Analisis Perbandingan Klasifikasi dan Penerapan *SMOTE* Dalam *Imbalanced Data* pada *Credit Card Default*

St Fatika Nabila Halim, dan Ulil Azmi
Departemen Aktuaria, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: ulil_azmi@statistika.its.ac.id

***Abstrak*—Perkembangan teknologi yang pesat melahirkan metode pembayaran elektronik, salah satunya adalah kartu kredit. Penggunaan kartu kredit dinilai memudahkan pemilik dalam bertransaksi. Kemudahan itu sering kali disalahgunakan oleh orang-orang yang tidak bijak, yaitu dengan cara belanja kompulsif. Perilaku belanja kompulsif menggunakan kartu kredit dapat berdampak terhadap risiko gagal bayar atau kartu kredit default. kartu kredit default adalah gagal melakukan pembayaran hutang pada tanggal jatuh tempo. Namun, kasus tersebut tidak selamanya terjadi dengan adanya penjagaan ketat dari pihak bank. Oleh karena itu, terjadi ketidakseimbangan data pada data kejadian yang disimpan dalam sistem. Dataset tidak seimbang menyulitkan metode klasifikasi karena hasil klasifikasi akan berfokus pada kelas mayoritas. Kondisi dataset tidak seimbang dapat diatasi dengan salah satu metode *oversampling*, yaitu *Synthetic Minority Over-sampling Technique* (SMOTE). Metode selanjutnya yang digunakan dalam penelitian ini setelah penerapan SMOTE adalah *random forest classifier* dan *extreme gradient boosting* (XGBOOST). Metode *random forest* mendapatkan nilai AUC yang meningkat 4,29% dari 58,73% menjadi 63,02%. Sementara metode XGBOOST mendapatkan nilai AUC yang juga meningkat 14,78% dari 58,00% menjadi 72,78%. Penentuan metode terbaik dilihat dari nilai AUC yang dihasilkan. Dari hasil tersebut dapat disimpulkan bahwa metode XGBOOST adalah metode terbaik dibandingkan dengan *random forest* karena memiliki nilai AUC yang lebih tinggi.**

***Kata Kunci*—Credit Card Default, Imbalanced Dataset, SMOTE, Random Forest, XGBOOST.**

I. PENDAHULUAN

DUNIA saat ini berada di era industri 4.0, dimana teknologi digital menjadi salah satu aset penting yang dibutuhkan para pelaku industri untuk mengembangkan usahanya. Kehadiran internet dan teknologi saat ini dinilai memberikan dampak yang besar dalam memudahkan aktivitas khususnya di bidang ekonomi digital. Ekonomi digital telah lahir dan berkembang seiring dengan penggunaan teknologi informasi dan komunikasi dunia yang semakin mengglobal.

Lahirnya ekonomi digital tidak lepas dari perkembangan industri perbankan. Dengan perkembangan teknologi informasi, industri perbankan menciptakan opsi/metode pembayaran elektronik yang sangat berbeda dengan metode pembayaran tradisional, yaitu kartu debit dan kartu kredit. Satu-satunya perbedaan mendasar antara sistem pembayaran elektronik dan sistem pembayaran tradisional adalah bahwa data dalam sistem pembayaran elektronik didigitalkan pada suatu sistem data keuangan dalam bentuk debet atau kredit [1]. Sistem pembayaran tunai dinilai mengurangi kenyamanan dalam melakukan transaksi ketika nilai transaksinya besar. Pembeli merasa mereka menimbulkan risiko keamanan yang relatif tinggi. Hal itu menyebabkan para konsumen atau nasabah lebih memilih menggunakan kartu kredit sebagai alat transaksi. Namun, dengan kemudahan itu juga sering kali disalahgunakan oleh orang-orang yang tidak bijak dalam menggunakan kartu kredit, salah satunya dengan perilaku belanja kompulsif. Perilaku belanja kompulsif adalah proses belanja berulang yang sering dan berlebihan yang disebabkan oleh kecanduan, depresi, atau kebosanan [2]. Perilaku belanja kompulsif dengan menggunakan kartu kredit dapat berdampak terhadap risiko gagal bayar atau *credit card default*.

Tabel 1.
Confusion Matrix

Confusion Matrix		**Nilai Sebenarnya**	
		True	***False***
Nilai Prediksi	***True***	TP (*True Positive*) *Correct result*	FP (*False Positive*) *Unexpected result*
	False	FN (*False Negative*) *Missing result*	TN (*True Negative*) *Correct absence of result*

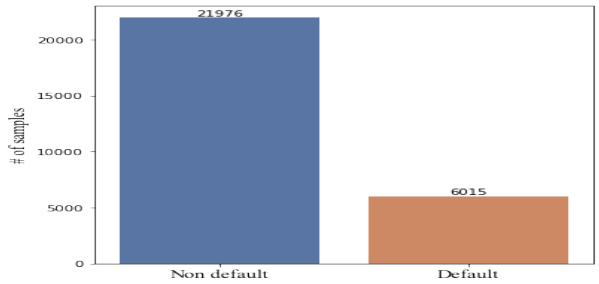
Tabel 2.
Statistika Deskriptif

	Limit Balance	***Age***	***Bill Amount***	***Pay Amount***
Min	10.000,00	21	45	14
Mean	166.341,45	35	$2{,}838 \times 10^5$	$3{,}323 \times 10^4$
Max	1.000.000,00	79	$5{,}263 \times 10^6$	$3{,}764 \times 10^6$
Std	130.295,92	9	$3{,}838 \times 10^5$	$6{,}197 \times 10^4$

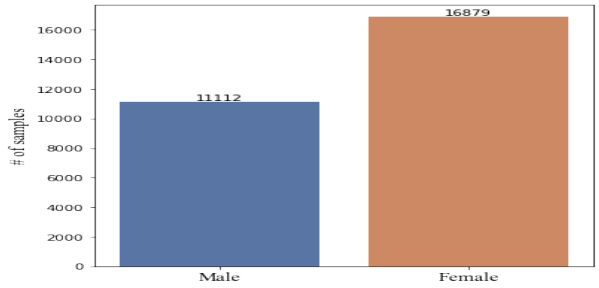
Credit card default adalah gagal melakukan pembayaran hutang pada tanggal jatuh tempo. Jika ini terjadi dengan kartu kredit, kreditur mungkin menaikkan suku bunga ke *default* (atau tingkat penalti) atau menurunkan batas kredit. Namun, kasus gagal bayar tidak selamanya terus terjadi karena adanya seleksi ketat dari pihak bank. Oleh karena itu, data transaksi yang tercatat pada sistem mengalami ketidakseimbangan.

