Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)
Vol 2, No.1, March 2019, pp. 18 - 27
p-ISSN: 2614-3372 | e-ISSN: 2614-6150

Data Mining Optimization Using Sample Bootstrapping and Particle Swarm Optimization in the Credit Approval Classification

[1]Andre Alvi Agustian, [2]Achmad Bisri
[1,2]Teknik Informatika, Fakultas Teknik, Universitas Pamulang
Email: [1]realvitian@gmail.com, [2]achmadbizri@gmail.com

Article Info

Article history:
Received Aug 12th, 2018
Revised Sept 20th, 2018
Accepted Des 26th, 2018

Keyword:
Algoritma C4.5
Naïve Bayes
Particle Swarm Optimization
Persetujuan Kredit
Sample Bootstrapping

ABSTRACT

Credit approval is a process carried out by the bank or credit provider company. Where the process is carried out based on credit requests and credit proposals from the borrower. Credit approval is often difficult for banks or credit providers. Where the number of requests and classifications must be made on various data submitted. This study aims to enable banks or credit card issuing companies to carry out credit approval processes effectively and accurately in determining the status of the submissions that have been made. This research uses data mining techniques. This study uses a Credit Approval dataset from UCI Machine Learning, where there is a class imbalance in the dataset. 14 attributes are used as system inputs. This study uses the C4.5 and Naive Bayes algorithms where optimization is needed using Sample Bootstrapping and Particle Swarm Optimization (PSO) in the algorithm so that the results of the research produce good accuracy and are included in the good classification. After using the optimization, it produces an accuracy rate of C4.5 which is initially 85.99% and the AUC value of 0.904 becomes 94.44% with the AUC value of 0.969 and Naive Bayes which initially has an accuracy value of 83.09% with an AUC value of 0.916 to 90 , 10% with an AUC value of 0.944.

Corresponding Author:
Andre Alvi Agustian,
Teknik Informatika, Fakultas Teknik,
Universitas Pamulang,
Email: realvitian@gmail.com

DOI: http://dx.doi.org/10.24014/ijaidm.v2i1.6299

1. PENDAHULUAN

Persetujuan kredit bagi pihak perbankan merupakan langkah penting guna mengklasifikasi layaknya calon peminjam kredit tersebut. Pihak perbankan dapat menilai dan mengklasifikasikan calon peminjam kredit dengan menilai risiko kredit. Untuk menilai risiko kredit, perlu untuk melihat lebih dekat situasi ekonomi dan hukum peminjam serta lingkungan yang relevan (misalnya industri, pertumbuhan ekonomi). Kualitas proses persetujuan kredit tergantung pada dua faktor, yaitu presentasi yang transparan dan komprehensif tentang risiko ketika memberikan pinjaman di satu sisi, dan penilaian yang memadai atas risiko-risiko ini di sisi lain. Lebih jauh lagi, tingkat efisiensi proses persetujuan kredit merupakan elemen pemeringkatan penting. Karena perbedaan yang cukup besar dalam sifat dari berbagai peminjam (misalnya orang-orang pribadi, perusahaan yang terdaftar, penguasa, dan lain-lain). Dan aset yang akan dibiayai (misalnya perumahan real estate, produksi, mesin, daln lain-lain). Oleh karena itu, perlu untuk membedakan, dan bagian ini menjelaskan kriteria penting yang harus diperhitungkan dalam mendefinisikan diferensiasi ini dalam hal risiko dan efisiensi [1].

Salah satu permasalahan yang sering timbul dalam perbankan adalah banyak nasabah yang tidak layak yang berakibat pada terhambatnya pembayaran angsuran, sehingga diperlukan sebuah sistem yang dapat mengklasifikasi calon nasabah mana yang dapat disetujui pengajuan kreditnya dan mana yang tidak dapat disetujui. Sehingga pihak perbankan dapat mengatasi permasalahan tersebut sejak dini. Dengan menggunakan teknologi di bidang *data mining* yang mengoptimasi proses pencarian informasi prediksi

kedalam *database* yang besar. Dengan ini diharapkan teknik *data mining* mampu menghasilkan informasi yang berguna dalam klasifikasi tentang nasabah yang layak dan tidak layak dalam hal pengajuan kredit. Dalam hal ini pihak bank mementingkan faktor keuntungan secara ekonomi [2].

Decision tree merupakan sebuah metode klasifikasi dan juga prediksi yang paling banyak dan sering digunakan karena sangat terkenal dan memiliki akurasi dan pemahaman yang mudah. *Decision tree* dapat mentransformasi informasi yang cukup banyak menjadi sebuah pohon keputusan yang memiliki *rule*. *Rule* yang dengan sangat mudah dipahami dengan bahasa alami. *Decision tree* juga berfungsi untuk mengeksplorasikan data, menemukan keterkaitan antara sejumlah calon variabel input dengan variabel target [3].

Namun metode *desicion tree* algoritma C4.5 memiliki kelemahan dalam mengkonstruksi pohon keputusan, dimana algoritma C4.5 membaca keseluruhan data *training* dari *storage* dan disimpan ke dalam memori. Metode *decision tree* algoritma C4.5 juga memiliki kelemahan lain, yaitu jika digunakan pada *dataset* yang besar, karena waktu komputasi yang diperlukan dalam mengklasifikasi dataset tersebut cukup tinggi [5].

