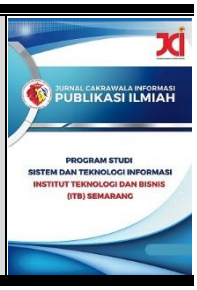
e-ISSN : 2830-6252
p-ISSN : 2830-6562

JCI, Vol. 2 No. 2 (2022) 1 – 14 | *https://doi.org/10.54066/jci.v2i2.233*

luceant opera vestra bona

Jurnal Cakrawala Informasi

Journal Homepage: http://www.itbsemarang.ac.id/sijies/index.php/jci
e-Mail: jci@itbsemarang.ac.id



Analisis dan Komparasi Algoritma Klasifikasi untuk Prediksi Kerugian *Tower Provider* Akibat Penalti yang Diberikan oleh Operator Telekomunikasi karena Keterlambatan Penyelesaian Pekerjaan oleh *Tower Provider*

Siska Narulita [1*]
Prihati [2]
Aji Priyambodo [3]

[1, 2, 3] Sistem dan Teknologi Informasi, Institut Teknologi dan Bisnis Semarang

INFO ARTIKEL

Histori artikel:

Diterima	:	13 November 2022
Revisi	:	16 Desember 2022
Disetujui	:	26 Desember 2022
Publikasi	:	30 Desember 2022

Kata kunci:
Algoritma Klasifikasi
Decision Tree
Naive Bayes
k-NN
Logistic Regression
Neural Network
Tower Provider
Penalti Operator Telekomunikasi

ABSTRACT

Penalty is a fine given by the telecommunications operator as the employer to the tower provider (telecommunication tower provider company). This penalty is given because the work completion time exceeds the specified time limit. To reduce the level of company losses caused by penalties from telecommunications operators, the tower provider itself must be able to take steps to prevent or avoid penalties from telecommunications operators by predicting the penalties that will be received by the tower provider. In this study used classification algorithms such as decision tree (C4.5), naive Bayes, k-nearest neighbor, logistic regression, and neural network. With these five classification algorithms, comparisons were made to obtain the level of accuracy of each classification algorithm using the 10-fold cross validation validation method and the t-test parametric difference test comparison method. From the parametric t-test test, the results of the decision tree algorithm (C4.5) are more dominant than other algorithms, next it can be said that the naive Bayes algorithm, logistic regression, and neural network have the same accuracy, however, the logistic regression algorithm and the neural network do not better than the k-nearest neighbor algorithm.

ABSTRAK

Penalti adalah denda yang diberikan oleh operator telekomunikasi selaku pemberi pekerjaan kepada *tower provider* (perusahaan penyedia menara telekomunikasi). Penalti ini diberikan karena waktu penyelesaian pekerjaan melebihi batas waktu yang ditentukan. Untuk mengurangi tingkat kerugian perusahaan yang diakibatkan oleh penalti dari operator telekomunikasi, maka *tower provider* itu sendiri harus bisa mengambil langkah untuk mencegah atau menghindari

* Korespondensi penulis: siskanarulita84@gmail.com

penalti dari operator telekomunikasi dengan cara memprediksi penalti yang akan akan diterima oleh *tower provider*. Pada penelitian ini digunakan algoritma klasifikasi seperti *decision tree* (C4.5), *naive bayes*, k-*nearest neighbor*, *logistic regression*, dan *neural network*. Dengan kelima algoritma klasifikasi ini dilakukan perbandingan untuk mendapatkan tingkat akurasi dari masing-masing algoritma klasifikasi menggunakan metode validasi 10-*fold cross validation* dan metode perbandingan uji beda parametrik t-*test*. Dari uji parametrik t-*test* diperoleh hasil algoritma *decision tree* (C4.5) lebih dominan daripada algoritma yang lain, berikutnya bisa dikatakan algoritma *naive bayes*, *logistic regression*, dan *neural network* memiliki akurasi yang sama, namun demikian algoritma *logistic regression* dan *neural network* tidak lebih baik dari algoritma k-*nearest neighbor*.

PENDAHULUAN

Di era informasi dewasa ini, peranan teknologi telekomunikasi dirasakan semakin penting, terutama bagi kehidupan masyarakat. Dalam beberapa tahun belakangan ini, perkembangan budaya, ilmu pengetahuan, dan pendidikan begitu cepat. Salah satu penyebabnya adalah dari kemajuan teknologi telekomunikasi. Karena kesadaran masyarakat akan pentingnya informasi, memicu perkembangan teknologi informasi, salah satunya di sektor telekomunikasi seluler.

Pernyataan dari Syeh Assery yang merupakan pengamat bisnis telekomunikasi yang dimuat pada harian Kedaulatan Rakyat tanggal 17 Juli 2008, bahwa dengan semakin meningkatnya sektor telekomunikasi maka memberikan peningkatan subsektor sarana pendukungnya antara lain pembangunan menara telekomunikasi [1]. Secara fungsional, menara telekomunikasi merupakan perangkat yang mendukung penyelenggaraan telekomunikasi dan salah satu sarana dan prasarana yang memungkinkan berfungsinya telekomunikasi. Sejak tahun 2002 bisnis penyewaan menara telekomunikasi mulai ada. Penyewaan menara telekomunikasi adalah usaha penyewaan yang dilakukan para penyedia, pembangun dan pengelola menara telekomunikasi untuk disewakan kepada operator telekomunikasi.

Masih menurut Syeh Assery, namun di sisi lain berbagai permasalahan yang berkaitan dengan pembangunan dan keberadaan menara telekomunikasi juga sedemikian rumit [1]. Cukup banyak menara telekomunikasi yang dianggap kurang memenuhi jaminan keamanan lingkungan dan kurang proporsional penempatannya bagi estetika tatakota. Juga sikap masyarakat terhadap keberadaan menara telekomunikasi yang dianggapnya berpotensi membahayakan lingkungan sekitar tempat tinggalnya. Bahkan indikasi timbulnya persaingan pendirian menara telekomunikasi yang tidak efisien karena tidak saling berbagi atau *sharing tower* [1].

Dari permasalahan yang telah diuraikan oleh Syeh Assery yang berkaitan dengan pembangunan dan keberadaan menara telekomunikasi, maka muncul SKB 3 Menteri tahun 2009 tentang Pedoman Pembangunan dan Penggunaan Bersama Menara Telekomunikasi [2]. Di dalam SKB 3 Menteri bab IV mengenai pembangunan dan pengelolaan menara pada pasal 5 ayat (1) disebutkan bahwa menara disediakan oleh penyedia menara, yang dalam hal ini adalah *tower provider* [2]. Seiring dengan hal itu, bermunculan perusahaan-perusahaan penyedia menara telekomunikasi atau *tower provider*.

