



## ARTICLE

# Prediksi Stroke Menggunakan Extreme Gradient Boosting

## *Stroke Prediction Using Extreme Gradient Boosting*

Danang Triantoro Murdiansyah\*

Prodi Informatika, Fakultas Informatika, Telkom University, Bandung, Indonesia

\*Penulis Korespondensi: danangtri@telkomuniversity.ac.id

(Disubmit 24-03-06; Diterima 24-03-15; Dipublikasikan online pada 24-09-05)

### Abstrak

Stroke merupakan salah satu penyakit yang paling banyak menyebabkan disabilitas dan kematian pada orang dewasa di seluruh dunia. Salah satu hal yang penting terkait stroke adalah pengobatan dini, sehingga stroke tidak berkembang ke level yang parah pada seseorang. Oleh karena itu prediksi stroke pada seseorang sebelum penyakit tersebut berkembang lebih jauh adalah sangat penting. Penelitian ini berisi prediksi stroke pada seseorang menggunakan algoritma berbasis machine learning, yaitu algoritma Extreme Gradient Boosting, disebut juga dengan XGBoost. Algoritma XGBoost dipilih karena memiliki potensi kemampuan yang baik untuk melakukan prediksi (klasifikasi). XGBoost telah banyak digunakan oleh para peneliti untuk mencapai hasil yang bagus dalam memecahkan berbagai kasus menggunakan machine learning. Pada penelitian ini model machine learning yang dirancang dengan menggunakan XGBoost dibandingkan dengan model machine learning lain yang telah digunakan sebelumnya, yaitu model jenis Stacking, Random Forest, dan Majority Voting. Hasil pengujian menunjukkan XGBoost dapat mencapai performa yang baik dalam seluruh metrik evaluasi, termasuk akurasi yang mendapatkan nilai 95.4%, namun XGBoost pada penelitian ini performanya belum bisa mengungguli Stacking dan Random Forest, yang mana Stacking menempati performa terbaik dengan nilai akurasi 98%.

**Kata kunci:** Stroke; Prediksi; Klasifikasi; XGBoost; Extreme Gradient Boosting

### Abstract

Stroke is one of the diseases that causes the most disability and death of adults in the world. One of the important things of stroke is early treatment, so that the stroke does not progress to a serious level. Therefore, predicting stroke of a person before the disease develops further is very important. This research contains prediction of stroke in a person using a machine learning algorithm, namely the Extreme Gradient Boosting, also called XGBoost. The XGBoost algorithm was chosen because it has good potential for prediction (classification). XGBoost has been widely used by researchers to achieve good results in solving various cases using machine learning. In this research, the machine learning model designed using XGBoost was compared with other machine learning models that had been used previously, namely Stacking, Random Forest and Majority Voting. The test results show that XGBoost can achieve good performance in all evaluation metrics, including accuracy which received a score of 95.4%, but in this study XGBoost's performance could not outperform Stacking and Random Forest, where Stacking has the best performance with an accuracy value of 98%.

**KeyWords:** Stroke; Prediction; Classification; XGBoost. Extreme Gradient Boosting

## 1. Pendahuluan

Penyakit stroke merupakan salah satu penyakit yang paling banyak menyebabkan distabilitas dan kematian pada orang dewasa di seluruh dunia [1]. Untuk di Amerika Serikat, sekitar 800.000 kasus stroke dilaporkan setiap tahun menyebabkan 200.000 kematian, dengan kata lain hampir 1 dari setiap 16 kematian disebabkan oleh stroke [2, 3]. Stroke juga merupakan salah satu penyakit yang membutuhkan biaya pengobatan yang tinggi [4], dengan perkiraan biaya mencapai lebih dari 60 miliar dolar per tahun untuk di Amerika Serikat saja [3]. Penyakit stroke dibagi menjadi 2 jenis, yaitu *ischemic stroke* dan *hemorrhagic stroke*. Lebih dari 80% kasus stroke merupakan *ischemic stroke*, sedangkan sisanya merupakan *hemorrhagic stroke* [2]. *Ischemic stroke* terjadi ketika terjadi sumbatan pada aliran darah ke otak, hal tersebut mengakibatkan kekurangan oksigen dan nutrisi ke jaringan otak. Sedangkan *hemorrhagic stroke* terjadi ketika pembuluh darah di otak pecah dan darah menyebar ke jaringan otak di sekitarnya. Pendarahan tersebut dapat menekan dan merusak sel-sel otak.

