Penerapan Extreme Gradient Boosting (XGBoost) dengan SMOTE untuk Deteksi Penipuan Kartu Kredit

Nicholas Anthony Suhartono[#1], Ventje J. Lewi Engel, M.T., CEH[*2]

[#]*Program Studi Informatika, Institut Teknologi Harapan Bangsa*
Jalan Dipatiukur No. 80-84, Bandung, Indonesia 40132
[1]anthony.nicholas84@gmail.com
[2]ventje@ithb.ac.id

Abstract*— Credit card is one of many payment methods that are commonly used globally. Credit card fraud is a fraud that involves stealing data from credit card owners and using it to make transactions. This type of fraud has significantly increased and resulted in substantial financial losses. The credit card fraud dataset typically includes imbalance classes because data labelled as fraud rarely occurs within normal transactions and becomes a challenge for machine learning classification algorithms. In this research, Extreme Gradient Boosting (XGBoost) will be implemented for classification along with Synthetic Minority Oversampling Technique (SMOTE) to oversample minority classes. K-fold Cross-validation and Grid Search are implemented to find the best XGBoost hyperparameter which is sensitive enough to detect a fraud transaction. Based on the test results, the highest recall that XGBoost achieved is 85,26% when hyperparameter n_estimators = 50, max_depth = 5, eta = 0,3, lambda = 0,5, gamma = 0,5.

***Keywords*— Classification, Oversampling, Credit Card Fraud, Imbalance, Extreme Gradient Boosting (XGBoost), Synthetic Minority Oversampling Technique (SMOTE)**

Abstrak*— Kartu kredit merupakan metode pembayaran yang umum digunakan oleh masyarakat. Kejahatan Penipuan kartu kredit adalah bentuk kejahatan penipuan yang melibatkan metode pembayaran kartu kredit yang terjadi ketika pelaku kejahatan menggunakan data pemilik kartu kredit untuk melakukan transaksi. Kejahatan ini mengalami peningkatan yang signifikan setiap tahunnya dan mengakibatkan kerugian yang cukup besar. Dataset penipuan kartu kredit memiliki jumlah data antar kelas yang sangat tidak seimbang (imbalance) sehingga menjadi salah satu masalah untuk melakukan pelatihan pada model machine learning. Pada penelitian ini, akan dilakukan klasifikasi menggunakan *Extreme Gradient Boosting* (XGBoost). Untuk mengatasi masalah data latih yang terbatas, akan dilakukan oversampling menggunakan metode *Synthetic Minority Oversampling Technique* (SMOTE). Proses pelatihan akan menggunakan metode *k-fold cross-validation* serta *grid search* untuk mencari *hyperparameter* terbaik yang cukup sensitif untuk mendeteksi penipuan kartu kredit. Untuk mendapatkan nilai sensitivitas tertinggi, maka pengujian akan menggunakan *confusion matrix* sebagai pengukuran khususnya untuk mencari nilai *recall* tertinggi. Berdasarkan hasil pengujian yang dilakukan, nilai *recall* tertinggi yang diperoleh sebesar 85,26% yang diperoleh dengan menggunakan *hyperparameter n_estimators* = 50, *max _depth* = 5, *eta* = 0,3, *lambda* = 0,5, dan *gamma* = 0,5.

***Kata Kunci*— Klasifikasi, Oversampling, Penipuan Kartu Kredit, Imbalance, Extreme Gradient Boosting (XGBoost), Synthetic Minority Oversampling Technique (SMOTE)**

I. Pendahuluan

Kartu kredit merupakan salah satu metode pembayaran yang umum di dunia finansial maupun bisnis. Penipuan kartu kredit adalah kejadian dimana penipu menggunakan identitas kartu kredit milik orang lain untuk melakukan transaksi. Jumlah penipuan transaksi kartu kredit meningkat setiap tahunnya, khususnya di Amerika Serikat. Pada tahun 2014, terjadi 55.553 penipuan kartu kredit, sedangkan pada tahun 2018, jumlahnya meningkat menjadi 157.688 kasus penipuan. Jumlah kerugian yang diakibatkan oleh transaksi kartu kredit di seluruh dunia pada tahun 2018 sangat besar, yaitu sebesar 24,26 miliar dollar Amerika Serikat. Penipuan kartu kredit dapat terjadi ketika penipu menggunakan informasi palsu untuk melakukan transaksi kartu kredit, dan pihak penerbit kartu kredit menerima transaksi tersebut atau ketika kartu kredit diterbitkan dengan benar namun transaksi melibatkan aktivitas yang bersifat curang atau penipuan. Deteksi penipuan kartu kredit memiliki tantangan tersendiri, karena dari banyaknya transaksi kartu kredit yang terjadi, hanya beberapa transaksi yang dapat dikategorikan kedalam penipuan. Oleh karena itu, diperlukan sebuah sistem yang sensitif untuk mendeteksi transaksi penipuan kartu kredit [1], [2], [3], [4], [5].