Naive Bayes merupakan metode klasifikasi sederhana yang diadaptasi dari teori *Bayesian* dengan asumsi independen yang kuat. *Naive Bayes* dapat menangani jumlah variabel independen yang berubah-ubah baik *continuos* maupun kategorial. Algoritma *Naive Bayes* membuat prediksi menggunakan teori *Bayesian* yang menggabungkan bukti dan pengetahuan sebelumnya dalam sebuah prediksi [4].

Metode *Naive Bayes* memiliki kekurangan dimana banyaknya celah untuk mengurangi keefektifan metoda ini dan akibatnya meloloskan dokumen ke dalam beberapa kelas meskipun jelas dokumen tidak layak berada di kelas tersebut [6]. *Naive Bayes* memiliki dua kelemahan utama, yaitu akurasi klasifikasinya menurun ketika atribut tidak independen, dan tidak dapat menangani nonparametric atribut kontinyu [7]. Namun kekurangan dari *Naïve Bayes* ukuran vektor fitur yang dihasilkan cukup besar dan memerlukan sebuah teknik untuk mengurangi ukuran vektor [8]. *Naïve Bayes* juga memiliki kelemahan dimana sebuah probabilitas tidak bisa mengukur seberapa besar tingkat keakuratan sebuah prediksi. Selain itu, *Naïve Bayes Classifier* juga memiliki kelemahan pada seleksi atribut sehingga dapat mempengaruhi nilai akurasi. Sehingga, *Naïve Bayes* perlu dilakukan optimasi dengan memberikan bobot pada atribut agar *Naïve Bayes* dapat bekerja lebih efektif [9].

Kelemahan tersebut dapat diatasi dengan metode optimasi yang diintegrasikan dengan metode-metode tersebut. Salah satu metode optimasi yang cukup populer dan sering digunakan yaitu *Particle Swarm Optimization (PSO). Particle Swarm Optimization (PSO)* adalah metode optimasi yang sudah terbukti efektif dan sering digunakan sebagai metode untuk memecahkan masalah optimasi yang bersifat multidimensi dan multiparameter pada *machine learning* seperti metode klasifikasi [10]. Ketika diaplikasikan pada beberapa kasus untuk mengoptimasi algoritma klasifikasi, PSO dapat lebih meningkatkan tingkat akurasi daripada *Genetic Algorithm* [11].

Kelemahan lainnya adalah *decision tree* diketahui tidak stabil. Artinya, jika kumpulan data asli diubah (terganggu) dalam beberapa cara, semisal *missing values*, dimana *dataset* yang digunakan terdapat *missing values*, maka penggolong yang dibangun dari data yang diubah dapat sangat berbeda dari pengklasifikasi aslinya [12].

Sample Bootstrapping diusulkan untuk mengatasi ketidakstabilan *decision tree* dengan prosedur resampling khusus. Sebuah titik data dipilih secara *random* dari data dan kemudian dikembalikan. Proses ini diulang n kali, dimana n adalah ukuran sampel. Sampel *bootstrap* yang dihasilkan terdiri dari n titik data namun akan berisi data yang direplikasi [12]. *Sample bootstrapping* juga dapat menangani *missing values* yang terjadi pada dataset dengan menggunakan *bootstrap* untuk memilih faktor aktif dalam sebuah percobaan dan memperkirakan data hilangnya [13].

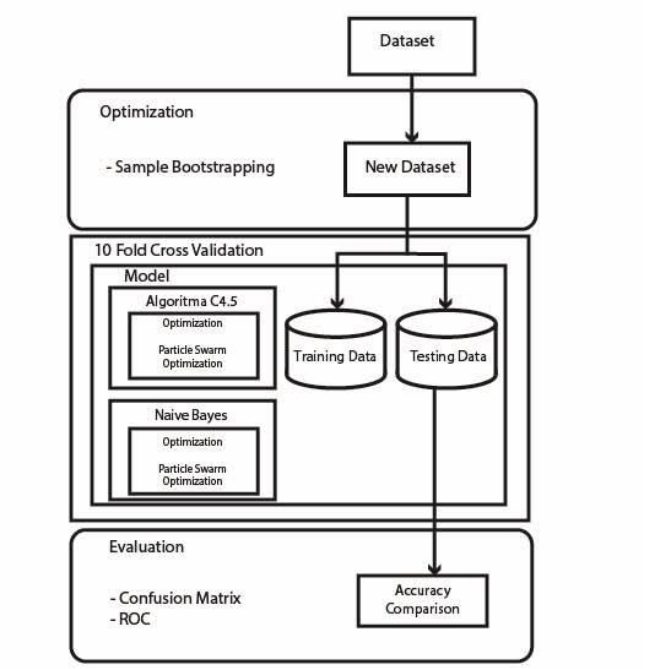
Dalam penelitian ini, metode optimasi *Sample Bootstrapping* akan digunakan pada metode klasifikasi algoritma C4.5 dan *Naive Bayes* untuk mengatasi ketidakstabilan pada metode *decision tree* algoritma C4.5 dan menangani *missing values* pada dataset yang digunakan sedangkan optimasi *Particle Swarm Optimization* digunakan untuk meningkatkan akurasi dalam klasifikasi persetujuan kredit yang dilakukan pihak bank atau penyedia kredit.

