JURNAL FASILKOM
Volume 16 No. 2 | Agustus 2026: 331-343

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Perbandingan Model *Deep Learning* LSTM, GRU, dan Bi-LSTM untuk Prediksi Hujan Harian Australia Menggunakan Teknik *SMOTE*

Ari Risnanto[1], Arief Hermawan[2], Donny Avianto[3] , Bayu T Nugroho[4]
[1234]Program Studi Teknologi Informasi, Magister Teknologi Informasi, Universitas Teknologi Yogyakarta
[1]arieriz@gmail.com*, [2]ariefdb@uty.ac.id, [3]donny@uty.ac.id, [4]bayutri2002@gmail.com

Abstract

Climate change causing increasingly erratic rainfall patterns, triggering an increase in hydrometeorological disasters such as floods, droughts, and declining agricultural productivity. Therefore, accurate rainfall prediction is crucial for mitigation and decision-making. However, previous research often focuses solely on accuracy metrics without evaluating the model's computational burden, and often ignores the problem of class imbalance in weather datasets. This study evaluates the performance and computational efficiency of LSTM, GRU, and Bi-LSTM deep learning models for daily rainfall prediction using the historical Australian meteorological dataset weatherAus. The novelty of this study lies in the comprehensive mapping between predictive quality and resource efficiency after dataset balancing. The preprocessing stage includes handling missing values, categorical data transformation, data leakage prevention, data sharing, and the application of SMOTE oversampling. The results of the area under the curve (AUC-ROC) evaluation show that the GRU model is superior with a value of 0.85, surpassing LSTM and Bi-LSTM, respectively, at 0.84. In the rain class recall metric, GRU again leads (0.70), compared to LSTM (0.67), and Bi-LSTM (0.57). Computational evaluation, GRU is significantly more efficient with the fastest training time (1,306.26 seconds), followed by LSTM (2,259.12 seconds), and Bi-LSTM (13,348.43 seconds). Peak RAM usage relatively comparable, GRU (2,053.77 MB), LSTM (1,971.47 MB), and the highest Bi-LSTM (2,242.60 MB). These findings conclude that GRU is recommended as the most optimal model that balances accuracy and efficiency, LSTM as an alternative, while Bi-LSTM is considered less effective. Future research recommended to explore hybrid architectures or ensemble learning to capture more complex spatiotemporal patterns.

Keywords: rain prediction, SMOTE, LSTM, GRU, Bi-LSTM

Abstrak

Perubahan iklim menyebabkan pola curah hujan semakin tidak menentu, sehingga memicu peningkatan bencana hidrometeorologi seperti banjir, kekeringan, dan penurunan produktivitas pertanian. Oleh karena itu, prediksi hujan yang akurat sangat krusial untuk mitigasi dan pengambilan keputusan. Namun, terdapat kesenjangan pada penelitian terdahulu sering kali hanya berfokus pada metrik akurasi tanpa mengevaluasi beban komputasi model, serta kerap mengabaikan permasalahan ketidakseimbangan kelas pada dataset cuaca. Penelitian ini mengevaluasi kinerja dan efisiensi komputasi model *deep learning* LSTM, GRU, dan Bi-LSTM untuk prediksi hujan harian menggunakan dataset historis meteorologi australia *weatherAus*. Kebaruan penelitian ini terletak pada pemetaan komprehensif antara kualitas prediktif dan efisiensi sumber daya setelah dataset diseimbangkan. Tahap pra-pemrosesan mencakup penanganan nilai hilang, transformasi data kategorikal, pencegahan data *leakage*, pembagian data, dan penerapan *oversampling* SMOTE. Hasil evaluasi *area under curve* (AUC-ROC), model GRU unggul memperoleh nilai (0,85) melampaui LSTM, dan Bi-LSTM masing-masing (0,84). Pada metrik *recall* kelas hujan, GRU kembali memimpin (0,70), dibandingkan LSTM (0,67), dan Bi-LSTM (0,57). Evaluasi komputasi, GRU secara signifikan lebih efisien dengan waktu pelatihan tercepat (1.306,26 detik), disusul LSTM (2.259,12 detik), dan Bi-LSTM (13.348,43 detik). Penggunaan RAM puncak relatif sepadan, GRU (2.053,77 MB), LSTM (1.971,47 MB), dan tertinggi Bi-LSTM (2.242,60 MB). Temuan ini menyimpulkan bahwa GRU direkomendasikan sebagai model paling optimal yang menyeimbangkan akurasi dan efisiensi, LSTM sebagai alternatif, sedangkan Bi-LSTM dinilai kurang efektif. Penelitian selanjutnya disarankan mengeksplorasi arsitektur hibrida atau *ensemble learning* guna menangkap pola spasiotemporal yang lebih kompleks.

