

Optimalisasi Hyperparameter pada Model Deteksi Transaksi Mencurigakan Menggunakan Grid-Search

1st Gilman Muslih Zakir
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

muslihgilman@student.telkomuniversity.ac.id

2nd Anggunmeka Luhur Prasasti
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

anggunmeka@telkomuniversity.ac.id

3rd Marisa W. Paryasto
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

marisaparyasto@telkomuniversity.ac.id

Abstrak — *Fraud*, sebagaimana didefinisikan oleh Association of Certified Fraud Examiners (ACFE) mencakup laporan keuangan yang keliru atau penipuan yang dibuat untuk memperoleh keuntungan yang tidak sah. Salah satu bentuk *fraud* adalah pencucian uang, di mana uang ilegal dipindahkan melalui sistem keuangan untuk membuatnya tampak sah. Panel Tingkat Tinggi International Financial Accountability, Transparency and Integrity (Panel FACTI) memperkirakan sekitar \$1,6 triliun (2,7% dari PDB global), dicuci setiap tahun. Adanya transaksi keuangan yang mencurigakan memerlukan deteksi dini oleh lembaga keuangan untuk mencegah penyalahgunaan. Salah satu hal yang ingin dicapai dengan penelitian ini adalah bagaimana cara meningkatkan akurasi dan efisiensi dalam mendeteksi transaksi mencurigakan menggunakan teknologi Machine Learning. Penggunaan teknologi machine learning merupakan salah satu Solusi untuk mengatasi tantangan dalam mendeteksi transaksi mencurigakan. Penelitian ini dilakukan dengan mengembangkan model deteksi transaksi mencurigakan menggunakan algoritma XGBoost, Decision Tree, dan Logistic Regression dengan menerapkan Hyperparameter tuning yang dibantu dengan pencarian hyperparameter terbaik menggunakan Grid-Search untuk mendapatkan performa terbaik dari model yang dikembangkan.

Kata kunci— *xgboost, decision tree, logistic regression, hyperparameter tuning, grid-search*

I. PENDAHULUAN

Fraud menurut Association of Certified Fraud Examiners (ACFE) adalah laporan keuangan yang keliru atau penipuan yang dibuat oleh suatu entitas atau individu dengan mengetahui bahwa hal tersebut dapat dilakukan untuk memperoleh keuntungan yang tidak sah [1]. Salah satu bentuk fraud adalah pencucian uang. Pencucian uang adalah ketika uang yang bersifat ilegal dipindahkan melalui sistem keuangan untuk membuat uang tersebut terlihat sah. Menurut Panel Tingkat Tinggi International Financial Accountability, Transparency and Integrity (Panel FACTI) sekitar \$1,6 Triliun atau setara dengan 2,7% dari PDB global dicuci setiap tahun [2].

Perpindahan atau pertukaran uang dapat dilakukan dengan cepat dan mudah. Perpindahan uang dapat mencapai

batas negara bahkan di luar wilayah. Adanya kegiatan transaksi juga disebut perpindahan uang. Menurut Undang-Undang No. 8 Tahun 2010 tentang Pencegahan dan Pemberantasan Tindak Pidana Pencucian Uang, transaksi keuangan adalah transaksi di mana uang ditempatkan, diserahkan, ditarik, ditransfer, atau dilakukan kegiatan lainnya yang terkait dengan uang. Dalam konteks ini, tidak dapat dipungkiri bahwa ada pihak-pihak yang tidak bertanggung jawab yang memicu transaksi keuangan yang mencurigakan. Sebagaimana yang telah dijelaskan oleh Rezim Anti Pencucian Uang, lembaga keuangan atau penyedia jasa keuangan memiliki tanggung jawab penting untuk mendeteksi secara dini adanya transaksi keuangan mencurigakan. Deteksi ini dapat dicapai melalui laporan transaksi keuangan mencurigakan yang dikirim kemudian ke lembaga intelijen keuangan terkait. Jika lembaga keuangan mencurigai atau memiliki alasan yang masuk akal untuk mencurigai bahwa dana yang ada berasal dari kegiatan kriminal atau terkait dengan pendanaan teroris, laporan transaksi keuangan yang mencurigakan dapat dimulai dengan praduga [3].

