Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 5 Iss. 3 July 2025, pp: 886-895
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Implementation of Feature Selection Using Boruta to Improve the Accuracy of the Lapser Prediction Model

Implementasi *Feature Selection* Menggunakan *Boruta* untuk Peningkatan Akurasi Model *Lapser Prediction*

Mochamad Gilang Saputra[1*], Bagus Jati Santoso[2]

[1]Departemen Manajemen Teknologi, Institut Teknologi Sepuluh Nopember, Indonesia
[2]Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember, Indonesia

E-Mail: [1]6032231011@student.its.ac.id, [2]bagus@if.its.ac.id

Received Mar 13th 2025; Revised May 13th 2025; Accepted Jun 18th 2025; Available Online Jun 24th 2025, Published Jun 24th 2025
Corresponding Author: Mochamad Gilang Saputra

Abstract

Predicting lapsing customers is a major challenge in the highly competitive data service sector, exacerbated by the high costs associated with acquiring new customers. This study proposes a feature selection approach using Boruta to enhance the accuracy of the lapse prediction model, employing a wrapper technique on Random Forest. The lapse prediction modeling process utilizes the Gradient Boosting machine learning algorithm, analyzed both before and after Boruta feature selection. Experimental results demonstrate that Boruta effectively improves key metrics (accuracy, recall, and AUC). Following the application of Boruta, the Gradient Boosting model achieved an accuracy of 75.10%, a recall of 74.42%, and an AUC of 82.18%. Prior to using Boruta, the model recorded an accuracy of 71.74%, a recall of 68.74%, and an AUC of 77.77%. These findings confirm that the proposed approach can predict lapsing customers at an earlier stage, thereby assisting policymakers in formulating more effective customer retention strategies, minimizing potential losses, and strengthening market competitiveness.

Keywords: Boruta, Feature Selection, Gradient Boosting, Lapser, Machine Learning

Abstrak

Memprediksi pelanggan *lapser* menjadi tantangan utama di sektor layanan data yang kompetitif, disertai tingginya biaya akuisisi pelanggan baru. Penelitian ini mengusulkan pendekatan *feature selection* menggunakan *Boruta* untuk meningkatkan akurasi *model lapser*, dengan menerapkan teknik *wrapper* pada *Random Forest*. Proses *modeling lapser prediction* menggunakan *a*lgoritma *machine learning Gradient Boosting* yang dianalisis sebelum dan sesudah seleksi fitur *Boruta*. Hasil eksperimen pada data menunjukkan bahwa *Boruta* efektif dalam meningkatkan metrik utama (akurasi, *recall*, dan AUC). Model Gradient Boosting meraih akurasi hingga 75.10%, *recall* 74.42%, dan AUC 82.18% setelah menggunakan *Boruta*. Sebelum menggunakan B*oruta* nilai akurasi 71.74%, *recall* 68.74%, dan AUC hanya 77.77%. Temuan tersebut menegaskan bahwa pendekatan yang diusulkan dapat memprediksi *lapser* secara lebih dini, serta membantu penyusun kebijakan menyusun strategi retensi pelanggan yang lebih efektif, sehingga meminimalkan potensi kerugian dan memperkuat daya saing di pasar.

Kata Kunci: *Boruta, Feature Selection, Gradient Boosting, Lapser, Machine Learning*

1. PENDAHULUAN

Industri layanan data mengalami perkembangan yang sangat pesat dalam beberapa tahun terakhir, sejalan dengan peningkatan kebutuhan masyarakat akan konektivitas dan ragam aplikasi digital. Kompetisi pasar pun semakin ketat, mendorong para penyedia layanan untuk terus berinovasi dalam menambah fitur dan meningkatkan kualitas layanan. Meski demikian, di tengah persaingan yang sengit, tantangan mempertahankan pelanggan agar tidak berhenti berlangganan (*lapser*) tetap menjadi isu krusial. Menurut beberapa penelitian, biaya memperoleh pelanggan baru *(customer acquisition cost)* sering kali lebih tinggi daripada biaya yang diperlukan untuk mempertahankan pelanggan lama [1]. Oleh karena itu, mendeteksi pelanggan yang berpotensi *lapser* secara dini menjadi prioritas penting bagi perusahaan.

Penurunan jumlah pelanggan akibat *lapser* berdampak langsung pada pendapatan dan profitabilitas, sekaligus dapat merusak citra perusahaan di mata publik. Pelanggan yang mengalami pengalaman buruk cenderung menyebarkan kesan negatif, memengaruhi keputusan calon pelanggan lainnya [2]. Di sisi lain, pelanggan yang puas dan bertahan lebih lama seringkali berkontribusi terhadap peningkatan revenue melalui pembelian produk tambahan *(cross-selling)* atau perluasan layanan *(up-selling).* Dengan demikian, model prediksi *lapser* yang andal dapat mendorong terciptanya strategi retensi yang efektif, misalnya berupa penawaran khusus atau program loyalitas yang tepat sasaran [3].

Dalam ranah *machine learning*, prediksi *lapser* sering dihadapkan pada dua kendala teknis utama: Dimensi Data yang Tinggi sebagai Penyedia layanan data umumnya mengumpulkan beragam atribut pelanggan, mulai dari data demografis, pola perilaku, hingga riwayat. Jumlah atribut yang melimpah tidak selalu berbanding lurus dengan peningkatan performa model, bisa menambah risiko *overfitting* jika banyak atribut tidak relevan [4], [5]; Ketidakseimbangan Kelas *(Imbalanced Data)* adalah proporsi pelanggan yang *lapser* cenderung kecil dibandingkan pelanggan *non-lapser*. Akibatnya, model cenderung mendominasi prediksi ke kelas mayoritas, sehingga mengorbankan *recall* pada kelas minoritas. Situasi ini membuat banyak model gagal mendeteksi pelanggan yang benar-benar berpotensi untuk *lapser* [6].

Metode *feature selection* menjadi solusi potensial untuk mengatasi masalah pertama. Dalam hal ini, teknik *Boruta* menonjol karena memanfaatkan *Random Forest* untuk menilai kepentingan setiap fitur, membandingkannya dengan *"shadow features"* (salinan acak dari fitur asli). Fitur yang *consistently* lebih penting daripada bayangan fitur dianggap memiliki andil signifikan dalam memprediksi variabel target [7]. Hal ini membedakannya dari metode filter tradisional yang hanya meninjau statistik antar-fitur, tanpa memperhitungkan interaksi non-linear.

