**A picture containing clipart

Description automatically generated**

**UNIVERSITAS INDONESIA**

***HYPERPARAMETER TUNING* PADA MODEL *EXTREME GRADIENT BOOSTING* UNTUK ESTIMASI CURAH HUJAN: STUDI KASUS KOTA PONTIANAK**

**PROPOSAL PENELITIAN TUGAS AKHIR S1**

**AURIWAN YASPER**

**1706032004**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**PROGRAM STUDI FISIKA**

**DEPOK**

**2023**

## HALAMAN PENGESAHAN

|  |  |
| --- | --- |
| Nama | : Auriwan Yasper |
| NPM | : 1706032004 |
| Program Studi | : Fisika |
| Judul Skripsi | : *Hyperparameter Tuning* pada *eXtreme Gradient Boosting* untuk estimasi curah hujan: Studi kasus Kota Pontianak |

Depok, 26 Februari 2023

A picture containing letter

Description automatically generatedDiajukan oleh

Auriwan Yasper

1706032004

Menyetujui

|  |  |
| --- | --- |
| Pembimbing I    Dr. Djati Handoko, M.Si  NIP/NUP: 1976902052008121001 | Pembimbing II  Maulana Putra, S.Si, M.T.  NIP/NUP: 198611272008011001 |

## DAFTAR ISI

[HALAMAN PENGESAHAN ii](#_Toc128284597)

[DAFTAR ISI iii](#_Toc128284598)

[DAFTAR GAMBAR iv](#_Toc128284599)

[DAFTAR TABEL v](#_Toc128284600)

[BAB I PENDAHULUAN 1](#_Toc128284601)

[1.1 Latar Belakang 1](#_Toc128284602)

[1.1 Perumusan Masalah 2](#_Toc128284603)

[1.2 Tujuan Penelitian 2](#_Toc128284604)

[1.3 Manfaat Penelitian 3](#_Toc128284605)

[1.4 Batasan Penelitian 3](#_Toc128284606)

[1.5 Sistematika Penulisan 3](#_Toc128284607)

[BAB 2 TINJAUAN PUSTAKA 4](#_Toc128284608)

[3.1 Landasan Teori 4](#_Toc128284609)

[3.2 Penelitian Terkait 11](#_Toc128284610)

[BAB 3 METODE PENELITIAN 33](#_Toc128284611)

[3.1 Alur Penelitian 33](#_Toc128284612)

[3.2 Tahapan Penelitian 34](#_Toc128284613)

[3.3 Tempat Penelitian 36](#_Toc128284614)

[3.4 Jadwal Penelitian 36](#_Toc128284615)

[3.5 Kesimpulan 37](#_Toc128284616)

[DAFTAR PUSTAKA 38](#_Toc128284617)

## DAFTAR GAMBAR

[Gambar 2. 1 Algoritma Gradient Boosting (Jolly, 2018) 6](#_Toc128292609)

[Gambar 2. 2 Perbedaan grid search dan random search (Deshpande & Kumar, 2018). 10](#_Toc128292610)

[Gambar 2. 3 Teknik Coarse to fine pada random search (Deshpande & Kumar, 2018). 11](#_Toc128292611)

[Gambar 2. 4 Hasil pelatihan dan tes RMSE selama iterasi (Anwar, Winarno, Hadikurniawati, & Novita, 2020). 12](#_Toc128292612)

[Gambar 2. 5 Ranking masing-masing atribut (Anwar, Winarno, Hadikurniawati, & Novita, 2020) 13](#_Toc128292613)

[Gambar 2. 6 RR prediksi versus RR sebenarnya (Anwar, Winarno, Hadikurniawati, & Novita, 2020) 14](#_Toc128292614)

[Gambar 2. 7 Auto Correlation Function plot untuk data training (R & I G, 2019) 15](#_Toc128292615)

[Gambar 2. 8 Prediksi versus data aktual (R & I G, 2019) 16](#_Toc128292616)

[Gambar 2. 9 Grafik prediksi vs aktual pada data tes (R & I G, 2019). 17](#_Toc128292617)

[Gambar 2. 10 Skor kepentingan fitur dataset (Ma, et al., 2020) 19](#_Toc128292618)

[Gambar 2. 11 Hasil penyetelan hyperparameter untuk XGBoost dan RF dievaluasi oleh . Proses penyetelan (a) N estimator di XGBoost dan RF, (b) max depth dan minimum child weight di XGBoost, dan (c) maximum depth dan minimum samples split dari Random Forest. (d) Peringkat kepentingan fitur diperoleh dari model XGBoost terlatih. Skor kepentingan dinormalisasi dengan membagi skor maksimum, dan skor maksimum ditetapkan menjadi 100 (Li, Shan, & Shek, 2022). 31](#_Toc128292619)

[Gambar 3. 1 Diagram alur penelitian 33](#_Toc128292646)

## DAFTAR TABEL

[Table 2. 1 Fitur yang ada pada data cuaca (Anwar, Winarno, Hadikurniawati, & Novita, 2020) 11](#_Toc128284630)

[Table 2. 2 Pelatihan RMSE dengan n\_rounds (n\_estimators) = 100, n\_fold = 10, eta (learning\_rate) = 0.3, max\_depth = 6 (Anwar, Winarno, Hadikurniawati, & Novita, 2020). 13](#_Toc128284631)

[Table 2. 3 Perbandingan masing-masing model (R & I G, 2019). 16](#_Toc128284632)

[Table 2. 4 Dataset yang digunakan dalam penelitian (Ma, et al., 2020). 18](#_Toc128284633)

[Table 2. 5 Tabel evaluasi model (Ma, et al., 2020) 19](#_Toc128284634)

[Table 2. 6 Kombinasi ruang pencarian dan jumlah komputasi masing-masing metode tuning (Qin, et al., 2021) 21](#_Toc128284635)

[Table 2. 7 Parameter hasil tuning yang akan digunakan dalam penelitian (Qin, et al., 2021) 21](#_Toc128284636)

[Table 2. 8 Tabel hasil performa model untuk dataset austria (Qin, et al., 2021) 22](#_Toc128284637)

[Table 2. 9 Pemilihan tuning hyperparameter (Qin, et al., 2021) 24](#_Toc128284638)

[Table 2. 10 hasil tuning hyperparameter dan hasil uji AUC (Hasan, et al., 2021) 25](#_Toc128284639)

[Table 2. 11 perbandingan performa masing-masing model untuk dataset kekuatan tekan (Nguyen, Vu, Vo, & Thai, 2021) 27](#_Toc128284640)

[Table 2. 12 perbandingan performa masing-masing model untuk dataset kekuatan tarik (Nguyen, Vu, Vo, & Thai, 2021) 28](#_Toc128284641)

[Table 2. 13 hasil tuning hyperparameter optimal (Kaushik & Birok, 2021) 29](#_Toc128284642)

[Table 2. 14 Perbandingan performa (Dalal, Seth, Radulescu, Secara, & Tolea, 2022) 32](#_Toc128284643)

[Tabel 3. 1 Tabel agenda dan jadwal penelitian 36](#_Toc128191001)

## BAB 1 PENDAHULUAN

### Latar Belakang

Curah hujan yang tinggi sangat mempengaruhi kehidupan manusia di berbagai sektor antara lain pertanian, transportasi, dan juga dapat mengakibatkan bencana alam seperti kekeringan, banjir, dan tanah longsor (Anwar, Winarno, Hadikurniawati, & Novita, 2020). Peningkatan rata-rata curah hujan secara signifikan menurunkan tingkat kerawanan pangan dalam aktivitas pertanian di suatu daerah, terutama pedesaan (Tankari, 2020). Hal ini menimbulkan dampak positif bagi para petani dalam hal kesuburan lahan pertanian mereka. Berbeda dengan perkotaan yang lebih beresiko terkena bencana banjir. Begitu juga dengan daerah perbukitan, curah hujan sangat berpengaruh terhadap bencana tanah longsor. Menurut data EM-DAT, sejak tahun 1908 hingga 2022, tanah longsor telah menyebabkan kerusakan besar dalam masyarakat dan telah menelan korban jiwa sebesar 67.169 dan kerugian ekonomi lebih dari 11 miliar dolar (Pham, Kim, Le, & Choi, 2022). Oleh karena itu, perkiraan cuaca sangat penting untuk menangani bencana yang mungkin akan terjadi.

Perkembangan teknologi dengan sistem yang dapat menganalisa data curah hujan dan melakukan perkiraan cuaca memberikan solusi yang tepat untuk mengatasi berbagai dampak negatif yang dihasilkan dari curah hujan. Algoritma *machine learning* merupakan salah satu teknologi alternatif yang dapat diterapkan. Cukup banyak model *machine learning* yang digunakan dalam mengolah data tersebut untuk mendapatkan prediksi yang akurat akan perkiraan hujan dimasa mendatang, diantaranya *Bayesian methode*, SVM (*Support Vector Machine*), ANN (*Artificial Neural Network*), atau variasinya seperti *Support Vector Regression* (SVR) yang memiliki performa bagus dalam melakukan prediksi curah hujan (Xiang, Gou, He, xia, & Wang, 2018). Pada penelitian ini, model yang dipilih merupakan model yang cukup populer dalam prediksi dengan set data historis yaitu eX*treme Gradient Boosting*.

