Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) p-ISSN: 2355-7699
Vol. 13, No. 3, Juni 2026, hlm. 521-530 e-ISSN: 2528-6579

PENERAPAN METODE STACKING ENSEMBLE LEARNING UNTUK DETEKSI PENIPUAN PADA TRANSAKSI FINANSIAL

Muh. Hafizh Izzaturrahim[*1], Ida Bagus Peling Prayoga[2], Andika Naufal Hilmy[3], Nugraha Priya Utama[4], Ayu Purwarianti[5]

[1,2,3,4,5] Institut Teknologi Bandung, Bandung, [1] Badan Riset dan Inovasi Nasional, Bandung
Email: [1] 23524029@mahasiswa.itb.ac.id, [2]23524068@mahasiswa.itb.ac.id, [3]23524069@mahasiswa.itb.ac.id, [4]utama@staff.stei.itb.ac.id, [5]ayu@staff.stei.itb.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 19 Desember 2024, diterima untuk diterbitkan: 15 Desember 2025)

Abstrak

Identifikasi *fraud* pada transaksi keuangan menjadi pekerjaan besar untuk mencegah kerugian yang dapat terjadi akibat *fraud*. Dengan perkembangan teknologi *machine learning* yang makin baik, eksperimen mengenai identifikasi *fraud* dapat dilakukan dengan menggunakan model *machine learning*. Sejumlah model *machine learning* dapat digunakan untuk identifikasi *fraud*, seperti *logistic regression*, *XGBoost, LightGBM,* dan *stacking ensemble*. Penelitian ini menerapkan metode eksperimen kuantitatif untuk menunjukkan kinerja dari *stacking ensemble* yang dipadukan dengan *Multi-Layer Perceptron* (MLP) sebagai *meta-learner*. Tahapan yang dilakukan yaitu eksplorasi data untuk memetakan karakteristik dataset IEEE-CIS Fraud Detection, *preprocessing* untuk menyiapkan data yang sesuai dengan model, pemodelan untuk klasifikasi biner, dan evaluasi terhadap data uji. Tantangan utama dalam deteksi *fraud* adalah data yang tidak seimbang antara transaksi otentik dan palsu. Meskipun demikian, Eksperimen dilakukan dengan pengolahan data dan *feature engineering* untuk mengatasi permasalahan dalam dataset. Pengolahan data dan *feature engineering* ditujukan untuk menangani pencilan, fitur yang tak relevan, dan nilai-nilai pada data yang hilang serta terduplikasi. Kemudian, pembuatan model dilakukan dengan *logistic regression, XGBoost, LightGBM,* dan *stacking ensemble* dengan *base learner* dari *XGBoost* dan *LightGBM* serta *meta-learner* menggunakan *Multi-Layer Perceptron*. Hasil dari eksperimen dan pengujian komparatif yang dilakukan menunjukkan bahwa model *stacking ensemble* memiliki kinerja yang paling tinggi dengan nilai AUC-ROC mencapai 0,9663, secara spesifik mengungguli model *LightGBM* (0,9661) dan *XGBoost* (0,9600).

Kata kunci: *fraud, klasifikasi, stacking, lightgbm, xgboost, MLP*

IMPLEMENTATION OF STACKING ENSEMBLE LEARNING METHOD FOR fraud DETECTION IN FINANCIAL TRANSACTIONS

Abstract

Fraud detection in financial transactions is a critical task to prevent potential losses caused by fraudulent activities. With the advancement of machine learning technology, experiments on fraud identification can now be conducted using various machine learning models. Several machine learning models can be used for fraud identification, such as logistic regression, XGBoost, LightGBM, and stacking ensemble. This study applies a quantitative experimental method to demonstrate the performance of a stacking ensemble combined with a Multi-Layer Perceptron (MLP) as a meta-learner. The methodology consists of several phases: data exploration to understand the IEEE-CIS Fraud Detection dataset's characteristics, data preprocessing to ensure suitability for the model, modeling for the purpose of binary classification, and evaluation against the test set. Despite this, the process involved data preprocessing and feature engineering to address issues within the dataset. These steps were undertaken to reduce outliers, eliminate irrelevant features, and handle missing or duplicated values. Model development was then carried out using logistic regression, XGBoost, LightGBM, and a stacking ensemble with XGBoost and LightGBM as base learners and MLP as the meta-learner. The results of the experiments and the comparative testing conducted indicate that the stacking ensemble model achieved the best performance with an AUC-ROC score of 0.9663, specifically outperforming the LightGBM (0.9661) and XGBoost (0.9600).

Keywords: *fraud, classification, stacking, lightgbm, xgboost, MLP*

1. PENDAHULUAN

Perkembangan sistem pembayaran elektronik yang pesat seiring transformasi digital di sektor keuangan telah meningkatkan kenyamanan dalam bertransaksi. Namun, bersamaan dengan itu muncul tantangan serius berupa peningkatan risiko penipuan (*fraud*). Penipuan dalam transaksi finansial tidak hanya mengakibatkan kerugian ekonomi secara langsung, tetapi juga mengancam kepercayaan pengguna terhadap sistem keuangan digital.

Skala ancaman ini tercermin dalam berbagai laporan global. Laporan Mastercard (2024) mencatat bahwa kerugian akibat *fraud* di sektor e-commerce mencapai USD 41 miliar hanya dalam satu tahun, menunjukkan bahwa solusi konvensional belum cukup efektif untuk mendeteksi pola serangan digital yang terus berkembang. Sementara itu, (Laxman et al., 2025) menyoroti meningkatnya ancaman kejahatan finansial global yang sejalan dengan melonjaknya nilai kerugian akibat penipuan digital di sektor keuangan. Temuan ini mempertegas perlunya sistem deteksi otomatis yang adaptif, efisien, dan mampu mengenali pola *fraud* yang kompleks dan tersembunyi.

Di Indonesia, Otoritas Jasa Keuangan (OJK) telah mewajibkan lembaga jasa keuangan untuk menerapkan strategi anti-fraud secara menyeluruh (OJK, 2024). Namun demikian, penerapan teknologi deteksi *fraud* masih terkendala oleh keterbatasan sistem yang mampu menangani data transaksi dalam jumlah besar, bersifat tidak seimbang, dan kompleks secara fitur.