Identifikasi yang akurat sangat penting untuk mendukung upaya penegakan hukum mengingat proses penipuan yang dilakukan oleh pelaku melibatkan langkah-langkah yang rumit. Untuk mengatasi tantangan ini, diperlukan sebuah teknologi yang andal dan efisien dalam penggunaan sumber daya. Salah satu upaya untuk mewujudkan hal tersebut adalah dengan mengembangkan sistem pendeteksi transaksi mencurigakan dengan memanfaatkan teknologi *machine learning*. Untuk membuat sistem yang baik diperlukan sebuah model dengan algoritma yang andal dalam mendeteksi sebuah transaksi. XGBoost, *Decision Tree*, dan *Logistic Regression* merupakan pilihan algoritma yang akan diterapkan pada model mengingat kemampuan algoritma-algoritma tersebut cukup baik dalam melakukan klasifikasi biner [4], [5]. Beberapa metode *preprocessing* dapat diterapkan untuk meningkatkan performa sebuah model salah satunya adalah dengan menerapkan batasan *hyperparameter* pada saat melatih model. *Grid-Search* dapat digunakan untuk menemukan *hyperparameter* yang optimal dengan cara mencari

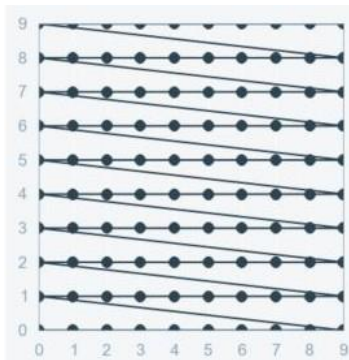
kombinasi *hyperparameter* secara sekuensial yang disusun ke dalam sebuah *grid* multidimensi. Proses ini diharapkan dapat meningkatkan performa model dalam mendeteksi transaksi mencurigakan.

II. KAJIAN TEORI

Hyperparameter adalah batasan yang menentukan bagaimana model *machine learning* belajar dan memengaruhi kemampuan model untuk memprediksi atau mengklasifikasikan data pengujian [6].

A. Grid-Search

Algoritma Grid-Search digunakan untuk menemukan *hyperparameter* yang optimal dari sebuah model untuk menghasilkan prediksi yang paling akurat [7]. *Hyperparameter* yang baik dapat ditemukan menggunakan *Grid-Search*. Strategi pencarian *grid* didasarkan pada pengujian semua kombinasi *hyperparameter* yang ditentukan dalam *grid* multidimensi. Performa dari kombinasi *hyperparameter* dievaluasi dengan menggunakan metrik performa. Konfigurasi dengan kinerja terbaik akan dipilih dan digunakan untuk melatih model *machine learning*. Di sisi lain, jika dibandingkan dengan algoritma pencarian lain, *grid-search* sangat memakan waktu akan tetapi cocok untuk penyesuaian beberapa *hyperparameter*, serta lebih mudah untuk diimplementasikan [8].



Gambar 1 Representasi Visual Grid-Search

B. Hyperparameter Tuning

Hyperparameter tuning adalah metode terbaik untuk menjalankan sejumlah kombinasi parameter yang berbeda untuk menilai kinerja pengklasifikasi. Menilai pengklasifikasi menggunakan data pelatihan akan mengarah pada masalah pembelajaran mesin yang mendasar yang disebut *overfitting*. *Overfitting* adalah situasi di mana sebuah model berkinerja buruk pada data uji dan sangat baik pada data pelatihan. Oleh karena itu, validasi silang atau *cross validation* digabungkan dengan metode *grid-search* untuk optimasi *hyperparameter* [9].