Untuk memperoleh prediksi dengan performa yang bagus, perlu adanya pengaturan perilaku dari model *machine learning*. Dalam pembelajaran mesin, terdapat sebuah nilai yang dapat mengontrol perilaku dari model yaitu *parameter* dan *hyperparameter*. Selama proses pembelajaran algoritma pembelajaran mempelajari pola-pola pada data dan selalu memperbaharui nilai yang mengontrol perilaku tersebut, nilai inilah yang disebut *parameter*. Setelah model berhasil mempelajari pola data, nilai parameter sudah menjadi bagian dari model tersebut dan tidak dapat diubah diluar pembelajaran. Sebagai contoh, *weight* dan *bias* dalam *neural network* merupakan parameter. Sebaliknya, *hyperparameter* digunakan hanya pada algoritma yang spesifik, dan nilainya diatur sebelum model mempelajari pola data. Nilai dari *hyperparameter* ini akan digunakan untuk memperbaharui *parameter* yang ada pada model selama proses pembelajaran. Sehingga nilai *hyperparameter* sangat mempengaruhi nilai parameter yang terdapat dalam suatu model pembelajaran mesin.

Penyesuaian *hyperparameter* bertujuan untuk mencari kumpulan nilai yang tepat untuk mengoptimalkan algoritma pembelajaran, sambil mengevaluasi nilai parameter yang disesuaikan hingga memperoleh nilai yang optimal. Kumpulan nilai *hyperparameter* tersebut ditujukan untuk mengoptimalkan kinerja model, mengurangi fungsi *loss* dan untuk mendapatkan hasil yang terbaik dengan *error* yang lebih sedikit. Perlu diperhatikan bahwa algoritma pembelajaran mengoptimalkan fungsi *loss* berdasarkan data *input* dan mencari solusi terbaik dalam pengaturan yang diberikan dan disini *hyperparameter* akan mencari kombinasi secara tepat. (Navas, Anyscale Blog, 2022).

### Perumusan Masalah

1. Bagaimana cara membuat estimasi curah hujan menggunakan model *eXtreme Gradient Boosting*?
2. Bagaimana cara mendapatkan kombinasi *hyperparameter* yang terbaik untuk model *eXtreme Gradient Boosting* dari data curah hujan?

### Tujuan Penelitian

1. Membuat estimasi curah hujan menggunakan model *eXtreme Gradient Boosting.*
2. Mendapatkan nilai terbaik dalam penyesuaian kombinasi *hyperparameter* yang tepat untuk memaksimalkan model *Extreme Gradient Boosting*.

### Manfaat Penelitian

Harapannya penelitian ini dapat menjadi sumber acuan dalam pemilihan *hyperparameter* model *eXtreme Gradient Boosting* untuk sistem estimasi curah hujan yang digunakan pada institusi terkait, serta berapa saja kombinasi penyesuaian *hyperparameter* *eXtreme Gradient Boosting* yang memiliki performa bagus dalam melakukan estimasi curah hujan.

### Batasan Penelitian

1. *Hyperparameter* yang akan di-*tuning* adalah *max\_depth, learning\_rate,* dan *n\_estimator* dengan fokus pada mencari nilai *learning\_rate* terbaik*.*
2. Data yang digunakan sebagai target adalah data klasifikasi curah hujan.
3. Data yang digunakan sebagai fitur adalah temperatur, kelembapan, kecepatan angin, dan tekanan udara Kota Pontianak.

### Sistematika Penulisan

Sistematika penulisan naskah penelitian terdiri dari lima bab, diantaranya:

|  |  |
| --- | --- |
| **BAB I** | **PENDAHULUAN**  Membahas latarbelakang, perumusan masalah, tujuan penelitian, manfaat penelitian, dan batasan masalah. |
| **BAB II** | **TINJAUAN PUSTAKA**  Berisi literatur mengenai landasan teori yang digunakan sebagai acuan pada penelitian, dan penelitian terkait yang sudah pernah dilakukan. |
| **BAB III** | **METODE PENELITIAN**  Berisi tentang alur penelitian, tahapan, dan metode apa saja yang digunakan dalam penelitian ini. |
| **BAB IV** | **PEMBAHASAN**  Berisi data-data yang diperoleh dan analisis terhadap hasil penelitian. |
| **BAB V** | **KESIMPULAN DAN SARAN**  Berisi tentang kesimpulan dan saran untuk penelitain selanjutnya. |

## BAB 2 TINJAUAN PUSTAKA

### Landasan Teori

#### *Machine Learning*

*Machine Learning* adalah sekumpulan metode yang dapat mendeteksi pola dalam data dan menggunakan pola tersebut untuk membuat prediksi di masa mendatang. Pembelajaran mesin telah mendapatkan nilai yang sangat besar di berbagai industri, mulai dari keuangan hingga perawatan kesehatan (Jolly, 2018). Istilah *Machine Learning* pertama kali dipopulerkan oleh Arthur Samuel pada tahun 1959. Arthur mengatakan bahwa *machine learning* merupakan cabang ilmu yang mempelajari bagaimana komputer dapat belajar sendiri tanpa harus diprogram secara explisit.

Secara garis besar, *machine learning* dapat dikategorikan menjadi tiga jenis utama:

* *Supervised Learning*: yaitu bentuk pembelajaran mesin di mana data kita dilengkapi dengan sekumpulan label atau variabel target yang berupa angka. Label/kategori ini biasanya milik satu fitur/atribut, yang umumnya dikenal sebagai variabel target. Misalnya, setiap baris data Anda dapat termasuk dalam kategori Sehat atau Tidak Sehat.
* *Unsupervised Learning*: yaitu salah satu bentuk pembelajaran mesin di mana algoritma mencoba mendeteksi/menemukan pola dalam data yang tidak memiliki variabel hasil/target. Dengan kata lain, kita tidak memiliki data yang disertakan dengan label yang sudah ada sebelumnya. Dengan demikian, algoritma biasanya akan menggunakan metrik seperti jarak untuk mengelompokkan data bersama-sama, bergantung pada seberapa jauh mereka satu sama lain.
* *Reinforcement Learning*: *Reinforcement Learning* didasarkan pada teori psikologi, yang dijabarkan setelah serangkaian percobaan yang dilakukan pada hewan. Tahap pertama adalah mendefinisikan tujuan yang ingin dicapai, *Reinforcement Learning* mencoba memaksimalkan imbalan yang diterima untuk pelaksanaan tindakan atau serangkaian tindakan yang memungkinkan kita mencapai tujuan yang telah ditentukan. *Reinforcement Learning* adalah sektor pembelajaran mesin yang sangat menarik, digunakan dalam segala hal mulai dari mobil otonom hingga video game. Ini bertujuan untuk membuat algoritma yang dapat belajar dan beradaptasi dengan perubahan lingkungan.

#### *Gradient Boosting*

Algoritma yang menggunakan teknik *boosting* akan membangun model dari data latih dan membuat model tambahan yang berfungsi untuk memperbaiki kesalahan dari model pertama. Model-model ini terus ditambahkan hingga data latih terprediksi dengan baik atau telah mencapai batas jumlah maksimal model yang ditentukan.

Algoritma *boosting* berfungsi untuk meningkatkan performa atau akurasi prediksi dengan menggabungkan beberapa model sederhana yang dianggap lemah (*weak learners*) menjadi sebuah model yang kuat (*strong ensemble learner*). Asal-usul dari algoritma ini berasal dari algoritma sederhana seperti regresi linier dan decision tree yang dimodifikasi untuk meningkatkan performa. Terdapat dua metode dalam algoritma boosting yaitu Adaptive boosting dan Gradient boosting, yang masing-masing memiliki cara untuk memperbaiki kesalahan pada model sebelumnya. Nama "boosting" merujuk pada fungsinya untuk meningkatkan performa atau akurasi prediksi (Kurikulum Developer Dicoding, 2023).

Algoritma *Adaptive boosting* dibangun untuk mengatasi masalah klasifikasi, sedangkan *Gradient boosting* dapat digunakan untuk mengatasi masalah regresi dan klasifikasi (Tattar, 2018). Oleh karena itu pada penelitian kali ini, model yang akan digunakan adalah salah satu pengembangan dari *gradient boosting* yaitu *eXtreme Gradient Boosting*.

*Gradient* dalam *Gradient Boosting* mengacu pada perbedaan antara nilai aktual dan prediksi, dan *Boosting* mengacu pada peningkatan, yaitu, memperbaiki error pada iterasi yang berbeda (Ciaburro, Ayyadevara, & Perrier, 2018). Pengklasifikasian pada model *Adaptive Boosting,* bobot ditambahkan ke contoh yang diprediksi oleh model dengan benar. Namun, dalam *Gradient Boosting*, kesalahan residual juga digunakan sebagai label di setiap pohon untuk membuat prediksi di masa mendatang. Konsep ini bisa dilihat pada gambar 2.1:

Diagram

Description automatically generated

Gambar 2. 1 Algoritma Gradient Boosting (Jolly, 2018)

Inilah yang terjadi pada diagram tersebut:

1. *Decision tree* pertama dilatih dengan data yang kita miliki, dan variabel Y sebagai target.
2. Kemudian hitung kesalahan residual untuk *tree* tersebut. Kesalahan residual merupakan selisih antara nilai *actual* dengan nilai prediksi.
3. *Tree* kedua sekarang dilatih, menggunakan residual sebagai target.
4. Proses membangun banyak pohon ini bersifat iteratif, dan berlanjut untuk jumlah *estimator* yang kita tetapkan.
5. Prediksi akhir dibuat dengan menjumlahkan nilai target yang diprediksi oleh *tree* pertama ke hasil penyusutan dan residual untuk semua *tree* lainnya. Penyusutan adalah faktor yang kita gunakan untuk mengontrol laju proses *Gradient boosting* yang kita inginkan.
6. Nilai penyusutan (*learning\_rate*) yang kecil menunjukkan bahwa algoritma akan belajar lebih cepat, dan oleh karena itu, harus dikompensasi dengan jumlah *estimator* dasar yang lebih besar (*decision tree*) untuk mencegah *overfitting*.
7. Nilai penyusutan (*learning\_rate*) yang lebih besar menunjukkan bahwa algoritma akan belajar lebih lambat, sehingga membutuhkan lebih sedikit *tree* untuk mengurangi waktu komputasi (Jolly, 2018).

