

# Random Forest Dengan Random Search Terhadap Ketidakseimbangan Kelas Pada Prediksi Gagal Jantung

Muhammad Ali Abubakar<sup>1</sup>, Muliadi<sup>2\*</sup>, Andi Farmadi<sup>3</sup>, Rudy Herteno<sup>4</sup>, Rahmat Ramadhani<sup>5</sup>

<sup>12345</sup>Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Ilmu Komputer, Universitas Lambung Mangkurat, Kota Banjarbaru, Indonesia

e-mail: <sup>1</sup>muhammadaliabubakar@icloud.com, <sup>2\*</sup>muliadi@ulm.ac.id, <sup>3</sup>andifarmadi@ulm.ac.id, <sup>4</sup>rudy.herteno@ulm.ac.id, <sup>5</sup>rahmat.ramadhani@ulm.ac.id

---

Informasi Artikel	Diterima: 04-12-2022	Direvisi: 15-12-2023	Disetujui: 04-01-2023
-------------------	----------------------	----------------------	-----------------------

---

## Abstrak

Prediksi keberlangsungan hidup pasien gagal jantung telah dilakukan pada penelitian untuk mencari tahu tentang kinerja, akurasi, presisi dan performa dari model prediksi ataupun metode yang digunakan dalam penelitian, dengan menggunakan dataset heart failure clinical records. Namun dataset ini memiliki permasalahan yaitu bersifat tidak seimbang yang dapat menurunkan kinerja model prediksi karena cenderung menghasilkan prediksi kelas mayoritas. Pada penelitian ini menggunakan pendekatan level algoritma untuk mengatasi ketidakseimbangan kelas yaitu teknik bagging dengan metode *Random Forest* lalu digabungkan dengan metode Hyper-Parameter Tuning agar kinerja yang dihasilkan menjadi lebih baik. Selanjutnya model dilatih dengan dataset dan dibandingkan dengan metode lain, hasilnya menunjukkan bahwa *Random Forest* dengan *Random Search* Hyper Parameter-Tuning mencapai nilai AUC sebesar 0,906 dan untuk model *Random Forest* tanpa *Random Search* memperoleh nilai AUC sebesar 0,866.

**Kata Kunci:** Ketidakseimbangan Kelas, *Random Forest*, *Random Search*

## Abstract

*Prediction of the survival of heart failure patients has been carried out in research to find out about the performance, accuracy, precision and performance of the prediction model or method used in the study, using the heart failure clinical records dataset. However, this dataset has a problem, namely being unbalanced which can reduce the performance of the prediction model because it tends to produce predictions for the majority class. This study uses an algorithm level approach to overcome class imbalance, namely the bagging technique with the Random Forest method and then combined with the Hyper-Parameter Tuning method so that the resulting performance is better. Then the model was trained with the dataset and compared with other methods, the results showed that the Random Forest with Random Search Hyper Parameter-Tuning achieved an AUC value of 0,906 and for the Random Forest model without Random Search the AUC value of 0,866 was obtained.*

**Keywords:** Class Imbalance, *Random Forest*, *Random Search*

## 1. Pendahuluan

Kelangsungan hidup manusia tidak pernah lepas dari organ tubuh yang bekerja terus menerus. Jantung merupakan organ utama tubuh manusia karena tugasnya yang sangat penting yaitu memompa darah dan mendistribusikannya ke seluruh tubuh dan nantinya darahnya yang membawa oksigen dan nutrisi yang dibutuhkan oleh tubuh. Permasalahan yang muncul antara lain bahaya dan kematian gagal jantung bila diderita oleh tubuh manusia, tingginya faktor kematian akibat penyakit jantung akibat kurangnya pengetahuan

masyarakat tentang gejala atau tanda bila seseorang mengidap penyakit ini (Fiquriansyah et al., 2022) sehingga hal ini perlu diramalkan agar dapat dideteksi sejak dini dan diobati atau diobati secepatnya. Menurut (Chicco & Jurman, 2020) Machine Learning dapat memprediksi kelangsungan hidup pasien dari datanya.

