AITI: Jurnal Teknologi Informasi, Volume 23 No. 2 Juni 2026, 305-318
ISSN 1693-8348 E-ISSN 2615-7128
DOI: 10.24246/aiti.v23.i2.305-318

JURNAL TEKNOLOGI INFORMASI

Prediksi kebangkrutan perusahaan menggunakan metode klasifikasi: Studi kasus pada industri

Ibrahim*, Hendry
Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
Jl. Dr. O. Notohamidjojo 1-10, Salatiga, Indonesia
Email: *672021178@student.uksw.edu

Riwayat Artikel

Diterima:	Direvisi:	Disetujui:
01-01-2024	01-02-2024	01-03-2024

Abstract

Corporate bankruptcy prediction is a crucial aspect of the financial sector because it can significantly affect investors, creditors, company management, and other stakeholders in making strategic decisions. This study aims to develop an accurate bankruptcy prediction model using the XGBoost algorithm optimized through hyperparameter tuning. In addition, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to address data imbalance by increasing the representation of the minority class. The model was also tested on two large-scale datasets (Taiwan and US) to assess its performance consistency and generalization capability. The results show that XGBoost with hyperparameter tuning achieves the best performance, with an accuracy of 98.94%, a precision of 0.98, and a recall of 1.0. Furthermore, the model demonstrated stable performance without indications of overfitting. These findings confirm that XGBoost with hyperparameter tuning can provide accurate, consistent, and reliable bankruptcy predictions and have strong potential for broader industrial implementation and larger-scale applications.

Keywords: *bankruptcy prediction, data imbalance, overfitting, SMOTE, model optimization, hyperparameter tuning*

Abstrak

Prediksi kebangkrutan perusahaan merupakan aspek penting dalam bidang keuangan karena dapat memberikan dampak signifikan terhadap investor, kreditur, manajemen perusahaan, serta pemangku kepentingan lainnya dalam pengambilan keputusan strategis. Penelitian ini bertujuan untuk mengembangkan model prediksi kebangkrutan perusahaan yang akurat menggunakan algoritma *XGBoost* yang dioptimalkan melalui proses penyesuaian *hiperparameter (hyperparameter tuning)*. Selain itu, penelitian ini juga menerapkan metode *Synthetic Minority Over-sampling Technique* (SMOTE) untuk mengatasi tantangan ketidakseimbangan data dengan meningkatkan representasi kelas minoritas. Model yang dikembangkan diuji pada dua *dataset* berskala besar (Taiwan dan US) untuk mengevaluasi konsistensi kinerja serta kemampuan generalisasi model. Hasil penelitian menunjukkan bahwa model *XGBoost* dengan *hyperparameter tuning* mampu menghasilkan performa terbaik dengan tingkat akurasi sebesar 98,94%, *precision* sebesar 0,98, dan *recall* sebesar 1,0. Selain itu, hasil pengujian menunjukkan bahwa model tersebut memiliki kinerja yang konsisten tanpa indikasi *overfitting*. Hasil pendekatan ini membuktikan bahwa *XGBoost* dengan penyesuaian *hiperparameter* mampu memberikan

prediksi kebangkrutan yang akurat, konsisten, dan dapat diandalkan untuk diterapkan dalam berbagai konteks industri serta pada skala penggunaan yang lebih luas.

Kata kunci: prediksi kebangkrutan, ketidakseimbangan data, *overfitting*, SMOTE, optimasi model, *hyperparameter tuning*

Pendahuluan

Prediksi kebangkrutan perusahaan penting karena berdampak signifikan pada investor, kreditor, karyawan, dan ekonomi secara keseluruhan [1]. Dengan kemampuan memprediksi kebangkrutan secara akurat, berbagai pihak dapat memitigasi risiko yang dapat mengganggu stabilitas ekonomi. Banyak perusahaan yang sebelumnya stabil kini menghadapi risiko kebangkrutan akibat persaingan ketat, perubahan kebijakan ekonomi, dan gangguan rantai pasokan [2]. Bahkan perusahaan stabil berisiko bangkrut karena krisis ekonomi, perubahan teknologi, dan pergeseran preferensi konsumen [3].

Data keuangan yang digunakan dalam prediksi kebangkrutan umumnya merupakan data yang tidak seimbang (*imbalanced*). Jumlah perusahaan yang bangkrut jauh lebih sedikit dibandingkan dengan perusahaan yang tidak bangkrut, menyebabkan model *machine learning* cenderung bias terhadap kelas mayoritas dan mengurangi akurasi dalam mendeteksi perusahaan yang berisiko bangkrut [4]. Model yang terlalu kompleks sering kali menyesuaikan diri secara berlebihan terhadap data pelatihan, sehingga kehilangan kemampuan untuk menggeneralisasi pola pada data baru (*overfitting*) [5].

