JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

PENGARUH PENYETELAN HYPERPARAMETER TERHADAP KINERJA PREDIKSI *RANDOM FOREST* PADA PENDETEKSIAN *SPAM*

NURSALMAN[1] , MUSTIKASARI[2]

Prodi Teknik Informatika, STMIK Dipanegara Makassar[1],
Prodi Teknik Informatika, Universitas Islam Negeri Alauddin Makassar[2],
E-mail: nursalman.halim@dipanegara.ac.id[1], mustikasari@uin-alauddin.ac.id[2]

ABSTRAK

Random Forest memiliki versi modifikasi dan sejumlah hyperparameter yang terpasang "default" pada aplikasi. Penelitian terdahulu telah membahas bahwa penyetelan hyperparameter dapat berpengaruh terhadap kinerja sistem. Namun, penyetelan hyperparameter secara manual bukan pekerjaan yang sederhana untuk sebuah algoritma yang kompleks. Mempertimbangkan bahwa banyak dataset keamanan data yang berdimensi tinggi, maka otomatisasi dan efisiensi menjadi pertimbangan dalam pekerjaan ini. Penyetelan hyperparameter bertujuan membentuk kombinasi hyperparameter yang dapat meningkatkan kinerja prediksi. Penelitian ini memuat perbandingan kinerja evaluasi hasil prediksi dengan penyetelan hyperparameter. Hasil yang diperoleh pada pendeteksian *spam* dengan 3-*fold* dan 5-*fold-cross-validation, variabel* menunjukkan bahwa kinerja prediksi meningkat menjadi 95% dan 95.4% pada *Accuracy* dan mencapai 98.5% dan 98.7% pada AUC, sementara ukuran tingkat kesalahan menurun hingga .50 dan .46 untuk MMCE dan .75 dan .80 untuk *Brier Score*.

Kata Kunci — penyetelan, hyperparameter, kinerja, prediksi.

I. PENDAHULUAN

Klasifikasi adalah salah satu metode data mining yang cukup populer diterapkan di berbagai bidang. Metode klasifikasi ini membagi sampel data sesuai kelas target. Model kelas kemudian melakukan prediksi kelas target / tujuan untuk setiap titik data. Pendekatan klasifikasi pada data *spam* yang dilakukan dalam penelitian ini dilakukan dengan menganalisa pola data *spam* dan bukan *spam*. Selanjutnya, model ini dapat digunakan untuk kasus prediksi data baru ke dalam kelas-kelas yang ada, penyetelan hyperparameter dilakukan untuk mengoptimasi hasil prediksi dan pekerjaan dilanjutkan pada tahap analisa pengaruh penyetelan hyperparameter pada kinerja prediksi menggunakan ukuran evaluasi. Beberapa metode klasifikasi tertentu dibutuhkan untuk jenis data yang besar dengan banyak attribut seperti yang umumnya ditemukan pada bidang keamanan data, semisal data

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

spam. Oleh karena itu, beberapa metode sering lebih dipilih untuk diterapkan karena sesuai dan mampu menangani data tertentu dengan baik.

Beberapa ringkasan dari beberapa pekerjaan terdahulu yang berfokus pada bidang keamanan data adalah penelitian oleh Yuan L. Antonius dan Rachmat, C, (2017), yang mendeteksi *spam* pada komentar berbahasa Indonesia di Instagram, menggunakan pendekatan pembelajaran mesin, pada penelitian ini pendeteksian bekerja dengan pendekatan pembelajaran terawasi yaitu *Naïve Bayes*. Sementara, fokus masalah penelitian yang kami lakukan tentang penyetelan parameter dan pengaruhnya pada prediksi dari metode terpilih yang digunakan.

Penelitian lainnya adalah yang dilakukan oleh Iftikhar Ahmad, dkk (2018), yang menggunakan pendekatan *Support Vector Machine*, *Random Forest*, and *Extreme Learning* untuk model klasifikasi *intrusion* atau gangguan, melalui penelitian yang bertujuan untuk meningkatkan kinerja klasifikasi untuk masalah terkait dataset. Ketiga algoritma dibandingkan berdasarkan kemampuan klasifikasinya. Pekerjaan validasi hasil prediksi menggunakan metode "*split*" data dengan porsi 80 : 20 dan 90 : 10 untuk selanjutnya dijadikan sebagai data latih dan data uji. Data diuji dalam tiga skenario proporsi data, yaitu *full samples*, *half* (1/2 *samples*) dan *quarter* (1/4 *samples*). Bagaimanapun, pengaturan skenario seperti ini masih memerlukan kajian mendalam tentang dataset yang digunakan. Pekerjaan penelitian kami berbeda dari strategi pencapaian kinerja prediksi, kriteria pengujian dan teknik validasi yang digunakan.