Diagnosa stroke secara dini sangatlah penting agar dapat segera mendapatkan perawatan, penundaan atau keterlambatan perawatan dapat memperburuk kerusakan otak, dan memperkecil kemungkinan pemulihan yang baik. Metode diagnosa secara tradisional memiliki keterbatasan dalam hal akurasi dan kecepatan, para peneliti mencari cara lain untuk melakukan diagnosa penyakit ini. Integrasi metode *machine learning* dalam dunia kesehatan merupakan pendekatan yang inovatif, diantaranya untuk prediksi dan diagnosa penyakit. Salah satu algoritma *machine learning* yang dalam penelitian ini mendapat perhatian adalah algoritma *Extreme Gradient Boosting*, atau disingkat dengan *XGBoost*. *XGBoost* telah menunjukkan potensi besar di dalam berbagai bidang, termasuk dalam dunia medis. Sebagai gambaran kemampuan *XGBoost*, dari 29 solusi pemenang kompetisi Kaggle, 17 solusi diantaranya menggunakan *XGBoost* [5]. Model *machine learning* dengan *XGBoost* menawarkan solusi dengan memanfaatkan kumpulan data untuk mengidentifikasi pola yang mengindikasikan stroke, sehingga membantu dalam deteksi dini dan penanganan medis yang tepat. *XGBoost* mampu menangani data yang hilang, menghindari *overfitting*, dan menyesuaikan dengan kumpulan data dinamis. Hal tersebut menjadikan *XGBoost* yang pada penelitian ini merupakan model yang cocok untuk memprediksi stroke yang kompleks.

Penelitian yang telah dilakukan peneliti lain sebelumnya terkait stroke, diantaranya adalah penelitian yang dilakukan oleh Miguel Monteiro dkk dengan papernya yang berjudul “*Using Machine Learning to Improve the Prediction of Functional Outcome in Ischemic Stroke Patients*” pada tahun 2018 [6]. Pada penelitian tersebut di prediksi fungsi organ terkait penyakit *ischemic stroke* pada pasien setelah mendapat perawatan, digunakan *Decision Tree*, *SVM*, *Random Forest*, dan *XGBoost* sebagai *classifier*, dan melibatkan data mRS (*modified Rankin Scale*) untuk penentuan kelas. Pada penelitian yang penulis lakukan tidak melibatkan mRS dan yang di prediksi adalah stroke secara dini, bukan fungsi organ setelah mendapat perawatan / tindakan. Penelitian lainnya yang berhubungan dengan penelitian ini adalah penelitian yang dilakukan oleh Monirul Islam dkk dengan papernya berjudul “*Stroke Prediction Analysis using Machine Learning Classifiers and Feature Technique*” pada tahun 2021 [7]. Pada penelitian tersebut di prediksi keberadaan stroke pada seseorang dengan menggunakan *classifier Naïve Bayes*, *Random Forest*, dan *AdaBoost*. Kemudian berdasarkan paper survei berjudul “*Machine Learning in Action: Stroke Diagnosis and Outcome Prediction*” yang ditulis oleh Shraddha Mainali dkk pada tahun 2021 [8], penelitian yang dilakukan dengan menggunakan *XGBoost* seperti yang penulis lakukan, belum pernah dilakukan oleh peneliti lain. Penelitian yang dilakukan penulis diharapkan dapat menambah kontribusi dalam penanganan penyakit stroke.

### 1.1 Data Penelitian

Data pada penelitian ini berdasarkan dataset dari Kaggle [9]. Dataset yang digunakan terdiri dari 11 atribut, yaitu umur (*age*), jenis kelamin (*gender*), status hipertensi (*hypertension*), status penyakit jantung (*heart disease*), status pernikahan (*ever married*), tipe pekerjaan (*work type*), tipe tempat tinggal (*residence type*), rata-rata kadar gula darah (*avg glucose level*), *index* berat badan (*body mass index*), status merokok (*smoking status*), dan status stroke. 1 atribut sebagai target kelas, yaitu status stroke, sedangkan 10 atribut lainnya sebagai *input* pada model *machine learning* yang dirancang. Pada dataset ini tidak terdapat nilai kosong (*null / missing value*), hanya saja dataset ini merupakan dataset yang tidak seimbang (*imbalance*). Untuk mengatasi data *imbalance*, pada penelitian ini digunakan SMOTE (*Synthetic Minority Over-sampling Technique*) [10, 11]. SMOTE adalah teknik augmentasi data, bekerja dengan menghasilkan data sintetis