Kata kunci: prediksi hujan, SMOTE, LSTM, GRU, Bi-LSTM

1. Pendahuluan

Hujan sebagai unsur meteorologi memiliki implikasi luas di berbagai sektor kehidupan. Akurasi dalam memprediksi hujan sangat krusial untuk perencanaan pertanian, manajemen sumber daya air, mitigasi bencana alam seperti banjir dan kekeringan, serta dalam mendukung sektor transportasi dan pariwisata [1]. Peningkatan akurasi prediksi hujan berkontribusi langsung pada pengambilan keputusan yang lebih efektif serta berdampak mengurangi kerugian ekonomi dan sosial. Namun, prediksi hujan adalah proses yang sangat kompleks, terutama karena sifat dinamis dan non-liniear. Variabilitas spasial dan temporal yang tinggi, serta interaksi multifaktor dari berbagai parameter atmosfer seringkali membatasi kemampuan model prediksi konvensional untuk memberikan hasil yang konsisten dan akurat [2].

Beberapa penelitian sebelumnya dalam prediksi hujan telah banyak mengandalkan metode statistik klasik seperti regresi linear, ARIMA, atau model berbasis fisika[3]. Meskipun metode-metode tersebut memberikan dasar pemahaman yang baik, tetapi sering

P-ISSN : 2089-3353
E-ISSN : 2808-9162

kesulitan dalam menangkap pola non-linear dan dependensi jangka panjang pada data deret waktu (*timeseries*) meteorologi [4]. Seiring dengan perkembangan teknologi komputasi dan ketersediaan data yang masif, pendekatan *Machine Learning* (ML) telah muncul sebagai alternatif [5]. Model ML konvensional seperti *Support Vector Machine* dan *Random Forest* telah menunjukkan peningkatan kinerja dibandingkan metode statistik, namun masih mempunyai keterbatasan dalam memproses data sekuensial kompleks dan berdimensi tinggi secara efisien [6].

Dalam beberapa tahun terakhir, *Deep Learning* (DL) telah merevolusi banyak bidang, termasuk prediksi cuaca, berkat kemampuannya untuk mempelajari representasi fitur hierarkis secara otomatis dari data mentah [7]. Khususnya arsitektur *Recurrent Neural Network* (RNN) seperti model *Long Short-Term Memory* (LSTM) dan model *Gated Recurrent Unit* (GRU) telah terbukti sangat efektif dalam menangani data deret waktu. Kemampuan LSTM dan GRU untuk menyimpan dan mengelola informasi urutan waktu yang panjang melalui mekanisme *gating* yang canggih, memungkinkan untuk menangkap dependensi temporal yang penting dalam data cuaca [8].

Penelitian terdahulu menggunakan pendekatan ML dengan kombinasi empat algoritma *Random Forest* (RM), *K-Nearest Neighbour* (KNN), *Decision Tree* (DT), dan *Support Vector Machine* (SVM) dalam prediksi hujan, suhu, dan kelembaban. Menggunakan dataset bersumber dari API OpenWeatherMap berisi parameter meteorologi harian. Untuk meningkatkan prediksi dan akurasi dilakukan *hyperparameter tuning*, pemilihan fitur, dan penyeimbangan kelas. Hasil evaluasi eksperimental menunjukkan bahwa RF mencapai akurasi 81,9% dalam memprediksi hujan, sedangkan model regresi suhu, dan kelembaban memperoleh skor R^2 masing-masing sebesar 96,5% dan 71,8% [9].

