Numeric	Duration(1),src_bytes(5),dst_bytes(6), wrong_fragment(8),urgent(9), hot(10),num_failed_logins(11),num_compromised(13),num_root(16),num_file_creations(17),num_shells(18),num_access_files(19),num_outbound_cmds(20), count(23),srv_count(24), serror_rate(25),srv_serror_rate(26), error_rate(27),srv_error_rate(28), same_srv_rate(29),diff_srv_rate(30),srv_diff_host_rate(31),dst_host_count(32),dst_host_srv_count(33),dst_host_same_srv_rate(34),dst_host_diff_srv_rate(35),dst_host_same_src_port_rate(36),dst_host_srv_diff_host_rate(37),dst_host_serror_rate(38),dst_host_srv_serror_rate(39),dst_host_error_rate(40),dst_host_srv_error_rate(41)

2.3. Preprocessing

1. Data Cleaning

Proses ini dilakukan untuk menjaga kualitas dataset yang akan digunakan untuk menjaga hasil dari pengklasifian. Ada beberapa fungsi dari data cleaning sebagai berikut:

- Memperbaiki struktur data yang tidak konsisten.
- Mengurangi data redundant.
- Menghilangkan outliers serta memperhalus noise pada data.
- Melengkapi missing value atau data hilang pada data test.

2. Data transformation and Data ScallingNaïve Bayes Classifier

Untuk mempermudah proses klasifikasi maka dari itu dilakukan perubahan bentuk data dari categorical kedalam bentuk numeric seperti pada data protocol type, Service, dan Flag. Setelahnya, dilakukan proses scaling data yang merupakan proses penting dikarenakan banyak algoritma machine learning yang akan bekerja tidak maksimal (bahkan salah), apabila variabel-variabel feature memiliki skala yang berbeda. Sebagai contoh, support vector machine (SVM) akan berperfora buruk jika datanya tidak melalui proses scaling terlebih dahulu, karena SVM menggunakan fungsi jarak (distance function) untuk optimasinya.

2.4. Naïve Bayes Classifier

Naïve Bayes Classifier merupakan algoritma yang digunakan untuk prediksi yang berbasikan pada probabilitas sederhana yang berdasarkan pada teorema bayes. Naïve bayes menggunakan model fitur independent [14]. Rumus dasar dari algoritma naïve bayes seperti pada persamaan 1.

$$P(y|X) = \frac{P(X|y).P(y)}{P(X)} \quad (1)$$

$P(y|X)$ merupakan nilai probabilitas dari *conditional probability* yang merupakan suatu hipotesa y terjadi ketika adanya evidence X terjadi. $P(X|y)$ adalah nilai probabilitas dari evidence X terjadi yang akan berdampak pada hipotesa y . Untuk $P(y)$ merupakan nilai Probabilitas awal dari hipotesa y yang tidak terpengaruh dengan evidence apapun. Sedangkan $P(X)$ adalah nilai Probabilitas awal evidence X yang tidak terpengaruh dengan hipotesa yang lain.

2.5. Support Vector Machine

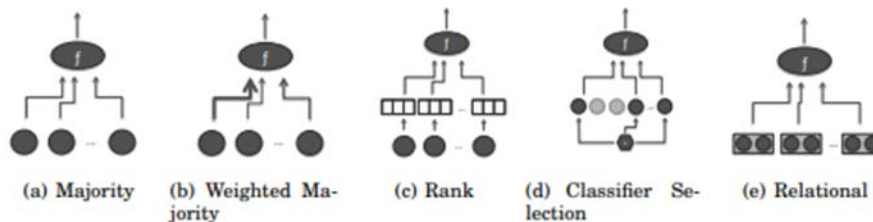
Support Vector Machine (SVM) adalah sebuah algoritma untuk melakukan prediksi secara klasifikasi ataupun regresi yang efektif. SVM diusulkan oleh Vapnik (1995) yang merupakan teknik supervised learning yang dilatih untuk mengklasifikasikan kategori yang berbeda dari banyak macam kasus. SVM menciptakan hyperplane dalam ruang dimensi yang tinggi, dimana pemisah terbaik yaitu yang bisa membagi data kedalam beberapa class dengan pemisah yang besar. Dan juga fungsi kernel berfungsi untuk memperkirakan nilai margin dan memaksimalkan hyperplane.