Data tidak seimbang atau *imbalanced dataset* adalah suatu kondisi dimana distribusi kelas data tidak seimbang, jumlah kelas data (*instance*) lebih sedikit atau lebih dari satu atau lebih dari jumlah kelas data lainnya [3]. Jenis masalah data tidak seimbang dapat diselesaikan dengan menerapkan kombinasi metode *oversampling*, yaitu teknik minoritas *resampling* dengan menggunakan *synthetic minority oversampling technique* (SMOTE) [4].

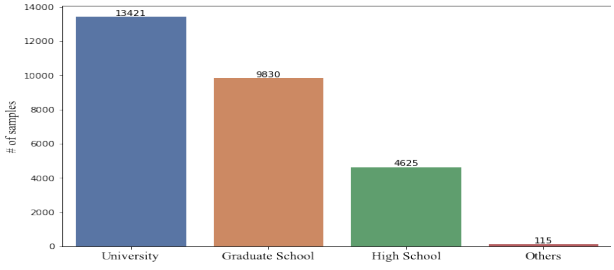
Dalam penelitian ini dilakukan analisis klasifikasi pada dataset tidak seimbang dengan melihat perbandingan hasil parameter evaluasi sebelum dan sesudah penerapan SMOTE menggunakan metode *Random Forest Classifier* dan *Extreme Gradient Boosting* (XGBOOST). Hasil yang diharapkan pada penelitian ini adalah pemilihan metode yang tepat untuk



Gambar 1. Diagram Batang Variabel *Default Payment.*



Gambar 2. Diagram Batang Variabel *Gender.*



Gambar 3. Diagram Batang Variabel *Education.*

menangani masalah *imbalanced dataset* dengan ketepatan klasifikasi yang baik.

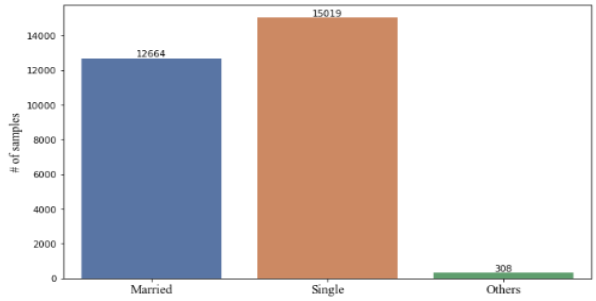
II. TINJAUAN PUSTAKA

A. *Credit Card Default*

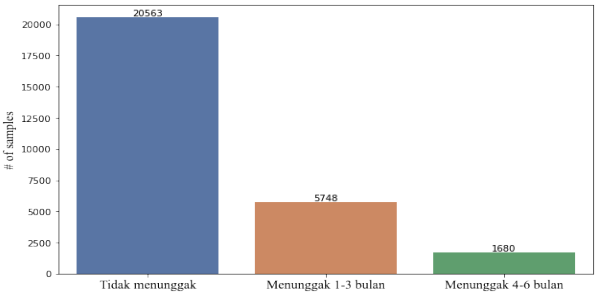
Default adalah kegagalan untuk melakukan pembayaran bunga atau pokok yang diperlukan atas suatu hutang, baik utang itu berupa pinjaman atau jaminan. Risiko gagal bayar merupakan pertimbangan penting bagi kreditur. Sebuah *default* mengurangi peringkat kredit peminjam dan dapat membatasi kemampuan mereka untuk meminjam dimasa depan.

B. *Dataset Tidak Seimbang*

Dataset tidak seimbang atau *imbalanced dataset* merupakan suatu keadaan dimana sebuah dataset memiliki perbandingan jumlah kelas yang tidak seimbang, atau dapat dikatakan jumlah kelas pada data yang satu jauh lebih sedikit atau bahkan jauh lebih banyak jika dibandingkan dengan kelas yang lainnya [3]. Ketika data tidak seimbang ini digunakan sebagai sampel pelatihan klasifikasi, model klasifikasi yang dihasilkan tidak mampu untuk memprediksi setiap kelas secara optimal [5]. Masalah ketidakseimbangan data dapat diselesaikan dengan menggunakan salah satu teknik *resampling*. Teknik ini digunakan untuk mengatur kembali sampel set data yang tidak seimbang sehingga distribusi kelas menjadi simetris selama fase pelatihan [6].



Gambar 4. Diagram Variabel *Marriage.*



Gambar 5. Diagram Batang Variabel *Pay.*

Tabel 3.
Confusion Matrix Data *Training Random Forest*

Confusion Matrix		**Nilai Sebenarnya**	
		True	***False***
Nilai Prediksi	***True***	19.732	2.616
	False	47	2.798

C. *Strategi Sampling*

Sampling adalah cabang ilmu statistika yang memfokuskan penelitian pada pemilihan data yang dihasilkan dari kumpulan data populasi. Teknik strategi *sampling* meliputi *oversampling* kelas minoritas atau *undersampling* kelas mayoritas [7]. Strategi *undersampling* dilakukan pada kelas mayoritas sedemikian rupa sehingga jumlah *instance* dari kelas mayoritas sama dengan jumlah kelas minoritas. Strategi ini dapat diterapkan dengan memilih kelas mayoritas secara acak. Sedangkan strategi *oversampling* dilakukan pada data kelas minoritas sedemikian rupa sehingga jumlah kelas minoritas mendekati jumlah kelas mayoritas. Strategi ini dapat diterapkan dengan menduplikasi kelas minoritas.