Tower provider saling bersaing. Operator telekomunikasi memberikan pekerjaan kepada *tower provider* tersebut untuk membangun sebuah *tower* atau hanya sekedar pekerjaan kolokasi

(*sharing* ke *tower provider* atau operator telekomunikasi lain). Kinerja dari perusahaan *tower provider* pun tidak lepas dari penilaian operator telekomunikasi. Semakin baik dan cepat kinerja *tower provider* dalam menyelesaikan pekerjaannya, maka akan diperhitungkan untuk mendapatkan pekerjaan lagi dari operator telekomunikasi tersebut. Oleh karena itu, tiap *tower provider* harus meningkatkan kinerjanya.

Ada banyak faktor yang mempengaruhi kinerja *tower provider*, misalnya faktor pemilihan *vendor* yang kompeten dan berkualitas, ketersediaan dana untuk melaksanakan pekerjaan, ada tidaknya *community issue* di lapangan, dan lain sebagainya. Jika pekerjaan yang dilakukan oleh *tower provider* tidak selesai sesuai dengan target yang diberikan oleh operator telekomunikasi, maka operator akan memberlakukan denda atau penalti. Tiap-tiap operator memberikan denda atau penalti yang berbeda-beda. Misal denda bisa berupa pemotongan nilai PO (*purchase order*), pembebasan masa sewa *tower* untuk beberapa bulan dan lain sebagainya. Semakin banyak pekerjaan yang tidak sesuai dengan target, akan semakin banyak pula penalti yang diberikan oleh operator telekomunikasi. Tentu saja perusahaan *tower provider* akan merugi dan ke depannya bisa tidak mendapatkan pekerjaan atau *order* lagi dari operator telekomunikasi. Oleh karena itu, *tower provider* harus bisa memprediksi kerugian yang akan diperoleh perusahaan karena penalti yang dikenakan oleh operator. Sehingga perusahaan dapat mengambil langkah-langkah untuk meningkatkan kinerja perusahaan.

Sudah banyak algoritma klasifikasi yang dipakai dalam analisis komparasi namun ada beberapa algoritma klasifikasi yang populer digunakan dalam analisis komparasi seperti algoritma *decision tree* (C4.5), *logistic regression*, *linear and quadratic discriminant analysis*, *neural network*, *least square support vector machines*, k-NN, *random forest*, CART, dan *naive bayes*. Dalam penelitian ini dilakukan perbandingan terhadap lima algoritma klasifikasi yaitu algoritma *decision tree* (C4.5), *naive bayes*, k-*nearest neighbor*, *logistic regression*, dan *neural network* menggunakan metode validasi 10-*fold cross validation* untuk *training* dan *testing dataset*, serta metode perbandingan uji beda parametrik t-*test* untuk membandingkan akurasi algoritma klasifikasi sehingga diperoleh algoritma yang memiliki akurasi terbaik untuk memprediksi kerugian *tower provider* akibat penalti dari operator telekomunikasi yang dikarenakan keterlambatan penyelesaian pekerjaan oleh *tower provider* itu sendiri.

TINJAUAN PUSTAKA

Analisis mengenai komparasi atau perbandingan algoritma klasifikasi sebelumnya telah dilakukan oleh beberapa peneliti, antara lain Lan Yu, *et al* (2011) yang melakukan perbandingan algoritma klasifikasi untuk pengendalian resiko kredit [3]. Algoritma klasifikasi yang digunakan dalam penelitiannya yaitu *logistic regression*, *discriminant analysis*, k-*nearest neighbor*, *naive bayes*, TAN *bayes*, *decision tree*, *associative classification* (CBA), *artificial neural network*, dan *support vector machines* (SVM). ROC *curve* dan metode *delong-pearson* digunakan untuk membandingkan performansi sembilan algoritma klasifikasi tersebut. Hasil yang diperoleh bahwa algoritma *decision tree* dan *support vector machines* memiliki performansi yang tidak memuaskan. Algoritma *associative classification*

menjadi pilihan terbaik untuk memprediksi peringkat kredit (*credit rating*).

Penelitian lainnya dilakukan oleh Romi Satria Wahono, *et al* (2014) yang melakukan penelitian perbandingan algoritma klasifikasi untuk prediksi cacat *software* [4]. Penelitian dilakukan menggunakan sepuluh algoritma klasifikasi yang dibedakan menjadi algoritma statistik tradisional (LR, LDA, dan NB), *nearest neighbor* (k-NN dan K*), NN, SVM, dan *decision tree* (C4.5, CART, dan RF). Dalam penelitian ini digunakan motode validasi *stratified* 10-*fold cross-validation* untuk *training* dan *testing dataset* serta menggunakan metode perbandingan non parametrik *friedman* dan *nemenyi post hoc test* untuk menguji perbedaan signifikan dari AUC diantara sepuluh algoritma klasifikasi tersebut. Hasil dari penelitian menunjukkan bahwa algoritma *logistic regression* (LR) mempunyai performansi terbaik untuk prediksi *cacat sofware*. NB, NN, SVM, dan K* juga mempunyai performansi yang baik, tidak ada perbedaan yang signifikan diantara algoritma tersebut. Kelompok algoritma *decision tree* (C4.5, CART, dan RF), cenderung memiliki performansi di bawahnya begitu juga dengan algoritma LDA dan k-NN.

Brown dan Mues (2012) melakukan komparasi atau perbandingan algoritma klasifikasi *dataset* nilai kredit untuk *class imbalanced*. Algoritma yang digunakan antara lain *logistic regression*, *linear and quadratic discriminant analysis*, *neural network*, *least square support vector machines*, *decision tree* (C4.5), k-NN, *random forest*, dan *gradient boosting* [5]. Untuk pengukuran performansi algoritma digunakan nilai dari AUC. *Friedman test* dan *nemenyi post hoc test* digunakan untuk menguji tingkat perbedaan signifikan diantara algoritma klasifikasi. Hasil penelitian menunjukkan bahwa algoritma *random forest* dan *gradient boosting* mempunyai akurasi yang baik untuk *dataset* nilai kredit dengan *class imbalanced*, sedangkan algoritma *decision tree* (C4.5), *quadratic discriminant analysis*, dan k-NN mempunyai akurasi yang buruk untuk *dataset* dengan *class imbalanced* yang besar.

