Komparasi Hasil Prediksi Dengan Menggunakan Hyperparameter Tuning Antara Random Search dan Grid Search

1st Ibnu Fazril
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
ibnufazril@student.telkomuniversity.ac.id

Anggunmeka Luhur Prasasti
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
anggunmeka@telkomuniversity.ac.id

Marisa W. Paryasto
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
marisaparyasto@telkomuniversity.ac.id

***Abstrak* — Peningkatan yang signifikan pada transaksi keuangan mencurigakan yang berpotensi merugikan lembaga keuangan dan masyarakat semakin luas. Pencucian uang dan penipuan finansial merupakan ancaman serius yang sulit dideteksi oleh sistem tradisional, yang sering kali tidak mampu mengimbangi kompleksitas metode kriminal yang semakin canggih. Masalah utama penelitian ini adalah bagaimana meningkatkan akurasi dan efisiensi dalam mendeteksi transaksi mencurigakan menggunakan teknologi Machine Learning. Penelitian ini dilakukan dengan mengembangkan model pendeteksi transaksi mencurigakan menggunakan algoritma XGBoost, Decision Tree, dan Logistic Regression dengan membandingkan pencarian parameter terbaik untuk Hyperparameter Tuning antara Random Search dan Grid Search dalam menghasilkan prediksi yang bernilai tinggi.**

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

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].

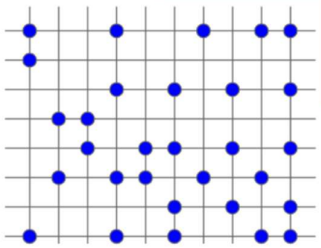
Perpindahan atau pertukaran uang dapat dilakukan dengan cepat dan mudah. Perpindahan uang dapat mencapai batas negara bahkan di luar wilayah. Adanya kegiatan transaksi juga disebut perpindahan uang. Menurut Undang-Undang No. 8 Tahun 2010 tentang Pencegahan dan Pemberantasan Tindak Pidana Pencucian Uang, 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].

Permasalahan yang sering dihadapi adalah ketika model tidak mampu untuk menghasilkan hasil prediksi yang tinggi padahal sudah diberlakukan *pre-processing*, untuk itu diperlukan adanya pengatur cara kerja tiap model dalam menghadapi kompleksitas data yang diberikan, metode itu dinamakan *Hyperparameter Tuning*, metode ini akan dipadukan dengan salah satu metode antara *Random Search* ataupun *Grid Search*, dengan memakai *Hyperparameter Tuning* akan membuat model siap menghadapi kompleksitas data dan akan menghasilkan hasil prediksi yang tinggi [4].

II. KAJIAN TEORI

A. Random Seach

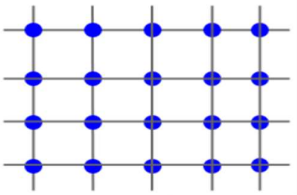
Random Seach merupakan sebuah metode pencarian parameter terbaik tiap model berdasarkan kombinasi acak yang ditemukan dalam proses pencarian, fungsi *Random* Search akan membuat model yang dipakai dalam pelatihan baik itu *Logistic Regression*, *Decision Tree*, maupun *XGBoost* akan menjadi lebih fleksibel terhadap data yang diberikan [5]



Gambar 1
Representasi Visual Random Search

B. Grid Search

Grid Search merupakan sebuah metode pencarian parameter terbaik tiap model berdasarkan kombinasi dari *grid* yang diberikan, fungsi *Grid* Search akan membuat model yang dipakai dalam pelatihan baik itu *Logistic Regression*, *Decision Tree*, maupun *XGBoost* akan menjadi lebih fleksibel terhadap data yang diberikan [6].



Gambar 2
Representasi Visual Grid Search

C. Hyperparameter Tuning

Hyperparameter tuning merupakan salah satu metode *Deep Learning* untuk membuat model menjadi lebih fleksibel dikarenakan parameter-parameter yang mengatur model untuk memprediksi bisa kita rubah sesuai dengan data yang diberikan baik data itu kompleks atau tidak, dan parameter ini sangat berguna untuk membuat prediksi model menjadi lebih akurat dan bisa mencegah *underfitting* maupun *overfitting* [7].