#### *Extreme Gradient Boosting*

Pada tahun 2014, Dr. Tianqi Chen yang berasal dari universitas washington mengembangkan sebuah model bernama *eXtreme Gradient Boosting* yang menjadikan *gradient boosting* sebagai dasar dalam membangun teorinya (Zhang & Gong, 2020). *Gradient boosting* merupakan algoritma yang dapat meminimalisir fungsi *loss* dan manjadi solusi untuk optimasi berbagai masalah khususnya pada regresi, klasifikasi dan ranking. Pada dasarnya, algoritma XGBoost mengadopsi konsep menyesuaikan parameter pembelajaran secara berulang untuk mengurangi nilai loss function (sebuah mekanisme evaluasi untuk model). XGBoost memanfaatkan model dengan struktur pohon regresi yang lebih teratur untuk memperbaiki kinerja dan mengurangi kompleksitas model agar tidak mengalami *overfitting*. Dengan menggunakan model pohon regresi yang lebih teratur, XGBoost mampu memberikan performa yang lebih baik (Sunata, Azrullah, & Rianto, 2020). Hasil prediksi akhir dari *XGBoost* adalah penjumlahan hasil prediksi dari setiap pohon regresi (Li & Zhang, 2020). Algoritma berbasis *decision tree* memiliki kinerja yang baik pada data dengan fitur kategorikal dan tidak terlalu berpengaruh terhadap data dengan kelas tidak seimbang (Pinata, Sukarsa, & Rusjayanthi, 2020). Dalam metode ini, diperlukan sebuah fungsi objektif yang berguna untuk mengevaluasi sejauh mana model yang dihasilkan sesuai dengan data latih. Fungsi objektif ini memiliki dua bagian penting, yaitu nilai yang hilang pada pelatihan dan nilai regularisasi, seperti yang terlihat pada persamaan (1).

Dimana L merupakan fungsi *loss*, dan Ω sebagai fungsi regularisasi, dan merupakan parameter dari model. Fungsi pelatihan yang hilang secara umum dapat ditulis seperti pada persamaan (2).

Dimana 𝑦𝑖 adalah nilai data *actual* dan merupakan nilai prediksi, sedangkan n merupakan banyak perulangan dalam model.

Diasumsikan data adalah , jadi ada n observasi. Setiap observasi memiliki m fitur dan variabel y. Maka nilai prediksi yang akan diperoleh model adalah , adalah seperti persamaan (3):

Disini adalah *regression tree*  dan adalah skor yang disediakan *k-tree* untuk observasi ke-i. Fungsi dipilih untuk meminimumkan nilai fungsi objektif seperti persamaan (4):

Dimana *l* adalah *loss function* dan Ω menunjukkan fungsi regularisasi seperti persamaan (5):

Dimana T adalah jumlah *leaves*  dan *w* adalah besar bobot *leaf.* Untuk menghindari *overfitting* dan model yang sederhana, pinalti untuk T diatur oleh , dan pinalti untuk *w* diatur oleh Hal yang membedakan dengan *gradient boosting* biasa adalah pinalti yang bernilai unik. Metode iterasi digunakan untuk mengurangi fungsi objektif. Dalam iterasi ke- *t,* maka ditambahkan untuk mengurangi fungsi objektif seperti persamaan (6):

Ekspansi Taylor digunakan untuk menyederhanakan persamaan ini. Dari persamaan Taylor kita bisa menurunkan rumus untuk mengurangi *loss* setelah *tree* dibagi dari *node* yang diberikan (Ma, et al., 2020), rumusnya bisa dilihat pada persamaan (7).

Dimana *l* mewakili subset dari pengamatan yang tersedia di *node* saat ini. adalah hipunan bagian dari observasi yang tersedia di *node* kiri dan merupakan hipunan bagian dari observasi yang tersedia di *node* kanan setelah *split*. Fungsi dan digunakan untuk menemukan *split* terbaik, dan didefenisikan sebagai persamaan (8) dan (9):

Fungsi objektif akhir hanya bergantung pada gradien orde pertama dan kedua dari fungsi kerugian pada setiap titik data dan parameter regularisasi γ

Selanjutnya *hyperparameter tuning* untuk *XGBoost* adalah sebagai berikut:

1. *max\_depth* dan *min\_child\_weight*: berfungsi untuk mengontrol arsitektur *tree*. *max\_depth* menentukan jumlah *node* maksimum dari *root* hingga *leaf* terjauh (angka defaultnya adalah 6). *min\_child\_weight* adalah *weight* minimum untuk membuat simpul baru *tree*.
2. *learning\_rate*: Ini menentukan jumlah koreksi pada setiap langkah, mengingat setiap putaran *boosting* mengoreksi kesalahan putaran sebelumnya. *learning\_rate* mengambil nilai dari 0 hingga 1, dan nilai defaultnya adalah 0,3.
3. *n\_estimators*: Ini menentukan jumlah pohon dalam *ensembel*. Nilai defaultnya adalah 100. Perhatikan bahwa jika menggunakan *vanilla XGBoost* dari *scikit-learn*, kita akan menggunakan *num\_boost\_rounds* daripada *n\_estimators*.

#### *Hyperparameter Tuning*

Bayangkan sebuah *sound system* yang memiliki sistem *speaker* dan *mixer* berkualitas tinggi. Anda pasti pernah melihat serangkaian tombol di konsol yang secara independen mengontrol parameter kualitas suara tertentu. *Bass*, *treble*, dan kenyaringan adalah beberapa kontrol yang harus diatur dengan benar untuk pengaturan suaranya agar enak didengar. Demikian pula, model *machine learning* yang mempunyai pengaturan atau penyesuaian berbagai parameter kontrol. Parameter ini disebut *hyperparameter*, dan proses pengontrolan berbagai parameter pada suatu nilai digunakan untuk mendapatkan kinerja terbaik model dengan waktu pelatihan/eksekusi serta akurasi dan generalisasi model. Mirip dengan contoh *equalizer* suara, beberapa *hyperparameter* perlu disetel bersama untuk performa optimal. Ada dua strategi yang biasanya digunakan saat memilih kombinasi hyperparameter:

1. ***Grid Search***: *Hyperparameter* diplot pada matriks dan kombinasi yang mendapatkan kinerja terbaik dipilih untuk model yang digunakan dalam skenario sebernarnya. *Grid Search* merupakan metode untuk penyesuaian hyperparameter dengan cara "*brute force*". Dalam metode ini, kita membuat kisi dari kemungkinan nilai *hyperparameter* yang bersifat diskrit dan model disesuaikan dengan setiap kombinasi nilai *hyperparameter* tersebut. Performa model untuk setiap kombinasi direkam dan kemudian dipilih kombinasi yang menghasilkan performa terbaik. *Grid Search* adalah algoritma *powerfull* yang dapat menemukan kombinasi terbaik dari hyperparameter secara tepat, akan tetapi metode ini memiliki kelemahan dalam kecepatannya. Menyesuaikan model dengan setiap kombinasi membutuhkan kapasitas komputasi yang tinggi dan waktu yang signifikan, sehingga proses akan sangat lama dan tidak efisien terutama jika sumber daya komputasi terbatas. (Navas, Anyscale, 2022).
2. ***Random Search***: Dalam kasus *random search*, nilai *hyperparameter* dipilih secara acak. Dalam hal ini, dengan jumlah iterasi yang sama dengan *grid search*, ada peluang yang lebih baik untuk mencapai nilai optimal untuk *hyperparameter*. Perbedaan antara *grid search* dan *random search* dapat dilihat pada gambar 2.2.

Chart, diagram, scatter chart

Description automatically generated

Gambar 2. 2 Perbedaan grid search dan random search (Deshpande & Kumar, 2018).

Variasi dari teknik *random search* dapat digunakan untuk mengurangi jumlah iterasi melalui ruang pencarian. Teknik ini secara luas dikategorikan sebagai *Coarse to fine search*. Dalam hal ini, pencarian acak dijalankan untuk beberapa iterasi dan setelah wilayah dengan kombinasi pengoptimalan yang lebih tinggi diidentifikasi, ruang pencarian dibatasi pada zona nilai *hyperparameter* yang lebih kecil. Dengan teknik ini, pencarian dibatasi pada suatu wilayah. Teknik *Coarse to fine* dapat divisualisasikan dalam gambar 2.3.

Chart, scatter chart

Description automatically generated

Gambar 2. 3 Teknik Coarse to fine pada random search (Deshpande & Kumar, 2018).

Selama iterasi pencarian awal, seluruh ruang dicari. Saat nilai hyperparameter optimal ditemukan, maka ruang pencarian dibatasi ke zona yang lebih halus. Dengan ini, *hyperparameter* disetel dengan halus dengan jumlah iterasi yang relatif lebih kecil.