2. METODOLOGI PENELITIAN

Metode yang diusulkan dalam penelitian ini menggunakan metode *decision tree* dan *naïve bayes* dimana dilakukan optimasi menggunakan *Particle Swarm Optimization* dan *Sample Bootstrapping* untuk meningkatkan kinerja metode klasifikasi yang digunakan dan menangani ketidakstabilan pada metode tersebut. Sedangkan untuk proses validasi menggunakan *10-fold cross validation.* Selanjutnya hasil pengukuran menggunakan T-Test. Model kerangka metode yang diusulkan terlihat pada Gambar 1.

Pada Gambar 1, dalam pengolahan awal *dataset* yang digunakan dioptimasikan dengan *Sample Bootstrapping* dimana dapat dihasilkan dataset yang stabil dan tidak terdapat *missing values* yang dapat

mempengaruhi akurasi penelitian. Selanjutnya *dataset* dibagi menjadi data *training* dan *testing.* Setelah itu dimasukan kedalam *classifier* dengan metode *decision tree* dan *naïve bayes.* Kemudian untuk lebih meningkatkan tingkat akurasi dan kinerja dari metode yang digunakan dilakukan optimasi menggunakan *Particle Swarm Optimization.* Lalu dilakukan validasi menggunakan *10-fold cross validation.* Hasil dari validasi akan menghasilkan data yang akan diukur dalam *Confusion Matrix* dan Kurva ROC berupa *Accuracy* dan AUC.



Gambar 1. Kerangka pemikiran model yang diusulkan

Menurut Efrondan Tibshirani [14], prosedur *Sample Bootstrapping* dapat dituliskan sebagai sebagai berikut:

1. Menentukan jumlah B, yaitu sampel independen Bootstrap $X^{*1}, X^{*2}, \dots, X^{*B}$ di mana setiap sampel mengandung n data yang diperoleh dari x (data awal).
2. Mengevaluasi replikasi di setiap sampel, dengan menggunakan Persamaan 1 dibawah ini.

$$Bootstrap\ 0^{*}(b) = s(X^{*b}) b = 1, 2, \dots, B \quad (1)$$

3. Mengestimasi standar error $seB_{F\ B}(\theta)$ dengan menggunakan standar deviasi untuk Bootstrap yang direplikasi B kali, dapat dilihat pada Persamaan 2 berikut.

$$se_B = \left\{ \sum_{b=1}^{B} \left[\hat{\theta} * (b) - \hat{\theta} * (.) \right]^2 \right\}^{1/2} \quad (2)$$

Dimana: $\hat{\theta} P^{*P}(.) = \sum_{b=1}^{B} \hat{\theta} * (b)/B$

Algoritma PSO dapat dijelaskan sebagai berikut [15]:

1. Inisialisasi swarm, P(t), dari partikel sedemikian hingga posisi $x_i(t)$ dari masing-masing partikel $Pi \in P(t)$ adalah acak.
2. Hitung performansi *(fitness)* F dari masing-masing partikel pada posisinya saat ini yaitu $F(x_i(t))$.
3. Bandingkan performansi tiap partikel saat ini terhadap performansi terbaik sebelumnya, yaitu pbest (*personal best*).
   Jika $F(x_i(t)) < pbest_i$ maka:
   a. $pbest_i \leftarrow F(x_i(t))$
   b. $x_{pbesti} \leftarrow x_i(t)$
4. Bandingkan performansi tiap partikel saat ini terhadap performansi terbaik dari seluruh partikel dalam kelompoknya, yaitu *gbest (global best*).
   Jika $F(x_i(t)) < gbest_i$ maka:

a. $gbest_i \leftarrow F(x_i(t))$
b. $x_{gbesti} \leftarrow x_i(t)$

5. Hitung vektor kecepatan tiap partikel, dapat dilihat pada Persamaan 3.

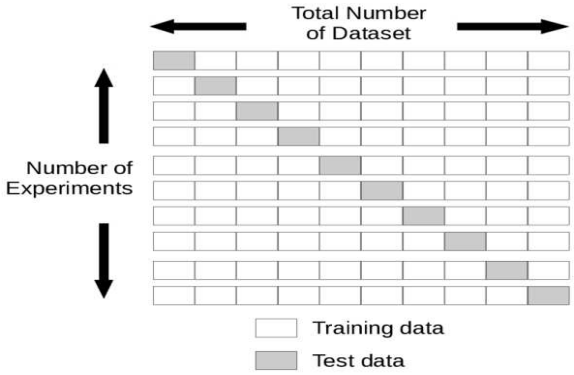
$$V_i(t) = V_i(t-1) = \rho_1 \, (x_{pbesti} - x_i(t-1)) + \rho_2(x_{gbesti} - x_i(t-1)) \quad (3)$$

Nilai ρ_1 dan ρ_2 didefinisikan sebagai : $\rho_1 = r_1c_1$ dan $\rho_2 = r_2c_2$, dimana c_1 dan c_2 disebut konstanta akselerasi, sedangkan r adalah suatu bilangan acak.

