

Prediksi Akun Bot pada Media Sosial X Menggunakan Random Forest, XGBoost dan SVM

Marcella Averina¹, Wilfridus Bambang Triadi Handaya², Albertus Joko Santoso³

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta

Jl. Babarsari No. 43, Sleman 55281, Daerah Istimewa Yogyakarta, Indonesia

Email: 1210711305@students.uajy.ac.id, wilfridus.handaya@uajy.ac.id,
joko.santoso@uajy.ac.id

Abstract. *The social media platform X is often misused, one example being the creation of fake accounts operated by bots. To address this issue, various machine learning models have been developed to detect bot accounts, particularly Random Forest, XGBoost, and Support Vector Machine (SVM). This study aims to compare the performance of these three models in predicting bot accounts on social media platform X and to determine the best model based on the f1-score. In addition, this research applies three hyperparameter tuning methods, namely grid search, random search, and Bayesian optimization. The results show that the XGBoost model with Bayesian optimization achieves the best performance. However, its performance decreases when applied to imbalanced datasets.*

Keywords: *bot, X, machine learning, hyperparameter tuning*

Abstrak. *Penggunaan media sosial X seringkali disalahgunakan, salah satunya dengan membuat akun palsu yang dijalankan oleh bot. Saat ini, telah banyak dikembangkan sistem untuk mendeteksi akun bot pada media sosial X dengan memanfaatkan model machine learning, khususnya algoritma Random Forest, XGBoost, dan SVM. Penelitian ini bertujuan untuk membandingkan performa dari ketiga model tersebut dalam memprediksi akun bot pada media sosial X dan menentukan model terbaik berdasarkan nilai f1-score. Pada penelitian ini juga dikembangkan penggunaan hyperparameter tuning grid search, random search, dan Bayesian optimization. Hasil penelitian ini menunjukkan bahwa model terbaik untuk prediksi akun bot pada media sosial X adalah XGBoost dengan hyperparameter tuning Bayesian optimization. Namun, model ini mengalami penurunan performa ketika digunakan untuk memprediksi akun bot pada data yang tidak seimbang.*

Kata Kunci: *bot, X, machine learning, hyperparameter tuning*

1. Pendahuluan

1.1 Latar Belakang

Penggunaan media sosial telah memengaruhi banyak aspek kehidupan manusia [1][2]. Tidak terbatas pada pertukaran informasi, media sosial menjadi sarana interaksi antar manusia, hiburan, media promosi, dan transaksi jual beli [3][4]. Hingga saat ini, terdapat berbagai jenis platform media sosial yang banyak digunakan, salah satunya X atau yang sebelumnya biasa disebut sebagai Twitter. X merupakan layanan media sosial berbasis *microblog* yang memungkinkan pengguna untuk berbagi konten teks singkat seperti berita, acara, maupun isi pikiran pengguna secara publik maupun pribadi [5][6].

Meningkatnya penggunaan media sosial seringkali menjadi alasan penyalahgunaan kegiatan yang dilakukan, salah satunya adalah pembuatan akun palsu yang dijalankan oleh bot [7]. Akun bot mampu menjalankan aktivitas seperti yang biasa dilakukan oleh manusia, misalnya mengikuti akun pengguna lain, mengunggah atau menyukai *tweet*, dan mengirim pesan langsung secara otomatis [8][9][10]. Kemampuan akun bot tersebut umumnya digunakan untuk memalsukan identitas pengguna akun, mengirim *spam*, membagikan tautan berbahaya, manipulasi, dan membuat berita palsu [11][12][13]. Dampaknya, akun bot dapat memengaruhi

kepercayaan pengguna, beredarnya informasi menyesatkan, dan rusaknya reputasi banyak pihak [14][15].

Untuk mengatasi keberadaan akun bot tersebut, telah dikembangkan beberapa sistem yang berguna untuk mendeteksi akun bot. Sistem tersebut umumnya memanfaatkan pengembangan model *machine learning* [3][16][17]. Berdasarkan studi literatur terhadap penelitian sebelumnya, model *machine learning* yang umumnya digunakan antara lain Random Forest, XGBoost, dan SVM. Model tersebut dipilih karena performanya yang cukup baik dalam menangani prediksi akun bot pada media sosial X. Namun, sebagian besar penelitian tersebut belum mengaplikasikan penggunaan *hyperparameter tuning* dan *cross-validation* dalam pengembangan dan evaluasi model.

Penelitian ini merupakan penerapan model *machine learning* dengan algoritma Random Forest, XGBoost, dan SVM dengan kernel RBF untuk memprediksi akun bot pada media sosial X. Ketiga model tersebut menggunakan *pipeline* yang masing-masing terdiri atas *data scaling*, *feature selection*, dan *classification*. Kemudian, dilakukan *hyperparameter tuning grid search*, *random search*, dan *Bayesian optimization* dengan *cross-validation* StratifiedKFold. Fitur numerik yang menjadi dasar dari prediksi berdasarkan pada hasil *feature selection* RFE dengan estimator Random Forest.

2. Tinjauan Pustaka

Hingga saat ini, model *machine learning* banyak digunakan untuk pengembangan model deteksi akun bot. Berdasarkan studi literatur terhadap penelitian yang sudah pernah dilakukan, sistem deteksi bot umumnya menggunakan algoritma Random Forest, XGBoost, dan SVM. Pemilihan model tersebut dikarenakan performanya yang cukup baik dalam menangani deteksi bot pada media sosial X.