III. METODE

Penelitian dilakukan dengan menggunakan *dummy dataset* yang berisi sekitar 10 ribu baris data dan berisi banyak variabel independen seperti Gaji yang akan digunakan sebagai pembelajaran model untuk memprediksi. Data yang dimasukkan ke dalam sistem pendeteksian harus melalui tahapan *pre-processing* atau pembersihan data seperti merubah baris data menjadi numerik dan lain-lain, setelah melalui tahapan itu sistem pendeteksian harus tahu variabel dependen mana yang ingin diprediksi dan juga data harus dibagi menjadi data latih dan data uji, data latih sebagai bahan pembelajaran model, dan data uji sebagai soal yang harus diprediksi oleh model, tentunya agar model tidak *bias* terhadap salah satu variabel independen maka diperlukan normalisasi data yang berguna untuk menyeimbangkan bobot variabel independen, selanjutnya akan diperlukan fitur *Random Seach* ataupun *Grid Search* untuk mencari parameter *Hyperparameter Tuning* yang sekiranya terbaik bagi tiap metode.

A. Implementasi Random Search

Random Search merupakan sebuah framework dari scikit-learn yaitu *RandomizedSearchCV*. Proses *Random Search* dalam mencari parameter terbaik membutuhkan *range* nilai parameter yang akan diuji atau beberapa nilai parameter yang berada dalam *grid* seperti halnya dalam *Grid Search*. Pelatihan dan pengujian yang dilakukan dalam *Random Search* diatur oleh variabel *Cross Validation* yang akan menguji sebuah kombinasi parameter terhadap beberapa pembagian data yang berbeda-beda sebanyak n, dan proses itu akan diulang sampai tidak ada lagi kombinasi yang bisa dilakukan atau membatasi pengulangan dengan cara memberi iterasi, sampai pada akhirnya akan ditinjau dan diberikan parameter apa saja yang terbaik dari proses yang dilakukan [8].

B. Implementasi Grid Search

Grid Search merupakan sebuah framework dari scikit-learn yaitu *GridSearchCV*. Proses *Grid Search* dalam mencari parameter terbaik membutuhkan nilai dari parameter yang berada dalam *grid*, sampai pada akhirnya akan ditinjau dan diberikan parameter apa saja yang terbaik dari proses yang dilakukan [9].

C. Grid

Setelah dilakukan proses *Random Search* dan *Grid Search*, nilai dari parameter tersebut dimasukkan ke dalam sistem pendeteksian, *grid* yang akan dipakai dalam pengujian ditampilkan dalam berikut ini:

TABEL 1

PARAMETER GRID UNTUK RANDOM SEARCH DAN GRID SEARCH

Model	Hyperparameter	Nilai
Logistic Regression	C	10, 50, 100
	Solver	liblinear, lbfgs, saga
Decision Tree	Max Depth	5, 8, 10
	Min Samples Split	10, 30, 50
	Min Samples Leaf	5, 10, 20
XGBoost	Max Depth	3, 4, 5, 6
	Learning rate	0.01, 0.05, 0.1
	n_estimators	100, 200, 300

Parameter yang akan digunakan adalah C (regularization strength) dan Solver (“liblinear”, “lbfgs”, dan

"saga") untuk model Logistic Regression. Solver adalah algoritma yang digunakan dalam optimasi. Solver "lbfgs" relatif berkinerja baik dibandingkan dengan metode lain dan menghemat banyak memori meskipun terkadang ada masalah dengan konvergensi sehingga dapat menggunakan opsi lain. Solver "liblinear" juga dapat digunakan mengingat data set yang digunakan memiliki banyak fitur atau berdimensi tinggi [10].

Hyperparameter "max_depth", "min_samples_split", dan "min_samples_leaf" digunakan untuk tuning model Decision Tree. Parameter "max_depth" menentukan kedalaman maksimum setiap pohon keputusan dari decision tree. Nilai yang lebih rendah dapat membantu untuk mencegah overfitting. Namun, nilai yang terlalu rendah dapat menyebabkan masalah underfitting. Nilai yang lebih tinggi membuat waktu komputasi yang lebih lama. Parameter "min_samples_split" adalah jumlah minimum sampel yang diperlukan pada sebuah node sebelum pemisahan dilakukan sedangkan "min samples_leaf" menentukan jumlah minimum sampel yang diperlukan dalam node daun. Nilai yang lebih tinggi dapat membantu mencegah overfitting [11].