6. Perbaharui posisi tiap partikel:
   a. x_i (t) x_i (t-1) + v_i (t)
   b. t= t +1
7. Kembali ke langkah 2 hingga tercapai kriteria penghentian.

Selanjutnya tahap evaluasi dilakukan untuk memperoleh *accuracy* dan AUC. Lalu dilakukan uji T-Test yang bertujuan untuk mengetahui adanya perbedaan hasil antara metode klasifikasi sebelum dioptimasi dan setelah dioptimasi. Evaluasi terhadap model untuk mengukur *accuracy* dengan *Confusion Matrix,* sedangkan kurva ROC untuk menghasilkan AUC dengan proses validasi *10-fold cross validation.*

Cross validasi adalah metode statistik untuk membandingkan algoritma pembelajaran dan mengevaluasinya. Prosesnya data dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Data *training* dan data *testing* harus cross-over secara berturut-turut sebanyak K kali, misalkan pada 10-*fold cross validation* maka data dibagi menjadi 10 bagian dan dilakukan 10 kali iterasi untuk pengujiannya [16], ilustrasi K-*Fold Cross Validation* dapat dilihat pada Gambar 2.



Gambar 2. Ilustrasi *K-Fold Cross Validation*

Berikut adalah proses pengukuran *Confusion Matrix*, dapat dilihat pada tabel 1.

Tabel 1. *Confusion Matrix*

Predictive Class	Actual Class	
	Ya	Tidak
Ya	TP	FN
Tidak	FP	TN

Akurasi merupakan persentase dari data yang diprediksi secara benar. Perhitungan akurasi dapat dilihat pada Persamaan 4.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Keterangan :
TP : *True positives*, merupakan jumlah data dengan kelas positif yang diklasifikasikan positif.
TN : *True negatives*, merupakan jumlah data dengan kelas negatif yang diklasifikasikan negatif.
FP : *False positives*, merupakan jumlah data dengan kelas positif diklasifikasikan negatif.
FN : *False negatives,* merupakan jumlah data dengan kelas negatif diklasifikasikan positif.

Pedoman umum untuk mengklasifikasikan keakuratan pengujian menggunakan AUC menurut Gorunescu [17]:

a. 0.90 - 1.00 = *Excellent Classification*.
b. 0.80 - 0.90 = *Good Classification*.
c. 0.70 - 0.80 = *Fair Classification*.

d. 0.60 - 0.70 = *Poor Classification*.
e. 0.50 - 0.60 = *Failure*

3. HASIL PENELITIAN

Penelitian dilakukan dengan menggunakan laptop berbasis Intel CoreTM i3 – 6006U 1.99GHz, dengan RAM 4 GB dan sistem operasi Microsoft Windows 7 Ultimate 64-bit. Sedangkan untuk aplikasi pengolahan data dirancang menggunakan bahasa pemrograman PHP dan aplikasi Rapidminer 5.0.

Data pengajuan kredit yang digunakan adalah dataset public *Credit Approval Data Set* dari UCI *Dataset Repository* dengan jumlah data sebanyak 690 dan 15 atribut. Dimana pada atribut kelas yaitu *Approved,* dengan *record Yes* sebanyak 307 *record* (44,5%) dan *record No* sebanyak 383 *record* (55,5%). Karakteristik dari dataset tersebut dapat dilihat pada tabel 2.

Tabel 2. Karakteristik *dataset Credit card Approval*

No	Atribut	Tipe Data	Keterangan
1	*Gender*	*Integer*	Kelamin
2	*Age*	*Integer*	Usia
3	*Debt*	*Integer*	Hutang
4	*Married*	*Binominal*	Status Pernikahan
5	*Bank Customer*	*Polynominal*	Nasabah Bank
6	*Education Level*	*Polynominal*	Pendidikan
7	*Etnichity*	*Polynominal*	Etnis/suku
8	*Years Employed*	*Integer*	Lama bekerja
9	*Prior Default*	*Binominal*	Kepemilikan kartu kredit
10	*Employed*	*Binominal*	Status karyawan
11	*Credit Score*	*Integer*	Riwayat kredit
12	*Driver License*	*Binominal*	SIM
13	*Citizen*	*Polynominal*	Kota
14	*Income*	*Integer*	Penghasilan
15	*Approved*	*Binominal*	Status Pengajuan (Label)

Dalam penelitian ini menggunakan metode *Decision Tree* dan *Naïve Bayes* dengan optimasi *Sample Bootstrapping* dan *Particle Swarm Optimization* dimana dilakukan validasi dengan *10-fold cross validation* dan evaluasi menggunakan *confusion matrix* dan kurva ROC untuk mendapatkan nilai a*ccuracy* dan AUC. Dimana perhitungan dan penerapan metode menggunakan aplikasi Rapidminer dan hasil penelitian adalah sebagai berikut:

3.1 Implementasi Metode Decision Tree Algoritma C4.5

Proses pengolahan data dan perhitungan penelitian menggunakan Rapidminer dengan metode *decision tree* algoritma C4.5 untuk menghasilkan *Accuracy* dan AUC. Berikut adalah perhitungan a*ccuracy* dan AUC yang dihasilkan dengan metode C4.5, dapat dilihat pada tabel 3 dibawah ini:

Tabel 3. *Confusion matrix decision tree*

Predictive Class	Actual Class	
	Ya	Tidak
Ya	93	23
Tidak	6	85

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$
$$Accuracy = \frac{93 + 85}{93 + 23 + 6 + 85}$$
$$Accuracy = \frac{178}{207}$$
$$Accuracy = 0{,}8599$$
$$Accuracy = 85{,}99\%$$

Dari jumlah data sebanyak 207 klasifikasi kelas dengan label Yes sebanyak 99 *record* dan label No sebesar 108 *record*. Data diprediksi yang sesuai dengan label Yes sejumlah 93, data yang diprediksi Yes tetapi kenyataannya No sejumlah 23, data yang diprediksi No tetapi kenyataannya Yes sejumlah 6, dan sedangkan data yang diprediksi No dan sesuai sejumlah 85. Sedangkan hasil perhitungan AUC yang didapatkan sebesar 0,904.

3.2 Implementasi Metode Naïve Baye

Proses pengolahan data dan perhitungan penelitian menggunakan Rapidminer dengan metode *Naïve Bayes* untuk menghasilkan a*ccuracy* dan AUC. Berikut adalah perhitungan *Accuracy* dan AUC yang dihasilkan dengan metode *Naïve Bayes*, dapat dilihat pada Tabel 4.

Tabel 4. *Confusion Matrix Naive Bayes*

Predictive Class	Actual Class	
	Ya	Tidak
Ya	80	16
Tidak	19	92

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$
$$Accuracy = \frac{80 + 92}{80 + 19 + 16 + 92}$$
$$Accuracy = \frac{172}{207}$$
$$Accuracy = 0{,}8309$$
$$Accuracy = 83{,}09\%$$

Dari jumlah data sebanyak 207 klasifikasi kelas dengan label Yes sebanyak 99 *record* dan label No sebesar 108 *record*. Data diprediksi yang sesuai dengan label Yes sejumlah 80, data yang diprediksi Yes tetapi kenyataannya No sejumlah 16, data yang diprediksi No tetapi kenyataannya Yes sejumlah 19, dan sedangkan data yang diprediksi No dan sesuai sejumlah 92. Sedangkan hasil perhitungan AUC yang didapatkan sebesar 0,916.

3.3 Implementasi Metode C4.5 + Sample Bootstrapping (C4.5 + SB)

Proses pengolahan data dan perhitungan penelitian menggunakan Rapidminer dengan metode C4.5 + *Sample Bootstrapping* untuk menghasilkan *accuracy* dan AUC. Berikut adalah perhitungan *accuracy* dan AUC yang dihasilkan dengan metode C4.5 + *Sample Bootstrapping*:

Tabel 5. *Confusion Matrix Decision Tree* C4.5 + *Sample Bootstrapping*

Predictive Class	Actual Class	
	Ya	Tidak
Ya	169	27
Tidak	22	196

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$
$$Accuracy = \frac{169 + 196}{169 + 22 + 27 + 196}$$
$$Accuracy = \frac{365}{414}$$
$$Accuracy = 0{,}8816$$
$$Accuracy = 88{,}16\%$$

Dari jumlah data sebanyak 414 klasifikasi kelas dengan label Yes sebanyak 196 *record* dan label No sebesar 218 *record*. Data diprediksi yang sesuai dengan label Yes sejumlah 169, data yang diprediksi Yes tetapi kenyataannya No sejumlah 27, data yang diprediksi No tetapi kenyataannya Yes sejumlah 22, dan sedangkan data yang diprediksi No dan sesuai sejumlah 196. Sedangkan hasil perhitungan AUC yang didapatkan sebesar 0,925.

3.4 Implementasi Metode Naïve Bayes + Sample Bootstrapping (Naive Bayes + SB)

Proses pengolahan data dan perhitungan penelitian menggunakan Rapidminer dengan metode *Naïve Bayes + Sample Bootstrapping* untuk menghasilkan *Accuracy* dan AUC. Berikut adalah perhitungan *Accuracy* dan AUC yang dihasilkan dengan metode *Naïve Bayes + Sample Bootstrapping*, dapat dilihat pada Tabel 6.

Tabel 6. *Confusion Matrix Naive Bayes + Sample Bootstrapping*

Predictive Class	Actual Class	
	Ya	Tidak
Ya	146	11
Tidak	45	212

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$
$$Accuracy = \frac{146 + 212}{146 + 45 + 11 + 212}$$
$$Accuracy = \frac{358}{414}$$
$$Accuracy = 0{,}8647$$
$$Accuracy = 86{,}47\%$$

Dari jumlah data sebanyak 414 klasifikasi kelas dengan label Yes sebanyak 157 *record* dan label No sebesar 257 *record*. Data diprediksi yang sesuai dengan label Yes sejumlah 146, data yang diprediksi Yes tetapi kenyataannya No sejumlah 11, data yang diprediksi No tetapi kenyataannya Yes sejumlah 45, dan sedangkan data yang diprediksi No dan sesuai sejumlah 212. Sedangkan hasil perhitungan AUC yang didapatkan sebesar 0,955.