II. Metodologi

A. Extreme gradient boosting (XGBoost)

Extreme gradient boosting (XGBoost) merupakan algoritme yang diciptakan oleh Tianqi Chen dari Universitas Washington. XGBoost tersedia dalam bentuk pustaka *python* yang dirilis pada tanggal 27 Maret 2014. Semula, XGBoost berawal dari proyek penelitian. Namun pada tahun 2015

XGBoost menjadi solusi paling dominan untuk menyelesaikan masalah klasifikasi dan regresi. Pada kompetisi yang diselenggarakan oleh Kaggle pada tahun 2015 yang terdapat 29 tantangan, 17 solusi diantaranya menggunakan XGBoost. Begitu juga dengan kompetisi yang diselenggarakan oleh KDDCup pada tahun 2015, XGBoost digunakan oleh setiap tim dalam peringkat 10 besar [6]. XGBoost diciptakan menggunakan prinsip *gradient boost* yang menggabungkan *weak learner* dengan *strong learner*. *Gradient-boosted tree* pada umumnya dibangun secara sekuensial, sedikit demi sedikit memperbaiki hasil prediksi di iterasi selanjutnya, tetapi XGBoost mampu membangun *tree* secara paralel. XGBoost memiliki performa prediksi lebih tinggi dengan mengendalikan kompleksitas model dan mengurangi *overfitting* melalui regularisasi (*regularization*).

B. Resampling

Data Imbalance merupakan istilah bagi dataset yang memiliki distribusi kelas label yang tidak seimbang, yaitu ketika salah satu label memiliki jumlah observasi yang sangat sedikit dibandingkan dengan label lainnya yang memiliki jumlah yang sangat besar. Untuk menyeimbangkan masing-masing label dapat menggunakan teknik *resampling*:

- Undersampling, yaitu mengurangi data kelas yang paling dominan atau kelas mayoritas
- *Oversampling* yaitu meningkatkan jumlah data kelas minoritas. Salah satu teknik yang paling populer untuk melakukan oversampling adalah Synthetic Minority Oversampling Technique (SMOTE).

C. Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) merupakan salah satu teknik statistik *oversampling* untuk meningkatkan jumlah data pada *dataset* untuk menangani masalah data *imbalance*. SMOTE membuat data baru dari data kelas minoritas sebagai masukan sementara data kelas mayoritas jumlahnya tidak berubah. Data baru yang dihasilkan SMOTE bukan hanya merupakan duplikat dari data minoritas, tetapi algoritme SMOTE mengambil sampel dari ruang fitur untuk setiap target kelas terhadap tetangga terdekatnya, dan membuat sampel baru yang menggabungkan fitur dari kelas target dengan fitur tetangganya [7].

D. K-fold Cross Validation

Ketika melatih dan menguji model menggunakan data yang sama, model mampu memprediksi dengan baik pada tahap pengujian, tetapi terdapat kemungkinan model mendapatkan performa buruk ketika memprediksi data baru. Hal ini disebut juga dengan *overfitting*. Untuk mengatasi ini, *dataset* bisa dibagi menjadi 2 bagian, data latih dan data uji. Namun, pada saat mencari *hyperparameter* terbaik, ada resiko model akan *overfitting* pada data uji ketika melakukan *hyperparameter tuning* [8]. Jika membagi *dataset* menjadi 3 bagian dengan menggunakan data validasi, data latih yang digunakan untuk melatih model akan semakin sedikit. Salah satu solusi untuk mengatasi masalah ini adalah menggunakan *k-fold cross validation*.

K-fold cross-validation atau *cross-validation* adalah salah satu metode validasi model dalam *machine learning*. *Cross-validation* digunakan apabila ingin menggunakan lebih banyak data dalam *dataset* untuk melatih model, dalam hal ini, melakukan *data splitting* menjadi data latih (*train set*) dan data uji (*test set*), sehingga tidak lagi diperlukan data validasi (*validation set*). *Cross-validation* kemudian digunakan pada data latih untuk mensimulasikan data validasi (*validation set*).

E. Hyperparameter Tuning

Hyperparameter tuning adalah proses pencarian kombinasi nilai *hyperparameter* model yang mampu menghasilkan performa yang baik. Ada beberapa metode yang digunakan dalam proses *hyperparameter tuning*, salah satunya adalah *grid search* [9].

Grid search merupakan metode yang paling mudah untuk diterapkan. Metode ini yaitu mencoba seluruh kombinasi dari *hyperparameter* kemudian menggunakan data latih untuk melatih model dan menilai performa model menggunakan data validasi. Kelebihan dari *grid search* adalah mencoba setiap kombinasi dari *hyperparameter* sehingga tentu akan menemukan kombinasi *hyperparameter* yang menghasilkan model dengan hasil evaluasi terbaik. Salah satu kelemahan dari *grid search* adalah waktu pemrosesannya yang lama, khususnya ketika jumlah data yang digunakan sangat banyak [9].