### Penelitian Terkait

#### *Rainfall Prediction Using Extreme Gradient Boosting.*

Penelitian mengenai prediksi curah hujan menggunakan eX*treme Gradient Boosting* pernah dilakukan oleh M T Anwar et al. tahun 2021. Data curah hujan yang digunakan berasal dari BMKG Tanjung Mas, Kota Semarang. Terdapat 11 fitur dalam dataset, akan tetapi yang digunakan hanya 8 fitur saja seperti disajikan dalam tabel 2.1. Dataset *training* terdiri dari data cuaca harian dari tahun 2013 hingga 2019, sedangkan data *testing* adalah data cuaca harian tahun 2020. Pada fase *training*, entri dengan *missing value* dihilangkan.

Table 2. 1 Fitur yang ada pada data cuaca (Anwar, Winarno, Hadikurniawati, & Novita, 2020)

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Tipe Data** | **Keterangan** |
| Tn | *Numeric* | Suhu minimum |
| Tx | *Numeric* | Suhu maksimum |
| Tavg | *Numeric* | Rata-rata suhu |
| RH\_avg | *Numeric* | Rata-rata kelembapan (%) |
| Ss | *Numeric* | *Sun exposure time (hours)* |
| ff\_x | *Numeric* | Kecepatan angin maksimum (m/s) |
| ff\_avg | *Numeric* | Rata-rata kecepatan angin (m/s) |
| RR | *Numeric* | Curah hujan (mm) |

Setelah dilakukan pelatihan model dalam 100 putaran, diperoleh nilai RMSE untuk pelatihan lebih baik daripada nilai saat training. Hal ini ditunjukkan pada gambar 2.4. Hal ini menunjukkan kecenderungan *overvitting* XGBoost seiring bertambahnya iterasi.

Chart

Description automatically generated

Gambar 2. 4 Hasil pelatihan dan tes RMSE selama iterasi (Anwar, Winarno, Hadikurniawati, & Novita, 2020).

Table 2.2 menunjukkan bahwa nilai kesalahan pelatihan, *10-fold* *cross validation* menghasilkan kesalahan yang lebih rendah daripada menggunakan dataset *full training* yang menunjukkan indikasi *overfitting.* Ketika data non-hujan dikecualikan kesalahannya meningkat. Kesalahan yang lebih tinggi ini mungkin disebabkan oleh hilangnya informasi tentang karakteristik hari tidak hujan. Sehingga mengurangi kemampuan model untuk memprediksi secara akurat jumlah curah hujan.

Table 2. 2 Pelatihan RMSE dengan n\_rounds (n\_estimators) = 100, n\_fold = 10, eta (learning\_rate) = 0.3, max\_depth = 6 (Anwar, Winarno, Hadikurniawati, & Novita, 2020).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Round** | ***RMSE* pelatiahan** | | | |
| **Dengan data *non-rainy*** | | **Tampa data *non-rainy*** | |
| **10-fold CV** | ***Complete Training*** | **10-fold CV** | ***Complete Training*** |
| 1 | 13.22+0.25 | 13.30 | 22.36+0.39 | 20.46 |
| 50 | 4.41+0.14 | 4.75 | 9.62+0.27 | 12.93 |
| 100 | 2.46+0.11 | 2.75 | 6.92+0.25 | 10.85 |

*XGBoost* mampu memeringkat atribut penting yang berkontribusi pada model. Pemeringkatan atribut ditunjukkan pada Gambar 2.5 Ini menunjukkan bahwa model prediksi curah hujan sangat dipengaruhi oleh kelembaban relatif rata-rata (RH\_avg) dan suhu minimum (Tn). Hasil ini sejalan dengan penelitian sebelumnya yang menggunakan model C4.5 untuk memprediksi apakah suatu hari hujan atau tidak (Anwar, S, Tantriyati, & Windarni, 2020).

Chart

Description automatically generated

Gambar 2. 5 Ranking masing-masing atribut (Anwar, Winarno, Hadikurniawati, & Novita, 2020)

Saat diuji dengan data cuaca pada tahun 2020, model tersebut memberikan MAE 8,8. Gambar 2.6 menunjukkan *scatter plot* dari prediksi RR terhadap RR aktual pada dataset uji. Ini menunjukkan bahwa banyak data terkonsentrasi pada nilai mendekati 0 yang sangat menantang untuk model. Garis biru adalah garis tren linier dengan korelasi R sebesar 0,555. Hasil ini lebih rendah dari penelitian terbaru (Lee, Pham, Lee, Ly, & Lee, 2020) yang menggunakan *Nonlinear Autoregressive* Neural Network dan memiliki R = 0,9. Saran penelitian masa depan untuk mengeksplorasi pengaturan parameter (*tuning*) *XGBoost* untuk meningkatkan kemampuan prediksinya. Penelitian lebih lanjut juga dapat menggabungkan data berbasis Bumi dengan data penginderaan jauh untuk membuat model yang lebih akurat.

Chart, scatter chart

Description automatically generated

Gambar 2. RR prediksi versus RR sebenarnya (Anwar, Winarno, Hadikurniawati, & Novita, 2020)

Pada penelitian ini disimpulkan bahwa model yang dibangun berdasarkan data cuaca historis selama 7 tahun yang dikumpulkan oleh stasiun cuaca menghasilkan prediksi yang akurat untuk estimasi curah hujan harian dengan RMSE 2.7 mm dan MAE pengujian 8.8 mm. Hasil penelitian juga menunnjukkan bahwa faktor yang paling mempengaruhi adalah kelembapan rata-rata dan suhu minimum. Penelitian selanjutnya sangat diharapkan adanya *tuning hyperparameter* *XGBoost* untuk meningkatkan akurasi model, dan tambahan data pengindraan jarak jauh.

#### Studi komparasi model *extreme gradient boosting, SARIMA, exponential smoothing,* dan *Neural Network* untuk peramalan data curah hujan

Penelitian selanjutnya dilakukan oleh Agata et al. pada tahun 2019 melakukan penelitian mengenai perbandingan antara beberapa algoritma *machine learning* yaitu *XGBoost*, SARIMA (*Seasonal Autoregressive Integrated Moving Average*), *exponential smoothing*, dan ANN (*Artificial Neural Network*)*.* Empat model ini dilakukan komparasi untuk mentukan model terbaik untuk melakukan prediksi curah hujan di kota Bandung dari tahun 2018-2019.

Untuk memahami data, langkah awal yang dilakukan peneliti adalah membuat pola sementara untuk data hujan dengan menggunakan *Auto Correlation Function* (ACF). Hasilnya bisa dilihat pada gambar 2.7.

Chart

Description automatically generated

Gambar 2. Auto Correlation Function plot untuk data training (R & I G, 2019)

Gambar diatas menunnjukkan *Autocorrelation function* dengan pola musim yang jelas dengan *time lag* 12 (s = 12 bulan). Ini menunnjukan kota Bandung memiliki pola musim berulang setiap tahunnya. Metode evaluasi yang digunakan adalah MAD (*Mean Absolute Deviance*), RMSED (*Root Mean Squared Error Deviance*)*,* dan MAPE (*Mean Absolute Percentage Deviance*). Perbandingan evaluasi masing masing model ditunjukkan dalam tabel 2.3.

Table 2. Perbandingan masing-masing model (R & I G, 2019).

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **MAD** | **RMSED** | **MAPE** |
| SARIMA | 116.9017 | 162.6560 | 65.7756 |
| *exponential smoothing* | 117.1898 | 158.5143 | 64.8473 |
| *Artificial Neural Network* | 117.1898 | 158.5143 | 64.8473 |
| *XGBoost* | 112.5225 | 152.8657 | 63.7136 |

Berdasarkan tabel diatas kita ketahui bahwa model dengan nilai MAD, RMSED, dan MAPE terendah adalah *XGBoost.* Selanjutnya tes dilakukan dengan data hujan tahun 2018-2019 dan diperoleh grafik prediksi vs aktual pada gambar 2.8.

A picture containing text, linedrawing

Description automatically generated

Gambar 2. 8 Prediksi versus data aktual (R & I G, 2019)

Secara jelasnya bisa dilihat pada gambar 2.9.

Chart

Description automatically generated with medium confidence

Gambar 2. 9 Grafik prediksi vs aktual pada data tes (R & I G, 2019).

#### *Very Short-Term Renewable Energy Power Prediction*

Selanjutnya penelitian yang dilakukan oleh Zhenchuan ma et al. tahun 2020 yang meneliti *XGBoost* sebagai model untuk prediksi kekuatan angin. Selain prediksi yang akurat *XGBoost* juga memiliki kelebihan dalam mengatur *thread* CPUs dalam melaksanakan tugas secara paralel (Ma, et al., 2020). Metode *tuning hyperparameter* yang digunakan pada penelitian ini adalah TPE (*Tree-Structured Parzen Estimator*).

Turbin Angin Sotavento merupakan sebuah turbin angin eksperimental, terletak di Galicia, Spanyol dengan koordinat 43.354377oN dan 7.881213oW. Galicia terletak di barat laut Spanyol, dekat samudra Atlantik dan memiliki iklim maritim dan Atlantik. Wilayah ini kaya akan sumber daya angin, khususnya, selalu ada angin kencang di bulan Januari, Februari, dan Maret. Angin yang bertiup datang dari barat daya dan tiba setelah melewati banyak air. Ada 24 turbin angin dengan 9 model yang berbeda di ladang turbin angin. Total daya nominal adalah 17,56MW.