Prediksi curah hujan juga dilakukan penelitian dengan studi kasus kota Tanjungpinang, Provinsi Kepulauan Riau. Prakiraan curah hujan sangat penting di daerah kepulauan karena beberapa alasan, diantaranya adalah pulau-pulau seringkali memiliki sumber daya air yang terbatas dan sangat bergantung pada curah hujan untuk pasokan air tawar. Prakiraan curah hujan yang akurat sangat penting untuk pengelolaan air yang tepat dan memastikan pasokan air yang cukup untuk konsumsi manusia dan pertanian. Penelitian ini menguji kinerja metode *Support Vector Machine* (SVM) dalam memprediksi hujan dengan tantangan utama pada kondisi geografis kepulauan yang unik dibandingkan dengan kondisi di daratan. *Dataset* tahun 2021 hingga 2023 bersumber dari Stasiun Meteorologi Raja Haji Fisabilillah Tanjung pinang, variabel terdiri dari suhu, kelembaban, kecepatan angin, dan curah hujan. Hasil yang diperoleh adalah nilai presisi sebesar 82% untuk curah hujan, dengan skor evaluasi kurva ROC sebesar 0,74 [10].

Beberapa penelitian terdahulu juga telah memanfaatkan pendekatan DL dalam prakiraan curah hujan. Penelitian di beberapa wilayah di negara India, mengeksplorasi prediksi curah hujan menggunakan model *Seasional Autoregressive Integrated Moving Average* (SARIMA) dan LSTM. SARIMA diterapkan untuk menangkap dinamika temporal pola curah hujan dan menunjukkan akurasi yang lebih tinggi, sementara LSTM diperkenalkan untuk meningkatkan prediksi curah hujan dengan menangkap dependensi temporal yang rumit dan meningkatkan fleksibilitas. Studi ini menggunakan data meteorologi deret waktu periode 2013 hingga 2022. Penelitian menyoroti kemampuan SARIMA dalam menangkap dinamika temporal dan mengakui kolaborasi dengan LSTM untuk meningkatkan ketepatan prediksi curah hujan dalam menghadapi perubahan iklim [11].

Penelitian lain menggunakan DL model GRU dalam prediksi intensitas curah hujan untuk empat kecamatan di kota Mataram, Nusa Tenggara Barat, yaitu kecamatan Ampenan, Cakranegara, Majeluk, dan Selaparang. Sumber data menggunakan data per jam dari MERRA Power NASA periode 2010 hingga 2021. Model GRU berhasil menangkap tren curah hujan secara musiman pada empat wilayah tersebut. Hasil evaluasi berdasarkan *Root Mean Squared Error* (RMSE) menunjukkan bahwa model GRU memberikan akurasi wajar dengan kesalahan yang lebih rendah di Ampenan dan Majeluk, sementara Cakranegara dan Selaparang dengan hasil nilai RMSE yang lebih tinggi. Hal ini menunjukkan tantangan dalam memprediksi kejadian curah hujan ekstrem. Dibandingkan model statistik konvensional, GRU menunjukkan kinerja unggul dalam memproses data meteorologi sekuensial. Temuan studi ini memiliki implikasi praktis prediksi curah hujan yang lebih akurat dapat mendukung strategi mitigasi banjir perkotaan, mengoptimalkan perencanaan pertanian, dan meningkatkan sistem peringatan dini untuk kesiapsiagaan bencana [12].

Penelitian lainnya mengembangkan model LSTM yang di optimasi dengan *Bayesian Optimization* (BO) untuk memprediksi curah hujan di tiga Kabupaten di Jawa Tengah yaitu Cilacap, Tegal, dan Semarang. Hasil evaluasi pada kombinasi model LSTM-BO menunjukkan peningkatan kinerja yang signifikan dibandingkan dengan model LSTM standar. Meskipun LSTM-BO unggul pada bulan dan lokasi tertentu, model LSTM standar cenderung lebih stabil dan konsisten dengan hasil mendekati observasi [13].

