Seleksi Fitur dan Penanganan *Imbalanced Data* menggunakan RFECV dan ADASYN

Irfan Pratama[1], Albert Yakobus Chandra[2], Putri Taqwa Prasetyaningrum[3]
Program Studi Sistem Informasi
Universitas Mercu Buana Yogyakarta
Yogyakarta, Indonesia
e-mail: [1]irfanp@mercubauana-yogya.ac.id, [2]albert.ch@mercubauana-yogya.ac.id, [3]putri@mercubauana-yogya.ac.id

Diajukan: 21 Juni 2021; Direvisi: 28 September 2021; Diterima: 30 September 2021

Abstrak

Proses data mining bekerja terhadap data yang tersedia. Jika dataset tidak tersedia sepenuhnya, hasil pengolahan data mining menjadi tidak optimal. Terdapat beberapa kondisi data yang perlu penanganan terlebih dahulu sebelum memasuki tahap data mining. Salah satunya ialah imbalanced class yang merupakan kondisi di mana distribusi data pada setiap kelas tidak proporsional. Sebagai salah satu cara untuk efisiensi proses klasifikasi, seleksi fitur dapat memenuhi kebutuhan tersebut karena hasil dari seleksi fitur adalah sebuah dataset dengan jumlah atribut yang lebih sedikit dari sebelumnya. Untuk menyelesaikan permasalahan imbalanced class, ADASYN digunakan dalam penelitian ini sebagai metode untuk menyeimbangkan proporsi kelas pada dataset. Sedangkan RFECV digunakan sebagai metode fitur seleksi yang dapat meningkatkan efisiensi pada proses klasifikasi. Setelah dilakukan evaluasi dari hasil klasifikasi pada dataset yang menggunakan seleksi fitur, didapatkan hasil klasifikasi yang lebih baik dibandingkan dengan hasil klasifikasi pada dataset tanpa seleksi fitur. Hal tersebut dibuktikan dengan perbandingan antara hasil terbaik dari akurasi klasifikasi dataset tanpa seleksi fitur. Hasil dari metode CART sebesar 85.1% yang merupakan hasil dari pengolahan data tanpa menggunakan metode fitur seleksi. sedangkan metode Bagging k-NN yang menghasilkan akurasi sebesar 88% yang di aplikasikan pada dataset dengan seleksi fitur. Sehingga dapat disimpulkan bahwa seleksi fitur dapat meningkatkan akurasi pada klasifikasi.

Kata kunci: *Data mining, Seleksi fitur, Imbalanced class, Bagging k-NN, RFECV, ADASYN.*

Abstract

The data mining process works on the available data. If the dataset is not fully available, the results of data mining processing will not be optimal. Several data conditions need to be handled first before entering the data mining stage. One of them is an imbalanced class, which is a condition where the distribution of data in each class is not proportional. As a way to improve the efficiency of the classification process, feature selection can meet these needs because the result of feature selection is a dataset with fewer attributes than before. To solve the imbalanced class problem, ADASYN is used in this study as a method to balance the class proportions in the dataset. While RFECV is used as a feature selection method that can increase efficiency in the classification process. After evaluating the classification results on the dataset using feature selection, the classification results are better than the classification results on the dataset without feature selection. This is proved by the comparison between the best results of dataset classification accuracy without feature selection. The result of the CART method is 85.1% which is the result of data processing without using the feature selection method. while the Bagging k-NN method which produces an accuracy of 88% is applied to the dataset with feature selection. So it can be concluded that feature selection can improve classification accuracy.

Keywords*: Data mining, Imbalanced class, Feature selection, Bagging k-NN, RFECV, ADASYN.*

1. Pendahuluan

Dalam perkembangan teknologi saat ini, di mana data menjadi sebuah komoditi yang berharga bagi sektor apa pun yang ada di dunia ini. Dengan menggunakan data, sebuah informasi dan pengetahuan dapat diperoleh. *Data mining* merupakan sebuah disiplin ilmu yang dapat mengubah kumpulan data-data

yang terlihat tidak merepresentasikan apa pun menjadi sebuah pengetahuan yang dapat digunakan oleh pemangku kepentingan dalam mengambil keputusan atau mengetahui kondisi saat ini[1]. *Data mining* menjadi salah satu tahap dari serangkaian proses penggalian pengetahuan yang lebih umum disebut sebagai *Knowledge Discovery in Database (KDD)*, dalam rangkaian pengolahan data tersebut terdapat beberapa tahapan yang dilalui. Tahapan-tahapan tersebut tergantung pada kebutuhan dari proses *data mining* yang akan dilakukan. Secara umum, tahapan dari rangkaian proses *KDD* adalah *data acquisition, preprocessing, data mining, evaluation, interpretation.* Pada rangkaian tersebut dimungkinkan terdapat proses-proses kecil yang dimaksudkan untuk menunjang proses *data mining* yang dilakukan. Sebagai contoh, jika *dataset* yang digunakan adalah data kategorial dan digunakan untuk proses klasifikasi mengalami ketidakseimbangan pada distribusi kelasnya. *Dataset* yang di dalamnya terdapat *missing values*, atau *dataset* yang memiliki begitu banyak atribut dan diinginkan untuk mencari tahu mana saja data yang penting untuk digunakan. Hal-hal tersebut memiliki mekanisme penyelesaiannya masing-masing secara spesifik. Untuk data yang distribusi kelasnya tidak seimbang dapat ditangani oleh metode-metode untuk *imbalanced class*. Untuk *dataset* yang memiliki *missing values* dengan jumlah yang banyak dapat menggunakan pendekatan *imputation*. Sedangkan untuk *dataset* yang memiliki atribut yang banyak dan menginginkan adanya efisiensi proses dan mencari atribut mana yang optimal dan penting dapat menggunakan seleksi fitur.