dari kelas minoritas. SMOTE membantu menjaga distribusi data seimbang dan dapat meningkatkan kinerja model *machine learning* [10]. Dataset yang digunakan terdiri dari 3426 *record*, dan setelah digunakan SMOTE dataset menjadi terdiri dari 6492 *record*.

## 1.2 Model Machine Learning

Pada penelitian ini digunakan *algoritma Extreme Gradient Boosting* [5], atau dikenal juga dengan nama *XGBoost*. *XGBoost* akan dibandingkan dengan model *machine learning* lain yang sudah digunakan oleh peneliti lain [12] terkait prediksi stroke, yaitu *MVoting* (*Majority Voting*), *RF* (*Random Forest*), dan *Stacking*. Perbandingan tersebut dipilih karena merupakan salah satu penelitian terbaru dalam prediksi penyakit stroke, serta model *MVoting*, *RF*, dan *Stacking* dipilih karena merupakan 3 model yang memberikan hasil terbaik pada penelitian tersebut.

### 1.2.1 Extreme Gradient Boosting

*Extreme Gradient Boosting*, disingkat sebagai *XGBoost*, merupakan algoritma *supervised learning* yang menggunakan pendekatan *ensemble learning* [13], dan berdasarkan algoritma *Gradient Boosting* [6, 7]. Algoritma *Gradient Boosting* menggabungkan banyak pembelajar yang lemah (*weak learner*), pada umumnya berupa *Decision Tree*, untuk membentuk model prediksi yang kuat. *XGBoost* merupakan *tree boosting system* yang sudah banyak digunakan oleh para peneliti untuk menyelesaikan berbagai kasus. *XGBoost* dapat digunakan untuk kasus klasifikasi dan regresi. Kelebihan lain yang dimiliki oleh *XGBoost* diantaranya adalah *regularization*, *tree pruning*, pemrosesan secara paralel, dan penanganan data numerikal serta kategorikal [5].

Langkah-langkah umum dari cara kerja *XGBoost* adalah sebagai berikut [5]. Langkah ke-1 adalah melakukan prediksi awal. Prediksi awal ini merupakan prediksi dengan cara yang sederhana, yaitu dengan menghitung rata-rata dari target. Langkah ke-2 adalah menghitung nilai *error*, yaitu menghitung selisih nilai prediksi dengan nilai aktual. Langkah ke-3 adalah membangun *tree* (*Decision Tree*) untuk melakukan perhitungan yang meminimalkan nilai *error*. Untuk meminimalkan nilai *error* ini pada umumnya digunakan metode *Gradient Descent*. Langkah ke-4 adalah menggunakan metode *ensemble* yang melibatkan prediksi dari *tree* sebelumnya, *learnig rate*, dan prediksi *tree* yang baru. Langkah ini akan menghasilkan prediksi dari metode *ensemble*. Langkah ke-5 yaitu memperbaharui (*update*) hasil prediksi. Pembaruan ini terus dilakukan secara *iteratif* berdasarkan banyak *tree*, atau berdasarkan kriteria pemberhentian. Langkah ke-6 adalah tahap *regularization*. *Regularization* dilakukan untuk mengontrol kompleksitas model dan untuk menghindari *overfitting*. Langkah ke-7 adalah mengulang kembali langkah ke-2 s.d. ke-6 secara terus menerus hingga banyak *tree* yang ditetapkan terbentuk, atau sampai kriteria pemberhentian lain yang ditentukan tercapai. Langkah ke-8 adalah melakukan penentuan hasil prediksi akhir. Prediksi akhir untuk setiap sampel diperoleh dengan menyimpulkan dari seluruh *tree* yang ada pada *ensemble*.