Data kecepatan angin dan arah angin dihasilkan pada menara anemometri. Tenaga angin yang dihasilkan diukur melalui sistem SCADA. Semua data yang digunakan dalam penelitian ini dirata-ratakan dan disimpan dengan interval 10 menit, mulai dari '1 Januari 2015' hingga '1 Januari 2018'. Catatan juga berisi nilai *missing value*, dan jumlah total catatan yang ditunjukkan pada Tabel 2.4 belum memperhitungkan *missing value*. Kumpulan data pelatihan mencakup 52069 data dan digunakan untuk melatih model *XGBoost*. Kumpulan data validasi berisi 12925 data dan digunakan untuk mengevaluasi kinerja model sehingga *hyperparameter* terbaik dapat dipilih. Kumpulan data uji berisi 12230 data dan digunakan untuk menguji kinerja peramalan model.

Table 2. 4 Dataset yang digunakan dalam penelitian (Ma, et al., 2020).

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Time Stamp awal** | **Time Stamp akhir** | **Jumlah data** |
| Data Latih | 2016-01-01 00:10 | 2017-01-01 00:00 | 52.069 |
| Data Validasi | 2017-01-01 00:10 | 2017-04-01 00:00 | 12.925 |
| Data Uji | 2017-04-01 00:10 | 2017-07-01 00:00 | 12.230 |

Saat XGBoost melatih model, XGBoost dapat menghitung rata-rata pengurangan loss function setiap fitur pada pemisahan node. Dengan ini, model dapat menghitung kontribusi setiap fitur pada pelatihan model. Semakin tinggi nilainya, semakin penting fitur ini (Ma, et al., 2020). Oleh karena itu, setelah model XGBoost dilatih, skor kepentingan setiap fitur dapat langsung diperoleh. Seperti disebutkan sebelumnya, jumlah total fitur kandidat adalah 133. Pada penelitian ini menggunakan fitur ini sebagai input ke algoritma XGBoost untuk melatih model dan hasilnya terdapat skor kepentingan untuk 40 fitur teratas, bisa dilihat pada Gambar 2.10.

Chart, histogram

Description automatically generated

Gambar 2. 10 Skor kepentingan fitur dataset (Ma, et al., 2020)

Metode evaluasi data pada penelitian ini adalah dengan menggunakan NRMSE (*Normalized Root Mean Square Error*), NMAPE (*Normalized Mean Absolute Percentage Error*) dan NPE (*Normalized Percentage Error*). Sedangkan metode prediksinya menggunakan tiga komparasi model yaitu *Persistence, Support Vector Regression* (SVR), dan *XGBoost.* Hasil evaluasi untuk masing-masing model dapat dilihat pada tabel 2.5.

Table 2. 5 Tabel evaluasi model (Ma, et al., 2020)

|  |  |  |  |
| --- | --- | --- | --- |
| **Kriteria evaluasi** | **Persistensi** | **SVR** | **XGBoost** |
| RMSE | 635.8632 | 522.7257 | 464.2721 |
| MAPE | 322.9744 | 326.2751 | 250.6030 |
| NRMSE | 3.6211 | 2.9768 | 2.6439 |
| NMAPE | 1.8393 | 1.8581 | 1.4271 |

Beradasarkan data tabel diatas dapat kita lihat bahwa model *XGBoost* memiliki akurasi prediksi yang tinggi dengan nilai NRMSE dan MAPE terendah.

***3.2.4* Optimalisasi *XGBoost* oleh *Adaptive Particle Swarm Optimization* untuk Penilaian Kredit**

Permasalahan *credit scoring* merupakan masalah yang cukup menantang. Perkembangan teknologi seperti *machine learning* telah menunjukkan performa yang memuaskan untuk mengatasi masalah kehidupan sehari-hari. Karena itu keuntungan yang disajikan dari kombinasi fitur dan pemilihan fitur, seperti *decision tree* dapat menyesuaikan data kredit yang memiliki dimensi tinggi dan kompleks. Salah satu model yang mengambil gagasan *decision tree* adalah *eXtreme Gradient Boosting* yang memperkuat *tree* untuk mengatasi kekurangan dengan mengintegrasikan model *tree* lainnya. Struktur model ditentukan oleh *hyperparameter*, yang ditujukan pada permasalahan umum yaitu masalah *manual tuning* yang menghabiskan waktu yang cukup lama, karena itu bebarapa metode pengoptimalan digunakan untuk melakukan *tuning*.

*Particle swarm optimization* (PSO) menggambarkan keadaan partikel dan hukum geraknya sebagai bilangan real kontinu, maka *hyperparameter* yang berlaku untuk *eXtreme Gradient Boosting* dapat ditemukan nilai optimalnya dalam ruang pencarian kontinu. Namun, *Particle swarm optimization* klasik cenderung jatuh ke optima lokal. Untuk mengatasi masalah ini, Chao Qin et al. mengusulkan model penilaian kredit *eXtreme Gradient Boosting* yang didasarkan pada *Particle swarm optimization* adaptif. Swarm split didasarkan pada ide pengelompokan dan dua jenis strategi pembelajaran akan digunakan untuk memandu partikel untuk meningkatkan keragaman subswarm, untuk mencegah algoritma jatuh ke optimal lokal.

Dalam penelitian ini, algoritman pembelajaran mesin yang akan digunakan adalah algoritma tradisional dan pengklasifikasi pembelajaran *ensemble* populer, serta empat metode pengoptimalan *hyperparameter* (pencarian grid (*Grid Search*), pencarian acak (*Random Search*), TPE (T*ree-structured Parzen Estimator*), dan PSO (*Particle swarm optimization*)) dan melakukan pembandingan performa masing-masing metode. Evaluasi model dalam penelitian ini dilakukan dengan empat set data kredit dan tujuh set data tolok ukur KEEL pada lima nilai evaluasi yang populer yaitu *accuracy*, *error rate* (erortipe I and eror tipe II), *Brier score*, and F1 *score*. Hasil menunjukkan bahwa model yang diusulkan yaitu *Adaptif Particle Swarm Sptimization* (APSO) mengungguli model lain secara rata-rata. Dalam tabel 2.6 disajikan berapa saja *hyperparameter* yang akan di-*tuning* dan hasil *tuning hyperparameter-*nya disajikan dalam table 2.7 (Qin, et al., 2021).

Table 2. 6 Kombinasi ruang pencarian dan jumlah komputasi masing-masing metode tuning (Qin, et al., 2021)

|  |  |  |  |
| --- | --- | --- | --- |
| **Hyper parameter** | ***APSO, PSO, TPE, RS*** | ***Grid Search*** | ***Default value*** |
| *Learning\_rate* | 0.1 | 0.1 | 0.1 |
| *Number\_of\_boosts* | 60 | 60 | 60 |
| *Maximum\_tree\_depth* | (1, 12) | 1, 2, 3 | 0.2 ∗ *number of features* |
| *Subsample\_ratio* | (0.9, 1) | 0.9, 0.95, 1 | 0.9 |
| *Column\_subsample\_ratio* | (0.9, 1) | 0.9, 0.95, 1 | 0,9 |
| *Minimum\_child\_weight* | (0, 4) | 0, 1, 2, 3, 4 | 2 |
| *Maximum\_delta\_step* | (0, 1) | 0.4, 0.6, 0.8, 1 | 1 |
| *Gamma* | (0, 0.01) | 0, 0.01 | 1 |
| *Number of computations* | 200 | 2160 | — |

Table 2. 7 Parameter hasil tuning yang akan digunakan dalam penelitian (Qin, et al., 2021)

|  |  |
| --- | --- |
| **Algoritma** | **Parameter** |
| *Decision Tree* | *Minimum samples leaf = 6*  *Maximum depth = 8*  *Minimum samples split = 2* |
| *Linear Regression* | Tidak ada pramater khusus |
| *Neural Network* | *Epoch = 1000*  *Learning rate = 0.01* |
| *Support Vector Machine* | *Kernel: RBF*  *C = 32*  *Gamma = 0.1* |
| *Adaptive Boosting*, *Bagging-Decision-Tree*, *Random Forest* | *N estimator = 100*  Siada parameter lainnya sama dengan *Decision Tree* |
| *Adaptive Boosting*-*Neural Network*, *Bagging-Neural Network* | *N estimator = 100*  Sisanya sama sama dengan *Neural Network* |
| *Gradient Boosting Decision Tree* | *N estimators = 100*  *Subsample = 0.9*  *Learning rate = 0.1*  *Min samples leaf = 1*  *Max depth = 7* |

Pengukuran performa dari model untuk salah satu dataset bisa kita lihat pada tabel dibawah

Table 2. 8 Tabel hasil performa model untuk dataset austria (Qin, et al., 2021)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Skor F1 | Tipe error II (%) | Tipe error I (%) | ACC (%) | Skor brier |
| *Linear Regression* | 0.7865 | 16.88 | **8.67∗** | 86.77 | 0.1019 |
| *Decision Tree* | 0.8543 | 13.95 | 17.41 | 84.51 | 0.1359 |
| *Neural Network* | 0.8612 | 16.87 | 12.06 | 85.27 | 0.1111 |
| AdaBoost | 0.8644 | 11.93 | 17.40 | 85.64 | 0.1034 |
| SVM | 0.3961 | 13.74 | 15.35 | 85.54 | 0.1012 |
| Bagging-DT | 0.4926 | 13.78 | 13.33 | 86.42 | 0.0987 |
| AdaBoost-NN | 84.59 | 16.36 | 14.22 | 0.8615 | 0.1174 |
| XGBoost | **87.58** | 12.97 | **11.79** | **0.0908** | **0.8756** |
| GBDT | 86.14 | 13.43 | 14.19 | 0.0991 | 0.6026 |
| *Random Forest* | 87.41 | 13.26 | 12.05 | 0.0971 | 0.7571 |
| Bagging-NN | 85.62 | 11.83 | 16.42 | 0.1062 | 0.8683 |