Tabel 1 Tabel perbedaan dengan penelitian sebelumnya

Penelitian	Fokus Utama	Analisis *Overfitting*	*Balancing Data*
U. Ali et al. [6]	Perbandingan algoritma untuk memprediksi kebangkrutan	Tidak	Tidak
Iparraguirre & Cabanillas [7]	Evaluasi performa 16 algoritma dan pembuktian efektivitas ensemble	Tidak	Tidak
P. Gnip et al. [8]	Pengaruh metode penanganan ketidakseimbangan data	Tidak	Ya
Penelitian ini	Optimasi model *XGBoost*, dua *dataset* (Taiwan & US), evaluasi komprehensif	Ya	Ya

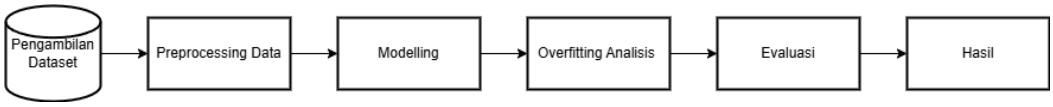
Tabel 1 membandingkan beberapa penelitian terkait prediksi kebangkrutan perusahaan. Sebagian besar studi sebelumnya hanya berfokus pada perbandingan performa algoritma tanpa mempertimbangkan masalah *overfitting* maupun ketidakseimbangan data. Penelitian oleh U. Ali et al. dan oleh Iparraguirre & Cabanillas mengevaluasi sejumlah algoritma, namun tidak menerapkan teknik *rebalancing* atau analisis *overfitting*. Sementara itu, studi yang dilakukan oleh P.Gnip et al menekankan pentingnya metode *rebalancing*, tetapi tidak membahas *overfitting*. Berbeda dengan studi sebelumnya, penelitian ini mengusulkan pendekatan terintegrasi menggunakan *XGBoost* dengan *hyperparameter tuning*, teknik *Synthetic Minority Oversampling Technique* (SMOTE) untuk

penyeimbangan data, serta analisis *overfitting* pada dua *dataset* berbeda (Taiwan dan US) untuk menghasilkan prediksi yang lebih andal dan dapat digeneralisasi.

Penelitian ini bertujuan untuk membangun model prediksi kebangkrutan perusahaan yang akurat dengan memanfaatkan algoritma *XGBoost* yang dioptimalkan melalui *hyperparameter tuning* serta penanganan ketidakseimbangan data menggunakan SMOTE. Dengan menguji model pada dua *dataset* berbeda (Taiwan dan US), penelitian ini diharapkan mampu menghasilkan model yang tidak hanya unggul dalam akurasi, tetapi juga mampu melakukan generalisasi dengan baik. Hasil dari penelitian ini penting sebagai kontribusi dalam pengembangan sistem pendukung keputusan di bidang keuangan, khususnya untuk deteksi dini risiko kebangkrutan yang dapat membantu dalam mengambil langkah preventif.

Metode Penelitian

Penelitian ini menggunakan bahasa pemrograman *Python* dan dijalankan dalam *Google Collaboratory*. Tahapan penelitian ini yaitu pengambilan *dataset*, *preprocessing* data, *modelling*, *overfitting* analisis, dan evaluasi. Tahapan penelitian ini tercantum pada Gambar 1.



Gambar 1 *Workflow* tahapan penelitian

A. Pengambilan *Dataset*

Dataset pertama adalah data *Taiwan company* yang diambil dari situs UC *Irvine Machine Learning Repository* [9]. *Dataset* kedua adalah data US *company* yang diambil dari situs *Kaggle* [10]. *Dataset* pertama sebagai *dataset* utama pada penelitian ini terdiri dari 95 fitur dan 13.198 *entries*, sedangkan *dataset* kedua sebagai *dataset* pembantu terdiri dari 21 fitur dan 78.682 *entries*. *Dataset* pertama merupakan data perusahaan yang dikumpulkan dari *Taiwan Economic Journal* dari tahun 1999 hingga 2009. Kebangkrutan perusahaan ditentukan berdasarkan peraturan bisnis Bursa Efek Taiwan. Sementara itu, *dataset* kedua merupakan *dataset* prediksi kebangkrutan yang berhubungan dengan perusahaan publik Amerika yang terdaftar di *New York Stock Exchange* (NYSE) dan Nasdaq. *Dataset* ini mencakup data akuntansi dari 8.262 perusahaan berbeda yang dicatat selama periode 1999 hingga 2018.

B. *Preprocessing Data*

Langkah pertama dalam penelitian ini adalah *preprocessing* data, yang dimulai dengan impor pustaka Python seperti *pandas*, *numpy*, dan *scikit-learn* untuk manipulasi data dan analisis lebih lanjut. Pengecekan terhadap nilai yang hilang (*missing values*) dilakukan, dan jika ditemukan, perbaikan dilakukan melalui penghapusan atau *imputation*, tergantung pada tingkat keparahan dan distribusi data

[11]. Proses ini bertujuan untuk memastikan integritas *dataset* yang akan digunakan pada tahap berikutnya.

Masalah ketidakseimbangan kelas pada *dataset* ini diatasi dengan menerapkan pendekatan SMOTE. Beberapa penelitian terbaru telah mengembangkan varian SMOTE untuk meningkatkan performa *oversampling* pada *dataset* yang sangat tidak seimbang [12]. Penerapan metode ini bertujuan agar algoritma *machine learning* tidak bias terhadap kelas mayoritas, sehingga dapat meningkatkan akurasi prediksi pada kelas minoritas.

Tahap akhir dari *preprocessing* adalah normalisasi data menggunakan *StandardScaler*. Metode ini digunakan untuk mengubah fitur numerik ke dalam skala yang sama, dengan mentransformasikan data agar memiliki nilai rata-rata (*mean*) nol dan standar deviasi satu [13].