Penelitian ini berfokus pada *feature selection* metode *Boruta* untuk membangun model *lapser* yang akurat. Setelah fitur tidak relevan berhasil dieliminasi, algoritma *Gradient Boosting* dipilih sebagai classifier karena memiliki reputasi yang baik dalam menangani masalah non-linearitas, memanfaatkan mekanisme *ensemble* untuk meningkatkan performa prediksi [8], [9]. *Gradient Boosting* merupakan perkembangan dari *Boosting* klasik yang membangun pohon keputusan secara iteratif, di mana tiap pohon baru difokuskan untuk memperbaiki kesalahan pohon sebelumnya [8]. Selain Gradient Boosting, beberapa studi terkini memanfaatkan *XGBoost* [10], yang dibangun di atas mekanisme *tree boosting* serupa namun peneliti memilih *Gradient Boosting* untuk tahap awal karena dianggap lebih sederhana untuk diimplementasikan sekaligus cukup andal dalam memodelkan data lapser, selaras dengan penjelasan Hastie, Tibshirani, dan Friedman [11], pendekatan ansambel seperti *Boosting* dan *Random Forest* terus berkembang dengan dukungan teori statistik mutakhir, memungkinkan pemodelan non-linear yang lebih tangguh sekaligus menangani berbagai jenis data. Selain untuk data pelanggan, metode ensemble *boosting* juga telah digunakan dalam klasifikasi data satelit, seperti ditunjukkan oleh Ouchra H [12], menegaskan fleksibilitas pendekatan ini untuk beragam tipe dataset dan domain berbeda.

Dalam industri layanan data yang sangat kompetitif, kemampuan untuk memprediksi pelanggan yang berpotensi berhenti berlangganan (*lapser*) memiliki nilai strategis yang tinggi. Kehilangan pelanggan tidak hanya berdampak langsung pada penurunan pendapatan, tetapi juga meningkatkan biaya akuisisi untuk mendapatkan pelanggan baru, yang secara umum jauh lebih mahal dibanding mempertahankan pelanggan yang sudah ada [13]. Oleh karena itu, deteksi dini terhadap perilaku lapser memungkinkan perusahaan untuk melakukan intervensi proaktif melalui program retensi yang tepat sasaran, seperti personalisasi penawaran atau pemberian insentif. Terlebih dalam era digital saat ini, pelanggan memiliki banyak alternatif layanan yang tersedia secara instan, sehingga loyalitas menjadi lebih rapuh. Model prediksi yang akurat dapat membantu perusahaan merespons dengan cepat terhadap perubahan pola perilaku pelanggan sebelum mereka benar-benar berhenti berlangganan [14], [15]. Dalam konteks ini, pengembangan sistem prediktif berbasis *machine learning* menjadi sangat penting, tidak hanya sebagai alat analitik, tetapi juga sebagai fondasi untuk pengambilan keputusan bisnis yang lebih responsif dan *data-driven*.

Penelitian terdahulu yang memanfaatkan *Boruta* kerap menunjukkan peningkatan akurasi dan interpretabilitas model . Misalnya, Kursa dan Rudnicki [7] membuktikan bahwa *Boruta* dapat menemukan fitur-fitur penting pada data genomik yang sangat berdimensi tinggi. Dalam konteks *lapser prediction*, pemilihan fitur yang tepat semakin relevan karena data pelanggan cenderung heterogen meliputi perilaku online, transaksi finansial, hingga informasi sosial. Sebagai lanjutan dari beragam model klasik dan modern, [16] memberikan landasan komprehensif untuk memahami metode ensemble, regularisasi, dan teknik tuning seperti *boruta* yang relevan bagi skenario prediksi lapser yang rumit. Dengan demikian, memadukan *Boruta* dan *Gradient Boosting* diharapkan mampu meningkatkan keandalan model dalam mendeteksi pelanggan *lapser*.

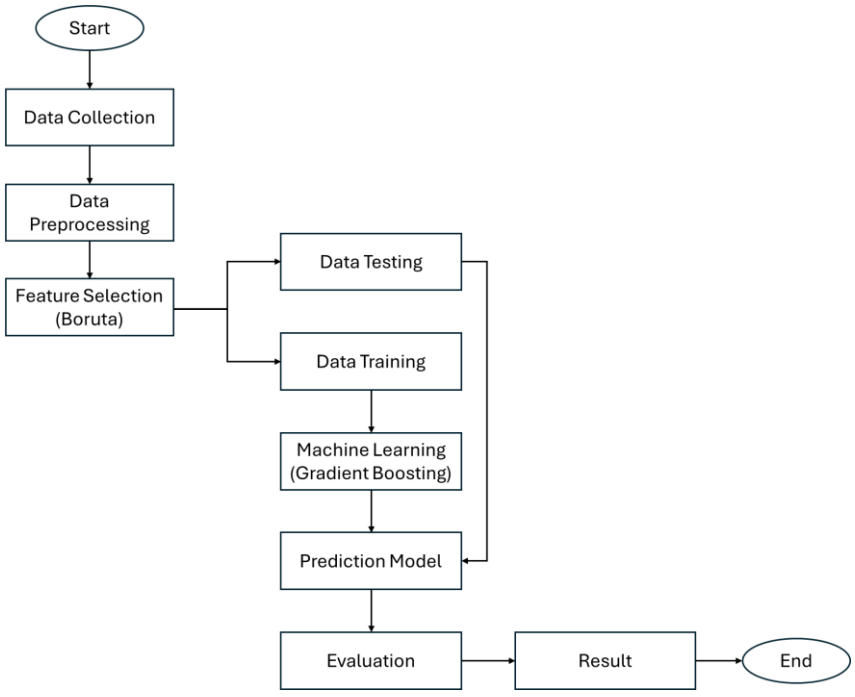
Secara spesifik, penelitian ini bertujuan untuk: Mengevaluasi efektivitas *Boruta* [17] dalam menyeleksi fitur yang relevan terhadap perilaku *lapser*; Menganalisis dampak terhadap performa model, terutama pada metrik *recall* dan AUC. Membandingkan kinerja *Gradient Boosting* sebelum dan sesudah proses seleksi fitur *Boruta*, ditinjau dari akurasi, *recall*, dan AUC.