Hendra Kurniawan dalam penelitian yang berjudul “Deteksi Twitter Bot Menggunakan Klasifikasi Decision Tree” menggunakan model Decision Tree dengan *feature selection information gain*, *gain ratio*, dan *gini index* untuk mendeteksi bot pada Twitter. Metrik evaluasi penelitian ini yaitu akurasi, *precision*, *recall*, dan nilai ROC-AUC. Hasil terbaik didapatkan dengan *feature selection gain ratio* pada rasio *train-test split* 90:10, yaitu akurasi 88,84% dan nilai AUC 0,965 [5].

A. Ramalingaiah, dkk. dalam penelitian yang berjudul “Twitter Bot Detection Using Supervised Machine Learning” menggunakan model *supervised learning* untuk mendeteksi bot Twitter, yaitu KNN, Logistic Regression, Naïve Bayes, dan Bag of Words. Rasio *train-test split* optimal pada penelitian ini yaitu 80:20. Akurasi paling tinggi dalam penelitian ini didapatkan pada model Decision Tree, tetapi performa paling stabil ditunjukkan oleh KNN. Penelitian ini juga menemukan bahwa model Bag of Words memberikan hasil paling sempurna, yaitu 96% pada *train set* dan 99,2% pada *test set* [8].

Aqilah A. Zahra, dkk. dalam penelitian yang berjudul “Bot Detection Application on Twitter Using Machine Learning with Random Forest Classifier Algorithm” menggunakan algoritma Random Forest pada rasio *train-test split* 80:20 untuk mendeteksi bot pada Twitter. Metrik evaluasi yang diukur dari pelatihan dan evaluasi model tersebut yaitu *accuracy*, *precision*, *recall*, dan *f1-score*. Hasilnya, didapatkan *accuracy* 96,79% dan *f1-score* 96% [11].

Abdulbasit A. Darem, dkk. dalam penelitian yang berjudul “Cybersecurity in social networks: An ensemble model for Twitter bot detection” menggunakan algoritma Logistic Regression, Decision Tree, Random Forest, XGBoost, SVM RBF, MLP, dan Ensemble Voting untuk mendeteksi bot pada media sosial, khususnya Twitter. Penelitian ini juga menggunakan *hyperparameter tuning grid search* dan validasi dengan *5-fold cross-validation*. Metrik evaluasi yang diukur yaitu akurasi, *precision*, *recall*, dan *f1-score*. Berdasarkan hasil pelatihan dan evaluasi model, Ensemble Voting menghasilkan akurasi 90,22% dan *f1-score* 89,54% [18].

Alexander Shevtsov, dkk. dalam penelitian yang berjudul “Identification of Twitter Bots Based on an Explainable Machine Learning Framework” menggunakan model *machine learning* XGBoost, Random Forest, dan SVM dengan seleksi fitur Random Forest, Lasso, serta

feature importance. Tujuannya untuk mendeteksi bot Twitter dengan studi kasus terkait Pemilu Amerika pada tahun 2020. Penanganan *class imbalance* dilakukan dengan SMOTE dan Tomek Links. Penelitian ini memberikan hasil terbaik berupa model XGBoost dengan *f1-score* 91,9%, PR-AUC 96,7%, dan ROC-AUC 97,9%. Selain itu, diberikan hasil interpretasi menggunakan SHAP [19].

Rachit Shukla, dkk. dalam penelitian yang berjudul “TweezBot: An AI-Driven Online Media Bot Identification Algorithm for Twitter Social Networks” menggunakan algoritma Decision Tree, Random Forest, Naïve Bayes, SVM MLP, dan Bag of Words untuk fitur deskripsi. Penelitian ini bertujuan untuk mendeteksi bot Twitter secara daring. Model tersebut dilatih dalam bentuk *pipeline* tiga lapis untuk menyaring *outlier*, *threshold* logis, dan integrasi model *machine learning*. Rasio *train-test split* yang digunakan dalam penelitian ini yaitu 70:30. Penelitian ini menghasilkan akurasi 99%, *precision* 99,24%, *recall* 93,85%, *f1-score* 98,94%, dan AUC-ROC 0,9966 [20].

Alothali, dkk., dalam penelitian yang berjudul “Hybrid feature selection approach to identify optimal features of profile metadata to detect social bots in Twitter” menggunakan model Random Forest, Naïve Bayes, SVM, dan Neural Network untuk mendeteksi bot Twitter berdasarkan profil. Hasil dari penelitian ini yaitu model terbaik didapatkan pada Random Forest melalui *cross-validation attribute evaluation*. Hasil evaluasi menunjukkan nilai AUC 94,3%, akurasi 89%, *precision* 83,8%, dan *f1-score* 83,5% [21].

Maryan Heidari, dkk. dalam penelitian yang berjudul “Deep Contextualized Word Embedding for Text-based Online User Profiling to Detect Social Bots on Twitter” menggunakan model algoritma Random Forest, SVM, Logistic Regression, dan Artificial Neural Network untuk mendeteksi bot Twitter berdasarkan profil. Hasil terbaik didapatkan dari model Random Forest. Berdasarkan perhitungan metrik evaluasi, didapatkan akurasi 92,3%, *f1-score* 91,2%, dan MCC 0,887 [22].

Berdasarkan tinjauan pustaka, cukup banyak penelitian yang telah menggunakan algoritma Random Forest, XGBoost, dan SVM. Penelitian tersebut juga memberikan hasil yang cukup baik. Namun, belum banyak yang mengembangkan penelitian serupa menggunakan *hyperparameter tuning grid search*, *random search*, dan *Bayesian optimization*. Oleh karena itu, penelitian ini mengembangkan penggunaan *hyperparameter tuning*.