Meskipun beberapa penelitian terdahulu telah menunjukkan keunggulan LSTM dan GRU dalam memproses data metorologi sekuensial, mayoritas studi tersebut berfokus pada optimalisasi akurasi dan mengabaikan masalah fundamental dalam dataset cuaca, yaitu ketidakseimbangan kelas. Di sisi lain penelitian yang berupaya menyeimbangkan kelas menggunakan teknik seperti SMOTE umumnya hanya diaplikasikan pada algortima ML konvensional seperti

P-ISSN : 2089-3353
E-ISSN : 2808-9162

SVM atau RF yang memiliki keterbatasan dalam menangkap dependensi temporal jangka panjang. Oleh karena itu, terdapat kesenjangan (*gap*) penelitian yang signifikan hingga saat ini, belum ada studi komprehensif yang secara khusus membandingkan performa arsitektur LSTM, GRU, dan Bi-LSTM secara bersamaan pada prediksi hujan harian menggunakan dataset yang telah diseimbangkan dengan teknik SMOTE, sembari mengukur metrik efisiensi komputasinya.

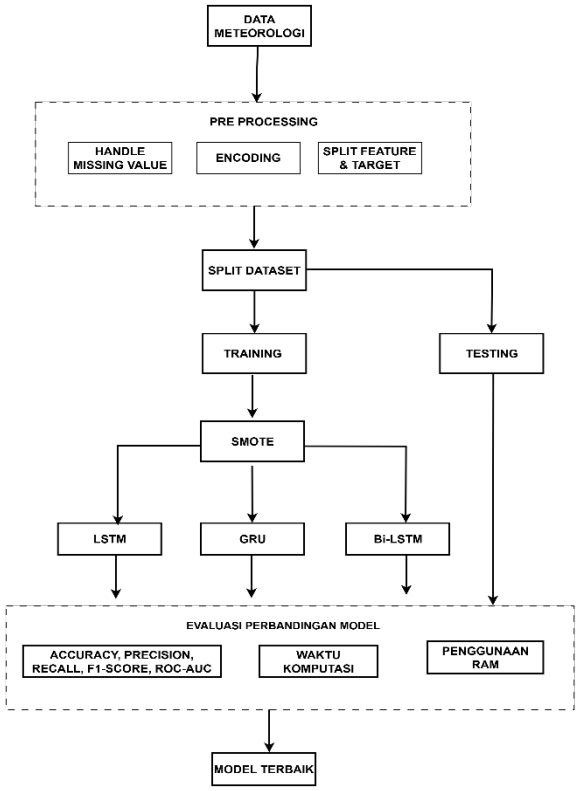
Berdasarkan kesenjangan tersebut, penelitian ini dirumuskan ke dalam dua pertanyaan penelitian (*Research Questions*/RQ) utama:
RQ1: Bagaimana pengaruh penerapan teknik SMOTE terhadap kinerja prediktif model LSTM, GRU, dan Bi-LSTM dalam mengatasi ketidakseimbangan kelas pada data curah hujan?
RQ2: Diantara ketiga arsitektur tersebut, model manakah yang paling optimal antara kinerja prediktif dan efisiensi komputasi?

Tujuan utama dari penelitian ini adalah untuk mengeksplorasi dan membandingkan kinerja dari tiga arsitektur DL tersebut dalam aspek akurasi, efisiensi waktu komputasi, serta penggunaan sumber daya *Random Acces Memory* (RAM) dalam prediksi hujan harian. Dalam konteks pemodelan, efisiensi waktu komputasi dan pemanfaatan sumber daya (RAM) adalah faktor penting yang secara signifikan mempengaruhi kinerja [14]. Kontribusi dari penelitian ini adalah memberikan panduan empiris dalam pengembangan sistem peringatan dini hidrometeorologi. Penelitian ini tidak hanya mengidentifikasi arsitektur dengan akurasi terbaik, tetapi juga memetakan model mana yang paling *robust* dan efisien untuk diimplementasikan pada lingkungan dengan sumber daya komputasi yang terbatas.