Dengan adanya penelitian ini, diharapkan dapat memberikan panduan teknis bagi praktisi di sektor layanan data dalam menerapkan pipeline feature selection dan balancing data sebagai upaya mencegah *lapser*. Bagian berikutnya (Metodologi Penelitian) akan menjelaskan kerangka metodologis penelitian, mulai dari pengumpulan data, proses *preprocessing*, metode *Boruta*, hingga konfigurasi *Gradient Boosting* [18].

2. METODOLOGI PENELITIAN

Pada gambar 1 menguraikan secara menyeluruh mengenai data yang digunakan, proses pengumpulan data *(data collection)*, prosedur pra-pemrosesan *(data preprocessing)*, metode *feature selection* menggunakan *Boruta*, serta konfigurasi algoritma *Gradient Boosting* yang menjadi model utama dalam penelitian ini. Keseluruhan tahapan penelitian dirancang berdasarkan rekomendasi dan praktik terbaik yang tercermin dalam berbagai literatur, guna memastikan metode yang diterapkan memiliki landasan teoritis yang kuat serta relevan bagi permasalahan nyata di industri layanan data [1],[19].

Penggunaan algoritma *Boruta* dalam penelitian ini didasarkan pada keunggulannya dalam melakukan seleksi fitur secara menyeluruh dan statistik. *Boruta* merupakan metode *wrapper* berbasis *Random Forest* yang bekerja dengan menambahkan *shadow features*, yaitu salinan acak dari fitur asli, untuk mengevaluasi signifikansi tiap fitur. Fitur yang secara konsisten menunjukkan skor lebih tinggi dibanding *shadow features* akan dipertahankan [7]. Keunggulan *Boruta* terletak pada kemampuannya untuk mempertimbangkan interaksi *non-linear* antar fitur, menjadikannya lebih unggul daripada metode filter biasa seperti chi-square atau *information gain* [20]. Dalam konteks prediksi pelanggan lapser yang memiliki data berdimensi tinggi, pendekatan ini membantu mengurangi *noise*, meningkatkan interpretabilitas, dan mempercepat proses pelatihan model tanpa mengorbankan akurasi. Studi sebelumnya menunjukkan bahwa *Boruta* berhasil meningkatkan performa model klasifikasi pada domain seperti *churn prediction* dan *credit scoring* [21],[22]. Selain itu, integrasi *Boruta* dengan model *Gradient Boosting* terbukti menghasilkan peningkatan metrik utama, seperti akurasi, *recall*, dan AUC. Oleh karena itu, kombinasi *Boruta* dan *Gradient Boosting* dipilih dalam penelitian ini sebagai pendekatan yang seimbang antara akurasi, efisiensi, dan stabilitas model.



Gambar 1. Metododologi Penelitian

2.1. Deskripsi dan Pengumpulan Data (*Data Collection)*

Penelitian ini menggunakan *big data* yang berasal dari industri layanan data, sebagaimana dijelaskan oleh Chen dan Zhang [5], dengan karakteristik yang kompleks dan beragam. *Dataset* yang digunakan mencakup sekitar 191 atribut, yang diklasifikasikan ke dalam tiga kelompok utama. Pertama, atribut demografis seperti usia pelanggan dan wilayah transaksi. Kedua, atribut perilaku yang mencakup jumlah transaksi, frekuensi pengisian ulang, total konsumsi kuota data, serta durasi berlangganan layanan. Ketiga, label target berupa status lapser yang bersifat *biner*, yaitu 0 untuk pelanggan *non-lapser* dan 1 untuk pelanggan *lapser*, mengikuti definisi yang umum digunakan dalam konteks industri layanan data [2],[3]. Secara keseluruhan, data yang dianalisis berjumlah sekitar 18 juta entri, dengan proporsi pelanggan lapser sebesar

26%. Untuk keperluan pelatihan model, data dibagi dengan rasio 80% untuk pelatihan (*training*) dan 20% untuk pengujian (*testing*) [1].

2.2. *Data Preprocessing*

Tahapan *data preprocessing* data berperan penting dalam memastikan kualitas data sebelum digunakan dalam pelatihan model machine learning [4], [5]. Proses ini mencakup penanganan *missing values, outlier, encoding* variabel kategorik, dan normalisasi data numerik. Fitur dengan *missing value* lebih dari 10% dihapus, sedangkan sisanya diimputasi menggunakan mean, median, atau mode, sesuai dengan karakteristik datanya. *Outlier* dideteksi menggunakan metode *Interquartile Range* (IQR) dan kemudian dibatasi (*capping*) agar tidak mengganggu proses pelatihan model [23]. Variabel kategorik seperti wilayah di *encoding* menggunakan *one-hot* atau label *encoding* agar dapat dibaca oleh algoritma *machine learning* [5], [9]. Selain itu, beberapa atribut numerik yang memiliki rentang nilai lebar dinormalisasi dengan *z-score* atau *min-max normalization* untuk meningkatkan stabilitas pelatihan model, meskipun tidak selalu diperlukan [4].

2.3. *Feature Selection using Boruta*

Boruta merupakan metode seleksi fitur berbasis *wrapper* yang menggunakan algoritma *Random Forest* untuk mengevaluasi pentingnya setiap fitur dalam *dataset*. Keunggulan utama *Boruta* terletak pada pendekatannya yang membandingkan setiap fitur asli dengan "*shadow features*", yaitu salinan acak dari fitur tersebut yang diperoleh melalui proses pengacakan baris data (*random permutation*) [7], [24]. *Dataset* yang telah diperluas dengan *shadow features* kemudian dilatih menggunakan *Random Forest* untuk menghasilkan nilai pentingnya (*feature importance*), yang dihitung berdasarkan metrik seperti *mean decrease accuracy* atau *mean decrease Gini* [19]. Selanjutnya, nilai kepentingan fitur asli dibandingkan dengan skor maksimum dari seluruh *shadow features*. Apabila suatu fitur menunjukkan nilai yang secara konsisten lebih tinggi dibanding shadow-nya, maka fitur tersebut dikategorikan sebagai fitur relevan. Sebaliknya, jika nilainya lebih rendah, fitur tersebut akan disingkirkan. Proses ini dilakukan secara iteratif hingga seluruh fitur dapat diklasifikasikan sebagai "*Confirmed"* atau "*Rejected"* [7], [25]. Dengan mempertimbangkan interaksi antar fitur secara menyeluruh, *Boruta* menawarkan pendekatan yang lebih komprehensif dibandingkan metode filter tradisional yang hanya mengandalkan hubungan linier atau korelasi parsial antar variabel [7].