Penelitian yang paling relevan telah dilakukan oleh Wicaksono, A. S., dan Supianto, A. A. (2018), yang berfokus pada masalah optimalisasi hyperparameter. dengan teknik *resampling.* Algoritma Genetika digunakan untuk mengoptimasi hyperparameter. Selanjutnya hyperparameter yang dihasilkan menjadi hyperparameter acuan pada algoritma pengklasifikasi data. Penelitian ini mampu mencapai nilai optimasi hyperparameter yang cukup signifikan dari metode grid search sebagai pembanding. Meskipun memiliki fokus yang sama pada optimalisasi hyperparameter, akan tetapi, metode dan pendekatan yang kami gunakan berbeda.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Penyetelan hyperparameter pada penelitian kami adalah untuk melihat hubungan atau pengaruh penyetelan pada setiap variabel hyperparameter dengan kinerja prediksi. Motivasi yang melatarbelakangi adalah bahwa hasil penyetelan hyperparameter dapat berpengaruh secara berbeda pada ukuran kinerja yang berbeda.

Berdasarkan literature review dan eksperimen pada penelitian ini, kami melakukan penelitian untuk pendeteksian *spam* dengan mempelajari pengaruh penyetelan hyperparameter pada kinerja algoritma *random forest* secara empiris dan menentukan variabel mana dari hyperparameter yang cenderung menghasilkan kinerja prediksi yang lebih baik pada beberapa ukuran kinerja evaluasi.

II.METODE PENELITIAN

Metode yang dilakukan dalam pekerjaan ini merupakan metode eksperimental. Metode ini merupakan salah satu metode kuantitatif dengan melakukan percobaan untuk melihat hasil dan menyelidiki kemungkinan sebab akibat.

a) Data

Data *spam* yang digunakan pada penelitian ini adalah data sekunder, yang diambil dari UCI Machine Learning Repository. Data *spam* yang digunakan adalah dataset *spambase* (D. Dua, C. Graff, 2019). Dataset *spambase* terdiri atas 4601 data dan 58 attribut.

b) Random Forest

Algoritma *random forest* (Breiman, 2001) dibangun dengan menggantungkan konstruksinya pada data, yang menjadikannya sebuah konstruksi dengan tingkat analisis matematika yang rumit. Sangat mendukung pekerjaan klasifikasi, regresi, dan *survival forests*. Algoritma ini digunakan untuk strategi prediksi berbasis model, dibangun berdasarkan kasus pengklasifikasian dan mengacu pada data sebelumnya, dimana telah diperoleh model penyaringan dan attribut acuan yang menjadi dasar pengklasifikasiannya.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Mempertimbangkan observasi ke-i dari set data pengujian $(i = 1,..,n)$, disini kami menyatakan label kelas sebagai y_i, yang berupa label biner 0 vs. 1. Nilai prediksi dihasilkan oleh pohon t (dengan $t = 1,..,T$) dilambangkan sebagai $(\hat{y}_{it})$, sedangkan $\hat{y}_i$ adalah singkatan dari nilai prediksi keluaran oleh keseluruhan *random forest*. Dalam kasus klasifikasi, $\hat{y}_i$ biasanya diperoleh dengan suara terbanyak. Untuk klasifikasi biner, ini setara dengan menghitung rata-rata yang sama seperti untuk regresi, yang menggunakan

$$\hat{p}_i = \frac{1}{T}\sum_{i=0}^{n} I\ (\hat{y}_{it} = 1)$$

dan dilambangkan sebagai $\hat{p}_i$ (singkatan dari probabilitas), dan akhirnya menurunkan $\hat{y}_i$ sebagai

$$\hat{y}_i = \begin{cases} 1 & if\ \hat{p}_i > 0.5, \\ 0 & otherwise \end{cases}$$