Penyakit jantung merupakan gangguan yang terjadi pada sistem pembuluh darah besar sehingga menyebabkan jantung dan peredaran darah tidak berfungsi dengan baik. Penyakit yang berhubungan dengan jantung dan pembuluh darah antara lain: gagal jantung,



penyakit jantung koroner, dan penyakit jantung rematik. Gagal jantung adalah tahap akhir dari semua penyakit jantung. Masalah yang muncul antara lain bahaya dan kematian gagal jantung bila diderita oleh tubuh manusia, sehingga hal ini perlu diantisipasi agar dapat diketahui lebih dini dan diobati atau diobati sesegera mungkin untuk menurunkan angka kematian atau memperpanjang kelangsungan hidup pasien (Rahayu et al., 2020).

Penelitian menggunakan dataset Heart Failure Clinical Records bukan yang pertama dilakukan, penelitian lain juga pernah melakukan hal yang sama. Penelitian yang dilakukan oleh (Novaldy & Herliana, 2021) dengan menggunakan metode Naïve Bayes dengan optimasi PSO menghasilkan akurasi 92,67% dan nilai AUC sebesar 0.908. Lalu penelitian yang dilakukan oleh (Nurmasani & Pristyanto, 2021) dengan menggunakan algoritma staking menghasilkan akurasi sebesar 81% dan AUC sebesar 0.87.

Namun permasalahan yang dihadapi pada dataset Heart Failure Clinical Records yang digunakan dalam penelitian ini adalah adanya ketidakseimbangan data yang dapat mempengaruhi kinerja suatu algoritma klasifikasi, karena algoritma klasifikasi bekerja dengan mengasumsikan distribusi kelas pada dataset relatif seimbang. dan biaya kesalahan klasifikasi adalah sama. Hal ini tentu saja dapat menimbulkan risiko kesalahan klasifikasi dataset yang mengakibatkan kinerja suatu algoritma klasifikasi menjadi tidak optimal. Oleh karena itu diperlukan suatu teknik atau metode yang dapat mengatasi permasalahan yang berkaitan dengan distribusi kelas yang tidak seimbang pada dataset.

Dataset dikatakan kelasnya tidak seimbang apabila jumlah kelas positif dan negatif berbeda (Pristyanto, 2019). Keberadaan distribusi kelas yang tidak seimbang dapat mempengaruhi performa dari suatu algoritma klasifikasi, karena suatu algoritma klasifikasi bekerja dengan mengasumsikan distribusi kelas pada dataset relatif seimbang dan biaya kesalahan klasifikasi yang sama. Selain itu, penanganan distribusi class yang tidak seimbang pada suatu data merupakan salah satu langkah penting untuk meminimalisir risiko yang akan timbul pada proses data mining (Widjiyati, 2021). Mengenai penanganan distribusi kelas yang tidak seimbang pada dataset, ada dua pendekatan yang dapat diterapkan, yaitu pendekatan level data dan pendekatan level algoritma. Pendekatan level data biasanya dilakukan pada tahap pre-processing data dengan mengubah atau mengoreksi kecondongan distribusi kelas yang terdapat pada dataset. Metode yang sering

digunakan dalam pendekatan level data adalah dengan menerapkan teknik resampling dan sintesis data. Pada pendekatan level algoritma, cara kerjanya adalah menyesuaikan operasi algoritma yang ada untuk membuat sebuah classifier lebih kondusif untuk klasifikasi kelas minoritas, atau dengan kata lain, untuk memodifikasi atau menggabungkan beberapa algoritma. Beberapa metode yang sering digunakan dalam pendekatan ini adalah cost sensitive learning, bagging, boosting, dan stacking. Menurut (Wahono & Suryana, 2013) teknik bagging baik untuk mengatasi ketidakseimbangan kelas karena efektivitasnya.

Penelitian yang menggunakan metode *Random Forest* pernah dilakukan oleh (Annisa, 2019) dengan menggunakan beberapa metode klasifikasi yaitu *Decision Tree*, *Naïve Bayes*, *k-Nearest Neighbour*, *Random Forest*, dan *Decision Stump*. Dari hasil penelitian yang dilakukan didapatkan bahwa *Random Forest* mendapatkan hasil akurasi terbaik sebesar 80,38%. Selain itu juga penelitian yang dilakukan oleh (Zailani & Hanun, 2020) menggunakan metode klasifikasi *Random Forest* dan mendapatkan nilai akurasi sebesar 87,88%.