2. Metode Penelitian

Kerangka konseptual metodologi dalam penelitian ini dirancang secara sistematis dengan tahapan pengumpulan data historis meteorologi, pra-pemrosesan data mencakup penanganan nilai hilang, transformasi data kategorikal, pencegahan data *leakage,* serta pemisahan fitur dan target. Tahapan selanjutnya pembagian data pelatihan dan pengujian, penerapan *oversampling* SMOTE pada data pelatihan untuk mengatasi masalah ketidakseimbangan kelas, pembangunan arsitektur DL model LSTM, GRU, Bi-LSTM, dan terakhir evaluasi model.

Pada tahap akhir, setiap model yang telah dilatih akan dikomparasi dan dievaluasi menggunakan dua pendekatan utama. Pendekatan pertama adalah evaluasi kualitas prediktif yang diukur menggunakan metrik klasifikasi standar (akurasi, *F1-score,* presisi, *recall,* AUC-ROC) untuk menilai kemampuan model dalam membedakan antar kelas. Pendekatan kedua adalah evaluasi efisiensi komputasi, yang dirancang untuk mengukur beban operasioanl masing-masing arsitektur. Tahapan penelitian dapat dilihat pada gambar 1.



Gambar 1. Tahapan Penelitian

2.1 Pengumpulan Data

Dataset penelitian menggunakan data meteorologi *weatherAus.csv* yang bersumber dari *Kaggle*, merupakan data historis meteorologi atau cuaca skala harian Australia selama sepuluh tahun. Periode tahun 2008 hingga 2018 terdiri 142.192 baris dan 24 variabel. Meskipun dataset ini merekam pengamatan hingga tahun 2018, pemilihan dataset *weatherAus* sangat representatif dan ideal untuk tujuan komparasi arsitektur (*algorithm benchmarking*). Data historis selama sepuluh tahun telah mencakup berbagai variasi pola cuaca jangka panjang yang esensial untuk melatih model sekuensial. Fokus utama dari penelitian ini adalah pada evaluasi kinerja dan efisiensi komputasi dari arsitektur LSTM, GRU, dan Bi-LSTM dalam memodelkan data deret waktu yang kompleks, bukan pada prediksi operasional secara *real-time*. Selain itu, penggunaan *dataset* publik yang terstandarisasi memungkinkan studi komparatif yang lebih adil (*fair comparison*), menghilangkan bias subjektif dalam pengumpulan data, serta memungkinkan penelitian lain mereplikasi eksperimen dengan hasil yang valid dan kredibel. Fitur atau variabel meteorologi dan deskripsi ditunjukkan pada tabel 1.

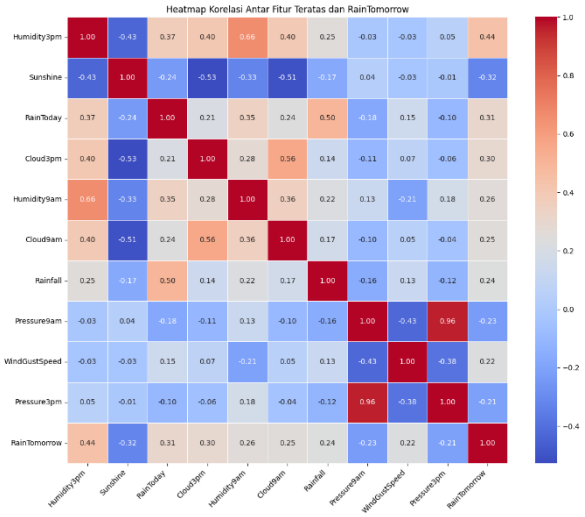
Tabel 1. Variabel dan Deskripsi Data Meteorologi

Variabel	Deskripsi (satuan)
Date	Tanggal observasi
Location	Nama lokasi oservasi
MinTemp	Suhu minimum (°C)