D. *Synthetic Minority Over-sampling Technique* (SMOTE)

SMOTE adalah salah satu teknik *oversampling* paling populer dan pendekatan pertama yang mencoba menyeimbangkan *dataset* dengan menghasilkan *instance* minoritas sintetis. Mekanisme SMOTE untuk menghasilkan *instance* sintetis didasarkan pada algoritma K-*Nearest Neighbor*. Teknik ini bekerja dengan mengelompokkan data berdasarkan tetangga terdekat. Tetangga terdekat dipilih berdasarkan jarak *Euclidean* antara dua data. Jarak *euclidean* antara kedua vektor dapat dihitung dengan persamaan berikut.

$$d(x_1, x_2) = \sqrt{(x_{(1),1} - x_{(2),1})^2 + (x_{(1),2} - x_{(2),2})^2 + \cdots + (x_{(1),p} - x_{(2),p})^2} \quad (1)$$

sedangkan *synthetic* data dilakukan dengan menggunakan persamaan berikut.

$$x_{syn} = x_i + (x_{knn} - x_i) \times \beta, i = 1,2, \ldots, n \quad (2)$$

Penentuan jarak untuk variabel kategorik menggunakan

Tabel 8.
Confusion Matrix Data Testing Random Forest

Confusion Matrix		**Nilai Sebenarnya**	
		True	***False***
Nilai Prediksi	***True***	2.124	476
	False	73	125

Tabel 9.
Parameter Evaluasi *Random Forest* Sebelum SMOTE

Parameter Evaluasi	**Nilai**	
	Data *Training*	**Data *Testing***
Precision	88,29%	81,69%
F1 *Score*	93,67%	88,55%
AUC	75,72%	58,73%

Tabel 10.
Confusion Matrix Data *Training* XGBOOST

Confusion Matrix		**Nilai Sebenarnya**	
		True	***False***
Nilai Prediksi	***True***	19.218	4.423
	False	561	991

Tabel 11.
Confusion Matrix Data *Testing* XGBOOST

Confusion Matrix		**Nilai Sebenarnya**	
		True	***False***
Nilai Prediksi	***True***	2.132	487
	False	65	114

jarak *Hamming* yang ditunjukkan pada persamaan berikut.

$$D(x_1, x_2) = \begin{cases} 1 & jika\ x_1 = x_2 \\ 0 & jika\ x_1 \neq x_2 \end{cases} \quad (3)$$

E. *Random Forest Classifier*

Random forest classifier adalah metode klasifikasi yang menggabungkan pohon klasifikasi saling independen (CART) dari distribusi yang sama dengan proses *voting* (jumlah maksimum) untuk mendapatkan prediksi klasifikasi. *Random forest* adalah evolusi dari metode *ensemble* yang awalnya dikembangkan oleh Leo Breiman dan digunakan untuk meningkatkan akurasi klasifikasi. Pada proses *bagging*, ketika pohon klasifikasi dengan banyak versi dibangkitkan oleh *bootstrap resampling* dan digabungkan ke dalam prediksi akhir, *Random Forest* menerapkan pohon klasifikasi tidak hanya pada data sampel, tetapi juga pada variabel prediktor proses pengacakan dilakukan. Proses ini dengan demikian menghasilkan kumpulan pohon klasifikasi dengan berbagai ukuran dan bentuk. Hasil yang diharapkan adalah kumpulan pohon klasifikasi dengan korelasi antar pohon yang rendah. Korelasi yang lebih kecil mengurangi hasil kesalahan prediksi untuk hutan acak [8].

F. *Extreme Gradient Boosting* (XGBOOST)

Extreme Gradient Boosting (XGBOOST) merupakan salah satu teknik dalam *machine learning* untuk analisis regresi dan klasifikasi berdasarkan *Gradient Boosting Decision Tree*. Metode XGBOOST menghubungkan antara *boosting* dan optimasi dalam membangun *Gradient Boosting Machine* (GBM). Metode *boosting* digunakan dalam membangun model baru untuk memprediksi kesalahan model sebelumnya. Model-model baru akan terus bertambah hingga perbaikan *error* tidak dapat dilakukan lagi.

G. K-fold Cross Validation

Cross validation adalah metode statistik untuk mengevaluasi dan membandingkan algoritma *machine learning* dengan membagi data menjadi dua segmen, data *training* yang digunakan untuk mempelajari atau melatih model dan data *testing* digunakan untuk memvalidasi model. Bentuk dasar dari *cross validation* adalah *k-fold cross validation* [9]. Pada *k-fold cross validation*, data dipartisi secara acak menjadi *k* bagian yang bersifat *mutually exclusive* D_1, D_2, ..., D_k, yang masing masing memiliki ukuran yang sama. Proses *training* dan *testing* dilakukan sebanyak *k* kali. Dalam iterasi ke-*i*, data partisi D_i diposisikan sebagai data *testing*, sementara partisi lain yang tersisa secara kolektif digunakan untuk melatih model [10].

Tabel 4.
Parameter Evaluasi XGBOOST Sebelum SMOTE

Parameter Evaluasi	**Nilai**	
	Data Training	**Data Testing**
Precision	81,29%	81,40%
F1 *Score*	88,52%	88,53%
AUC	57,73%	58,00%

Tabel 5.
Perbandingan Jumlah Kategori Sebelum dan Sesudah SMOTE

	Kategori	Sebelum SMOTE	Sesudah SMOTE
Data *training*	*Non default* (0)	19.763	19.763
	Default (1)	5.429	19.763

Tabel 6.
Hyperparameter Random Forest Sesudah SMOTE

Parameter	***Grid Search Values***	***Best Parameter***
n_estimators	50, 75, 100, 200	100
max_features	sqrt, log2	sqrt
max_depth	7, 8, 9, 10, 12, 15	15
min_samples_leaf	2, 3, 4, 5, 6, 7, 8, 9	6
criterion	entropy, gini	entropy

Tabel 7.
Confusion Matrix Data *Training Random Forest* Sesudah SMOTE

Confusion Matrix		**Nilai Sebenarnya**	
		True	***False***
Nilai Prediksi	***True***	19.184	3.149
	False	705	2.154

H. *Hyperparameter Tuning*

Optimasi *hyperparameter* adalah ilmu menyetel *hyperparameter* dari algoritma untuk mendapatkan kinerja yang optimal. Algoritma yang berbeda memiliki tipe *hyperparameter* yang berbeda pula. Dalam penelitian ini digunakan metode *grid search* dalam mengaplikasikan *hyperparameter tuning*. *Grid search* merupakan metode alternatif yang digunakan untuk mencari parameter terbaik dalam suatu model, sehingga metode yang digunakan dapat memprediksi data yang digunakan secara akurat.