Data Mining

Data mining merupakan proses untuk menemukan pola (*pattern*) dari suatu data. Pola (*pattern*) yang ditemukan harus memiliki arti atau mengandung informasi penting [6]. *Data mining* adalah proses analisis dari *dataset* (berukuran besar) untuk menemukan suatu hubungan tak terduga serta meringkas data tersebut dengan cara baru yang dapat dimengerti dan bermanfaat bagi pemilik data [7].

Klasifikasi

Klasifikasi adalah proses menempatkan objek tertentu dalam satu set kategori, berdasarkan sifat masing-masing objek [8]. Proses klasifikasi didasarkan pada empat komponen [9]:

1. *Class*

   Class adalah variabel tidak bebas (*dependent*) yang merupakan variabel kategorik yang merepresentasikan label objek setelah diklasifikasi.
2. Prediktor

   Prediktor merupakan variabel bebas (*independent*) yang direpresentasikan oleh karakteristik data (atribut) yang menjadi dasar dari klasifikasi.
3. *Dataset training*

   Dataset training merupakan kumpulan data yang berisi komponen *class* (label) dan prediktor (atribut) yang digunakan untuk

training model untuk mengenali *class* berdasarkan variabel prediktor (atribut) yang tersedia.

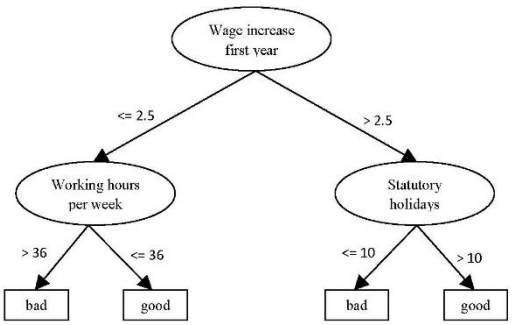
4. *Dataset testing*

   Dataset testing berisi data baru yang akan diklasifikasikan menggunakan model yang telah dibangun atau ditemukan, dengan demikian akurasi dari model klasifikasi (performansi model) tersebut dapat dievaluasi.

Algoritma *Decision Tree* (C4.5)

Algoritma *decision tree* (C4.5) adalah algoritma klasifikasi yang memiliki struktur seperti pohon terdiri atas *node* internal (akar) dan *node* cabang. *Node* internal (akar) merupakan atribut dan sebuah *node* cabang mewakili *class* [10].

Algoritma *decision tree* (C4.5) merupakan algoritma klasifikasi yang mengklasifikasikan sampel data secara *top-down* mulai dari simpul akar dan bergerak sesuai dengan hasil pengujian dari *node* internal sampai *node* cabang dicapai dan *class* ditetapkan [11].



Gambar 1. *Decision Tree* Data Negosiasi Tenaga Kerja

Tahapan-tahapan algoritma *decision tree* (C4.5):

1. Memilih atribut sebagai akar yang didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung gain digunakan rumus:

$$\text{Gain (S, A)} = Entropy(S) - \sum_{i-1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i)$$

S = himpunan kasus

A = atribut

n = jumlah atribut A

|Si| = jumlah kasus pada partisi ke-i

|S| = jumlah kasus dalam S

Nilai entropi dapat dihitung dengan persamaan:

$$Entropy(S) = \sum_{i=1}^{n} - p_i * \log_2 p_i$$

S = himpunan (*dataset*) kasus

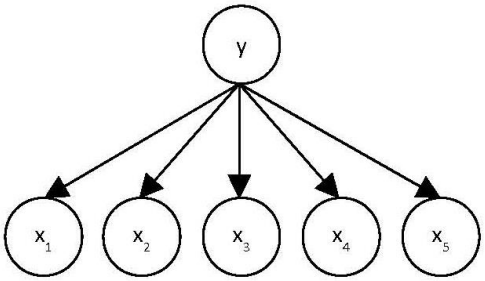
n = banyaknya partisi

pi = probabilitas yang didapat dari jumlah kasus pada partisi ke-i dibagi total kasus

2. Membuat cabang untuk setiap nilai didalam akar tersebut,
3. Membagi atribut dalam cabang,
4. Ulangi proses untuk setiap atribut sampai semua atribut pada cabang memiliki *class* yang sama.

Algoritma *Naive Bayes* (NB)

Algoritma *naive bayes* adalah salah satu bentuk sederhana dari jaringan *Bayesian* untuk klasifikasi. Jaringan *Bayes* dapat dilihat sebagai bagan *acyclic* yang diarahkan dengan gabungan distribusi probabilitas melalui rangkaian diskrit dan variabel stokastik [12].



Gambar 2. Hubungan Variabel pada *Naive Bayes*

Tahapan-tahapan algoritma *naive bayes* (NB):

1. Menghitung P(B), jumlah kemungkinan dari semua data,
2. Menentukan atribut yang memiliki probabilitas yang sama P(A|B). Nilai P(A|B) dapat dihitung menggunakan rumus:

$$P(A \mid B) = \frac{P(B \mid A).P(A)}{P(B)}$$

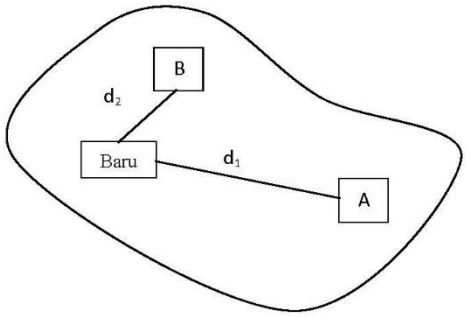
B = label/*class* pada sebuah tabel
A = atribut/prediksi
P(B) = probabilitas atau kemungkinan dari suatu *class*
P(B|A) = probabilitas dari *class* B yang memberikan ketentuan terhadap atribut A
P(A|B) = ketentuan dari atribut A yang memberikan probabilitas pada *class* B

3. Menghitung hasil dengan probabilitas data sesuai hipotesis,
4. Mencari nilai maksimum dari hasil hitung.

Algoritma k-*Nearest Neighbor* (k-NN)

Algoritma k-*nearest neighbor* adalah algoritma klasifikasi yang mengklasifikasikan titik data dengan mengambil suara terbanyak yang paling mirip dengan k titik data [13].