3. Metodologi Penelitian

Penelitian ini berfokus pada perbandingan performa model Random Forest, XGBoost, dan SVM dalam mendeteksi bot pada X. Algoritma dengan performa terbaik diintegrasikan ke dalam aplikasi web dan *chatbot* Telegram. Tahapan yang dilakukan dalam penelitian ini yaitu (1) studi literatur, (2) pengembangan *dataset*, (3) pengembangan model *machine learning*, (4) pengembangan arsitektur sistem, (5) pengembangan aplikasi web menggunakan *framework* Streamlit, (6) pengembangan *chatbot* Telegram.

Studi literatur merupakan tahap pembelajaran terkait penelitian sistem deteksi akun bot media sosial X yang sudah ada sebelumnya. Pembelajaran dilakukan untuk mengidentifikasi perbandingan *dataset*, model *machine learning* yang digunakan, dan performa antar model. Hasil dari perbandingan tersebut digunakan sebagai dasar pengembangan model *machine learning* dalam penelitian ini.

Penelitian ini menggunakan *dataset* Cresci 2017 dengan dua subset data, yaitu “fake_users.csv” berupa data akun bot dan “genuine_users.csv” berupa data akun asli [23]. Sebelum digunakan dalam pelatihan model *machine learning*, data diberi label kelas *is_bot* dengan nilai 1 untuk akun bot dan 0 untuk akun asli. Setelah pelabelan data, dilakukan *preprocessing data*, *exploratory data analysis* (EDA), dan *feature engineering*.

Pelatihan model dilakukan dalam *pipeline* yang terdiri atas *data scaling*, *feature selection*, dan *classifier* masing-masing algoritma. *Hyperparameter tuning* yang digunakan yaitu *grid search*, *random search*, dan *Bayesian optimization*. Hasil evaluasi model dibuat dalam bentuk tabel metrik evaluasi rata-rata *macro*, nilai metrik evaluasi berdasarkan kelas akun

asli, dan nilai metrik evaluasi berdasarkan kelas akun bot. Model terbaik juga ditampilkan dalam bentuk *confusion matrix* untuk mengetahui perbandingan hasil prediksi benar dan salah.

Model *machine learning* dikembangkan pada Python *notebook*, hasil evaluasi model terbaik disimpan ke dalam dokumen pickle (.pkl) yang digunakan sebagai *backend* dari sistem yang akan diimplementasikan ke dalam aplikasi web dan *chatbot* Telegram. Antarmuka web dikembangkan dengan *framework* Streamlit. *Chatbot* Telegram dikembangkan menggunakan Telegram API yang diintegrasikan dengan perintah yang dijalankan menggunakan skrip Python.

Aplikasi web pada sistem ini memiliki dua menu, yaitu prediksi akun bot dan menu yang mengarahkan pengguna ke *chatbot* Telegram. Pada *chatbot* Telegram, pengguna dapat melakukan prediksi dengan konsep yang sama seperti berkirim pesan singkat di Telegram. Proses prediksi pada *chatbot* Telegram berdasarkan pada model *machine learning* yang sama dengan model yang digunakan pada aplikasi web.

Pengujian model *machine learning* dilakukan menggunakan *dataset* eksternal botometer-feedback-2019 yang sudah memiliki label *ground truth*. Sebelum pengujian, dilakukan *mapping dataset* untuk mengambil fitur-fitur yang akan digunakan dalam pengujian. Setelah *mapping dataset*, pengujian dilakukan dengan membandingkan antara *ground truth* dan hasil prediksi.

4. Hasil dan Diskusi

4.1 Preprocessing Data dan Exploratory Data Analysis (EDA)

Dataset Cresci 2017 terdiri atas dua subset data, yaitu “fake_users.csv” dan “genuine_users.csv”. Kedua dokumen *dataset* tersebut diunggah ke Google Colab untuk diberi label *is_bot*. *Dataset* digabungkan dan diambil fitur numerik utama berupa *followers_count*, *friends_count*, *favourites_count*, *statuses_count*, dan label *is_bot*.

Pada *dataset* ini terdapat 6.825 baris data yang terdiri atas 3.351 data akun bot dan 3.474 data akun asli. Diketahui *dataset* ini relatif seimbang. Hal itu karena rasio persentase antar kelas data cukup dekat, yaitu 50,9% akun asli dan 49,1% akun bot.

Preprocessing data terdiri atas *data cleansing*, penghapusan kolom yang tidak digunakan, konversi tipe data numerik, pengisian *missing values* dengan median, dan normalisasi distribusi data. Setelah *preprocessing data*, kolom yang dipertahankan hanya fitur numerik utama. Distribusi data juga menjadi lebih simetris dan mendekati normal setelah normalisasi menggunakan transformasi log (np.log1p).

Pada tahap EDA, dilakukan analisis statistik terhadap fitur numerik utama setelah *preprocessing data*. Hasilnya, akun asli memiliki tingkat interaksi dan aktivitas yang lebih tinggi jika dibandingkan dengan akun bot. Proses kemudian dilanjutkan dengan *outlier removal* menggunakan IQR dan Z-Score.

Penelitian ini juga melakukan pembuatan fitur baru pada tahap *feature engineering*. Fitur baru yang dibuat yaitu *followers_friends_ratio* dan *followers_friends_difference*. Dengan demikian, fitur numerik utama pada penelitian ini berkembang menjadi lima fitur asli dan dua fitur turunan. Rincian fitur numerik utama dapat dilihat pada Tabel 1.

Tabel 1. Fitur Numerik Utama Setelah Feature Engineering

Nama Fitur	Definisi
<i>followers_count</i>	Jumlah pengikut.
<i>friends_count</i>	Jumlah akun yang diikuti oleh pengguna.
<i>favourites_count</i>	Jumlah <i>tweet</i> yang disukai oleh pengguna.
<i>statuses_count</i>	Jumlah <i>tweet</i> yang diunggah oleh pengguna.
<i>listed_count</i>	Jumlah daftar publik yang memasukkan akun pengguna.
<i>followers_friends_ratio</i>	Rasio jumlah pengikut (<i>followers_count</i>) dengan jumlah akun yang diikuti (<i>friends_count</i>).
<i>followers_friends_difference</i>	Selisih jumlah pengikut dengan jumlah akun yang diikuti.