Random Forest adalah salah satu jenis metode bagging yang memiliki cara kerja dengan membangkitkan sejumlah pohon dari data sampel dimana pembuatan satu pohon selama proses pelatihan tidak bergantung pada pohon sebelumnya dan kemudian keputusan diambil berdasarkan yang paling banyak suara. Dua konsep yang mendasari Random Forest adalah membangun komposit pohon melalui bagging dengan mengganti dan memilih fitur secara acak untuk setiap pohon yang dibangun.

Untuk mengetahui perbedaan penelitian ini dengan penelitian sebelumnya, hasil klasifikasi akan dibandingkan dengan menggunakan model Random Forest. Namun, penelitian ini akan dilakukan dengan menyetel model Hyper-Parameter. Perlu dilakukan tuning pada Hyper-Parameter untuk mendapatkan nilai Hyper-Parameter yang optimal sehingga hasil model dapat meningkat. Pada penelitian yang dilakukan oleh (Leonardo et al., 2020) menghasilkan peningkatan yang signifikan pada saat melakukan optimasi menggunakan *Random Search*. Selain itu juga penelitian yang dilakukan oleh (Valarmathi & Sheela, 2021) menggunakan 3 dataset yang berbeda, *Random Search* memberikan peningkatan hasil prediksi pada ketiga dataset tersebut.

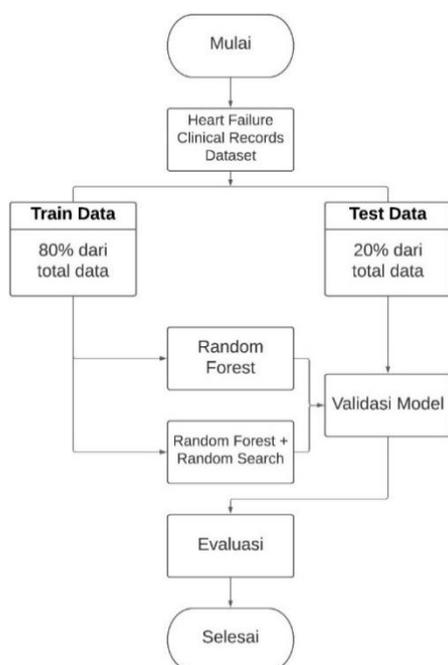
Pemilihan Hyper-Parameter yang tepat dapat memengaruhi kinerja model. metode *Random Search* akan memilih nilai untuk setiap hyperparameter secara independen menggunakan distribusi probabilitas dan

mengambil nilai sample secara random (Firdaus, 2022). metode *Random Search* baik untuk Tuning Hyper-Parameter karena efektivitas waktu komputasinya. Random search akan memiliki range percobaan yang lebih besar dibandingkan grid search jika jumlah percobaan yang ditentukan sama. Random search juga akan efektif digunakan apabila jumlah dimensi parameter yang dicobakan besar (Syukron et al., 2020).

Dari permasalahan yang telah dijabarkan sebelumnya, pada penelitian ini mengusulkan peningkatan model *Random Forest* menggunakan Hyper-Parameter Tuning *Random Search* terhadap ketidakseimbangan kelas (Class Imbalance) pada dataset Heart Failure Clinical Records.

## 2. Metode Penelitian

Alur penelitian dapat dilihat pada gambar 1 di bawah ini :



Gambar 1. Alur Penelitian

### Pengumpulan Data

Dataset yang digunakan pada penelitian ini adalah Dataset Heart Failure Clinical Records yang dirujuk berdasarkan penelitian yang dilakukan oleh (Chicco & Jurman, 2020). Pada dataset ini berisi informasi tentang atribut apa saja yang digunakan sebagai fitur untuk melakukan prediksi/Dataset Heart Failure Clinical Records sendiri memiliki 299 instance/sample dengan 13 fitur, untuk daftar nama dari fitur dapat dilihat pada tabel di bawah ini.

Dataset ini memiliki dua kelas yang terdiri dari 203 kelas non deceased (tidak

meninggal) dan 96 kelas deceased (meninggal) dengan perbandingan persentase pada kelas non deceased (tidak meninggal) sebesar 67,89% dan persentase kelas deceased (meninggal) sebesar 32,11%. Kelas non deceased (tidak meninggal) dilambangkan dengan angka 1 dan kelas deceased (meninggal) dilambangkan dengan angka 0.