Pada kenyataannya, tidak semua *dataset* tersedia secara sempurna. Terlebih jika data tersebut adalah data asli yang secara aktual didapatkan dari proses observasi atau pengambilan data primer. Jika terdapat dua atau lebih kelompok yang memisahkan antar data-data tersebut atau yang biasa disebut klasifikasi, proporsi sebaran data antar masing-masing kelas menjadi penting untuk diperhatikan. Proses *data mining* yang bekerja terhadap data yang tersedia akan mengolah data seperti apa adanya. Jika ketersediaan data tidak dipastikan terlebih dahulu, maka hasil dari proses *data mining* menjadi tidak optimal. Kondisi data dengan sebaran tiap kelas yang tidak proporsional disebut dengan *imbalanced class*. Kondisi tersebut dapat menyebabkan bias pada hasil klasifikasi yang dilakukan karena jumlah data yang dapat digunakan untuk pembelajaran model menjadi tidak seimbang antara satu kelas dengan kelas lainnya. Terdapat beberapa mekanisme yang dapat menangani masalah tersebut yang terbagi menjadi dua pendekatan yaitu pendekatan level data, dan pendekatan level algoritme[2]. Metode-metode yang secara umum digunakan untuk menangani kondisi seperti itu adalah metode dari pendekatan level data. seperti yang dilakukan oleh [3] yang menggunakan SMOTE sebagai metode untuk menangani ketidakseimbangan kelas pada *dataset* yang digunakan. Terdapat metode lain yaitu ADASYN yang merupakan pemutakhiran dari metode SMOTE.

Terdapat penelitian-penelitian terdahulu yang memiliki permasalahan *imbalanced class* dan diselesaikan menggunakan mekanisme *oversampling* seperti penelitian yang dilakukan oleh [4] yang mengembangkan sebuah metode prediksi yang digunakan untuk menyelesaikan masalah pada *"lysine succinylation sites"* dengan menggunakan 6 fitur sekuensial bernama Inspector. Metode tersebut terdiri dari *Edited Nearest-Neighbor* (ENN) sebagai metode *undersampling* dan *Adaptive Synthetic Sampling Approach* (ADASYN) sebagai metode *oversampling* pada saat proses penanganan *imbalanced class* pada data *training*. Kemudian diklasifikasikan menggunakan metode *Random Forest* dan menghasilkan akurasi prediksi sebesar 90% dan AUC pada analisis ROC sebesar 0.96. Penelitian lain dilakukan oleh [5] yang menggunakan metode *oversampling* SMOTE dalam *extreme learning machine* untuk menangani *imbalanced class*. Dikatakan bahwa SMOTE meningkatkan signifikansi dari sampel kelas minoritas dengan membuat sampel sintetis dalam konteks penentuan klasifikasi yang kemudian membentuk sebuah model klasifikasi untuk *imbalanced class data* yang disebut SMOTE-CSELM. Penelitian yang lain dilakukan oleh [6] yang menggunakan ADASYN untuk menangani *imbalanced class* pada *churn prediction dataset* menggunakan metode *Backpropagation*. Penelitian tersebut menghasilkan akurasi sebesar 96.31%.

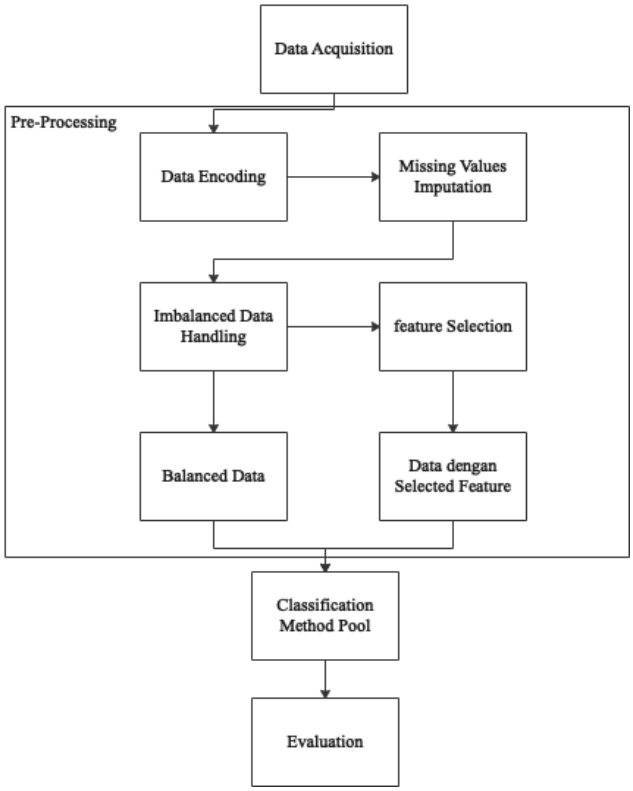
Dalam kaitannya dengan optimalisasi proses pengolahan data, fungsi dari proses seleksi fitur menghasilkan *dataset* yang lebih ramping karena pengurangan jumlah atribut yang diperoleh dari pengukuran atribut-atribut yang memiliki pengaruh signifikan terhadap kelas. Terdapat metode-metode umum yang digunakan untuk mengetahui atribut mana yang memberikan pengaruh terhadap kelas dari *dataset* yang digunakan. Pendekatan berupa *filter-based* seperti *pearson correlation* dan *chi-squared, wrapper-based* seperti *Recursive Feature Elimination (RFE),* dan metode *embedded* seperti *Lasso* dapat digunakan sebagai metode seleksi fitur [7]. Seleksi fitur dapat meningkatkan kinerja dari metode *data mining* yang digunakan karena hanya memproses fitur-fitur terbaik saja dari data yang ada. Penggunaan RFECV sebagai metode seleksi fitur telah digunakan oleh penelitian-penelitian terdahulu. Seperti pada penelitian yang dilakukan oleh [8] dalam penyelesaian masalah deteksi gangguan pada keamanan jaringan dengan menggunakan klasifikasi Bernoulli Naïve Bayes dan *Recursive feature Elimination with Cross-Validation* (RFECV). RFECV yang berfungsi sebagai mekanisme seleksi fitur dapat menentukan fitur-fitur yang penting dari *dataset* yang ada dan kemudian diklasifikasikan. Hasil dari penelitian tersebut adalah

nilai AUC ROC sebesar 0.93. Penelitian lain dilakukan oleh [9] yang menggunakan RFECV untuk menghilangkan fitur yang tidak penting yang justru memberikan bias terhadap hasil klasifikasi pada permasalahan pengelompokan nanomaterial yang diklasifikasikan menggunakan *Random Forest*. Penelitian tersebut menghasilkan akurasi sebesar 82%.