C. *Modelling*

Model utama yang digunakan dalam penelitian ini adalah *XGBoost* dan *Random Forest*. Kedua model ini dipilih karena memiliki performa tinggi, kemampuan generalisasi yang baik, serta keunggulan dalam menangani *dataset* yang besar dan kompleks. Model *XGBoost* dalam penelitian ini dirumuskan sebagai kumpulan *base learners* atau model dasar yang dinyatakan dalam Persamaan (1).

$$\hat{y}_i = \sum_{t=1}^{m} f_t(x_i) \tag{1}$$

di mana $\hat{y}_i$ merupakan hasil prediksi akhir untuk observasi ke-i, $f_t(x_i)$ adalah fungsi prediktor pada iterasi ke-t, dan m adalah jumlah total model dasar yang digunakan. Setiap fungsi f_t dibentuk untuk meminimalkan fungsi objektif yang dinyatakan dalam Persamaan (2).

$$L(t) = \sum_{i=1}^{n} \ell\big(y_i, \hat{y}_i(t-1)\big) + f_t(x_i) + \Omega(f_t) \tag{2}$$

dengan $L(t)$ adalah fungsi kerugian total pada iterasi ke-t, y_i adalah nilai aktual, $\hat{y}_i(t-1)$ adalah prediksi dari model hingga iterasi sebelumnya, dan ℓ menyatakan fungsi kerugian (*loss function*) seperti *log loss* atau *mean squared error* yang digunakan untuk menghitung selisih antara prediksi dan nilai aktual. Sementara itu, $\Omega(f_t)$ adalah fungsi regularisasi yang mengontrol kompleksitas model untuk mencegah *overfitting*. Seluruh proses pembentukan model dilakukan secara bertahap dan berulang, di mana setiap model baru ditambahkan untuk memperbaiki kesalahan sebelumnya, sehingga diperoleh model akhir yang memberikan hasil prediksi sebaik mungkin. Salah satu kelebihan *XGBoost* adalah kemampuannya dalam menangani *dataset* besar dan kompleks dengan kecepatan tinggi, serta memiliki mekanisme bawaan untuk mengatasi *overfitting*, seperti *regularization* L1 dan L2 [14].

Random Forest merupakan algoritma pembelajaran *ensemble* yang memanfaatkan pendekatan *bootstrap aggregating* (*bagging*), yaitu dengan membangun sejumlah besar pohon keputusan yang dilatih secara independen menggunakan subset acak dari data pelatihan. Tujuan dari pendekatan ini adalah untuk mengurangi varians prediksi dan meningkatkan akurasi model secara keseluruhan. Prediksi akhir dari model *Random Forest* diperoleh dengan menggabungkan hasil prediksi dari seluruh pohon yang terbentuk. Untuk kasus klasifikasi, penggabungan dilakukan melalui pendekatan *majority voting*, di mana kelas yang memperoleh suara terbanyak dari seluruh pohon akan menjadi prediksi akhir. Secara matematis, prediksi akhir untuk input x seperti dirumuskan dalam Persamaan (3).

$$\hat{y} = \text{mode}\{h_1(x), h_2(x), \dots, h_K(x)\} \quad (3)$$

di mana h_K menyatakan hasil prediksi dari pohon ke-k, dan K adalah jumlah total pohon dalam hutan. Rumus ini menunjukkan bahwa prediksi akhir y ditentukan oleh nilai modus (kelas terbanyak) dari seluruh prediksi yang diberikan oleh masing-masing pohon keputusan. Kelebihan utama *Random Forest* adalah kemampuannya dalam menangani data dengan jumlah fitur yang besar dan beragam, serta kemampuannya memberikan *feature importance* untuk analisis interpretabilitas model [15]. Meskipun demikian, kelemahan *Random Forest* terletak pada potensi *overfitting*, terutama jika *dataset* memiliki terlalu banyak pohon keputusan yang dalam (*deep trees*) atau mengandung *outliers* yang signifikan.

Pelatihan model *machine learning* dilakukan melalui proses *oversampling*, validasi silang, dan optimasi *hyperparameter* untuk mengevaluasi performa pada data mentah. Data dibagi menjadi set pelatihan (80%) dan pengujian (20%) menggunakan *stratified sampling*, dengan teknik SMOTE diterapkan pada set pelatihan untuk menangani ketidakseimbangan kelas. Optimasi *hyperparameter* menggunakan *RandomizedSearchCV* dengan metrik *F1-score*, sedangkan validasi silang menggunakan *StratifiedKFold* untuk memastikan generalisasi model [16]. Model *Random Forest* dan *XGBoost* digunakan tanpa seleksi fitur karena keduanya efektif dalam menangani *dataset* kompleks. *Random Forest* memiliki kemampuan menghitung *feature importance*, sementara *XGBoost* unggul dalam efisiensi, dukungan *parallelization*, serta fitur bawaan untuk mengatasi *overfitting* [17].

D. *Overfitting Analysis*

Overfitting analysis dilakukan dengan menggunakan *learning curves* dan *Receiver Operating Characteristic* (ROC) untuk menganalisis performa dua model utama, yaitu *Random Forest* dan *XGBoost*. *Learning curves* menunjukkan perbandingan antara skor akurasi pada data pelatihan dan validasi terhadap ukuran *dataset*, yang membantu dalam mengidentifikasi apakah model mengalami *overfitting* atau *underfitting* [18]. Jika skor pada data pelatihan jauh lebih tinggi

dibandingkan data validasi, maka ini mengindikasikan bahwa model terlalu menyesuaikan diri dengan data pelatihan dan kehilangan kemampuan untuk menggeneralisasi pada data baru. Sebaliknya, ROC digunakan untuk mengukur kemampuan model dalam membedakan antara kelas positif dan negatif dengan menampilkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai ambang keputusan.

