Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 344-353
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5355

Deteksi Penipuan Kartu Kredit Menggunakan Support Vector Machine dengan Optimasi Grid Search dan Genetic Algorithm

Lailan Sahrina Hasibuan*, Fatimah Alfiatul Jannah

Fakultas Matematika dan Ilmu Pengetahuan Alam, Departemen Ilmu Komputer, Institut Pertanian Bogor, Bogor, Indonesia
Email: 1,*lailan.sahrina@apps.ipb.ac.id, 2,ftmhalfijannah@apps.ipb.ac.id
Email Penulis Korespondensi: lailan.sahrina@apps.ipb.ac.id
Submitted: **16/06/2024**; Accepted: **29/06/2024**; Published**: 29/06/2024**

Abstrak−Transaksi kartu kredit mengalami peningkatan yang signifikan setiap tahunnya. Seiring dengan meningkatnya pemakaian kartu kredit maka meningkat pula risiko tindak penipuan oleh oknum yang tidak bertanggung jawab. Transaksi penipuan kartu kredit dapat dideteksi melalui bantuan *machine learning*. Permasalahan utama dalam mengolah data transaksi ialah dimensinya yang besar, kelas yang tidak seimbang, dan diperlukan proses pendeteksian dengan waktu komputasi yang singkat. Maka dari itu dibutuhkan model serta algoritma optimasi yang tepat untuk mengatasi permasalahan tersebut. Tujuan penelitian ini adalah membangun model pendeteksian transaksi penipuan kartu kredit yang mampu menghasilkan performa baik serta waktu komputasi singkat menggunakan metode *support vector machine* (SVM) dengan optimasi *grid search* dan *genetic algorithm*. Dari tiga model yang dibangun, diperoleh bahwa model SVM menggunakan dataset awal yang diseimbangkan dengan ADASYN dan pencarian parameter terbaik dengan *grid search* sebagai teknik optimasi *hyperparameter* mampu melakukan pendeteksian dengan baik dan waktu komputasi yang singkat. Model ini mampu mendeteksi transaksi fraud dengan sentitifitas 99% dan spesifisitas 99% serta waktu pelatihan model yang paling singkat diantara dua model lainnya.

Kata Kunci: Algoritma Genetika; Kartu Kredit; Optimasi; SVM; Transaksi Penipuan

Abstract−Credit card transactions have increased significantly every year. Along with the increasing use of credit cards, the risk of fraud by irresponsible people also increases. Credit card fraud can be detected with the help of machine learning. The main problem that often encountered is the transaction data has very large dimensions, unbalanced classes, and requires a detection process with a short computation time. Therefore we need a model that can produce good performance with short computation time using the support vector machine (SVM) method with grid search and genetic algorithm optimization. From the three models built, it was found that the SVM model using an initial dataset which was balanced using ADASYN and searching for the best parameters using grid search as a hyperparameter optimization technique was able to carry out good detection and short computing time. This model is able to detect fraudulent transactions with 99% sensitivity and 99% specificity and the shortest model training time among the other two models.

Keywords: Genetic Algorithm; Credit Card; Optimization; SVM; Fraudulent Transaction

1. PENDAHULUAN

Transaksi menggunakan kartu kredit merupakan salah satu metode pembayaran yang terbilang cukup praktis karena pelanggan dapat melakukan transaksi walaupun tidak membawa uang tunai. Dari data yang dirilis oleh Bank Indonesia [1] transaksi kartu kredit mengalami peningkatan yang signifikan per tahunnya. Pada Juli 2022 nilai transaksi kartu kredit meningkat 54.3% dari tahun sebelumnya. Tidak hanya nilai transaksi, volume transaksi kartu kredit juga mengalami peningkatan. Volume transaksi mengalami kenaikan sebesar 34.8% dari tahun sebelumnya yaitu dari 20.9 juta menjadi 28.13 juta kali di bulan Juli 2022 dari 16,58 juta unit pemegang kartu kredit di Indonesia. Terlepas dari banyaknya masyarakat indonesia yang menggunakan kartu kredit, tidak dipungkiri bahwa masih terdapat risiko pemakaiannya. Salah satu risiko yang sering ditemui adalah terkait tindak kejahatan fraud. Fraud atau yang bisa disebut juga sebagai penipuan, dalam konteks transaksi kartu kredit dapat diartikan sebagai penggunaan kartu kredit seseorang untuk melakukan transaksi tanpa sepengetahuan pemilik demi keuntungan pribadi [2].

Seiring dengan berkembangnya teknologi, aksi penipuan kartu kredit dapat terdeteksi dengan pembangunan model dari data histori transaksi melalui bantuan pembelajaran mesin. Pembelajaran mesin menggunakan data transaksi kartu kredit memiliki tantangan tersendiri. Permasalahan utamanya adalah biasanya data transaksi memiliki dimensi sangat besar dan kelas yang tidak seimbang yang mana kelas transaksi asli lebih banyak dibandingkan kelas transaksi penipuan. Data yang tidak seimbang membutuhkan perhatian khusus karena berpengaruh besar terhadap nilai akurasi [3]. Selain itu dalam proses transaksi kartu kredit durasi waktu antara pelanggan melakukan pembayaran dan pembayaran sampai ke rekening tujuan biasanya sangat singkat. Untuk mencegah kerugian yang besar, pendeteksian penipuan kartu kredit haruslah berjalan dengan cepat, maka diperlukan proses pendeteksian dengan waktu komputasi yang singkat [4].

Pola penipuan yang bersifat dinamis memerlukan penelitian berkelanjutan. Sudah ada beberapa pendekatan yang dilakukan untuk melakukan pendeteksian fraud pada transaksi kartu kredit. Teknik yang pernah digunakan antara lain adalah Logistic Regression (LR), Decision Tree (DT), Naive Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), dan beberapa algoritma lainnya [5].

Pada penelitian yang dilakukan [6], selain mempertimbangkan nilai akurasi juga memperhatikan waktu komputasi yang dihasilkan. Dengan menggunakan deep convolutional network (DCNN) dilakukan perbandingan antara tiga teknik optimasi yaitu RMSprop, Adagrad dan Adam. Hasilnya adalah model dengan metode optimasi

This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 344-353
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5355

Adam menghasilkan akurasi paling optimum yaitu sebesar 99% dan kecepatan proses 583 detik. Selain itu optimasi Adam ini juga hanya membutuhkan sedikit memori dan dapat menangani data yang ukurannya besar.

