

Analisis Optimasi Algoritma Klasifikasi Support Vector Machine, Decision Trees, dan Neural Network Menggunakan Adaboost dan Bagging

Agus Heri Yunial

Teknik Informatika, Universitas Pamulang, Jl. Raya Puspitek No.46, Buaran, Serpong, Tangerang Selatan, Banten, Indonesia, 15310
e-mail: dosen02525@unpam.ac.id

Submitted Date: August 26th, 2020
Revised Date: September 29th, 2020

Reviewed Date: September 22nd, 2020
Accepted Date: September 30th, 2020

Abstract

The accuracy value of a classification algorithm shows whether the algorithm is good or not in classifying data which can affect the results of the classification method in data mining processing. In this study, the author will analyze the effect of optimization using the adaboost and bagging methods on the results of the classification algorithm accuracy value on support vector machines, decision trees, and neural networks. This study uses a software in data mining processing that is using the Weka application version 3.8.1. The test method used was a percentage split of 70%. In this study, the results show that adaboost optimization can increase the accuracy value of the support vector machine algorithm from 88.93% to 89.10%, decision trees from 90.24% to 90.36%, and neural network from 88.53% to 88.61%, while bagging optimization can only increase Algoritma decision trees become 90.55%, and the neural network becomes 90.38%, because the accuracy value of the support vector machine algorithm is the same as the accuracy value of bagging, which is 88.93%.

Keywords: Data Mining; Clasification; AdaBoost; Bagging; Support Vector Machine; Decision Trees; Neural Network

Abstrak

Nilai akurasi dari suatu algoritma klasifikasi menunjukkan baik tidaknya algoritma tersebut dalam pengklasifikasian data yang bisa mempengaruhi hasil dari metode klasifikasi tersebut dalam pengolahan data mining. Pada penelitian ini, penulis akan melakukan analisis pengaruh dari optimasi menggunakan metode adaboost dan bagging terhadap hasil dari nilai akurasi algoritma klasifikasi pada support vector machine, decision trees, dan neural network. Penelitian ini menggunakan salah satu perangkat lunak dalam pengolahan data mining yaitu menggunakan aplikasi Weka versi 3.8.1. Metode test yang digunakan adalah percentage split sebesar 70%. Pada penelitian ini didapatkan hasil bahwa optimasi adaboost mampu meningkatkan nilai akurasi dari algoritma support vector machine dari 88.93% menjadi 89.10% , decision trees dari 90.24% menjadi 90.36%, maupun neural network dari 88.53% menjadi 88.61%, sedangkan untuk optimasi bagging hanya mampu meningkatkan algoritma decision trees menjadi 90.55%, dan neural network menjadi 90.38%, karena nilai akurasi pada algoritma support vector machine sama nilainya dengan nilai akurasi dari bagging yaitu sebesar 88.93%.

Kata Kunci : Data Mining, Klasifikasi, AdaBoost, Bagging, Support Vector Machine, Decision Trees, Neural Network.

1. Pendahuluan

Kegiatan transaksi saat ini hampir seluruhnya, khususnya perbankan dilakukan secara digital dan terkomputerisasi yang pastinya tersimpan dalam setiap database. Dalam database tersebut banyak tumpukan data-data yang

sebenarnya bisa dijadikan informasi yang bermanfaat. Sebuah data bisa menjadi informasi yang sangat berguna jika dilakukan teknik pengolahan data yang sesuai.

Data mining adalah salah satu metode pengolahan data yang bisa menggali informasi

yang tersembunyi dari sebuah database. Ada beberapa metode pengolahan data dalam data mining, seperti klasifikasi, asosiasi, klustering dan berbagai metode lainnya. Pada penelitian ini, penulis akan membahas salah satu metode dalam data mining yaitu klasifikasi.

Klasifikasi merupakan suatu kegiatan mencari fungsi atau model yang menerangkan perbedaan kelas suatu data atau membedakan cara atau konsep, untuk memprediksi kelas yang tidak diketahui labelnya dari suatu objek. Ada banyak algoritma pada data mining untuk dapat digunakan dalam metode klasifikasi antara lain Neural Network, Decision Tree, k-Nearest Neighbor, Naive Bayes, Support Vector Machines, Zero-R, dan masih banyak yang lainnya.

Nilai akurasi dari suatu algoritma klasifikasi menunjukkan baik tidaknya suatu algoritma tersebut digunakan pada suatu metode klasifikasi yang bisa mempengaruhi data yang dihasilkan dari penggunaan algoritma tersebut. Suatu algoritma bisa ditingkatkan kembali nilai akurasi dengan beberapa metode atau algoritma lain, contohnya adaboost dan bagging. Dari beberapa penelitian membuktikan bahwa kedua metode tersebut terbukti dapat meningkatkan nilai akurasi dari suatu algoritma khususnya klasifikasi. Dengan demikian tingkat keberhasilan dari prediksi algoritma tersebut lebih tepat dan sesuai dalam membrikan labelnya. Pada penelitian ini, penulis menggunakan optimasi adaboost dan bagging untuk meningkatkan nilai akurasi dari algoritma klasifikasi Support Vector Machine, Decision Trees, dan Neural Network dengan menggunakan sebuah dataset yang disediakan oleh UCI machine learning yaitu dataset Bank marketing.

Sebelumnya, terdapat banyak penelitian yang dilakukan mengenai klasifikasi data mining dengan membandingkan beberapa algoritma klasifikasi. Beberapa literatur mengenai perbandingan kinerja dari beberapa algoritma klasifikasi menggunakan dataset yang disediakan oleh UCI Machine Learning, khususnya dataset Bank Marketing. Contohnya adalah penelitian yang dilakukan oleh Niharika Sharma dkk (Sharma, Kaur, Gandotra, & Sharma, 2015) yang hasil penelitiannya menunjukkan algoritma Random forest sebagai algoritma yang memiliki nilai akurasi tertinggi sebesar 71%, sedangkan untuk SVM 54%, NN 63%, dan DT 66%.

Penelitian berikutnya yaitu yang sudah dilakukan Ali Keles bersama Ayturk Keles (Keles & Keles, 2015) yang hasil penelitiannya menunjukkan algoritma Neural Networks memiliki

tingkat akurasi tertinggi dengan nilai akurasi 88.31%, sedangkan SVM 72.13% dan DT 72.84%.