Setelah fitur turunan dibuat, nilai data kosong (NaN) diisi dengan nol. Data akhir setelah seluruh rangkaian proses *preprocessing data* dan *outlier removal* terdapat 5.918 baris yang terdiri atas 2.716 akun asli dan 3.202 akun bot. Data tersebut yang kemudian digunakan pada pelatihan dan evaluasi model.

4.2 Pengembangan Model Machine Learning

Dataset dibagi ke dalam *train set* dan *test set* dengan rasio 80:20, yaitu 80% *train set* dan 20% *test set*. Setelah *preprocessing* terdapat 5.918 data di dalam *dataset* Cresci 2017. Artinya, pembagian data dalam *train-test split* yaitu 4.734 *train set* dan 1.184 *test set*. *Train set* digunakan sebagai data pelatihan dan *test set* digunakan sebagai data dalam pengujian model.

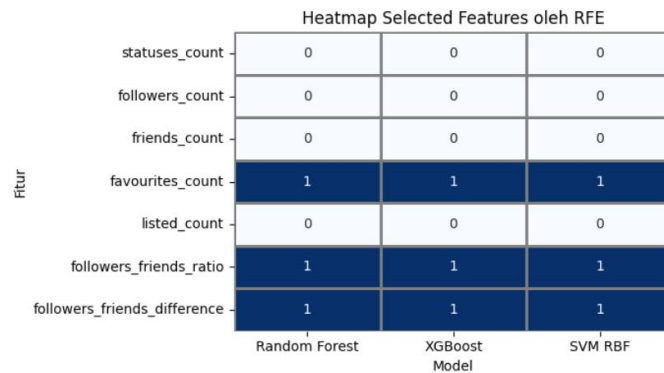
Algoritma *machine learning* dalam penelitian ini yaitu Random Forest, XGBoost, dan SVM RBF. Masing-masing model dilatih dalam *pipeline* yang terdiri atas *data scaler*, *feature selection*, dan *classification*. *Data scaling* menggunakan StandardScaler dan *feature selection* menggunakan Recursive Feature Elimination (RFE) dengan estimator Random Forest untuk memilih tiga fitur terbaik yang berkontribusi dalam prediksi akun bot. Validasi model dilakukan menggunakan Stratified 5-fold *cross-validation*.

Pelatihan model dilakukan dalam bentuk *baseline* model menggunakan parameter *default* dan *hyperparameter tuning* menggunakan *grid search*, *random search*, dan *Bayesian optimization*. Parameter yang diatur oleh model Random Forest yaitu *n_estimators* dan *max_depth*. Pada model XGBoost, parameter yang diatur yaitu *n_estimators*, *max_depth*, dan *learning_rate*. Pengaturan parameter SVM dengan kernel RBF yaitu pada *C* dan *gamma*. Pengaturan parameter dapat dilihat pada Tabel 2.

Tabel 2. Pengaturan Parameter pada Hyperparameter Tuning

Algoritma	Hyperparameter Tuning	Parameter	Nilai
Random Forest	Grid Search	n_estimators	[100, 150]
		max_depth	[3, 5]
	Random Search	n_estimators	[100, 150]
		max_depth	[3, 5]
	Bayes Optimization	n_estimators	Integer(100, 150)
		max_depth	Integer(3, 5)
XGBoost	Grid Search	n_estimators	[100, 150]
		max_depth	[3, 5]
		learning_rate	[0.1, 0.3]
	Random Search	n_estimators	[100, 150]
		max_depth	[3, 5]
		learning_rate	[0.1, 0.3]
	Bayes Optimization	n_estimator	Integer(100, 150)
		learning_rate	Integer(3, 5)
SVM RBF	Grid Search	max_depth	Real(0.1, 0.3, log-uniform)
		C	[0.1, 1]
	Random Search	gamma	['scale', 'auto']
		C	[0.1, 1]
	Bayes Optimization	gamma	['scale', 'auto']
		C	Real(0.1, 1, log-uniform)
		gamma	Categorical(['scale', 'auto'])

Pada akhir pelatihan model, dilakukan visualisasi hasil seleksi fitur oleh RFE Random Forest dalam bentuk *heatmap*. Berdasarkan *heatmap* pada Gambar 1, ketiga algoritma baik Random Forest, XGBoost, maupun SVM RBF memiliki hasil yang sama. *Heatmap* tersebut menunjukkan bahwa fitur yang paling berkontribusi dalam prediksi akun bot pada media sosial X yaitu jumlah *tweet* yang disukai, rasio pengikut dengan akun yang diikuti, dan selisih antara pengikut dengan akun yang diikuti.



Gambar 1. Heatmap Hasil Feature Selection

Evaluasi model dilakukan dengan *test set*. Metrik evaluasi yang diukur yaitu akurasi, *precision*, *recall*, *f1-score*, dan nilai ROC-AUC. Perhitungan metrik evaluasi dibagi menjadi tiga bagian, yaitu nilai rata-rata *macro* seperti pada Tabel 3, evaluasi kelas akun asli seperti pada Tabel 4, dan evaluasi kelas akun bot seperti pada Tabel 5. Hasil evaluasi diurutkan berdasarkan nilai *f1-score* tertinggi.