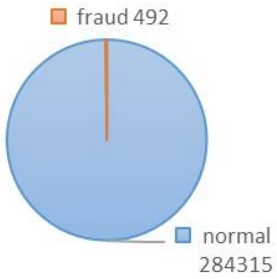
Terakhir adalah penelitian yang dilakukan oleh Siswanto [7], adapun tujuan dari penelitian ini untuk menghasilkan data latih yang ideal dari dataset yang tidak seimbang menggunakan Adaptive Synthetic Sampling (ADASYN) agar dapat digunakan untuk proses pelatihan model Support Vector Machine (SVM). Dibuat tiga model yaitu pemodelan dengan SVM menggunakan data awal yang tidak seimbang, pemodelan SVM dengan menerapkan metode resampling ADASYN, serta yang ketiga pemodelan SVM yang dilatih dengan menggunakan data *Support Vector* dari model pertama lalu diseimbangkan dengan ADASYN. Hasil penelitian menunjukkan bahwa model ketiga mampu mengklasifikasikan transaksi anomali dengan sangat baik. Nilai akurasi yang dihasilkan sebesar 74.4%, nilai sensitivitas sebesar 100% dan nilai spesifisitas yang diperoleh sebesar 74.3%.

Dari beberapa penelitian terdahulu menunjukan bahwa selain dari hasil performa, kecepatan proses komputasi juga penting diperhatikan. Maka dari itu selain dibutuhkan algoritma pembelajaran mesin yang tepat juga harus diperhatikan mengenai algoritma optimasi. Pada penelitian ini digunakan model pembelajaran mesin SVM dengan optimasi GA karena algoritma ini terbukti handal untuk melakukan optimasi [8], [9], [10]. Lalu model tersebut dibandingkan dengan model pembelajaran mesin SVM dengan optimasi grid search, metode mana yang paling baik dari sisi performa dan waktu komputasi.

2. METODOLOGI PENELITIAN

2.1 Dataset Penelitian

Data yang digunakan dalam penelitian ini merupakan data sekunder yang diambil dari situs kaggle. Data tersebut merupakan data transaksi kartu kredit selama dua hari pada bulan September 2013 oleh pemegang kartu di Eropa. Dari total 284,807 jumlah transaksi 492 diantaranya merupakan data penipuan. Dataset ini memiliki ketidakseimbangan yang tinggi yaitu 1:578 antara transaksi fraud dan normal. Gambar 1 menunjukkan perbandingan jumlah transaksi fraud dan normal.



Gambar 1. Perbandingan data fraud dan normal

Dataset ini memiliki 31 atribut yang bersifat numerik. Tiga atribut diantaranya adalah *time* yang menyatakan selisih waktu antara transaksi pertama dan transaksi saat ini, *amount* yang menyatakan nominal transaksi, dan *class* yang mengkodekan jenis transaksi dengan 0 mengkode transaksi normal dan 1 mengkode transaksi fraud. Atribut lainnya yang berjumlah 28 atribut tidak diketahui keterangannya karena telah mengalami transformasi menggunakan metode PCA (*Principal Component Analysis*) guna menjaga kerahasiaan data [11].

2.2 Tahapan Penelitian

Penelitian ini terdiri atas enam tahapan utama yaitu praproses data, optimasi parameter menggunakan dataset kecil, pembentukan dataset untuk pelatihan model, pembentukan model, evaluasi model dan terakhir adalah analisis hasil yang diperoleh. Skema tahapan penelitian dapat dilihat pada Gambar 2. Penjelasan lebih lanjut mengenai metode penelitian diberikan pada subbab di bawah ini.

2.3 Praproses Data

Praproses data dilakukan agar pembentukan model lebih efektif dan efisien. Pertama dilakukan eksplorasi data untuk mengidentifikasi karakteristik dan sebaran data maupun langkah yang tepat untuk menangani data. Selanjutnya pengecekan *missing value*, karena pada data tidak ditemukan adanya *missing value* maka tidak dilakukan penghapusan data. Setelah itu seluruh atribut data dinormalisasi agar memudahkan dalam proses klasifikasi dan meningkatkan kinerja model [12][13]. Normalisasi dapat meningkatkan performa model, karena pada dataset yang telah dinormalisasi, setiap fiturnya memiliki kontribusi yang seimbang terhadap pembentukan model [14] [15]. Berbeda dengan data yang belum dinormalisasi, maka fitur yang memiliki jangkauan paling luas akan mempengaruhi model relatif lebih besar dibandingkan fitur lainnya yang jangkauannya lebih sempit.

Terdapat banyak metode normalisasi yang dapat digunakan untuk praproses data, penelitian [15] merekomendasikan min-max normalization pada kasus klasifikasi kanker payudara. Pada prinsipnya, dataset kanker payudara dan transaksi penipuan kartu kredit memiliki karakteristik yang sama, yaitu dataset yang tidak seimbang

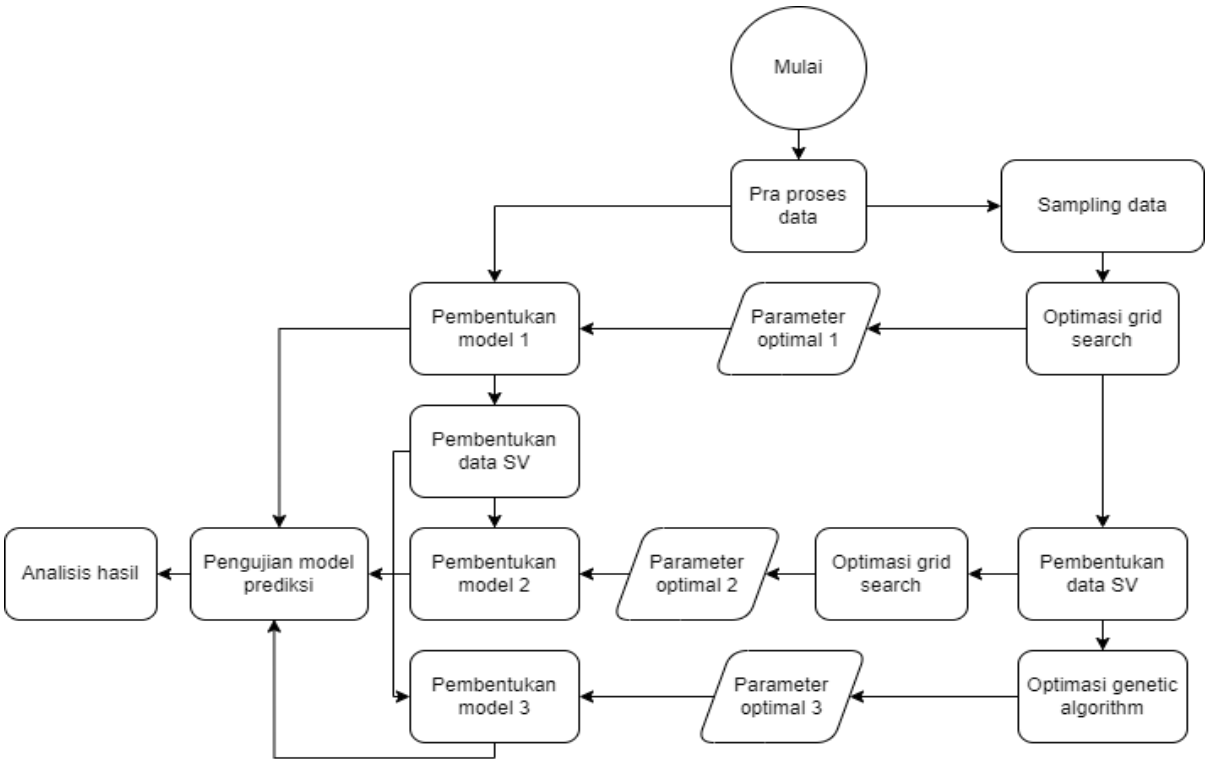
Copyright © 2024 **Lailan Sahrina Hasibuan**, Page 345
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 344-353
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5355