Penelitian yang dilakukan oleh Hilda Amalia (Amalia, 2016) yang hasil penelitiannya menunjukkan algoritma SVM dengan nilai akurasi tertinggi sebesar 89.27%, sedangkan NN 88.41% dan DT 88.48%.

Penelitian yang dilakukan oleh Olatunji Apampa (Apampa, 2016) yang hasil penelitiannya menunjukkan algoritma Decision Trees memiliki performa yang lebih baik yang memiliki nilai akurasi 76.6%.

Dari beberapa literatur di atas dapat disimpulkan algoritma klasifikasi yang memiliki nilai akurasi terbesar adalah Support Vector Machine dengan nilai akurasi 89.27%, Decision trees dengan nilai akurasi 88.48%, dan Neural Network dengan nilai akurasi 88.41%.

Selain literatur di atas yang membandingkan beberapa algoritma klasifikasi pada data mining, terdapat juga beberapa literatur yang menunjukkan kinerja terbaik dari algoritma Support Vector Machine, Neural Network, dan Decision Trees dibandingkan dengan algoritma klasifikasi yang lain dengan menggunakan beberapa dataset yang berbeda dengan dataset di atas. Contohnya adalah Penelitian yang Dewi Sartika bersama Dana Indra Sensuse (Sartika & Sensuse, 2017). Penelitian ini melakukan perbandingan dari klasifikasi seperti Nearest Neighbour, Naive Bayes, dan Decision Trees dalam menentukan pola pakaian dengan menggunakan aplikasi Weka. Hasil dari penelitiannya didapatkan nilai akurasi Decision Trees yang paling unggul yaitu mencapai 75.6%.

Penelitian yang dilakukan Rizki Djuwi Probo, R. Rumani M bersama Budhi Irawan (Probo, Irawan, & M, 2016). Penelitian ini membandingkan algoritma K-Nearest Neighbors dan Support Vector Machine. Pada penelitian mereka didapatkan nilai akurasi dari SVM sebesar 69,55%, sedangkan nilai akurasi KNN sebesar 49,62%.

Penelitian yang dilakukan oleh Ahmad Ilham (Ilham, 20017). Penelitian ini membandingkan kinerja dari beberapa algoritma klasifikasi. Penelitian tersebut menyimpulkan bahwa nilai akurasi dari algoritma Neural Network sebesar dengan nilai akurasi 97.14% lebih unggul dari Linear Regression, Decision Trees, Support Vector Machine, dan Naive Bayes.

Penelitian yang dilakukan oleh Khairul Sani, bersama Silmi Fauziati dan Wing Wahyu Winarno (Sani, Winarno, & Fauziati, 2016). Pada penelitian tersebut didapatkan bahwa algoritma trees C4.5

nilai akurasi sebesar 98.54 % dan nilai tersebut lebih besar dari nilai akurasi algoritma Naive Bayes dan Neural Network.

Atas dasar itulah pada penelitian ini akan mencoba mengoptimalkan nilai akurasi dari algoritma klasifikasi terbaik dari literatur di atas seperti Support Vector Machine, Decision Trees, dan Neural Network yang menggunakan dataset Bank Marketing. Optimasi adaboost dan bagging adalah optimasi yang akan digunakan penulis pada penelitian kali ini, karena dalam beberapa penelitian lain optimasi tersebut mampu meningkatkan akurasi dari beberapa algoritma di atas. Hal ini berdasarkan pada literatur penelitian yang dilakukan oleh Eka Listiana dan Much Aziz Muslim (Listiana & Muslim, 2017). Penelitian mereka menggunakan optimasi adaboost pada algoritma Support Vector Machine. Hasil penelitian mereka optimasi yang dilakukan pada algoritma Support Vector Machine menggunakan adaboost menggunakan data chronic kidney disease dapat meningkatkan nilai akurasi sebanyak 37%.

Penelitian yang dilakukan oleh Achmad Bisri dari Universitas Pamulang dan Romi Satria Wahono (Bisri & Wahono, 2015). Mereka menggunakan optimasi adaboost pada algoritma Decision Trees. Setelah dilakukan optimasi nilai akurasi yang didapatkan adalah 90,45% yang sebelumnya hanya 87,18%.

Penelitian yang dilakukan oleh Rizki Tri Prasetyo dan Pratiwi (Prasetyo & Pratiwi, 2015). Mereka menggunakan optimasi bagging pada algoritma klasifikasi naïve bayes, k-nearest neighbor, dan decision trees. Pada penelitian mereka menunjukkan nilai akurasi dari algoritma di atas memiliki kecenderungan meningkat. Misalnya saja pada algoritma C4.5 dengan database Heart-disease memiliki nilai akurasi 73.70% dan setelah

Tools dalam pengolahan optimasi algoritma klasifikasi pada penelitian ini adalah menggunakan Weka. Karena dalam beberapa literatur ditemukan aplikasi Weka mampu mengklasifikasikan data dengan baik. Contohnya pada penelitian yang dilakukan Dakić Dušanka dkk (Dušanka, Darko, Srdjan, Marko, & Teodora, 2017). Penelitian ini membandingkan beberapa aplikasi pengolahan data mining yang bersifat opensource seperti Weka, rapidminer, apache spark, azure ML studio, dan H2O. Penelitian ini menggunakan metode klasifikasi dengan algoritma decision trees dan dataset iris dari UCI Machine Learning. Penelitian ini menunjukkan hasil bahwa aplikasi Weka memiliki kinerja cukup baik dibanding dengan aplikasi lainnya. Terlihat dari nilai akurasi, TPR–

True positive rate (sensitivity) yaitu 0.98, dan TNR–True negative rate (specificity) 0.99.

Penelitian yang dilakukan oleh Hayden Wimmer (Wimmer & Powell, 2015). Penelitian ini membandingkan aplikasi Orange, Tanagra, Rapidminer, KNIME, R, dan Weka dalam mengolah data. Penilaian yang dibandingkan adalah berdasarkan nilai matrix yang dihasilkan. Dan diperoleh dari beberapa metode data mining yang digunakan, ternyata Weka menunjukkan bahwa aplikasi open source yang paling mendukung dalam beberapa metode yang dilakukan.