Dari permasalahan yang disampaikan sebelumnya dan dari penelitian-penelitian terkait yang telah dilakukan sebelumnya, penelitian ini berfokus pada proses penanganan *imbalanced class* yang terjadi pada *dataset* dan seleksi fitur untuk mempercepat proses pengolahan data dan meningkatkan performa dari metode klasifikasi. Selain itu, dari penelitian terdahulu yang telah disampaikan sebelumnya belum terdapat penelitian yang menggunakan kombinasi antara ADASYN dan RFECV untuk menangani *dataset* karyawan.

2. Metode Penelitian

Pada bagian ini akan dijelaskan alur dari penelitian ini. Dimulai dari pengambilan data, tahapan *pre-processing* untuk menghasilkan *dataset* yang siap untuk di proses menggunakan metode klasifikasi yang ditentukan. Tahapan *pre-processing* juga meliputi, *missing values imputation*, *feature selection*, dan *imbalanced data handling* yang merupakan poin utama dari penelitian ini. Untuk menguji perbandingan apakah proses *feature selection* dan *imbalanced data handling* dapat memberikan perbaikan pada akurasi yang dihasilkan oleh metode-metode klasifikasi yang ditentukan. Metode klasifikasi yang digunakan untuk menguji *dataset* yang digunakan adalah *Random Forest*, *Decision tree*, CART, *Naïve Bayes*, *Logistic Regression*, K-NN, *Support Vector Machine*, *Bagging*, dan *Stacking* di mana dua metode terakhir merupakan *ensemble method* yang dinilai memiliki performa yang lebih kuat dibandingkan dengan model-model klasifikasi tunggal. Gambar dari alur penelitian ini dapat dilihat pada Gambar 1.

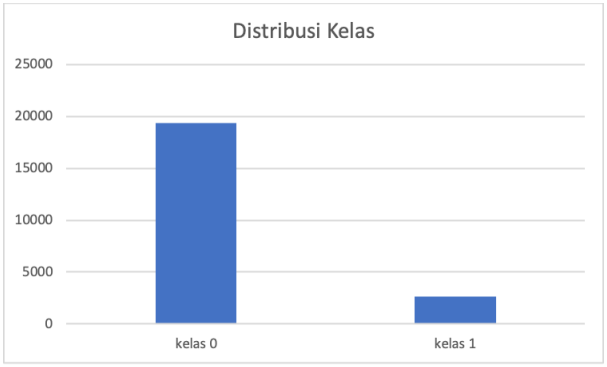


Gambar 1. Diagram penelitian.

2.1. *Data Acquisition*

Data yang digunakan pada penelitian ini adalah *dataset* yang diambil dari web kaggle.com dan juga dijadikan *dataset* untuk kompetisi *data science*. Data disajikan dalam bentuk CSV *file* yang akan digunakan dari setiap proses yang dilakukan pada penelitian ini. Data tersebut berisi 220005 data dari pegawai dengan semua data-data terkait demografis pegawai dan juga rekam jejak dari pekerjaan pegawai tersebut seperti lama bekerja, usia, pencapaian atau *achievement* yang sudah pernah didapatkan, dan lain-lain. Jumlah distribusi untuk setiap kelas pada *dataset* ini adalah 19337 untuk kelas 0 (bukan karyawan

dengan *best performance*) dan 2668 untuk kelas 1 (karyawan *best performance*). Jika dilihat secara langsung, proporsi jumlah data pada kedua kelas sangat tidak seimbang. Hal tersebut dapat menyebabkan bias pada hasil klasifikasi yang dilakukan jika kondisi tersebut tidak ditangani terlebih dahulu. Bentuk visual dari distribusi kelas pada *dataset* ini dapat dilihat pada Gambar 2 dan deskripsi *dataset* yang digunakan dapat dilihat pada Tabel 1 dan sampel *dataset* pada Tabel 2.



Gambar 2. Distribusi kelas.

Dari Gambar 2 dapat dilihat bahwa secara jumlah data pada setiap kelas sangat jauh berbeda. Perbedaan proporsi tersebut akan sangat mempengaruhi hasil proses *data mining* yang akan dilakukan selanjutnya. Karena hasil pada proses klasifikasi akan condong atau bias ke arah data-data yang ada pada kelas mayoritas. Oleh karena itu, data ini tidak dapat langsung di proses ke tahap *data mining* akan tetapi akan menjalani tahap *pre-processing* untuk memastikan semua kondisi yang tidak ideal di dalam *dataset* sudah terselesaikan.

Tabel 1. Deskripsi atribut *dataset*.

No	Attribute	Tipe Data	Deskripsi
1	Job_level	Nominal	Level pekerjaan / jabatan dari karyawan tersebut saat ini ditunjukkan oleh kode JG03-JG06
2	job_duration_in_current_job_level	Float	Lama bekerja karyawan tersebut pada level pekerjaan / jabatan saat ini.
3	person_level	Nominal	Level person dari karyawan tersebut ditunjukkan oleh kode PG01-PG08
4	job_duration_in_current_person_level	Float	Lama bekerja pada level person saat ini.
5	job_duration_in_current_branch	Float	Lama pekerjaan di kantor cabang saat ini
6	Employee_type	Nominal	Jenis ikatan kerja ditunjukkan dengan RM_Type_A-RM-Type_A.
7	Employee_status	Nominal	Status karyawan (kontrak / tetap)
8	Gender	Nominal	Jenis Kelamin
9	Age	Numerik	Umur ditunjukkan dengan Tahun lahir
10	marital_status_maried(Y/N)	Nominal	Status Pernikahan
11	number_of_dependences	Numerik	Jumlah tanggungan
12	number_of_dependences (male)	Numerik	Jumlah tanggungan laki-laki
13	number_of_dependences (female)	Numerik	Jumlah tanggungan Perempuan
14	Education_level	Nominal	Level Pendidikan
15	GPA	Float	Nilai Indeks Akademik
16	Year_graduated	Numerik	Tahun Lulus
17	job_duration_as_permanent_worker	Numerik	Lama bekerja sebagai pegawai
19	job_duration_from_training	Numerik	Lama bekerja sejak pelatihan
20	branch_rotation	Numerik	Jumlah rotasi bekerja di cabang berbeda
21	job_rotation	Numerik	Jumlah rotasi pekerjaan
22	assign_of_otherposition	Numerik	Jumlah penugasan pada posisi berbeda
23	Annual leave	Numerik	Banyaknya cuti tahunan
24	Sick_leaves	Numerik	Banyaknya izin sakit
25	Avg_achievement_%	Float	Rerata persentase pencapaian kinerja
26	Last_achievement_%	Float	Persentase capaian terakhir
27	Achievement_above_100%_during3quartal	Numerik	Jumlah pencapaian kinerja pada *quartal* ketiga yang di atas 100%
28	achievement_target_1	Nominal	*Range* ketercapaian target kesatu (achiev_ > 50%, …)
29	achievement_target_2	Nominal	*Range* ketercapaian target kedua (achiev_ > 50%, …)
30	achievement_target_3	Nominal	Ketercapaian capaian target ketiga (tercapai / tidak tercapai)
31	Best Performance	Binary	Kelas dari *dataset*, menunjukkan ketercapaian seorang karyawan pada *best performance*. (0 dan 1)

Tabel 2. Sampel data.