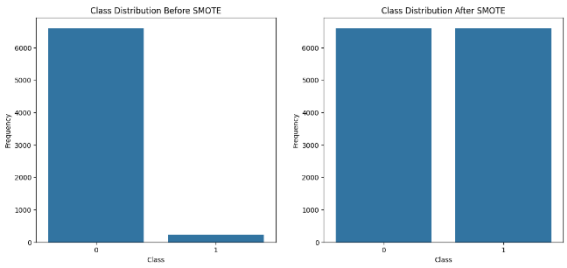
Pada hasil analisis, *Random Forest* menunjukkan potensi *overfitting* dalam beberapa skenario, terutama pada data pelatihan yang sangat bervariasi. Di sisi lain, *XGBoost* relatif lebih stabil berkat mekanisme bawaan seperti regularisasi L1 dan L2 yang membantu mengendalikan kompleksitas model. ROC untuk kedua model menunjukkan nilai *Area Under Curve* (AUC) yang tinggi, mengindikasikan performa prediktif yang baik. Namun, pola dalam *learning curves* membantu dalam mengidentifikasi kebutuhan pengaturan lebih lanjut pada model *Random Forest*, seperti pengurangan kedalaman pohon atau penyesuaian parameter lainnya untuk mengurangi kompleksitas dan meningkatkan kemampuan generalisasi [19].

E. Evaluasi

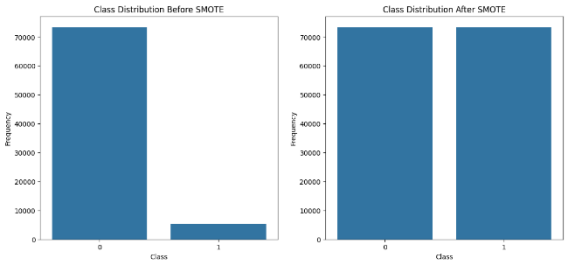
Evaluasi model dilakukan dengan menggunakan laporan klasifikasi yang mencakup *accuracy*, *precision*, *recall*, dan *F1-score* untuk mengukur performa secara menyeluruh. *Accuracy* menunjukkan persentase prediksi yang benar dari keseluruhan data, sedangkan *precision* mengukur ketepatan model dalam memprediksi kelas positif. *Recall* menilai kemampuan model dalam mendeteksi seluruh kasus positif yang sebenarnya, dan *F1-score* digunakan untuk menyeimbangkan antara *precision* dan *recall*, terutama dalam situasi di mana data tidak seimbang. Selain metrik tersebut, *confusion matrix* juga digunakan untuk menggambarkan distribusi hasil prediksi model terhadap data aktual, sehingga membantu memastikan bahwa model memiliki performa yang baik pada data pengujian dan tidak bias terhadap salah satu kelas.

Hasil dan Pembahasan

Pada tahap awal sebelum pemodelan, dilakukan proses *checking* data untuk mengidentifikasi nilai *null* dan memastikan kualitas data yang akan digunakan. Setelah itu, data dibagi menjadi fitur (X) dan target (y), kemudian dilakukan normalisasi menggunakan *StandardScaler* untuk menghindari ketidakseimbangan skala antar fitur yang dapat memengaruhi performa model. Untuk menangani ketidakseimbangan kelas dalam data target, diterapkan metode SMOTE, yaitu teknik *oversampling* yang menambahkan data sintetis pada kelas minoritas agar proporsi antar kelas menjadi lebih seimbang dan model tidak bias terhadap kelas mayoritas.



Gambar 2 *Before and after* SMOTE pada *dataset* Taiwan



Gambar 3 *Before and after* SMOTE pada *dataset* US

Perbandingan grafik data sebelum dan sesudah penerapan SMOTE menunjukkan peningkatan jumlah sampel pada kelas minoritas, yang bertujuan untuk meningkatkan kemampuan model dalam mengenali pola pada data tersebut. Proses ini dilakukan untuk meminimalkan bias terhadap kelas mayoritas dan memastikan bahwa model memiliki performa yang adil pada kedua kelas. Setelah melalui rangkaian proses *preprocessing* data, penelitian dilanjutkan dengan penjabaran hasil evaluasi dari setiap model yang telah diuji, untuk mengetahui efektivitas pendekatan yang diterapkan dalam menangani data tidak seimbang.

Tabel 2 Tabel hasil setiap model pada *Taiwan dataset*

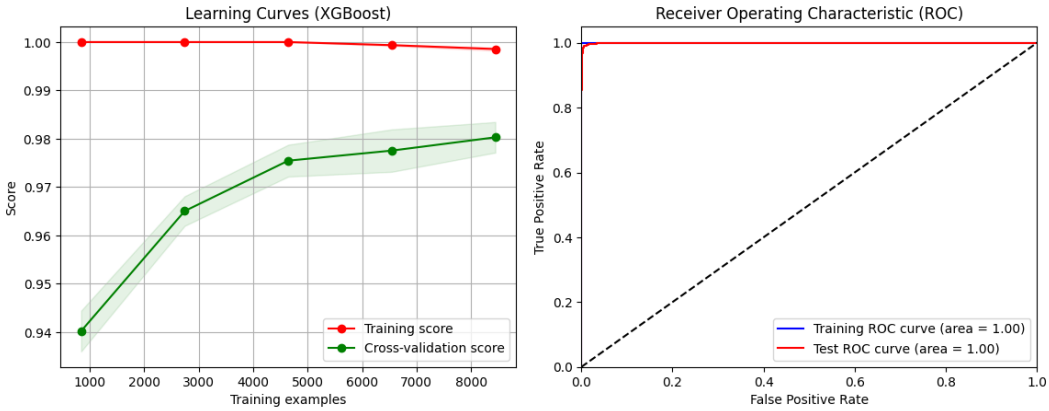
Nama Model	Model Score (%)	Precision	Recall	F1 Score
XGBoost	98,75	0,99	0,99	0,99
Random Forest	98,48	0,98	0,98	0,98