Dalam beberapa tahun terakhir, machine learning muncul sebagai solusi potensial dalam mendeteksi *fraud* berbasis data historis (offline), terutama pada skenario yang tidak memungkinkan pemrosesan *real-time* (Ali et al., 2022). Model *supervised learning* seperti *logistic regression*, XGBoost, dan LightGBM telah banyak digunakan dalam konteks ini karena kemampuannya mempelajari pola dari data terdahulu. Akan tetapi, efektivitas model tunggal sering kali menurun ketika dihadapkan pada data yang sangat tidak seimbang. Misalnya, dominasi kelas non-fraud menyebabkan model cenderung bias, sebagaimana dilaporkan oleh Patel et al. (2024) dan Achary & Shelke (2023). Solusi seperti penyesuaian ambang klasifikasi atau penggunaan metrik alternatif seperti AUC-ROC membantu, namun belum menyelesaikan masalah pada tingkat fitur dan distribusi kelas.

Sebagai alternatif, pendekatan *ensemble learning* mulai banyak dilirik karena mampu menggabungkan kekuatan dari berbagai model dasar. Teknik seperti *Random Forest* dan *stacking ensemble* terbukti unggul dalam mengatasi ketimpangan data tanpa harus menggunakan teknik *oversampling* seperti *Synthetic Minority Over-sampling TEchnique* (SMOTE) (Suganya et al., 2023). Model *stacking* secara khusus memiliki keunggulan dalam menggabungkan prediksi dari model yang heterogen, memungkinkan pembelajaran non-linier dari hasil model dasar. Namun, efektivitas *stacking* sangat tergantung pada pemilihan meta-learner dan konfigurasi model dasar.

Penelitian terbaru oleh Khan et al. (2024) menggunakan *stacking ensemble* dengan *base learner* XGBoost, LightGBM, dan CatBoost, serta XGBoost sebagai *meta-learner* pada dataset IEEE-CIS. Studi ini menunjukkan performa tinggi, namun belum membandingkan secara eksplisit performa berbagai kombinasi *meta-learner*, termasuk *logistic regression* yang jauh lebih ringan dan efisien secara komputasi. Hal ini menunjukkan adanya *research gap* dalam pemahaman pengaruh pemilihan *meta-learner* terhadap performa akhir pada skema *stacking*, khususnya pada konteks deteksi *fraud* berbasis data tabular.

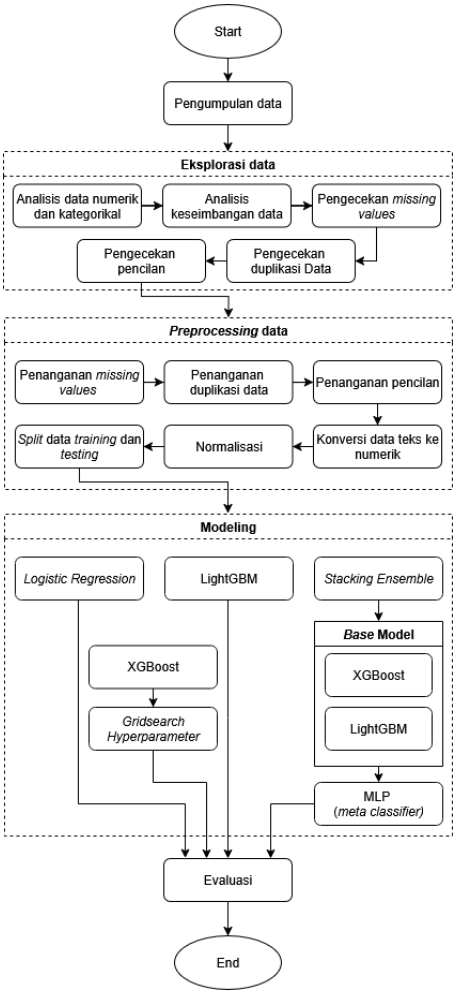
Oleh karena itu, penelitian ini bertujuan untuk melakukan evaluasi empiris terhadap efektivitas *stacking ensemble* dibandingkan dengan model individual (*logistic regression*, XGBoost, dan LightGBM) dalam mendeteksi fraud pada dataset IEEE-CIS. Penelitian ini tidak mengusulkan metode baru, melainkan menyumbang kontribusi orisinal berupa analisis komprehensif terhadap perbandingan performa klasifikasi baik dari segi akurasi, AUC-ROC, maupun efisiensi komputasi, tanpa menggunakan teknik *oversampling* seperti SMOTE. Penelitian ini juga menyoroti peran MLP sebagai *meta-learner* yang belum banyak dieksplorasi dalam konfigurasi *stacking ensemble* pada domain ini.

2. METODE PENELITIAN

Penelitian ini menggunakan beberapa tahapan pokok dalam pengembangan model *machine learning* dengan tujuan membandingkan hasil evaluasi dari berbagai pengembangan model untuk melakukan deteksi *fraud* pada transaksi finansial. Tahapan yang dilakukan antara lain pengumpulan data, eksplorasi data, *preprocessing* data, *modeling*, dan evaluasi model. Diagram dari setiap proses dapat dilihat pada Gambar 1.

2.1. Dataset

Penelitian ini menggunakan dataset yang berasal dari kompetisi IEEE-CIS Fraud Detection yang telah melalui beberapa tahap adaptasi dan pra-pemrosesan. Sumber data diperoleh melalui platform Kaggle-CIS (Kaggle, 2019), yang memuat rekam jejak transaksi finansial lengkap dengan label *fraud* (penipuan) maupun non-fraud. Untuk menjaga kerahasiaan informasi sensitif, beberapa atribut dalam dataset sengaja dilakukan penyamaran data (*data masking*).



Gambar 1. Alur Penelitian

Dataset ini terdiri dari empat *file* dalam format *Comma-Separated Values* (CSV), yaitu *train_+ transaction*, *train_identity*, *test_transaction*, dan *test_identity*. Berkas *train_transaction* dan *test_transaction* berisi informasi utama terkait transaksi finansial, sementara *train_identity* dan *test_identity* mengandung fitur tambahan mengenai identitas pengguna serta perangkat yang digunakan. Penelitian ini hanya menggunakan *file train_transaction* dan *train_identity* yang didesain memiliki label sebagai fitur dependen.