2.6. Random Forest

Random Forest merupakan algoritma yang digunakan untuk pengklasifian dan analisis regresi, membuat pohon keputusan pada tahap pelatihan, output class merupakan hasil dari suara mayoritas [15]. Random forest dapat mencapai akurasi yang tinggi karena dapat menangani overfitting.

2.7. Ensemble learning

Ensemble learning adalah sebuah komposisi dari beberapa learner atau classifier yang lemah dengan sebuah learner yang memiliki hasil prediksi yang bagus. Penelitian ini menerapkan parameter atau teknik Voting. Voting adalah teknik yang berhubungan dengan bagaimana output masing-masing dari anggota ensemble digunakan pada saat prediksi, dimana struktur sangat mempengaruhi dari voting. Berikut beberapa macam voting method pada Gambar 1 [16].



Gambar 1. Macam-macam metode Voting

2.8. Confusion matrix

Pada penelitian ini pengujian terhadap hasil klasifikasi akan dilanjutkan dengan pencarian nilai accuracy rate, recall, precision, dan f-measure.

1. Accuracy, merupakan *ratio* prediksi yang benar (positif atau negatif) terhadap nilai dari keseluruhan data dengan rumus seperti pada persamaan 2.

$$\frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

2. Recall, merupakan *ratio* prediksi yang benar positif dibandingkan dengan data yang benar positif dengan rumus seperti pada persamaan 3:

$$\frac{TP}{TP+FN} \quad (3)$$

3. Precision, merupakan *ratio* prediksi benar positif dibandingkan dengan kesemua hasil positif yang telah diprediksi sebelumnya dengan rumus seperti pada persamaan 4.

$$\frac{TP}{TP+FP} \tag{4}$$

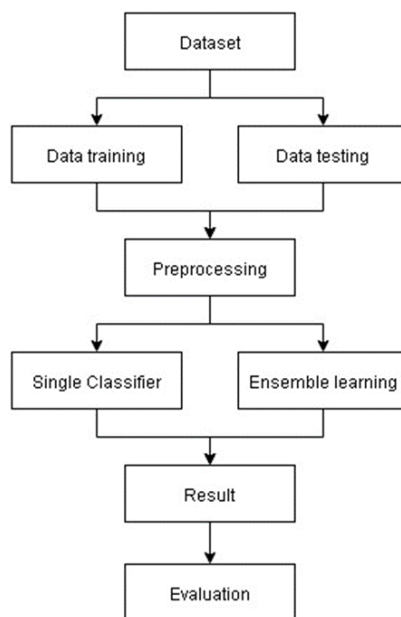
4. F-Measure, merupakan perbandingan antara nilai rata-rata *precision* dan *recall* yang telah dibobotkan dengan rumus seperti pada persamaan 5.

$$\frac{2*Recall*Precision}{Recall+Precision} \tag{5}$$

TP adalah True Positive yang merupakan akumulasi nilai positif yang terklasifikasi dengan benar oleh sistem. Untuk TN adalah True Negative, merupakan akumulasi nilai negatif yang terklasifikasi dengan benar oleh sistem. FP adalah False Positive yang merupakan akumulasi nilai positif namun terklasifikasi salah oleh system. Sementara, FN adalah False Negative yang merupakan akumulasi nilai negatif namun terklasifikasi salah oleh sistem.

2.9. Alur penelitian

Terdapat beberapa langkah dalam penelitian ini untuk mempermudah dalam pengambilan Tindakan, seperti ditunjukkan pada Gambar 1.



Gambar 1. Diagram alur penelitian

3. HASIL DAN PEMBAHASAN

3.1. Implementasi Proprocessing

1. Data Cleaning

Sebelum melakukan proses data cleaning terdapat beberapa kesenjangan data atau data missing pada dataset yang digunakan seperti yang ditunjukkan Gambar 2, dan terdapat perbedaan jumlah *categories* pada data training dan testing yaitu selisih 6 buah *categories* pada fitur *service*.

```

Training set:
Feature 'protocol_type' has 3 categories
Feature 'service' has 70 categories
Feature 'flag' has 11 categories
Feature 'label' has 23 categories

Test set:
Feature 'protocol_type' has 3 categories
Feature 'service' has 64 categories
Feature 'flag' has 11 categories
Feature 'label' has 38 categories

```

Gambar 2. Jumlah *categories* pada data train dan test

Proses selanjutnya setelah didapati *categories* yang hilang dan dilanjutkan dengan menyelaraskan dimensi data, seperti yang ditunjukkan pada potongan kode berikut.

```

for col in difference:
testdf_cat_data[col] = 0
print(df_at_data.shape)
print(testdf_cat_data.shape)

```

2. Data Transformation dan Data Scalling

Proses selanjutnya dilakukan tranformasi categorical data menjadi numerical seperti ditunjukkan pada Tabel 2.