### 1. Pembagian Data

Pada tahap ini dilakukan pembagian data menjadi dua bagian, yaitu data training (latih) dan data testing (uji). Data training digunakan untuk melatih algoritma dalam pembentukan sebuah model, sedangkan data testing digunakan untuk mengukur kinerja yang didapatkan dari data training. Data training dan data testing dibagi dengan proporsi 80% untuk data training dan sisanya 20% untuk data testing secara stratify, yaitu membagi dataset dengan mempertahankan proporsi kelas yang sama.

Data training digunakan untuk melatih model, sedangkan data testing merupakan representasi dari data baru yang belum pernah dilihat sebelumnya atau unseen data yang digunakan untuk mengetahui kinerja dari model yang telah dilatih.

### Random Forest

*Random Forest* (Pohon Acak) merupakan salah satu jenis metode bootstrap aggregating (bagging) yang memiliki cara kerja dengan membangkitkan sejumlah pohon dari data sampel di mana pembuatan satu pohon pada saat proses training tidak bergantung terhadap pohon sebelumnya kemudian dalam pengambilan keputusannya diambil berdasarkan voting terbanyak. Dua konsep yang menjadi dasar dari *Random Forest* adalah membangun gabungan dari pohon via bagging dengan penggantian dan penyeleksian fitur secara acak untuk tiap pohon yang dibangun. *Random Forest* mempunyai dua parameter utama, yaitu parameter  $m$  yang merupakan presentasi dari jumlah pohon yang akan dipakai dan parameter  $k$  yang merupakan representasi dari banyaknya fitur maksimal yang dipertimbangkan ketika proses percabangan pada pohon (Widjiyati, 2021).

Pada metode *Random Forest* terdapat *BalancedRandomForestClassifier* yang mana ini merupakan penyeimbangan data menggunakan *Random undersampling (RUS)*, yaitu menghitung selisih antara kelas mayoritas dan minoritas kemudian dilakukan perulangan selisih hasil perhitungan, selama perulangan data kelas mayoritas dihapus secara acak, sehingga jumlah kelas mayoritas sama dengan minoritas (Erna & Romi Satria, 2015).

Selama data di train, *Random Forest* menghasilkan beberapa Decision Tree (Pohon

Keputusan) secara acak yang diterapkan pada subset data, yang berisi subset baik instance data maupun fitur. Pada akhirnya, *Random Forest* memeriksa semua hasil biner dari Decision Tree ini dan memilih hasil akhirnya melalui suara mayoritas.

### Random Search

*Random Search* sebenarnya lebih praktis daripada grid search karena dapat diterapkan bahkan ketika menggunakan sekelompok komputer yang dapat gagal, dan memungkinkan eksperimen untuk mengubah "resolusi" dengan cepat : menambahkan percobaan baru ke set atau mengabaikan percobaan yang dadal adalah keduanya layak. Tentu saja, pencarian acak mungkin dapat ditingkatkan dengan mengotomatiskan apa yang dilakukan pencarian manual, yaitu, pengoptimalan berurutan. *Random Search* memiliki semua keuntungan praktis dari grid search (kesederhanaan konseptual, kemudahan implementasi, paralelisme sepele) dan perdagangan pengurangan kecil dalam efisiensi dalam ruang dimensi rendah untuk peningkatan besar dalam efisiensi dalam ruang pencarian dimensi tinggi.

### Evaluasi

Untuk mengevaluasi model klasifikasi digunakan AUC untuk mengetahui kinerja dan performa dari model *Random Forest* dan akan dibandingkan dengan model *Random Forest* yang menggunakan *Random Search* sebagai Parameter Tuning. AUC adalah ukuran kinerja yang populer dalam ketidakseimbangan kelas, nilai AUC yang tinggi menunjukkan kinerja yang lebih baik. Sehingga untuk memilih model mana yang terbaik, dapat dilakukan dengan menganalisa nilai AUC (Pujianto, 2016). Evaluasi hasil klasifikasi dari model *Random Forest* dan *Random Forest* dengan *Random Forest* pada dataset dengan menggunakan AUC untuk mendapatkan nilai performa dari setiap model pengklasifikasi. AUC dipilih sebagai metode evaluasi arena menurut penelitian terdahulu yang sudah dipaparkan menjelaskan bahwa AUC cocok untuk mengevaluasi nilai kinerja atau performa prediksi yang menggunakan dataset dengan permasalahan ketidakseimbangan data.