Berdasarkan Literatur Review di atas, maka penelitian ini mencoba membandingkan kinerja algoritma Support Vector Machines, Decision Trees, dan Neural Network sebelum dan sesudah dioptimasi dengan teknik adaboost dan bagging dengan menggunakan sekumpulan data berupa dataset dalam hal ini adalah data Bank Marketing yang disediakan oleh UCI Machine Learning dengan menggunakan salah satu aplikasi pengolahan data mining yaitu Weka versi 3.8.1.

Atas dasar itulah penelitian ini dibuat untuk mengetahui penggunaan teknik adaboost dan bagging dalam meningkatkan keakurasian dari algoritma Support Vector Machine, Decision Trees, dan Neural Network pada klasifikasi data mining.

2. Metode Penelitian

2.1 Analisis Kebutuhan

Penelitian ini menggunakan perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut:

- a. Perangkat Keras
Laptop dengan spesifikasi seperti Processor : Core I5, Ram : 8 GB dan Hardisk 1 TB
- b. Perangkat Lunak
 - Sistem Operasi Windows 10
 - Microsoft Office 2016
 - Aplikasi Weka versi 3.8.1
- c. Data
Data yang digunakan pada penelitian ini adalah dataset yang disediakan situs UCI Machine Learning dengan alamat yaitu data Bank marketing yang dapat diunduh pada situs (<https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>). Dataset tersebut terdiri dari beberapa atribut seperti:

Tabel 1. Atribut dataset bank marketing

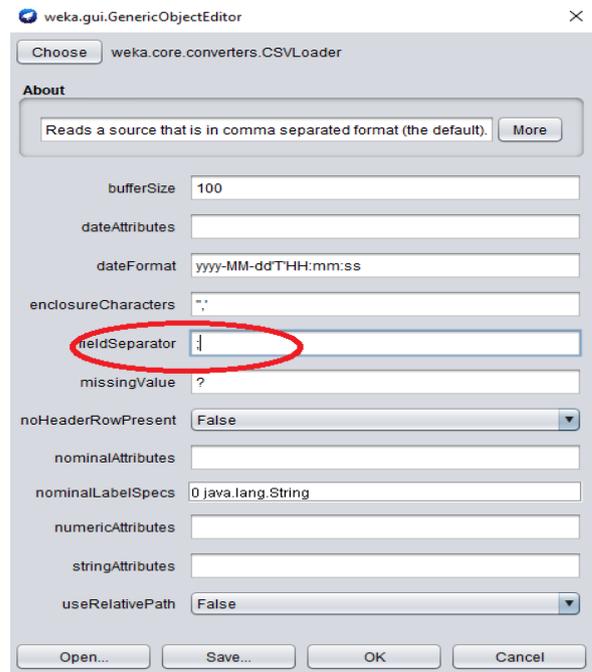
Nama Atribut	Tipe	Keterangan
<i>Age</i>	Numerik	Umur
<i>Job</i>	Kategori	Pekerjaan
<i>Martial</i>	Kategori	Status pernikahan
<i>Education</i>	Kategori	Pendidikan terakhir
<i>Default</i>	Kategori	Memiliki hutang yang menunggak?
<i>Balance</i>	Numerik	Saldo nasabah (dalam mata uang Euro)
<i>Housing</i>	Kategori	Memiliki pinjaman rumah?
<i>Loan</i>	Kategori	Adakah pinjaman?
<i>Contact</i>	Kategori	Jenis komunikasi
<i>Day</i>	Numerik	Tanggal kontak terakhir
<i>Month</i>	Kategori	Bulan kontak terakhir dalam tahun ini
<i>Duration</i>	Numerik	Waktu terakhir melakukan panggilan (waktu dalam detik)
<i>Campaign</i>	Numerik	Selama penawaran kredit, berapa kontak yang dilakukan
<i>Pdays</i>	Numerik	Berapa hari nasabah telah dihubungi dari penawaran sebelumnya (-1 berarti nasabah belum pernah dihubungi sebelumnya)
<i>Previous</i>	Numerik	Sebelum penawaran, berapa panggilan yang telah dilakukan pada nasabah
<i>Poutcome</i>	Kategori	Penawaran yang lalu
<i>Output</i>	Kategori	Hasil penawaran saat ini

Terdapat 2 file dataset yang diunduh yaitu bank.csv dan bank-full.csv. Data yang akan digunakan sebagai data adalah data bank-full.csv yang terdiri dari 45211 instances, 17 attributes. Dalam aplikasi Weka versi 3.8.1 ini, dataset tersebut perlu dilakukan konversi menjadi format .arff agar data bisa diproses pada aplikasi Weka.

Cara mengkonversinya adalah ditunjukkan pada Gambar 1 dan Gambar 2. Pada saat data ini dibuka pada Weka, akan muncul peringatan seperti Gambar 1. Pilih saja Use Converter dan ubah fieldSeparator dari koma [,] menjadi titik koma [;], lalu klik OK. Kemudian save dengan format .arff.



Gambar 1. Peringatan pada saat membuka file bank-full



Gambar 2. Form converter Weka

2.2 Perancangan Penelitian

Langkah pertama dalam penelitian ini adalah melakukan identifikasi masalah yaitu dengan menentukan batasan masalah yang akan dibahas sehingga tujuan penelitian ini tidak akan keluar dari pembatasan masalah tersebut. Kemudian langkah kedua membuat latar belakang masalah untuk menerangkan latar belakang masalah yang terjadi. Kemudian langkah ketiga adalah memilih metode untuk digunakan dalam penelitian dan mencari literatur terkait yang mendukung untuk penggunaan metode tersebut dalam menyelesaikan permasalahan-permasalahan yang ada dalam penelitian.

Langkah keempat adalah mencari data yang akan diteliti dengan menggunakan metode yang sudah ditetapkan agar permasalahan penelitian bisa terjawab dan memberikan hasil yang diharapkan sesuai tujuan penelitian yang sudah dibuat. Langkah kelima yaitu penulisan akhir berupa jurnal ilmiah tertulis agar dapat bisa berguna untuk penelitian selanjutnya serta dapat dipertanggungjawabkan hasilnya.

2.3 Teknik Analisis

Pada penelitian ini, akan dilakukan optimasi dari 3 algoritma klasifikasi pada data mining yaitu Support Vector Machine, Decision Trees, dan Neural Network untuk mengolah dataset Bank Marketing dari UCI Machine Learning dengan menggunakan aplikasi Weka versi 3.8.1 dengan metode optimasi adaboost dan bagging.