2.4. *Model Gradient Boosting*

Setelah fitur-fitur yang tidak relevan dieliminasi melalui proses seleksi menggunakan *Boruta*, *dataset* hasil *balancing* kemudian digunakan dalam pelatihan model prediktif menggunakan algoritma *Gradient Boosting* sebagai *classifier* [4], [8]. *Gradient Boosting* bekerja dengan membangun pohon keputusan secara bertahap, di mana setiap pohon baru diarahkan untuk memperbaiki kesalahan prediksi (*residual* error) dari pohon sebelumnya. Pendekatan iteratif ini menjadikan *Gradient Boosting* sangat efektif dalam *menangani* data tabular yang kompleks, termasuk dalam kasus prediksi pelanggan *lapser* atau *churn* [1]. Dalam implementasinya, beberapa parameter utama yang dikonfigurasi meliputi fungsi *loss*, *learning rate*, jumlah estimators, kedalaman maksimum pohon, dan early stopping. *Fungsi* loss yang digunakan adalah *logistic loss* atau *binary cross-entropy*, yang merupakan pilihan umum dalam klasifikasi *biner* [26]. Nilai *learning rate* ditetapkan dalam rentang kecil (0.01 hingga 0.1) untuk memastikan proses pembelajaran berlangsung stabil meskipun memerlukan lebih banyak iterasi untuk konvergen. Jumlah pohon atau n_*estimators* dipilih berdasarkan hasil eksperimen awal, dengan kisaran umum antara 100 hingga 300 pohon [8]. Parameter *max_depth* digunakan untuk mengatur kedalaman maksimal pohon, di mana pohon yang lebih dangkal (kedalaman 3–5) cenderung menghindari risiko *overfitting* namun memiliki keterbatasan dalam fleksibilitas model. Selain itu, strategi *early stopping* juga diterapkan apabila skor validasi tidak menunjukkan peningkatan setelah sejumlah iterasi tertentu, sebagai upaya mencegah *overfitting* dan meningkatkan generalisasi model terhadap data uji [4].

2.5. Evaluasi Model

Model dievaluasi berdasarkan beberapa metrik utama: (1) *Accuracy*, proporsi prediksi benar dibandingkan total prediksi [27]. Rumus akurasi ditunjukkan pada persamaan 1. (2) *Recall*, mencerminkan kemampuan model mendeteksi *lapser* secara benar. Recall difokuskan karena biaya kehilangan pelanggan *lapser* (FN) lebih besar ketimbang salah memprediksi pelanggan aktif sebagai *lapser* (FP) [1], [3]. Rumus recall ditunjukkan pada persamaan 2. (3) Area Under Curve (AUC), tabel 1 AUC Receiver Operating Characteristics (ROC) Curve, mengukur keseimbangan true positive rate dan false positive rate [6].

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

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

Tabel 1. Nilai AUC [19]

Nilai AUC	Interpretasi
0.9-1.00	*Excellent Classification*
0.8-0.9	*Good Classification*
0.7-0.8	*Fair Classification*
0.6-0.7	*Poor Classification*
0.5-0.6	*Failure*

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan rangkaian hasil eksperimen yang dilakukan setelah menerapkan tahapan metodologi sebagaimana dijelaskan pada Bab 2. Fokus utama adalah menilai dampak seleksi fitur menggunakan *Boruta*, proses penyeimbangan data, serta konfigurasi *Gradient Boosting* terhadap kinerja prediksi *lapser*. Pembahasan lebih lanjut juga mengulas interpretasi temuan dan keterbatasan yang dihadapi selama penelitian.

3.1. Karakteristik Data Awal

Analisis awal terhadap dataset menunjukkan bahwa terdapat ketidakseimbangan kelas (*class imbalance*) antara pelanggan *lapser* dan *non-lapser*. Dari total jumlah data yang besar, sekitar 26% di antaranya berlabel *lapser*, dengan proporsi kelas lapser secara umum berada dalam kisaran 15–25%. Ketimpangan ini berpotensi menyebabkan bias pada model prediktif karena algoritma cenderung mengutamakan kelas mayoritas (*non-lapser*), sehingga diperlukan teknik balancing untuk memastikan bahwa model dapat mengenali pola dari kelas minoritas secara efektif [28]. Selain analisis distribusi kelas, penelitian ini juga mengidentifikasi sejumlah fitur utama yang memiliki pengaruh signifikan terhadap perilaku pelanggan dalam konteks berhenti berlangganan.

Salah satu fitur penting adalah wilayah, yang tidak hanya mencerminkan lokasi geografis pelanggan, tetapi juga mengandung informasi implisit mengenai infrastruktur jaringan, kondisi sosial-ekonomi, serta karakteristik pasar di wilayah tersebut. Faktor-faktor ini terbukti memengaruhi intensitas dan pola konsumsi layanan data. Fitur penting lainnya adalah pendapatan (*income*), di mana pelanggan dengan daya beli lebih tinggi cenderung membeli layanan data dalam jumlah besar, meskipun perilaku mereka dapat berubah secara signifikan jika terdapat insentif atau promosi menarik dari kompetitor. Selain itu, atribut seperti konsumsi kuota dan frekuensi pengisian ulang menjadi indikator utama dalam menganalisis kecenderungan pelanggan untuk lapser. Penurunan drastis dalam konsumsi kuota atau frekuensi pengisian ulang selama periode tertentu sering kali menjadi sinyal awal bahwa pelanggan tersebut berpotensi berhenti berlangganan.

3.2. *Data Preprocessing*

Proses *data preprocessing* data dilakukan untuk memastikan bahwa dataset yang digunakan memiliki kualitas yang memadai dan bebas dari masalah umum seperti nilai hilang (*missing values*), *outlier* ekstrem, serta format variabel yang tidak sesuai. Tahapan pertama dalam preprocessing adalah penanganan *missing values*. Fitur dengan nilai hilang lebih dari 10% dipertimbangkan untuk dihapus karena dapat menimbulkan distorsi pada hasil analisis. Sebaliknya, jika persentase missing values berada di bawah ambang tersebut, dilakukan imputasi menggunakan metode statistik sederhana seperti mean, median, atau mode, tergantung pada jenis dan distribusi data. Sebagai contoh, pada fitur seperti usia kartu dan frekuensi pengisian ulang yang memiliki proporsi *missing* sekitar 2–3%, dilakukan imputasi dengan nilai rata-rata (*mean*), karena metode ini tidak mengubah distribusi data secara signifikan.