P-ISSN : 2089-3353
E-ISSN : 2808-9162

MaxTemp	Suhu maksimum (°C)
Rainfall	Curah hujan (mm)
Evaporation	Total penguapan (mm)
Sunshine	Durasi penyinaran matahari (jam)
WindGusDir	Arah angin (derajat)
WindGustSpeed	Kecepatan angin (km/h)
WindDir9am	Arah angin pada jam 9 pagi (derajat)
WindDir3pm	Arah angin pada jam 3 sore (derajat)
WindSpeed9am	Kecepatan angin (km/h) jam 9 pagi
WindSpeed3pm	Kecepatan angin (km/h) jam 3 sore
Humidity9am	Kelembaban udara (%) jam 9 pagi
Humidity3pm	Kelembaban udara (%) jam 3 sore
Pressure9am	Tekanan udara (hPa) jam 9 pagi
Pressure3pm	Tekanan udara (hPa) jam 3 sore
Cloud9am	Tutupan awan (oktas) jam 9 pagi
Cloud3pm	Tutupan awan (oktas) jam 3 sore
Temp9am	Temperatur udara (°C) jam 9 pagi
Temp3pm	Temperatur udara (°C) jam 3 pagi
RainToday	Indikator apakah hujan (ya/tidak)
RISK_MM	Jumlah curah hujan untuk hari berikutnya (mm)
RainTomorrow	Indikator apakah hujan keesokan hari (ya/tidak)

Gambar 2, *Heatmap* menjelaskan interpretasi korelasi antar fitur teratas, menunjukkan hubungan linear antara setiap pasang fitur paling berpengaruh dan fitur target *RainTomorrow*. Nilai korelasi berkisar dari -1 hingga 1. Nilai positif (mendekati 1) berarti memiliki hubungan kuat, jika satu fitur meningkat maka yang lain cenderung meningkat. Nilai negatif (mendekati -1) menunjukkan hubungan negatif kuat, jika satu fitur meningkat maka fitur lain cenderung menurun. Nilai mendekati 0 menunjukkan hubungan linear yang sangat lemah atau tidak ada [15].



Gambar 2. *Heatmap* Korelasi Antar Fitur Teratas

Dari analisis visualisasi *heatmap* menunjukkan fitur paling berpengaruh meliputi *Humidity3pm, Sunshine, RainToday, Cloud3pm, Humidity9am, cloud9am, Pressure9am,* dan, *Pressure3pm.* Korelasi dengan antar fitur *Humidity3pm* (kelembabaan tinggi sore hari) dan *Sunshine* (durasi pendek penyinaran matahari) adalah inidikator utama untuk memprediksi hujan pada besok hari, hal ini sejalan dengan pemahaman meteorologi umum. Dengan memahami korelasi antara fitur-fitur teratas penting dalam membangun model prediksi hujan yang akurat karena berpengaruh langsung terhadap variabel target prediksi [16].

Sebagai gambaran awal mengenai data penelitian, tabel 2 menyajikan sampel dataset meteorologi merepresentasikan struktur data dan variabel yang digunakan. Sampel dataset tersebut memperlihatkan struktur data serta variabel-variabel yang menjadi masukan dalam tahapan pemrosesan, analisis, dan pembangunan model prediksi hujan.

Tabel 2. Sampel *Dataset* Meteorologi

Date	*Temp3 pm*	*Max Temp*	*Pressure9am*	*Humidity9am*	*Sunshine*
20/6/2017	20.9	21.8	1030.6	87	8.5
21/6/2017	22.4	23.4	1029.4	90	8.5
22/6/2017	24.5	25.3	1029.4	94	8.5
23/6/2017	26.1	26.9	1022.3	92	8.5
24/6/2017	26.0	27	1018.8	72	8.5

2.2. Pra-pemrosesan Data

Pra-pemrosesan data dapat didefiniskan sebagai serangkaian langkah atau proses untuk mempersiapkan data mentah (*raw data*) sebelum digunakan dalam pemodelan. Langkah-langkah ini meliputi memuat *library* dan dataset, pembersihan data, penanganan data hilang (*missing values*), transformasi fitur, dan normalisasi atau standarisasi data [17]. Pada penelitian ini, tahapan pra-pemrosesan dilakukan penanganan nilai hilang, transformasi data ketegorikal dengan *label encoding*, transformasi data kategorikal dengan *label encoding*, pencegahan data *leakage,* pembagian data pelatihan dan pengujian, normalisasi menggunakan min-max, serta mengatasi permasalahan ketidakseimbangan kelas dengan teknik *oversampling* SMOTE.