III. METODE

Penelitian dilakukan menggunakan *dataset* transaksi keuangan nasabah tahun 2022. *Dataset* ini terdiri data demografis dan transaksi nasabah. Data yang digunakan dalam penelitian ini merupakan data historis yang sudah dilabeli untuk memudahkan proses klasifikasi. *Dataset* ini terdiri dari 10.661 baris data transaksi keuangan yang sebagian besar terdiri dari transaksi mencurigakan dengan total transaksi mencurigakan sebanyak 5.559 transaksi atau sebesar 52.15% dan 5.101 transaksi yang wajar atau 47.85%

dari total data transaksi keuangan secara keseluruhan. Dengan jumlah tersebut, diharapkan agar model tidak mengalami *oversampling* pada kelas tertentu. Selain itu, dilakukan pemeriksaan dan penanganan pada *missing values*. Penanganan yang dilakukan adalah dengan menghapus *missing values* dikarenakan jumlah yang hanya sedikit sehingga tidak memengaruhi *dataset* secara keseluruhan. Setelah melakukan serangkaian proses analisis data, selanjutnya melakukan *preprocessing* pada *dataset*, seperti mentransformasi tipe data semua kolom yang bertipe *object* ke dalam bentuk *integer* menggunakan *label encoding*, analisis korelasi antara fitur dengan fitur dan fitur dengan target, normalisasi data, serta persiapan data sebelum pelatihan untuk membantu mentransformasi data dalam membangun model *machine learning* yang baik [10].

A. Implementasi Grid-Search

Proses pencarian *hyperparameter* dilakukan menggunakan modul *GridSearchCV* yang disediakan oleh *library* *Scikit-learn*. Sebelum memulai pencarian, pertama-tama menginisiasi model mengingat algoritma pencarian *hyperparameter* bekerja dengan cara melatih model menggunakan kombinasi *hyperparameter* yang telah disusun ke dalam sebuah *grid* multidimensi. *Hyperparameter* tersebut kemudian dimasukkan ke dalam sebuah *dictionary* dengan masing-masing nilai *hyperparameter* yang telah ditentukan. *Dictionary* yang berisi *grid* dari beberapa kombinasi *hyperparameter* akan digunakan sebagai parameter variabel *Grid-Search*. Algoritma pencarian *Grid-Search* juga diatur dengan parameter *cross validation* dengan jumlah *n* yang artinya proses pelatihan akan dibagi menjadi *n* lipatan (*n-1* variasi skala pembagian data pelatihan dan sisanya menjadi data pengujian) [11]. Penilaian dilakukan menggunakan akurasi pelatihan tertinggi yang dihasilkan oleh kombinasi *hyperparameter* yang terbaik.

B. Implementasi Hyperparameter Tuning

Kombinasi *hyperparameter* yang diperoleh setelah mengimplementasikan *Grid-Search* akan diterapkan pada model.

TABEL I.
Kombinasi Grid Hyperparameter Tiap Model

| Model | Hyperparameter | Nilai |
|----------------------------|-------------------|---|
| <i>Logistic Regression</i> | C | 0.01, 0.1 |
| | Penalty | 'l2', 'l1' |
| | Solver | 'newton-cg' (hanya mendukung penalty 'l2' atau 'None'), 'lbfgs', 'liblinear', 'sag' |
| <i>Decision Tree</i> | Criterion | 'gini', 'entropy' |
| | Max_depth | None, 10, 20, 30, 40, 50 |
| | Min_samples_leaf | 1, 2, 5, 10 |
| | Min_samples_split | 2, 5, 10 |
| <i>XGBoost</i> | Learning_rate | 0.01, 0.1, 0.2 |
| | Max_depth | 3, 6, 9 |
| | n_estimators | 50, 100, 200, 300 |

Beberapa *hyperparameter* telah ditentukan untuk mengatur jalannya pelatihan model. Misalnya *hyperparameter* C digunakan untuk mengatur seberapa kuat

regularisasi model *logistic regression*, 'max_depth' pada *decision tree* dan *XGBoost* untuk mengatur kedalaman pohon selama pelatihan, dan 'n_estimators' atau *number of estimators* digunakan untuk menentukan jumlah pohon yang akan digunakan selama proses pembelajaran model. Secara keseluruhan, nilai-nilai *hyperparameter* ini dipilih berdasarkan pengalaman praktik terbaik, dan hasil eksperimen dari para ahli yang menunjukkan bahwa rentang tersebut umumnya menghasilkan performa yang baik dalam berbagai skenario.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pencarian Grid-Search

Setelah menyelesaikan pencarian menggunakan *Grid-Search*, diperoleh beberapa *hyperparameter* terbaik untuk ketiga model. Detail *hyperparameter* yang didapatkan dapat dilihat pada Tabel II.