Data Bank Marketing dari UCI yang akan digunakan pada penelitian ini. Pada dataset tersebut terdapat 2 buah file, yaitu file bank dan bank-full. Yang akan digunakan pada penelitian ini adalah bank-full dengan informasi file sebagai berikut:

- Ukuran data : 4,503 KB
- Jumlah Record/Instances : 45211
- Jumlah Atributs : 17

Pada penelitian ini akan dibandingkan hasil nilai akurasi sebelum dan setelah dilakukan optimasi dengan adaboost dan bagging pada algoritma Support Vector Machine, Decision Trees, dan Neural Network. Dan akan dianalisa sejauh mana teknik optimasi tersebut dalam meningkatkan nilai akurasi pada algoritma tersebut. Serta dapat dilihat algoritma yang memiliki nilai akurasi terbaik dari hasil perbandingan yang diperoleh.

Percentage split (70%), salah satu mode yang akan digunakan pada penelitian ini yaitu mengevaluasi hasil akurasi dari algoritma dengan menggunakan 70% dari dataset sebagai data training dan 30% dari dataset sebagai data testing. Hasil nilai akurasi akan ditulis pada Tabel 2.

Tabel 2. Form perbandingan nilai akurasi sebelum dan setelah optimasi

Algoritma	Mode Pengujian		
	Nonoptimasi	Adaboost	Bagging
Support Vector Machine			
Decision Tree			
Neural Network			

3. Hasil

Proses perbandingan nilai akurasi dari algoritma klasifikasi pada penelitian ini menggunakan aplikasi Weka versi 3.8.1 dengan metode pengujian yang dilakukan adalah Percentage split sebanyak 70%. Yang artinya dataset akan dibagi sebesar 70% untuk data training untuk pembentukan model, dan 30% sebagai data testing yang akan dibagi secara acak oleh sistem.

3.1 Hasil Akurasi Algoritma Support Vector Machine

Pada algoritma klasifikasi Support Vector Machine yang dilakukan pengukuran akurasi menggunakan aplikasi Weka dengan metode testing percentage split didapat nilai akurasi sebagai berikut:

a. Support Vector Machine tanpa optimasi

Setelah dilakukan pengolahan data mining pada aplikasi Weka dengan algoritma support vector machine tanpa dilakukan optimasi didapat nilai akurasi sebesar 88.93% seperti terlihat pada Gambar 3.

```

=== Summary ===

Correctly Classified Instances      12062          88.9331 %
Incorrectly Classified Instances    1501           11.0669 %
Kappa statistic                    0.2396
Mean absolute error                 0.1107
Root mean squared error             0.3327
Relative absolute error             53.2834 %
Root relative squared error         102.572 %
Total Number of Instances          13563

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.986   0.820   0.899     0.986   0.940     0.295   0.583   0.898   no
          0.180   0.014   0.628     0.180   0.280     0.295   0.583   0.211   yes
Weighted Avg.   0.889   0.724   0.866     0.889   0.861     0.295   0.583   0.816

=== Confusion Matrix ===

      a    b  <-- classified as
11770  173 |  a = no
 1328   292 |  b = yes
    
```

Gambar 3. Hasil akurasi support vector machine tanpa optimasi

Dapat dilihat pada Gambar 3 bahwa jumlah instances untuk dijadikan sebagai data test sebanyak 13.563 instances. Confusion matrix yang diperoleh jika dibuatkan tabel bisa terlihat pada Tabel 3.

Tabel 3. Confusion matrix support vector machine

	Prediksi (yes)	Prediksi (no)
Aktual (yes)	11770	173
Aktual (no)	1328	292

Dari tabel di atas bisa diketahui bahwa:

$$TP = 11770$$

$$FN = 173$$

$$FP = 1328$$

$$TN = 292$$

$$\begin{aligned} \text{Nilai akurasi} &= (TP+TN)/(TP+TN+FP+FN) \\ &= (11770+292)/(11770+292+1328+173) \\ &= 12062/13563 = 0.889331269 \end{aligned}$$

Jadi nilai akurasi yang diperoleh sebesar 88.93%.

b. Support Vector Machine dengan ditambahkan optimasi adaboost

```

=== Summary ===

Correctly Classified Instances      12085          89.1027 %
Incorrectly Classified Instances    1478           10.8973 %
Kappa statistic                    0.4286
Mean absolute error                 0.1453
Root mean squared error             0.2742
Relative absolute error             69.9781 %
Root relative squared error        84.5493 %
Total Number of Instances          13563

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0.953   0.564   0.926     0.953   0.939     0.433   0.882    0.978    no
                0.436   0.047   0.556     0.436   0.489     0.433   0.882    0.508    yes
Weighted Avg.   0.891   0.502   0.881     0.891   0.885     0.433   0.882    0.922

=== Confusion Matrix ===

  a    b  <-- classified as
11379  564 |  a = no
  914   706 |  b = yes
    
```

Gambar 4. Hasil akurasi support vector machine dengan adaboost

Kita dapat melihat dari Gambar 4 bahwa jumlah instances untuk dijadikan sebagai data test sebanyak 13.563 instances. Confusion matrix yang diperoleh jika dibuatkan tabel bisa terlihat pada Tabel 5.

Tabel 5. Confusion matrix support vector machine dengan adaboost

	Prediksi (yes)	Prediksi (no)
Aktual (yes)	11379	564
Aktual (no)	914	706

Jumlah iterasi pada optimasi adaboost bisa mempengaruhi nilai akurasi yang dihasilkannya. Untuk itulah dilakukan beberapa percobaan optimasi dengan beberapa jumlah iterasi seperti pada Tabel 4.

Tabel 4. Nilai akurasi support vector machine dengan adaboost periterasinya

Jumlah Iterasi	Nilai Akurasi
1	88.93%
5	89.36%
10	89.10 %
20	89.10 %

Setelah dilakukan pengolahan data mining pada aplikasi Weka dengan algoritma support vector machine dan dilakukan optimasi menggunakan adaboost didapat nilai akurasi yang nilainya stabil pada nilai 89.10% dimulai pada iterasi ke 10. Hasil nilai akurasi dapat terlihat pada Gambar 4.