2.3. SMOTE

Synthetic Minority Over Sampling Technique (SMOTE) merupakan teknik *oversampling* digunakan untuk menangani masalah ketidakseimbangan kelas (*class imbalance*) pada dataset. Pada kasus prediksi hujan, data umumnya tidak seimbang karena jumlah hari tidak hujan lebih banyak dibandingkan hari hujan. Kondisi ini membuat model lebih cenderung untuk memprediksi kelas mayoritas, sehingga akurasi terhadap kelas minoritas menjadi rendah. SMOTE bekerja dengan menghasilkan data sintetis baru pada kelas minoritas berdasarkan kedekatan antar sampel menggunakan metode *K-Nearest Neighbors* (KNN) [18]. Dengan demikian, distribusi data menjadi lebih seimbang sehingga model mampu mempelajari pola kelas minoritas dengan lebih baik.

$$x_{baru} = x_i + \lambda(x_{nn} - x_i) \quad (1)$$

Persamaan (1) adalah rumus SMOTE, dengan x_{baru} merupakan data sintetis baru, x_i data minoritas asli, x_{nn} tetangga terdekat (*nearest neighbor*), dan λ sebagai bilangan acak antara 0 dan 1, dengan ketentuan jika $\lambda = 0$ data sama dengan titik awal, jika $\lambda = 1$ maka data

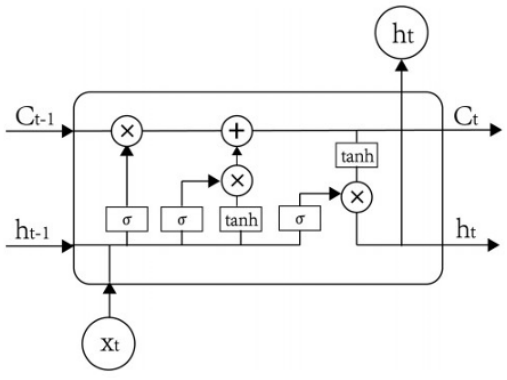
P-ISSN : 2089-3353
E-ISSN : 2808-9162

sama dengan tetangga, jika $0 < \lambda < 1$ maka data sintetis berada diantara keduannya [19].

2.4. Implementasi Model LSTM, GRU, dan Bi-LSTM

Deep Learning adalah cabang dari kecerdasan buatan (*Artificial Intelegence*) yang memproses dan menganalisa data dengan menggunakan struktur jaringan saraf tiruan terdiri dari banyak lapisan [20]. Lapisan neuron buatan yang terhubung bekerja secara hierarki untuk proses belajar, yang mencakup mengenali fitur sederhana hingga kompleks. Tujuan utama adalah untuk meniru cara otak manusia bekerja mengenali pola, memahami bahasa, dan membuat keputusan [21].

Long Short Term Memory (LSTM) merupakan arsitektur jaringan saraf tiruan yang digunakan pada data sekuensial atau data yang berurutan seperti data *timeseries,* susunan teks atau kalimat, video maupun audio [22]. LSTM ditemukan pada tahun 1997 dan merupakan pengembangan dari *Recurent Neural Network* (RNN) yang telah ditemukan lebih dahulu pada tahun 1986. Keterbatasan performa RNN dalam mengatasi data jangka waktu yang panjang, menjadi dasar dikembangkannya LSTM. Hingga LSTM ditemukan dan mempunyai kinerja lebih baik dalam menangani data jangka waktu yang panjang dengan kemampuan memori sel yang lebih baik dari RNN. LSTM bekerja dengan mekanisme memori dan gerbang (*gate*) untuk mengatur aliran informasi pada data sekuensial. *Forget Gate* berfungsi menetukan informasi dari *state* sebelumnya apakah perlu disimpan atau dilupakan, *Input Gate* menambahkan informasi baru ke memori, dan *Output Gate* menentukan informasi yang akan menjadi output [23]. Arsitektur model LSTM dapat dilihat pada gambar 3.