Selanjutnya, untuk mendeteksi dan menangani nilai ekstrem (*outlier*), digunakan pendekatan *Interquartile Range* (IQR) yang memanfaatkan kuartil pertama (Q1) dan ketiga (Q3) untuk menetapkan ambang batas. Nilai-nilai yang berada di luar rentang (Q1 – 1.5IQR) hingga (Q3 + 1.5IQR) diidentifikasi sebagai *outlier*. Alih-alih menghapus data tersebut, penelitian ini menerapkan teknik capping, yaitu membatasi nilai-nilai ekstrem pada ambang maksimum yang masih dianggap wajar, misalnya membatasi konsumsi kuota pada nilai persentil ke-99. Pendekatan ini mempertahankan informasi penting dalam data tanpa mengorbankan kestabilan model.

Untuk variabel kategorik, dilakukan proses encoding agar dapat diproses oleh algoritma *machine learning*. *One-hot encoding* digunakan pada fitur dengan jumlah kategori terbatas, seperti jenis kelamin, sedangkan label *encoding* diterapkan pada fitur dengan banyak kategori, seperti wilayah. Misalnya, kategori wilayah A, B, dan C direpresentasikan dengan angka 1, 2, dan 3 untuk menghindari ledakan dimensi. Keseluruhan proses *preprocessing* ini bertujuan untuk memastikan bahwa data yang digunakan pada tahap seleksi fitur dan pelatihan model berada dalam kondisi bersih, terstruktur, dan stabil, sehingga dapat mengurangi potensi bias serta meningkatkan akurasi dan ketahanan model dalam prediksi.

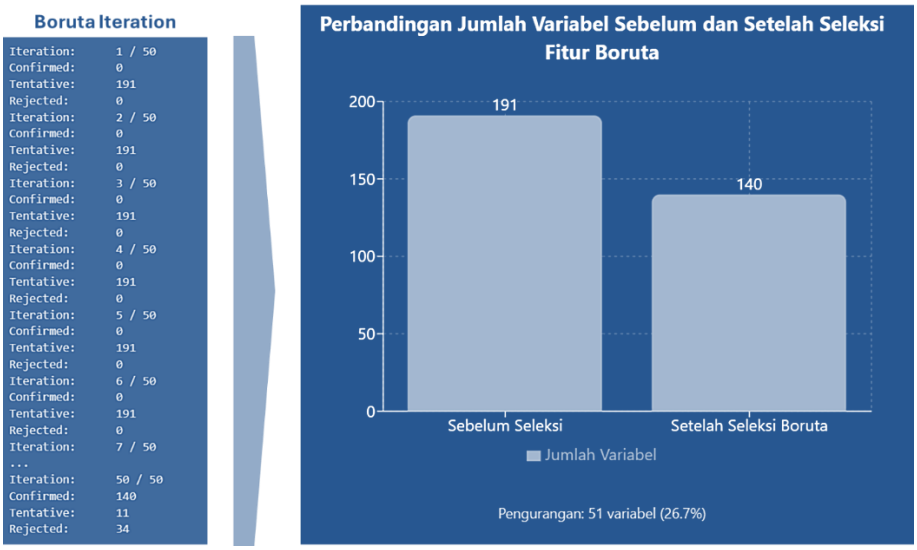
3.3. Hasil *Feature Selection using Boruta*

Boruta Feature Selection digunakan dalam proses ini karena memiliki metode yang efektif untuk mengidentifikasi fitur-fitur yang paling relevan dalam dataset. Dengan menggunakan *Boruta*, dapat mengurangi dimensi data, meningkatkan performa model, dan membuat model lebih mudah diinterpretasikan seperti di gambar 2.

```
from boruta import BorutaPy
from pyspark.ml.classification import RandomForestClassifier, GBTClassifier, LinearSVC, LogisticRegression
from sklearn.ensemble import RandomForestClassifier
import numpy as np

df_pandas_sample_10[features] = df_pandas_sample_10[features].fillna(df_pandas_sample_10[features].mean()).clip(-1e9,1e9)

X = df_pandas_sample_10[features].values
Y = df_pandas_sample_10['y'].values.ravel()
print(X.shape)
print(Y.shape)

rf = RandomForestClassifier(n_jobs=-1, class_weight='balanced', max_depth=5)

boruta_feature_selector = BorutaPy(rf, n_estimators='auto', verbose=2, random_state=4242, max_iter = 50, perc = 90)

boruta_feature_selector.fit(X, Y)

boruta_feature_selector.support_

boruta_feature_selector.ranking_

X_filtered = boruta_feature_selector.transform(X)
X_filtered.shape

final_features = list()
indexes = np.where(boruta_feature_selector.support_ == True)
for x in np.nditer(indexes):
    final_features.append(features[x])
print(final_features)
```

Gambar 2. *Feature Selection Boruta Code*

Setelah tahap preprocessing data, seperti pembersihan *missing values*, penanganan *outlier*, dan *encoding*, dilakukan seleksi fitur menggunakan *Boruta*. Pada gambar 3 dari sekitar 191 fitur yang dianalisis, *Boruta* mengidentifikasi sekitar 140 fitur yang relevan. Fitur-fitur ini memiliki skor kepentingan yang jauh lebih tinggi dibandingkan *shadow features* dalam beberapa iterasi. Beberapa fitur seperti Usia Kartu, Pendapatan, Konsumsi Kuota, dan Frekuensi Pengisian Ulang secara konsisten memperoleh skor tinggi, yang sejalan dengan literatur mengenai faktor-faktor penentu *lapser*. Proses seleksi ini dilakukan dalam 50 iterasi, yang menurut pengalaman peneliti, jumlah iterasi tersebut sudah cukup untuk mendapatkan hasil yang optimal mengingat waktu komputasi yang dibutuhkan cukup lama. Pada setiap iterasi, *Boruta* mengevaluasi kepentingan variabel untuk membantu membangun model yang lebih efisien, akurat, dan mudah diinterpretasikan. Pengurangan sebesar 26,7% dalam jumlah fitur menunjukkan bahwa *Boruta* efektif dalam menyederhanakan dataset tanpa menghilangkan informasi penting.