Dari Tabel 5 bisa diketahui bahwa:

$$TP = 11379$$

$$FN = 564$$

$$FP = 914$$

$$TN = 706$$

$$\begin{aligned} \text{Nilai akurasi} &= (TP+TN)/(TP+TN+FP+FN) \\ &= (11379+706)/(11379+706+914+564) \\ &= 12085/13563 = 0.89102 \end{aligned}$$

Jadi nilai akurasi yang diperoleh sebesar 89.10%.

c. Support Vector Machine tanpa ditambahkan optimasi bagging

Jumlah iterasi pada optimasi bagging bisa mempengaruhi nilai akurasi yang dihasilkan. Untuk itulah dilakukan beberapa percobaan optimasi dengan beberapa jumlah iterasi yang ditunjukkan pada Tabel 6.

Setelah dilakukan pengolahan data mining pada aplikasi Weka dengan algoritma Support Vector Machine dengan penambahan optimasi bagging didapat nilai akurasi yang stabil pada nilai

88.93% dari optimasi ke 1 sampai optimasi 10 nilai akurasi tidak berubah. Hasil nilai akurasi ditunjukkan oleh Gambar 5.

Tabel 6. Nilai akurasi support vector machine dengan bagging periterasinya

Jumlah Iterasi	Nilai Akurasi
1	88.93%
5	88.93%
10	88.93%

```

=== Summary ===

Correctly Classified Instances      12062          88.9331 %
Incorrectly Classified Instances    1501           11.0669 %
Kappa statistic                    0.2396
Mean absolute error                 0.1107
Root mean squared error             0.3327
Relative absolute error             53.2834 %
Root relative squared error         102.572 %
Total Number of Instances          13563

=== Detailed Accuracy By Class ===

              TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
              -----  -----  -
              0.986   0.020   0.899     0.986   0.940     0.295   0.583    0.898    no
              0.180   0.014   0.628     0.180   0.280     0.295   0.583    0.211    yes
Weighted Avg.   0.889   0.724   0.866     0.889   0.861     0.295   0.583    0.816

=== Confusion Matrix ===

      a    b  <-- classified as
11770  173 |    a = no
 1328  292 |    b = yes
    
```

Gambar 5. Hasil akurasi support vector machine dengan bagging

Dapat dilihat pada Gambar 5 bahwa jumlah instances untuk dijadikan sebagai data test sebanyak 13.563 instances. Confusion matrix yang diperoleh jika dibuatkan tabel bisa terlihat pada Tabel 7.

Tabel 7. Confusion matrix support vector machine dengan bagging

	Prediksi (yes)	Prediksi (no)
Aktual (yes)	11770	173
Aktual (no)	1328	292

Dari Tabel 7 bisa diketahui bahwa:

TP = 11770

FN = 173

FP = 1328

TN = 292

Nilai akurasi = $(TP+TN)/(TP+TN+FP+FN)$

$$= (11770+292)/(11770+292+1328+173)$$

$$= 12062/13563 = 0.889331269$$

Jadi nilai akurasi yang diperoleh sebesar 88.93%.

3.2 Hasil Akurasi Algoritma Decision Tree

Pada algoritma kalsifikasi decision trees yang dilakukan pengkuruan akurasi menggunakan aplikasi Weka dengan metode testing percentage split didapat nilai akurasi sebagai berikut:

a. Decision Trees tanpa ditambahkan optimasi

Setelah dilakukan pengolahan data mining pada aplikasi Weka dengan algoritma decision trees tanpa dilakukan optimasi didapat nilai akurasi sebesar 90.24% yang ditunjukkan oleh Gambar 6.

```

=== Summary ===

Correctly Classified Instances      12240          90.2455 %
Incorrectly Classified Instances    1323           9.7545 %
Kappa statistic                    0.4801
Mean absolute error                 0.1261
Root mean squared error             0.2784
Relative absolute error             60.7022 %
Root relative squared error        85.8373 %
Total Number of Instances          13563

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0.962   0.533   0.930     0.962   0.946     0.486   0.837    0.953    no
                0.467   0.038   0.622     0.467   0.533     0.486   0.837    0.498    yes
Weighted Avg.   0.902   0.474   0.893     0.902   0.896     0.486   0.837    0.898

=== Confusion Matrix ===

      a    b  <-- classified as
11484  459 |  a = no
   864   756 |  b = yes
    
```

Gambar 6. Hasil akurasi decision trees tanpa optimasi

Kita dapat melihat dari gambar tersebut diatas bahwa jumlah instances untuk dijadikan sebagai data test sebanyak 13.563 instances. Confusion matrix yang diperoleh jika dibuatkan tabel bisa terlihat pada tabel berikut:

Tabel 8. Confusion matrix decision tree

	Prediksi (yes)	Prediksi (no)
Aktual (yes)	11484	459
Aktual (no)	864	756

Dari Tabel 8 bisa diketahui bahwa:

TP = 11484

FN = 459

FP = 864

TN = 756

Nilai akurasi = $(TP+TN)/(TP+TN+FP+FN)$

$$= (11484+756)/(11484+756+864+459)$$

$$= 12240/13563 = 0.902455$$

Jadi nilai akurasi yang diperoleh sebesar 90.24%.

b. Decision Trees dengan ditambahkan optimasi adaboost

Jumlah iterasi pada optimasi adaboost bisa mempengaruhi nilai akurasi yang dihasilkannya. Untuk itulah dilakukan beberapa percobaan optimasi dengan beberapa jumlah iterasi ditunjukkan pada Tabel 9.

Setelah dilakukan pengolahan data mining pada aplikasi Weka dengan algoritma decision trees dan dilakukan optimasi menggunakan

adaboost didapat yang stabil pada nilai 90.36% dari optimasi ke 500 sampai optimasi 700 nilai akurasi tidak berubah. Hasil nilai akurasi dapat terlihat pada Gambar 7.