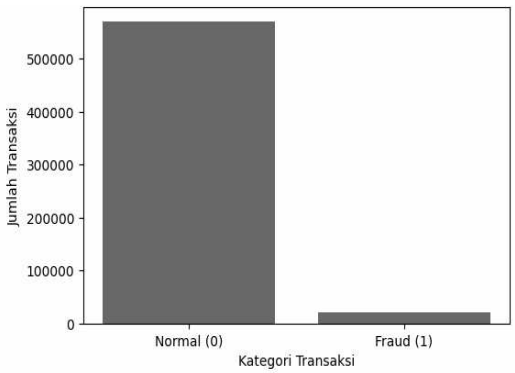
Dari segi karakteristik teknis, dataset ini mengandung variasi tipe data baik numerik maupun kategorikal. Dataset memiliki 590.540 baris dan 434 kolom. Dengan volume data yang cukup besar ini, total penyimpanan yang diperlukan mencapai 1,26 *gigabyte* (GB). Dataset ini memiliki fitur yang mencakup berbagai aspek transaksi, seperti waktu, jumlah nominal, informasi kartu, serta perangkat pengguna. Label *isfraud* digunakan untuk menandai apakah suatu transaksi terindikasi *fraud* atau tidak. Deskripsi kolom-kolom lainnya pada dataset ini dijelaskan pada Tabel 1.

Dataset memiliki karakter berupa jumlah data berlabel yang tak seimbang antara transaksi *fraud* atau transaksi *legitimate* (bukan *fraud*) dengan jumlah transaksi *fraud* yang jauh lebih sedikit yang terlihat dari Gambar 2. Data *fraud* memiliki persentase 3,5% dari keseluruhan dataset. Dataset ini memiliki kolom yang sangat banyak, tetapi banyak data kosong pada sejumlah kolom. Sejumlah kolom yang terdapat pada dataset ini berisi data yang telah disamarkan. Banyaknya data yang perlu dibersihkan cocok digunakan untuk ketahanan model.

Dataset ini bersumber dari transaksi pada sebuah *e-commerce* berskala besar yang dimiliki oleh Vesta Corporation dengan jumlah transaksi yang banyak sehingga memiliki representasi yang baik terhadap analisis *fraud detection* serta dapat mencerminkan kondisi di dunia nyata dengan variasi pola *fraud* yang beragam.

Tabel 1. Deskripsi Fitur

kolom	deskripsi
TransactionID	ID unik yang diberikan untuk setiap transaksi.
TransactionDT	Waktu transaksi dalam bentuk *timestamp* relatif (delta).
isfraud	Label transaksi, 1 untuk transaksi *fraud* dan 0 untuk transaksi normal.
TransactionAmt	Jumlah nominal transaksi dalam satuan mata uang tertentu.
ProductCD	Kode produk yang terkait dengan transaksi. Produk ini bisa berupa kartu kredit, debit, atau bentuk pembayaran lainnya.
card1 - card6	Informasi terkait kartu pembayaran yang digunakan dalam transaksi, Seperti ID unik kartu, bank penerbit atau tipe kartu.
addr1, addr2	Informasi alamat terkait transaksi.
dist1, dist2	Jarak terkait transaksi, kemungkinan antara pembeli dan penerima.
P_emaildomain	Domain email pembeli..
R_emaildomain	Domain email penerima.
C1-C14	Fitur numerik anonim
D1 - D15	Fitur terkait perbedaan waktu antara transaksi satu dengan transaksi lainnya.
M1 - M9	Fitur biner yang mencerminkan metode pembayaran atau status transaksi.
V1 - V339	Variabel anonim
id_1 - id_38	Berisi berbagai fitur terkait identitas pengguna dan perangkat yang digunakan, termasuk kemungkinan parameter risiko atau informasi tambahan dari penyedia layanan pembayaran.
DeviceType	Jenis perangkat yang digunakan untuk melakukan transaksi, seperti "desktop" atau "mobile".
DeviceInfo	Detail spesifik tentang perangkat, seperti model ponsel atau sistem operasi (misalnya, "Windows", dan "iPhone").



Gambar 2. Distribusi Transaksi Normal dan *Fraud*

2.2. Eksplorasi Data

Eksplorasi data dilakukan dengan empat tahap, yaitu eksplorasi deskriptif data numerik dan kategorikal, keseimbangan data, analisis data hilang, duplikasi data, dan pencilan. Adapun contoh satu data yang digunakan dapat dilihat pada Tabel 2 yang menampilkan hanya beberapa kolom saja. Tabel tersebut merupakan salah satu data yang dilakukan transpose untuk memperlihatkan lebih banyak kolom.

Tabel 2. Contoh Data

column	*value*
TramsactionID	2987000
isFraud	0
TransactionDT	86400
TransactionAmt	68,5
ProductCD	W
id_36	*NaN*
id_37	*NaN*
id_38	*NaN*
DeviceType	*NaN*
DeviceInfo	*NaN*

Berdasarkan eksplorasi deskriptif data numerik, sebagaimana di kolom *mean, std, min,* dan *max* pada Tabel 3, karakter distribusi pada tiap kolom berbeda satu sama lain. Perbedaan karakter distribusi utamanya ditandai dengan rata-rata sebagaimana di kolom *mean* yang bervariasi dan standar deviasi sebagaimana di kolom *std* yang mana semakin besar nilainya, semakin tersebar datanya. Beberapa kolom memiliki nilai pencilan akibat nilai standar deviasi yang tinggi, nilai maksimum yang jauh di atas rata-rata sebagaimana di kolom *max*, atau nilai minimum yang jauh di bawah rata-rata sebagaimana di kolom *min*. Misalnya, pada kolom *TransactionAmt* dengan nilai rata-rata 135,03, standar deviasi mencapai 239,15, dan nilai maksimum mencapai 31.937,39. Karakter pada kolom ini seharusnya ditemukan di kolom penanda waktu dan kolom identifikasi unik transaksi yang memiliki karakter serupa. Selain itu, sejumlah kolom memiliki persebaran nilai yang jauh lebih terpusat yang ditandai dengan nilai standar deviasi yang kecil.