3.5 Implementasi Metode C4.5 + Sample Bootstrapping + Particle Swarm Optimization (C4.5 + SB + PSO)

Proses pengolahan data dan perhitungan penelitian menggunakan Rapidminer dengan metode *C4.5 + Sample Bootstrapping + Particle Swarm Optimization* untuk menghasilkan *Accuracy* dan AUC. Berikut adalah perhitungan *Accuracy* dan AUC yang dihasilkan dengan metode *C4.5 + Sample Bootstrapping + Particle Swarm Optimization*, dapat dilihat pada Tabel 7.

Tabel 7. *Confusion Matrix Decision Tree with Optimization*

Predictive Class	Actual Class	
	Ya	Tidak
Ya	164	11
Tidak	12	227

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$
$$Accuracy = \frac{164 + 227}{164 + 11 + 12 + 227}$$
$$Accuracy = \frac{391}{414}$$
$$Accuracy = 0{,}9444$$
$$Accuracy = 94{,}44\%$$

Dari jumlah data sebanyak 414 klasifikasi kelas dengan label Yes sebanyak 176 *record* dan label No sebesar 238 *record*. Data diprediksi yang sesuai dengan label Yes sejumlah 164, data yang diprediksi Yes tetapi kenyataannya No sejumlah 11, data yang diprediksi No tetapi kenyataannya Yes sejumlah 12, dan sedangkan data yang diprediksi No dan sesuai sejumlah 227. Sedangkan hasil perhitungan AUC yang didapatkan sebesar 0,969.

3.6 Implementasi metode Naive Bayes + Sample Bootstrapping + Particle Swarm Optimization (Naive Bayes + SB + PSO)

Proses pengolahan data dan perhitungan penelitian menggunakan Rapidminer dengan metode *Naive Bayes + Sample Bootstrapping + Particle Swarm Optimization* untuk menghasilkan *Accuracy* dan AUC. Berikut adalah perhitungan *Accuracy* dan AUC yang dihasilkan dengan metode *Naive Bayes + Sample Bootstrapping + Particle Swarm Optimization*, dapat dilihat pada Tabel 8.

Tabel 8. *Confusion Matrix Naive Bayes with Optimization*

Predictive Class	Actual Class	
	Ya	Tidak
Ya	154	23
Tidak	18	219

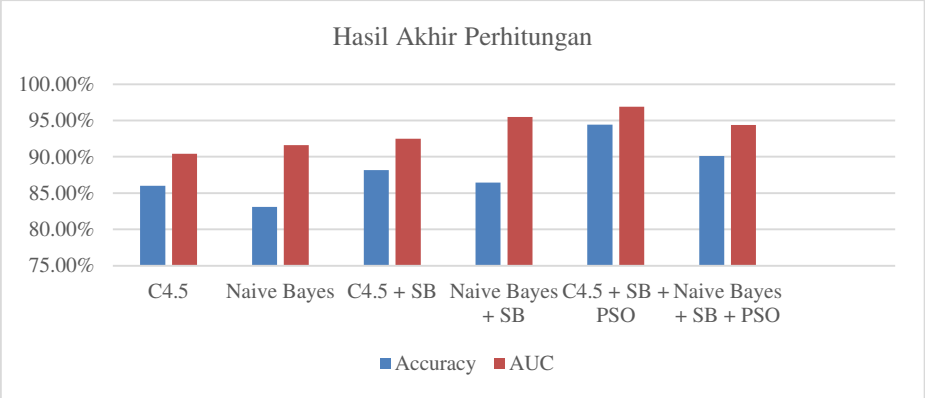
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$
$$Accuracy = \frac{154 + 219}{154 + 23 + 18 + 219}$$
$$Accuracy = \frac{373}{414}$$
$$Accuracy = 0{,}9010$$
$$Accuracy = 90{,}10\%$$

Dari jumlah data sebanyak 414 klasifikasi kelas dengan label Yes sebanyak 172 *record* dan label No sebesar 242 *record*. Data diprediksi yang sesuai dengan label Yes sejumlah 154, data yang diprediksi Yes tetapi kenyataannya No sejumlah 23, data yang diprediksi No tetapi kenyataannya Yes sejumlah 18, dan sedangkan data yang diprediksi No dan sesuai sejumlah 219. Sedangkan hasil perhitungan AUC yang didapatkan sebesar 0,944.

Dapat dilihat perbandingan AUC dari enam metode yang dilakukan. Metode C4.5 menghasilkan AUC sebesar 0,904 dan C4.5 + *Sample Bootstrapping* menghasilkan AUC sebesar 0,925, sedangkan *Naive Bayes* menghasilkan AUC sebesar 0,916 dan *Naive Bayes* + *Sample Bootstrapping* menghasilkan AUC sebesar 0,955. Setelah dilakukan optimasi dengan *Sample Bootstrapping* dan *Particle Swarm Optimization* dapat dilihat bahwa metode C4.5 meningkat AUCnya menjadi 0,969 begitu juga dengan *Naive Bayes* meningkat menjadi 0,944, dapat dilihat pada Tabel 9.