Hyperparameter yang digunakan dalam *hyperparameter tuning* pada klasifikasi menggunakan metode *Random Forest* adalah sebagai berikut.

a. *n_estimator*, menunjukkan banyaknya pohon dalam hutan
b. *max_depth*, menunjukkan kedalaman maksimum pohon.
c. *min_samples_leaf,* jumlah minimum sampel yang diperlukan untuk membagi node internal
d. *max_features*, jumlah fitur yang perlu dipertimbangkan saat mencari pemisah terbaik.
e. *criterion*, untuk mengukur kualitas split. Kriteria yang

Tabel 16.
Confusion Matrix Data *Testing Random Forest* Sesudah SMOTE

Confusion Matrix		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	1.889	459
	False	198	253

Tabel 17.
Parameter Evaluasi *Random Forest* Sesudah SMOTE

Parameter Evaluasi	Nilai	
	Data *Training*	Data *Testing*
Precision	85,89%	80,45%
F1 *Score*	90,87%	85,18%
AUC	68,53%	63,02%

Tabel 18.
Hyperparameter XGBOOST Sesudah SMOTE

Parameter	*Grid Search Values*	*Best Parameter*
max_depth	5,8,9,10,12,15	5
min_child_weight	1,2,3,4,5,6,7,8	4
learning_rate	0.025, 0.05, 0.1, 0.2, 0.3	0.1
gamma	0.1, 0.2, 0.3, 0.4, 1.0, 1.5, 2.0	2.0

Tabel 19.
Confusion Matrix Data *Training* XGBOOST Sesudah SMOTE

Confusion Matrix		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	19.094	3.211
	False	795	2.092

didukung adalah 'gini' untuk ketidakmurnian dan "entropi" untuk perolehan informasi.

Hyperparameter yang digunakan dalam *hyperparameter tuning* pada klasifikasi menggunakan metode XGBOOST adalah sebagai berikut.

a. *max_depth*, menunjukkan kedalaman maksimum pohon.
b. *min_child_weight*, jumlah minimum berat instance yang dibutuhkan pada seorang anak.
c. *eta* (*learning_rate*), penyusutan ukuran langkah digunakan dalam pembaruan untuk mencegah *overfitting*.
d. *gamma*, pengurangan kerugian minimum yang diperlukan untuk membuat partisi lebih lanjut pada *leaf node* pohon.

I. Parameter Evaluasi

Hasil parameter evaluasi dapat diketahui dengan bantuan *confusion matrix* (CM). CM memiliki informasi perbandingan antara hasil klasifikasi dari keluaran model dan hasil klasifikasi yang sebenarnya. Empat komponen yang membentuk hasil klasifikasi CM adalah *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Isi dari *confusion matrix* atau yang disebut *cross tabulation* dan dapat dilihat pada Tabel 1.

Dalam penelitian ini digunakan beberapa parameter evaluasi seperti *Precision*, F1 *score*, dan *Area Under Curve* (AUC). Persamaan dari parameter evaluasi tersebut adalah sebagai berikut.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4}$$

$$F1\ score = \frac{2\times precision\times recall}{precision+recall} \tag{5}$$

$$AUC = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \tag{6}$$

Tabel 12.
Confusion Matrix Data *Testing* XGBOOST Sesudah SMOTE

Confusion Matrix		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	1.974	349
	False	113	363

Tabel 13.
Parameter Evaluasi XGBOOST Sesudah SMOTE

Parameter Evaluasi	Nilai	
	Data *Training*	Data *Testing*
Precision	85,60%	84,97%
F1 *Score*	90,50%	89,52%
AUC	67,72%	72,78%

Tabel 14.
Perbandingan Parameter Evaluasi Pada Data *Training*

Parameter	*Random Forest*		XGBOOST	
	Sebelum SMOTE	Sesudah SMOTE	Sebelum SMOTE	Sesudah SMOTE
Precision	88,29%	85,89%	81,29%	85,60%
F1 Score	93,67%	90,87%	88,52%	90,50%
AUC	75,72%	68,53%	57,73%	67,72%

Tabel 15.
Perbandingan Parameter Evaluasi Pada Data *Testing*

Parameter	*Random Forest*		XGBOOST	
	Sebelum SMOTE	Sesudah SMOTE	Sebelum SMOTE	Sesudah SMOTE
Precision	81,69%	80,45%	81,40%	84,97%
F1 Score	88,55%	85,18%	88,53%	89,52%
AUC	58,73%	63,02%	58,00%	72,78%

III. METODOLOGI PENELITIAN

Data yang digunakan pada penelitian ini berupa data *credit card default* dari salah satu bank penting (penerbit uang tunai dan kartu kredit) di Taiwan yang diambil pada Oktober 2005. Dalam penelitian ini terdapat 8 variabel independen yang terdiri dari *limit balance*, *age*, *bill amount* dan *pay amount* dengan tipe data rasio serta *gender*, *education*, *marriage*, dan *pay* dengan tipe data nominal. Kategori variabel *gender* adalah *male* dan *female*. Kategori variabel *education* adalah *university*, *graduate school*, *high school*, dan *others*. Kategori variabel *marriage* adalah *married*, *single* dan *others*. Kategori variabel *pay* adalah tidak menunggak, menunggak 1-3 bulan, dan menunggak 4-6 bulan.

IV. HASIL DAN PEMBAHASAN

A. *Preprocessing* Data

Penelitian ini melakukan modifikasi pada variabel *bill amount* dan *pay amount*. Data asli memiliki enam variabel *bill amount* berbeda yang menunjukkan nilai *bill amount* untuk bulan April hingga September. Kemudian dilakukan penggabungan sehingga menghasilkan satu variabel *bill amount* dengan cara menjumlahkan ke-enam variabel tersebut. Hal yang sama juga berlaku pada variabel *pay amount*. Selain itu juga dilakukan penghapusan satu kategorik yang tidak memiliki keterangan pada variabel *education* dan *marriage* serta data yang bernilai negatif pada variabel *bill amount* dan *pay amount*.

B. Statistika Deskriptif Pada Variabel Numerik

Statistika deskriptif pada variabel numerik dapat dilihat pada Tabel 2 yang meliputi nilai minimum, rata-rata, maksimum, dan standar devisasi dari variabel *limit balance*, *age*, *bill amount*, dan *pay amount*.