Tabel 3. Eksplorasi Deskriptif Data Numerik

column	*count*	*mean*	*std*	*min*	*max*
TransactionID	590540	3282269,5	170474,35	2987000	3577539
TransactionDT	590540	7372311,31	4617223,64	86400	15811131
TransactionAmt	590540	135,03	239,15	251	31937,39
card1	590540	9898,73	4901,17	1000	18396
card2	581607	362,55	157,79	100	600
card3	588975	153,19	11,33	100	231
card5	586281	199,27	41,24	100	237
addr1	524834	290,73	101,74	100	540
addr2	524834	86,80	2,69	10	102
dist1	238269	118,50	371,87	0	10286
dist2	37627	231,85	529,05	0	11623

Pada data kategorikal, eksplorasi deskriptif dilakukan dengan mencari jumlah data unik, modus beserta frekuensinya dan jumlah data tak kosong. Sebagaimana pada kolom *count* pada Tabel 4, beberapa *column* memiliki nilai *null* yang banyak yang ditandai dengan nilai pada kolom *count* yang sedikit dibandingkan dengan jumlah *row* pada dataset, misalnya pada kolom *R_emaildomain*, *DeviceType*, dan *DeviceInfo*. Selain *column* dengan nilai *null* yang banyak, terdapat *column* dengan nilai unik yang banyak, seperti pada *column DeviceInfo* dengan 1.786 nilai unik sebagaimana ditandai di kolom *unique* pada Tabel 4. Adapun pada *column ProductCD*, *card4*, dan *card6* didominasi oleh satu nilai, masing-masing bernilai *W*, *visa*, dan *debit* dengan frekuensi, sebagaimana di kolom *freq* pada Tabel 3, yang melebihi separuh dari jumlah data tak kosong sebagaimana di kolom *count* pada tabel yang sama.

Analisis keseimbangan data dilakukan untuk menentukan penanganan yang dilakukan terhadap model. Seperti yang terlihat pada Gambar 2, dataset memiliki karakter yang tidak seimbang untuk kedua kelas *fraud* dan bukan *fraud,* dimana data bukan *fraud* sangat mendominasi dataset. Penanganan terhadap ketidakseimbangan dilakukan dengan menentukan model yang tahan terhadap ketidakseimbangan data serta memilih metrik evaluasi yang berfokus pada kemampuan model dalam mendeteksi kelas minoritas, dalam hal ini kelas *fraud*.

Tabel 4. Eksplorasi Deskriptif Data Kategorikal

column	*count*	*unique*	*top*	*freq*
ProductCD	590540	5	W	439670
card4	588963	4	visa	384767
card6	588969	4	debit	439938
P_emaildomain	496084	59	gmail.com	228355
R_emaildomain	137291	60	gmail.com	57147
DeviceType	140810	2	desktop	85165
DeviceInfo	118666	1786	Windows	47722
transaction_date	590540	182	2017-12-23	6852

Tabel 5. Fitur dengan Data Hilang Tertinggi

column	*data type*	*null frequency*	*null percentage*
id_24	*float64*	585793	0,9920
id_25	*float64*	585408	0,9913
id_26	*float64*	585377	0,9913
id_21	*float64*	585381	0,9913
id_08	*float64*	585385	0,9913
id_07	*float64*	585385	0,9913
id_27	*object*	585371	0,9912
id_23	*object*	585371	0,9912
id_22	*float64*	552913	0,9912

Tabel 6. Hasil Analisis Duplikasi Data

TransactionID	*isFraud*	*TransactionDT*	*ProductCD*	*card1*	*card2*
295456	0	7310775	W	7919	194
295457	0	7310775	W	7919	194
453414	0	11576951	W	1359	314
453415	0	11576951	W	1359	314
570433	0	15129302	W	2744	490
570434	0	15129302	W	2744	490

Eksplorasi terhadap *missing values* atau data yang hilang dilakukan dengan melihat persentase data yang hilang dari setiap fitur. Tabel 5 menunjukkan bahwa terdapat sembilan fitur dengan didominasi data yang hilang hingga mencapai 99%. Hal ini juga menunjukkan bahwa hampir setiap baris data memiliki nilai yang hilang.

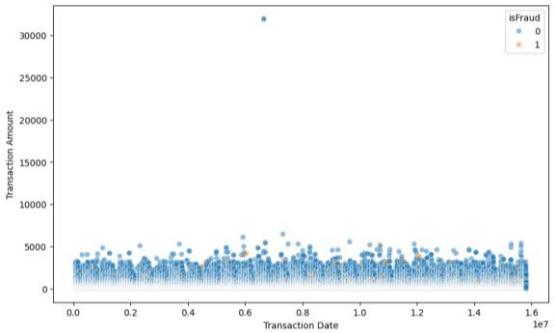
Analisis data duplikasi dilakukan dengan melihat data yang sama selain *TransactionID.* Hasil eksplorasi ditunjukkan pada Tabel 6 yang menampilkan beberapa kolom saja. Tabel 6 menunjukkan bahwa dataset ini memiliki enam data yang terduplikasi dengan satu data memiliki satu data dengan nilai identik.

Analisis pencilan dilakukan dengan melihat persebaran *TransactionAmt* atau nominal transaksi terhadap *TransactionDT* atau tanggal transaksi. Dari *scatter plot* pada Gambar 3, terlihat bahwa ada transaksi dengan nilai yang menyimpang jauh di atas transaksi lainnya di kisaran 30.000. Data tersebut dianggap pencilan, mengingat nominal transaksi lainnya berada di rentang 0-7.000.

2.3. Data *Preprocessing*

Data *preprocessing* dilakukan agar memperoleh data dengan format yang bersih dan konsisten. Tahapan yang dilakukan antara lain penanganan data hilang, penanganan data terduplikasi, penanganan pencilan, konversi ke data numerik, normalisasi data, dan pemisahan data latih dan data uji untuk digunakan ke dalam model.

Penanganan data hilang dilakukan dengan menghapus fitur dengan persentase data hilang di atas 99%. Selain itu juga dilakukan imputasi sesuai dengan tipe data untuk fitur yang masih memiliki data hilang. Fitur data numerik diisi dengan nilai -999 dan fitur data *object* atau teks diganti dengan nilai "*missing*" agar model tetap dapat membaca data yang hilang. Pengisian dengan nilai tertentu juga dimaksudkan untuk mempertahankan identitas suatu transaksi dalam satu baris data.



Gambar 3. Analisis Pencilan

Eksplorasi sebelumnya dari Gambar 5 menunjukkan ada tiga data yang terduplikasi. Penanganan terhadap duplikasi data dilakukan dengan menghapus masing-masing data yang terduplikasi yang berjumlah tiga data. Hal ini bertujuan untuk menghindari kebocoran data saat pemisahan data latih dan data uji.

Penanganan pencilan dilakukan dengan menghapus data yang dianggap pencilan. Hal ini dapat mempengaruhi hasil analisis data jika tidak ditangani terutama dalam proses normalisasi. Terdapat dua data dengan nominal transaksi bernilai lebih dari 30.000 yang mana kedua transaksi bukan termasuk kelas *fraud* sehingga tidak mewakili pola *fraud.*

Penyesuaian tipe data dilakukan dengan mengubah fitur bertipe *object* atau teks menjadi fitur numerik menggunakan *library LabelEncoder.* Hal ini bertujuan untuk menyesuaikan kebutuhan input *machine learning* dengan mengubah data *object* menjadi data numerik.