Tabel 3. Perhitungan Metrik Evaluasi Macro

Model	Accuracy	Precision Macro	Recall Macro	F1-Score Macro	ROC-AUC
XGBoost - Bayes	0.991554	0.991791	0.991214	0.991491	0.999375
Random Forest - Grid	0.990709	0.990862	0.990434	0.990641	0.999252
Random Forest - Bayes	0.990709	0.990862	0.990434	0.990641	0.999312
Random Forest - Random	0.990709	0.990862	0.990434	0.990641	0.999252
SVM RBF - Bayes	0.990709	0.990862	0.990434	0.990641	0.997196
SVM RBF - Random	0.990709	0.990862	0.990434	0.990641	0.997167
SVM RBF - Grid	0.990709	0.990862	0.990434	0.990641	0.997167
SVM RBF - Base	0.990709	0.990862	0.990434	0.990641	0.997167
Random Forest - Base	0.989865	0.990252	0.989373	0.989786	0.998043
XGBoost - Grid	0.989865	0.990252	0.989373	0.989786	0.999364
XGBoost - Random	0.989865	0.990252	0.989373	0.989786	0.999364
XGBoost - Base	0.989020	0.989317	0.988593	0.988937	0.999249

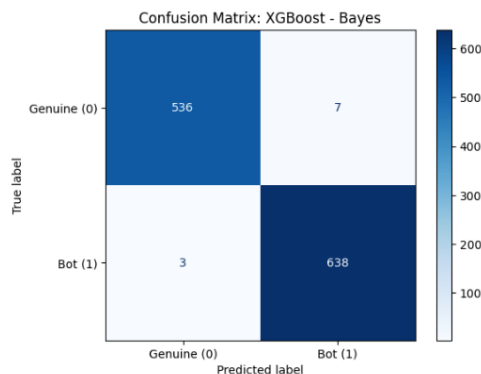
Tabel 4. Hasil Perhitungan Metrik Evaluasi Kelas Akun Asli

Model	Accuracy	Precision Akun Asli	Recall Akun Asli	F1-Score Akun Asli	ROC-AUC
XGBoost - Bayes	0.991554	0.994434	0.987109	0.990758	0.999375
Random Forest - Grid	0.990709	0.992593	0.987109	0.989843	0.999252
Random Forest - Bayes	0.990709	0.992593	0.987109	0.989843	0.999312
Random Forest - Random	0.990709	0.992593	0.987109	0.989843	0.999252
SVM RBF - Bayes	0.990709	0.992593	0.987109	0.989843	0.997196
SVM RBF - Random	0.990709	0.992593	0.987109	0.989843	0.997167
SVM RBF - Grid	0.990709	0.992593	0.987109	0.989843	0.997167
SVM RBF - Base	0.990709	0.992593	0.987109	0.989843	0.997167
Random Forest - Base	0.989865	0.994413	0.983425	0.988889	0.998043
XGBoost - Grid	0.989865	0.994413	0.983425	0.988889	0.999364
XGBoost - Random	0.989865	0.994413	0.983425	0.988889	0.999364
XGBoost - Base	0.989020	0.992565	0.983425	0.987974	0.999249

Tabel 5. Perhitungan Metrik Evaluasi Kelas Akun Bot

Model	Accuracy	Precision Bot	Recall Bot	F1-Score Bot	ROC-AUC
XGBoost - Bayes	0.991554	0.989147	0.99532	0.992224	0.999375
Random Forest - Grid	0.990709	0.989130	0.99376	0.991440	0.999252
Random Forest - Bayes	0.990709	0.989130	0.99376	0.991440	0.999312
Random Forest - Random	0.990709	0.989130	0.99376	0.991440	0.999252
SVM RBF - Bayes	0.990709	0.989130	0.99376	0.991440	0.997196
SVM RBF - Random	0.990709	0.989130	0.99376	0.991440	0.997167
SVM RBF - Grid	0.990709	0.989130	0.99376	0.991440	0.997167
SVM RBF - Base	0.990709	0.989130	0.99376	0.991440	0.997167
Random Forest - Base	0.989865	0.986090	0.99532	0.990683	0.998043
XGBoost - Grid	0.989865	0.986090	0.99532	0.990683	0.999364
XGBoost - Random	0.989865	0.986090	0.99532	0.990683	0.999364
XGBoost - Base	0.989020	0.986068	0.99376	0.989899	0.999249

Berdasarkan nilai *f1-score macro* tertinggi pada masing-masing model, didapatkan hasil bahwa model XGBoost dengan *Bayesian optimization* menjadi model terbaik dengan nilai *f1-score macro* 99,1%. Hasil tersebut dibuktikan dengan *confusion matrix* untuk mengetahui jumlah prediksi benar dan salah pada masing-masing kelas data seperti yang divisualisasikan oleh Gambar 2. *Confusion matrix* dibuat berdasarkan pengujian terhadap 1.184 *test set* dan diketahui model XGBoost dengan *Bayesian optimization* mampu memberikan 536 *true negative* dan 638 *true positive* sehingga total kesalahan yang terjadi hanya ada 10 prediksi.

**Gambar 2. Confusion Matrix XGBoost dengan Bayesian optimization**

Setelah didapatkan XGBoost dengan *Bayesian optimization* sebagai model terbaik, model tersebut disimpan ke dalam dokumen pickle (.pkl). Dokumen ini dapat diunduh untuk kemudian diimplementasikan ke dalam sistem. Penggunaan dokumen pickle ini sebagai *backend* pada sistem aplikasi.

4.3 Pengujian Model

Pengujian model XGBoost dengan *Bayesian optimization* yang terpilih sebagai model terbaik dilakukan pada *dataset* eksternal botometer-feedback-2019. Tujuan pengujian ini untuk mengetahui kinerja model yang telah dilatih dengan *dataset* Cresci 2017. *Dataset* botometer-feedback-2019 memiliki label *ground truth* sehingga pengujian dilakukan dengan membandingkan hasil prediksi dengan nilai *ground truth*.