Table 2. Tranformasi data

NO	protocol_type	service	flag	Numerical protocol_type	Numerical service	Numerical Flag
0	tcp	ftp_data	SF	1	20	9
1	udp	other	SF	2	44	9
2	tcp	private	S0	1	49	5
3	tcp	http	SF	1	24	9
4	tcp	http	SF	1	24	9

3. Feature Selection

Proses ini akan memilih 13 fitur menggunakan RFE (Recursive Feature Selection) dengan Random Forest Classifier sebagai estimator yang nantinya akan menghasilkan data baru yang digunakan pada proses klasifikasi seperti pada potongan kode berikut:

```

RFC_CLF = RandomForestClassifier(n_estimators=10,n_jobs=2,max_depth = 5)
RFE_FS = RFE(estimator=RFC_CLF, n_features_to_select=13, step=1)

```

4. Naïve Bayes Classifier

Proses klasifikasi dibangun dengan Naïve bayes seperti pada potongan kode berikut:

```

CLF_NB = GaussianNB(priors=None, var_smoothing=1e-09)

```

5. Ensemble Learning

Proses pengklasifikasin dengan metode ini dibangun menggunakan 2 classifier yaitu SVM kernel RBF dan Random Forest seperti pada potongan kode berikut:

```

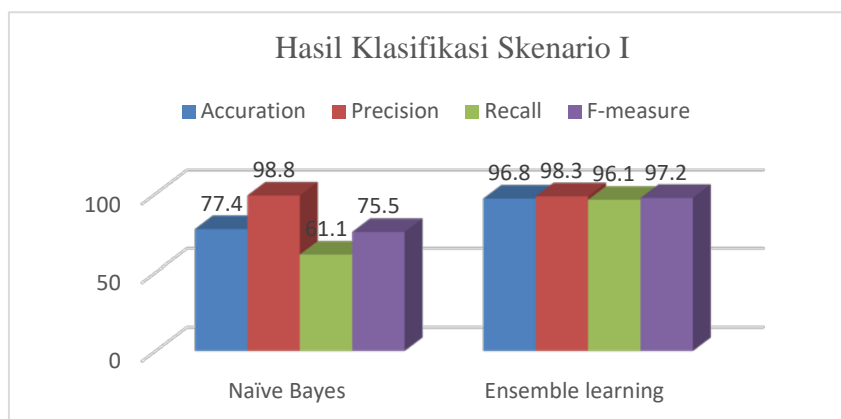
CLF_RF = RandomForestClassifier(n_estimators=10,n_jobs=2)
CLF_SVM = SVC(kernel='rbf',random_state=None, gamma=0.1 , C=10)
CLF_ELE = VotingClassifier(estimators=[('rf', CLF_RF), ('svm',
CLF_SVM)], voting='hard')

```

3.2. Hasil Ujicoba

Setelah dilakukan proses *preprocessing*, selanjutnya dilakukan proses klasifikasi menggunakan algoritma *naïve bayes* dan *ensemble learning*. Dari hasil klasifikasi dilanjutkan dengan proses validasi serta didapatkan nilai akurasi.