TABEL II.
Hasil Pencarian Grid-Search

| Model | Hyperparameter |
|----------------------------|-----------------------|
| <i>Logistic Regression</i> | C: 0.01 |
| | Penalty: 'l2' |
| | Solver: 'newton-cg' |
| <i>Decision Tree</i> | Criterion: 'entropy' |
| | Max_depth: 40 |
| | Min_samples_leaf: 10 |
| | Min_samples_split: 10 |
| <i>XGBoost</i> | Learning_rate: 0.1 |
| | Max_depth: 6 |
| | n_estimators: 300 |

Pencarian *grid-search* menghasilkan *hyperparameter* yang akan mengatur proses pelatihan model. *Hyperparameter* tersebut di antaranya C atau kekuatan regularisasi pelatihan model *logistic regression* sebesar 0.01, *criterion* 'entropy' sebagai kriteria pembagian pada node menunjukkan bahwa model memilih pemisahan berdasarkan *information gain*, yang lebih sensitif terhadap distribusi kelas [12], dan *number of estimators* atau 'n_estimators' untuk mengatur jumlah pohon keputusan yang digunakan dalam *boosting* model *XGBoost* sebanyak 300.

B. Pelatihan Model dengan Hyperparameter Tuning

Untuk dapat menerapkan *hyperparameter* pada tiap model, inisialisasi tiap model dengan menambahkan parameter yang berisi *hyperparameter* sesuai hasil pencarian *grid-search*. Perbandingan hasil pengujian model tanpa dan dengan menggunakan *hyperparameter tuning* sebelum pelatihan dapat dilihat pada Tabel III.

TABEL III. PERBANDINGAN HASIL PENGUJIAN MODEL

| Model | Accuracy | Accuracy (Hyperparameter Tuning) |
|----------------------------|----------|----------------------------------|
| <i>Logistic Regression</i> | 62.98% | 63.70% |
| <i>Decision Tree</i> | 91.40% | 92.03% |
| <i>XGBoost</i> | 95.72% | 96.15% |

Pengujian dilakukan dengan membagi *dataset* menjadi 70% data pelatihan dan 30% data pengujian. Model *XGBoost* memiliki akurasi tertinggi di antara model lain sebesar 96,15%.

Model *logistic regression* mengalami kenaikan akurasi sebesar 0,72%. Meskipun tidak signifikan, setelah menetapkan *hyperparameter* seperti C, *penalty*, dan *solver*, model mengalami sedikit perbaikan. Selain itu, ada beberapa faktor yang memengaruhi performa model *logistic regression* di luar dari penetapan *hyperparameter*, seperti pola yang kompleks pada *dataset*, multikolinearitas antara fitur pada *dataset*, dan faktor lainnya.

Selain itu, model *decision tree* mengalami kenaikan sebesar akurasi sebesar 0,63% setelah *hyperparameter tuning*. Meskipun tidak memberikan dampak yang terlalu signifikan, dapat dikatakan model *decision tree* memiliki performa yang baik dalam mendeteksi transaksi mencurigakan. Begitu juga dengan model *XGBoost*, model ini memiliki performa yang sangat baik meskipun hanya mengalami kenaikan sebesar 0,43% setelah melakukan *hyperparameter tuning*. Meskipun demikian, *hyperparameter tuning* merupakan salah satu tahap persiapan yang perlu dilakukan sebelum melatih model, mengingat proses ini dapat meningkatkan performa ketiga model, terutama dalam menghasilkan prediksi yang lebih akurat untuk mendeteksi transaksi mencurigakan.