Tabel 2 menunjukkan nilai standar deviasi variabel *limit balance* adalah 130.295,92 dimana lebih kecil dari nilai *mean* sehingga dapat diartikan *limit balance* memiliki tingkat variasi data yang rendah. Keberagaman nilai *limit balance* dipengaruhi beberapa hal yang menjadi persyaratan dari pihak bank, seperti pendapatan bulanan, ada atau tidaknya utang, jumlah kredit yang diminta, dan lain sebagainya. Variabel *age* juga memiliki nilai standar deviasi 9 tahun yang lebih kecil dari nilai *mean* sehingga dapat diartikan bahwa variabel *age* memiliki tingkat variasi data yang rendah. Tingkat variasi data variabel *age* lebih rendah disebabkan umur minimal dalam pembuatan kartu kredit adalah 21 tahun dan memiliki nilai *range* 58 tahun. Variabel *bill amount* juga memiliki nilai standar deviasi sebesar 383.812 NT yang lebih besar dari nilai *mean* sehingga dapat diartikan bahwa variabel *bill amount* memiliki tingkat variasi data yang tinggi. Variasi jumlah tagihan yang tinggi disebabkan seberapa sering pengguna menggunakan kartu kredit sebelum tanggal pembayaran. Variabel *pay amount* juga memiliki nilai standar deviasi sebesar 61.979 NT yang lebih besar dari nilai *mean* sehingga dapat diartikan bahwa variabel *pay amount* memiliki tingkat variasi data yang tinggi. Variasi jumlah pembayaran yang tinggi disebabkan oleh kemampuan pengguna dalam membayar tagihan kartu kreditnya. Namun terdapat peraturan yang mengatakan bahwa jumlah pembayaran minimum kartu kredit adalah 10% dari jumlah tagihan.

C. Karakteristik Variabel Kategorik

Berikut ini penjelasan terkait karakteristik pada variabel kategorik yang terdiri dari variabel *gender*, *education*, *marriage*, *pay*, dan *default payment*.

1) Karakteristik Variabel Y (Default Payment)

Variabel *default payment* adalah variabel yang menunjukkan status apakah nasabah tersebut berpotensi mengalami *credit card default* atau tidak. Karakteristik data variabel Y (*default payment*) digambarkan melalui diagram batang yang dapat dilihat pada Gambar 1.

Gambar 1 menunjukkan jumlah kartu kredit nasabah tidak termasuk kategori *default* lebih banyak dibanding kategori *default*. Jumlah kartu kredit nasabah yang masuk dalam kategori *non default* sebanyak 21.976 orang atau 78,51% sedangkan jumlah kartu kredit nasabah yang masuk dalam kategori *default* sebanyak 6.015 orang atau 21,49%.

2) Karakteristik Variabel X_2 (Gender)

Variabel *gender* adalah variabel yang menunjukkan jenis kelamin pengguna kartu kredit. Karakteristik data variabel X_2 (*gender*) digambarkan melalui diagram batang yang dapat dilihat pada Gambar 2.

Gambar 2 menunjukkan jumlah nasabah perempuan lebih banyak jika dibandingkan dengan laki-laki. Jumlah nasabah laki-laki sebanyak 11.112 orang atau 39,7% sedangkan jumlah nasabah perempuan sebanyak 16.879 orang atau 60,3%.

3) Karakteristik Variabel X_3 (Education)

Variabel *education* adalah variabel yang menunjukkan pendidikan terakhir para pengguna kartu kredit. Karakteristik data variabel X_3 (*education*) digambarkan melalui diagram batang yang dapat dilihat pada Gambar 3.

Gambar 3 menunjukkan jenjang pendidikan nasabah paling banyak adalah lulusan *university* sebanyak 13.421 orang atau 47,95%, lulusan *graduate school* sebanyak 9.830 orang atau 35,12%, lulusan *high school* sebanyak 4625 orang atau 16,52% dan lulusan lainnya sebanyak 115 orang atau 0,41%.

4) Karakteristik Variabel X_4 (Marriage)

Variabel marriage adalah variabel yang menunjukkan status pernikahan pengguna kartu kredit. Karakteristik data variabel X_4 (*marriage*) digambarkan melalui diagram batang yang dapat dilihat pada Gambar 4.

Gambar 4 menunjukkan lebih banyak nasabah masih berstatus *single* atau belum menikah. Jumlah nasabah *single* sebanyak 15.019 orang atau 53,66%, jumlah nasabah *married* sebanyak 12.664 orang atau 45,24%, dan jumlah nasabah dengan keterangan lainnya sebanyak 308 orang atau 1,1%.

5) Karakteristik Variabel X_6 (Pay)

Variabel *pay* adalah variabel yang menunjukkan apakah pengguna kartu kredit pernah melakukan penunggakan pembayaran kartu kredit atau tidak. Karakteristik data variabel X_6 (*pay*) digambarkan melalui histogram yang dapat dilihat pada Gambar 5.

Gambar 5 menunjukkan lebih banyak nasabah kartu kredit tidak pernah menunggak. Jumlah nasabah yang tidak pernah menunggak sebanyak 20.563 orang atau 73,46%, jumlah nasabah yang pernah menunggak 1-3 bulan sebanyak 5.748 orang atau 20,54%, dan jumlah nasabah yang pernah menunggak 4-6 bulan sebanyak 1.680 orang atau 6%.

D. Stratified k-fold Cross Validation

Pembagian data *training* dan data *testing* pada penelitian ini menggunakan metode *k-fold cross validation* dengan jumlah k=10. Hasil yang didapatkan adalah data *training* sebanyak 25.193 data dengan persentase 90% dan nilai akurasi sebesar 89,42%. Sementara data *testing* sebanyak 2.798 data dengan persentase 10% dan nilai akurasi sebesar 80,37%. Dari hasil *k-fold* didapatkan *subset* terbaik berada pada subset *fold-4* dengan nilai akurasi sebesar 81,03%. Penggunaan nilai *accuracy* hanya digunakan untuk pemilihan nilai *subset* terbaik.

E. Analisis Sebelum SMOTE

Hasil analisis yang didapatkan sebelum penerapan SMOTE untuk masing-masing metode adalah sebagai berikut.

1) Random Forest

Hasil klasifikasi *random forest* pada data *training* digambarkan pada tabel *confusion matrix* pada tabel 3.