Tabel 2 menunjukkan hasil evaluasi algoritma *XGBoost* dan *Random Forest* pada *Taiwan dataset*. Model *XGBoost* memberikan performa terbaik dengan akurasi 98,75%, serta nilai *precision*, *recall*, dan *F1-score* masing-masing sebesar 0,99. Model *Random Forest* menyusul dengan akurasi 98,48% dan metrik evaluasi lainnya yang juga tinggi dan konsisten. Hasil ini menunjukkan bahwa *XGBoost* dan *Random Forest* memiliki kemampuan yang unggul dalam klasifikasi kebangkrutan pada *Taiwan dataset* tanpa memerlukan proses seleksi fitur tambahan.

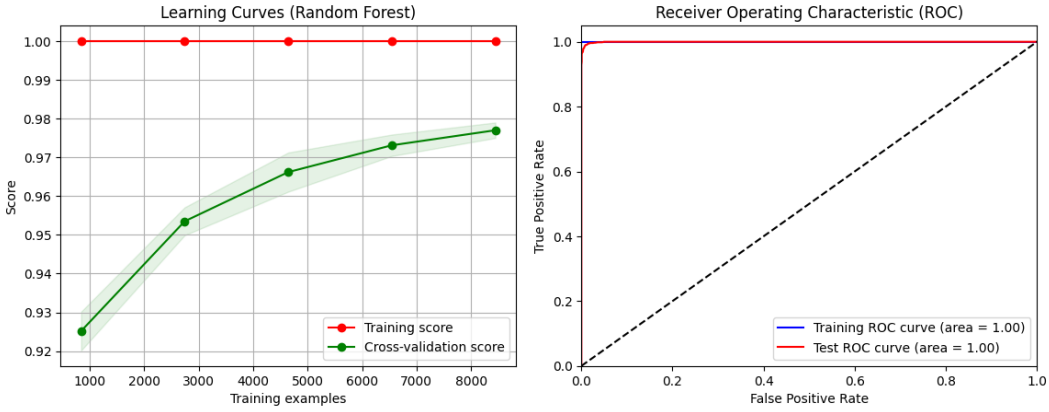
Tabel 3 Tabel hasil setiap model pada *US dataset*

Nama Model	Model Score (%)	Precision	Recall	F1 Score
XGBoost	88,09	0,88	0,88	0,88
Random Forest	89,52	0,90	0,90	0,90

Tabel 3 menyajikan hasil evaluasi algoritma *XGBoost* dan *Random Forest* pada *US dataset*. Model *Random Forest* menunjukkan performa tertinggi dengan akurasi 89,52%, serta nilai *precision*, *recall*, dan *F1-score* masing-masing sebesar 0,90. Sementara itu, *XGBoost* mencatat akurasi sebesar 88,09% dengan nilai *precision*, *recall*, dan *F1-score* yang identik yaitu 0,88. Hasil ini menunjukkan bahwa perbedaan performa antar model pada *US dataset* relatif kecil, meskipun *Random Forest* memberikan hasil terbaik secara keseluruhan.