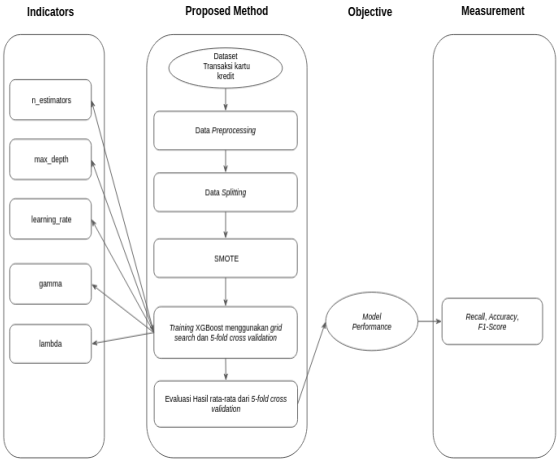
III. Perancangan Sistem

1) Kerangka Pemikiran

Gambar 1 menunjukkan kerangka pemikiran dari metode yang diusulkan untuk melakukan deteksi transaksi penipuan kartu kredit. Satuan ukur untuk mengukur hasil penelitian adalah *confusion matrix* (*recall, accuracy, f-score)*.

Berikut merupakan beberapa indikator yang akan diuji:

1. *n_estimators* merupakan banyaknya tree yang akan dibentuk oleh XGBoost
2. *max_depth* merupakan kedalaman maksimum tree yang dapat dibentuk oleh XGBoost.
3. *Learning rate* atau eta (η): Merupakan besaran nilai yang harus diambil untuk meminimalkan loss function. Pada XGBoost, Nilai ini akan digunakan sebagai nilai scaling pada nilai output setiap tree. Learning rate yang terlalu besar mengakibatkan model akan *overshoot*, jika nilainya terlalu kecil maka proses pelatihan model akan semakin lama.
4. *gamma* (γ) merupakan konstanta yang mencegah pembangunan tree terlalu dalam sehingga mencegah *overfitting.*
5. *lambda* (λ) merupakan konstanta regularisasi pada XGBoost yang berperan dalam mencegah *overfitting*.



Gambar 1 Kerangka Pemikiran

2) Flowchart Global

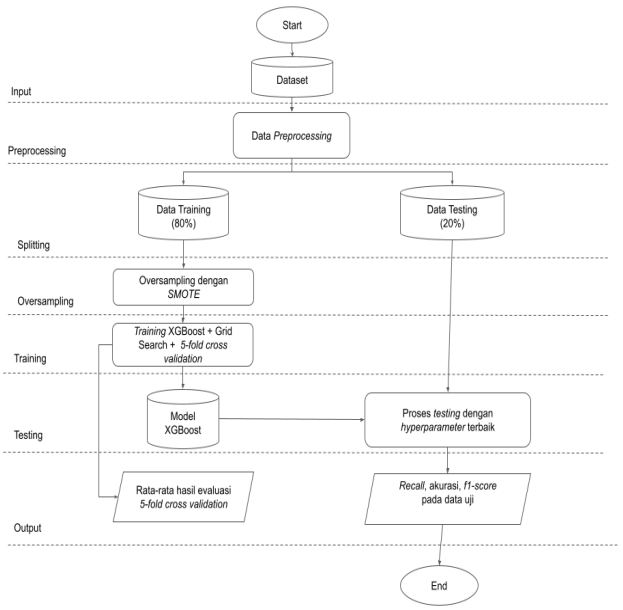
Dalam sistem deteksi transaksi penipuan kartu kredit terdapat beberapa proses, yang berisi data *preprocessing*, data *splitting*, penerapan teknik *oversampling* menggunakan SMOTE pada data latih, proses training XGBoost menggunakan *5-fold cross validation* dan *grid search*, evaluasi rata-rata hasil metrik *5-fold cross-validation*, memilih hyperparameter terbaik untuk selanjutnya dilakukan proses pengujian pada data uji, dan evaluasi menggunakan *confusion matrix*.

Berikut adalah uraian proses global yang dilakukan dalam penelitian ini yang terlihat pada Gambar 2:

1. Masukan berupa file CSV berisi data transaksi kartu kredit [10]. *File* CSV akan disimpan kedalam bentuk *DataFrame*, yang kemudian akan dilakukan *data preprocessing* untuk standarisasi data. Tahap data *preprocessing* menghilangkan *missing value* pada data dan menghilangkan data yang duplikat.
2. Data *splitting* membagi data menjadi data latih dan data uji. Data uji akan digunakan sebagai data *hold out* yang berarti data uji akan digunakan sebagai evaluasi terakhir dan tidak akan digunakan dalam proses training dan *hyperparameter tuning*.
3. Menerapkan teknik *oversampling* SMOTE pada data latih untuk memperbanyak data dengan label *fraud* (*Class* = 1).
4. Proses pelatihan pada XGBoost akan menggunakan *5-fold cross validation* dengan menggunakan kombinasi *hyperparameter* hasil dari *grid-search*.
5. Memilih *hyperparameter* terbaik berdasarkan hasil rata-rata evaluasi *5-fold cross-validation*. Model XGboost akan dibentuk kembali menggunakan *hyperparameter* terbaik berdasarkan rata-rata evaluasi *5-fold cross-validation* untuk dilakukan proses pengujian menggunakan data uji yang telah ditetapkan sebagai data *hold out* untuk evaluasi akhir, dengan hasil pengukuran menggunakan *confusion matrix* untuk mengetahui *precision*, *recall*, *accuracy*, dan *f-score*.
6. Keluaran berupa XGBoost dengan *hyperparameter* yang memiliki rata-rata evaluasi *5-fold cross-validation* terbaik dan waktu pemrosesan terbaik serta hasil pengujian berupa *confusion matrix* pada data uji sebagai data *hold out*.