Dokumen *dataset* ini terdiri atas data profil akun media sosial X dengan format JSON dan dokumen label kelas dalam format tsv. Sebelum pengujian, kedua dokumen digabungkan untuk *mapping dataset* dan pemeriksaan kolom. Setelah itu, ambil kolom yang dibutuhkan untuk pengujian dan simpan hasil *mapping* ke dalam dokumen csv. Sampel data hasil *mapping* seperti pada Tabel 6.

Tabel 6. Sampel Data Hasil Mapping Dataset

followers_ count	friends_ count	favourites_ count	followers_friends_ ratio	followers_friends_ difference	favourites_ log	label
790	3218	12898	0.245494	2428	9.464905	human
16039	10960	10835	1.463412	5079	9.290629	bot
7618	7800	3116	0.976667	182	8.044626	human
388	674	1728	0.575668	286	7.455298	bot
19677	13611	5946	1.445669	6066	8.690642	human

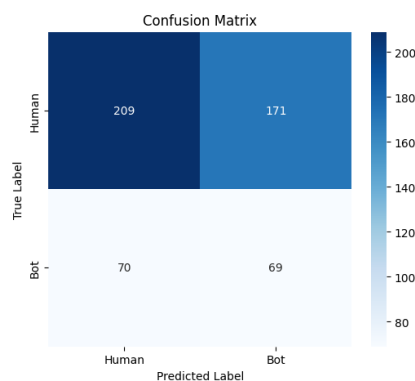
Dokumen csv hasil *mapping* tersebut diujikan pada model XGBoost dengan *Bayesian optimization* yang didapatkan dari pelatihan dan evaluasi serta telah disimpan dalam bentuk dokumen pickle (.pkl). Pengujian menghasilkan label prediksi yang membedakan akun asli dengan akun bot. Hasil prediksi dibandingkan dengan nilai *ground truth* seperti pada Tabel 7.

Tabel 7. Sampel Data Hasil Pengujian

followers_ count	friends_ count	favourites_ count	favourites_ _log	followers_friends_ ratio	followers_friends_ difference	ground_truth	prediction
790	3218	12898	9.465	0.245	2428	Human	Bot
16039	10960	10835	9.291	1.463	5079	Bot	Human
7618	7800	3116	8.045	0.977	182	Human	Human
388	674	1728	7.455	0.576	286	Bot	Bot
19677	13611	5946	8.691	1.446	6066	Human	Human

Pengujian juga dilakukan dengan menghitung metrik evaluasi *dataset* botometer-feedback-2019 terhadap model XGBoost dengan *Bayesian optimization*. Perhitungan metrik evaluasi terdiri atas akurasi, *precision*, *recall*, *f1-score*, dan nilai ROC-AUC. Selain itu, dilakukan juga perhitungan *confusion matrix*.

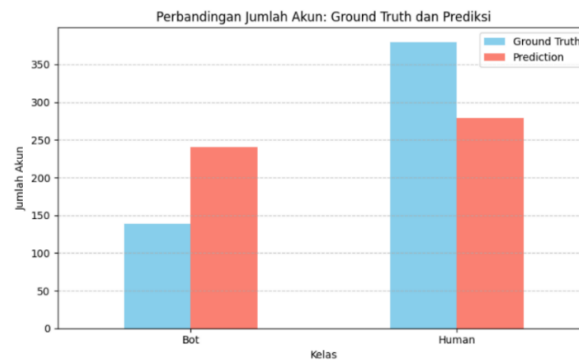
Berdasarkan perhitungan metrik evaluasi, didapatkan nilai *f1-score* 36,4%, akurasi 53,6%, *precision* 28,7%, *recall* 49,6%, dan nilai ROC-AUC 52,3%. Hasil tersebut didapatkan dari pengujian terhadap 519 data pada *dataset* botometer-feedback-2019. Pada perhitungan *confusion matrix* yang ditunjukkan oleh Gambar 3, didapatkan 209 *true negative*, 171 *false positive*, 70 *false negative*, dan 69 *true positive*.

**Gambar 3. Confusion Matrix Hasil Pengujian Model**

Berdasarkan analisis terhadap *dataset* eksternal, terjadi ketidakseimbangan pada *dataset* botometer-feedback-2019. Karakteristik *dataset* yang tidak seimbang memengaruhi hasil pengujian sehingga terjadi perbedaan signifikan antara hasil pengujian dengan hasil pelatihan dan evaluasi model. Berdasarkan pengujian, dapat diketahui bahwa model XGBoost dengan *Bayesian optimization* kurang optimal dalam menangani prediksi terhadap *dataset* yang tidak seimbang.

Pada Gambar 4, diketahui bahwa perbandingan jumlah data pada *dataset* botometer-feedback-2019 tidak seimbang. Berdasarkan grafik, data akun asli berjumlah lebih dari 350

sedangkan data akun bot kurang dari 150. Namun, hasil prediksi memberikan perbandingan jumlah data yang lebih seimbang daripada nilai data berdasarkan *ground truth*.



Gambar 4. Hasil Analisis Dataset Pengujian

Justin M. Johnson dan Taghi M. Khoshgotfaar menekankan pendapat serupa dalam penelitiannya. Jumlah kelas data yang tidak seimbang mengakibatkan bias hasil prediksi karena model mengabaikan kelas data yang lebih sedikit [13]. Hal itu menyebabkan banyaknya *false positive* dan menurunkan nilai *precision*. Rendahnya nilai *precision* berdampak pada nilai *f1-score* sehingga jika nilai *precision* turun maka nilai *f1-score* juga turun.