Gambar 3. *Boruta Iteration* dan Perbandingan Jumlah Variabel

Dalam penelitian ini, baik pada gambar 2 dan 3 metode *Boruta* diterapkan sebagai teknik seleksi fitur dengan pendekatan *wrapper* berbasis *Random Forest.* Implementasi *Boruta* dilakukan menggunakan *library BorutaPy* di *Python*, yang secara default mengadopsi *RandomForestClassifier* dari *scikit-learn* sebagai *estimator*. Jumlah maksimum iterasi (*max_iter*) yang digunakan ditetapkan sebesar 50 untuk memastikan proses seleksi mencapai konvergensi, yaitu ketika tidak ada lagi fitur yang statusnya berubah dalam iterasi berturut-turut. Parameter lainnya disesuaikan mengikuti best *practice* dan tuning awal, seperti n_*estimators* pada *Random Forest*, *max_depth*) agar *Random Forest* dapat mempelajari relasi kompleks antar fitur, dan nilai *random_state* diatur agar proses bersifat replikatif. Kriteria pemilihan fitur mengikuti pendekatan *Boruta* klasik, yaitu sebuah fitur akan dikonfirmasi sebagai "*Confirmed*" jika *consistently* memiliki *importance score* yang lebih tinggi dari *shadow features* secara statistik (menggunakan uji dua arah berdasarkan *Z-score*). Fitur dengan skor lebih rendah secara signifikan akan ditandai sebagai "*Rejected*", sementara fitur yang tidak signifikan disebut "*Tentative*" hingga iterasi berakhir. Hasil akhir seleksi menghasilkan daftar fitur terpilih yang kemudian digunakan dalam pelatihan model *Gradient Boosting*.

3.4. Model *Gradient Boosting*

Gambar 4 menampilkan kode Python yang digunakan untuk melatih dan mengevaluasi model klasifikasi menggunakan *Gradient Boosting Trees* (GBT) dengan *PySpark*. Kode tersebut mengatur inisialisasi GBT dengan beberapa parameter, seperti kolom label, kolom fitur, dan jumlah iterasi maksimum. Selanjutnya, model dilatih menggunakan *dataset training*, dan prediksi dilakukan pada dataset testing. Evaluasi model menggunakan metrik *Area Under the Precision-Recall* (AUC-PR) untuk mengukur performa model. Selain itu, kode ini juga melakukan evaluasi pada dataset real test guna memeriksa kinerja model dalam kondisi nyata. Hasil prediksi kemudian dikelompokkan dan diubah menjadi *DataFrame Pandas*.

```
# Initialize classifiers with class weighting
from pyspark.ml.classification import GBTClassifier
from pyspark.ml.evaluation import BinaryClassificationEvaluator, MulticlassClassificationEvaluator

gbt = GBTClassifier(labelCol=label_col, featuresCol="scaled_features", maxIter=10)
classifiers = [gbt]
classifier_names = ['Gradient Boosting']

# Train and evaluate each classifier
for classifier, name in zip(classifiers, classifier_names):
    print("\nTraining", name, f":{datetime.now()}")
    model = classifier.fit(train_set)
    predictions = model.transform(test_set)
    pd_pred = predictions.groupby(['y','prediction']).count().toPandas()
    evaluation(pd_pred)

    # Evaluate using AUC-PR
    evaluator = BinaryClassificationEvaluator(labelCol=label_col,metricName="areaUnderPR")
    auc_pr = evaluator.evaluate(predictions)
    print(name, "AUC-PR:", auc_pr)

    ## Real Test
    print("\n",name, "\nReal Next Month Test")
    predictions = model.transform(real_test)
    pd_pred = predictions.groupby(['y','prediction']).count().toPandas()
    evaluation(pd_pred)

    # Evaluate using AUC-PR
    evaluator = BinaryClassificationEvaluator(labelCol=label_col,metricName="areaUnderPR")
    auc_pr = evaluator.evaluate(predictions)
    print(name, "AUC-PR:", auc_pr)
```

Gambar 4. *Machine Learning Gradient Boosting Code*

3.5. Analisis dan Evaluasi Model

Pada tabel 2 Penggunaan *Boruta* untuk seleksi fitur telah meningkatkan performa model *Gradient Boosting* baik pada data pelatihan maupun data uji nyata. Peningkatan yang signifikan dalam akurasi, *recall*, dan AUC-PR menunjukkan bahwa fitur-fitur yang dipilih oleh *Boruta* lebih relevan dan informatif, sehingga membantu model untuk lebih baik dalam memprediksi kelas yang benar dan mengurangi overfitting. Ini menunjukkan bahwa *Boruta* adalah alat yang efektif untuk meningkatkan performa model dalam kasus ini.

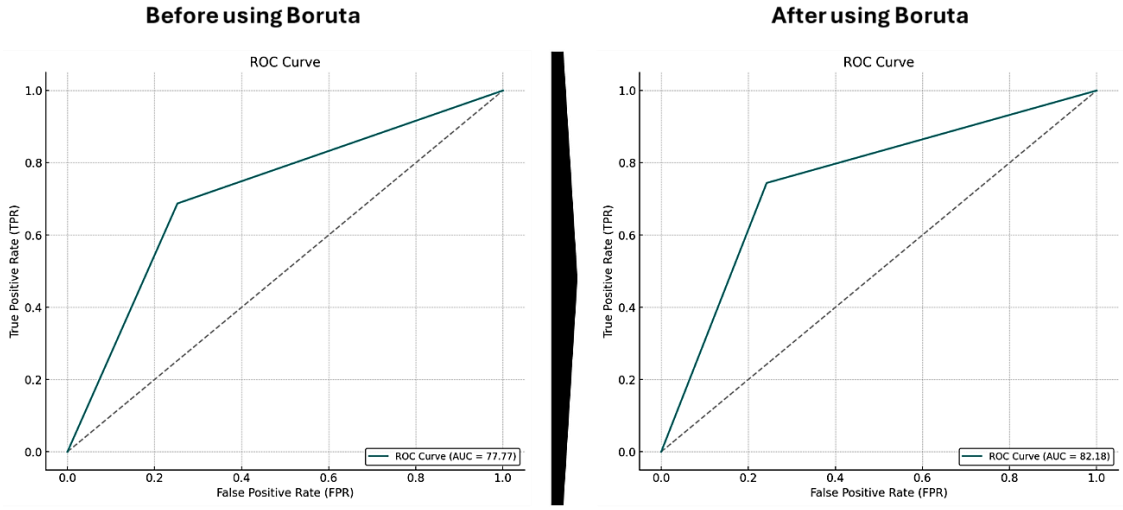
Peningkatan akurasi menunjukkan bahwa seleksi fitur dengan *Boruta* membantu model untuk lebih baik dalam memprediksi kelas yang benar. Ini menunjukkan bahwa fitur-fitur yang dipilih oleh *Boruta*. Peningkatan recall yang signifikan menunjukkan bahwa model lebih baik dalam mengidentifikasi instance positif. Ini penting dalam kasus di mana *false negative* memiliki konsekuensi yang tinggi. Sedangkan AUC-PR yang lebih tinggi menunjukkan bahwa model memiliki keseimbangan yang lebih baik antara precision dan *recall* setelah seleksi fitur. Ini menunjukkan bahwa model lebih baik dalam memprediksi kelas positif tanpa terlalu banyak false positive, serta nilai akurasi setelah menggunakan *Boruta* lebih tinggi dibandingkan tidak menggunakan *Boruta* dengan delta 3.36%.