3) Dataset

Dalam penelitian ini, dataset berupa 1 file dengan format *Comma Separated Value* (CSV). Fitur yang digunakan dalam penelitian adalah fitur Time, V1-V28, *Amount*, dan *Class* karena seluruh fitur tersebut memuat data penting mengenai transaksi yang terjadi. Data yang akan dipakai dari dataset sejumlah 47773 dengan 47300 diantaranya merupakan transaksi non-fraud dan 473 diantaranya merupakan transaksi *fraud*, sehingga *dataset* ini sangat *imbalance* dengan rasio sekitar 1% transaksi penipuan dari dataset yang akan dipakai. Dalam penelitian ini, *dataset* akan dibagi menjadi 20% data uji dan 80% data latih. Data latih terdiri dari 38.218 data dengan 37.840 diantaranya merupakan data transaksi bukan penipuan dan 378 diantaranya merupakan data transaksi penipuan. Data uji terdiri dari sebanyak 9.555 data dengan 9.460 diantaranya merupakan data transaksi bukan penipuan dan 95 diantaranya merupakan data transaksi penipuan.



Gambar 2 Flowchart Global

IV. Hasil dan Pembahasan

Tabel I
Kombinasi Hyperparameter yang akan diuji

n_estimator	max_depth	eta	lambda	gamma
50	5	0,1	0	0,5
100	10	0,3	0,5	1
200	15	-	-	-

Pengujian dilakukan akan menguji 5 *Hyperparameter* XGBoost, diantaranya adalah *n_estimator, max_depth, eta, lambda,* dan *gamma*. Proses pelatihan *5-fold cross validation* dan metode *grid search* untuk mencari kombinasi *hyperparameter* terbaik. Dengan menggunakan *grid search* maka akan dihasilkan 72 kombinasi *hyperparameter* XGBoost yang akan diuji. Kombinasi *hyperparameter* yang diuji dapat dilihat pada Tabel I. Metode *oversampling* SMOTE diterapkan untuk meningkatkan rasio jumlah data kelas positif hingga perbandingan 2 banding 3 terhadap kelas mayoritas atau kelas negatif.

A. Pengujian Utama dengan Oversampling SMOTE

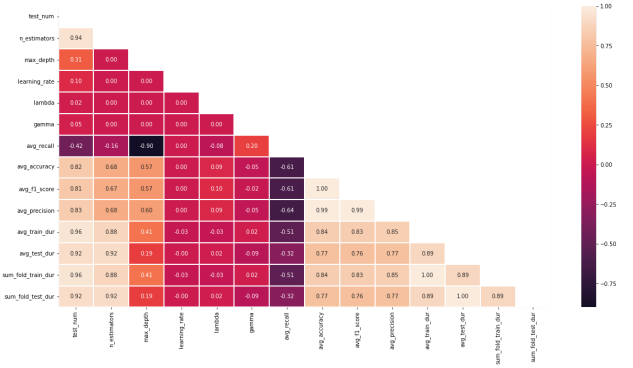
Berdasarkan Tabel II yang menunjukkan hasil *5-fold cross validation,* nilai *recall* terbaik diperoleh oleh kombinasi pengujian nomor 7 yang terjadi ketika *hyperparameter n_estimators = 50, max_depth = 5, eta = 0,3, lambda = 0,5, gamma = 0,5* dengan nilai *recall* tertinggi sebesar 85,26%. Nilai *f-score* dan akurasi terbaik diperoleh oleh kombinasi pengujian nomor 58 dan kombinasi pengujian nomor 62 dengan nilai *f-score* 87,05% dan nilai akurasi sebesar 99,75% yang terjadi ketika *hyperparameter n_estimators = 200, max_depth = 10, eta = 0,3, lambda = 0, gamma = 1.*

Tabel II
Hasil 5-Fold Cross Validation

No	Recall	Accuracy	F-score
7	0,8598	0,9928	0,7034
62	0,8439	0,9975	0,8705

Tabel III
Hasil Pengujian Dengan Oversampling SMOTE

No	Recall	Accuracy	F-score
7	0,8526	0,9909	0,6506
62	0,8421	0,9977	0,8791



Gambar 3 Matriks Korelasi Hasil Pengukuran Pengujian dengan *Hyperparameter* yang diuji

Selanjutnya dilakukan pengujian terhadap data uji menggunakan kombinasi *hyperparameter* nomor 7 dan 62. Tabel III yang menunjukkan hasil pengujian XGBoost pada data uji, nilai *recall* tertinggi diperoleh sebesar 85,25% pada kombinasi *hyperparameter* nomor 7. Nilai akurasi tertinggi diperoleh oleh kombinasi nomor 62 sebesar 99,77%. Nilai *f-score* tertinggi diperoleh sebesar 87,91% oleh kombinasi hyperparameter nomor 62.