antara kelas target dan non-target. Oleh karen itu, pada penelitian ini menggunakan normalisasi min-max. Formula (1) di bawah menunjukkan rumus matematis normalisasi min-max.

$$X_{new} = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (1)$$

Pada formula di atas, X adalah nilai data yang akan dinormalisasi pada salah satu fitur, sementara $\min(X)$ dan $\max(X)$ adalah nilai minimum dan maksimum data pada fitur tersebut. Selanjutnya, X_{new} adalah nilai baru untuk data tersebut setelah dinormalisasi. Setelah proses normalisasi, nilai data baru, yaitu X_{new}, pada setiap fitur akan merentang antara 0 dan 1.



Gambar 2. Metode penelitian

2.3 Optimasi Parameter

Pemodelan pada penelitian ini menggunakan *Support Vector Machine* (SVM). SVM pertama kali diperkenalkan oleh Cortes dan Vapnik pada tahun 1995 [16]. SVM menggunakan konsep *hyperplane* dalam memisahkan data ke dalam kelas-kelas yang berbeda. Perbedaan *hyperplane* pada SVM adalah *hyperplane* dibentuk sedemikian rupa sehingga data-data pada kelas yang berbeda terpisah oleh jarak yang maksimum. Data yang yang membentuk *hyperplane* atau jarak tersebut adalah disebut sebagai data *support vector*. Selain itu, SVM mengusung konsep baru untuk klasifikasi, yaitu toleransi terhadap salah mengklasifikasikan data. Hal ini sesuai dengan konsep di dunia nyata bahwa sering kali suatu dataset tidak dapat dipisahkan secara tegas [16][17].

Pada penelitian ini, kernel yang digunakan pada SVM adalah *Radial Basis Function* (RBF) yang memiliki parameter gamma. Oleh karena itu, parameter yang perlu dioptimasi pada penelitian ini adalah C dan gamma.

Performa SVM dalam melakukan klasifikasi data sangat dipengaruhi oleh nilai parameternya [18][19]. SVM memiliki dua jenis parameter yaitu, parameter SVM dan parameter kernel. Kernel yang digunakan pada penelitian ini adalah *Radial Basis Function* (RBF) yang memiliki parameter gamma. Oleh kare itu, parameter yang perlu dioptimasi adalah C dan gamma. Parameter C mengatur toleransi kesalahan yang diperbolehkan pada proses klasifikasi, sementara parameter gamma mengatur pengaruh suatu sampel data terhadap bentuk *hyperplane*.

Optimasi parameter ini dilakukan dengan metode *grid search* dan *genetic algorithm*. Tahap ini menggunakan data sebanyak 5% dari data asli, karena kompleksitas SVM adalah $O(n^3)$. Kompleksitas tersebut menyatakan bahwa banyaknya komputasi yang perlu dilakukan SVM sebanding dengan banyaknya data yang digunakan dipangkatkan dengan tiga, sehingga pelatihan model dengan data yang relatif banyak dapat memakan waktu yang sangat lama.

Optimasi *gride search* diterapkan pada dua jenis dataset yaitu dataset 5% yang diseimbangkan dengan ADASYN dan dataset *support vector* yang seimbang juga. Sementara optimasi dengan *genetic algorithm* diterapkan pada dataset *support vector* yang seimbang saja. Proses optimasi parameter dengan metode *grid search* diterapkan dengan menggunakan konsep 10-fold cross validation. Pemilihan parameter terbaik berdasarkan nilai AUC (Area Under ROC). Tabel 1 menunjukkan kombinasi dataset dan teknik optimasi yang digunakan.

Copyright © 2024 **Lailan Sahrina Hasibuan**, Page 346
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 344-353
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5355

Tabel 1. Kombinasi metode optimasi dan dataset

Model	Dataset	Teknik optimasi
Model 1	Dataset 5% seimbang	*Grid search*
Model 2	Support vector Model 1 seimbang	*Grid Search*
Model 3	Support vector model 1 seimbang	*Genetic Algorithm*

2.4 Pembentukan Dataset

Machine learning pada umumnya, termasuk SVM, memiliki kerentanan terhadap dataset yang tidak seimbang. Model yang dilatih menggunakan dataset tidak seimbang akan cenderung mengelompokkan data ke dalam kelas mayor, yaitu kelas dengan jumlah data lebih banyak. Kenyataanya, kebanyakan kasus di dunia nyata menunjukkan bahwa sering kali kelas minorlah yang penting untuk diidentifikasi, misalnya pada kasus identifikasi kanker [20], bidang bioinformatika untuk identifikasi SNPs (*Single Nucleotide Polymophisms*) [21], keamanan jaringan untuk identifikasi serangan [22]. Oleh karena itu, pembentukan model harus ditur sedemikian rupa sehingga model mampu mengenali data-data yang berasal dari kelas minor. Pada penelitia ini, cara yang digunakan adalah membentuk dataset seimbang yang akan digunakan sebagai data latih pada proses pembuatan model.