### 1.2.2 Majority Voting

*Majority Voting*, disingkat sebagai *MVoting*, bekerja berdasarkan hasil dari sejumlah model *classifier* yang berbeda. *MVoting* menentukan label kelas yang benar berdasarkan mayoritas suara (hasil) dari model-model *classifier* tersebut. Langkah-langkah umum dari cara kerja *MVoting* adalah sebagai berikut [9, 10]. Pada langkah ke-1 dilakukan tahap pelatihan pada model-model dengan menggunakan data latih. Pada langkah ke-2 model-model melakukan tahap pengujian dengan menggunakan data uji, dan setiap model memberikan prediksi terhadap label kelas. Langkah ke-3 adalah tahap pengambilan keputusan, yaitu hasil dari prediksi setiap model digabungkan, dan label kelas yang dipilih adalah label yang mendapatkan suara terbanyak (mayoritas). Khusus pada penelitian ini, model-model *classifier* yang termasuk dalam *MVoting* adalah *RF* (*Random Forest*) [18], *J48* [19], dan *RepTree* [20].

### 1.2.3 Random Forest

*Random Forest*, atau disingkat dengan *RF*, adalah sebuah algoritma *machine learning* yang merupakan kombinasi dari *prediktor tree* (*Decision Tree*) di mana setiap *tree* bergantung pada nilai dari vektor acak yang diambil secara independen, dengan distribusi yang sama untuk seluruh *tree* dalam *forest* [18]. *RF* merupakan salah satu jenis dari model *ensemble learning*, yaitu model yang menggabungkan beberapa model prediksi, dalam hal ini adalah sejumlah *tree*, untuk meningkatkan kinerja prediksi dibandingkan dengan

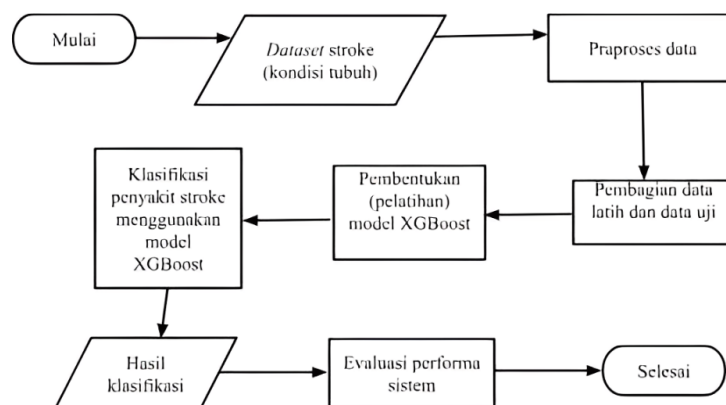
hanya menggunakan 1 model prediksi. Setiap *tree* memberikan hasil klasifikasinya sendiri, kemudian penentuan hasil akhir diperoleh melalui pemungutan suara terbanyak.

#### 1.2.4 Stacking

*Stacking* bekerja dengan pendekatan *ensemble learning* yang memanfaatkan beberapa model *classifier* heterogen yang prediksinya kemudian digabungkan dalam sebuah *meta-classifier* [21]. Pada *Stacking* ada yang disebut *base* model dan *meta-model*. *Base* model dilatih dengan menggunakan data *training*, sedangkan *meta-model* dilatih dengan menggunakan hasil dari *base* model. Dalam penelitian ini, *Stacking* terdiri dari *Naive Bayes*, *Random Forest*, *RepTree*, dan *J48*. Model-model tersebut berlaku sebagai *base* model, yang hasil prediksinya digunakan untuk melatih *Logistic Regression* yang berlaku sebagai *meta-model*.