Random forest (RF) adalah teknik pembelajaran ensemble yang terdiri dari agregasi sejumlah besar *T* = pohon keputusan. Prediksi diperoleh untuk observasi baru dengan menggabungkan prediksi yang dibuat oleh *T* pohon tunggal. Penelitian ini mengimplementasi teknik *Random Forest Generator* (Marvin N. Wright, 2017), yang disingkat "*ranger"* disediakan pada perangkat lunak open source yang dirilis di bawah lisensi GNU GPL-3, merupakan algoritma modifikasi dari algoritma aslinya yaitu *random forest*. Algoritma modifikasi *random forest* ini sering diterapkan untuk klasifikasi data berdimensi tinggi dengan kebutuhan efesiensi memori yang ketat.

c) Penyetelan Hyperparameter

Random forest bekerja dengan sejumlah hyperparameter. Untuk mendapatkan trade-off yang sesuai, kita dapat mengacu pada beberapa variable berikut ini.

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Tabel 1: Variabel hyperparameter *random forest* dan nilai standar / *default*

Hyperparameter	Nilai Standar / *default*
num_trees → Jumlah dari pohon di hutan	($\sqrt{p}$ dimana p adalah jumlah variabel prediktor)
mtry → Jumlah pohon untuk dibangun menjadi pohon keputusan	(500, 1000)
min.node.size → Jumlah minimum observasi di terminal node	1 (satu) untuk klasifikasi
splitting.rule → Memisahkan kriteria di node	(Gini, p-value, random)
sample.size → Jumlah observasi yang diambil untuk setiap pohon	(n)
replacement → Pengamatan dengan atau tanpa penggantian	("FALSE" = tanpa penggantian)

Setiap hyperparameter memiliki keutamaannya masing-masing dan akan mempengaruhi output dari prediksi. Jika kinerja *random forest* dengan nilai hyperparameter standar dapat ditingkatkan dengan memilih alternative nilai lainnya, pertanyaan selanjutnya adalah bagaimana pilihan ini harus dilakukan.

Berikut ini beberapa uraian mengenai masing-masing hyperparameter dari literatur dan eksperimen yang dapat membantu menjadi acuan penyetelan hyperparameter. Hyperparameter pertama adalah *num.trees* atau jumlah pohon di hutan, Peningkatan jumlah pohon secara linier meningkatkan akurasi model. Semakin luas ukuran *forest* semakin baik keakuratannya.

Hyperparameter kedua adalah *mtry*. Nilai *mtry* yang lebih rendah menghasilkan pohon yang berbeda dengan korelasi yang rendah, dan dapat menghasilkan stabilitas yang lebih baik saat digabungkan. Namun nilai yang rendah tersebut dapat memicu pemilihan kandidat kecil secara acak yang merupakan variabel sub optimal sehingga menyebabkan pohon berkinerja lebih buruk. Hyperparameter ketiga yaitu *min.node.size*. Meskipun dibeberapa aplikasi, nilai *min.node.size* sebagai jumlah minimum observasi di terminal node sering di atur ke = 1 untuk klasifikasi, namun berdasarkan beberapa simulasi, lebih tepat untuk menyetel variabel ini ke nilai yang lebih tinggi dari nilai default karena hal ini mengurangi waktu proses secara empiris.

Secara umum kami belum menemukan ada aturan pemisahan atau *splitting.rule* yang dapat dibuktikan lebih unggul dari pada lainnya terkait dengan

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

kinerja. Metode ini mendukung pemilihan variabel dengan banyak kemungkinan pemisahan (misalnya, variabel kontinu atau variabel kategori dengan banyak kategori) dan variabel.

Pengambilan ukuran sampel atau *sample.size* dengan penggantian atau *replacement* tidak terlalu kami soroti, karena disini kami merujuk pada pendapat Martínez-Muñoz dan Suárez (2010) yang telah meneliti bahwa pengambilan sampel dengan penggantian atau tanpa penggantian ketika parameter ukuran sampel disetel secara optimal tidak akan menunjukkan perbedaan kinerja yang substansial.