Algoritma k-*nearest neighbor* (k-NN) merupakan algoritma yang bertujuan untuk mengklasifikasi objek baru berdasarkan atribut dan *training samples* (Larose, 2005) [14]. Dimana hasil dari sampel uji yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada k-NN. Algoritma k-NN menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari sampel uji yang baru [15].



Gambar 3. Ilustrasi Kedekatan Kasus

Tahapan-tahapan algoritma k-*nearest neighbor* (k-NN):

1. Tentukan parameter K (jumlah tetangga terdekat),
2. Hitung jarak antara data baru dengan semua *data training*. Jarak antara data baru dengan semua *data training* (jarak *euclidian*) dapat diperoleh dengan menggunakan rumus:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i y_i)^2}$$

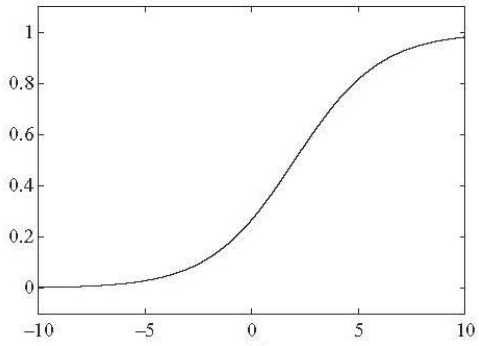
d = jarak *euclidian* (jarak antara titik pada *data training* x dan titik *data testing* y yang akan diklasifikasi
x1, x2, ..., xi = nilai atribut
y1, y2, ..., yi = nilai atribut
i = nilai atribut
n = dimensi atribut

3. Urutkan jarak tersebut dan tetapkan tetangga terdekat berdasarkan jarak minimum ke-K,
4. Periksa *class* dari tetangga terdekat,

5. Gunakan mayoritas sederhana dari *class* tetangga terdekat sebagai nilai prediksi data baru.

Algoritma *Logistic Regression* (LR)

Algoritma *logistic regression* adalah algoritma klasifikasi yang menerapkan model regresi linear berganda (*multiple*) dimana variabel *class* (y) bernilai diskrit [10].



Gambar 4. Contoh Fungsi *Logistic Regression*

Tahapan-tahapan algoritma *logistic regression*:

1. Memetakan dataset,
2. Menduga koefisien β,
3. Menghitung nilai:

$$g_0(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

4. Menghitung nilai: $e^{g_0(x)}$
5. Menghitung nilai:

$$p_1(x) = \frac{e^{g_0(x)}}{1 + e^{g_0(x)}}$$

6. Menghitung fungsi *likelihood*:

$$p_i^{Y_i} * (1 - p_i)^{1-Y_i}$$

7. Mencari nilai natural logaritma dari fungsi *likelihood*,
8. Memaksimumkan fungsi *likelihood* dan koefisien β,
9. Menghitung nilai:

$$p_0(x) = p(Y = 0) = \frac{e^{g_0(x)}}{e^{g_0(x)} + e^{g_1(x)}}$$

$$p_1(x) = p(Y = 1) = \frac{e^{g_1(x)}}{e^{g_0(x)} + e^{g_1(x)}}$$

dengan $g_0(x) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k$ dan $g_1(x) = 0$

x = nilai attribut

Y = nilai variabel *class*/label

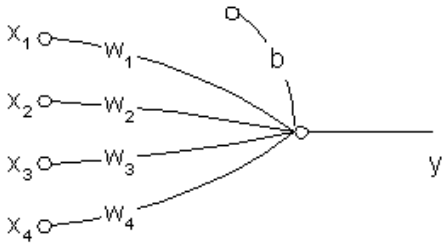
β = koefisien regresi

p0(x) = probabilitas Y = 1 berdasarkan kondisi x

p1(x) = probabilitas Y = 0 berdasarkan kondisi x

Algoritma *Neural Network* (NN)

Algoritma *neural network* adalah model matematika yang merepresentasikan fungsi yang dimiliki otak manusia [16].



Gambar 5. *Neural Network* Satu Lapis

Tahapan-tahapan algoritma *neural network*:

1. Inisialisasi bobot-bobot (termasuk juga bias), termasuk perubahan bobot awal,
2. Mengambil nilai x1, x2, x3 dan x4 juga nilai target,
3. Menghitung keluaran jaringan *neuron* dengan rumus:

$$y = X * W^T + b$$

$$y = x_1 w_1 + x_2 w_2 + x_3 w_3 + x_4 w_4 + b$$

y = *output* atau keluaran jaringan *neuron*

x = *input* atau data masukan

w = bobot *link neuron*

b = bias

4. Menghitung parameter $\theta = w$,
5. Menghitung *error* keluaran $e = \text{target} - y$,
6. Menyimpan bobot-bobot ke dalam variabel bobot lama,
7. Menghitung perubahan bobot-bobot pada lapisan keluaran menggunakan rumus:

$$\Delta w_i = \eta e x_i \text{ , } \Delta \eta e$$
$$w_i(baru) = w_i(lama) + \Delta w_i(baru) + \alpha \Delta w_i(lama)$$
$$b(baru) = b(lama) + \Delta b(baru) + \alpha \Delta b(lama)$$

8. Menyimpan perubahan-perubahan bobot dan bias ke variabel perubahan lama,
9. Kembali langkah 2.

Metode Validasi k-*Fold Cross Validation*

Dalam k-*fold cross validation*, *dataset* X dibagi secara *random* menjadi K bagian yang sama ukurannya, Xi, i = 1,...,K. Untuk menghasilkan masing-masing pasangan, ditentukan bagian K sebagai *data testing* (validasi) dan mengkombinasikannya dengan sisa bagian K-1 sebagai *data training* [17].

$$V_1 = X_1 \quad T_1 = X_2 \cup X_3 \cup \ldots \cup X_k$$
$$V_2 = X_2 \quad T_2 = X_1 \cup X_3 \cup \ldots \cup X_k$$
$$\vdots$$
$$V_k = X_k \quad T_k = X_1 \cup X_2 \cup \ldots \cup X_{k-1}$$

k-*fold cross validation* adalah sebuah teknik intensif komputer yang menggunakan keseluruhan data yang ada sebagai *training set* dan *test set*. Metode *cross validation* akan menghindari tumpang tindih pada *data testing*. Tahapan dalam metode *cross validation*:

1. Membagi data menjadi k bagian dengan ukuran yang sama,
2. Gunakan masing-masing bagian untuk *testing*, sisanya sebagai *training*.