No	Job_level	Job_duration_in current_job_level	Person_level	Employee_type	Gender	…	Best_performance
1	JG04	1.17	PG03	RM_type_A	Male	…	0
2	JG04	1.83	PG03	RM_type_A	Male	…	1
3	JG03	0.75	PG01	RM_type_B	Male	…	0
4	JG03	0	PG01	RM_type_B	Male	…	0
5	JG04	1.17	PG03	RM_type_A	Male	…	0
6	JG04	0.75	PG03	RM_type_B	Male	…	0
7	JG04	1.83	PG03	RM_type_B	Female	…	1
8	JG03	0.75	PG01	RM_type_B	Male	…	0
9	JG04	1.83	PG03	RM_type_B	Male	…	0
10	JG04	1.17	PG03	RM_type_A	Male	…	0

2.2. *Pre-Processing*

Pada tahap *Pre-Processing* ini terdapat beberapa langkah yang dilakukan untuk mempersiapkan data mentah menjadi data yang siap digunakan untuk proses *data mining* selanjutnya. Langkah-langkah tersebut meliputi: transformasi data/*data encoding*; pengisian *missing values*; *imbalanced data handling*; seleksi fitur.

2.2.1. Transformasi Data/*Data Encoding*

Pada tahap ini, data-data yang memiliki bentuk data nominal akan di transformasi atau di-*encode* menjadi bentuk numerik. Bentuk transformasi data menggunakan mekanisme sederhana yaitu mengubah semua data nominal dengan kode numerik 0-n, di mana n adalah varian terakhir dari data nominal pada atribut tersebut. Contoh hasil transformasi data dapat dilihat pada Tabel 3.

Tabel 3. Hasil transformasi data.

No	Job_level	Job_duration_in current_job_level	Person_level	Employee_type	Gender	…	Best_performance
1	4	1.17	3	0	1	…	0
2	4	1.83	3	0	1	…	1
3	3	0.75	1	1	1	…	0
4	3	0	1	1	1	…	0
5	4	1.17	3	0	1	…	0
6	4	0.75	3	1	1	…	0
7	4	1.83	3	1	0	…	1
8	3	0.75	1	1	1	…	0
9	4	1.83	3	1	1	…	0
10	4	1.17	3	0	1	…	0

2.2.2. Penanganan *Missing Values*

Tahapan ini memiliki peran cukup penting dari keseluruhan proses penelitian ini, pengisian *missing values* dimaksudkan untuk menjaga jumlah data yang bisa digunakan untuk proses *data mining* selanjutnya. *Dataset* yang digunakan memiliki jumlah *missing values* yang tidak sedikit. Rangkuman jumlah *missing values* pada setiap atribut yang terdapat pada data ini dapat dilihat pada Gambar 3.

```
job_level                                  0
job_duration_in_current_job_level          0
person_level                               0
job_duration_in_current_person_level       0
job_duration_in_current_branch             0
Employee_type                             12
Employee_status                            0
gender                                     0
age                                        0
marital_status_maried(1/0)                 0
number_of_dependences                      0
number_of_dependences (male)               0
number_of_dependences (female)             0
Education_level                         3608
GPA                                     7452
year_graduated                          3946
job_duration_as_permanent_worker        2055
job_duration_from_training                 0
branch_rotation                            0
job_rotation                               0
assign_of_otherposition                    0
annual leave                               0
sick_leaves                                0
Avg_achievement_%                       6289
Last_achievement_%                      6302
Achievement_above_100%_during3quartal   6302
achievement_target_1                    6727
achievement_target_2                    6727
achievement_target_3                    6727
Best Performance                           0
```

Gambar 3. Jumlah *missing values* per atribut.

Missing values pada *dataset* ini akan diselesaikan menggunakan pendekatan *imputation*, yaitu proses mengisi *missing values* dengan nilai-nilai hasil estimasi menggunakan metode tertentu. Selain *imputation* pendekatan penanganan *missing values* dapat berupa penghapusan baris data yang memiliki *missing values*, akan tetapi pendekatan tersebut akan mengakibatkan jumlah *dataset* semakin berkurang drastis jika jumlah *missing values* yang terdapat pada *dataset* berjumlah banyak.

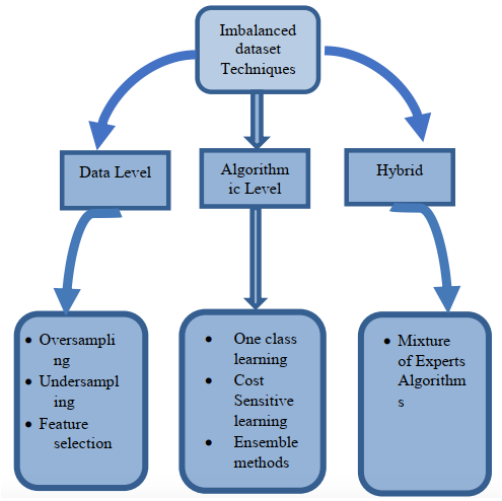
Metode yang digunakan untuk melakukan *missing values imputation* pada *dataset* ini adalah Missforest [10][11]. Missforest merupakan teknik *missing values imputation* berbasis pada metode Random Forest di mana metode ini menggunakan model Random Forest pada data yang tersedia pada *dataset* untuk kemudian melakukan estimasi pada *missing values* pada *dataset* tersebut [11]. MissForest dapat digunakan untuk pengisian *missing values* pada *dataset* dengan tipe data yang *heterogeny* (*mixed data-type*) [10]. *Dataset* hasil *missing values imputation* menggunakan metode Missforest dapat dilihat pada Tabel 4.