1. Skenario I (Klasifikasi tanpa *Feature selection*)



Gambar 3. Hasil Skenario I

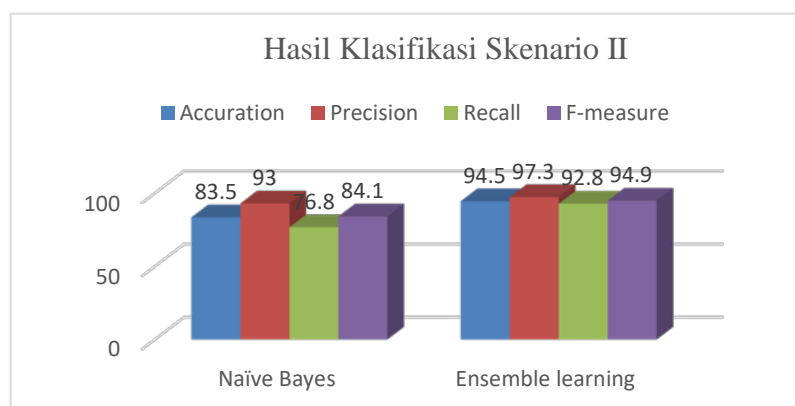
Dari Gambar 3, pada *naïve bayes classifier* menghasilkan nilai akurasi 77,4%, presisi 98,8%, *recall* 61,1%, dan F-1 75,5%. Adapun *ensemble learning* menghasilkan nilai akurasi yang jauh lebih baik dari *naïve bayes* yaitu 96,8% serta nilai presisi 98%, *recall* 96,1% dan F-1 97,2%. Tabel 3 menampilkan perbandingan waktu proses pada kedua pengklasifikasian.

Tabel 3. Perbandingan waktu proses

No	Classifier	Waktu Proses
1	<i>naïve bayes</i>	0.140 detik
2	<i>ensemble learning</i>	15.342 detik

Pada Tabel 3 diketahui, *ensemble learning* memiliki waktu proses lebih lama dari *naïve bayes* yaitu 15.342 detik dan 0.140 detik, hal ini didasari oleh *ensemble learning* menggunakan 2 buah *classifier* yaitu algoritma *SVM kernel RBF* dan *Random Forest* sehingga dibutuhkan waktu lebih untuk melakukan proses klasifikasi.

2. Skenario II (Klasifikasi + *Feature selection*)



Gambar 3. Skenario II

Dari Gambar 3, *Naïve Bayes Classifier* menghasilkan nilai akurasi 83,5%, presisi 93%, *recall* 76,8%, dan F-1 84,1%. Adapun *Ensemble Learning* menghasilkan nilai akurasi yang jauh lebih baik dari *Naïve Bayes* yaitu 94,5% serta nilai presisi 97,3%, *recall* 92,8% dan F-1 94,9%. Berikut pada Tabel 4 menampilkan perbandingan waktu proses pada kedua pengklasifikasian.

Pada Tabel 5 diketahui, *Ensemble Learning* memiliki waktu proses lebih lama dari *Naïve Bayes* yaitu 5,234 detik dan 0,0195 detik.

Tabel 4. Perbandingan waktu proses

No	Classifier	Waktu Proses
1	<i>naïve bayes</i>	0,0195 detik
2	<i>ensemble learning</i>	5,234detik

3. Perbandingan dengan penelitian sejenis (Benchmark)

Table 5. Perbandingan hasil dengan beberapa penelitian sejenis

Metode	Tahun	Feature Selection	Accuracy	Presicion	Statistik Waktu
<i>Naïve bayes</i>	2019	RFE	83,5	93,0	0,0195 detik
<i>Ensemble learning</i>	2019	RFE	94,5	97,3	5,234 detik
Proposed (A Two-Stage Classifier Ensemble) [9]	2019	Hybrid	85,979	88,0	-
SVM [17]	2019	MBGWO	81,58	-	-
Bagging J48 [18]	2018	Gain Ratio	84,25	-	-
2 Tier Classifier [19]	2017	LDA	83,240	-	-

Hasil terbaik ditandai dengan teks cetak tebal, dan hasil dari penelitian ini ditandai dengan teks cetak miring. Hasil analisis perbandingan dari beberapa penelitian dapat dilihat pada Tabel 5 yaitu metode yang diusulkan *Ensemble learning* dan *feature selection* sangat bersaing dalam pendekatan yang efektif untuk kasus pendeteksian intursi anomali. Selain itu hasil *accuracy* yang didapat sangat tinggi jika dibandingkan dengan beberapa penelitian di atas yaitu 94,5% serta nilai *presicion* 97,3%. Penelitian ini mengusulkan performa waktu eksekusi (statistik waktu) dengan hasil bahwa penggunaan *feature selection* pada pengklasifikasian dapat mengurangi waktu proses yang sangat signifikan. Pada beberapa penelitian yang dibandingkan tidak memuat waktu proses.