### 3. Hasil dan Pembahasan

Dataset yang digunakan pada penelitian ini adalah Dataset Heart Failure Clinical Records yang memiliki dua kelas yang terdiri dari 203 kelas non deceased dan 96 kelas deceased. Dari hasil penelitian yang dilakukan didapatkan dua buah model yaitu

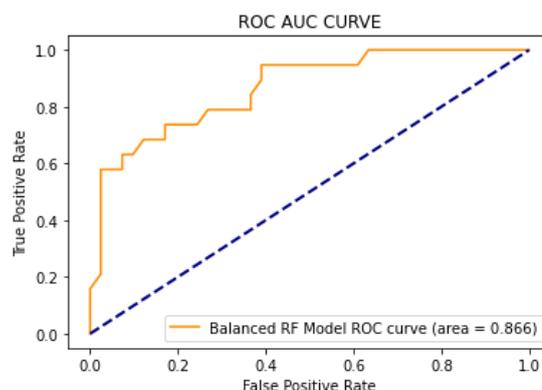
### Klasifikasi *Random Forest*

Klasifikasi dilakukan dengan melibatkan seluruh dataset Heart Failure Clinical Records. Model *Random Forest* yang digunakan menggunakan Hyper-Parameter default berdasarkan dokumentasi dari library *Random Forest* itu sendiri.

Klasifikasi dilakukan dengan melatih model *Random Forest* menggunakan seluruh data training, yaitu sebanyak 180 instance. Kemudian *Random Forest* melakukan *BalancedRandomForestClassifier* yang mana data training diseimbangkan menggunakan Teknik Undersampling. Model *Random Forest* kemudian dievaluasi dengan melakukan prediksi terhadap data testing. Hasil prediksi kemudian akan dimasukkan ke rumus AUC untuk mendapatkan kinerja berupa nilai AUC. Hasil evaluasi berupa nilai performa yang disajikan dalam tabel di bawah ini.

Tabel 1. Hasil evaluasi AUC dengan model *Random Forest* tanpa *Random Search*

Model	Nilai AUC
<i>Random Forest</i>	0,866



Gambar 2. Kurva ROC-AUC Model *Random Forest*

### Klasifikasi *Random Forest* dengan *Random Search*

Setelah melakukan klasifikasi *Random Forest* tanpa Hyper-Parameter Tuning, selanjutnya pada tahap ini dilakukan klasifikasi *Random Forest* dengan Hyper-Parameter Tuning menggunakan teknik *Random Search*.

Setelah dataset dibagi menjadi data training dan data testing, *Random Forest* melakukan *BalancedRandomForestClassifier* yang mana data training diseimbangkan menggunakan Teknik Undersampling. selanjutnya seluruh data training digunakan untuk melakukan hyper-parameter tuning dengan *Random Search*. *Random Search* merupakan teknik hyper-parameter tuning yang memilih konfigurasi berdasarkan ruang hyper-parameter secara acak. Teknik ini sepenuhnya

acak dan tidak menggunakan kecerdasan dalam memilih titik percobaan. *Random Search* akan menghasilkan kandidat Hyper-Parameter sebanyak  $n$  iterasi, kemudian setiap kandidat Hyper-Parameter akan divalidasi dengan aturan cross validation. Hasil validasi dari setiap kandidat hyper-parameter kemudian disimpan. Konfigurasi hyper-parameter optimal dipilih berdasarkan nilai AUC cross validation tertinggi dari semua kandidat yang telah dihasilkan.