Tabel 4. Sampel atribut sebelum & setelah *missing values imputation.*

No	Atribut Last_achievement_% (sebelum *missing values imputation*)	Atribut Last_achievement_% (setelah *missing values imputation*)
1	?	33.0385
2	46.8	46.8
3	?	32.1895
4	?	31.4021
5	?	38.1723
6	?	22.6998
7	?	33.0385

2.2.3. *Imbalanced Data Handling*

Imbalanced data merupakan sebuah kondisi di mana distribusi dari kelas yang terdapat pada *dataset* tidak seimbang jumlahnya. Selisih jumlah data yang terdapat pada masing-masing kelas akan dapat mempengaruhi performa dari metode klasifikasi jika tidak ditangani karena akan menimbulkan bias terhadap kelas mayoritas[12]. Hal yang membuat data *imbalanced* tidak ideal untuk langsung diproses adalah permasalahan proporsi data yang dapat mengurangi kinerja dari algoritma klasifikasi standar. Terdapat beberapa cara menangani kondisi *imbalanced* pada sebuah *dataset* dan dapat dikategorikan menjadi dua jenis pendekatan, yaitu pendekatan level data, pendekatan level algoritme, dan *hybrid* antara keduanya [2]. Ilustrasi dari metode untuk masing-masing jenis pendekatan dalam menangani *imbalanced dataset* dapat dilihat pada Gambar 4.



Gambar 4. Penanganan *imbalanced dataset* [2].

Pendekatan level data untuk menyelesaikan *imbalanced dataset* dapat disebut sebagai metode eksternal karena dilakukan di luar proses *data mining* secara umum (masih ada di tahap pra-pemrosesan). Pendekatan ini mencoba untuk menyeimbangkan proporsi data dengan cara mengurangi jumlah data pada kelas mayoritas, atau meningkatkan jumlah data pada kelas minoritas dengan metode sintetisasi yang juga disebut *Undersampling* dan *Oversampling*. Secara umum pendekatan pada level data memiliki 3 metode yaitu *oversampling, undersampling*, dan *feature selection*.

Pada pendekatan level algoritme atau biasa disebut pendekatan internal dikarenakan terjadi pada pemrosesan data dengan mengimplementasikan metode baru atau meningkatkan kapabilitas metode klasifikasi yang ada untuk dapat menangani permasalahan bias yang disebabkan oleh *imbalanced data.* Pendekatan level algoritme dikategorikan menjadi *ensemble-based method, threshold methods, one class learning, cost sensitive learning,* dan *active learning methods.* Pendekatan ketiga adalah model *hybrid* yang merupakan kombinasi antara kedua pendekatan yang telah dijelaskan sebelumnya. Sebagaimana disampaikan pada penelitian [2] bahwa penanganan *imbalanced dataset* terbukti efektif menggunakan SMOTE yang mana adalah metode *Oversampling.* Sehingga pada penelitian ini metode *Oversampling* akan menjadi solusi untuk menangani *imbalanced data*. Metode *Oversampling* yang digunakan pada penelitian ini adalah *Adaptive Synthetic* (ADASYN). ADASYN merupakan metode *resampling* yang termasuk ke dalam kategori *oversampling,* dan ADASYN merupakan sebuah pengembangan dari metode SMOTE. Dengan menggunakan metode yang lebih mutakhir daripada SMOTE diharapkan dapat meningkatkan performa dari metode klasifikasi yang digunakan nantinya [13].

2.2.4. *Feature Selection*

Seperti yang dapat dilihat pada bagian deskripsi *dataset* bahwa jumlah atribut yang terdapat pada dataset yang digunakan berjumlah 30 atribut dan 1 atribut kelas. Namun apakah semua atribut yang terdapat pada *dataset* tersebut memberikan kontribusi terhadap kelas atribut atau justru mengganggu proses klasifikasi dan dianggap sebagai *noise*. Untuk dapat mengetahui hal tersebut, maka digunakanlah sebuah metode *feature selection* atau *feature ranking* untuk dapat mengetahui berapa jumlah optimal dan mana saja atribut yang memberikan kontribusi terhadap kelas atribut dan dapat meningkatkan performa klasifikasi. Pada penelitian ini metode *feature selection* yang digunakan adalah *Recursive Feature Elimination and Cross-Validation (RFECV)* [8][9]. Penggunaan *feature selection* dapat meningkatkan akurasi dari klasifikasi karena asumsinya hanya atribut-atribut yang dianggap penting saja yang akan digunakan, dan hal tersebut otomatis akan memangkas waktu pemrosesan karena data yang diolah menjadi semakin sedikit. Tetapi karena data yang diolah menjadi sedikit, akan dikhawatirkan juga justru melemahkan kinerja klasifikasi karena dimungkinkan ada data-data yang sebenarnya penting justru tereliminasi oleh proses *feature selection* tersebut. Oleh sebab itu, pada penelitian ini. Kedua hasil akan dibandingkan untuk menunjukkan apakah penggunaan *feature selection* dapat meningkatkan akurasi dari klasifikasi atau tidak. Metode RFECV di implementasikan menggunakan *library* yang terdapat pada scikit.learn Python dengan parameter *cross-validation* = 10.

2.2.5. Klasifikasi

Pada tahap ini, *dataset* yang telah melalui tahap *pre-processing* akan melalui proses klasifikasi untuk kemudian diuji akurasinya pada tahap evaluasi. Karena data yang digunakan adalah *dataset* klasifikasi karyawan yang artinya setiap kelas memiliki pola tersendiri. Metode untuk melakukan klasifikasi sudah sangat banyak dan sudah berkembang menjadi model-model dengan pendekatan yang berbeda-beda pula. Metode-metode dasar dari klasifikasi yang sudah sering digunakan adalah *Decision Tree* yang pertama kali muncul pada penelitian [14]. Untuk model *Decision Tree* saja saat ini sudah ada algoritma hasil pengembangannya yaitu *Random Forest* yang dikembangkan pertama kali oleh Breiman pada penelitian [15]. Selain pendekatan model pohon keputusan seperti dua metode yang telah disebutkan, masih ada model lain yang menggunakan pendekatan *mathematical modelling* seperti *support vector machine* [16][17]. Algoritma klasifikasi lain adalah *Naïve Bayess* yang menggunakan konsep probabilitas dalam proses klasifikasinya[18]. Dari sekian banyak metode klasifikasi baik bentuk dasar maupun pengembangan pasti memiliki keunggulan dan kelemahan masing-masing. Pada penelitian ini, proses perbandingan antara metode-metode klasifikasi akan dilakukan untuk mencari tahu manakah metode yang terbaik yang dapat digunakan pada *dataset* ini. Karena setiap karakteristik data mungkin membutuhkan pendekatan yang berbeda pada pengolahannya.