Terdapat dua jenis dataset yang digunakan pada penelitian ini, yaitu dataset awal yang diseimbangkan menggunakan ADASYN dan dataset *support vector* yang diseimbangkan menggunakan ADASYN juga. ADASYN merupakan metode resampling yang pertama kali diperkenalkan pada tahun 2008. Teknik ini didasari oleh teknik resampling SMOTE [23], namun dengan menambahkan proses adaptif dalam pembangkitan data sintetis berdasarkan tingkat kompelksitas data minor. Semakin kompleks sifat suatu data minor, semakin banyak data sintetis yang dibangkitkan untuk data tersebut [24]. Kompleksitas data minor diukur berdasarkan sifat tetangga terdekatnya, jika data minor tersebut dikelilingi oleh banyak data mayor maka kompleksitasnya tinggi, begitu juga sebaliknya. Gambar 3 di bawah ini menunjukkan algoritma ADASYN yang diperkenalkan oleh [24].

Sementara untuk dataset kedua dibentuk dengan menggunakan data *support vector* (SV) yang diperoleh dari model pertama. Pemilihan data SV didasarkan pada teori bahwasanya *support vector* adalah data yang membentuk *hyperplane*, sehingga posisinya berada antara daerah kelas mayor dan minor. Penentuan kelas data selain data SV ditentukan berdasarkan jarak data tersebut terhadap data SV. Oleh karena itu, pada penelitian ini diasumsikan bahwa penggunaan dataset SV memiliki potensi tinggi untuk meningkatkan performa model. Hal ini didukung oleh hasil penelitian yang pernah dilakukan sebelumnya [25]. Dataset SV ini selanjutnya diseimbangkan menggunakan teknik resampling ADASYN. Teknik ADASYN diimplementasikan dengan bantuan library smotefamily.

Input
Training data set D_{tr} with m samples $\{x_i, y_i\}$, $i = 1, \ldots, m$, where x_i is an instance in the n dimensional feature space X and $y_i \in Y = \{1, -1\}$ is the class identity label associated with x_i. Define m_s and m_l as the number of minority class examples and the number of majority class examples, respectively. Therefore, $m_s \leq m_l$ and $m_s + m_l = m$.

Procedure
Calculate the degree of class imbalance:

$$d = \frac{m_s}{m_l}, where\ d \in (0,1]$$

If $d < d_{th}$ then (d_{th} is a preset threshold for the maximum tolerated degree of class imbalance ratio):

(a) Calculate the number of synthetic data examples that need to be generated for the minority class:

$$G = (m_l - m_s) \times \beta, where\ \beta \in [0,1]$$

β is a parameter used to specify the desired balance level after generation of the synthetic data. $\beta = 1$ means a fully balanced data set is created after the generalization process.

(b) For each example $x_i \in$ minority class, find K *nearest neighbors* based on the Euclidean distance in n dimensional space, and calculate the ratio r_i defined as:

$$r_i = \frac{\Delta_i}{K}, i = 1, \ldots, m_s$$

where Δ_i is the number of examples in the K *nearest neighbors* of x_i that belong to the majority class, therefore $r_i \in [0, 1]$

(c) Normalize r_i according to $\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i$, so that $\hat{r}_i$ is a density distribution ($\sum_i \hat{r}_i = 1$)

(d) Calculate the number of synthetic data examples that need to be generated for each minority example x_i:

$$g_i = r_i \times G$$

where G is the total number of synthetic data examples that need to be generated for the minority class

(e) For each minority class data example x_i, generate g_i synthetic data examples according to the following steps:
Do the **Loop** from 1 to g_i:

i. Randomly choose one minority data example, g_{zi}, from the K *nearest neighbors* for data x_i.

ii. Generate the synthetic data example:

$$s_i = x_i + (x_{zi} - x_i) \times \lambda$$

where $(x_{zi} - x_i)$ is the difference vector in n dimensional spaces, and λ is a random number: $\lambda \in [0, 1]$.

End **Loop**

Gambar 3. Algoritma ADASYN dari [24]

Copyright © 2024 **Lailan Sahrina Hasibuan**, Page 347
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 344-353
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5355

2.5 Pembentukan Model

Model yang dibentuk pada penelitian ini terdiri atas tiga buah model SVM. Parameter yang digunakan adalah parameter optimal yang diperoleh dari optimasi parameter pada tahap sebelumnya. Model SVM pertama dibangun menggunakan dataset awal yang diseimbangkan menggunakan ADASYN. Pembentukan model SVM kedua menggunakan dataset *support vector* yang telah diseimbangkan menggunakan ADASYN, data SV diperoleh dari model SVM yang pertama. Pembentukan model SVM ketiga menggunakan dataset SV juga dengan parameter yang diperoleh dari optimasi menggunakan *genetic algorithm*. Kombinasi dataset dan optimasi parameter tertera pada Tabel 1 subbab 3.2.

2.6 Evaluasi Model

Evaluasi model dilakukan dengan menguji model untuk memprediksi data uji. Hasil evaluasi dilihat matriks konfusi dan memperhatikan nilai akurasi, sensitivitas dan spesifisitas [26]. Selain dari performa yang dihasilkan, lamanya waktu komputasi juga menjadi bahan evaluasi model. Dari ketiga model tersebut dievaluasi dengan membandingkan baiknya performa dan kecepatan waktu yang dihasilkan. Gambar 3 menunjukkan metrik yang digunakan untuk mengevaluasi performa model.