Untuk model XGBoost sendiri menggunakan parameter "n_estimators" dan "learning_rate" untuk tuning. Mengatur "learning_rate" dapat membuat model menjadi lebih stabil dan kuat. "n_estimators" atau number of estimators digunakan untuk menentukan jumlah iterasi. Jumlah pohon yang lebih banyak memberikan model kesempatan lebih besar untuk menangkap pola, tetapi juga meningkatkan resiko overfitting jika tidak disertai dengan pengaturan learning rate yang tepat [12].

IV. HASIL DAN PEMBAHASAN

A. Hasil Pencarian Random Search

Setelah menyelelesaikan pencarian maka parameter terbaik akan ditampilkan, berikut parameter terbaik yang didapat oleh *Random Search*:

TABEL 2

KOMBINASI GRID HYPERPARAMETER TIAP MODEL UNTUK RANDOM SEARCH

Model	Hyperparameter	Nilai
Logistic Regression	C	100
	Solver	lbfgs
Decision Tree	Max Depth	5
	Min Samples Split	30
	Min Samples Leaf	10
XGBoost	Max Depth	3
	Learning rate	0.05
	n_estimators	100

Parameter C bernilai 100 (regularisasi lemah), *Solver* menggunakan optimisasi lbfgs, *Max Depth* memiliki kedalaman pohon sebanyak 5 pertanyaan, *Min Samples Split* hanya boleh membagi data yang akan dibagi mencapai 30 baris data, dan untuk *Min Samples leaf* yaitu hasil pembagian data minimal harus mempunyai 10 baris data, *Max Depth* model *XGBoost* memiliki kedalaman pohon sebanyak 3 pertanyaan, *Learning Rate* sebesar 0.05 yang artinya model tidak terlalu cepat mempelajari data latih cukup stabil, dan terakhir yaitu *n_estimators* yaitu membuat 100 pohon keputusan untuk pelatihan.

B. Hasil Pencarian Grid Search

Setelah menyelelesaikan pencarian maka parameter terbaik akan ditampilkan, berikut parameter terbaik yang didapat oleh *Grid Search*:

TABEL 3

KOMBINASI GRID HYPERPARAMETER TIAP MODEL UNTUK GRID SEARCH

Model	Hyperparameter	Nilai
Logistic Regression	C	100
	Solver	saga
Decision Tree	Max Depth	10
	Min Samples Split	30
	Min Samples Leaf	10
XGBoost		
	Learning rate	0.1
	n_estimators	300

Parameter C bernilai 100 (regularisasi lemah), *Solver* menggunakan optimisasi saga, *Max Depth* memiliki kedalaman pohon sebanyak 10 pertanyaan, *Min Samples Split* hanya boleh membagi data yang akan dibagi mencapai 30 baris data, dan untuk *Min Samples leaf* yaitu hasil pembagian data minimal harus mempunyai 10 baris data, *Max Depth* model *XGBoost* memiliki kedalaman pohon sebanyak 5 pertanyaan, *Learning Rate* sebesar 0.1 yang artinya model lambat mempelajari data latih akan tetapi sangat stabil dalam mempelajari data latih, dan terakhir yaitu *n_estimators* yaitu membuat 300 pohon keputusan untuk pelatihan.

C. Komparasi Hasil Prediksi Antara Random Search Dan Grid Search

Setelah dilakukan pelatihan terhadap model yang digunakan, maka dilampirkan hasil prediksi antara *Tuning* memakai *Random Search* dan Tuning memakai *Grid Search:*

TABEL 4

HASIL PREDIKSI MODEL

Model	Akurasi Tes (Random Search)	Akurasi Tes (Grid Search)
Logistic Regression	62.34%	62.29%
Decision Tree	89.12%	91.09%
XGBoost	93.90%	95.50%