#### *Optimalisasi hiperparameter lanjutan untuk meningkatkan prediksi spasial dari tanah longsor dangkal menggunakan XGBoost*

Algoritma pembelajaran mesin secara progresif telah menjadi bagian dari praktik pemetaan kerentanan tanah longsor karena ketangguhannya dalam menangani mekanisme tanah longsor yang rumit dan non-linier. Namun, struktur internal algoritma semacam itu berisi sekumpulan konfigurasi *hyperparameter* yang pengaturannya benar sangat penting untuk mendapatkan kinerja tertinggi yang dapat dicapai. Penelitian ini menyelidiki keefektifan dan kekokohan algoritma pengoptimalan lanjutan, termasuk *Genetic Algorithm*, *Random Search*, *Bayesian Optimization with Tree-structure Parzen Estimators*, *Bayesian Optimization* *with* *Gausian Process*, dan metode *Hyperband*, untuk mengoptimalkan *hyperparameter* algoritma *eXtreme Gradient Boosting (XGBoost)* dalam prediksi spasial tanah longsor. Terdapat 12 faktor penyebab yang dianggap menghasilkan *landslide susceptibility maps* (LSM) untuk provinsi Trabzon Turki, di mana tanah longsor dangkal translasi terjadi di mana-mana. Metrik akurasi yang akan digunakan untuk mengukur keefektifan strategi pengoptimalan pada algoritme XGBoost adalah *overall accuracy* (OA, *F1-score*), *precision*, A*rea Under the receiver operating characteristic Curve* (AUC), *recall*, dan uji signifikansi statistik.

Dibandingkan dengan model *XGBoost* dengan pengaturan default, model yang dioptimalkan memberikan peningkatan signifikan hingga 13% dalam hal akurasi keseluruhan, yang juga dipastikan dengan uji McNemar. Analisis AUC menunjukkan bahwa model memiliki kinerja yang serupa secara statistik, metode GA (0,942) dan *Hyperband* (0,922) memiliki kemampuan prediksi tertinggi, diikuti oleh *Bayesian Optimization* *with* *Gausian Process* (0,920), *Bayesian Optimization with Tree-structure Parzen Estimators* (0,899), dan *Random Search* (0,894). Analisis efisiensi biaya komputasi menunjukkan bahwa pendekatan *Hyperband* (40,3 detik) jauh lebih cepat (sekitar 13 kali) daripada GA dalam *tuning hyperparameter*, dan dengan demikian algoritma pengoptimalan terbaik untuk masalah ini adalah *hyperband* (Kavzoglu & Teke, 2022). Pemilihan *hyperparameter* bisa dilihat pada tabel 2.9.

Table 2. 9 Pemilihan tuning hyperparameter (Qin, et al., 2021)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Hyper parameter** | **Raung pencarion** | | **Nilai Optimal** | | | | |
| **Batas atas** | **Batas bawah** | **BO-GP** | **RS** | **Hyperband** | **GA** | **BO-TPE** |
| Min child weight | 20 | 1 | 17 | 7 | 3 | 16 | 5 |
| Learning rate | 1.0 | 0.1 | 0.3 | 0.7 | 0.5 | 0.1 | 0.6 |
| Maximum depth | 20 | 1 | 3 | 11 | 4 | 1 | 2 |
| Gamma | 0.2 | 0.0 | 0.0 | 0.0 | 0.2 | 0.1 | 0.1 |
| Colsample by tree | 0.7 | 0.5 | 0.5 | 0.5 | 0.7 | 0.7 | 0.6 |
| N estimator | 500 | 100 | 100 | 390 | 17 | 150 | 500 |
| subsample | 1.0 | 0.8 | 0.9 | 1.0 | 1.0 | 0.9 | 0.9 |

#### Mengkaitkan Klasifikasi Serapan Vaksin Campak dan Faktor Pendukungnya Menggunakan Model Ensemble Machine Learning

Campak adalah salah satu masalah kesehatan masyarakat yang signifikan yang bertanggung jawab atas tingginya angka kematian di seluruh dunia, terutama di negara-negara berkembang. Dengan menggunakan data survei demografi dan kesehatan yang representatif secara nasional, penggunaan vaksin campak telah diklasifikasikan, dan faktor-faktor yang mendasarinya diidentifikasi melalui pendekatan *Ensemble Machine Learning* (ML) (Hasan, et al., 2021).

Pertama, nilai yang hilang diperhitungkan dengan menggunakan berbagai pendekatan, dan kemudian beberapa teknik pemilihan fitur telah diterapkan untuk mengidentifikasi atribut penting untuk memprediksi vaksinasi campak. Teknik pengoptimalan *hyperparameter* yaitu *grid search* telah diterapkan untuk menyetel *hyperparameter* kritis dari berbagai model ML, seperti Naive Bayes, *random forest*, *decision tree*, *XGboost*, dan *lightgbm*. Dataset yang digunakan adalah data BDHS (*Bangladesh Demographic and Health Survey*) dan performanya akurasinya diuji dengan metode AUC (*area under the receiver operating characteristic curve*).

Secara individual, lightgbm yang dioptimalkan memberikan presisi tertinggi dan AUC masing-masing sebesar 79,90% dan 77,80%. Hasil ini meningkat ketika lightgbm yang dioptimalkan dipadukan dengan *XGboost*, memberikan presisi dan AUC masing-masing sebesar 84,60 % dan 80,0 %. Hasil penelitian ini mengungkapkan bahwa teknik imputasi median statistik dengan metode pemilihan atribut berbasis *XGboost* dan pengklasifikasi *lightgbm* memberikan hasil individual terbaik. Performa meningkat ketika ansambel berbobot yang diusulkan dari pendekatan *XGboost* dan *lightgbm* diadaptasi dengan preprocessing yang sama dan direkomendasikan untuk penggunaan vaksin campak. Signifikansi dari pendekatan yang diusulkan penelitian ini adalah menggunakan atribut minimum yang dikumpulkan dari anak dan anggota keluarga mereka dan menghasilkan akurasi 80,0%, membuatnya mudah dijelaskan oleh pengasuh dan petugas kesehatan (Hasan, et al., 2021).

Hasil *tuning hyperparameter* beberapa model ditunjukkan pada tabel 2.10 seperti berikut ini.

Table 2. 10 hasil tuning hyperparameter dan hasil uji AUC (Hasan, et al., 2021)

|  |  |  |
| --- | --- | --- |
| **Model** | **Hasil tuning hyperparameter** | **AUC** |
| **GNB** | Variasi atribut terbesar untuk stabilitas (var\_smothing = 0.01) |  |
| **BNB** | Parameter halus laplace (alpha = 0.0), fitur binarizing threshold (binarize = 0.4), dan kelas prior (fit\_prior = true) |  |
| **RF** | Fungsi pembagi (criterion = gini), jumlah pembagian fitur (max\_features = auto), jumlah daun node (max\_leaf\_nodes = none), dan jumlah daun node sample (min\_sample\_leaf = 1.0) |  |
| **Decision Tree** | Fungsi pembagi (criterion = entrophy), kebijakan partisi node (splitter = best), split feature number (max\_feature = log2), jumlah internal node sample (min\_samples\_split = 0.1) dan jumlah daun sample node’s (min\_sample\_leaf = 0.05) |  |
| **XGBoost** | Jumlah hessian (min\_child\_weight = 1.0), loss minimum (gamma = 1.0), rasion instance subdsample (subsample = 0.25), rasio subsample kolom (colsample\_bytree = 1.0), kedalaman pohon (max\_depth = 5) |  |
| **LGBoost** | Base learner daun pohon (num\_leaves = 25), jumlah pohon (n\_estimators = 50), rasion subsample (subsample = 0.25), rasio subsample kolom (cosample\_bytree = 1.0), kedalaman pohon (max\_depth = -1) |  |

#### Model pembelajaran mesin yang efisien untuk prediksi kekuatan beton

Dalam penelitian ini, *machine learning* akan diimplementasikan untuk memprediksi kekuatan tekan dan tarik beton HPC *(High Performance Concrete).* Algoritma yang akan digunakan adalah GBR (*Gradient Boosting Regressor*),SVR (*Support Vector Regressor*), MLP (*MultiLayer Perceptron*), dam *XGBoost* (*eXtreme Gradient Boosting*). Metode *tuning* untuk *hyperparameter* yang digunakan adalah *Random Search.*

Dalam persiapan data, nilai yang hilang atau *missing value* akan diisi dengan rata-rata dari data yang tersedia yang memungkintakan lebih banyak informasi yang dapat digunakan dalam proses pelatihan. Hasil prediksi model menunjukkan bahwa model yang terbaik yang memiliki performa bagus adalah GBR dan *XGBoost* dibandingkan dengan model SVR dan MLP. Hasil *tuning hyperparameter* bisa dilihat pada tabel 2.11 dan 2.12 (Nguyen, Vu, Vo, & Thai, 2021).