Semua data dinormalisasi dengan menggunakan *Z-score* untuk menyelaraskan nilai numerik di semua fitur. Fitur yang dinormalisasi akan mengubah rata-rata menjadi nol dan standar deviasinya bernilai satu. Normalisasi dibantu menggunakan *library Standarscaler* dari Python. Tujuannya agar setiap fitur memiliki bobot yang seimbang pada saat pelatihan model.

Dataset dibagi menjadi data latih untuk pelatihan model dan data uji untuk validasi model, dengan pembagian 80% untuk data latih dan 20% untuk data uji. Selain itu digunakan parameter *stratify* untuk memastikan bahwa proporsi sampel setiap kelas dipertahankan secara proporsional baik dalam data latih maupun data uji. Hasil akhir dari *preprocessing* yaitu pengurangan data sebanyak lima baris data dan sembilan kolom.

2.4. *Modelling*

2.4.1. *Logistic Regression*

Logistic regression dijadikan titik awal sebagai model *baseline* dalam eksperimen ini. Model ini dipilih karena kemudahannya dalam implementasi, interpretasi hasil, serta efisiensi komputasi, menjadikannya cocok untuk memberikan gambaran awal performa klasifikasi. *Logistic regression* bekerja dengan melakukan regresi dan *fitting* terhadap fungsi logistik dari sejumlah data pada dataset. *Logistic regression* masih digunakan secara luas dalam studi deteksi fraud karena sifatnya yang linier, sederhana, dan mudah dikendalikan, meskipun performanya cenderung terbatas ketika diterapkan pada data yang kompleks dan tidak seimbang (Ali et al., 2022).

2.4.2. XGBoost

XGBoost digunakan sebagai pendekatan lanjutan dalam eksperimen ini karena dikenal sebagai salah satu algoritma boosting paling andal dalam kompetisi data science. *XGBoost* bekerja dengan membangun *decision tree* yang kemudian dikoreksi pada setiap *error* yang muncul dari hasil prediksi.

Decision tree lama kemudian dikoreksi menjadi *decision tree* baru untuk kemudian semuanya dikumpulkan untuk memberikan prediksi akhir. Algoritma ini efektif dalam menangani kompleksitas data dan mendeteksi anomali, terutama pada data yang tidak seimbang, berkat mekanisme regularisasi tingkat lanjut dan struktur pohon berbasis *boosting* (Zhang et al., 2022).

Dalam implementasinya, model XGBoost dijalankan menggunakan parameter *default* dan penyetelan *hyperparameter* menggunakan *grid search*. Penyetelan terbatas pada tiga parameter utama: *learning_rate, max_depth,* dan *n_estimators*, menggunakan *grid search* dengan dua kombinasi nilai pada masing-masing parameter, menyesuaikan batas komputasi yang tersedia. XGBoost memerlukan waktu pelatihan yang lebih lama dan penggunaan memori yang lebih tinggi dibandingkan LightGBM, menjadikannya kurang efisien untuk skenario dengan kebutuhan komputasi minimal (Achary & Shelke, 2023).

2.4.3. LightGBM

LightGBM menjadi kandidat kuat untuk deteksi fraud karena dirancang untuk mencapai performa tinggi pada data berskala besar. Model ini menunjukkan efisiensi komputasi yang tinggi dengan hasil klasifikasi yang kompetitif dalam dataset besar seperti IEEE-CIS (Bakhtiari et al., 2023).

Model ini menggunakan *base algorithm* yang sama dengan XGBoost, yaitu *decision tree* yang diimprovisasi pada tiap kesalahan deteksi. Model ini menggunakan teknik *histogram-based boosting* yang membuat proses pelatihan lebih cepat dan efisien. Dalam eksperimen ini, parameter disesuaikan berdasarkan literatur dan hasil uji coba: *learning_rate=0,01, max_depth=12, n_estimators =10000, bagging_fraction=0,8,* dan *feature_fraction =0,4.*

2.4.4. *Multi-Layer Perceptron* (*MLP*)

Model MLP digunakan sebagai model pembanding berbasis jaringan saraf klasik dalam eksperimen ini. MLP dipilih karena MLP memungkinkan pembelajaran pola dari data yang kompleks, termasuk dari kolom-kolom yang saling berelasi pada dataset *fraud*. Hasil yang diperoleh Hernandez Aros et al. (2024), menyatakan bahwa MLP merupakan salah satu model yang banyak digunakan dalam studi kasus deteksi fraud di bidang keuangan karena mengenali pola non-linear dan kompleks.

MLP merupakan jaringan saraf bertipe *feedforward* yang terdiri dari lapisan input, dua *hidden layer* (masing-masing 256 dan 128 neuron), dan satu lapisan output dengan fungsi aktivasi sigmoid untuk klasifikasi biner. Pada penelitian ini, fungsi aktivasi ReLU digunakan pada lapisan tersembunyi, dan algoritma optimasi Adam digunakan dengan *learning_rate* sebesar 0,001 dan *batch_size* sebesar 1024.

Studi yang dilakukan Chellapilla et al. (2024) menunjukkan bahwa penggunaan MLP sebagai *meta-learner stacking ensemble* dapat mengurangi *false positive* dibandingkan model tunggal. MLP dapat meningkatkan kinerja hasil prediksi terhadap data yang tidak seimbang serta relevan untuk dataset berdimensi tinggi, yaitu 280 ribu transaksi.

2.4.5. *Stacking Ensemble*

Arsitektur *stacking ensemble* menggunakan sejumlah model dasar dalam klasifikasi yang kemudian diteruskan ke *meta-learner* untuk melakukan prediksi akhir. Arsitektur ini dipilih karena memungkinkan pemilihan ragam model dan konfigurasinya, kinerja yang relatif lebih baik, dan dapat mengoreksi kesalahan prediksi yang mungkin dibuat oleh *base learner model*. Dalam penelitian ini, tiga model dasar digunakan sebagai *base learners*, yaitu *Logistic regression, XGBoost,* dan *LightGBM,* yang masing-masing mewakili model linier dan *boosting*. Prediksi dari ketiga model dasar tersebut kemudian menjadi input bagi meta-learner berupa MLP, yang bertugas melakukan klasifikasi akhir. Pemilihan tersebut didasarkan pada kemampuannya dalam menggeneralisasi dengan baik dan efisiensi komputasi.