4. KESIMPULAN

Didapati hasil dari penggunaan ensemble learning berhasil meningkatkan nilai akurasi dan juga penggunaan feature selection sangat membantu memilih feature baru yang penting serta mengurangi waktu proses dibandingkan penggunaan semua feature. Kekurangan implementasi IDS menggunakan ensemble learning terdapat pada waktu proses yang cukup lama sehingga sangat menghambat proses pendeteksian anomali secara real-time. Hal ini terbukti setelah dilakukan 2 skenario di atas, dapat diambil kesimpulan bahwa kedua metode yang digunakan yaitu ensemble learning dan features selection mampu melakukan peningkatan pada proses klasifikasi terutama pada dataset NSL-KDD.

5. SARAN

Untuk penelitian selanjutnya pemilihan fitur penting diharapkan bisa lebih dikembangkan sehingga didapati hasil klasifikasi yang lebih maksimal. Tidak terlepas dari peningkatan nilai akurasi juga sebaiknya waktu proses yang saat ini lebih lama karena penggunaan ensemble learning masih belum mendukung sistem penteksian yang real-time. Diharapkan kedepannya ada pengembangan anomali yang dapat dideteksi sehingga cakupan deteksi IDS yang dibangun lebih luas.

DAFTAR PUSTAKA

- [1] B. A. Tama and K.-H. Rhee, 2017, An extensive empirical evaluation of classifier ensembles for intrusion detection task, *Comput. Syst. Sci. Eng.*, Vol. 32, No. 2, Hal. 149–158,
- [2] Hansen, L. K., Salamon, P., 1990, Neural network ensembles, *IEEE transactions on pattern analysis and machine intelligence* 12, Vol. 10, Hal. 993–1001.
- [3] Schapire, R. E., 1990, The strength of weak learnability *Machine learning*, Vol. 5, No. 2, Hal. 197–227.
- [4] Dietterich, T. G., 2000, Ensemble Methods in Machine Learning - In: *Multiple Classifier Systems MCS 2000, Lecture Notes in Computer Science*, vol 1857, Springer, Berlin, Heidelberg
- [5] Ho, T. K., 2002, Multiple classifier combination: Lessons and next steps. In *Hybrid methods in pattern recognition*, World Scientific, Hal. 171–198.
- [6] Woźniak, M., Graña, M., Corchado, E., 2014, A survey of multiple classifier systems as hybrid systems, *Information Fusion*, vol. 16, pp. 3–17, Mar. 2014.
- [7] Kuncheva, L. I., 2014, *Combining pattern classifiers: methods and algorithms* 2nd Ed, Wiley, New Jersey
- [8] Shrivasa, A. K., Dewangan, A. K., 2014, An Ensemble Model for Classification of Attacks with Feature Selection based on KDD99 and NSL- KDD Data Set, *International Journal of Computer Applications*, No. 15, Vol. 99, Hal. 0975-8887
- [9] Tama, B. A., Comuzzi, M., Rhee, K. H., 2019, TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System, *IEEE Access*, Vol. 7, Hal. 94497–94507.
- [10] Ahmad, I., Basher, M., Iqbal, M., Rahim, A., 2018, Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection, *IEEE Access*, Vol. 6, Hal. 33789–33795.
- [11] Nkiama, H., Said, S. Z. M., Saidu, M., 2016, A Subset Feature Elimination Mechanism for Intrusion Detection System, *International Journal of Advanced Computer Science and Applications*, No. 4, Vol. 7, Hal. 148–157
- [12] M. A. Ambusaidi, X. He, S. Member, P. Nanda, S. Member, and Z. Tan, 2016, Building an intrusion detection system using a filter-based feature selection algorithm, Vol. 9340, Hal. 1–13
- [13] Tavallae, M., Bagheri, E., Lu, W., Ghorbani, A. A., 2009, *IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE Symposium on Computational Intelligence for Security and Defense Applications - CISDA 2009*, Hal. 1–6
- [14] Vembandasamy, K., Sasipriya, R., Deepa, E., 2015, Heart Diseases Detection Using Naive Bayes Algorithm, *International Journal of Innovative Science Engineering and Technology (IJSET)*, No. 9, Vol. 2, Hal. 441–444
- [15] Liu, Y., Wang, Y., Zhang, J., 2012, New machine learning algorithm: Random forest, *International Conference on Information Computing and Applications in Information Computing and Applications*, Hal. 246–252.
- [16] Gomes, H. M., Barddal, J. P., Enembreck, A. F., Bifet, A., 2017, A survey on ensemble learning for data stream classification, *ACM Computing Surveys*, No. 2, Vol. 50
-