## 2. Metode

Tahapan – tahapan dari sistem prediksi stroke pada penelitian ini, yang dibangun dengan *algoritma machine learning XGBoost*, ditunjukkan oleh diagram alir pada Gambar 1. Input dari sistem prediksi stroke yang dibangun pada penelitian ini adalah data kondisi tubuh. Sumber dataset stroke tersebut bersumber dari Kaggle [9]. Sebagaimana dijelaskan mengenai data penelitian, dataset stroke terdiri dari 11 atribut, yang mana sebagai target kelas adalah status stroke. Dataset tersebut diolah terlebih dahulu pada tahap praproses data, sebelum dilakukan pembagian data latih (*training*) dan data uji (*testing*). Pada tahap praproses data, dilakukan proses SMOTE (*Synthetic Minority Over-sampling Technique*) [10] untuk menangani data *imbalance*. Setelah proses SMOTE, dilakukan normalisasi data. Normalisasi data dibutuhkan karena setiap atribut memiliki satuan dan juga skala yang berbeda. Sampai proses normalisasi, maka tahap praproses data selesai. Selanjutnya dilakukan pembagian data latih dan data uji. Pembagian data latih serta data uji dilakukan dengan menggunakan *stratify* [22]. Dengan menggunakan *stratify* dihasilkan data latih sebanyak 4869 *record* (75%), dan data uji sebanyak 1623 *record* (25%). Kemudian berlanjut ke proses pembentukan (pelatihan) model *XGBoost* dengan menggunakan data latih. Dalam pembentukan model *XGBoost*, nilai parameter-parameter *XGBoost* ditentukan dengan menggunakan *Bayesian Optimization* [23]. Setelah model *XGBoost* terbentuk, dilakukan klasifikasi penyakit stroke menggunakan model *XGBoost* tersebut. Proses klasifikasi akan menghasilkan 2 kelas, yaitu seseorang dinilai mengidap stroke dan tidak mengidap stroke. Kemudian hasil klasifikasi tersebut akan dievaluasi untuk mengetahui performa sistem. Evaluasi dilakukan dengan menggunakan metrik *recall*, *precision*, *F1-Measure*, *AUC (Area Under The ROC Curve)*, dan akurasi.



Gambar 1. Diagram Alir Sistem Prediksi Stroke

## 3. Hasil

Pada penelitian ini model *machine learning* yang dirancang diimplementasikan dengan menggunakan bahasa pemrograman Python 3 dan *library xgboost 2.0.0*. Model *machine learning XGBoost* yang dirancang pada penelitian ini, performanya akan dibandingkan dengan performa model lain yang telah dilakukan oleh peneliti lain sebelumnya [12]. Model pembandingan tersebut adalah *MVoting (Majority Voting)*, *RF*

(*Random Forest*), dan *Stacking*. Penelitian tersebut [12] dipilih sebagai pembanding karena merupakan salah satu penelitian terbaru dalam prediksi penyakit stroke, dan model *MVoting*, RF, dan *Stacking* dipilih karena merupakan 3 model yang memberikan hasil terbaik pada penelitian tersebut.

Pada penelitian ini nilai dari parameter yang ada pada *XGBoost* dioptimasi dengan menggunakan *Bayesian Optimization* [23]. Dengan menggunakan *Bayesian Optimization* dicari nilai optimal dari parameter-parameter, diantaranya adalah parameter kedalaman maksimum (*max depth*) dengan ruang pencarian antara 2 s.d. 20, parameter kecepatan belajar (*learning rate*) antara 0.05 s.d. 0.4, dan parameter subsampel (*subsample*) antara 0.5 s.d. 0.9. Untuk implementasi *Bayesian Optimization* digunakan *library Hyperopt*. Dari proses optimasi tersebut didapat nilai parameter yang menghasilkan performa terbaik, yaitu pada kedalaman maksimum bernilai 18, kecepatan belajar bernilai 0.381138230167329, dan subsampel bernilai 0.6174783474555894. Sebelumnya dataset telah diolah dengan menggunakan menggunakan metode SMOTE [2, 3] dan *stratify* [22]. Penggunaan *stratify* dapat membantu mengatasi *overfitting* dan *underfitting*, sedangkan SMOTE digunakan untuk mengatasi data *imbalance*. Pada Tabel 1 dapat dilihat performa dari model *XGBoost* yang dirancang, dan perbandingannya dengan model lain. Angka yang dicetak tebal menunjukkan angka yang terbesar daripada angka lainnya dalam 1 kolom.