Gambar 3 merupakan matriks korelasi antara hasil pengukuran pengujian dengan *hyperparameter* yang diuji. Berdasarkan gambar tersebut, terlihat bahwa nilai *recall* (*avg_recall*) memiliki korelasi yang negatif dengan *hyperparameter max_depth* dan *n_estimators*, dan korelasi positif dengan *hyperparameter gamma*. *Lambda* memiliki korelasi yang sedikit negatif dengan *recall* tetapi memiliki korelasi positif dengan *f-score*. Dapat dilihat juga bahwa ketika nilai *recall* memiliki korelasi negatif, maka *f-score* (*avg_f1_score*) memiliki korelasi yang positif, hal ini dapat disebabkan juga karena *precision* (*avg_precision*) juga memiliki korelasi positif sehingga mempengaruhi perhitungan *f-score*.

B. Hasil Pengujian tanpa menggunakan Oversampling SMOTE

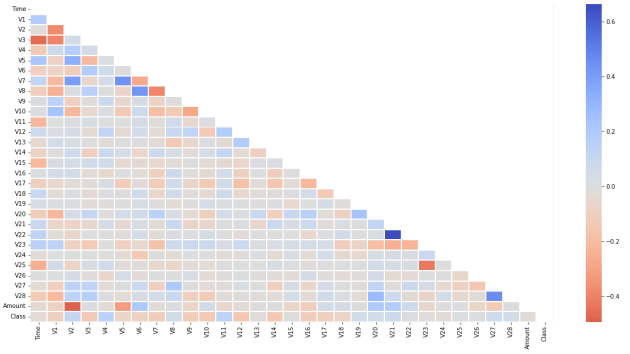
Pada bagian ini dilakukan pengujian tambahan XGBoost tanpa menerapkan oversampling SMOTE pada data latih. Pengujian tambahan bertujuan untuk membandingkan hasil pengujian yang menerapkan SMOTE dan yang tidak menerapkan SMOTE. Berdasarkan pengujian sebelumnya, kombinasi nomor 7 dan 62 merupakan kombinasi yang masuk ke dalam hasil pengujian terbaik.

Tabel IV
Hasil Pengujian XGBoost Tanpa Oversampling SMOTE

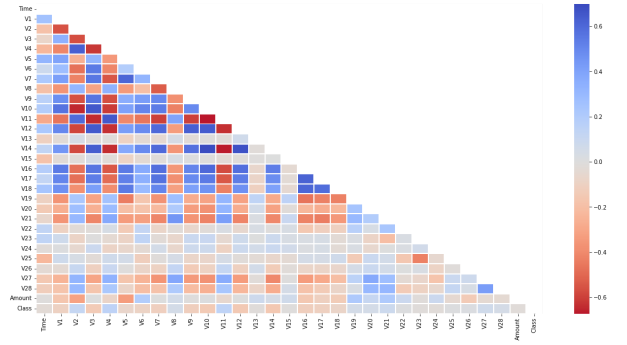
No	Recall	Accuracy	F-score
7	0,7895	0,9974	0,8571
62	0,7789	0,9973	0,8506

Berdasarkan pengujian yang dilakukan, terlihat bahwa kombinasi nomor 7 dan 62 memperoleh nilai recall yang lebih rendah jika dibandingkan dengan pengujian yang menggunakan SMOTE sebagai teknik oversampling. Pada pengujian XGBoost tanpa SMOTE, kombinasi nomor 7 memperoleh nilai *recall* tertinggi sebesar 78,95%. Sedangkan kombinasi nomor 62 memperoleh nilai recall sebesar 77,89%. Pada pengujian XGBoost dengan SMOTE, kombinasi nomor 7 memperoleh recall sebesar 85,26% dan kombinasi nomor 62 memperoleh nilai *recall* sebesar 84,21% seperti pada Tabel III. Terdapat perbedaan yang cukup signifikan, yaitu nilai recall mengalami penurunan sebesar 7-8%. Sehingga berdasarkan perbandingan pengujian XGBoost dengan SMOTE dan tanpa SMOTE, terdapat perbedaan nilai *recall* yang signifikan, dan nilai *recall* sangat penting dalam melakukan deteksi transaksi penipuan kartu kredit karena menekan *false negative*, yaitu kesalahan dalam mendeteksi transaksi penipuan kartu kredit.

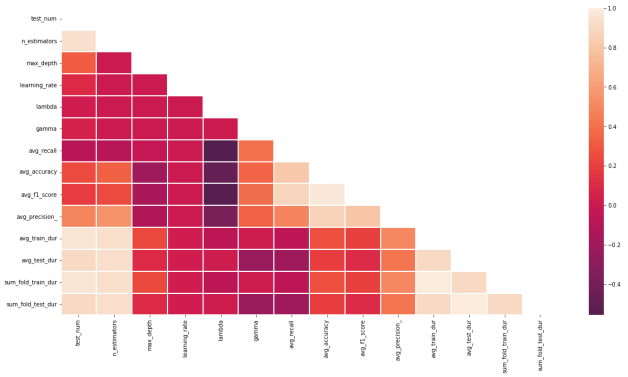
Gambar 4 dan 5 menunjukkan perbandingan matriks korelasi antar fitur tanpa menggunakan *oversampling* SMOTE dengan yang menggunakan *oversampling* SMOTE Berdasarkan kedua gambar tersebut, hasilnya dengan menggunakan proses oversampling SMOTE, lebih banyak fitur-fitur yang mampu ditangkap korelasinya, baik korelasi positif ataupun korelasi negatif dibandingkan dengan apabila tidak menggunakan proses oversampling SMOTE. Perbedaan korelasi ini dapat disebabkan oleh data yang akan digunakan berubah, karena proses oversampling meningkatkan jumlah kelas minoritas.