Penyetelan hyperparameter pada penelitian ini adalah menggunakan teknik *sequential model-based optimization* (SMBO) yang telah tersedia dalam library tuneRanger di R (Bischl et al., 2017). Sementara teknik penyetelan konfigurasi hyperparameter prediksi secara otomatis yang diperoleh dari hasil evaluasi hyperparameter menerapkan dan mengevaluasi model pada "tuneRanger". Lebih lanjut tentang pengaruh beberapa hyperparameter akan dibahas di bagian pembahasan.

d) Strategi dan Ukuran Evaluasi Kinerja

Strategi umum untuk mengevaluasi kinerja algoritma dengan nilai hyperparameter berbeda adalah *k-fold cross-validation*. Jumlah *k* lipatan biasanya dipilih antara 2 hingga 10. Rata-rata hasil dari beberapa pengulangan dari seluruh prosedur validasi silang memberikan hasil yang lebih dapat diandalkan karena variansi dari estimasi berkurang (Seibold et al., 2017). Untuk random forest, strategi lain dimungkinkan, yaitu menggunakan observasi Out-Of-Bag (OOB) untuk mengevaluasi algoritma yang terlatih. Secara umum, hasil dari strategi ini dapat diandalkan, namun bias akan teramati pada situasi khusus (Janitza, 2017). Oleh karena itu, kami lebih memilih menggunakan validasi silang *k-fold* pada studi ini dengan penggunaan $k = 3$ dan $k = 5$ dan masing-masing diulang sebanyak k.

Ukuran evaluasi adalah ukuran yang bergantung pada masalah pembelajaran. Dalam klasifikasi, dua ukuran yang paling sering dipertimbangkan

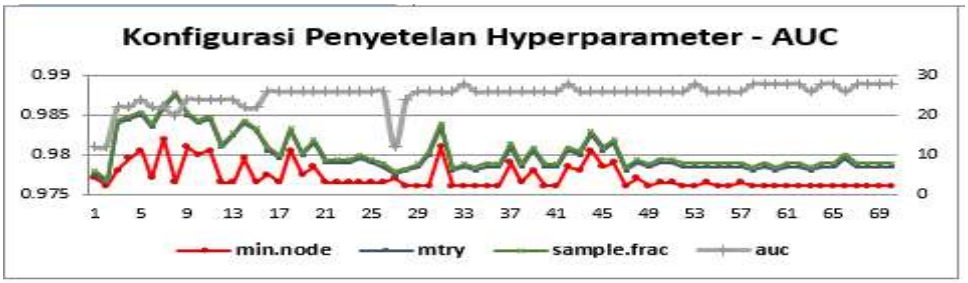
JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

adalah "classification rate" yang mana kami menggunakan "Accuracy" dan ukuran Area Under the Curve (AUC). Dua ukuran umum lainnya yang didasarkan pada probabilitas adalah "Brier Score" dan "mean misclassification error" atau MMCE untuk alasan evaluasi kinerja prediksi klasifikasi biner dan evaluasi kesalahan pengelompokan kelas pada model.

III.HASIL DAN PEMBAHASAN

Pada bagian ini, kami menjalankan algoritma random forest generator, dan D = spambase dataset, yang digunakan sebagai input. Hasil perbandingan klasifikasi prediksi untuk 4 variabel berbeda pada konfigurasi hyperparameter dalam eksperimen yang dilakukan, dapat dilihat pada Gambar 1.

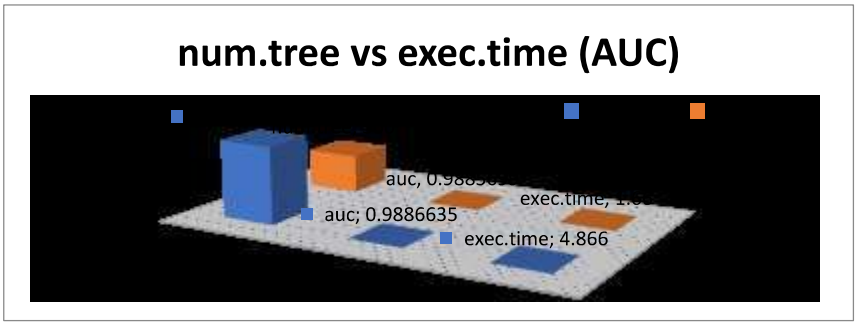
Pada eksperimen pertama, ditetapkan untuk hyperparameter jumlah pohon atau num.tree = 1000, dan dilakukan penyetelan hyperparameter secara otomatis sebanyak 70 kali iterasi ditambah 30 kali pemanasan diawal, algoritma klasifikasi menghasilkan tingkat akurasi prediksi rata-rata AUC = 98,7 %. Selama 70 kali iterasi tersebut itu, diperoleh nilai prediksi AUC tertinggi adalah 98,9% dengan mtry bervariasi antara 4 atau 5 sementara dan hasil prediksi AUC yang terendah yaitu 98,1% dengan mtry bervariasi antara 2 atau 7. Dengan penyetelan otomatis ini diperoleh rekomendasi konfigurasi hyperparameter untuk nilai mtry = 5, min.node = 3, sample.frac = 0.860, dengan capaian tingkat kinerja prediksi salah satu metode yaitu AUC = 0.988 atau 98,8%. *Kedalaman pohon mtry merupakan hyperpa*rameter yang kritis didalam eksperimen ini, jika nilai yang lebih kecil dipilih untuk kedalaman maka model akan mengalami kekurangan penyusunan atau mereduksi nilai kinerja prediksi.