Tabel 3 menunjukkan hasil klasifikasi pada data *testing* yang nilai prediksi dan kenyataannya benar (*true positive*) adalah sebanyak 19.732, sedangkan yang nilai prediksi dan kenyataannya salah (*true negative*) yaitu sebanyak 2.798. Selain itu, juga diketahui jumlah hasil klasifikasi yang diprediksikan salah tetapi kenyataannya benar (*false positive*)

yaitu sebanyak 2.616, sedangkan yang diprediksikan benar tetapi kenyataannya salah (*false negative*) yaitu sebanyak 47. Selanjutnya yaitu hasil klasifikasi *random forest* pada data *testing* yang digambarkan pada tabel *confusion matrix*

Tabel 4 menunjukkan hasil klasifikasi pada data *testing* yang nilai prediksi dan kenyataannya benar (*true positive*) adalah sebanyak 2.124, sedangkan yang nilai prediksi dan kenyataannya salah (*true negative*) yaitu sebanyak 125. Selain itu, juga diketahui jumlah hasil klasifikasi yang diprediksikan salah tetapi kenyataannya benar (*false positive*) yaitu sebanyak 476, sedangkan yang diprediksikan benar tetapi kenyataannya salah (*false negative*) yaitu sebanyak 73. Dari hasil *confusion matrix* pada data *training* dan data *testing* tersebut dapat diketahui hasil dari masing-masing parameter evaluasi yang telah dirangkum pada tabel 5.

Tabel 5 menunjukkan hasil parameter evaluasi yang didapatkan dari hasil klasifikasi dengan metode *random forest* sebelum dilakukan SMOTE. Hasil paramater evaluasi pada data *training* menunjukkan model tersebut dapat memprediksikan dengan tepat sebesar 88,29%. Kemudian F1-*score* menujukkan kinerja model dalam mengklasifikasi tiap kategorinya. Didapatkan kinerja model dalam mengklasifikasikan sebesar 93,67%. Sementara itu berdasarkan nilai AUC yang didapat, hasil klasifikasi pada data *training* diklasifikasikan dengan akurat sebesar 75,72% yang berarti model diklasifikasikan dengan baik. Lalu hasil parameter evaluasi pada data *testing* menunjukkan model tersebut dapat memprediksikan dengan tepat sebesar 81,69. Didapatkan kinerja model dalam memprediksikan sebesar 88,55%. Sementara itu berdasarkan nilai AUC, hasil klasifikasi pada data *testing* diklasifikasikan dengan akurat sebesar 58,73% yang berarti model diklasifikasikan cukup baik.

2) XGBOOST

Metode *k-fold cross validation* yang diterapkan pada metode XGBOOST dengan *k=10* menghasilkan nilai akurasi untuk setiap kombinasi (*subset fold*). *Subset* atau kombinasi *k-fold cross validation* terbaik berada pada *subset fold*-10 dengan nilai akurasi sebesar 80,78%. Hasil klasifikasi XGBOOST pada data *training* digambarkan pada tabel *confusion matrix*

Tabel 6 menunjukkan hasil klasifikasi pada data *training* yang nilai prediksi dan kenyataannya benar (*true positive*) adalah sebanyak 19.218, sedangkan yang nilai prediksi dan kenyataannya salah (*true negative*) yaitu sebanyak 991. Selain itu, juga diketahui jumlah hasil klasifikasi yang diprediksikan salah tetapi kenyataannya benar (*false positive*) yaitu sebanyak 4.423, sedangkan yang diprediksikan benar tetapi kenyataannya salah (*false negative*) yaitu sebanyak 561. Selanjutnya yaitu hasil klasifikasi XGBOOST pada data *testing* digambarkan pada tabel *confusion matrix*.

Tabel 7 menunjukkan jumlah hasil klasifikasi pada data *testing* yang nilai prediksi dan kenyataannya benar (*true positive*) adalah sebanyak 2.132, sedangkan yang nilai prediksi dan kenyataannya salah (*true negative*) yaitu sebanyak 114. Selain itu, juga diketahui jumlah hasil klasifikasi yang diprediksikan salah tetapi kenyataannya benar (*false positive*) yaitu sebanyak 487, sedangkan yang diprediksikan benar tetapi kenyataannya salah (*false negative*) yaitu sebanyak 65. Dari hasil *confusion matrix* pada data *training* dan data *testing* tersebut dapat diketahui hasil dari masing-masing parameter evaluasi yang telah dirangkum pada tabel 8.

Tabel 8 menunjukkan hasil parameter evaluasi yang didapatkan dari hasil klasifikasi dengan metode XGBOOST sebelum dilakukan SMOTE. Hasil paramater evaluasi pada data *training* menunjukkan model tersebut dapat memprediksikan dengan tepat sebesar 81,29%. Kemudian F1-*score* menujukkan kinerja model dalam mengklasifikasi tiap kategorinya. Didapatkan kinerja model dalam mengklasifikasikan sebesar 88,52%. Sementara itu berdasarkan nilai AUC yang didapat, hasil klasifikasi pada data *training* diklasifikasikan dengan akurat sebesar 57,73% yang berarti model diklasifikasikan dengan cukup baik. Lalu hasil parameter evaluasi pada data *testing* menunjukkan model tersebut dapat memprediksikan dengan tepat sebesar 81,40%. Didapatkan kinerja model dalam memprediksikan sebesar 88,53%. Sementara itu berdasarkan nilai AUC, hasil klasifikasi pada data *testing* diklasifikasikan dengan akurat sebesar 58,00% yang berarti model diklasifikasikan dengan cukup baik.

F. SMOTE

Penerapan SMOTE dilakukan terlebih dahulu sebelum melakukan klasifikasi ulang untuk setiap metode. Jumlah perbandingan kelas sebelum dan sesudah penerapan metode SMOTE dapat dilihat pada tabel 9.

Tabel 9 menunjukkan jumlah data untuk kategori *non default* dan *default* sebelum dan sesudah penerapan metode SMOTE. Sebelum penerapan metode SMOTE terdapat 19.763 data untuk kategori *non default* dan sebanyak 5.429 data untuk kategori *default*. Kemudian setelah penerapan metode SMOTE, jumlah data untuk kategori *non default* maupun *default* telah seimbang dengan jumlah data sebanyak 19.763 data.

G. Analisis Setelah Resampling Data

Hasil analisis yang didapatkan setelah dilakukan *resampling* data menggunakan metode SMOTE untuk masing-masing metode adalah sebagai berikut.

1) Random Forest

Pada analisis *random forest* kali ini menggunakan *hyperparameter tuning* dengan metode *gridsearchCV*. *Hyperparameter tuning* yang digunakan adalah tabel 10.