4.4 Implementasi Model

Model XGBoost dengan *Bayesian optimization* yang terpilih menjadi model terbaik disimpan ke dalam dokumen pickle (.pkl). Dokumen model tersebut digunakan sebagai *backend* dari aplikasi web yang dikembangkan dengan *framework* Streamlit. Implementasi model juga dilakukan pada *chatbot* Telegram.

Pada aplikasi web yang divisualisasikan Gambar 5, pengguna dapat mengisi data profil akun yang ingin diprediksi. Data tersebut yaitu jumlah pengikut, jumlah akun yang diikuti, dan jumlah *tweet* yang disukai. Sistem akan memprediksi apakah akun tersebut merupakan akun asli atau akun bot berdasarkan fitur yang telah dipilih oleh RFE, yaitu *favourites_count*, *followers_friends_ratio*, dan *followers_friends_difference*.

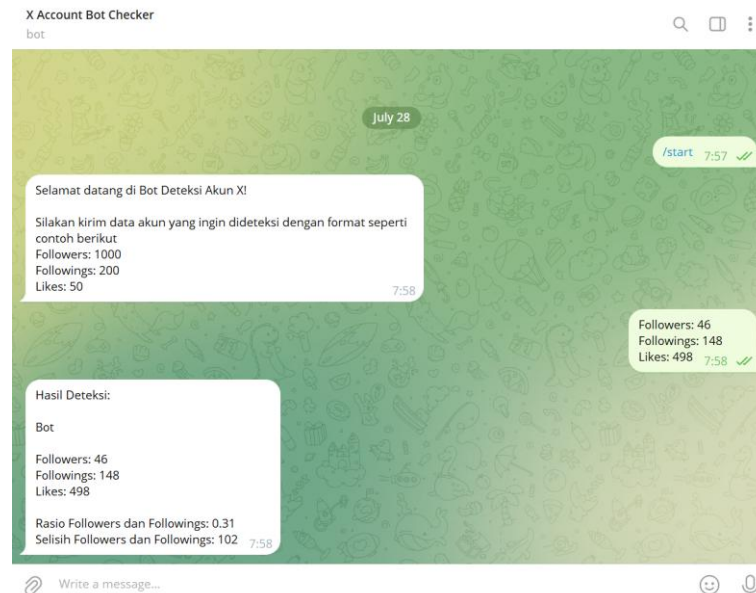
pengikut	Jumlah akun yang diikuti	Jumlah tweet yang disukai	Rasio followers dan followings	Selisih fol
0	2452	2792	5311	0.8782

Gambar 5. Implementasi pada Aplikasi Web

Pengguna diminta mengisi jumlah pengikut, jumlah akun yang diikuti, dan jumlah *tweet* yang disukai karena. Fitur turunan berupa rasio dan selisih jumlah pengikut dengan akun yang

diikuti akan otomatis dihitung oleh sistem. Setelah itu, sistem menampilkan hasil prediksi beserta rincian informasi profil berdasarkan data yang telah diisi oleh pengguna

Pada *chatbot* Telegram seperti yang divisualisasikan Gambar 6, konsepnya sama dengan aplikasi web. Perbedaannya dengan aplikasi web hanya pada pendekatan yang dapat dilakukan pengguna dalam melakukan deteksi. Pada *chatbot* Telegram, konsepnya sama seperti berkirim pesan singkat via Telegram.



Gambar 6. Implementasi pada Chatbot Telegram

5. Kesimpulan dan Saran

Model Random Forest, XGBoost, dan SVM RBF memberikan performa yang baik dalam prediksi akun bot pada media sosial X dengan nilai *f1-score* berkisar 98,9% hingga 99,1%. Dari ketiga model tersebut, XGBoost dengan *Bayesian optimization* memberikan performa terbaik dalam prediksi akun bot X pada *dataset* Cresci 2017 yang dibuktikan oleh nilai *f1-score* 99,1%. Berdasarkan nilai *f1-score*, *Bayesian optimization* meningkatkan performa model lebih efektif dibandingkan dengan *grid search* dan *random search*. Model terbaik disimpan dalam format pickle untuk diimplementasikan pada aplikasi web dan *chatbot* Telegram. Namun, performa menurun pada pengujian *dataset* eksternal yang tidak seimbang sehingga penelitian di masa mendatang dapat menggunakan *dataset* yang dibentuk dari kolom yang sama dengan Cresci 2017 melalui *scraping* dari media sosial X.

Referensi

- [1] A. Dehghan, K. Siuta, A. Skorupka, A. Dubey, A. Betlen, D. Miller, W. Xu, B. Kamiński, P. Prałat, "Detecting bots in social-networks using node and structural embeddings," *J Big Data*, vol. 10, no. 1, Des 2023, doi: 10.1186/s40537-023-00796-3.
- [2] J. Giroux, G. Ariyaratne, A. C. Nwala, C. Fanelli, "Unmasking social bots: how confident are we?," *EPJ Data Sci*, vol. 14, no. 1, Des 2025, doi: 10.1140/epjds/s13688-025-00536-y.
- [3] M. Aljabri, R. Zagrouba, A. Shaahid, F. Alnasser, A. Saleh, D. M. Alomari, "Machine learning-based social media bot detection: a comprehensive literature review," *Soc Netw Anal Min*, vol. 13, no. 1, Des 2023, doi: 10.1007/s13278-022-01020-5.