Analisis komparatif menunjukkan bahwa penerapan *Boruta* mampu meningkatkan performa model secara signifikan. Akurasi meningkat dari 71.74% menjadi 75.10%, recall dari 68.74% menjadi 74.42%, dan AUC dari 77.77% menjadi 82.18%. Hal ini menunjukkan bahwa *Boruta* efektif dalam menyaring fitur relevan dan mengurangi *noise* pada *dataset* berdimensi tinggi. Meski demikian, metode ini memiliki keterbatasan, seperti potensi bias saat menangani fitur yang saling berkorelasi dan waktu komputasi yang tinggi pada dataset besar. Selain itu, penggunaan satu jenis algoritma (*Gradient Boosting*) membatasi variasi hasil. Penelitian lanjutan disarankan untuk membandingkan *Boruta* dengan metode seleksi fitur lain seperti SHAP atau RFE, serta menguji model alternatif seperti *Neural Network* atau *CatBoost* untuk validasi hasil yang lebih luas.

Tabel 2. Perbandingan Hasil Evaluasi Sebelum dan Sesudah Menggunakan *Boruta*

	Gradient Boosting				Delta
	Before using Boruta		After Using Boruta		
Training & Evaluate	Accuracy	71.74%	Accuracy	75.10%	3.36%
	Recall	68.74%	Recall	74.42%	5.68%
	AUC-PR	77.77%	AUC-PR	82.18%	4.41%
Real Test	Accuracy	72.81%	Accuracy	75.74%	2.93%
	Recall	65.04%	Recall	66.84%	1.80%
	AUC-PR	55.88%	AUC-PR	62.32%	6.44%

Pada gambar 5 ROC adalah kurva yang menunjukkan keseimbangan antara True Positive Rate dan *False Positive Rate* untuk berbagai ambang batas klasifikasi. AUC, yang merupakan luas area di bawah kurva ROC, mengindikasikan seberapa efektif model dalam membedakan antara kelas positif dan negatif. Setelah menerapkan *Boruta*, nilai AUC mencapai 82,18%, menunjukkan bahwa model memiliki kemampuan klasifikasi yang sangat baik dan dapat membedakan kelas positif dan negatif dengan efektif, mendekati kinerja ideal. Dibandingkan dengan sebelum menggunakan *Boruta*, nilai AUC mengalami peningkatan yang signifikan. Berdasarkan kriteria nilai AUC pada tabel 1, nilai AUC setelah menggunakan *Boruta* termasuk dalam kategori *"Good Classification"*, sehingga model klasifikasi ini dapat diterima dan diimplementasikan dengan baik.



Gambar 5. Grafik ROC Perbandingan Sebelum dan Sesudah Menggunakan *Boruta*

Pada tabel 3 Secara keseluruhan, model setelah *Boruta* lebih layak untuk diterapkan karena performa klasifikasinya lebih optimal dan stabil, dengan keseimbangan yang lebih baik antara mendeteksi positif dan meminimalkan kesalahan prediksi negatif. Hal ini menunjukkan bahwa *Boruta* efektif dalam meningkatkan kualitas model melalui seleksi fitur yang relevan, sehingga model menjadi lebih efisien dan akurat.

Tabel 3. Analisa Perbandingan ROC

Aspek	Before using Boruta (AUC = 77.77)	After using Boruta (AUC = 82.18)
Kemampuan Klasifikasi (AUC)	Cukup baik	Baik, lebih tinggi
Kemampuan Deteksi (TPR)	Meningkat, tapi lambat	Lebih cepat meningkat
False Positive Rate (FPR)	Lebih tinggi di awal	Lebih rendah
Dekat ke Sudut Kiri Atas (Ideal)	Tidak terlalu dekat	Lebih dekat
Potensi Penggunaan	Bisa digunakan, perlu peningkatan	Lebih layak diterapkan

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penggunaan metode *feature selection Boruta* secara signifikan meningkatkan kinerja model prediksi pelanggan *lapser*. Dengan menyaring 191 fitur menjadi 140 fitur relevan, *Boruta* berhasil menyederhanakan data tanpa kehilangan informasi penting, sekaligus meningkatkan akurasi model *Gradient Boosting* dari 71.74% menjadi 75.10%, *recall* dari 68.74% menjadi 74.42%, dan AUC dari 77.77% menjadi 82.18%. Hasil ini menunjukkan bahwa *Boruta* efektif dalam mengurangi *noise* dan meningkatkan fokus model pada fitur-fitur utama seperti usia kartu, pendapatan, konsumsi kuota, dan frekuensi pengisian ulang. Model yang dihasilkan berpotensi digunakan oleh industri layanan data untuk mendeteksi pelanggan berisiko tinggi lebih awal dan merancang strategi retensi yang lebih tepat sasaran. Selain mendukung pengambilan keputusan berbasis data, hasil seleksi fitur juga memberi wawasan strategis mengenai perilaku pelanggan.

Namun, penelitian ini memiliki keterbatasan, seperti ketergantungan pada satu metode seleksi fitur (*Boruta*) dan satu algoritma klasifikasi (*Gradient Boosting*), serta belum mempertimbangkan dinamika waktu (*time-series behavior*) dalam data pelanggan. Untuk pengembangan ke depan, disarankan untuk membandingkan *Boruta* dengan metode seleksi lain seperti SHAP atau *Recursive Feature Elimination*, serta mengeksplorasi pendekatan ensembel dan model time-aware untuk meningkatkan akurasi prediksi dalam skenario bisnis yang lebih dinamis.