Table 2. 11 perbandingan performa masing-masing model untuk dataset kekuatan tekan (Nguyen, Vu, Vo, & Thai, 2021)

Table

Description automatically generated

Table 2. 12 perbandingan performa masing-masing model untuk dataset kekuatan tarik (Nguyen, Vu, Vo, & Thai, 2021)

Graphical user interface, table

Description automatically generated

#### Prediksi Gagal Jantung menggunakan algoritma Xgboost dan seleksi fitur menggunakan permutasi fitur

Jantung adalah organ vital tubuh manusia yang menyediakan darah kaya oksigen bagi tubuh. Jika jantung berhenti bekerja, maka manusia akan mati dalam rentang waktu beberapa menit. Penelitian ini menganalisis kinerja algoritma *XGBoost* dalam prediksi kematian di antara pasien gagal jantung. Performa *XGBoost* dibandingkan dengan berbagai algoritma prediksi lainnya. *Tuning hyperparameter* telah dilakukan pada algoritma *XGBoost* untuk meningkatkan kinerja model dan mengurangi *overfitting*. Pada bagian kedua dari penelitian ini, pemilihan fitur menggunakan permutasi fitur telah dilakukan dengan menggunakan fitur penting dari dataset yang telah dipilih. Pencarian *hyperparameter* menggunakan metode *RandomSearchCV* dari *sklearn*. Lipatan bertingkat atau *stratified fold* digunakan selama *tuning hyperparameter* untuk memastikan bahwa data pelatihan dan pengujian tetap seimbang selama pembuatan dan prediksi model. Menggunakan pemilihan fitur, telah ditetapkan bahwa menggunakan sejumlah kecil fitur penting dapat meningkatkan kinerja model dengan jumlah yang signifikan dan dapat mengurangi *overfitting*.

Table 2. 13 hasil tuning hyperparameter optimal (Kaushik & Birok, 2021)

|  |  |  |
| --- | --- | --- |
| ***Hyperparameter*** | ***Optimal Value achieved*** | ***Description*** |
| *Minimum child weight* | 1 | Minimal weight yang dibutuhkan anak *tree* |
| *subsample* | 0.4 | Rasio subsample pelatihan |
| *gamma* | 3.5 | Parameter emangkasan leaf |
| *Colsample bytree* | 0.9 | Kolom subsample |
| *Maximum depth* | 3 | Maksimum keadalam pohon |
| *N estimators* | 125 | Jumlah pohon yang dibutuhkan |
| *Learning rate* | 0.15 | Parameter skala learning rate |

#### Prediksi pembelajaran mesin sifat magnetik gelas logam berbasis Fe dengan mempertimbangkan kemampuan pembentukan kaca

*Metallic Glass* berbasis Fe (MG) telah menunjukkan nilai komersial yang besar karena sifat magnetik lunaknya yang sangat baik. Prediksi magnetisme dengan pertimbangan kemampuan pembentukan kaca sangat penting untuk mengembangkan MG berbasis Fe. Namun, teori atau model yang dibuat berdasarkan fisika *condense matter* menunjukkan akurasi yang terbatas dan beberapa pengecualian. Dalam penelitian ini, berdasarkan 618 sampel MG berbasis Fe yang dikumpulkan dari penelitian yang yang telah dipublikasi, model *machine learning* (ML) dilatih dengan baik untuk memprediksi magnetisasi jenuh () dari MG berbasis Fe. *Glass forming ability* diperlakukan sebagai fitur menggunakan data eksperimen dari wilayah cairan superdingin (Tx). Tiga algoritma ML yang digunakan, yaitu ANN (*Artificial Neural Network*) , dan RF (*Random Forest*), *XGBoost* (*eXtreme Gradient Boosting*). Melalui pemilihan fitur dan *tuning hyperparameter*, *XGBoost* menunjukkan kinerja prediktif terbaik pada dataset uji yang dipisahkan secara acak dengan koefisien determinasi MAPE (*mean absolute percent error*) sebesar 5,563%, (R2) sebesar 0,942, dan RMSE (*root mean squared error*) sebesar 0,078 T. Berbagai peringkat kepentingan fitur yang diturunkan oleh model *XGBoost* menunjukkan bahwa Tx memainkan peran penting dalam performa prediksi model. Penelitian ini menunjukkan metode ML yang diusulkan dapat secara bersamaan menggabungkan GFA dan fitur lainnya dalam termodinamika, kinetika, dan struktur untuk memprediksi sifat magnetik MG berbasis Fe dengan akurasi yang sangat baik. Hasil *tuning hyperparameter* bisa dilihat pada gambar 2.11.

Chart

Description automatically generated

Gambar 2. 11 Hasil penyetelan hyperparameter untuk XGBoost dan RF dievaluasi oleh . Proses penyetelan (a) N estimator di XGBoost dan RF, (b) max depth dan minimum child weight di XGBoost, dan (c) maximum depth dan minimum samples split dari Random Forest. (d) Peringkat kepentingan fitur diperoleh dari model XGBoost terlatih. Skor kepentingan dinormalisasi dengan membagi skor maksimum, dan skor maksimum ditetapkan menjadi 100 (Li, Shan, & Shek, 2022).

#### Memprediksi Penipuan dalam Layanan Pembayaran Keuangan melalui Model XGBoost Hyper-Parameter-Tuned yang Dioptimalkan

Transaksi online, layanan medis, transaksi keuangan, dan perbankan semuanya memiliki andil dalam aktivitas penipuan. Pendapatan tahunan yang dihasilkan oleh penipuan melebihi $1 triliun. Meskipun penipuan berbahaya bagi organisasi, hal itu dapat diungkap dengan bantuan solusi cerdas seperti *rules engine* dan *machine learning*. Dalam penelitian ini, teknik hibrid unik digunakan untuk mengidentifikasi penipuan pembayaran finansial dengan menggabungkan *nature-inspired-based* *hyperparameter* dengan beberapa model *supervised learning*, seperti yang diterapkan dalam versi Algoritma *modified* *XGBoost*. Pertama, peneliti membagi sampel kumpulan data pembayaran keuangan lengkap untuk digunakan sebagai kumpulan pengujian. Data yang digunakan 70% data untuk pelatihan dan 30% untuk pengujian. Rekaman yang diketahui tidak sah atau palsu diprediksi, sedangkan yang menimbulkan kecurigaan diselidiki lebih lanjut menggunakan sejumlah algoritma pembelajaran mesin. Model dilatih dan divalidasi menggunakan teknik *10-fold cross-validation*.

Pada penelitian ini algoritma *nature-inspired-based hyperparemeter* yang diusulkan peneliti menyediakan kemampuan modifikasi pada model *XGBoost* yang meningkatkan akurasi dari model dalam menentukan mana saja yang merupakan penipuan. Penelitian ini menunjukkan performa yang cukup bagus dengan akurasi 99.68%, sedangkan *error rate* untuk memprediksi penipuan diharapkan sekitar 0.32%. Perbandingan dengan model lain bisa kita lihat pada tabel 2.14.

Table 2. 14 Perbandingan performa (Dalal, Seth, Radulescu, Secara, & Tolea, 2022)

|  |  |  |
| --- | --- | --- |
| **Algoritma** | **AUC** | **Akurasi** |
| Neural Network | 0.172 | 92.18 |
| Random Forest | 0.573 | 96.15 |
| Logistic Regression | 0.655 | 92.737 |
| Quest | 0.569 | 96.648 |
| Random Tree | 0.652 | 96.089 |
| Bayesian Network | 0.5 | 91.06 |
| Tree-AS | 0.681 | 96.648 |
| C5.0 | 0.5 | 96.648 |
| Discriminant algorithm | 0.667 | 97.207 |
| Modified XGBoost | 0.9988 | 99.68 |

## BAB 3 METODE PENELITIAN

### Alur Penelitian

Diagram dari alur penelitian dapat dilihat pada gambar 3.1

Diagram

Description automatically generated

Gambar 3. 1 Diagram alur penelitian

Dataset yang akan digunakan adalah data mentah, sehingga data perlu diproses terlebih dahulu. Pada proses ini nilai *missing value* akan dihilangkan, karena data yang hilang akan mempengaruhi proses pelatihan. Selanjutnya data yang sudah siap dipilih fitur yang akan digunakan dalam proses pelatihan. Fitur ini mencakup kelembapan udara, data curah hujan, kecepatan angin, suhu udara, dan klasifikasi hujan atau tidak hujan sebagai target. Selanjutnya data dibagi menjadi data *train* dan *test* dan dilakukan optimasi *hyperparameter* untuk mendapatkan *paremter* terbaik untuk model *eXtreme Gradient Boosting.* Terakhir model dievaluasi sebarapa baik model dalam melakukan estimasi dan berapa skor akurasi, presisi dan *metrics* evaluasi lainnya.

### Tahapan Penelitian

Penelitian dilakukan dengan beberapa tahapan untuk mendapatkan hasil yang optimal. Tahapannya adalah sebagai berikut:

#### Idetifikasi Masalah

latar belakang masalah dalam penelitian ini adalah mengenai dampak yang ditimbulkan oleh curah hujan. Curah hujan sangat mempengaruhi kehidupan manusia mulai dari dampak positif maupun dampak negatif. Dalam mengatasi dampak negatif maka perlu adanya sistem yang dapat memperkirakan curah hujan yang akan terjadi. Oleh karena itu *eXtreme Graadient Boosting* diusulkan sebagai model untuk estimasi curah hujan.