Biasanya digunakan metode *stratification* sebelum proses *cross validation*. Estimasi tingkat kesuksesan dirata-rata untuk mendapat estimasi total.

Tabel 1. *Stratified 10-Fold Cross Validation*

Testing	*Dataset*									
1	■									
2		■								
3			■							
4				■						
5					■					
6						■				
7							■			
8								■		
9									■	
10										■

Uji Beda Parametrik

Uji beda parametrik yaitu metode pengujian yang mempertimbangkan jenis sebaran atau distribusi data, yaitu apakah data menyebar secara normal atau tidak. Dengan kata lain, data yang akan dianalisis menggunakan uji parametrik harus memenuhi asumsi normalitas. Pada umumnya, jika data tidak menyebar normal, maka data seharusnya dikerjakan dengan metode non parametrik, atau setidak-tidaknya dilakukan transformasi terlebih dahulu agar data mengikuti sebaran normal, sehingga bisa dikerjakan dengan statistik parametrik.

Uji parametrik membutuhkan input dari user, seperti jumlah interval jumlah maksimum [18]. Ciri-ciri uji parametrik:

1. Data dengan skala interval dan rasio,
2. Data menyebar atau berdistribusi normal.

Keunggulan uji parametrik:

1. Syarat-syarat parameter dari suatu populasi yang menjadi sampel biasanya tidak diuji dan

dianggap memenuhi syarat, pengukuran terhadap data dilakukan dengan kuat,

2. Observasi bebas satu sama lain dan ditarik dari populasi yang berdistribusi normal serta memiliki varian yang homogen.

Kelemahan uji parametrik:

1. Populasi harus memiliki varian yang sama,
2. Variabel-variabel yang diteliti harus diukur setidaknya dalam skala interval,
3. Dalam analisis varian ditambahkan persyaratan rata-rata populasi harus normal dan bervarian sama dan harus merupakan kombinasi linear dari efek-efek yang ditimbulkan.

Tabel 2. Contoh Uji Beda Parametrik

Jenis data	Metode
2 sampel independen	*Independent Sample T-Test*
2 sampel berhubungan	*Paired Sample T-Test*
2 < sampel	*Anova*

Dalam pengklasifikasian keakuratan dari tes diagnostik menggunakan *Area Under Curve* (AUC), sebuah sistem nilai yang disajikan [9].

Tabel 3. Keterangan Nilai AUC

AUC	Keterangan
0.90 – 1.00	*excellent classification*
0.80 – 0.90	*good classification*
0.70 – 0.80	*fair classification*
0.60 – 0.70	*poor classification*
< 0.60	*failure*

T-*Test*

T-*test* dengan dua sampel digunakan untuk menentukan apakah rata-rata dua populasi adalah sama [19]. T-*test* dua sampel untuk data berpasangan didefinisikan:

H_0 = $\mu_1 = \mu_2$

H_a = $\mu_1 \neq \mu_2$

Test statistik = $$T = \frac{\overline{Y}_1 \overline{Y}_2}{\sqrt{s_1^2 / N_1 + s_2^2 / N_2}}$$

N1 = ukuran sampel

N2 = ukuran sampel

$\overline{Y}_1, \overline{Y}_2$ = rata-rata sampel

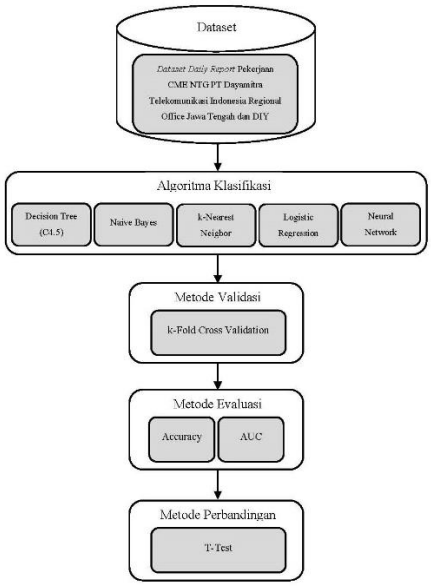
s_1^2, s_2^2 = varian sampel

jika varian sama maka diasumsikan:

$$T = \frac{\overline{Y}_1 \overline{Y}_2}{s_p \sqrt{1/N_1 + 1/N_2}}, \text{ dimana } s_p^2 = \frac{(N_1 - 1)s_1^2 + (N_2 - 1)s_2^2}{N_1 + N_2 - 2}$$

METODE PENELITIAN

Metode penelitian yang diusulkan dapat dilihat pada Gambar 6. Strukturnya terdiri dari 1) *dataset* 2) algoritma klasifikasi 3) metode validasi 4) metode evaluasi, dan 5) metode perbandingan.



Gambar 6. Struktur Metode yang Diusulkan

Dataset

Dataset yang digunakan dalam penelitian ini merupakan *dataset private* yang diperoleh dari salah satu *tower provider* nasional di Jawa Tengah dan DIY. *Dataset* berupa data *daily report*

pekerjaan *Civil, Mechanical, and Electrical* (CME). Struktur datanya sebagai berikut:

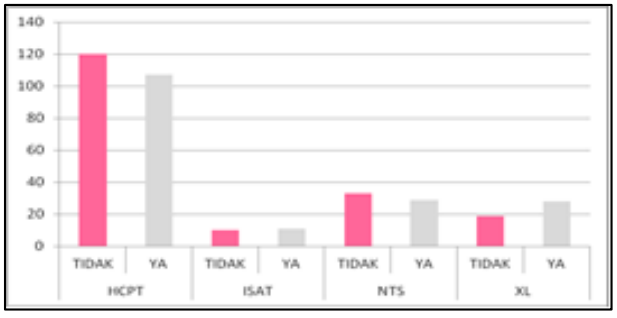
Tabel 4. Struktur *Dataset Daily Report* Pekerjaan CME

No.	Attribut	Tipe Data	Keterangan
1	Operator/Tenant	Polynominal	*Attribute*
2	Status Site	Binominal	*Attribute*
3	Tenant ID	Nominal	*Attribute*
4	Tenant Name	Nominal	*Attribute*
5	Tower Owner	Polynominal	*Attribute*
6	Outdoor/Indoor	Binominal	*Attribute*
7	Tower Type	Binominal	*Attribute*
8	Tower Height	Integer	*Attribute*
9	Site Type	Polynominal	*Attribute*
10	RFC Date	Date	*Attribute*
11	CME Start Date	Date	*Attribute*
12	Foundation Complete Date	Date	*Attribute*
13	Soft RFI Date	Date	*Attribute*
14	RFI Actual Date	Date	*Attribute*
15	Penalti	Binominal	*Label*

Penelitian ini mengambil sampel sebanyak 357 *record data* dengan jumlah atribut empat belas dan satu label.