Tabel 9. Hasil akhir perhitungan Accuracy dan AUC

Metode	Accuracy	AUC
C4.5	85,99%	0,904
Naive Bayes	83,09%	0,916
C4.5 + *SB*	88,16%	0,925
Naive Bayes + SB	86,47%	0,955
C4.5 + *SB + PSO*	94,44%	0,969
Naive Bayes + SB + PSO	90,10%	0,944



Gambar 3. Grafik hasil akhir perhitungan

Dapat dilihat pada Gambar 3, menunjukkan hasil akhir dari perhitungan yang menghasilkan akurasi dan AUC yang didapatkan yaitu, metode C4.5 dengan optimasi *Sample Bootstrapping* dan *Particle Swarm Optimization* menghasilkan akurasi terbesar yaitu 94,44% dan AUC 0,969. Sedangkan metode lain diantaranya C4.5 menghasilkan akurasi sebesar 85,99% dan AUC 0,904, C4.5 + *Sample Bootstrapping* 88,16% dan AUC 0,925, *Naive Bayes* 83,09% dan AUC 0,916, *Naive Bayes* + *Sample Bootstrapping* 86,47% dan AUC 0,955, *Nave Bayes* + *Sample Bootstrapping* + *Particle Swarm Optimization* 90,10% dan AUC 0,944.

Selanjutnya dilakukan pengujian hipotesis menggunakan uji *multi-paired sample t-Test Decision Tree* dan *Naive Bayes* dengan metode tersebut diintegrasikan dengan optimasi *Sample Bootstrapping* dan *Particle Swarm Optimization*. *T-Test* adalah hubungan antara variabel respon dengan varibel prediktor [18].

Hipotesis nol (H_0) menyatakan bahwa tidak ada perbedaan yang signifikan, sedangkan hipotesis alternatif (Ha) menyatakan bahwa adanya perbedaan yang signifikan sesudah diterpakan optimasi pada metode yang digunakan. Hasil pengujian T-test dapat dilihat pada Tabel 10.

Tabel 10. Hasil pengujian T-Test

	0.849 +/- 0.025	0.920 +/- 0.017	0.829 +/- 0.035	0.896 +/- 0.019
0.849 +/- 0.025		0.000	0.193	0.000
0.920 +/- 0.017			0.000	0.011
0.829 +/- 0.035				0.000
0.896 +/- 0.019				

Pada Tabel 10 menunjukan hasil dari t-test yang dilakukan dengan empat metode, bahwa untuk nilai uji t memiliki aturan apabila *P-value* < 0,05 terdapat perbedaan pada taraf signifikan yaitu 5%, dan apabila *P-value* > 0,05, maka tidak ada perbedaan antara sebelum optimasi dan sesudah optimasi. Hasil yang didapat dari nilai uji t-test berikut adalah sebagai berikut:

1. Algoritma C4.5 pada Algoritma C4.5 dengan *Sample Bootstrapping* dan *Particle Swarm Optimization* bernilai 0.000 berarti dimana p < 0.005 atau 0.000 < 0.005, yang artinya Ho tertolak, ada perbedaan yang signifikan.
2. Algoritma C4.5 pada Algoritma *Naive Bayes* bernilai 0.193 berarti dimana p > 0.005 atau 0.193 > 0.005, yang artinya Ha tertolak, tidak ada perbedaan yang signifikan.
3. Algoritma C4.5 pada Algoritma *Naive Bayes* dengan *Sample Bootstrapping* dan *Particle Swarm Optimization* bernilai 0.000 berarti dimana p < 0.005 atau 0.000 < 0.005, yang artinya Ho tertolak, ada perbedaan yang signifikan.
4. Algoritma C4.5 dengan *Sample Bootstrapping* dan *Particle Swarm Optimization* pada algoritma *Naive Bayes* bernilai 0.000 berarti dimana p < 0.005 atau 0.000 < 0.005, yang artinya Ho tertolak, ada perbedaan yang signifikan.
5. Algoritma C4.5 dengan *Sample Bootstrapping* dan *Particle Swarm Optimization* pada algoritma *Naive Bayes* dengan *Sample Bootstrapping* dan *Particle Swarm Optimization* bernilai 0.011 berarti dimana p > 0.005 atau 0.011 > 0.005, yang artinya Ha tertolak, tidak ada perbedaan yang signifikan.
6. Algoritma *Naive Bayes* pada algoritma *Naive Bayes* dengan *Sample Bootstrapping* dan *Particle Swarm Optimization* bernilai 0.000 berarti dimana p < 0.005 atau 0.000 < 0.005, yang artinya Ho tertolak, ada perbedaan yang signifikan.

Dari hasil penelitian ini didapat algoritma C4.5 dengan optimasi *Sample Bootstrapping* dan PSO mempunyai hasil akurasi yang lebih baik dibandingkan dengan algoritma C4.5, *Naïve Bayes* dan *Naïve Bayes* dengan optimasi.