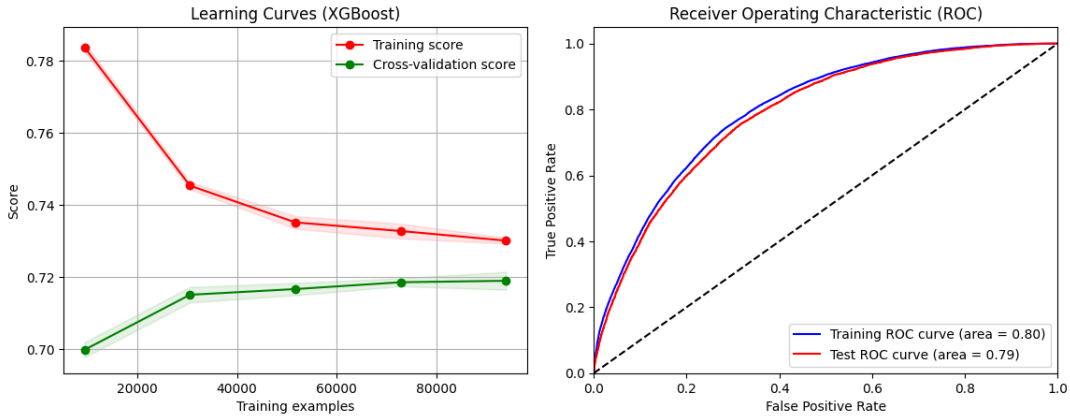


Gambar 4 *Learning curves* dan *ROC curve* pada *XGBoost Taiwan dataset*

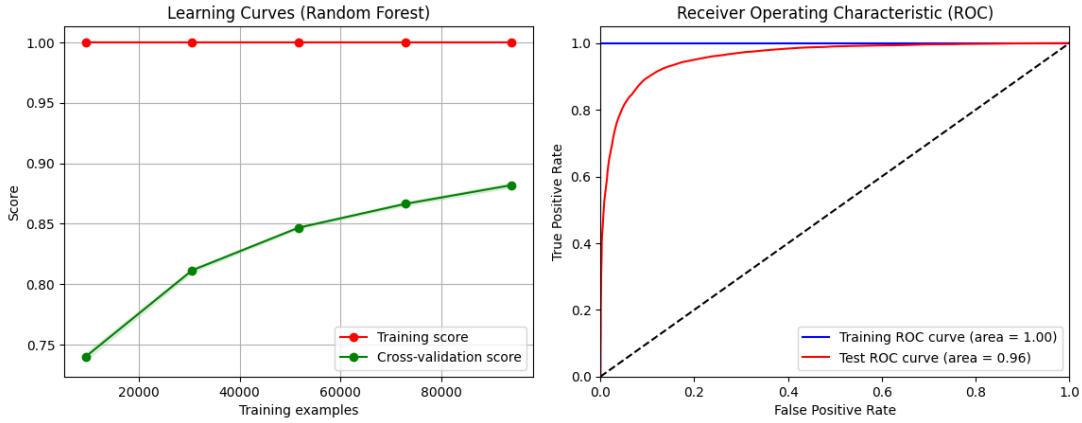


Gambar 5 *Learning curves* dan ROC *curve* pada *Random Forest Taiwan dataset*

Gambar 4 dan Gambar 5 menunjukkan bahwa model *XGBoost* dan *Random Forest* tidak mengalami *overfitting*, ditandai dengan skor pelatihan yang mendekati 1,00 dan skor validasi silang yang stabil, masing-masing sekitar 0,98 dan 0,97. Hal ini menandakan kemampuan generalisasi yang baik dari kedua model. ROC *curve* pada data pelatihan dan pengujian menunjukkan nilai AUC sebesar 1,00, yang mengindikasikan performa klasifikasi yang sangat baik. Meskipun *Random Forest* menunjukkan sedikit perbedaan antara skor pelatihan dan validasi, potensi *overfitting* tetap perlu diwaspadai, terutama pada data yang kompleks atau jumlah pohon yang terlalu banyak. Untuk meningkatkan kepercayaan terhadap kestabilan kedua model, dilakukan pengujian tambahan menggunakan *dataset* pembanding.



Gambar 6 *Learning curves* dan ROC *curve* pada *XGBoost US dataset*



Gambar 7 *Learning curves* dan ROC *curve* pada *Random Forest US dataset*

Gambar 6 dan Gambar 7 menunjukkan perbandingan performa model *XGBoost* dan *Random Forest* pada *dataset* pembanding. Model *XGBoost* tidak mengalami *overfitting*, ditunjukkan dengan perbedaan skor pelatihan dan *cross-validation* yang mengecil seiring bertambahnya data pelatihan. Kedua kurva stabil dengan skor yang hampir sama, menandakan bahwa penambahan data tidak memberikan peningkatan signifikan pada performa. Nilai AUC sebesar 0,80 pada data pelatihan dan 0,79 pada data uji memperkuat bukti kemampuan generalisasi model ini. Sebaliknya, model *Random Forest* menunjukkan potensi *overfitting*, dengan skor pelatihan tetap di angka 1,00 dan skor validasi hanya sekitar 0,90, yang mengindikasikan bahwa model terlalu menyesuaikan diri pada data pelatihan dan kurang mampu menggeneralisasi ke data baru. Berdasarkan hasil tersebut, penelitian ini berfokus pada pengembangan model *XGBoost* yang terbukti lebih efektif dalam menangani kompleksitas data. Untuk meningkatkan akurasi prediksi, percobaan selanjutnya akan difokuskan pada proses *hyperparameter tuning* terhadap model *XGBoost*.

Tabel 4 Tabel hasil model *XGBoost* dengan penyesuaian *hyperparameter tuning*

Nama Model	Model *Score* (%)	*Precision*	*Recall*	F1 *Score*	ROC-AUC *Score*
XGBoost with Hyper-parameter Tuning – Taiwan	98,94	0,98	1,00	0,99	0,99
XGBoost with Hyper-parameter Tuning – US	90,15	0,88	0,93	0,90	0,90

Tabel 4 menunjukkan hasil model *XGBoost* dengan *hyperparameter tuning* untuk *Taiwan dataset dan US dataset*, di mana akurasi mencapai 98,94% dan 90,15%. Model pada *Taiwan dataset* memiliki *precision* 0,98, *recall* 1,00, dan *F1-score* 0,99. ROC-AUC mendekati 1, menunjukkan performa hampir sempurna. Model pada *US dataset* mencapai akurasi sebesar 90,15%, dengan nilai *precision* sebesar 0,88, *recall* 0,93, dan *F1-score* 0,90. Skor ROC-AUC yang mendekati 1 menandakan kemampuan model dalam membedakan kedua kelas secara optimal.

Model *Extreme Gradient Boosting* (*XGBoost*) dilatih menggunakan serangkaian nilai parameter yang diuji dalam *param_grid* melalui metode *RandomizedSearchCV*, yang mengombinasikan *cross-validation* berlapis (*StratifiedKFold*) sebanyak 10 kali dengan pencarian acak terhadap kombinasi parameter. Proses ini bertujuan untuk menemukan konfigurasi *hyperparameter* terbaik yang menghasilkan performa model paling optimal. Adapun *hyperparameter* utama yang disesuaikan dalam proses ini meliputi: *n_estimators*, yaitu jumlah pohon keputusan yang dibangun oleh model; *learning_rate*, yang mengontrol laju pembelajaran pada setiap iterasi; *max_depth*, yakni kedalaman maksimum dari setiap pohon; *min_child_weight*, yang menentukan bobot minimum dari jumlah sampel yang dibutuhkan untuk melakukan pemisahan pada node; *gamma*, sebagai nilai ambang batas minimal untuk pemisahan lebih lanjut; *subsample*, yaitu proporsi sampel pelatihan yang digunakan untuk membangun setiap pohon; *colsample_bytree*, yang merupakan proporsi fitur yang dipilih secara acak untuk setiap pohon; serta *scale_pos_weight*, yang berfungsi untuk menangani ketidakseimbangan kelas dalam data target.