Tabel 2. Matriks konfusi

	Aktual	
Prediksi	P	N
P	TP	FP
N	FN	TN

$$\textit{True Positive Rate (TPR)} = \frac{TP}{(TP+FN)} \quad (1)$$

$$\textit{False Positive Rate (FPR)} = \frac{FP}{(TN+FP)} \quad (2)$$

$$\textit{Precision} = \frac{TP}{(TP+FP)} \quad (3)$$

$$\textit{Recall} = TPR$$

$$\textit{Sensitivity} = TPR$$

$$\textit{Specificity} = \frac{TN}{(TN+FP)} \quad (4)$$

$$F_{measure} = \frac{2*Precision*Recall}{(Precision+Recall)} \quad (5)$$

$$\text{Akurasi} = \frac{TP+TN}{(TP+FP+FN+TN)} \quad (6)$$

3. HASIL DAN PEMBAHASAN

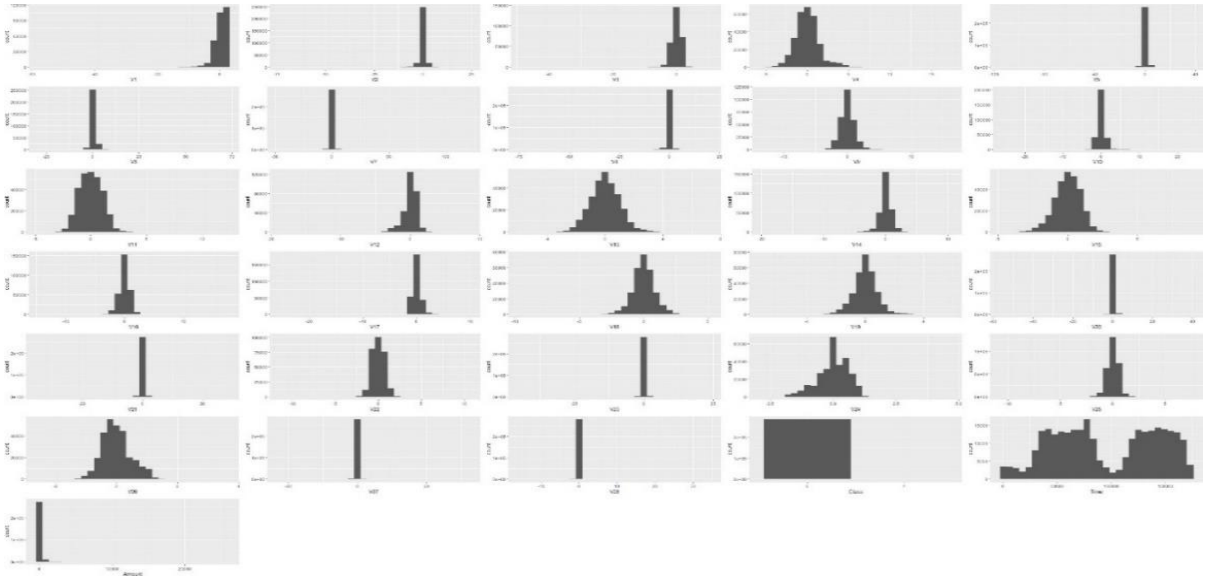
3.1 Praproses Data

Data asli yang diperoleh dari kaggle [11] yang diolah sebelumnya dilakukan eksplorasi terlebih dahulu agar dapat dianalisis tahapan praproses yang diperlukan pada data. Pertama dilakukan eksplorasi mengenai keberadaan missing value dengan fungsi sum() dan is.na(). Keluaran yang dihasilkan menunjukkan angka 0, hal ini berarti tidak terdapat *missing value* pada data. Pengecekan *missing value* dilakukan karena SVM tidak dapat bekerja pada dataset yang memiliki *missing value*, sehingga jika terdapat data yang *missing value* maka data data tersebut harus dihapus. Pada bahasa R, *library* yang digunakan untuk menjalankan SVM adalah e1071 dengan fungsi *train* untuk melatih model. Fungsi *train* ini telah diatur untuk menghapus data sampel yang mengandung *missing value* pada atributnya.

Selain keberadaan *missing value*, performa SVM juga dipengaruhi oleh rentangan nilai pada atribut-atributnya, atribut yang memiliki rentangan yang luas akan memiliki pengaruh yang relatif lebih kuat dibandingkan atribut yang rentangannya relatif lebih sempit. Grafik pada Gambar 4 menunjukkan bahwa dataset ini memiliki atribut-atribut yang rentangan nilainya berbeda-beda. Oleh karena itu, proses normalisasi nilai pada atribut-atribut dataset juga diperlukan, sehingga setiap atribut memiliki rentangan yang seragam. Pada penelitian ini, proses normalisasi dilakukan menggunakan metode min-max normalization sesuai pada formula (1) subbab 3.1. Setelah proses normalisasi setiap atribut memiliki rentangan nilai yang sama yaitu [0,1] [15]. Atribut ke-31 yang menyimpan informasi kelas data, normal atau fraud, tidak dinormalisasi karena atribut ini hanya memiliki dua kemungkinan kelas dan dikode dengan angka 0 dan 1 untuk kelas normal dan kelas fraud.

Copyright © 2024 **Lailan Sahrina Hasibuan**, Page 348
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 344-353
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5355



Gambar 4. Grafik persebaran dataset kartu kredit

3.2 Optimasi Parameter

Pencarian nilai parameter yang optimum dilakukan pada data sampling sebesar 5% dari dataset awal, yaitu sebanyak 18374 data yang terdiri atas 18355 data normal dan sisanya 39 data merupakan fraud. Penggunaan data 5% pada proses pencarian parameter optimum untuk menghindari lamanya proses pencarian ini karena besarnya ukuran data asli. Teknik penggunaan data sampel kecil untuk pencarian parameter optimum seperti ini juga dilakukan oleh [27] dan [28]. Selanjutnya, data 5% tersebut diseimbangkan menggunakan teknik ADASYN, sehingga jumlahnya seimbang antara transaksi normal dan fraud. Banyaknya data pada dataset 5% seimbang adalah 36671, yang terdiri dari 18335 transaksi normal dan 18336 transaksi fraud.

Dataset yang seimbang ini digunakan untuk mencari parameter optimum untuk model SVM pertama. Dari model SVM pertama yang optimal, data-data pembentuk *hyperplane* diekstrak dan dijadikan sebagai dataset baru. Dataset ini berjumlah 2797 transaksi yang terdiri atas 1857 transaksi normal dan 940 transaksi fraud. Berdasarkan jumlah ini, dapat dilihat bahwa terjadi penurunan ketidakseimbangan data antara data awal dan data *support vector* dari 1:578 menjadi 1:1.98. Selain itu, jumlah data *support vector* juga mengalami penurunan sebesar 9.89 kali dibandingkan data aslinya. Penurunan jumlah data ini berarti terjadi penurunan waktu komputasi pada proses pelatihan. Data *support vector* ini selanjutnya diseimbangkan menggunakan ADASYN sehingga jumlahnya menjadi 3709 transaksi, yang terdiri dari 1857 transaksi normal dan 1852 transaksi fraud. Data *support vector* yang seimbang tersebut digunakan untuk mencari parameter yang optimal menggunakan metode *grid search* dan *genetic algorithm*. Tabel 3 menunjukkan banyaknya data yang digunakan untuk proses optimasi parameter optimum.