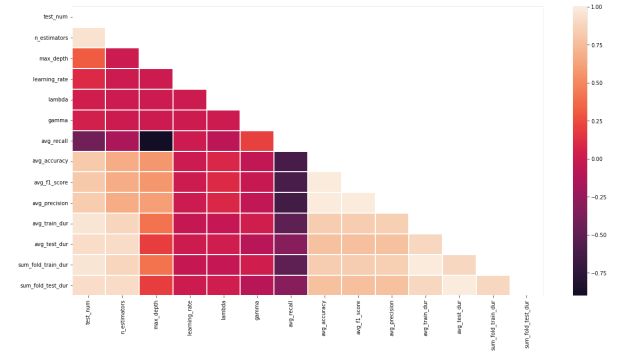
Gambar 4 Matriks korelasi tanpa menggunakan oversampling SMOTE



Gambar 5 Matriks korelasi dengan menggunakan oversampling SMOTE



Gambar 6 Matriks korelasi hasil pengujian dengan *hyperparameter* tanpa *oversampling* SMOTE

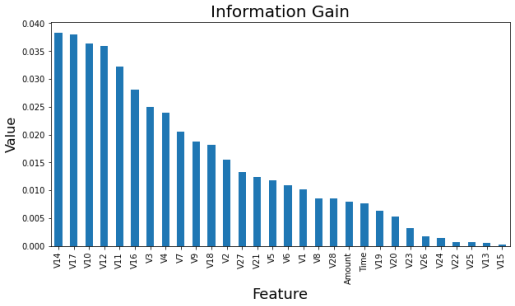


Gambar 7 Matriks korelasi hasil pengujian dengan *hyperparameter* dengan *oversampling* SMOTE

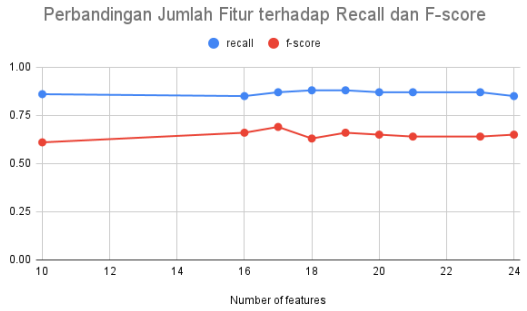
Gambar 6 dan 7 menunjukkan perbandingan matriks korelasi pengukuran hasil pengujian dengan hyperparameter yang diuji tanpa oversampling SMOTE dan dengan menggunakan oversampling SMOTE. Berdasarkan gambar tersebut, terdapat perbedaan diantaranya adalah pada korelasi pengukuran tanpa menggunakan oversampling SMOTE, *lambda* merupakan *hyperparameter* yang lebih banyak mempengaruhi *recall* sedangkan pengujian yang menggunakan SMOTE yaitu pada Gambar 7 mampu menangkap korelasi *max_depth* dengan *recall* yang lebih besar. Sehingga dapat disimpulkan bahwa perbedaan tanpa menggunakan *oversampling* SMOTE dan dengan menggunakan proses *oversampling* SMOTE mempengaruhi korelasi data karena *oversampling* meningkatkan jumlah data minoritas dengan membuat data sintetis sehingga karakteristik data berubah dan berbeda dari data sebelum dilakukan *oversampling* SMOTE, selain itu *oversampling* SMOTE juga berperan dalam meningkatkan nilai *recall* yang signifikan.

C. Pengujian XGBoost pada Data yang Berbeda

Pengujian ini dilakukan untuk mengetahui apakah seleksi fitur mampu meningkatkan performa XGBoost, maka dilakukan pengujian XGBoost dengan seleksi fitur menggunakan *mutual information* (*information gain*).



Gambar 8 Nilai information gain untuk setiap fitur yang telah diurutkan dari yang tertinggi hingga terendah



Gambar 9 Hasil pengujian untuk menentukan jumlah fitur yang digunakan

Gambar 8 menunjukkan nilai *information gain* pada setiap fitur. Dapat terlihat bahwa fitur yang memperoleh nilai information gain yang tinggi diantaranya adalah fitur V14, V17, V10, V12, V11, dan V16 dan nilai information gain terendah diperoleh diantaranya diperoleh oleh fitur V24, V22, V25, V13, dan V15. Berdasarkan Gambar 9 terlihat bahwa dengan menggunakan 18 dan 19 fitur dengan information gain tertinggi, keduanya memperoleh nilai *recall* tertinggi sebesar 88,42% dan nilai *f-score* masing-masing sebesar 63,39% dan 66,14%. Sebanyak 19 fitur dipilih karena memiliki nilai f-score yang lebih baik daripada menggunakan 18 fitur.