Tabel 5 *Confusion matrix XGBoost with hyperparameter tuning Taiwan dataset*

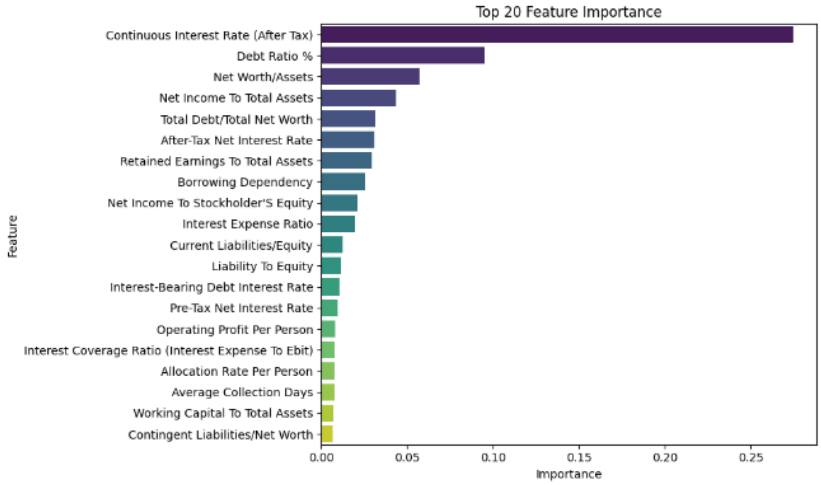
		Predicted	
		Positive	*Negative*
Actual	*Positive*	1293	27
	Negative	1	1319

Tabel 6 *Confusion matrix XGBoost with hyperparameter tuning US dataset*

		Predicted	
		Positive	*Negative*
Actual	*Positive*	12854	1839
	Negative	1056	13636

Tabel 5 menunjukkan hasil *confusion matrix* model *XGBoost* pada *Taiwan dataset*, di mana dari 1.320 data kelas 0, sebanyak 1.293 diklasifikasikan dengan

benar sebagai *True Negative* dan 27 sebagai *False Positive*. Untuk 1.320 data kelas 1, model berhasil mengklasifikasikan 1.319 sebagai *True Positive* dan hanya satu data sebagai *False Negative*. Sementara itu, Tabel 6 menyajikan hasil *confusion matrix XGBoost* dengan *hyperparameter tuning* pada *US dataset*, yang juga menunjukkan performa klasifikasi yang baik. Dari 14.693 data kelas 0, sebanyak 12.854 diklasifikasikan dengan benar sebagai *True Negative*, dan 1.839 diklasifikasikan salah sebagai *False Positive*. Untuk 14.692 data kelas 1, model mampu mengklasifikasikan 13.636 data sebagai *True Positive*, dan 1.056 sebagai *False Negative*. Hasil ini menunjukkan bahwa model memiliki kemampuan yang seimbang dalam mengenali kedua kelas dan mampu meminimalkan kesalahan klasifikasi, baik pada *dataset* Taiwan maupun US.



Gambar 8 *Top 20 feature importance* pada *XGBoost*

Gambar 8 menunjukkan *top 20 feature importance* pada model prediksi kebangkrutan perusahaan. Fitur *Continuous Interest Rate* (*After Tax*) memiliki nilai penting tertinggi, tetapi kurang dari 30%, menandakan tidak ada fitur dominan. Fitur seperti *Debt Ratio %* dan *Net Worth/Assets* juga berperan penting tanpa perbedaan signifikan. Ini menunjukkan bahwa model tetap efektif tanpa seleksi fitur, karena tidak ada fitur yang terlalu mendominasi.

Berdasarkan hasil penelitian ini, model *XGBoost* yang dioptimalkan dengan *hyperparameter tuning* menunjukkan performa yang sangat baik, dengan akurasi mencapai 98,94% dan skor ROC-AUC sebesar 0,99, yang menunjukkan kemampuan klasifikasi yang sangat baik. Performa yang sangat baik dari *XGBoost* juga tercermin dalam nilai *precision*, *recall*, dan *F1 score* yang tinggi, masing-masing mencapai 0,98, 1,00, dan 0,99, menunjukkan bahwa model ini sangat baik dalam mendeteksi kebangkrutan perusahaan tanpa kesalahan klasifikasi yang signifikan.

Dalam perbandingan dengan penelitian sebelumnya, seperti yang dilakukan oleh Muslim et al. [20], model *XGBoost* dalam penelitian mereka hanya mencapai akurasi 97%, sementara penelitian ini mencatatkan akurasi lebih tinggi dengan

XGBoost yang mencapai 98,94%. Hasil ini menunjukkan bahwa *XGBoost* dengan *hyperparameter tuning* dapat mengungguli model yang digunakan dalam studi sebelumnya, baik dalam hal akurasi maupun dalam kemampuan mengatasi masalah ketidakseimbangan kelas.