Tabel 9. Nilai akurasi decision trees dengan adaboost periterasinya

Jumlah Iterasi	Nilai Akurasi
1	90.24%
5	89.46%
10	89.51%
20	90.15%
30	90.06%
40	90.23%
50	90.11%
60	90.21%
70	90.15%
80	90.27%
90	90.32%
100	90.30%
150	90.30%
200	90.26%
300	90.47%
400	90.30%
500	90.36%
600	90.36%
700	90.36%
800	90.33%
900	90.37%
1000	90.29%

```

== Summary ==

Correctly Classified Instances      12255      90.3561 %
Incorrectly Classified Instances    1308       9.6439 %
Kappa statistic                    0.479
Mean absolute error                 0.0964
Root mean squared error             0.31
Relative absolute error              46.3966 %
Root relative squared error         95.5971 %
Total Number of Instances          13563

== Detailed Accuracy By Class ==

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
-----
Weighted Avg.  0.904  0.482  0.894  0.904  0.897  0.487  0.743  0.884

== Confusion Matrix ==

      a      b  <-- classified as
11514  429 |      a = no
      879  741 |      b = yes
    
```

Gambar 7. Hasil akurasi decision trees dengan adaboost

Kita dapat melihat dari Gambar 7 bahwa jumlah instances untuk dijadikan sebagai data test sebanyak 13.563 instances. Confusion matrix yang diperoleh jika dibuatkan tabel bisa terlihat pada Tabel 10.

Tabel 10. Confusion matrix decision trees dengan adaboost

	Prediksi (yes)	Prediksi (no)
Aktual (yes)	11514	429
Aktual (no)	879	741

Dari Tabel 10 bisa diketahui bahwa:

TP = 11514

FN = 429

FP = 879

TN = 741

Nilai akurasi = $(TP+TN)/(TP+TN+FP+FN)$

$$= (11514+741)/(11514+741+879+429)$$

$$= 12255/13563 = 0.90356$$

Jadi nilai akurasi yang diperoleh sebesar 90.36%.

c. Decision Trees tanpa ditambahkan optimasi bagging

Jumlah iterasi pada optimasi bagging bisa mempengaruhi nilai akurasi yang dihasilkannya.

Untuk itulah dilakukan beberapa percobaan optimasi dengan beberapa jumlah iterasi seperti pada Tabel 11.

Tabel 11. Nilai akurasi decision trees dengan bagging periterasinya

Jumlah Iterasi	Nilai Akurasi
1	89.11 %
5	90.13%
10	90.31%
20	90.35%
30	90.30%
40	90.58%
50	90.48%
60	90.59%
70	90.50%
80	90.59%
90	90.55%
100	90.55%

Setelah dilakukan pengolahan data mining pada aplikasi Weka dengan algoritma decision trees dengan penambahan optimasi bagging didapat nilai akurasi yang memiliki nilai stabil sebesar 90.55% dimulai pada iterasi ke 90. Hasil nilai akurasi dapat terlihat pada Gambar 8.

```

=== Summary ===

Correctly Classified Instances      12281          90.5478 %
Incorrectly Classified Instances    1282           9.4522 %
Kappa statistic                    0.5035
Mean absolute error                0.1241
Root mean squared error            0.2566
Relative absolute error            59.773 %
Root relative squared error        79.1097 %
Total Number of Instances          13563

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.961   0.506   0.933     0.961   0.947     0.508   0.925   0.988   no
          0.494   0.039   0.634     0.494   0.555     0.508   0.925   0.592   yes
Weighted Avg.   0.905   0.450   0.898     0.905   0.900     0.508   0.925   0.941

=== Confusion Matrix ===

      a    b  <-- classified as
11480  463 |  a = no
   819  801 |  b = yes
    
```

Gambar 8. Hasil akurasi decision trees dengan bagging

Kita dapat melihat bahwa jumlah instances untuk dijadikan sebagai data test sebanyak 13.563 instances. Confusion matrix yang diperoleh jika dibuatkan tabel bisa terlihat pada tabel berikut:

Tabel 12. Confusion matrix decision trees dengan bagging

	Prediksi (yes)	Prediksi (no)
Aktual (yes)	11480	463
Aktual (no)	819	801

Dari tabel di atas bisa diketahui bahwa:

TP = 11480

FN = 463

FP = 819

TN = 801

$$\begin{aligned}
 \text{Nilai akurasi} &= (TP+TN)/(TP+TN+FP+FN) \\
 &= (11480+801)/(11480+801+819+463) \\
 &= 12281/13563 = 0.905478
 \end{aligned}$$

Jadi nilai akurasi yang diperoleh sebesar 90.55%.

3.3 Hasil Akurasi Algoritma Neural Network

Pada algoritma kalsifikasi Neural Network yang dilakukan pengukuran akurasi menggunakan aplikasi Weka dengan metode testing adalah percentage split didapat nilai akurasi sebagai berikut:

a. Neural Network tanpa optimasi

Setelah dilakukan pengolahan data mining pada aplikasi Weka dengan algoritma neural network tanpa dilakukan optimasi didapat nilai akurasi sebesar 88.53% yang ditunjukkan oleh Gambar 9.

```

=== Summary ===

Correctly Classified Instances      12008          88.535 %
Incorrectly Classified Instances    1555          11.465 %
Kappa statistic                    0.4631
Mean absolute error                0.125
Root mean squared error            0.2999
Relative absolute error            60.1772 %
Root relative squared error        92.4668 %
Total Number of Instances          13563

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0.933   0.462   0.937     0.933   0.935     0.463   0.882   0.980   no
          0.538   0.067   0.519     0.538   0.528     0.463   0.882   0.524   yes
Weighted Avg.   0.885   0.415   0.887     0.885   0.886     0.463   0.882   0.925

=== Confusion Matrix ===

      a    b  <-- classified as
11137  806 |  a = no
   749  871 |  b = yes
    
```

Gambar 9. Hasil akurasi neural network tanpa optimasi

Kita dapat melihat dari Gambar 9 di atas bahwa jumlah instances untuk dijadikan sebagai data test sebanyak 13.563 instances. Confusion matrix yang diperoleh jika dibuatkan tabel bisa terlihat pada Tabel 13.