Jika dibandingkan dengan pengujian sebelumnya, hasil pengujian menggunakan seleksi fitur yang terbaik memperoleh nilai recall sebesar 88,42% dan *f-score* sebesar 66,14%. Pada hasil pengujian menggunakan SMOTE, kombinasi *hyperparameter* nomor 7 memperoleh nilai recall sebesar 85,25% dan *f-score* sebesar 65,06%. Dalam hal ini terdapat perbedaan nilai *recall* tertinggi sekitar 0,0316 atau sekitar 3% dan perbedaan nilai *f-score* sebesar 0,0108 atau hanya sekitar 1%. Berdasarkan hasil diatas dapat disimpulkan bahwa menggunakan seleksi fitur dengan *information gain* tidak berpengaruh banyak dalam meningkatkan nilai *recall* ataupun *f-score* dalam pengujian ini.

D. Pengujian XGBoost pada Data yang Berbeda

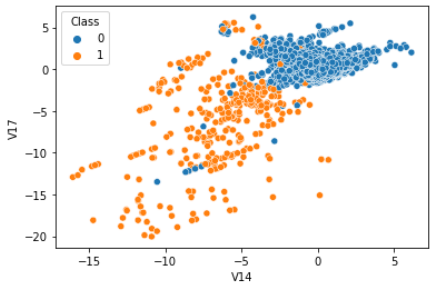
Pengujian ini bertujuan untuk menguji XGBoost dengan dataset lain untuk mengetahui performa algoritme XGBoost dengan oversampling SMOTE. Dalam melakukan pengujian dengan menggunakan dataset kartu kredit lain, terdapat kendala dimana dataset sulit didapatkan karena data mengenai kartu kredit merupakan data yang bersifat privasi dan jarang untuk dipublikasikan. Oleh karena itu, untuk mencoba menguji XGBoost menggunakan dataset yang berbeda, data uji pada pengujian sebelumnya akan menjadi data latih, sedangkan data latih pada pengujian sebelumnya akan menjadi data uji. Dengan menukar posisi data uji sebagai data latih dan data latih sebagai data uji, maka pengujian ini menggunakan data pelatihan dan data pengujian yang berbeda dengan pengujian sebelumnya. Pada data latih, terdapat 9555 data dengan 9460 diantaranya merupakan kelas negatif atau *non-fraud* dan 95 diantaranya dikategorikan sebagai kelas positif atau *fraud*. Pada data uji, terdapat 1953 data dengan 1575 merupakan kelas negatif dan 378 diantaranya merupakan kelas positif.

Dalam pengujian ini, kombinasi nomor 7 meraih nilai *recall*, akurasi dan *f-score* tertinggi. jika dibandingkan dengan pengujian sebelumnya yaitu pada Tabel III, nilai *recall* pada kombinasi nomor 7 dan 62 tidak berbeda jauh.

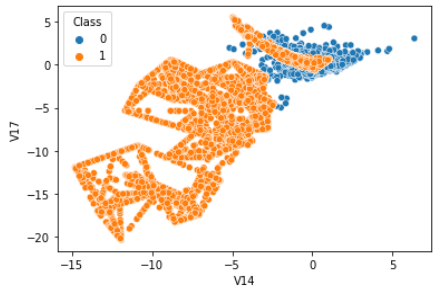
Kenaikan nilai *f-score* jika dibandingkan dengan pengujian sebelumnya disebabkan nilai *precision*. Kenaikan *precision* berarti model lebih baik dalam mengklasifikan kelas bukan penipuan, dan berkurangnya *false positive* yaitu kondisi ketika data diklasifikasikan sebagai penipuan tetapi data tersebut bukan penipuan. Hal ini disebabkan oleh proporsi data yang berbeda antara pengujian ini dengan pengujian sebelumnya, dimana dalam pengujian ini data latih yang digunakan lebih sedikit dibandingkan pengujian sebelumnya, sehingga berpengaruh ke dalam proses *oversampling* SMOTE. Gambar 10, 11, dan 12 menunjukkan perbandingan penyebaran data sebelum dilakukan *oversampling* dan setelah dilakukan *oversampling* SMOTE pada pengujian ini dan dibandingkan dengan pengujian utama. Pada Gambar 11 terlihat bahwa data sintetik hasil oversampling yang terbentuk lebih sedikit dibandingkan dengan Gambar 12. Selain itu, pada Gambar 11 terdapat ada beberapa data pada kelas negatif yang tertutupi oleh kelas positif namun jumlahnya lebih sedikit dibandingkan Gambar 12. Hal ini yang membuat model XGBoost memperoleh *f-score* dan *precision* yang lebih tinggi, sehingga lebih sedikit model mengalami klasifikasi *false positive*. Sehingga berdasarkan pengujian ini, dapat disimpulkan bahwa model XGBoost tidak mengalami perbedaan nilai *recall* yang signifikan, namun hanya mengalami kenaikan *f-score* yang disebabkan oleh kenaikan precision yang diakibatkan karena perbedaan proporsi data.