Tabel 3. Banyaknya transaksi normal dan fraud pada setiap dataset kecil

Model pelatihan	Banyaknya data	Jumlah normal	Jumlah fraud
Grid search + data asli (model pertama)	36671	18335	18336
Grid search + data *support vector* (model kedua)	3709	1857	1852
Genetic algorithm + data *support vector* (model ketiga)	3709	1857	1852

Area pencarian nilai parameter C menggunakan *grid search* adalah 0.1, 1, 10, 100, dan 1000, sementara untuk nilai gamma adalah 0.0001, 0.001, 0.01, 0.1, dan 1. Metrik yang dievaluasi adalah AUC (Area Under ROC). Tabel 4 menunjukkan evaluasi model untuk setiap kombinasi parameter C dan gamma. Dari tabel tersebut dapat dilihat bahwa nilai parameter terbaik untuk dataset asli yang seimbang maupun dataset *support vector* yang seimbang adalah 10 untuk parameter C dan 0.1 untuk parameter gamma berdasarkan nilai AUC. Berdasarkan nilai-nilai pada metrik-metrik lainnya yaitu AUC, sensitivitas dan spesifisitas dapat dilihat bahwa nilai-nilai tersebut tidak berbeda jauh. Namun, karena nilai-nilai ini mewakili dataset yang berukuran kecil yaitu 5% saja dari data asli, maka perbedaan nilai yang sedikit ini bisa saja memiliki peluang memberikan hasil yang berbeda ketika diterapkan pada data aslinya yang berukuran besar. Oleh karena itu, pada penelitian ini tidak dilakukan pengujian beda nyata untuk memeriksa perbedaan nilai secara signifikan menurut statistika.

Tabel 4. Nilai kebaikan model pada pelatihan *grid search*

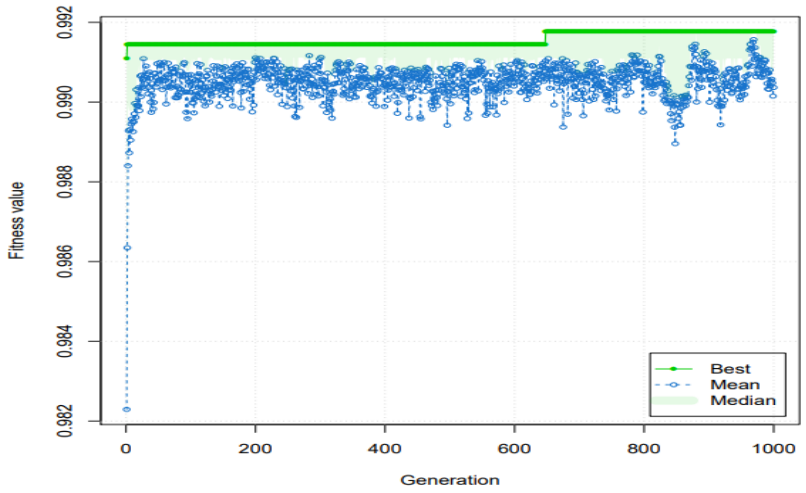
Model	Gamma	C	AUC	Sensitivitas	Spesifisitas
Grid search raw data seimbang	**0.1**	0.1	0.9999136	0.9991143	0.9962158
		1	0.9999964	0.9996779	0.9989533

Copyright © 2024 **Lailan Sahrina Hasibuan**, Page 349
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 344-353
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5355

		10	**0.9999977**	**0.9996779**	**0.9993559**
		100	0.9999431	0.9996779	0.9989533
		1000	0.9999432	0.9996779	0.9989533
Grid search data SV seimbang	**0.1**	0.1	0.9928204	0.9849587	0.9537263
		1	0.9976468	0.9901961	0.9791401
		10	**0.9991964**	**0.9928105**	**0.9863212**
		100	0.9978464	0.9915033	0.9856634
		1000	0.9964230	0.9901961	0.9863170

Pencarian nilai optimum menggunakan algoritma *genetic algorithm* dilakukan pada area [0.0001, 0.1] hingga [1, 1000], yang mana dimensi pertama menyatakan nilai untuk parameter gamma dan dimensi kedua menyatakan parameter C. Ukuran populasi adalah 100, maksimum iterasi adalah 1000, peluang *crossover* 0.8 dan peluang mutasi adalah 0.1. Gambar 5 menunjukkan grafik perubahan nilai fitnes, AUC, terhadap iterasi selama proses pencarian nilai parameter optimum menggunakan *genetic algorithm*. Berdasarkan grafik tersebut dapat dilihat bahwa proses pencarian berhenti setelah iterasi mencapai nilai maksimum yaitu 1000. Pada iterasi ke-1000 ini, AUC yang diperoleh adalah 0.992.



Gambar 5. Grafik nilai fitnes dan iterasi pada optimasi parameter menggunakan *genetic algorithm*.

Perbandingan antara performa model (AUC), waktu pencarian, nilai C, nilai gamma tersaji pada Tabel 5. Berdasarkan data-data tersebut dapat disimpulkan bahwa nilai AUC untuk semua model hampir sama, namun waktu komputasi yang dibutuhkan oleh *genetic algorithm* meningkat secara eksponensial jika dibandingkan dengan *grid search* baik pada dataset asli yang seimbang maupun dataset *suppport vector*. Algoritma GA membutuhkan waktu $4.9 * 10^6$ menit yang setara dengan 3 hari 9 jam, sementara *grid search* hanya membutukan waktu 8 menit dan 8.5 menit. Tingginya waktu komputasi yang dibutuhkan oleh *genetic algorithm* karena pada setiap iterasi akan dibangkitkan 100 populasi yang merupakan kombinasi nilai C dan gamma. Setiap kombinasi nilai C dan gamma digunakan untuk satu model SVM dan diuji nilai fitnesnya (AUC). Dengan prosedur seperti ini, *genetic algorithm* membangkitkan 100 model pada setiap iterasi, sementara jumlah iterasi adalah 1000, sehingga total model SVM yang dibangkitkan adalah 10^5 model SVM. Jumlah ini sangat kontras dengan banyaknya model yang dibangkitkan oleh *grid search* yang hanya membangkitkan 25 model SVM.

Tabel 5. Parameter Optimal SVM

Model pelatihan	C	Gamma	AUC Model	Waktu (menit)
Grid search + data asli seimbang (model pertama)	10	0.1	0.9999977	8.077
Grid search + data sv seimbang (model kedua)	10	0.1	0.9991964	8.528
Genetic algorithm + data support vector (model ketiga)	49.742	0.157	0.992	$4.9 * 10^6$ (3.39 hari)

3.3 Pembentukan Dataset

Dataset yang diperoleh dari kaggle [11] memiliki 284807 jumlah transaksi yang terdiri dari 284315 transaksi normal dan 492 transaksi fraud. Dataset ini dibagi menjadi dua dataset yaitu dataset latih dan dataset uji dengan perbandingan 70:30, sehingga dataset latih terdiri atas 192933 transaksi dan dataset uji terdiri atas 91874 transaksi. Setiap dataset memiliki distribusi yang sama antara transaksi fraud dan normal dengan data aslinya, yaitu 1:578. Dataset uji yang berjumlah 91874 dibiarkan tetap tidak seimbang sesuai kondisi aslinya. Hal ini dilakukan agar dataset uji memiliki karakteristik yang sama dengan data aslinya sehingga proses pengujian model bisa menggambarkan kondisi sebenarnya pada dunia nyata.