Tabel 10 menunjukkan nilai *hyperparameter* yang digunakan untuk klasifikasi dari proses pencarian menyeluruh dengan *gridsearchCV*. Pencarian nilai parameter terbaik pada setiap parameter dilakukan dengan menggunakan *cross validation* 5-*fold estimate* (CV) atau lima kali pengulangan untuk setiap nilai *hyperparameter*-nya. Nilai *hyperparameter* terbaik yang didapatkan berdasarkan Tabel 10 kemudian digunakan dalam membuat model klasifikasi pada data *training* dan data *testing*. Metode *k-fold cross validation* yang diterapkan pada metode *random forest* setelah penerapan SMOTE dengan *k=10* menghasilkan *subset* terbaik pada *subset fold-8* dengan nilai akurasi sebesar 81,10%. Hasil klasifikasi metode *random forest* pada data *training* digambarkan pada tabel 11.

Tabel 11 menunjukkan jumlah hasil klasifikasi pada data *training* yang nilai prediksi dan kenyataannya benar (*true positive*) adalah sebanyak 19.184, sedangkan yang nilai

prediksi dan kenyataannya salah (*true negative*) yaitu sebanyak 2.154. Selain itu, juga diketahui jumlah hasil klasifikasi yang diprediksikan salah tetapi kenyataannya benar (*false positive*) yaitu sebanyak 3.149, sedangkan yang diprediksikan benar tetapi kenyataannya salah (*false negative*) yaitu sebanyak 705. Selanjutnya yaitu hasil klasifikasi *random forest* pada data *testing* digambarkan pada tabel *confusion matrix*.

Tabel 12 menunjukkan jumlah hasil klasifikasi pada data *testing* yang nilai prediksi dan kenyataannya benar (*true positive*) adalah sebanyak 1.889, sedangkan yang nilai prediksi dan kenyataannya salah (*true negative*) yaitu sebanyak 253. Selain itu, juga diketahui jumlah hasil klasifikasi yang diprediksikan salah tetapi kenyataannya benar (*false positive*) yaitu sebanyak 459, sedangkan yang diprediksikan benar tetapi kenyataannya salah (*false negative*) yaitu sebanyak 198. Dari hasil *confusion matrix* tersebut dapat diketahui hasil dari masing-masing parameter evaluasi yang telah dirangkum pada tabel 13.

Tabel 13 menunjukkan hasil parameter evaluasi yang didapatkan dari hasil klasifikasi dengan metode *random forest* sesudah dilakukan SMOTE. Hasil paramater evaluasi pada data *training* menunjukkan model tersebut dapat memprediksikan dengan tepat sebesar 85,89%. Kemudian F1-*score* menujukkan kinerja model dalam mengklasifikasi tiap kategorinya. Didapatkan kinerja model dalam mengklasifikasikan sebesar 90,87%. Sementara itu berdasarkan nilai AUC yang didapat, hasil klasifikasi pada data *training* diklasifikasikan dengan akurat sebesar 68,53% yang berarti model diklasifikasikan dengan cukup baik. Lalu hasil parameter evaluasi pada data *testing* menunjukkan model tersebut dapat memprediksikan dengan tepat sebesar 80,45%. Didapatkan kinerja model dalam memprediksikan sebesar 85,18%. Sementara itu berdasarkan nilai AUC, hasil klasifikasi pada data *testing* diklasifikasikan dengan akurat sebesar 63,02% yang berarti model diklasifikasikan dengan cukup baik.

2) XGBOOST

Selain metode *random forest*, metode XGBOOST kali ini juga menggunakan *hyperparameter tuning* dengan metode *gridsearchCV*. *Hyperparameter tuning* yang digunakan dapat dilihat Tabel 14.

Tabel 14 menunjukkan nilai *hyperparameter* yang digunakan untuk klasifikasi dari proses pencarian menyeluruh dengan *gridsearchCV*. Pencarian nilai *hyperparameter* terbaik pada setiap *hyperparameter* dilakukan dengan menggunakan 5-*fold cross validation* atau lima kali pengulangan untuk setiap *hyperparameter*-nya. Nilai *hyperparameter* terbaik yang didapatkan berdasarkan Tabel 16 kemudian digunakan dalam membuat model klasifikasi. Metode *k-fold cross validation* yang diterapkan pada metode XGBOOST setelah penerapan SMOTE dengan *k=10* menghasilkan *subset* terbaik pada *subset fold-8* dengan nilai akurasi sebesar 80,85%. Hasil klasifikasi XGBOOST pada data *training* digambarkan pada tabel *confusion matrix* pada Tabel 15.

Tabel 15 menunjukkan jumlah hasil klasifikasi pada data *training* yang nilai prediksi dan kenyataannya benar (*true positive*) adalah sebanyak 19.094, sedangkan yang nilai prediksi dan kenyataannya salah (*true negative*) yaitu sebanyak 2.092. Selain itu, juga diketahui jumlah hasil klasifikasi yang diprediksikan salah tetapi kenyataannya benar (*false positive*) yaitu sebanyak 3.211, sedangkan yang diprediksikan benar tetapi kenyataannya salah (*false negative*) yaitu sebanyak 795. Selanjutnya yaitu hasil klasifikasi XGBOOST pada data *testing* digambarkan pada tabel *confusion matrix*.

Tabel 16 menunjukkan jumlah hasil klasifikasi pada data *testing* yang nilai prediksi dan kenyataannya benar (*true positive*) adalah sebanyak 1.974, sedangkan yang nilai prediksi dan kenyataannya salah (*true negative*) yaitu sebanyak 363. Selain itu, juga diketahui jumlah hasil klasifikasi yang diprediksikan salah tetapi kenyataannya benar (*false positive*) yaitu sebanyak 349, sedangkan yang diprediksikan benar tetapi kenyataannya salah (*false negative*) yaitu sebanyak 113. Dari hasil *confusion matrix* tersebut dapat diketahui hasil dari masing-masing parameter evaluasi yang telah dirangkum pada tabel 17.