V. KESIMPULAN

Berdasarkan hasil pengamatan, penerapan *Grid-Search* untuk mencari kombinasi *hyperparameter* optimal pada model *Logistic Regression*, *Decision Tree*, dan *XGBoost* menunjukkan hasil yang cukup positif dalam meningkatkan akurasi model. *Hyperparameter tuning* menghasilkan peningkatan akurasi pada semua model yang diuji, dengan model *XGBoost* mencapai akurasi tertinggi sebesar 96,15%, diikuti oleh *Decision Tree* dan *Logistic Regression* dengan akurasi masing-masing sebesar 92,03% dan 63,70%. Dengan serangkaian percobaan beserta hasil yang didapatkan, dapat disimpulkan bahwa teknik *hyperparameter tuning* menggunakan *Grid-Search*, dapat memberikan perbaikan dalam kinerja model deteksi transaksi mencurigakan.

REFERENSI

- [1] J. Yao, J. Zhang and L. Wang, "A financial statement fraud detection model based on hybrid data mining methods," *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 57-61, 2018, doi: 10.1109/ICAIBD.2018.8396167.
- [2] R. Frumerie, "Money Laundering Detection using Tree Boosting and Graph Learning Algorithms," M.S. thesis, Dept. Mathematics., KTH., Stockholm, Sweden, 2021. [Online]. Available: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1663255&dsid=6464>
- [3] N. Alfa, S. Mawar, N. H. Siahaan, R. Putri, "Memahami Transaksi Keuangan Mencurigakan," ppatk.go.id. Accessed: Oct. 10, 2023. [Online]. Available: https://www.ppatk.go.id/siaran_pers/read/953/memahami-transaksi-keuangan-mencurigakan.html
- [4] L. Wahab and H. Jiang, "A comparative study on machine learning based algorithms for prediction of motorcycle crash severity," *PLOS ONE*, vol. 14, no. 4,

- pp. 1-17, Apr. 2019. [Online]. Available: <https://doi.org/10.1371/journal.pone.0214966>.
- [5] J. Watt, R. Borhani, and A. K. Katsaggelos, "Linear Two-Class Classification" in *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge: Cambridge University Press, 2020, ch. 6, sec. 2, pp. 125. [Online]. Available: <https://books.google.com/>
- [6] C. Arnold, L. Biedeback, A. Küpfer, and M. Neunhoeffler, "The role of hyperparameters in machine learning models and how to tune them," *Political Science Research and Methods*, vol. 1, no. 1, pp. 1-8, Feb. 2024. doi: 10.1017/psrm.2023.61.
- [7] Y. A. Albastaki and W. Awad, Eds., "Using an Artificial Neural Network to Improve Email Security" in *Implementing Computational Intelligence Techniques for Security Systems Design*, Hershey, PA: Information Science Reference, IGI Global, 2020, p. 137.
- [8] R. Guido, M. C. Groccia, and D. Conforti, "A hyperparameter tuning approach for cost-sensitive support vector machine classifiers," in *Soft Computing*, vol. 27, no. 18, pp. 12863-12881, Sep. 2023, doi: 10.1007/s00500-022-06768-8.
- [9] G. SijiGeorgeC and B. Sumathi, "Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction," *International Journal of Advanced Computer Science and Applications*, vol. 11, pp. 1-8, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:222433720>.
- [10] G. SijiGeorgeC and B. Sumathi, "Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction," *International Journal of Advanced Computer Science and Applications*, vol. 11, pp. 1-8, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:222433720>.
- [11] N. Beheshti, "Cross validation and grid search using sklearn's GridSearchCV on random forest model," *Towards Data Science*, Feb. 5, 2022. [Online]. Available: <https://towardsdatascience.com>.
- [12] A. Saud, S. Shakya, and B. Neupane, "Analysis of Depth of Entropy and GINI Index Based Decision Trees for Predicting Diabetes," *Indian Journal of Computer Science*, vol. 6, pp. 19-28, Jan. 2022, doi: 10.17010/ijcs/2021/v6/i6/167641.