Copyright © 2024 **Lailan Sahrina Hasibuan**, Page 350
This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 344-353
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5355

Data latih yang berjumlah 192933 transaksi terdiri atas 192613 transaksi normal dan 320 transaksi fraud diseimbangkan menggunakan algoritma ADASYN sehingga jumlah datanya menjadi 385199 transaksi yang terdiri atas 192601 transaksi normal dan 192598 transaksi fraud. Dataset seimbang ini digunakan untuk membangun model SVM yang pertama. Selanjutnya dari model SVM tersebut, data *support vector* pembentuk *hyperplane* diekstrak. Dataset ini memiliki jumlah 9885 transaksi yang terdiri atas 7001 transaksi normal dan 2884 transaksi fraud. Ketidakseimbangan pada data *support vector* mengalami penurunan dibandingkan ketidakseimbangan pada data asli yaitu dari 1:578 menjadi 1: 2.43 atau menurun sekitar 237 kali. Tabel 6 menunjukkan jumlah transaksi fraud dan normal untuk semua dataset.

Tabel 6. Jumlah transaksi fraud dan normal pada setiap dataset

Dataset	Banyaknya data	Jumlah normal	Jumlah fraud	Perbandingan fraud:normal
Data awal	284807	284315	492	1:578
Data latih	192933	192613	320	1:602
Data latih awal seimbang	385199	192601	192598	1:1
Data *support vector*	9885	7001	2884	1:2.43
Data latih *support vector* seimbang	14037	7001	7036	1:1
Data uji	91874	91702	172	1:533

3.4 Pembentukan Model Pertama

Pembentukan model yang pertama menggunakan dataset awal yang seimbang yang berjumlah 385199 transaksi. Parameter C dan gamma yang digunakan 10 dan 0.1. Pelatihan model menggunakan dataset ini membutuhkan waktu 4.29 jam (4 jam 17 menit). Model SVM yang dihasilkan pada pelatihan ini diuji pada dataset uji yang memiliki jumlah 91874 transaksi. Hasil pengujian ditampilkan menggunakan matriks konfusi yang tersaji pada Tabel 7. Berdasarkan matriks konfusi tersebut, model SVM memperoleh sensitivitas sebesar 99% dan spesifitas sebesar 99%.

Tabel 7. Matriks konfusi pengujian model pertama

	Fraud (Aktual)	Normal (Aktual)
Fraud (Prediksi)	171	4
Normal (Prediksi)	1	91698

3.5 Pembentukan Model Kedua

Pembentukan model kedua sama dengan pembentukan model sebelumnya, namun menggunakan dataset yang berbeda. Dataset yang digunakan pada pemodelan kedua adalaha dataset *support vector* dari model pertama yang telah diseimbangkan. Dataset ini memiliki 14037 transaksi. Parameter yang digunakan adalah 10 dan 0.1 untuk C dan gamma. Pelatihan model menggunakan dataset ini membutuhkan waktu 4.32 jam (4 jam 19 menit). Setelah model berhasil dibentuk, model tersebut diuji terhadap dataset uji yang sudah disiapkan. Hasil pengujian tersaji dalam matriks konfusi pada Tabel 8. Berdasarkan nilai-nilai pada matriks konfusi, model SVM yang kedua memperoleh sensitivitas sebesar 99% dan spesifisitas juga 99%.

Tabel 8. Matriks konfusi pengujian model kedua

	Fraud (Aktual)	Normal (Aktual)
Fraud (Prediksi)	171	20
Normal (Prediksi)	1	91682

3.6 Pembentukan Model Ketiga

Pembentukan model yang ketiga sama dengan pembentukan model yang kedua, dataset yang digunakan adalah sama yaitu dataset *support vector* yang telah diseimbangkan dari model SVM pertama yang pertama. Namun, nilai parameter yang digunakan berbeda yaitu 49.742 untuk parameter C dan 0.157 untuk parameter gamma. Pelatihan model menggunakan dataset ini membutuhkan waktu 82.77 jam (3 hari 10 jam 42 menit). Setelah model dibentuk, model ketiga ini juga diuji pada dataset uji yang sama. Hasil pengujiannya ditampilkan dalam bentuk matriks konfusi yang tersaji pada Tabel 9. Berdasarkan nilai-nilai pada matriks konfusi, model SVM yang ketiga memperoleh sensitivitas sebesar 99% dan spesifisitas juga 99%.

Tabel 9. Matriks konfusi model ketiga

	Fraud (Aktual)	Normal (Aktual)
Fraud (Prediksi)	171	12
Normal (Prediksi)	1	91690

This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 344-353
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5355

3.7 Analisis Hasil

Pada tabel dapat dilihat bahwa model pertama dan kedua memiliki tingkat akurasi, sensitivitas, serta spesifisitas yang sama. Waktu pelatihan yang dibutuhkan antara kedua model pun tidak terlalu jauh, namun waktu komputasi berpengaruh terhadap banyaknya data serta rentang parameter yang diatur. Semakin banyak data yang digunakan maka semakin banyak juga waktu komputasi yang dibutuhkan. Model ketiga menunjukkan perbedaan hasil dari model pertama dan kedua. Karena permasalahan utama pada klasifikasi kartu kredit ini adalah datanya yang tidak seimbang, maka parameter kebaikan yang lebih diperhatikan adalah sensitivitas. Jika dilihat dari seberapa baik model dapat mengklasifikasikan data fraud ke dalam kelas yang tepat serta membutuhkan waktu komputasi yang rendah, maka model pertama lebih unggul daripada model lainnya. Hal ini juga berarti ADASYN dan optimasi parameter menggunakan grid search dengan matriks evaluasi ROC dapat menangani permasalahan ketidakseimbangan data, hal ini sesuai dengan penelitian [29][30].

Tabel 9. Perbandingan performa model

Model	Akurasi	Sensitivitas	Spesifisitas	Waktu Komputasi (jam)
Model 1	99%	99%	99%	4.29
Model 2	99%	99%	99%	4.32
Model 3	99%	99%	99%	82.77

4. KESIMPULAN

Penelitian ini berfokus kepada pembangunan model klasifikasi yang dapat menangani data tidak seimbang sehingga dapat menghasilkan model dengan performa baik dan waktu komputasi singkat. Terdapat 3 model yang dibandingkan, model pertama yaitu pemodelan ADASYN menggunakan pembelajaran mesin SVM dengan grid search sebagai optimasi parameter, merupakan model yang paling baik diantara kedua model lainnya. Dengan lamanya waktu pelatihan model selama 4 jam 17 menit, model ini memiliki tingkat akurasi sebesar 99%, spesifisitas 99%, dan sensitivitas sebesar 99%.