Pendekatan *stacking* terbukti efektif karena mampu menangkap kompleksitas data melalui kombinasi kekuatan dari model-model yang memiliki arsitektur berbeda. Talukder et al. (2024) menyatakan bahwa integrasi model-model *ensemble* dalam skema bertingkat mampu meningkatkan akurasi deteksi penipuan secara signifikan, terutama pada dataset besar yang tidak seimbang seperti transaksi keuangan. Selain itu, (Suganya et al., 2023) menjelaskan bahwa *stacking ensemble* mampu mengurangi risiko *overfitting* dan meningkatkan *robustness* model, karena menggabungkan keputusan dari berbagai model yang memiliki kekuatan dan kelemahan yang saling melengkapi. Kombinasi model dalam *stacking ensemble* terbukti mampu mengintegrasikan keunggulan prediksi dari model *boosting* dan linier, sehingga meningkatkan kemampuan diskriminasi terhadap pola yang kompleks (Suganya et al., 2023; Talukder et al., 2024).

2.5. Evaluasi Model

Evaluasi model dilakukan untuk mengukur setiap algoritma dalam mendeteksi transaksi fraud pada data historis yang tidak seimbang. Metrik utama yang digunakan adalah *Area Under Curve - Receiver Operating Characteristic* (AUC-ROC), karena metrik ini mampu menilai performa klasifikasi pada berbagai ambang batas dan lebih representatif dalam konteks data tidak seimbang dibandingkan matriks akurasi.

Selain AUC-ROC, metrik lain yang turut dihitung adalah *precision* dan *recall* untuk memberikan evaluasi komprehensif terhadap kemampuan model dalam mengidentifikasi kelas minoritas (*fraud*). Evaluasi menggunakan *precision* dan *recall* terhadap model terbaik yang didapat dengan *confusion matrix* terhadap hasil model terbaik. Dalam konteks deteksi *fraud, confusion matrix* memiliki informasi berikut:

- *True Positive* (TP): data berlabel *fraud* dan terdeteksi model sebagai *fraud*
- *False Positive* (FP): data berlabel *fraud* dan terdeteksi model bukan *fraud*
- *False Negative* (FN): data berlabel bukan *fraud* dan terdeteksi model sebagai *fraud*
- *True Negative* (TN): data berlabel bukan *fraud* dan terdeteksi model bukan *fraud*

Analisis selanjutnya dilakukan dengan menghitung *precision* dan *recall. Precision* dapat dihitung dengan rumus pada persamaan (1). Menurut (Ileberi et al., 2021) *precision* digunakan untuk melihat rasio prediksi benar positif (*fraud*) terhadap semua prediksi positif, sementara *recall* dihitung untuk mengukur kemampuan model mengidentifikasi semua kasus *fraud* yang sebenarnya dengan persamaan (2).

$$Precision = \frac{TP}{TP + TN} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

3. HASIL DAN PEMBAHASAN

Eksperimen dilakukan menggunakan data yang sudah dilakukan *preprocessing* dimana data awal memiliki dimensi 590.540 baris dan 434 kolom menjadi 590.535 dan 425 kolom. Pembagian dataset menjadi data *train* dan *data test* dilakukan secara acak dengan *random seed*.

Eksperimen dilakukan menggunakan model *Logistic regression, XGBoost, LightGBM,* dan *stacking ensemble* menggunakan *base learner XGBoost* dan *LightGBM* serta *meta-learner* MLP. *Logistic regression, XGBoost,* dan *LightGBM* dilakukan dua eksperimen, yaitu dengan nilai *hyperparameter default* dan *tuning hyperparameter.*

Hyperparameter XGBoost di Tabel 7 didapatkan dari hasil pencarian terbaik menggunakan strategi *grid search* dengan nilai [0,1, 0,3] untuk *learning_rate,* [6, 9] untuk *max_depth,* dan [100, 200] untuk *n_estimators.* Selain yang disebutkan di Tabel 4, *XGBoost* menggunakan nilai *default library xgboost. XGBoost* dilakukan eksperimen dengan dan tanpa *tuning hyperparameter.* Model juga digunakan untuk *stacking* tetapi hanya *XGBoost* dengan *tuning hyperparameter.*

Hyperparameter untuk model LightGBM didapatkan dengan menggunakan hasil *tuning* dari (Ge et al., 2020) seperti pada Tabel 8. Selain dari yang disebutkan pada Tabel 8 menggunakan nilai *default* dari *library lightgbm.* Sama seperti *XGBoost,* eksperimen dilakukan dengan nilai *default* dan dengan *tuning hyperparameter.* LightGBM dengan *tuning* juga digunakan sebagai *base learner* dalam *stacking ensemble.*

Tabel 7. *Hyperparameter XGBoost*

nama	deskripsi	nilai
learning_rate	Besar pembaharuan bobot dilakukan setiap iterasi	0,1
max_depth	Jumlah maksimal pohon	9
n_estimators	Jumlah estimator	200

Tabel 8. *Hyperparameter LightGBM*

nama	deskripsi	nilai
learning_rate	Besar pembaharuan bobot dilakukan setiap iterasi	0,01
max_depth	Jumlah maksimal pohon	12
n_estimators	Jumlah estimator	10000
bagging_fraction	Data yang di-sampling acak sebelum membuat setiap pohon	0,8
feature_fraction	Fitur yang di-sampling acak sebelum membuat setiap pohon	0,4
boosting_type	Algoritma *boosting*	gbdt

Tabel 9. Hasil Evaluasi Model

model	nilai AUC-ROC
Logistic regression	0,8359
Logistic regression tuning	0,8369
XGBoost	0,9419
XGBoost tuning	0,9600
LightGBM	0,9312
LightGBM tuning	0,9661
Stacking ensemble	**0,9663**

Evaluasi dilakukan terhadap 20% dari dataset yang sudah dibagi secara proporsional untuk kelas data *fraud* dan bukan menggunakan *stratify* saat melakukan *split data training*. Berdasarkan hasil yang disajikan di Tabel 9, model *stacking ensemble* menjadi model terbaik dalam mendeteksi *fraud* untuk data transaksi finansial dengan nilai AUC-ROC 0,9663 yang tidak berbeda jauh dengan LightGBM dengan *tuning,* dengan perbedaan 0,0002 saja. Eksperimen ini sejalan dengan hasil dari penelitian (Ge et al., 2020) dimana LightGBM menjadi model kinerja terbaik dengan nilai AUC-ROC 0,955. Kecepatan inferensinya menjadikan LightGBM cocok untuk deployment real-time di sistem yang harus memproses data dalam jumlah besar secara cepat (Ge et al., 2020).