Tabel 13. Confusion matrix neural network

	Prediksi (yes)	Prediksi (no)
Aktual (yes)	11137	806
Aktual (no)	749	871

Dari tabel di atas bisa diketahui bahwa:

$$TP = 11137$$

$$FN = 806$$

$$FP = 749$$

$$TN = 871$$

$$\text{Nilai akurasi} = (TP+TN)/(TP+TN+FP+FN)$$

$$= (11137+871)/(11137+871+749+806)$$

$$= 12008/13563 = 0.88535$$

Jadi nilai akurasi yang diperoleh sebesar 88.53%.

b. Neural Network dengan ditambahkan optimasi adaboost

Jumlah iterasi pada optimasi adaboost bisa mempengaruhi nilai akurasi yang dihasilkannya. Untuk itulah dilakukan beberapa kali percobaan optimasi dengan beberapa jumlah iterasi sebagai berikut:

Tabel 14. Nilai akurasi neural network dengan adaboost periterasinya

Jumlah Iterasi	Nilai Akurasi
1	88.53%
5	88.53%
10	88.61%
20	88.61%

Setelah dilakukan pengolahan data mining pada aplikasi Weka dengan algoritma neural network dengan optimasi adaboost yang nilainya stabil pada nilai 88.61% dimulai pada iterasi ke 10 dan tidak berubah pada iterasi berikutnya. Hasil nilai akurasi dapat terlihat pada Gambar 10.

```

=== Summary ===

Correctly Classified Instances      12018      88.6087 %
Incorrectly Classified Instances    1545      11.3913 %
Kappa statistic                    0.4643
Mean absolute error                 0.1328
Root mean squared error             0.3007
Relative absolute error             63.9397 %
Root relative squared error         92.7211 %
Total Number of Instances          13563

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
Weighted Avg.   0.934   0.066   0.523     0.536   0.529     0.464   0.828    0.418   yes
                0.886   0.417   0.887     0.886   0.887     0.464   0.828    0.895

=== Confusion Matrix ===
      a    b  <-- classified as
11150  793 |    a = no
   752  868 |    b = yes
    
```

Gambar 10. Hasil akurasi neural network dengan adaboost

Kita dapat melihat dari Gambar 10 bahwa jumlah instances untuk dijadikan sebagai data test sebanyak 13.563 instances. Confusion matrix yang diperoleh jika dibuatkan tabel bisa terlihat pada Tabel 15.

Tabel 15. Confusion matrix neural network dengan adaboost

	Prediksi (yes)	Prediksi (no)
Aktual (yes)	11150	793
Aktual (no)	752	868

Dari Tabel 15 bisa diketahui bahwa:

$$TP = 11150$$

$$FN = 793$$

$$FP = 752$$

$$TN = 868$$

$$\text{Nilai akurasi} = (TP+TN)/(TP+TN+FP+FN)$$

$$= (11150+868)/(11150+868+752+793)$$

$$= 12018/13563 = 0.886087149$$

Jadi nilai akurasi yang diperoleh sebesar 88.61%.

c. Neural Network tanpa ditambahkan optimasi bagging

Jumlah iterasi pada optimasi bagging bisa mempengaruhi nilai akurasi yang dihasilkannya.

Untuk itulah dilakukan beberapa percobaan optimasi dengan beberapa jumlah iterasi yang ditunjukkan pada Tabel 16.

Tabel 16. Nilai akurasi neural network dengan bagging periterasinya

Jumlah Iterasi	Nilai Akurasi
1	88.93%
5	90.05%
10	90.17%
20	90.27%
30	90.23%
40	90.34%
50	90.27%

60	90.39%
70	90.38%
80	90.33%
90	90.37%
100	90.39%
110	90.38%
150	90.38%

Setelah dilakukan pengolahan data mining pada aplikasi Weka dengan algoritma neural network dengan optimasi bagging didapat nilai akurasi yang stabil sebesar 90.38% dimulai dari iterasi ke 110. Hasil nilai akurasi dapat terlihat pada Gambar 11.

```

=== Summary ===

Correctly Classified Instances      12259          90.3856 %
Incorrectly Classified Instances    1304           9.6144 %
Kappa statistic                    0.466
Mean absolute error                 0.1167
Root mean squared error             0.263
Relative absolute error             56.1812 %
Root relative squared error         81.1015 %
Total Number of Instances          13563

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MOC      ROC Area  PRC Area  Class
                0.968   0.569   0.926     0.968   0.947     0.478   0.919   0.988   no
                0.431   0.032   0.646     0.431   0.517     0.478   0.919   0.599   yes
Weighted Avg.   0.904   0.505   0.893     0.904   0.895     0.478   0.919   0.941

=== Confusion Matrix ===

  a    b  <-- classified as
11561  382 |  a = no
  922   698 |  b = yes
    
```

Gambar 11. Hasil akurasi neural network dengan bagging

Kita dapat melihat bahwa jumlah instances untuk dijadikan sebagai data test sebanyak 13.563 instances. Confusion matrix yang diperoleh jika dibuatkan tabel bisa terlihat pada tabel berikut:

Tabel 17. Confusion matrix neural network dengan bagging

	Prediksi (yes)	Prediksi (no)
Aktual (yes)	11561	382
Aktual (no)	922	698

Dari Tabel 17 bisa diketahui bahwa:

TP = 11561

FN = 382

FP = 922

TN = 698

Nilai akurasi = $(TP+TN)/(TP+TN+FP+FN)$

$$= (11561+698)/(11561+698+922+382)$$

$$= 12957/13563 = 0.90385$$

Jadi nilai akurasi yang diperoleh sebesar 90.38%.

4. Kesimpulan

Dari hasil penelitian yang membandingkan antara optimasi adaboost dan bagging ini pada algoritma support vector machine, decision trees, dan neural network pada dataset bank marketing dapat disimpulkan beberapa hal sesuai tujuan dari penelitian ini, di antaranya:

- Jumlah iterasi yang diberikan pada parameter aplikasi Weka mampu mempengaruhi nilai dari akurasi dari optimasi menggunakan adaboost dan bagging baik pada algoritma support vector machine, decision trees, maupun neural network.
- Pada algoritma support vector machine optimasi adaboost mampu meningkatkan nilai akurasi lebih besar dari pada optimasi bagging.
- Pada algoritma decision trees dan neural network, optimasi bagging mampu

meningkatkan nilai akurasi lebih besar dari pada optimasi adaboost.