REFERENSI

[1] B. Larivière and D. Van den Poel, "Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: The case of financial services," *Expert Syst Appl*, vol. 27, no. 2, pp. 277–285, 2004, doi: https://doi.org/10.1016/j.eswa.2004.02.002.

[2] Y. Duan, R. Zbigniew W, A. Lu, A. Tzacheva, and M. Khouja, "Recommender System for Improving Churn Rate," 2022.

[3] S. Neslin, S. Gupta, W. Kamakura, J. Lu, and C. Mason, "Defection Detection: Measuring and Understanding the Predictive Accuracy of Customer Churn Models," *Journal of Marketing Research American Marketing Association ISSN*, vol. 43, pp. 204–211, Apr. 2006, doi: 10.1509/jmkr.43.2.204.

[4] R. Agrawal, "A Modified K-Nearest Neighbor Algorithm Using Feature Optimization," *International Journal of Engineering and Technology*, vol. 8, pp. 28–37, Feb. 2016.

[5] C. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," *Inf Sci (N Y)*, vol. 275, pp. 314–347, Aug. 2014, doi: 10.1016/j.ins.2014.01.015.

[6] H. He and E. A. Garcia, "Learning from Imbalanced Data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, pp. 1263–1284, Oct. 2009, doi: 10.1109/TKDE.2008.239.

[7] M. B. Kursa and W. R. Rudnicki, "Feature selection with the boruta package," *J Stat Softw*, vol. 36, no. 11, pp. 1–13, 2010, doi: 10.18637/jss.v036.i11.

[8] J. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, vol. 29, Nov. 2000, doi: 10.1214/aos/1013203451.

[9] A. Liaw and M. Wiener, "Classification and Regression by RandomForest," *Forest*, vol. 23, Nov. 2001.

[10] M. P. Parmar and M. Shilpa Serasiya, "Telecom Churn Prediction Model using XgBoost Classifier and Logistic Regression Algorithm," *International Research Journal of Engineering and Technology*, 2021, [Online]. Available: www.irjet.net

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*. 2009.

[12] H. Ouchra, A. Belangour, and A. Erraissi, "Machine Learning Algorithms for Satellite Image Classification Using Google Earth Engine and Landsat Satellite Data: Morocco Case Study," *IEEE Access*, p. 1, Jul. 2023, doi: 10.1109/ACCESS.2023.3293828.

[13] S. Wu, "Customer Churn Prediction in Telecom Based on Machine Learning," *Highlights in Science, Engineering and Technology*, vol. 94, pp. 113–118, Apr. 2024, doi: 10.54097/snc09915.

[14] B. Zhang, "Customer Churn in Subscription Business Model—Predictive Analytics on Customer Churn," *BCP Business & Management*, vol. 44, pp. 870–876, Apr. 2023, doi: 10.54691/bcpbm.v44i.4971.

[15] Y.-J. Han, J. Moon, and J. Woo, "Prediction of Churning Game Users Based on Social Activity and Churn Graph Neural Networks," *IEEE Access*, vol. PP, p. 1, Jan. 2024, doi: 10.1109/ACCESS.2024.3429559.

[16] F. Sohil, M. Sohail, and J. Shabbir, "An introduction to statistical learning with applications in R: by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, New York, Springer Science and Business Media, 2013, $41.98, eISBN: 978-1-4614-7137-7," *Stat Theory Relat Fields*, vol. 6, p. 1, Sep. 2021, doi: 10.1080/24754269.2021.1980261.

[17] N. Farhana, A. Firdaus, M. F. Darmawan, and M. F. Ab Razak, "Evaluation of Boruta algorithm in DDoS detection," *Egyptian Informatics Journal*, vol. 24, no. 1, pp. 27–42, Mar. 2023, doi: 10.1016/j.eij.2022.10.005.

[18] A. Alsahaf, N. Petkov, V. Shenoy, and G. Azzopardi, "A framework for feature selection through boosting," *Expert Syst Appl*, vol. 187, p. 115895, Jan. 2022, doi: 10.1016/J.ESWA.2021.115895.
[19] F. Gorunescu, *Data Mining: Concepts, models and techniques*. 2011.
[20] A. Jovic, K. Brkić, and N. Bogunovic, *A review of feature selection methods with applications*. 2015. doi: 10.1109/MIPRO.2015.7160458.
[21] A. Bhatnagar, "Customer Churn Prediction using Machine Learning Approach: A Comprehensive Study," *Journal of Information Systems Engineering and Management*, vol. 10, pp. 80–92, Mar. 2025, doi: 10.52783/jisem.v10i25s.3944.
[22] M. Ganiyu, O. E. Johnson, and O. V Johnson, "Credit Scoring Prediction Using Boruta Feature Selection with Different Sampling Techniques," in *2024 International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG)*, 2024, pp. 1–9. doi: 10.1109/SEB4SDG60871.2024.10630264.
[23] A. M. Sharifnia, D. E. Kpormegbey, D. K. Thapa, and M. Cleary, "A Primer of Data Cleaning in Quantitative Research: Handling Missing Values and Outliers," *J Adv Nurs*, 2025, doi: 10.1111/jan.16908.
[24] F. Degenhardt, S. Seifert, and S. Szymczak, "Evaluation of variable selection methods for random forests and omics data sets," *Brief Bioinform*, vol. 20, no. 2, pp. 492–503, Mar. 2019, doi: 10.1093/bib/bbx124.
[25] H. Gholami, A. Mohammadifar, S. Golzari, D. G. Kaskaoutis, and A. L. Collins, "Using the Boruta algorithm and deep learning models for mapping land susceptibility to atmospheric dust emissions in Iran," *Aeolian Res*, vol. 50, Mar. 2021, doi: 10.1016/j.aeolia.2021.100682.
[26] Z. Zhang, L. Shi, and D.-X. Zhou, "Classification with Deep Neural Networks and Logistic Loss," *Journal of Machine Learning Research*, vol. 25, no. 125, pp. 1–117, 2024, [Online]. Available: http://jmlr.org/papers/v25/22-0049.html
[27] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Sci Rep*, vol. 14, no. 1, p. 6086, 2024, doi: 10.1038/s41598-024-56706-x.
[28] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res. (JAIR)*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.