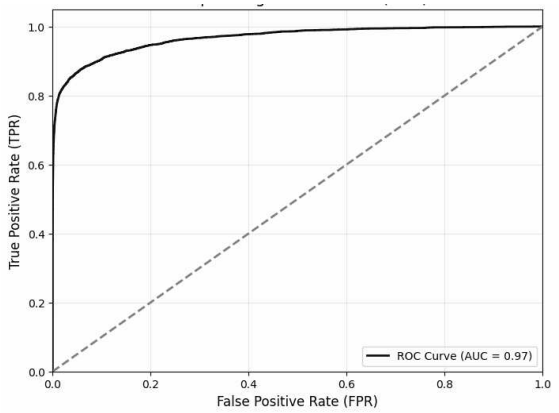
Sementara itu, hasil klasifikasi *logistic regression* berbeda cukup jauh dibandingkan model lain. Hal ini karena ketika eksperimen dengan nilai *default*, model sudah mencapai iterasi maksimal sementara model belum konvergen. *Tuning* dilakukan dengan mengubah nilai *max_iter* 10.000 untuk memastikan proses pelatihan mencapai konvergensi. Model mencapai konvergensi tetapi nilai AUC-ROC sedikit peningkatannya. Saat digunakan pada dataset dengan dominasi kelas mayoritas (non-fraud), *logistic regression* menunjukkan keterbatasan dalam mendeteksi kelas minoritas. Akurasinya moderat, dan

sensitivitas terhadap kelas *fraud* rendah, menjadikannya kurang ideal untuk *deployment* dalam skenario nyata yang memerlukan deteksi anomali secara lebih presisi (Patel et al., 2024).

Stacking ensemble menurut (Airlangga, 2024) dapat meminimalisir *false positive* dan *false negative* dimana penting dalam mendeteksi *fraud.* Eksperimen *stacking ensemble* yang dilakukan menggabungkan dua model yang tahan terhadap data yang tidak seimbang sehingga dari *base learner*, *stacking ensemble* dapat memadukan kelebihan dari kedua model tersebut. Selain itu MLP sebagai *meta learner* juga berperan meningkatkan hasil. Selain itu Bakhtiari et al. (2023) menunjukkan bahwa pendekatan ensemble berbasis boosting seperti XGBoost dan LightGBM memberikan kinerja yang unggul dalam mendeteksi fraud pada dataset IEEE-CIS, dibandingkan model neural network klasik seperti MLP.

Stacking ensemble memiliki kompleksitas tinggi mengingat *base learner* yang digunakan sudah cukup kompleks dengan *tuning hyperparameter* setiap model dalam *base learner* sehingga dapat memberikan evaluasi terbaik dibandingkan model lain. *Stacking ensemble* membutuhkan perangkat komputasi yang mumpuni sebagai akibat dari pembangunan model yang kompleks dan waktu pelatihan yang lebih lama. Selain itu, dampak dari kompleksitas model juga menyebabkan interpretabilitas dari hasil model lebih sulit dijelaskan, sesuai dengan hasil penelitian sebelumnya (Airlangga, 2024).

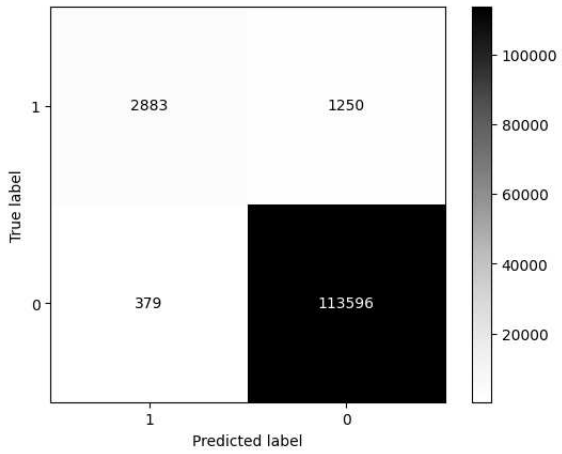
Model terbaik memberikan klasifikasi *fraud* dengan baik. Dari kurva Gambar 4, terlihat bahwa kurva ROC yang ditunjukkan warna biru berada jauh dari garis diagonal sebagai *baseline* dari AUC ROC. Semakin model mendekati angka 1, maka semakin baik model melakukan generalisasi dan semakin model mendekati angka 0,5 model semakin mengindikasikan tidak dapat melakukan generalisasi.



Gambar 4. Grafik kurva ROC

Evaluasi selanjutnya dilakukan dengan analisis *confusion matrix* model terbaik terhadap 118.108 data uji. Hasilnya seperti yang terlihat pada Gambar 5, didapatkan nilai TP 2.883, FP 379, FN 1.250, dan TN 113.596. Nilai tersebut bisa digunakan untuk menghitung dan menganalisis kemampuan model terbaik.

Dari persamaan (1) dan (2), didapatkan *precision* dan *recall* berturut-turut 88,4% dan 69,8%. Artinya, model terbaik dapat mendeteksi *fraud* sebanyak 88,4% dari seluruh *fraud*. Sementara itu, dari nilai *recall* memberikan bahwa dari seluruh data, model terbaik mampu mendeteksi *fraud* sebesar 69,8%.



Gambar 5. *Confusion matrix* model terbaik

4. KESIMPULAN DAN SARAN

Penelitian bertujuan untuk mendapatkan hasil evaluasi model *stacking ensemble* dibandingkan model lain, dengan menggunakan MLP sebagai *meta-learner*. Hasil *preprocessing* tidak memberikan perubahan signifikan karena hanya mengurangi data sebanyak lima baris dan sembilan kolom. Dibandingkan dengan jumlah datanya, yaitu 590.540 baris dan 434 kolom, perubahan datanya terlalu kecil.