Tabel 5. Perbandingan Jumlah Penalti Operator

Operator	Status Site	Penalti		Grand Total
		TIDAK	YA	
HCPT	B2S	23	12	35
	Colo	97	95	192
ISAT	Colo	10	11	21
NTS	Colo	33	29	62
XL	Colo	19	28	47
Grand Total		**182**	**175**	**357**



Gambar 7. Grafik Jumlah Penalti Operator

Dari tabel 5 diketahui bahwa dari 35 *project site* B2S (*Build to Suite*) yang diberikan oleh operator HCPT yang terkena penalti 12 *site*, sedangkan dari *project colo* 192 *site*, yang terkena penalti sebanyak 95 *site*. *Colo* operator Indosat dengan total *project* 21 *site*, yang terkena penalti sebesar 11 *site*. Dari total 62 *project colo site* dari operator NTS/Axis, 29 *site* terkena penalti. Dan *project colo* XL sebanyak 47 *site*, yang terkena penalti 28 *site*.

Algoritma Klasifikasi

Penelitian ini bertujuan membandingkan akurasi lima algoritma klasifikasi untuk memprediksi kerugian *tower provider* akibat penalti dari operator telekomunikasi. Pada penelitian ini digunakan lima algoritma klasifikasi, yaitu *decision tree* (C4.5), *naive bayes*, k-*nearest neighbor*, *logistic regression*, dan *neural network*. Kelima algoritma ini dipilih karena algoritma ini populer digunakan dalam analisis komparasi.

Metode Validasi

Dalam penelitian digunakan metode validasi 10-*fold cross validation* untuk *training* dan *testing dataset*. *Dataset* dibagi menjadi sepuluh bagian, sembilan bagian digunakan untuk *data training* dan satu bagian digunakan untuk *data testing*. Proses validasi dilakukan sebanyak sepuluh kali. Peneliti menggunakan metode validasi 10-*fold cross validation* karena metode ini menjadi standar metode validasi dari penelitian yang telah dilakukan sebelumnya.

Metode Evaluasi

Peneliti menentukan nilai *accuracy* dari *confusion matrix* dan nilai *area under curve* (AUC) dari ROC *curve* sebagai indikator tingkat akurasi performansi dari algoritma klasifikasi. Istilah

accuracy sering digunakan dalam konteks metode klasifikasi. *Accuracy* mengacu pada pengukuran tingkat keakuratan atau prediksi dari suatu model atau metode klasifikasi (Sammut, 2011). Nilai *accuracy* dalam penelitian ini diperoleh dari tabel *confusion matrix RapidMiner*. AUC mengukur performansi metode klasifikasi berdasarkan ROC *curve*. Nilai AUC ditunjukkan dalam skala 0 sampai 1 dimana angka 0 menunjukkan tingkat negatif dan angka 1 menunjukkan tingkat positif (Sammut, 2011).

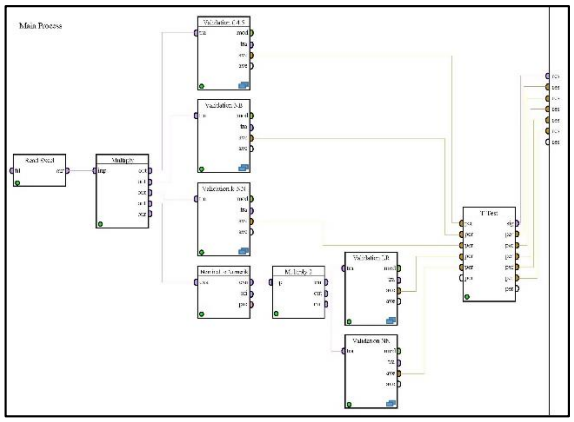
Metode Perbandingan

Peneliti menggunakan metode perbandingan uji beda parametrik t-*test* untuk membandingkan akurasi algoritma klasifikasi. Nilai akurasi yang diperoleh dibandingkan menggunakan t-*test* untuk memastikan apakah ada perbedaan signifikan pada akurasi algoritma. Jika perbedaan antara dua rata-rata akurasi tidak signifikan, dapat dikatakan bahwa akurasi algoritma tidak dapat dibedakan dan jika perbedaannya signifikan, maka salah satu algoritma memiliki akurasi yang tidak bagus dibandingkan algoritma yang lain [20].

PEMBAHASAN DAN HASIL

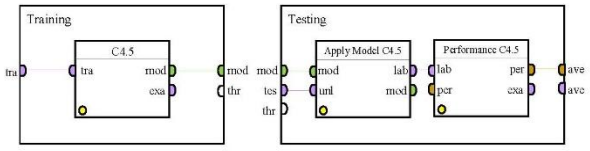
***Training* dan *Testing Dataset* Menggunakan Algoritma Klasifikasi**

Pada penelitian ini digunakan *tool RapidMiner Studio* 6.3.000 untuk mengukur performansi algoritma klasifikasi dengan *confusion matrix* dan ROC *curve*. Berikut skema proses pengukuran performansi lima algoritma klasifikasi (*decision tree* (C4.5), *naive bayes*, k-*nearest neighbor* (k-NN), *logistic regression*, dan *neural network*):

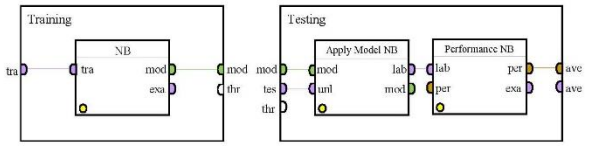


Gambar 8. Skema *Main Process* Penghitungan Performansi Algoritma Klasifikasi

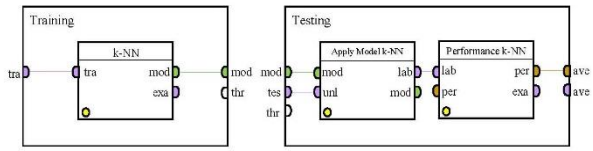
Di dalam proses validasi masing-masing algoritma klasifikasi terdapat sub proses. Berikut skema untuk masing-masing proses validasi algoritma klasifikasi:



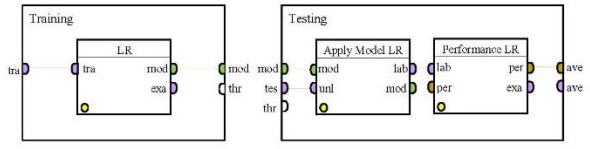
Gambar 9. Skema Sub Proses Validasi Algoritma *Decision Tree* (C4.5)



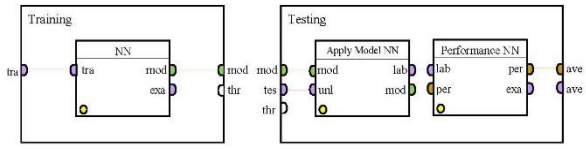
Gambar 10. Skema Sub Proses Validasi Algoritma *Naive Bayes*



Gambar 11. Skema Sub Proses Validasi Algoritma k-*Nearest Neighbor*



Gambar 12. Skema Sub Proses Validasi Algoritma Logistic Regression



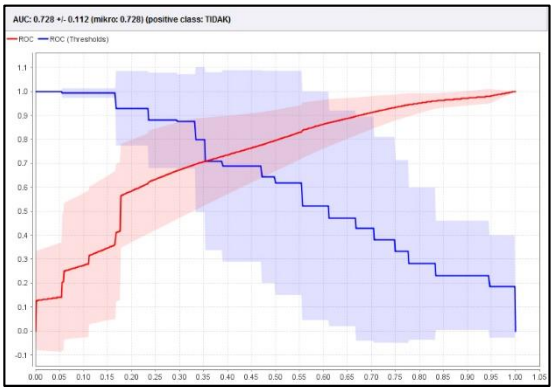
Gambar 13. Skema Sub Proses Validasi Algoritma *Neural Network*

Nilai Akurasi Algoritma Klasifikasi dengan *Confusion Matrix* dan ROC *Curve*

Dari proses validasi yang menggunakan metode k-*fold cross validation*, diperoleh nilai akurasi masing-masing algoritma klasifikasi:

Tabel 6. *Confusion Matrix Algoritma Decision Tree* (C4.5)

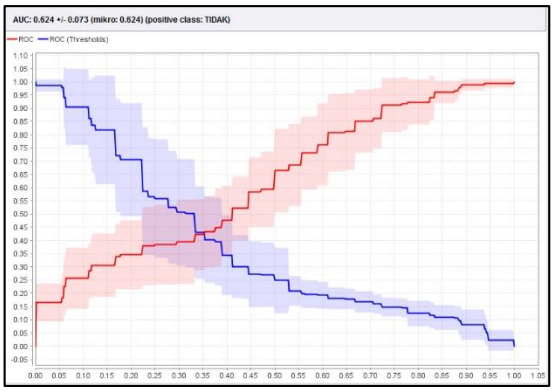
accuracy: 69.74% +/- 11.31% (mikro: 69.75%)			
	true YA	true TIDAK	class precision
pred. YA	119	52	69.59%
pred. TIDAK	56	130	69.89%
class recall	68.00%	71.43%	



Gambar 14. ROC *Curve* Algoritma *Decision Tree* (C4.5)

Tabel 7. *Confusion Matrix* Algoritma *Naive Bayes*

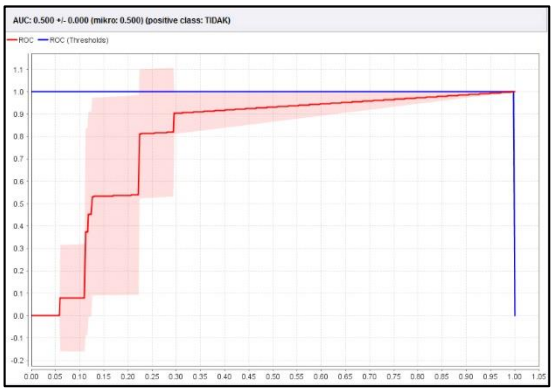
accuracy: 56.33% +/- 8.89% (mikro: 56.30%)			
	true YA	true TIDAK	class precision
pred. YA	133	114	53.85%
pred. TIDAK	42	68	61.82%
class recall	76.00%	37.36%	



Gambar 15. ROC *Curve* Algoritma *Naive Bayes*

Tabel 8. *Confusion Matrix* Algoritma k-*Nearest Neighbor*

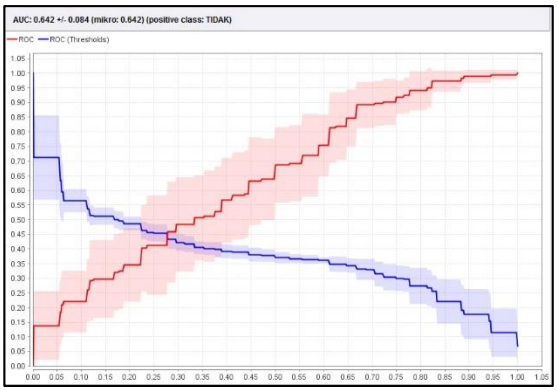
accuracy: 86.23% +/- 6.40% (mikro: 86.27%)			
	true YA	true TIDAK	class precision
pred. YA	147	21	87.50%
pred. TIDAK	28	161	85.19%
class recall	84.00%	88.46%	



Gambar 16. ROC *Curve* Algoritma k-*Nearest Neighbor*

Tabel 9. *Confusion Matrix* Algoritma *Logistic Regression*

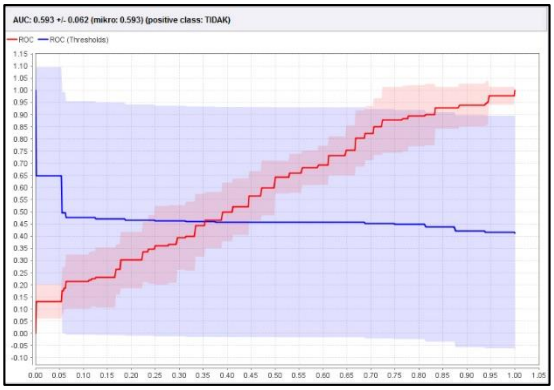
accuracy: 55.76% +/- 5.31% (mikro: 55.74%)			
	true YA	true TIDAK	class precision
pred. YA	151	134	52.98%
pred. TIDAK	24	48	66.67%
class recall	86.29%	26.37%	