REFERENCES

[1] Bank Indonesia, "Laporan Kelembagaan Bank Indonesia Triwulan 2 2022," Jakarta, 2022.
[2] S P Maniraj, Aditya Saini, Shadab Ahmed, and Swarna Deep Sarkar, "Credit Card Fraud Detection using Machine Learning and Data Science," *Int. J. Eng. Res.*, vol. 08, no. 09, pp. 110–115, 2019.
[3] I. Benchaji, S. Douzi, and B. El Ouahidi, *Using genetic algorithm to improve classification of imbalanced datasets for credit card fraud detection*, vol. 66. Springer International Publishing, 2019.
[4] E.-A. MINASTIREANU and G. MESNITA, "An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection," *Inform. Econ.*, vol. 23, no. 1/2019, pp. 5–16, 2019.
[5] N. Yousefi, M. Alaghband, and I. Garibay, "A Comprehensive Survey on Machine Learning Techniques and User Authentication Approaches for Credit Card Fraud Detection," pp. 1–27, 2019.
[6] J. I.-Z. Chen and K.-L. Lai, "Deep Convolution Neural Network Model for Credit-Card Fraud Detection and Alert," *J. Artif. Intell. Capsul. Networks*, vol. 3, no. 2, pp. 101–112, 2021.
[7] D. Dharmawan, "Deteksi Anomali Pada Transaksi Kartu Kredit Menggunakan Adaptive Synthetic Resampling(ADASYN) dan Support Vector Machine (SVM)," 2022.
[8] C. Li, N. Ding, Y. Zhai, and H. Dong, "Comparative study on credit card fraud detection based on different support vector machines," *Intell. Data Anal.*, vol. 25, no. 1, pp. 105–119, 2021.
[9] Z. Pooranian, M. Shojafar, R. Tavoli, M. Singhal, and A. Abraham, "A hybrid metaheuristic algorithm for job scheduling on computational grids," *Inform.*, vol. 37, no. 2, pp. 157–164, 2013.
[10] B. Alhijawi and A. Awajan, "Genetic algorithms: theory, genetic operators, solutions, and applications," *Evol. Intell.*, vol. 17, no. 3, pp. 1245–1256, Jun. 2023.
[11] "Credit Card Fraud Detection." [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data. [Accessed: 16-Jun-2024].
[12] N. G. Ramadhan, "Comparative Analysis of ADASYN-SVM and SMOTE-SVM Methods on the Detection of Type 2 Diabetes Mellitus," *Sci. J. Informatics*, vol. 8, no. 2, pp. 276–282, 2021.
[13] M. A. Umar and C. Zhanfang, "Effects of Feature Selection and Normalization on Network Intrusion Detection," pp. 1–25, 2020.
[14] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, vol. 97, p. 105524, Dec. 2020.
[15] H. Henderi, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," *IJIIS Int. J. Informatics Inf. Syst.*, vol. 4, no. 1, pp. 13–20, 2021.
[16] C. Cortes and V. Vapnik, "Support-Vector Networks," vol. 297, pp. 273–297, 1995.
[17] A. S. Desuky and S. Hussain, "An Improved Hybrid Approach for Handling Class Imbalance Problem," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 3853–3864, 2021.
[18] A. Alsarhan, M. Alauthman, E. Alshdaifat, A. R. Al-Ghuwairi, and A. Al-Dubai, "Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, no. 5, pp. 6113–6122, 2023.
[19] D. J. Kalita, V. P. Singh, and V. Kumar, "A novel adaptive optimization framework for SVM hyper-parameters tuning in

This Journal is licensed under a Creative Commons Attribution 4.0 International License

Building of Informatics, Technology and Science (BITS)
Volume 6, No 1, Juni 2024 Page: 344-353
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v6i1.5355

non-stationary environment: A case study on intrusion detection system," *Expert Syst. Appl.*, vol. 213, p. 119189, Mar. 2023.
[20] C. Y. Guo and Y. C. Chou, "A novel machine learning strategy for model selections - Stepwise Support Vector Machine (StepSVM)," *PLoS ONE*, vol. 15, no. 8 August. 2020.
[21] L. S. Hasibuan, W. A. Kusuma, and W. B. Suwamo, "Identification of single nucleotide polymorphism using support vector machine on imbalanced data," *Proc. - ICACSIS 2014 2014 Int. Conf. Adv. Comput. Sci. Inf. Syst.*, pp. 375–379, 2014.
[22] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *J. Big Data*, vol. 8, no. 1, 2021.
[23] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," 2002.
[24] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *Proc. Int. Jt. Conf. Neural Networks*, no. 3, pp. 1322–1328, 2008.
[25] L. S. Hasibuan, S. Nabila, N. Hudachair, and M. A. Istiadi, "Evaluation of F-Measure and Feature Analysis of C5.0 Implementation on Single Nucleotide Polymorphism Calling," *Indones. J. Artif. Intell. Data Min.*, vol. 1, no. 1, p. 1, 2018.
[26] G. Varoquaux and O. Colliot, "Evaluating Machine Learning Models and Their Diagnostic Value," *Neuromethods*, vol. 197, pp. 601–630, 2023.
[27] B. D. O'Fallon, W. Wooderchak-Donahue, and D. K. Crockett, "A support vector machine for identification of single-nucleotide polymorphisms from next-generation sequencing data," *Bioinformatics*, vol. 29, no. 11, pp. 1361–1366, 2013.
[28] P. Yang, Z. Zhang, B. B. Zhou, and A. Y. Zomaya, "Sample Subset Optimization for Classifying Imbalanced Biological Data," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6635 LNAI, no. PART 2, pp. 333–344, 2011.
[29] H. Mohammedqasim, A. Ahmed Jasim, A. Mohammedqasem, and O. Ata, "Enhancing Predictive Performance in Covid-19 Healthcare Datasets: a Case Study Based on Hyper Adasyn Over-Sampling and Genetic Feature Selection," *J. Eng. Sci. Technol.*, vol. 19, no. 2, pp. 598–617, 2024.
[30] A. F. Pulungan, D. Selvida, and A. I. Silitonga, "Combination of ADASYN and Random Forest for Classification of Imbalanced Lung Cancer Dataset," *AIP Conf. Proc.*, vol. 2987, no. 1, Apr. 2024.

Copyright © 2024 **Lailan Sahrina Hasibuan**, Page 353
This Journal is licensed under a Creative Commons Attribution 4.0 International License