Dalam perkembangannya, metode klasifikasi sudah sampai di mana terdapat teknik yang menggabungkan lebih dari satu metode klasifikasi untuk membentuk sebuah algoritme klasifikasi yang lebih kuat. Teknik tersebut adalah *ensemble method*, *ensemble method* mengombinasikan lebih dari satu metode klasifikasi dasar dengan tujuan menghasilkan sebuah model yang lebih akurat dibanding dengan metode klasifikasi secara individu [19]. Terdapat beberapa model *ensemble method* yang popular digunakan pada penelitian-penelitian terkait klasifikasi di antaranya, *Stacking, Bagging (Bootstrap Aggregation),* dan *Boosting. Stacking* merupakan sebuah *ensemble method* yang menggunakan lebih dari satu metode klasifikasi dasar. Salah satu dari metode tersebut akan digunakan untuk dilatih menggunakan dataset yang ada, dan kemudian membentuk *dataset* baru. Metode yang digunakan pada tahap awal disebut sebagai *meta learner*. *Dataset* baru yang terbentuk kemudian akan menjadi *input* untuk metode klasifikasi selanjutnya yang kemudian mengeluarkan hasil akhir klasifikasi. Model kedua adalah *Bagging*, metode ini membentuk beberapa model yang dibentuk dari sebuah algoritma klasifikasi yang sama dengan menggunakan sub-sampel yang diambil dari *dataset* secara acak menggunakan metode *bootstrap sampling*. [20]. Model ketiga adalah *Boosting*, metode ini secara umum adalah sebuah metode yang dapat mengonversikan model yang lemah menjadi model yang kuat. *Boosting* secara *incremental* membentuk sebuah *ensemble* dengan melatih setiap model menggunakan data yang sama dengan penyesuaian bobot dari data berdasarkan hasil prediksi terakhir [1]. Penelitian ini menggunakan beberapa algoritma klasifikasi yang akan dibandingkan satu sama lain untuk mencari tahu mana metode yang paling baik untuk *dataset* yang digunakan. Metode-metode tersebut adalah, *Logistic Regression*[18], *Decision Tree* [21], *Naïve Bayes* [22], *k-NN* [23], *CART, Support Vector Machine (SVM), Random Forest* [15], *Stacking* [24][25], *Bagging* [20].

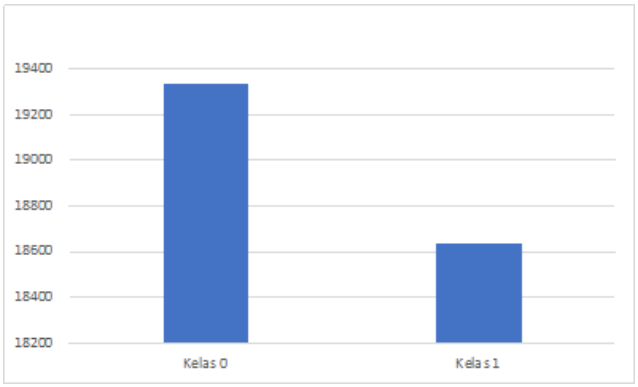
2.2.6. Evaluasi

Pada tahap evaluasi, penelitian ini menggunakan metrik pengukuran *k-fold cross-validation* untuk setiap metode klasifikasi yang digunakan. Cara kerja metode ini adalah membagi *dataset* menjadi subset dengan proporsi tertentu secara acak sebanyak nilai *k* yang dikehendaki. Lalu untuk setiap *subset* akan melalui proses *training-testing* di mana salah satu dari *subset* tersebut akan digunakan sebagai data *test*. Proses tersebut akan diulangi sampai semua *subset* telah berperan menjadi data *test*. Tidak terdapat jumlah baku terkait nilai *k* pada metode pengukuran akurasi ini. Tetapi beberapa penelitian terdahulu menggunakan $k = 5$ atau $k = 10$, di mana saat nilai *k* semakin besar maka perbedaan kuantitas antara data latih dan subset yang telah di sampling ulang menjadi lebih kecil. Ketika perbedaan tersebut menjadi semakin kecil, nilai bias pada metode ini juga menjadi semakin kecil [7]. Pada penelitian ini nilai koefisien *k* yang digunakan adalah $k = 10$.

3. Hasil dan Pembahasan

3.1. *Imbalanced Data*

Pada tahap penanganan *imbalanced data*, data ditangani menggunakan metode ADASYN. Jumlah dari hasil sintetisasi metode ADASYN memang tidak sama persis, tetapi relatif seimbang. Hasil dari penanganan *imbalanced data* dapat dilihat pada Gambar 5.

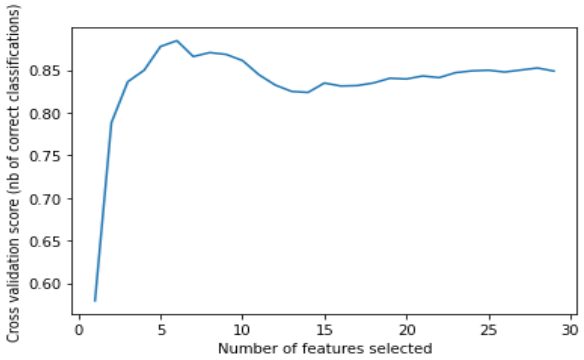


Gambar 5. Sintetisasi menggunakan ADASYN.

Pada Gambar 5 dapat dilihat bahwa secara angka jumlah data pada masing-masing kelas tidak sama persis, tetapi model penyelesaian *imbalanced data* menggunakan ADASYN memang tidak pernah menghasilkan sintetisasi data yang sama untuk masing-masing kelasnya. Hal ini dimungkinkan oleh jumlah data pada kelas minoritas yang terlalu jauh selisihnya dengan kelas mayoritas. Setelah tahap penanganan *imbalanced data* selesai, kemudian data akan menjalani dua jalur berbeda, yaitu tahap seleksi fitur yang akan dibahas pada bagian selanjutnya dan tanpa fitur seleksi yang kemudian nantinya akan dibandingkan seberapa berbeda kedua model tersebut.