Tabel 1. Performa Model *Machine Learning*

Model Machine Learning	Precision	Recall	F-Measure	AUC	Akurasi
XGBoost	0.943	0.966	0.954	0.954	0.954
MVoting [12]	0.93	0.93	0.93	0.93	0.93
RF [12]	0.966	0.966	0.966	0.986	0.97
Stacking [12]	<b>0.974</b>	<b>0.974</b>	<b>0.974</b>	<b>0.989</b>	<b>0.98</b>

#### 4. Pembahasan

Berdasarkan eksperimen yang hasilnya dirangkum dalam Tabel 1, model *Stacking* unggul dari semua model lainnya dalam semua metrik evaluasi (*precision*, *recall*, *F-Measure*, *AUC*, akurasi). Sedangkan performa *XGBoost* secara umum berada di bawah RF dan *Stacking*, kecuali dalam metrik *recall* saja yang menyamai RF, yaitu bernilai 0.966. Untuk seluruh metrik evaluasi, *XGBoost* hanya unggul dari *MVoting*. Jika dilihat dari metrik AUC, model RF dan *Stacking* memiliki nilai AUC yang hampir sama, sedangkan *XGBoost* terpaut 0.035 lebih kecil dari *Stacking*, yang mana *Stacking* memiliki nilai AUC tertinggi dengan nilai 0.989. Dengan hasil tersebut model RF dan *Stacking* artinya memiliki kemampuan diskriminasi yang hampir sama, yaitu kemampuan untuk membedakan pasien stroke dari pasien yang tidak stroke. *XGBoost* berada di urutan ke-3 dalam hal kemampuan tersebut.

Seluruh model, yaitu *XGBoost*, *MVoting*, RF, dan *Stacking*, merupakan model *ensemble*, yang berarti setiap model terdiri dari beberapa model di dalamnya. *XGBoost* terdiri dari model-model *Decision Tree*. *MVoting* untuk kasus ini [12], terdiri dari RF, J48, dan *RepTree*. RF [12] terdiri dari model-model *Decision Tree*. Salah satu perbedaan RF dan *XGBoost* adalah RF membangun *tree* secara independen dengan fitur dan subset data random, sedangkan *XGBoost* membangun *tree* secara *sekuensial*, *tree* baru dibentuk berdasarkan *tree* sebelumnya dengan perhitungan untuk memperkecil *error*. *Stacking* untuk kasus ini [12], terdiri dari *Naïve Bayes*, RF, J48, *RepTree*, dan *Logistic Regression*. Dengan komposisi demikian pada model-model tersebut, dapat disimpulkan bahwa *Stacking* memiliki komposisi model paling lengkap, yaitu melibatkan model model yang mengandung jenis *Bayesian* [24], *Decision Tree* [25], dan *Logistic Regression* [26]. Dengan komposisi model paling lengkap, *Stacking* dapat menangani segala kondisi dataset dengan paling baik [27], sehingga pada kasus penelitian ini wajar jika *Stacking* dapat mencapai nilai metrik evaluasi terbaik sebagaimana terlihat pada Tabel 1.

Pada penelitian ini, *XGBoost* dan RF sama-sama berbasis *Decision Tree*, namun secara umum *XGBoost* berada di bawah RF performanya. Hal tersebut kemungkinan besar karena *XGBoost* membutuhkan *hyperparameter tuning* dengan benar-benar baik, sebab *XGBoost* memiliki parameter cukup banyak [28], sedangkan RF memiliki parameter yang lebih sedikit sehingga lebih mudah dilakukan *hyperparameter tuning* untuk mendapatkan hasil yang optimal. Pada penelitian ini *hyperparameter tuning XGBoost* dilakukan dengan

menggunakan *Bayesian Optimization* [23]. Opsi lain untuk melakukan *hyperparameter tuning XGBoost* di antaranya adalah dengan menggunakan *Grid Search*, namun berdasarkan penelitian sebelumnya [29], *Bayesian Optimization* dapat memberikan hasil yang lebih baik daripada *Grid Search*, oleh sebabnya *Bayesian Optimization* dipilih dalam penelitian ini. Dalam menggunakan *Bayesian Optimization* peneliti harus bisa memberikan ruang pencarian nilai parameter yang tepat untuk mendapatkan hasil yang optimal, namun memberikan ruang pencarian tersebut termasuk permasalahan yang tidak mudah [30]. Oleh karena itu *hyperparameter tuning XGBoost* memerlukan kajian tersendiri untuk kasus pada penelitian ini, dan merupakan permasalahan yang rencananya akan diselesaikan pada penelitian di masa mendatang.