Gambar 1. Grafik konfigurasi penyetelan hyperparameter dan kinerja prediksi AUC

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Pada eksperimen ke-2, digunakan paket dan pengaturan variabel standar yang sama dengan eksperimen pertama, dan selanjutnya "trade-off" diamati untuk melihat perbandingan hyperparameter num.tree dan exec.time, dengan hasil prediksi yang diperoleh. Jumlah pohon sebaiknya ditetapkan tinggi (hingga titik tertentu yang dapat diamati): semakin tinggi jumlah pohon, semakin baik hasil dalam hal kinerja, namun tentunya akan memberi konsekuensi yang buruk pada *exec.time (*waktu eksekusi). Sebagaimana yang ditunjukkan pada Gambar 2.



Gambar 2. *Trade-off* antara jumlah pohon (num.tree) dan waktu (exec.time) terhadap hasil prediksi AUC

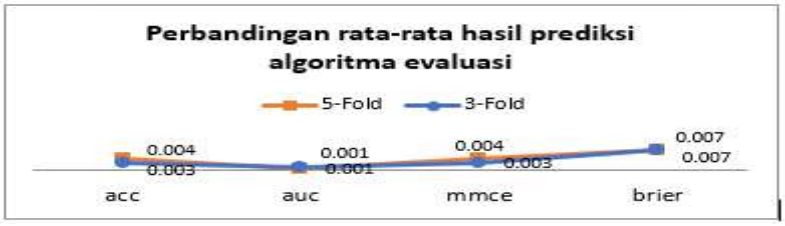
Pada eksperimen ke-3, dilanjutkan dengan penyetelan hyperparameter num.tree = 1000. Iterasi berlangsung sebanyak jumlah k dan m kali merujuk pada pola *k-fold cross validation* yang dipilih sebagai metode validasi. Eksperimen dieksekusi secara berurutan dengan skenario 3-times-3-fold dan 5-times-5-fold validation. Hasil yang dijadikan sebagai output adalah perbandingan ukuran kinerja prediksi antara algoritma "ranger" standar (tanpa penyetelan hyperparameter) dan algoritma "ranger" dengan penyetelan hyperparameter.

Tabel 2: Kinerja klasifikasi rata-rata dari 4 (empat) evaluasi ukuran kinerja yang digunakan dengan validasi *k-fold cross validation*, dimana k=3 dan 5.

	Accuracy	AUC	MMCE	Brier Score	
penyetelan vs standar (3-fold)	0.947	0.985	0.053	0.087	tanpa penyetelan
	0.950	0.986	0.050	0.080	dengan penyetelan
penyetelan vs standar (5-fold)	0.950	0.986	0.050	0.083	tanpa penyetelan
	0.954	0.987	0.046	0.075	dengan penyetelan

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

Selanjutnya, diperoleh output nilai dari 4 (empat) algoritma evaluasi prediksi yaitu Accuracy, AUC, MMCE serta Brier Score, dan diperoleh hasil sebagaimana yang disajikan dalam Tabel 2. serta rangkuman proses keseluruhan perbandingan pada Gambar 3 dalam makalah ini. Dapat diamati pada Tabel 2, bahwa algoritma evaluasi mengalami peningkatan kinerja prediksi. Peningkatan hasil prediksi pada 2 metode evaluasi, yaitu Accuracy = .004 dan .003, AUC = 0.001 untuk 3-fold dan 5-fold dan penurunan nilai pada 2 metode lainnya yaitu MMCE = 0.003 dan .0.004 serta Brier Score = 0.007 untuk kedua teknik lipat atau k-fold, dan ini telah sesuai dengan kecendrungan nilai optimal untuk setiap metode.