Gambar 17. ROC *Curve* Algoritma *Logistic Regression*

Tabel 10. *Confusion Matrix* Algoritma *Neural Network*

accuracy: 51.84% +/- 4.61% (mikro: 51.82%)			
	true YA	true TIDAK	class precision
pred. YA	93	90	50.82%
pred. TIDAK	82	92	52.87%
class recall	53.14%	50.55%	



Gambar 18. ROC *Curve* Algoritma *Neural Network*

Dari proses pengukuran akurasi masing-masing algoritma dengan *confusion matrix* dan ROC *curve*, diperoleh nilai *accuracy* dan AUC:

Tabel 11. Perbandingan *Accuracy* dan AUC

	DT (C4.5)	NB	k-NN	LR	NN
Accuracy	0.6974	0.5633	0.8623	0.5576	0.5184
AUC	0.7280	0.6240	0.5000	0.6420	0.5930

Akurasi dari masing-masing algoritma klasifikasi diuji dengan uji beda parametrik t-*test*, diperoleh tabel uji t-*test*:

Tabel 12. T-*Test Significance*

	DT (C4.5)	NB	k-NN	LR	NN
DT (C4.5) 0.697 +/- 0.113		**0.009**	**0.001**	**0.002**	**0.000**
NB 0.563 +/- 0.089			**0.000**	0.867	0.224
k-NN 0.862 +/- 0.064				**0.000**	**0.000**
LR 0.558 +/- 0.053					0.114
NN 0.518 +/- 0.046					

Nilai yang dicetak tebal berarti lebih kecil dari *alpha* = 0.050 yang mengindikasikan adanya perbedaan signifikan diantara nilai rata-rata aktual. Dari tabel 13 dapat ditarik kesimpulan, algoritma *decision tree* (C4.5) memiliki akurasi paling bagus (dominan) terhadap algoritma yang lain. Berikutnya ada algoritma *naive bayes*, *logistic regression*, dan *neural network*, tidak ada perbedaan signifikan diantara algoritma tersebut. Namun demikian, algoritma *logistic regression* dan *neural network* tidak lebih baik dari algoritma k-*nearest neighbor* (k-NN).

KESIMPULAN

Penelitian dengan membandingkan lima algoritma klasifikasi *decision tree* (C4.5), *naive bayes*, k-*nearest neighbor*, *logistic regression*, dan *neural network* untuk memprediksi kerugian *tower provider* yang diakibatkan oleh penalti dari operator telekomunikasi, menggunakan metode validasi k-*fold cross validation*, serta dilakukan uji beda terhadap akurasi masing-masing algoritma dengan uji beda parametrik t-*test* menunjukkan bahwa algoritma *decision tree* (C4.5) memiliki akurasi paling bagus (dominan) terhadap algoritma

yang lain. Berikutnya ada algoritma *naive bayes*, *logistic regression*, dan *neural network*, tidak ada perbedaan signifikan diantara algoritma tersebut. Namun demikian, algoritma *logistic regression* dan *neural network* tidak lebih baik dari algoritma k-*nearest neighbor* (k-NN). Setelah diperoleh algoritma yang paling bagus tingkat akurasinya yakni algoritma *decision tree* (C4.5), maka algoritma tersebut dapat digunakan untuk memprediksi kerugian *tower provider* yang diakibatkan oleh penalti dari operator telekomunikasi sehingga *tower provider* bisa mengambil langkah terbaik untuk mengurangi bahkan meniadakan tingkat kerugian.

DAFTAR PUSTAKA

[1] S. Assery, "Pengguna Seluler Indonesia Semakin Meningkat," *Kedaulatan Rakyat*, Yogyakarta, p. 20, 2008.

[2] M. K. dan I. dan K. B. K. P. M. Menteri Dalam Negeri, Menteri Pekerjaan Umum, *Pedoman Pembangunan dan Penggunaan Bersama Menara Telekomunikasi*. 2009.

[3] L. Yu, X. Yao, S. Wang, and K. K. Lai, "Expert Systems with Applications Credit Risk Evaluation using a Weighted Least Squares SVM Classifier with Design of Experiment for Parameter Selection," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 15392–15399, 2011, doi: 10.1016/j.eswa.2011.06.023.

[4] R. S. Wahono, N. S. Herman, and S. Ahmad, "A Comparison Framework of Classification Models for Software Defect Prediction," vol. 20, no. 10, pp. 1945–1950, 2014, doi: 10.1166/asl.2014.5640.

[5] I. Brown and C. Mues, "An Experimental Comparison of Classification Algorithms for Imbalanced Credit Scoring Data Sets," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3446–3453, 2012, doi: 10.1016/j.eswa.2011.09.033.

[6] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining Third Edition*. Elsevier Inc., 2011.

[7] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. London: The MIT Press, 2001.

[8] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annu. Eugen.*, vol. 7, pp. 179–188, 1936, doi: 10.1111/j.1469-1809.1936.tb02137.x.

[9] F. Gorunescu, *Data Mining Concepts, Models, and Techniques*. Verlag Berlin Heidelberg: Springer, 2011.

[10] T. W. Liao and E. Triantaphyllou, *Recent Advances in Data Mining of Enterprise Data: Algorithms and Applications*. World Scientific, 2007.

[11] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

[12] B. Kitchenham *et al.*, "Systematic Literature Reviews in Software Engineering – A Tertiary Study," *Inf. Softw. Technol.*, vol. 52, no. 8, pp. 792–805, 2010, doi: 10.1016/j.infsof.2010.03.006.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Second. Stanford, California: Springer, 2008.

[14] D. T. Larose, *Discovering Knowledge in Data an Introduction to Data Mining*. 2005.

[15] X. Wu and V. Kumar, *The Top Ten Algorithms in Data Mining*, 1st Editio. New York: Chapman and Hall, 2009.

[16] C. M. Bishop, *Neural Network for Pattern Recognition*. Oxford: Clarendon Press, 1995.

[17] E. Alpaydin, *Introduction to Machine Learning Second Edition*, 2nd ed. The MIT Press, 2010.

[18] O. Maimon and L. Rokach, "Introduction to Knowledge Discovery in Database," in *Data Mining and Knowledge Discovery Handbook*, Tel Aviv, 2014, pp. 1–17.

[19] G. W. Snedecor and W. G. Cochran, *Statistical Methods*. Ames, Iowa: The Iowa State Press, 1989.

[20] H. Crc and M. Hofmann, *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. CRC Press, 2014.