## 5. Simpulan

Stroke merupakan penyakit yang sangat penting untuk dideteksi secara dini agar dapat segera dilakukan tindak lanjut, dan tidak berkembang ke tahap yang parah. Pada penelitian ini disajikan prediksi penyakit stroke pada seseorang dengan menggunakan pendekatan *machine learning*. Model *machine learning XGBoost* yang dirancang pada penelitian ini berhasil memberikan hasil yang baik, yaitu *precision* dengan nilai 94.3%, *recall* dengan nilai 96.6%, *F-Measure* dengan nilai 95.4%, *AUC* dengan nilai 95.4%, dan akurasi dengan nilai 95.4%. Walaupun *XGBoost* berhasil mencapai hasil yang baik, namun belum bisa mengungguli model *Stacking* dan *RF* yang telah digunakan peneliti lain sebelumnya. Pada penelitian ini digunakan *SMOTE* untuk menangani data *imbalance* dan *Bayesian Optimization* untuk optimasi nilai *hyperparameter*. Peneliti pada penelitian ini melihat bahwa *XGBoost* masih memiliki potensi untuk ditingkatkan performanya. Walaupun menggunakan *Bayesian Optimization* untuk optimasi nilai *hyperparameter*, peneliti pada penelitian ini melihat optimasi nilai *hyperparameter* masih dapat ditingkatkan sehingga memberikan hasil yang lebih baik. Hal tersebut karena ruang pencarian solusi dari nilai *hyperparameter* yang luas. Perlu dilakukan kajian tersendiri untuk melakukan optimasi *hyperparameter* pada kajian penelitian ini, yang mana hal tersebut dapat menjadi kesempatan permasalahan pada kajian penelitian berikutnya.

## Pustaka

- [1] W. Wang et al., "A systematic review of machine learning models for predicting outcomes of stroke with structured data," *PLOS ONE*, vol. 15, no. 6, p. e0234722, Jun. 2020, doi: 10.1371/journal.pone.0234722.
- [2] K. A. Blackham et al., "Endovascular therapy of acute ischemic stroke: report of the Standards of Practice Committee of the Society of NeuroInterventional Surgery," *Journal of NeuroInterventional Surgery*, vol. 4, no. 2, pp. 87–93, Mar. 2012, doi: 10.1136/neurintsurg-2011-010243.
- [3] "Heart Disease and Stroke Statistics—2023 Update: A Report From the American Heart Association | *Circulation*." Accessed: Feb. 29, 2024. [Online]. Available: <https://www.ahajournals.org/doi/full/10.1161/CIR.0000000000001123>
- [4] A. S. Ferrell and G. W. Britz, "Developments on the horizon in the treatment of neurovascular problems," *Surg Neurol Int*, vol. 4, no. Suppl 1, pp. S31–S37, Mar. 2013, doi: 10.4103/2152-7806.109194.
- [5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in *KDD '16*. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [6] M. Monteiro et al., "Using Machine Learning to Improve the Prediction of Functional Outcome in Ischemic Stroke Patients," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1953–1959, Nov. 2018, doi: 10.1109/TCBB.2018.2811471.
- [7] M. M. Islam, S. Akter, M. Rokunojjaman, J. H. Rony, A. Amin, and S. Kar, "Stroke Prediction Analysis using Machine Learning Classifiers and Feature Technique," *International Journal of Electronics and Communications Systems*, vol. 1, no. 2, Art. no. 2, Dec. 2021, doi: 10.24042/ijecs.v1i2.10393.
- [8] S. Mainali, M. E. Darsie, and K. S. Smetana, "Machine Learning in Action: Stroke Diagnosis and Outcome Prediction," *Frontiers in Neurology*, vol. 12, 2021, Accessed: Feb. 29, 2024. [Online]. Available: <https://www.frontiersin.org/journals/neurology/articles/10.3389/fneur.2021.734345>