TABEL IV
HASIL PENGUJIAN XGBOOST PADA DATA LAIN

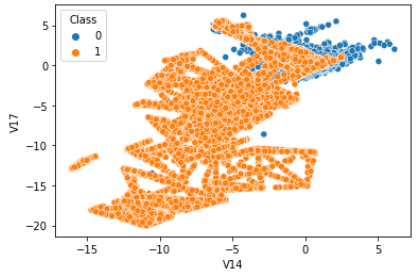
No	Recall	Accuracy	F-score
7	0,8598	0,9708	0,9194
62	0,8413	0,9677	0,9099



Gambar 10 Penyebaran data sebelum oversampling



Gambar 11 Penyebaran data pada pengujian dengan menggunakan dataset yang ditukar setelah oversampling



Gambar 12 Perbandingan penyebaran data sebelum dan setelah dilakukan oversampling SMOTE

V. Simpulan

Dalam kasus deteksi penipuan kartu kredit yang menekankan sensitivitas (*recall*) untuk deteksi transaksi penipuan (*fraud*) dengan rasio jumlah data yang tentunya lebih rendah dari kelas non-fraud, maka kombinasi nomor 7 merupakan hyperparameter terbaik karena memperoleh nilai recall tertinggi.

Recall tertinggi pada data uji diperoleh oleh kombinasi nomor 7 sesuai pada Tabel III dengan nilai recall tertinggi sebesar 85,26%. Akurasi dan f-score tertinggi diperoleh oleh kombinasi nomor 62 dengan nilai masing- masing 99,77% dan 87,91%. Dengan menggunakan seleksi fitur, nilai *recall* hanya meningkat sebesar 3%.

Berdasarkan hasil pengujian yang dilakukan, semakin besar nilai *n_estimator* dan *max_depth* akan menurunkan nilai *recall* tetapi meningkatkan nilai *f-score* karena adanya *trade-off* antara nilai *recall* dan *precision* yang mempengaruhi perhitungan *f-score*. Selain itu tingginya nilai *n_estimator* dan *max_depth* akan mengakibatkan model cenderung mengalami *overfit*. Semakin besar nilai *gamma* meningkatkan nilai *recall* serta semakin besar nilai *lambda* menurunkan nilai *recall* Hal ini karena *gamma* dan *lambda* berperan untuk mencegah adanya *overfitting*, tentunya apabila nilainya terlalu besar maka akan menyebabkan model mengalami *underfitting*, tetapi jika nilainya terlalu kecil maka model tetap akan mengalami *overfitting*.

Pengujian tanpa menggunakan *oversampling* SMOTE terdapat penurunan *recall* yang signifikan sebesar 7-8%, sehingga menggunakan SMOTE diperlukan untuk melakukan *oversampling* data agar model menjadi lebih sensitif dalam mendeteksi data transaksi penipuan kartu kredit.

Dalam penelitian lanjutan, dapat dilakukan pengujian terhadap pengaruh *hyperparameter* yang terdapat dalam teknik oversampling SMOTE seperti rasio kelas yang akan digunakan dan jumlah tetangga yang perlu dipertimbangkan untuk melakukan deteksi penipuan kartu kredit. Selain itu juga dapat menambah jumlah pengujian terhadap jumlah *hyperparameter lambda* dan *gamma.*

Daftar Referensi

[1] S. P. Maniraj, A. Saini, S. Ahmed dan S. D. Sarkar, "Credit Card Fraud Detection using Machine Learning and Data Science," International Journal of Engineering Research Technology (IJERT), vol. 8, no. 9, 2019.

[2] I. Kaur dan M. Kalra, "Ensemble Classification Method for Credit Card Fraud Detection," International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 3, 2019.

[3] A. A. Taha dan S. J. Malebary, "An Intelligent Approach to Credit Card Fraud Detection Using an Optimized Light Gradient Boosting Machine," IEEE Access, vol. 8, 2020..

[4] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid dan H. Zeineddine, "An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection," IEEE Access, vol. 7, 2019..

[5] S. Marabad, "Credit Card Fraud Detection using Machine Learning," Asian Journal of Convergence in Technology, vol. 7, no. 2, 2021.

[6] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", dalam ACM SIGKDD International Conference, 2016.

[7] Microsoft. "Synthetic Minority Oversampling Technique". [Daring]. Tersedia: www.docs.microsoft.com/en-us/previous-versions/azure/machine-learning/studio-module-reference/smote [29 Januari 2022]

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol 12, 2011.

[9] A. Burkov. *The Hundred-Page Machine Learning Book*, 2019.

[10] Kaggle. "Credit Card Fraud Detection". [Daring]. Tersedia: www.kaggle.com/mlg-ulb/creditcardfraud [10 Oktober 2021]

Nicholas Anthony Suhartono, menerima gelar Sarjana Komputer pada program studi Informatika di Institut Teknologi Harapan Bangsa (ITHB). Saat ini bekerja sebagai Software Engineer di perusahaan pembuatan perangkat lunak.

Ventje Jeremias Lewi Engel, menerima gelar Sarjana Teknik dari Institut Teknologi Bandung (ITB) pada tahun 2012 dan Magister Teknik dari Institut Teknologi Bandung pada tahun 2013. Aktif sebagai dosen di Prodi Informatika Institut Teknologi Harapan Bangsa (ITHB). Minat penelitian pada bidang *Deep Learning, Cybersecurity*, dan *Malware Analysis*.