- [4] W. Dracewicz & M. Sepczuk, “Detecting Fake Accounts on Social Media Portals—The X Portal Case Study,” *Electronics (Switzerland)*, vol. 13, no. 13, Jul 2024, doi: 10.3390/electronics13132542.
- [5] H. Kurniawan, “Deteksi Twitter Bot Menggunakan Klasifikasi Decision Tree,” *Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan*, vol. 09, no. 01, hlm. 31–37, 2020.
- [6] D. M. Siddiqi & D. W. Albert, “Detecting Malicious Twitter Bots Using Machine Learning,” *International Journal of Information Technology and Computer Engineering*, 12 (3), doi: 10.62647/ijitce.2024.v12.i3.pp40-44.
- [7] S. Gurajala, J. S. White, B. Hudson, B. R. Voter, dan J. N. Matthews, “Profile Characteristics of Fake Twitter Accounts,” *Big Data Soc*, vol. 3, no. 2, Des 2016, doi: 10.1177/2053951716674236.
- [8] A. Ramalingaiah, S. Hussaini, dan S. Chaudhari, “Twitter Bot Detection Using Supervised Machine Learning,” *Journal of Physics: Conference Series*, IOP Publishing Ltd, Agu 2021. doi: 10.1088/1742-6596/1950/1/012006.
- [9] J. Luo dan C. Jin, “Fusing Content and Social Relationships: A Multi-Modal Heterogeneous Graph Transformer Approach For Social Bot Detection,” *EPJ Data Sci*, vol. 14, no. 1, Des 2025, doi: 10.1140/epjds/s13688-025-00583-5.
- [10] N. Jalal dan K. Z. Ghafoor, “Machine Learning Algorithms for Detecting and Analyzing Social Bots Using a Novel Dataset,” *ARO-The Scientific Journal of Koya University*, vol. 10, no. 2, hlm. 11–21, Des 2022, doi: 10.14500/aro.11032.
- [11] A. A. Zahra, W. Widyawan & S. Fauziati, “Development of Bot Detection Applications on Twitter Social Media Using Machine Learning with a Random Forest Classifier Algorithm,” *IJITEE (International Journal of Information Technology and Electrical Engineering)*, 4(2), 66-73, 2020.
- [12] O. Arranz-Escudero, L. Quijano-Sanchez, dan F. Liberatore, “Enhancing misinformation countermeasures: a multimodal approach to twitter bot detection,” *Soc Netw Anal Min*, vol. 15, no. 1, Des 2025, doi: 10.1007/s13278-025-01435-w.
- [13] R. G. Al-Azawi dan S. O. Al-Mamory, “Unsupervised Machine Learning for Bot Detection on Twitter: Generating and Selecting Features for Accurate Clustering,” *Inteligencia Artificial*, vol. 27, no. 73, hlm. 142–158, Jun 2024, doi: 10.4114/intartif.vol27iss73pp142-158.
- [14] F. K. Alarfaj, H. Ahmad, H. U. Khan, A. M. Alomair, N. Almusallam, dan M. Ahmed, “Twitter Bot Detection Using Diverse Content Features and Applying Machine Learning Algorithms,” *Sustainability (Switzerland)*, vol. 15, no. 8, Apr 2023, doi: 10.3390/su15086662.
- [15] L. Li, O. Vászrhelyi, dan B. Vedres, “Social bots spoil activist sentiment without eroding engagement,” *Sci Rep*, vol. 14, no. 1, Des 2024, doi: 10.1038/s41598-024-74032-0.
- [16] M. Kolomeets, O. Tushkanova, V. Desnitsky, L. Vitkova, dan A. Chechulin, “Experimental Evaluation: Can Humans Recognise Social Media Bots?,” *Big Data and Cognitive Computing*, vol. 8, no. 3, Mar 2024, doi: 10.3390/bdcc8030024.
- [17] R. Kenny, B. Fischhoff, A. Davis, dan C. Canfield, “Improving Social Bot Detection Through Aid and Training,” *Hum Factors*, vol. 66, no. 10, hlm. 2323–2344, Okt 2024, doi: 10.1177/00187208231210145.

- [18] A. A. Darem, A. A. Alhashmi, M. H. Alanazi, A. F. Alanezi, Y. Said, L. A. Darem, M. M. Hussain, "Cybersecurity in social networks: An ensemble model for Twitter bot detection," *International Journal of Advanced and Applied Sciences*, vol. 11, no. 11, hlm. 130–141, Nov 2024, doi: 10.21833/ijaas.2024.11.014.
- [19] A. Shevtsov, C. Tzagkarakis, D. Antonakaki, dan S. Ioannidis, "Identification of Twitter Bots Based on an Explainable Machine Learning Framework: The US 2020 Elections Case Study," *AAAI Conference on Artificial Intelligence*, 2022. [Daring]. Tersedia pada: www.aaai.org
- [20] R. Shukla, A. Sinha, dan A. Chaudhary, "TweezBot: An AI-Driven Online Media Bot Identification Algorithm for Twitter Social Networks," *Electronics (Switzerland)*, vol. 11, no. 5, Mar 2022, doi: 10.3390/electronics11050743.
- [21] E. Alothali, K. Hayawi, dan H. Alashwal, "Hybrid feature selection approach to identify optimal features of profile metadata to detect social bots in Twitter," *Soc Netw Anal Min*, vol. 11, no. 1, Des 2021, doi: 10.1007/s13278-021-00786-4.
- [22] M. Heidari, J. H. Jones, dan O. Uzuner, "Deep Contextualized Word Embedding for Text-based Online User Profiling to Detect Social Bots on Twitter," dalam *IEEE International Conference on Data Mining Workshops, ICDMW*, IEEE Computer Society, Nov 2020, hlm. 480–487. doi: 10.1109/ICDMW51313.2020.00071.
- [23] S. Cresci, A. Spognardi, M. Petrocchi, M. Tesconi, dan R. Di Pietro, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race," dalam *26th International World Wide Web Conference 2017, WWW 2017 Companion*, International World Wide Web Conferences Steering Committee, 2017, hlm. 963–972. doi: 10.1145/3041021.3055135.