- [9] "Stroke Prediction Dataset." Accessed: Jan. 28, 2024. [Online]. Available: <https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>
- [10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.
- [11] G. A. Pradipta, R. Wardoyo, A. Musdholifah, I. N. H. Sanjaya, and M. Ismail, "SMOTE for Handling Imbalanced Data Problem : A Review," in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, Nov. 2021, pp. 1–8. doi: 10.1109/ICIC54025.2021.9632912.
- [12] E. Dritsas and M. Trigka, "Stroke Risk Prediction with Machine Learning Techniques," *Sensors (Basel)*, vol. 22, no. 13, p. 4670, Jun. 2022, doi: 10.3390/s22134670.
- [13] T. N. Rincy and R. Gupta, "Ensemble Learning Techniques and its Efficiency in Machine Learning: A Survey," in *2nd International Conference on Data, Engineering and Applications (IDEA)*, Feb. 2020, pp. 1–6. doi: 10.1109/IDEA49133.2020.9170675.
- [14] V. K. Ayyadevara, "Gradient Boosting Machine," in *Pro Machine Learning Algorithms: A Hands-On Approach to Implementing Algorithms in Python and R*, V. K. Ayyadevara, Ed., Berkeley, CA: Apress, 2018, pp. 117–134. doi: 10.1007/978-1-4842-3564-5\_6.
- [15] G. Biau and B. Cadre, "Optimization by Gradient Boosting," in *Advances in Contemporary Statistics and Econometrics: Festschrift in Honor of Christine Thomas-Agnan*, A. Daouia and A. Ruiz-Gazen, Eds., Cham: Springer International Publishing, 2021, pp. 23–44. doi: 10.1007/978-3-030-73249-3\_2.
- [16] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information Fusion*, vol. 6, no. 1, pp. 63–81, Mar. 2005, doi: 10.1016/j.inffus.2004.04.008.
- [17] G. Seni and J. Elder, *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Morgan & Claypool Publishers, 2010.
- [18] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [19] P. Pandey and R. Prabhakar, "An analysis of machine learning techniques (J48 & AdaBoost)-for classification," in *2016 1st India International Conference on Information Processing (IICIP)*, Aug. 2016, pp. 1–6. doi: 10.1109/IICIP.2016.7975394.
- [20] S. Chandra and S. Maheshkar, "Verification of static signature pattern based on random subspace, REP tree and bagging," *Multimed Tools Appl*, vol. 76, no. 18, pp. 19139–19171, Sep. 2017, doi: 10.1007/s11042-017-4531-2.
- [21] B. Pavlyshenko, "Using Stacking Approaches for Machine Learning Models," in *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, Aug. 2018, pp. 255–258. doi: 10.1109/DSMP.2018.8478522.
- [22] X. Zhao, J. Liang, and C. Dang, "A stratified sampling based clustering algorithm for large-scale data," *Knowledge-Based Systems*, vol. 163, pp. 416–428, Jan. 2019, doi: 10.1016/j.knosys.2018.09.007.
- [23] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimizationb," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, Mar. 2019, doi: 10.11989/JEST.1674-862X.80904120.
- [24] H. K. S. Tayal, P. M. George, P. Singla, and U. Kose, *Bayesian Reasoning and Gaussian Processes for Machine Learning Applications*. CRC Press, 2022.
- [25] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, Art. no. 01, Mar. 2021, doi: 10.38094/jastt20165.

- [26] L. M. Gladence, M. Karthi, and V. M. Anu, “A Statistical Comparison of Logistic Regression and Different Bayes Classification Methods for Machine Learning,” vol. 10, no. 14, 2015.
- [27] A. Rácz, D. Bajusz, and K. Héberger, “Multi-Level Comparison of Machine Learning Classifiers and Their Performance Metrics,” *Molecules*, vol. 24, no. 15, Art. no. 15, Jan. 2019, doi: 10.3390/molecules24152811.
- [28] S. Putatunda and K. Rama, “A Comparative Analysis of Hyperopt as Against Other Approaches for Hyper-Parameter Optimization of XGBoost,” in *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning*, in SPML '18. New York, NY, USA: Association for Computing Machinery, Nov. 2018, pp. 6–10. doi: 10.1145/3297067.3297080.
- [29] L. Sun, “Application and Improvement of Xgboost Algorithm Based on Multiple Parameter Optimization Strategy,” in *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Dec. 2020, pp. 1822–1825. doi: 10.1109/ICMCCE51767.2020.00400.
- [30] D. Eriksson and M. Poloczek, “Scalable Constrained Bayesian Optimization,” in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, PMLR, Mar. 2021, pp. 730–738. Accessed: Feb. 19, 2024. [Online]. Available: <https://proceedings.mlr.press/v130/eriksson21a.html>