#### Studi Literature

Kajian pustaka dilakukan berupa mengkaji literatur dan beberapa penelitian terkait yang sudah pernah dilakukan sebelumnya tentang topik yang akan dibahas. Kajian literatur yang dibahas adalah mengenai pembelajaran mesin, algoritma *decision tree,* dan *random forest* yang menjadi landasan dalam *gradient boosting,* lalu algoritma *eXtreme Gradient Boosting.*

#### Data Collecting

Data yang digunakan adalah data curah hujan stasiun cuaca yang telah dikumpulkan oleh Badan Meteorologi Klimatologi dan Geofisika (BMKG)dalam kurun waktu tertentu untuk Kota Pontianak.

#### Data Understanding

Pada tahap ini data yang telah dikumpulkan dipahami secara keseluruhan mencakup informasi dalam data dan menentukan kualitasnya. Pemahaman data sangat penting untuk menghindari masalah yang tidak terduga selama tahapan berikutnya, yaitu tahap persiapan data (*data preparation*).

#### Data Preparation

Secara umum, hal-hal yang akan Anda lakukan dalam *data preparation* adalah sebagai berikut:

* Menggabungkan data.
* Menyeleksi data yang akan digunakan.
* Melakukan proses transformasi data
* Membagi data menjadi data training dan test.

#### Modeling dan Evaluation

Pemodelan dilakukan dalam beberapa iterasi. Pada tahap ini, peneliti menggunakan algoritma *XGBoost* untuk membuat model estimasi. Model ini akan dilatih dengan default *hyperparameter* dan melakukan evaluasi performa dengan *metrics:*

1. Akurasi: merupakan persentase dari prediksi benar yang dilakukan oleh model dari semua total prediksi. Dapat dituliskan dalam persamaan (10).
2. Presisi: Rasio *true* positif terhadap jumlah total prediksi positif yang dibuat oleh model. Dihitung dalam persamaan (11).
3. *Recall* (sensitivitas): Rasio *true* positif terhadap jumlah total positif actual dalam kumpulan data. Dihitung sebagai persamaan (12).
4. F1-score: Rata-rata *harmonic* dari presisi dan *recall*. Ini memberikan kepentingan yang sama untuk presisi dan *recall*. Ini dihitung sebagai persamaan (13).

#### Model Optimization (*Hyperparameter tuning*)

Tahap ini model akan di-*tuning* padatiga variasi *hyperparameter-*nyayaitu *max\_depth, learning\_rate,* dan *n\_estimator.* Pada penelitian ini fokus parameter yang akan dituning adalah untuk mencari nilai *learning\_rate* yang terbaik untuk model *eXtreme Gradient Boosting* dan parameter lainnya dibiarkan dengan nilai default*.*

### Tempat Penelitian

Penelitian ini dilakukan di Laboratorium Cisco, Departement Fisika, FMIPA Universitas Indonesia, Kota Depok.

### Jadwal Penelitian

Berikut ini adalah agenda penelitian dan jadwal pelaksanannya:

Tabel 3. 1 Tabel agenda dan jadwal penelitian

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Kegiatan** | **2023** | | | | | | |
| **Bulan ke-** | | | | | | |
| **2** | **3** | **4** | **5** | **6** | **7** |  |
| Tahap I (persiapan) | | | | | | | | |
| 1 | Kajian Literatur |  |  |  |  |  |  |  |
| 2 | Penyusunan Proposal |  |  |  |  |  |  |  |
| Tahap II (Penelitian) | | | | | | | | |
| 3 | Pengolahan data |  |  |  |  |  |  |  |
| 4 | Eksplorasi data |  |  |  |  |  |  |  |
| 5 | Analisis data |  |  |  |  |  |  |  |
| 6 | Hyperparameter Optimization |  |  |  |  |  |  |  |
| Tahap II (Penulisan) | | | | | | | | |
| 7 | Pembuatan naskah skripsi |  |  |  |  |  |  |  |
| 8 | Sidang skripsi |  |  |  |  |  |  |  |
| 9 | Revisi Skripsi dan pembuatan manuskrip makalah ilmiah |  |  |  |  |  |  |  |

### Kesimpulan

Metode yang akan digunakan dalam penelitian ini adalah *eXtreme Gradient Boosting* untuk premasalah klasifikasi. Harapannya penelitian ini dapat diselesaikan dengan hasil estimasi curah hujan yang memiliki presisi yang baik, dan dapat diterapkan dalam melakukan estimasi curah hujan di instansi terkait.

## BAB 4 HASIL DAN PEMBAHASAN

### 4.1 *Data Understanding*

## DAFTAR PUSTAKA

Anwar, M. T., S, N., Tantriyati, V., & Windarni, V. (2020). Rain Prediction Using Rule Based Machine Learning Approach. *Adv. Sustain. Sci. Eng. Technol*, 2.

Anwar, M. T., Winarno, E., Hadikurniawati, W., & Novita, M. (2020). Rainfall prediction using Extreme Gradient Boosting. *Journal of Physics: Conference Series Annual Conference on Science and Technology (ANCOSET 2020)*, 1.

Ciaburro, G., Ayyadevara, V. K., & Perrier, A. (2018). *Hands-On Machine Learning on.* Birmingham: Packt Publishing Ltd.

Dalal, S., Seth, B., Radulescu, M., Secara, C., & Tolea, C. (2022). Predicting Fraud in Financial Payment Services through Optimized Hyper-Parameter-Tuned XGBoost Model. *Mathematics*, 1-17.

Deshpande, A., & Kumar, M. (2018). *Artificial Intelligence for Big Data.* Birmingham: Packt Publishing Ltd.

Hasan, M. K., Jawad, M. T., Dutta, A., Awal, M. A., Islam, A. M., Masud, M., & Al-Amri, J. F. (2021). Associating Measles Vaccine Uptake Classification and Its Underlying Factors Using an Ensemble of Machine Learning Models. *IEEE Access*, 119613.

Jolly, K. (2018). *Machine Learning with Scikit-Learn Quick Start Guide.* Birmingham: Packt Publishing Ltd.

Kaushik, S., & Birok, R. (2021). Heart Failure prediction using Xgboost algorithm and feature selection using feature permutation. *ICECCT*, 1.

Kavzoglu, T., & Teke, A. (2022). Advanced hyperparameter optimization for improved spatial prediction of shallow landslides using extreme gradient boosting (XGBoost). *Bulletin of Engineering Geology and the Environment*, 201.

Kurikulum Developer Dicoding. (2023, February 8). *Machine Learning Terapan: Model Development dengan Boosting Algorithm*. Diambil kembali dari Dicoding: https://www.dicoding.com/academies/319/tutorials/18590

Lee, V. M., Pham, B. T., Lee, T.-T., Ly, H.-B., & Lee, L. M. (2020). Daily Rainfall Prediction Using Nonlinear Autoregressive Neural Network micro-electronics and Telecomunication Engineering. *Springer*, 213.

Li, S., & Zhang, X. (2020). Research on orthopedic auxiliary classification and prediction model based on XGBoost algorithm. *Neural Computing and Applications*, 1971-1979.

Li, X., Shan, G., & Shek, C. H. (2022). Machine learning prediction of magnetic properties of Fe-based metallic glasses considering glass forming ability. *Journal of Materials Science & Technology*, 113-120.

Ma, Z., Chang, H., Sun, Z., Liu, F., Li, W., Zhao, D., & Chen, C. (2020). Very Short-Term Renewable Energy Power Prediction Using XGBoost Optimized by TPE Algorithm. *IEEE*, 1237.

Navas, J. (2022, February 8). *Anyscale*. Diambil kembali dari What is hyperparameter tuning?: https://www.anyscale.com/blog/what-is-hyperparameter-tuning

Navas, J. (2022, Februari 8). *Anyscale Blog*. Diambil kembali dari Anyscale: https://www.anyscale.com/blog/what-is-hyperparameter-tuning

Nguyen, H., Vu, T., Vo, T. P., & Thai, H.-T. (2021). Efficient machine learning models for prediction of concrete strengths. *Construction and Building Materials*, 8.

Pham, K., Kim, D., Le, C. V., & Choi, H. (2022). Dual tree‑boosting framework for estimating. *Landslides*, 2249.

Pinata, N. N., Sukarsa, I. M., & Rusjayanthi, N. K. (2020). Prediksi Kecelakaan Lalu Lintas Di Bali Dengan Xgboost Pada Python. *Jurnal Ilmiah Merpati*, 188-196.

Qin, C., Zhang, Y., Bao, F., Zhang, C., Liu, P., & Liu, P. (2021). XGBoost Optimized by Adaptive Particle Swarm Optimization for Credit Scoring. *Mathematical Problems in Engineering*, 12.

R, A., & I G, M. J. (2019). A comparison of extreme gradient boosting, SARIMA, exponential smoothing, and neural network models for forecasting rainfall data. *Journal of Physics: Conference Series* , 4.

Sunata, H., Azrullah, F. J., & Rianto, Y. (2020). „Komparasi Tujuh Algoritma Identifikasi Fraud ATM Pada PT. Bank Central Asia Tbk 1,2,3. *Jurnal Teknik Informatika dan Sistem Informasi*, 441-450.

Tankari, M. R. (2020). Rainfall variability and farm households’ food insecurity. *Food Security*, 576.

Tattar, P. N. (2018). *Hands-On Ensemble Learning .* Birmingham: Packt Publishing Ltd.

Xiang, Y., Gou, L., He, L., xia, S., & Wang, W. (2018). A SVR–ANN combined model based on ensemble EMD for rainfall. *Applied Soft Computing Journal*, 874-875.

Zhang, D., & Gong, Y. (2020). „The Comparison Of Lightgbm And Xgboost Coupling Factor Analysis . *IEEE Access*, 8.