Dari tabel diatas, model *Logistic Regression* mendapatkan akurasi tes sebesar 62.34% untuk *Random Search* dan mendapatkan akurasi tes sebesar 62.29% untuk *Grid Search*, untuk model *Logistic Regression* mendapatkan akurasi tes sebesar 89.12% untuk *Random Search* dan mendapatkan akurasi tes sebesar 91.09% untuk *Grid Search,* untuk model terakhir yaitu *XGBoost* mendapatkan akurasi tes sebesar 93.90% untuk *Random Search* dan mendapatkan akurasi tes sebesar 95.50% untuk *Grid Search. Random search* hanya menang persentase di model *Logistic Regression*, sementara *Grid Search* menang persentase untuk model *Decision Tree*, dan model *XGBoost*.

V. KESIMPULAN

Berdasarkan hasil komparasi yang dilakukan, *Grid Search* sangat bisa diandalkan dalam mencari prediksi model dengan parameter yang sedikit, dikarenakan *Grid Search* mencari semua kombinasi parameter, sementara *Random Search* hanya mencari kombinasi acak dari *grid*

parameter yang diberikan dan akan menghentikan pencarian jika tidak ada lagi kombinasi acak yang bisa dilakukan, tetapi di pengujian lain yang menggunakan cukup banyak parameter, *Grid Search* tidak cukup andal karena pencarian parameter yang sangat lama, sementara *Random Search* melakukan pencarian dengan cukup cepat karena memakai grid (1,2,3), dan bukan range (1-50) serta membatasi iterasi yang dilakukan.

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&dswid=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] W. Nugraha and A. Sasongko, "Hyperparameter Tuning on Classification Algorithm with Grid Search," *SISTEMASI*, vol. 11, no. 2, p. 391, May 2022, doi: https://doi.org/10.32520/stmsi.v11i2.1750.

[5] Y. Özüpak, "Machine learning-based fault detection in transmission lines: A comparative study with random search optimization," Bulletin of the Polish Academy of Sciences Technical Sciences, pp. 153229–153229, 2025, doi: https://doi.org/10.24425/bpasts.2025.153229.

[6] A. F. D. Putra, M. N. Azmi, H. Wijayanto, S. Utama, and I. G. P. W. Wedashwara Wirawan, "Optimizing Rain Prediction Model Using Random Forest and Grid Search Cross-Validation for Agriculture Sector", *MATRIK*, vol. 23, no. 3, pp. 519–530, Jul. 2024, doi: 10.30812/matrik.v23i3.3891.

[7] M. Arifin and S. Adiyono, "Hyperparameter Tuning in Machine Learning to Predicting Student Academic Achievement," International Journal of Artificial Intelligence Research, vol. 8, no. 1.1, 2024, doi: https://doi.org/10.29099/ijair.v8i1.1.1214.

[8] D. A. Anggoro and S. S. Mukti, "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure," International Journal of Intelligent Engineering and Systems, vol. 14, no. 6, pp. 198–207, Aug. 2021, doi: https://doi.org/10.22266/ijies2021.1231.19.

[9] M. B. Prayoga, N. Cahyono, Subektiningsih, and Kamarudin, "PENERAPAN GRID SEARCH UNTUK OPTIMASI MODEL MACHINE LEARNING DALAM KLASIFIKASI SENTIMEN KOMENTAR YOUTUBE," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 3, pp. 3817–3824, June 2025, doi: https://doi.org/10.36040/jati.v9i3.13375.

[10] M. Gusarov, "Do I need to tune logistic regression hyperparameters?" Medium.com. Accessed: Aug. 11, 2024. [Online.] Available: https://medium.com/codex/do-i-need-to-tune-logistic-regression-hyperparameters-1cb2b81fc a69

[11] L. Owen, "Understanding the Hyperparameters of Popular Algorithms" in Hyperparameter Tuning with Python. Birmingham, UK: Packt Publishing Ltd., 2022, ch. 11, pp. 219–225.

[12] S. F. N. Islam, A. Sholahuddin, and A. S. Abdullah, "Extreme gradient boosting (XGBoost) method in making forecasting application and analysis of USD exchange rates against rupiah," in Journal of Physics: Conference Series, vol. 1722, no. 1, pp. 12-16, Jan. 2021, doi: 10.1088/1742-6596/1722/1/012016.