Hasil menunjukkan bahwa *stacking* memberikan hasil evaluasi terbaik untuk mendeteksi *fraud* dalam transaksi finansial dengan nilai AUC-ROC 0,9663. Artinya, model *machine learning* dapat membedakan kelas dengan baik. Meskipun demikian, kebutuhan komputasi yang lebih tinggi untuk mengembangkan model *Stacking* tidak sebanding dengan hasil peningkatan yang tidak signifikan. *Hyperparameter* berperan penting dalam peningkatan nilai AUC-ROC. Oleh karena itu, dalam mengembangkan sistem deteksi *fraud* sebaiknya memberikan sumber daya lebih untuk melakukan *fine-tuning* model dibandingkan mengadopsi model kompleks seperti *stacking*, kecuali jika peningkatan kecil dalam kinerja model benar-benar dibutuhkan.

Secara umum, penelitian ini masih memiliki keterbatasan dalam eksplorasi data dan *preprocessing* sehingga penelitian selanjutnya dapat mempelajari pengaruh dari *preprocessing* lain terhadap hasil evaluasi seperti pembentukan fitur baru dari hasil kombinasi fitur yang ada. Penentuan transaksi *fraud* juga dapat dilakukan dengan pendekatan identifikasi fitur pelaku *fraud* mengingat dataset juga

memberikan fitur terkait identitas pelaku transaksi. Selain itu *tuning hyperparameter* XGBoost juga masih bisa ditingkatkan dengan variasi rentang nilai dan *hyperparameter* lebih banyak seperti pada eksperimen LightGBM.

Hasil *stacking* tidak memberikan peningkatan yang signifikan membuktikan bahwa penggunaan model dengan *tuning* seperti *XGBoost* atau LightGBM saja sudah bagus digunakan sebagai model tunggal. Terlebih jika terdapat keterbatasan perangkat komputasi untuk pelatihan model mengingat eksperimen model *stacking ensemble* menggunakan base model XGBoost dan LightGBM dengan *hyperparameter* yang sama dengan kedua model tunggal, dimana hasil yang dihasilkan tidak berbeda jauh dengan model tunggal.

Hasil dari *confusion matrix* menunjukkan bahwa model menunjukkan kemampuan baik dan mampu meminimalisir *false negative.* Akan tetapi, *recall* 69,8% mengindikasikan bahwa 30,2% data *fraud* tidak terdeteksi. *Recall* yang rendah dapat menyebabkan kerugian langsung ke perusahaan karena banyak transaksi *fraud* yang terlewat, sedangkan *precision* rendah dapat menyebabkan banyaknya komplain dari pengguna karena banyak yang terdeteksi sebagai transaksi *fraud* meskipun sebenarnya bukan. Perusahaan perlu memikirkan dampak mana yang menimbulkan kerugian lebih besar.

DAFTAR PUSTAKA

ACHARY, R., & SHELKE, C. J. 2023. Fraud detection in banking transactions using machine learning. *Proceedings of IITCEE 2023*, 1–6. https://doi.org/10.1109/IITCEE57236.2023.10091067

AIRLANGGA, G. 2024. A hybrid ensemble approach for enhanced fraud detection: Leveraging stacking classifiers to improve accuracy in financial transaction. *Journal of Computer System and Informatics (JoSYC)*, *5*(4), 1118–1127. https://doi.org/10.47065/josyc.v5i4.5840

ALI, A., ABD RAZAK, S., OTHMAN, S. H., & EISA, T. A. E. 2022. Financial fraud detection based on machine learning: A systematic literature review. *Applied Sciences, 12*(19), 9637. https://doi.org/10.3390/app12199637

BAKHTIARI, S., NASIRI, Z., AND VAHIDI, J. 2023 Credit card fraud detection using ensemble data mining methods. Multimedia Tools and Applications, vol. 82, pp. 29057–29075, Aug. 2023. https://doi.org/10.1007/s11042-023-14698-2

CHELLAPILLA, V., CHIKKAM, S., MELINATI, J. S., & EZHILARASAN, M. 2024. Credit card fraud detection using a stacking ensemble approach with LSTM and random forest machine learning techniques. International Education & Research Journal, 10(4), 16–19.

GE, D., GU, J., CHANG, S., & CAI, J. 2020. Credit card fraud detection using LightGBM model. In 2020 International Conference on E-Commerce and Internet Technology (ECIT) (pp. 232–236). IEEE. https://doi.org/10.1109/ECIT50008.2020.00060

HERNANDEZ AROS, L., BUSTAMANTE MOLANO, L. X., GUTIERREZ-PORTELA, F., MORENO HERNANDEZ, J. J., AND RODRÍGUEZ BARRERO, M. S. 2024. Financial fraud detection through the application of machine learning techniques: A literature review. Humanities and Social Sciences Communications, vol. 11, art. 1130, Sep. 2024. https://doi.org/10.1057/s41599-024-03606-0

ILEBERI, E., SUN, Y., & WANG, Z. 2021. Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost. IEEE Access, 9, 165286–165294. https://doi.org/10.1109/ACCESS.2021.3134330

KAGGLE. 2019. IEEE-CIS fraud detection. https://www.kaggle.com/competitions/ieee-fraud-detection/data

LAXMAN, V., AHMAD, A., & PANDA, S. K. 2025. Emerging threats in digital payment and financial crime. Journal of Financial Innovation and Digital Threats, vol. 3, no. 2, pp. 45–60, 2025. https://doi.org/10.1016/j.jdec.2025.04.002

MASTERCARD. 2024. *2024 global fraud report*. https://www.mastercard.com/global-fraud-report-2024

Otoritas Jasa Keuangan. 2024. *POJK No. 17/POJK.03/2023 tentang Strategi Anti-fraud*. Jakarta: OJK.

PATEL, S., PANDEY, M., & R., D. 2024. fraud detection in financial transactions: A machine learning approach. *Proceedings of ICONSTEM 2024*, 1–6. https://doi.org/10.1109/ICONSTEM60960.2024.10568903

SUGANYA, S. S., NISHANTH, S., & MOHANADEVI, D. 2023. Ensemble learning approaches for fraud detection in financial transactions. In *Proceedings of the 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS-2023)*. IEEE. https://doi.org/10.1109/ICACRS58579.2023.10404382

TALUKDER, M. A., KHALID, M., & UDDIN, M. A. 2024. An integrated multistage ensemble machine learning model for fraudulent transaction detection. *Journal of Big Data, 11*, Article 168. https://doi.org/10.1186/s40537-024-00996-5

ZHANG, P., JIA, Y., & SHANG, Y. 2022. Research and application of XGBoost in imbalanced data. *International Journal of Distributed Sensor Networks, 18*(6). https://doi.org/10.1177/15501329221106935