3.2. Seleksi Fitur

Data yang telah melalui tahap *preprocessing* yaitu tahap pengisian *missing values* dan penanganan *imbalanced class* selanjutnya akan melalui tahapan seleksi fitur. Seleksi fitur adalah sebuah tahapan di mana memilih atau melakukan filtrasi terhadap atribut data yang banyak kemudian menjadi lebih sedikit dengan tujuan menghilangkan atribut/fitur yang tidak memiliki kontribusi atau pengaruh terhadap kelas data. Dengan menggunakan metode RFECV, fitur data yang berjumlah 29 atribut kemudian akan dilihat pada jumlah fitur dan fitur yang mana saja yang paling optimal untuk digunakan untuk tahap klasifikasi, hasil dari metode RFECV dapat dilihat pada Gambar 6.



Gambar 6. Seleksi fitur menggunakan RFECV.

Dari Gambar 6 dapat dilihat bahwa jumlah optimal yang didapat untuk *dataset* ini adalah 6 fitur dengan hasil *cross-validation* mencapai lebih dari 0.85(85%). Dengan jumlah fitur yang telah diketahui maka dari itu dapat diambil kesimpulan bahwa tidak semua fitur pada *dataset* ini optimal untuk digunakan pada proses klasifikasi, ada lebih banyak data yang mengandung *noise* dibanding data yang menunjang hasil klasifikasi. Hal tersebut dapat dilihat dari hasil grafik yang menurun cukup drastis pada jumlah fitur di atas 6. setelah itu, identifikasi mana saja fitur yang menjadi bagian dari ke-enam fitur yang optimal tersebut akan dilakukan. Menggunakan fungsi yang terdapat pada *library* RVECV untuk menampilkan *rangking* atau peringkat dapat dilihat mana saja fitur yang dianggap optimal. Hasil dari keluaran fungsi tersebut dapat dilihat pada Gambar 7.

```
[ ] selects.ranking_

    array([24,  1, 22,  1,  1,  1, 21, 14,  8, 23, 20, 18, 17,  6,  3,  9, 12,
            7, 13,  5, 19, 10, 15,  1,  1, 16,  2, 11,  4])
```

Gambar 7. Peringkat fitur.

Pada Gambar 7 dapat dilihat peringkat dari setiap fitur atau atribut yang ada pada *dataset*. Urutan yang tertera pada *array* tersebut adalah urutan dari atribut yang ada pada *dataset*. Jika dijabarkan dan dituliskan dalam bentuk nama atribut, atribut atau fitur yang berada pada peringkat 1 adalah antara lain:

- "job_duration_in_current_job_level"
- "job_duration_in_current_person_level"
- "job_duration_in_current_branch"
- "Employee_type"
- "Avg_achievement_%"
- "Last_achievement_%"

Menggunakan hasil seleksi fitur tersebut, maka *dataset* akan di reduksi jumlah atributnya berdasarkan hasil tersebut. Pada tahapan selanjutnya, dua model *dataset* (dengan fitur seleksi dan tanpa fitur seleksi) akan diklasifikasi menggunakan metode-metode yang telah ditentukan sebelumnya yang kemudian dievaluasi menggunakan metrik pengukuran k-*fold cross validation*.

3.2.1. Klasifikasi dan Evaluasi

Untuk mendapatkan hasil yang komprehensif terkait klasifikasi dari dua buah model *dataset* yang telah disiapkan dan kemudian mengetahui mana model klasifikasi yang tepat dan cocok untuk digunakan untuk *dataset* ini, beberapa metode klasifikasi yang berbeda digunakan untuk diketahui hasil evaluasinya menggunakan *k-fold cross validation.* Hasil akurasi dari masing-masing metode klasifikasi untuk masing-masing dataset berbeda dapat dilihat pada Tabel 5.

Tabel 5. Tabel perbandingan akurasi klasifikasi.

No	Metode Klasifikasi	Akurasi	
		Tanpa Fitur Seleksi	Hasil RFECV
1	Random Forest	78%	88%
2	Logistic Regression	63.4%	87.9%
3	Decision tree	78%	85%
4	Naïve Bayes	55.5%	85.2%
5	k-NN	81.1%	86.6%
6	CART	**85.1%**	79.1%
7	SVM	55.5%	87.9%
8	Stacking	82%	87.9%
9	Bagging k-NN	68%	**88%**

Dari tabel 6 dapat dilihat perbandingan dari masing-masing metode klasifikasi yang digunakan terhadap dua buah model *dataset* yaitu *dataset* yang tidak melalui proses seleksi fitur, dan *dataset* yang melalui tahap seleksi fitur. Pada hasil klasifikasi tanpa seleksi fitur, hasil terbaik dihasilkan oleh metode CART dengan 85.1% akurasi. Sedangkan pada *dataset* dengan seleksi fitur, hasil akurasi klasifikasi terbaik dihasilkan oleh metode Bagging k-NN yang merupakan salah satu dari *ensemble technique* dengan akurasi sebesar 88%. Secara umum hasil klasifikasi dari metode-metode yang digunakan pada *dataset* dengan proses seleksi fitur lebih baik dibandingkan pada *dataset* asli tanpa proses seleksi fitur.

4. Kesimpulan

Dari permasalahan yang telah dikemukakan pada bagian pendahuluan bahwa pengolahan data yang efektif dan efisien diperlukan untuk mempercepat proses pengambilan keputusan, terutama untuk menentukan pola untuk mengetahui pegawai yang berada pada *best performance* atau tidak. Sebagai salah satu cara untuk mengefisienkan proses klasifikasi, seleksi fitur dapat memenuhi kebutuhan tersebut karena hasil dati seleksi fitur adalah sebuah *dataset* dengan jumlah atribut yang lebih sedikit dari sebelumnya. Setelah dilakukan evaluasi dari hasil klasifikasi pada *dataset* yang menggunakan seleksi fitur, didapatkan hasil klasifikasi yang lebih baik dibandingkan dengan hasil klasifikasi pada *dataset* tanpa seleksi fitur. Hal tersebut dibuktikan dengan perbandingan antara hasil terbaik dari akurasi klasifikasi *dataset* tanpa seleksi fitur yang dihasilkan oleh CART sebesar 85.1% tidak lebih baik dari metode Bagging k-NN yang di

aplikasikan pada *dataset* dengan seleksi fitur yang menghasilkan 88% akurasi. Sehingga dapat disimpulkan bahwa seleksi fitur dapat meningkatkan akurasi pada klasifikasi.