Gambar 3. Perbandingan hasil rata-rata algoritma evaluasi pada *random forest* dengan penyetelan parameter (tuneRanger) dan tanpa penyetelan (Ranger)

Sehingga dapat dikatakan bahwa optimalisasi kinerja prediksi dapat dicapai melalui penyetelan hyperparameter jika dibandingkan dengan hasil prediksi dari algoritma standar sebagai pembanding.

IV.KESIMPULAN

Kesimpulan yang dapat diambil adalah bahwa peningkatan jumlah pohon atau *num.tree* secara linier meningkatkan akurasi model. Semakin luas ukuran hutan lebih baik keakuratannya, tetapi keakuratannya tidak akan berubah pada tingkat tertentu bahkan jika ada peningkatan jumlah pohon. Algoritma *random forest* memiliki beberapa hyperparameter yang dapat memengaruhi kinerjanya. Hyperparameter *min.node.size atau j*umlah minimum observasi di terminal node, *mtry*, dan *sample.frac* adalah parameter yang mengontrol keacakan algoritma. Dari parameter ini, *mtry* paling berpengaruh pada kinerja algoritma baik menurut literatur dan berdasarkan eksperimen yang kami kerjakan. Nilai terbaik *mtry* tergantung pada jumlah variabel yang berpengaruh pada hasil. Ukuran sampel dan

JURNAL INSTEK
INFORMATIKA SAINS DAN TEKNOLOGI
P –ISSN : 2541-1179, E-ISSN : 2581-1711
Ojs :http://journal.uin-alauddin.ac.id/index.php/instek/index
Email : instek@uin-alauddin.ac.id

ukuran node memiliki pengaruh yang rendah pada kinerja namun tetap layak untuk disesuaikan sebagaimana yang juga kami tunjukkan secara empiris dalam eksperimen kami.

DAFTAR PUSTAKA

Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE access*, *6*, 33789-33795.

Biau, G. and Scornet, E. (2016) A Random Forest guided tour. Test, 25, 197–227.

Bischl, B., Richter, J., Bossek, J., Horn, D., Thomas, J. and Lang, M. (2017) mlrMBO: A modular framework for model-based optimization of expensive black-box functions. ArXiv preprint arXiv:1703.03373. URL: https://arxiv.org/abs/1703.03373.

Breiman, L. (1996) Out-of-bag estimation. Tech. rep., UC Berkeley, Department of Statistics.— (2001) Random forests. Machine Learning, 45, 5–32.

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Janitza, S., & Hornung, R. (2018). On the overestimation of random forest's out-of-bag error. PLoS One, 13, e0201904.

Lukito, Y., & Rahmat, A (2017). Deteksi Komentar Spam Bahasa Indonesia Pada Instagram Menggunakan Naive Bayes. *Ultimatics: Jurnal Teknik Informatika*, *9*(1), 50-58.

Martínez-Muñoz, G., & Suárez, A. (2010). Out-of-bag estimation of the optimal sample size in bagging. Pattern Recognition, 43, 143–152.

Probst, P., Bischl, B. and Boulesteix, A.-L. (2018) Tunability: Importance of hyperparameters of machine learning algorithms. ArXiv preprint arXiv:1802.09596. URL: https://arxiv.org/abs/1802.09596.

Seibold, H., Bernau, C., Boulesteix, A.-L., & De Bin, R. (2018). On the choice and influence of the number of boosting steps for high-dimensional linear cox-models. Computational Statistics, 33, 1195–1215.

Wicaksono, A. S., & Supianto, A. A. (2018). Hyper parameter optimization using genetic algorithm on machine learning methods for online news popularity prediction. Int. J. Adv. Comput. Sci. Appl, 9(12), 263-267.

Wright, M. N. and Ziegler, A. (2017) ranger: A fast implementation of random forests for high dimensional data in C++ and R. Journal of Statistical Software, 77, 1–17.