Simpulan

Penelitian ini bertujuan untuk mengembangkan model prediksi kebangkrutan perusahaan dengan menggunakan algoritma *XGBoost* yang dioptimalkan melalui *hyperparameter tuning*. Berdasarkan hasil eksperimen yang dilakukan, model *XGBoost* yang diterapkan menunjukkan performa terbaik dengan akurasi mencapai 98,94%, *precision* 0,98, *recall* 1,0, dan skor F1 0,99. Hal ini mengindikasikan bahwa seluruh fitur yang digunakan dalam model ini memberikan kontribusi yang signifikan dalam memprediksi kebangkrutan perusahaan. Temuan ini juga menunjukkan bahwa model *XGBoost* dengan *hyperparameter tuning* mampu menangani tantangan ketidakseimbangan data dengan baik, yang diperkuat dengan penerapan SMOTE untuk meningkatkan representasi kelas minoritas.

Daftar Pustaka

[1] P. R. Masdiantini and N. M. S. Warasniasih, "Laporan Keuangan dan Prediksi Kebangkrutan Perusahaan," *JIA (Jurnal Ilmiah Akuntansi)*, vol. 5, no. 1, pp. 196–220, 2020.

[2] G. K. Putra, A. Khafid, and E. S. Hastuti, "Analisis Kebangkrutan Dengan Metode Springate (Studi Kasus Pada Pt. Pp Properti Tbk Tahun 2020-2022)," *Solusi*, vol. 21, no. 3, pp. 270–277, 2023.

[3] M. Sitorus and S. M. Yulita, "Analisis Potensi Kebangkrutan Pada PT. Prima Mulia Engineering dengan Metode Altman Z-Score untuk Periode 2017-2020," *Jurnal Teknologi dan Manajemen*, vol. 21, no. 1, pp. 1–8, 2023.

[4] A. Nugroho and E. Rilvani, "Penerapan Metode Oversampling SMOTE Pada Algoritma Random Forest Untuk Prediksi Kebangkrutan Perusahaan.," *Techno. com*, vol. 22, no. 1, 2023.

[5] A. C. N. Heryanto, "Analisis Prediksi Kebangkrutan Perusahaan Dengan Model Grover," *COMPETITIVE Jurnal Akuntansi Dan Keuangan*, vol. 4, no. 2, pp. 54–65, 2020.

[6] U. Ali, S. Fahad, and A. Ali, "Machine Learning Approaches for Predicting Company Bankruptcy: A Comparative Study," 2024.

[7] O. Iparraguirre-Villanueva and M. Cabanillas-Carbonell, "Predicting business bankruptcy: A comparative analysis with machine learning models," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 3, p. 100375, 2024.

[8] P. Gnip, R. Kanász, M. Zoričak, and P. Drotár, "An experimental survey of imbalanced learning algorithms for bankruptcy prediction," *Artif Intell Rev*, vol. 58, no. 4, p. 104, 2025.

[9] UCI Machine Learning Repository, "Taiwanese Bankruptcy Prediction." [Online]. Available: https://archive.ics.uci.edu/ml/*dataset*s/Taiwanese+Bankruptcy+Prediction

[10] U. Singh, "US Company Bankruptcy Prediction *Dataset*." [Online]. Available: https://www.kaggle.com/*dataset*s/utkarshx27/american-companies-bankruptcy-prediction-*dataset*

[11] A. Andi, "Komparasi Kinerja Algoritma Random Forest dan Support Vector Machine berbasis Adaptive Boosting untuk Analisis Kelayakan Kredit Nasabah," *Jurnal TIMES*, vol. 13, no. 2, pp. 326–333, 2024.

[12] X. W. Liang, A. P. Jiang, T. Li, Y. Y. Xue, and G. T. Wang, "LR-SMOTE—An improved unbalanced data set oversampling based on K-means and SVM," *Knowl Based Syst*, vol. 196, p. 105845, 2020.

[13] R. Saputra, A. G. Alamsyah, M. Tjoanda, K. Nick, and A. Cornelius, "Analisis Prediksi Saham Tesla menggunakan Algoritma Long Short Term Memory (LSTM)," *Journal of Computer Science and Information Technology*, vol. 2, no. 1, pp. 81–90, 2024.

[14] F. A. Mohammad, A. M. Rizki, and A. N. Sihananto, "Peramalan Tingkat Inflasi di Indonesia Menggunakan Artificial Bee Colony dan *XGBoost*," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, 2024.

[15] M. FADLI and R. A. Saputra, "Klasifikasi dan evaluasi performa model Random Forest untuk prediksi stroke," *Jurnal Teknik*, vol. 12, no. 2, 2023.

[16] S. Prusty, S. Patnaik, and S. K. Dash, "SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer," *Frontiers in Nanotechnology*, vol. 4, p. 972421, 2022.

[17] A. A. Dhani, "Perbaikan Akurasi Random Forest dengan Anova dan Smote pada Klasifikasi Data Stunting," 2024.

[18] M. Imani, A. Beikmohammadi, and H. R. Arabnia, "Comprehensive Analysis of Random Forest and *XGBoost* Performance with SMOTE, ADASYN, and GNUS Under Varying Imbalance Levels," *Technologies (Basel)*, vol. 13, no. 3, p. 88, Feb. 2025, doi: 10.3390/technologies13030088.

[19] L. Barreñada, P. Dhiman, D. Timmerman, A.-L. Boulesteix, and B. Van Calster, "Understanding *overfitting* in random forest for probability estimation: a visualization and simulation study," *Diagn Progn Res*, vol. 8, no. 1, p. 14, 2024.

[20] M. A. Muslim and Y. Dasril, "Company bankruptcy prediction framework based on the most influential features using *XGBoost* and stacking ensemble learning," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, pp. 5549–5557, 2021.