Tabel 17 menunjukkan hasil parameter evaluasi yang didapatkan dari hasil klasifikasi dengan metode XGBOOST sesudah dilakukan SMOTE. Hasil paramater evaluasi pada data *training* menunjukkan model tersebut dapat memprediksikan tepat sebesar 85,60%. Kemudian F1-*score* menujukkan kinerja model dalam mengklasifikasi tiap kategorinya. Didapatkan kinerja model dalam mengklasifikasi adalah sebesar 90,50%. Sementara itu berdasarkan nilai AUC yang didapat, hasil klasifikasi pada data *training* diklasifikasikan dengan akurat sebesar 67,72% yang berarti model diklasifikasikan dengan cukup baik. Lalu hasil parameter evaluasi pada data *testing* menunjukkan model tersebut dapat memprediksikan dengan tepat sebesar 84,97%. Didapatkan kinerja model dalam memprediksikan adalah sebesar 89,52%. Sementara itu berdasarkan nilai AUC, hasil klasifikasi pada data *testing* diklasifikasikan dengan akurat sebesar 72,78% yang berarti model diklasifikasikan dengan cukup baik.

H. Perbandingan Hasil Klasifikasi

Perbandingan klasifikasi berdasarkan paramater evaluasi dilakukan untuk memilih manakah metode yang lebih baik dari kedua metode yang digunakan. Hasil perbandingan yang digunakan adalah ketepatan klasifikasi menggunakan data *training* dan data *testing* pada masing-masing metode. Hasil yang didapatkan pada data *training* digambarkan pada Tabel 18.

Tabel 18 menunjukkan hasil evaluasi parameter pada data *training* untuk masingk-masing metode sebelum dan sesudah penerapan metode SMOTE. Berdasarkan hasil evaluasi paramater sebelum penerapan metode SMOTE, dapat disimpulkan bahwa metode yang memiliki nilai *precision,* F1 score dan AUC yang lebih tinggi pada masing-masing kategori adalah metode *random forest*. Selanjutnya berdasarkan hasil evaluasi parameter setelah penerapan metode SMOTE, metode yang memiliki nilai *precision,* F1 *score*, dan AUC yang lebih tinggi untuk masing-masing kategori adalah metode *random forest*. Sehingga dapat disimpulkan bahwa metode *random forest* memiliki hasil parameter evaluasi yang lebih baik dibanding XGBOOST pada data *training*. Hasil evaluasi yang baik dari kedua metode sebelum penerapan SMOTE dapat terjadi ketika terdapat *overfitting* pada model sehingga akan menghasilkan

nilai evaluasi yang cenderung tinggi. Salah satu cara mengatasi adanya *overfitting* pada model adalah dengan penerapan metode SMOTE. Oleh karena itu diperoleh nilai evaluasi yang menurun karena metode SMOTE dapat mengatasi masalah *overfitting*. Lalu perbandingan hasil evaluasi data *testing* untuk masing-masing metode ditunjukkan pada tabel 19.

Tabel 19 menunjukkan hasil evaluasi parameter pada data *testing* untuk masing-masing metode sebelum dan sesudah *resampling* data. Berdasarkan hasil evaluasi paramater sebelum penerapan metode SMOTE, dapat disimpulkan bahwa metode yang memiliki nilai *precision*, F1 *score*, dan AUC yang lebih tinggi adalah metode *random forest*. Selanjutnya berdasarkan hasil evaluasi parameter setelah penerapan metode SMOTE, metode yang memiliki nilai *precision,* F1 *score*, dan AUC yang lebih tinggi untuk masing-masing kategori adalah metode XGBOOST. Sehingga dapat disimpulkan bahwa metode XGBOOST memiliki hasil parameter evaluasi yang lebih baik dibanding *random forest* pada data *testing*. Pemilihan metode terbaik berdasarkan nilai AUC yang dihasilkan sesudah penerapan metode SMOTE karena nilai AUC merupakan rangkuman dari keseluruhan parameter evaluasi yang digunakan. Berdasarkan hasil perbandingan diatas, dapat disimpulkan bahwa metode yang lebih baik dalam menangani permasalahan *imbalanced dataset* adalah metode XGBOOST.

V. KESIMPULAN DAN SARAN

Berdasarkan pembahasan sebelumnya dapat disimpulkan bahwa setelah penerapan SMOTE, metode XGBOOST merupakan metode terbaik dalam penelitian ini karena memiliki nilai AUC yang lebih tinggi dibanding metode *random forest*.

Pada penelitian serupa selanjutnya disarankan untuk menggunakan data *imbalanced* lainnya seperti *costumer churn*, data penyakit, *fraud*, dan lain sebagainya. Selain itu juga dapat menggunakan nilai *hyperparameter* yang berbeda untuk mendapatkan hasil klasifikasi yang lebih baik.

DAFTAR PUSTAKA

[1] A. Nugroho, *E-Commerce: Memahami Perdagangan Modern di Dunia Maya*. Bandung : Informatika Bandung, 2006.

[2] J. Z. and R. P. M. Solomon, *Consumer Behaviour: Buying, Having, and Being*, 4th ed. New York: Pearson Prentice Hall, 2001.

[3] A. Ali, S. Mariyam Shamsuddin, A. Ralescu, and A. L. Ralescu, "Classification with class imbalance problem: a review," *Classification Int. J. Advance Soft Compu. Appl*, vol. 5, no. 3, p. 30, 2013,

[4] A. Bisri and R. Rachmatika, "Integrasi gradient boosted trees dengan SMOTE dan bagging untuk deteksi kelulusan mahasiswa," *JNTETI*, vol. 8, no. 4, pp. 309–314, 2019, [Online]. Available: https://forlap.ristekdikti.go.id

[5] N. Soonthornphisaj, T. Sira-Aksorn, and P. Suksankawanich, "Media Comment Management using SMOTE and Random Forest Algorithms," in *Proceedings - 2018 IEEE/ACIS 19th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2018*, 2018, pp. 129–134. doi: 10.1109/SNPD.2018.8441039.

[6] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst Appl*, vol. 73, pp. 220–239, May 2017, doi: 10.1016/j.eswa.2016.12.035.

[7] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[8] L. Breiman, J. Friedman, R. Olshen, and Stone, *Classification and Regression Trees*. New York: Chapman Hall, 2007.

[9] P. , T. L. , Refaeilzadeh and H. Liu, "Cross-Validation. Encyclopedia of Database Systems ," 2009.

[10] H. Jiawei, K. Micheline, and P. Jian, *Data Mining: Concepts and Techniques* , 3rd ed. USA: Morgan Kaufmann, 2012.