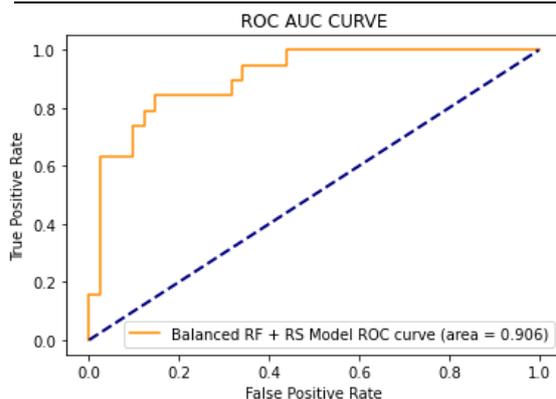
*Random Search* dengan melakukan pencarian secara menyeluruh terhadap Hyper-Parameter yang telah ditetapkan. Nilai hyper-parameter terbaik dari proses hyper-parameter Tuning akan digunakan dalam pembentukan sebuah model.

Setelah divalidasi, konfigurasi Hyper-Parameter dan hasil AUC kemudian disimpan sebagai kandidat. *Random Search* dengan 50 iterasi. Setelah 50 iterasi, kandidat hyper-parameter dengan AUC tertinggi dipilih sebagai konfigurasi hyper-parameter optimal yang kemudian digunakan untuk melatih model *Random Forest* dengan data training penuh.

Untuk nilai objektif yang didapatkan pada proses hyper-parameter tuning menggunakan metode *Random Search* didapatkan kandidat terbaik pada iterasi ke 35 dengan nilai AUC sebesar 0.906 dengan rincian hyper-parameter yaitu,  $n\_estimator = 69$ ,  $max\_depth = 2$ ,  $criterion = gini$ ,  $max\_features = auto$ . Konfigurasi hyper-parameter ini kemudian digunakan untuk final training, yaitu dengan melatih ulang model *Random Forest* dengan melakukan prediksi terhadap data testing. Hasil evaluasi berupa nilai performa yang disajikan dalam tabel di bawah ini.

Tabel 2. Hasil evaluasi AUC dengan model *Random Forest* dengan *Random Search*

Model	Nilai AUC
<i>Random Forest</i> dengan <i>Random Search</i>	0,906



Gambar 3. Kurva ROC-AUC Model *Random Forest* dengan *Random Search*

Untuk mengetahui pengaruh *Hyper-parameter tuning Random Search* pada kinerja *Random Forest* dalam melakukan klasifikasi pada dataset, maka dilakukan perbandingan antara kinerja model *Random Forest* tanpa *Random Search* dengan kinerja model *Random Forest* dengan *Hyper-parameter tuning* dengan menggunakan nilai AUC. dapat dilihat perbandingan anatara kinerja model *Random Forest* tanpa *Random Search* dengan kinerja model *Random Forest* dengan *Random Search* pada Tabel Berikut.

Tabel 3. Hasil Perbandingan

Model	Nilai AUC
<i>Random Forest</i>	0,866
<i>Random Forest</i> dengan <i>Random Search</i>	0,906

Dari hasil perbandingan yang dilakukan, terjadi peningkatan hasil kinerja model sebesar 0,04, sehingga dapat disimpulkan bahwa *Random Search* dapat meningkatkan kinerja model dari *Random Forest* pada dataset Heart Failure Clinical Records.

#### 4. Kesimpulan

Pada penelitian ini mengusulkan penanganan ketidakseimbangan kelas menggunakan *Random Forest* (Pohon Acak) dan peningkatan performa model menggunakan Hyper parameter Tuning *Random Search* (Pencarian Acak) yang kemudian dilakukan pengukuran kinerja model menggunakan AUC. Penelitian dilakukan dengan menggunakan dataset Heart Failure Clinical Records. Untuk split data (pembagian data) dilakukan pembagian sebesar 80% untuk data training (latih) dan 20% untuk data testing (uji). Berdasarkan penelitian yang dilakukan, didapatkan dua model klasifikasi, pertama model *Random Forest* dengan nilai AUC sebesar 0,866, yang kedua model *Random Forest* dengan Hyper parameter Tuning *Random Search* dengan nilai AUC 0,906. Dari kedua model klasifikasi, dapat disimpulkan bahwa penambahan Hyper parameter Tuning *Random Search* memberikan peningkatan hasil kinerja model klasifikasi menggunakan metode *Random Forest* pada prediksi penyakit gagal jantung.