4. KESIMPULAN

Dari hasil penelitian dan pengujian, model *Decision Tree* yang menggunakan algoritma C4.5 dengan optimasi PSO dan *Sample Bootstrapping* mendapatkan nilai 94.44%. Sementara hasil pengujian model *Naive Bayes* menggunakan optimasi PSO dan *Sample Bootstrapping* mendapatkan nilai akurasi 90,10%. Maka dapat ditarik kesimpulan bahwa optimasi PSO dan *Sample Bootstrapping* dapat meningkatkan kinerja metode klasifikasi yang digunakan, dan metode terbaik dalam penelitian ini yaitu *decision tree* dengan optimasi *Sample Bootstrapping* dan PSO dengan nilai pengujian 94,44% dengan nilai AUC sebesar 0,969.

REFERENSI

[1] Oesterreichische Nationalbank , Credit Approval Process and Credit Risk Management, Vienna: Oesterreichische Nationalbank , 2004.

[2] I. H. Witten dan E. Frank, Data Mining, Practical Machine Learning Tool and Techniques, San Francisco: Morgan Kauffman, 2005.

[3] K. dan E. T. Luthfi, Algoritma Data Mining, Yogyakarta: Andi Offset, 2009.

[4] L. Fausett, Fudamental of Neural Networks, New Jersey: Prentice-Hall, Inc, 1994.

[5] T. A. Setiawan, R. S. Wahono dan A. Syukur, "Integrasi Metode Sample Bootstrapping dan Weighted Principal Component Analysis untuk Meningkatkan Performa k Nearest Neighbor pada Dataset Besar," *Journal of Intelligent Systems,* vol. 1, no. 2, p. 1, 2015.

[6] S. R. Wicaksono, Sistem Berbasis Pengetahuan, Malang: Seribu Bintang, 2018.

[7] M. M. Arroyo dan L. E. Sucar, "Learning an Optimal Naive Bayes Classifier," *The 18th International Conference on Pattern Recognition,* vol. 3, pp. 1236-1239, 2006.

[8] C. Darujati dan A. B. Gumelar, "Pemanfaatan Teknik Supervised untuk Klasifikasi Teks Bahasa

Indonesia," vol. 16, no. 1, 2012.

[9] H. Muhammad, C. A. Prasojo, N. A. Sugianto, L. Suryatiningsih dan I. Cholissodin, "Optimasi Naive Bayes Classifier Dengan Menggunakan Particle Swarm Optimization pada Data Iris," *Jurnal Teknologi Informasi dan Ilmu Komputer,* vol. 4, no. 3, pp. 180-184, 2017.

[10] J. Kennedy dan R. Eberhart, "Particle Swarm Optimization," *Proc. IEEE Int. Conf. Neural Network,* 1995.

[11] T. Sousa, A. Silva dan A. Neves, "Particle Swarm based Data Mining Algorithms for classification tasks," *Parallel Computing,* no. 30, pp. 767-783, 2004.

[12] H. Ishwaran dan J. S. Rao, "Decision Trees, Advanced Techniques in Constructing," dalam *Encyclopedia of Medical Decision Making*, California, SAGE Publications, Inc., 2009, pp. 328-332.

[13] M. Qumsiyeh dan G. Saughnessy, "Comparison of Re-sampling Methods to Generalized Linear Models and Transformations in Factorial and Fractional Factorial Designs," *Journal of Modern Applied Statistical Methods,* vol. 11, no. 1, pp. 95-105, 2012.

[14] B. Efron dan R. J. Tibshirani, An Introduction to the Bootstrap, New York: Chapman & Hall, 1993.

[15] A. Engelbrecht, Computational Intelligence: An Introduction, New York: Halsted Press New York, 2002.

[16] J. Han, M. Kamber dan J. Pei, Data Mining Concepts and Techniques, Massachusetts: Morgan Kauffman, 2012.

[17] F. Gorunescu, Data Mining Concepts, Models & Techniques, Berlin: Springer-Verlag Berlin Heidelberg, 2011.

[18] D. T. Larose, Discovering Knowledge in Data, New Jersey: John Wiley & Sons, Inc., 2005.

BIBLIOGRAFI PENULIS

Andre Alvi Agustian, lahir di Tangerang, 22 Agustus 1994. Lulus di Sekolah Dasar Negeri 3 Larangan Utara pada 2006. Bersekolah di SMPN 1 Patuk, Yogyakarta hingga tahun 2009, lanjut Sekolah Menegah Kejuruan jurusan Teknik Komputer Jaringan dan lulus di Tahun 2012. Masuk Unniversitas Pamulang pada tahun 2013. Memperoleh gelar Sarjana Komputer (S.Kom) di bidang Teknik Informatika pada tahun 2018.

Achmad Bisri, memperoleh gelar Sarjana Komputer (S.Kom) dibidang Teknik Informatika dari STMIK Banten Jaya, Serang-Banten, gelar Magister Komputer (M.Kom) dibidang Teknik Informatika dari STMIK Eresha, Jakarta. Dia saat ini sebagai Dosen di Universitas Pamulang (Unpam), Tangerang Selatan. Minat Penelitiannya saat ini meliputi Software Engineering, Intelligence System, Machine Learning dan Data Mining.