5. Saran

Pada penelitian ini dapat diketahui kinerja algoritma adaboost dan bagging dalam mengoptimalkan nilai akurasi dari algoritma klasifikasi support vector machine, decision trees, dan neural network. Hasil penelitian ini mungkin akan mendorong penelitian lanjutan dalam pengolahan data mining, khususnya optimasi dari algoritma klasifikasi pada dataset perbankan. Beberapa saran terkait hasil penelitian ini dan penelitian selanjutnya di antaranya:

- a. Penelitian ini bisa menjadi referensi untuk melakukan klasifikasi data bank dalam menentukan resiko kredit nasabah, yaitu dengan melihat hasil akurasi tertinggi dari penelitian ini.
- b. Penelitian ini berfokus pada perbandingan hasil akurasi sebagai pembanding kinerja dari optimasinya, dimungkinkan penelitian selanjutnya bisa membandingkan waktu proses atau kinerja lainnya dari hasil optimasi yang telah dilakukan.
- c. Dapat dilakukan penelitian lanjutan untuk optimasi adaboost pada algoritma decision trees sehingga mendapatkan nilai akurasi yang stabil.

Referensi

- Amalia, H. (2016). Komparasi Metode Data Mining Untuk Prediksi Pemasaran Bank Secara Langsung. *Seminar Nasional Ilmu Pengetahuan dan Teknologi Komputer Nusa Mandiri*, 1-6.
- Apampa, O. (2016). Evaluation Of Classification And Ensemble Algorithms For Bank Customer Marketing Response Prediction. *Journal of International Technology and Information Management*, 85-100.
- Bisri, A., & Wahono, R. S. (2015). Penerapan Adaboost Untuk Penyelesaian Ketidakeimbangan Kelas Pada Penentuan Kelulusan Mahasiswa Dengan Metode Decision Tree. *Journal Of Intelligent Systems*, 27-32.
- Dušanka, D., Darko, S., Srdjan, S., Marko, A., & Teodora, L. (2017). A Comparison of Contemporary Data Mining Tools. *XVII International Scientific Conference on Industrial Systems*, 150-155.
- Fernanda, J. W., & Otok, B. W. (2012). Boosting Neural Network Dan Boosting Cart Pada Klasifikasi Diabetes Militus Tipe II. *Jurnal Matematika*, 33-49.
- Ilham, A. (2017). Komparasi Algoritma Klasifikasi Dengan Pendekatan Level Data Untuk Menangani Data Kelas Tidak Seimbang. *Jurnal Ilmiah Ilmu Komputer*, 1-6.
- Keles, A., & Keles, A. (2015). IBMMS Decision Support Tool For Management Of Bank Telemarketing Campaigns. *International Journal of Database Management Systems*, 1-15.
- Kotsiantis, S. B., & Pintelas, P. E. (2004). Combining Bagging and Boosting. *International Journal Of Computational Intelligence*, 324-333.
- Kurniawan, D., & Supriyanto, C. (2013). Optimasi Algoritma Support Vector Machine (Svm) Menggunakan Adaboost Untuk Penilaian Risiko Kredit. *Jurnal Teknologi Informasi*, 38-49.
- Li, X., Wang, L., & Sung, E. (2008). AdaBoost with SVM-based component classifiers. *Engineering Applications of Artificial Intelligence*, 785-795.
- Listiana, E., & Muslim, M. A. (2017). Penerapan Adaboost Untuk Klasifikasi Support Vector Machine Guna Meningkatkan Akurasi Pada Diagnosa Chronic Kidney Disease. *Prosiding Snatif Ke -4* (pp. 875-881). Kudus: Universitas Muria Kudus.
- Nurdiyah, D., & Muwakhid, I. A. (2016). Perbandingan Support Vector Machine Dan K-Nearest Neighbor Untuk Klasifikasi Telur Fertile Dan Infertile Berdasarkan Analisis Texture GLCM. *Jurnal Transformatika*, 29-34.
- Prasetyo, R. T., & Pratiwi. (2015). Penerapan Teknik Bagging Pada Algoritma Klasifikasi Untuk Mengatasi Ketidakeimbangan Kelas Dataset Medis. *Jurnal Informatika*, 395-403.
- Probo, R. D., Irawan, B., & M, R. R. (2016). Analisis Dan Implementasi Perbandingan Algoritma KNN (K Nearest Neighbor) Dengan SVM (Support Vector Machine) Untuk Prediksi Penawaran Produk. *E-Proceeding Of Engineering* (pp. 4988-4995). Universitas Telkom.
- Sani, K., Winarno, W. W., & Fauziati, S. (2016). Analisis Perbandingan Algoritma Classification Untuk Authentication Uang Kertas. *Jurnal Informatika*, 1130-1139.
- Sartika, D., & Senses, D. I. (2017). Perbandingan Algoritma Klasifikasi Naive Bayes, Nearest Neighbour, dan Decision Tree pada Studi Kasus Pengambilan Keputusan Pemilihan Pola Pakaian. *JATISI*, 151-161.
- Sharma, N., Kaur, A., Gandotra, S., & Sharma, B. (2015). Evaluation and Comparison of Data Mining Techniques Over Bank Direct Marketing. *International Journal of Innovative Research in Science, Engineering and Technology*, 7141-7147.

- Suwondo, A., Asmarajati, D., & Surahman, H. (2013). Algoritma C4.5 Berbasis Adaboost Untuk Prediksi Penyakit Jantung Korone. *Prosiding Seminar Nasional Teknologi Dan Teknopreneur (SNTT)* (pp. 1-11). Wonosobo: Fastikom UNSIQ Wonosobo.
- Verbeke, W., Dejaeger, K., Martens, D., Hur, J., & Baesens, B. (2012). New Insights into Churn Prediction in the Telecommunication Sector: A Profit Driven Data Mining Approach. *European Journal of Operational Research*, 218(1), 211-229. doi:10.1016/j.ejor.2011.09.031
- Wahbeh, A. H., Al-Radaideh, Q. A., Al-Kabi, M. N., & Al-Shawakfa, E. M. (2008). A Comparison Study between Data Mining Tools over some Classification Methods. *International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence*, 18-26.
- Wimmer, H., & Powell, L. M. (2015). A Comparison of Open Source Tools for Data Science. *Proceedings of the Conference on Information Systems Applied Research* (pp. 1-9). Wilmington: Information Systems & Computing Academic Professionals.
- Wisaeng, K. (2013). A Comparison of Different Classification Techniques for Bank Direct Marketing. *International Journal of Soft Computing and Engineering*, 116-119.
- Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3rd ed.). Burlington: Morgan Kaufmann.
- Yap, B. W., Rani, K. A., Rahman, H. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets. *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*. 285, pp. 13-22. Singapore: Springer. doi:10.1007/978-981-4585-18-7_2