Daftar Pustaka

[1] H. Jiawei and M. Kamber, *Data mining: concepts and techniques*. 2001.

[2] V. S. Spelmen and R. Porkodi, "A Review on Handling Imbalanced Data," *Proc. 2018 Int. Conf. Curr. Trends Towar. Converging Technol. ICCTCT 2018*, pp. 1–11, 2018, doi: 10.1109/ICCTCT.2018.8551020.

[3] H. Kuswanto and A. Naufal, "Evaluation of performance of drought prediction in Indonesia based on TRMM and MERRA-2 using machine learning methods," *MethodsX*, vol. 6, no. March, pp. 1238–1251, 2019, doi: 10.1016/j.mex.2019.05.029.

[4] Y. Zhu, C. Jia, F. Li, and J. Song, "Inspector: a lysine succinylation predictor based on edited nearest-neighbor undersampling and adaptive synthetic oversampling," *Anal. Biochem.*, vol. 593, no. January, p. 113592, 2020, doi: 10.1016/j.ab.2020.113592.

[5] B. S. Raghuwanshi and S. Shukla, "SMOTE based class-specific extreme learning machine for imbalanced learning," *Knowledge-Based Syst.*, vol. 187, p. 104814, 2020, doi: 10.1016/j.knosys.2019.06.022.

[6] A. Aditsania, Adiwijaya, and A. L. Saonard, "Handling imbalanced data in churn prediction using ADASYN and backpropagation algorithm," *Proceeding - 2017 3rd Int. Conf. Sci. Inf. Technol. Theory Appl. IT Educ. Ind. Soc. Big Data Era, ICSITech 2017*, vol. 2018-January, pp. 533–536, 2017, doi: 10.1109/ICSITech.2017.8257170.

[7] M. Kuhn and K. Johnson, *Applied Predictive Modeling with Applications in R*, vol. 26. 2013.

[8] M. Artur, "Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features," *Procedia Comput. Sci.*, vol. 190, no. 2019, pp. 564–570, 2021, doi: 10.1016/j.procs.2021.06.066.

[9] A. Bahl *et al.*, "Recursive feature elimination in random forest classification supports nanomaterial grouping," *NanoImpact*, vol. 15, no. June, p. 100179, 2019, doi: 10.1016/j.impact.2019.100179.

[10] D. J. Stekhoven and P. Bühlmann, "Missforest-Non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012, doi: 10.1093/bioinformatics/btr597.

[11] F. Tang and H. Ishwaran, "Random forest missing data algorithms," *Stat. Anal. Data Min.*, vol. 10, no. 6, pp. 363–377, 2017, doi: 10.1002/sam.11348.

[12] T. E. Tallo and A. Musdholifah, "The Implementation of Genetic Algorithm in Smote (Synthetic Minority Oversampling Technique) for Handling Imbalanced Dataset Problem," in *2018 4th International Conference on Science and Technology (ICST)*, 2018, pp. 1–4.

[13] Y. Pristyanto, I. Pratama, and A. F. Nugraha, "Data level approach for imbalanced class handling on educational data mining multiclass classification," in *2018 International Conference on Information and Communications Technology, ICOIACT 2018*, 2018, vol. 2018-Janua, doi: 10.1109/ICOIACT.2018.8350792.

[14] S. Gavankar and S. Sawarkar, "Decision Tree: Review of Techniques for Missing Values at Training, Testing and Compatibility," in *2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, 2015, pp. 122–126, doi: 10.1109/AIMS.2015.29.

[15] L. Breiman, "Random Forests," *Int. J. Adv. Comput. Sci. Appl.*, 2001.

[16] R. Berwick, "An Idiot's Guide to Support vector machines (SVMs): A New Generation of Learning Algorithms Key Ideas," pp. 1–28, 2003.

[17] L. Mohan, J. Pant, P. Suyal, and A. Kumar, "Support Vector Machine Accuracy Improvement with Classification," *Proc. - 2020 12th Int. Conf. Comput. Intell. Commun. Networks, CICN 2020*, pp. 477–481, 2020, doi: 10.1109/CICN49253.2020.9242572.

[18] M. R. Romadhon and F. Kurniawan, "A Comparison of Naive Bayes Methods, Logistic Regression and KNN for Predicting Healing of Covid-19 Patients in Indonesia," *3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIConCIT 2021*, pp. 41–44, 2021, doi: 10.1109/EIConCIT50028.2021.9431845.

[19] S. Ahmed, A. Mahbub, F. Rayhan, R. Jani, S. Shatabda, and D. M. Farid, "Hybrid Methods for Class Imbalance Learning Employing Bagging with Sampling Techniques," *2nd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. CSITSS 2017*, pp. 1–5, 2018, doi: 10.1109/CSITSS.2017.8447799.

[20] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996, doi: 10.1007/BF00058655.

[21] L. Rokach and O. Maimon, *Data Mining With Decision Trees: Theory and Applications*, 2nd ed.

River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2014.

[22] D. Soni, "Introduction to Naive Bayes Classification," 2018. [Online]. Available: https://towardsdatascience.com/introduction-to-naive-bayes-classification-4cffabb1ae54. [Accessed: 06-May-2019].

[23] C. Tang, P. Xu, Z. Luo, G. Zhao, and T. Zou, "Automatic facial expression analysis of students in teaching environments," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9428, pp. 439–447, 2015, doi: 10.1007/978-3-319-25417-3_52.

[24] K. Leartpantulak and Y. Kitjaidure, "Music genre classification of audio signals using particle swarm optimization and stacking ensemble," *iEECON 2019 - 7th Int. Electr. Eng. Congr. Proc.*, pp. 1–4, 2019, doi: 10.1109/iEECON45304.2019.8938995.

[25] J. Ling and G. Li, "A two-level stacking model for detecting abnormal users in Wechat activities," *Proc. - 2019 Int. Conf. Inf. Technol. Comput. Appl. ITCA 2019*, pp. 229–232, 2019, doi: 10.1109/ITCA49981.2019.00057.