Saran yang diberikan berdasarkan penelitian ini adalah pada penelitian selanjutnya disarankan untuk melakukan seleksi fitur terhadap dataset. Kemudian dapat dicoba melakukan penyeimbangan data terlebih dahulu dengan menggunakan pendekatan level data.

## Referensi

- Annisa, R. (2019). *Analisis Komparasi Algoritma Klasifikasi Data Mining*. 3(1).
- Chicco, D., & Jurman, G. (2020). Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Medical Informatics and Decision Making*, 5, 1–16.
- Erna, I., & Romi Satria, W. (2015). Penggunaan Random Under Sampling untuk Penanganan Ketidakseimbangan Kelas pada Prediksi Cacat Software Berbasis Neural Network. *Journal of Software Engineering*, 1(2), 92–100.
- Fiqriansyah, R., Akbar, F., Andiko, V. C., Ahmad, K. G., Rasywir, E., Meisak, D., Pratama, Y., & Feranika, A. (2022). *Penerapan Algoritma Naïve Bayes Untuk Mengetahui Pasien Penyakit Gagal Jantung Jurnal Informatika Dan Rekayasa Komputer ( JAKAKOM )*. 2(September).
- Firdaus, I. A. (2022). Deteksi Infeksi Mycoplasma Pneumoniae Pneumonia Menggunakan Komparasi Algoritma Klasifikasi Machine Learning. *Journal of Information Technology and Computer Science*, 3(28), 35–42.
- Leonardo, V. A., Christian, Q. L., Jose, G. C., Alexandra, M., & Marcelo, J. (2020). Evaluating hyper-parameter tuning using random search in support vector machines for software effort estimation. *Promise 2020 - Proceedings of the 16th ACM International Conference on Predictive Models and Data Analytics in Software Engineering, Co-Located with Esec/Fse 2020*, 31–40. <https://doi.org/10.1145/3416508.3417121>
- Novaldy, F., & Herliana, A. (2021). *Penerapan Pso Pada Naïve Bayes Untuk*. 3(1), 37–43.
- Nurmasani, A., & Pristyanto, Y. (2021). *Algoritme Stacking Untuk*. VIII.
- Pristyanto, Y. (2019). Penerapan Metode Ensemble Untuk Meningkatkan Kinerja Algoritme Klasifikasi Pada Imbalanced Dataset. *Jurnal Teknoinfo*, 13(1), 11. <https://doi.org/10.33365/jti.v13i1.184>
- Pujianto, U. (2016). Strategi Resampling Berbasis Centroid Untuk Menangani Lunak. *Tekno*, 25(Maret), 1–6.
- Rahayu, S., Purnama, J. J., Pohan, A. B., Nugraha, F. S., & Nurdiani, S. (2020). Prediction Of Survival Of Heart Failure Patients Using. *Jurnal Pilar Nusa Mandiri*, 255–260.
- Syukron, M., Santoso, R., & Widiharih, T. (2020). Perbandingan Metode Smote Random Forest Dan Smote Xgboost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data. *Jurnal Gaussian*, 9, 227–236.
- Valarmathi, R., & Sheela, T. (2021). Heart disease prediction using hyper parameter optimization (HPO) tuning. *Biomedical Signal Processing and Control*, 70(March), 103033. <https://doi.org/10.1016/j.bspc.2021.103033>
- 3
- Wahono, R. S., & Suryana, N. (2013). Combining Particle Swarm Optimization based Feature Selection and Bagging Technique for So ware Defect Prediction. *International Journal of Software Engineering and Its Applications, Vol 7*, 153–166.
- Widjiyati, N. (2021). Implementasi Algoritme Random Forest Pada Klasifikasi Dataset Credit Approval Implementation of Random Forest Algorithm in The Classification of Credit Approval Dataset. *Jurnal Janita Informatika Dan Sistem Informasi*, 1(1), 1–7. <https://doi.org/10.25008/janitra.v1i1.118>
- Zailani, A. U., & Hanun, N. L. (2020). Penerapan Algoritma Klasifikasi Random Forest Untuk Penentuan Kelayakan Pemberian Kredit Di Koperasi Mitra Sejahtera. *Infotech: Journal of Technology Information*, 6(1), 7–14. <https://doi.org/10.37365/jti.v6i1.61>