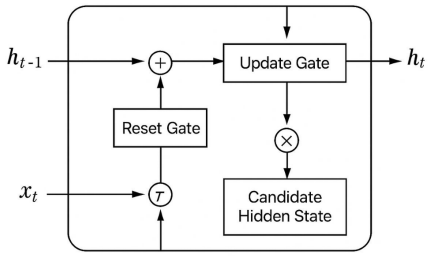


Gambar 3. Arsitektur Model LSTM

Persamaan (2) adalah rumus output LSTM, dimana h_t sebagai *hidden state* atau output tersembunyi, o_t nilai *output gate*, *tanh* sebagai fungsi aktivasi sering digunakan pada *Artificial Neural Network* (ANN), dan C_t sebagai *cell state* jalur memori utama yang membawa informasi penting dari waktu sebelumnya.

$$h_t = o_t \,.\, tanh(C_t) \quad (2)$$

Gated Recurren Unit (GRU) adalah jenis arsitektur jaringan varian dari RNN yang diperkenalkan pada tahun 2014, dirancang untuk secara efektif memproses data sekuensial sambil mempertahankan parameter yang lebih sedikit dibandingkan dengan LSTM.



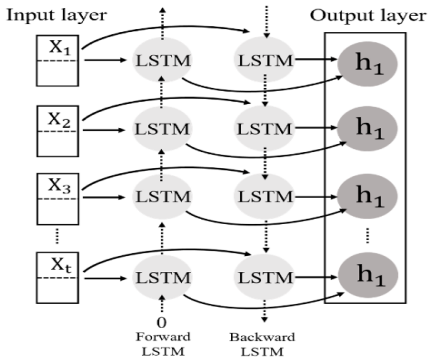
Gambar 4. Arsitektur Model GRU

GRU menggunakan dua gerbang untuk mengelola informasi sehingga memungkinkan mempertahankan data penting dalam urutan yang panjang tanpa masalah kehilangan gradien yang sering mempengaruhi RNN. Terlihat pada gambar 4, arsitektur GRU terdiri dari dua gerbang yaitu *Reset Gate* dan *Update Gate*. *Reset Gate* menentukan banyaknya informasi masa lalu yang harus dilupakan, sedangkan *Update Gate* mengontrol berapa banyak informasi baru untuk dimasukkan ke dalam keadaan saat ini. Penyederhanaan arsitektur GRU meningkatkan efisiensi dan kinerja komputasi dalam berbagai aplikasi, seperti klasifikasi teks dan pengenalan ucapan [24].

$$h_t = (1 - z_t)\, x\, h_{t-1} + z_t\, x\, h_t \quad (3)$$

Persamaan (3) h_t adalah *hidden state* atau *ouptut*, z_t sebagai *update gate*, h_{t-1} merujuk pada *hidden state* dari langkah waktu sebelumnya.

Bidirectional Long Short Term Memory (Bi-LSTM) adalah perluasan dari arsitektur jaringan saraf tiruan LSTM yang dapat memproses data berurutan kedua arah sekaligus, lapisan maju (*forward*) memproses data dari langkah pertama hingga langkah terakhir, sedangkan lapisan mundur (*backward*) memproses data secara terbalik dari langkah terakhir hingga langkah pertama [25].



Gambar 5. Arsitektur Model Bi-LSTM

Arsitektur model Bi-LSTM dapat dilihat pada gambar 5. Persamaan (4) adalah langkah maju (*forward* LSTM)

P-ISSN : 2089-3353
E-ISSN : 2808-9162

memproses data dari urutan $x_1 \rightarrow x_2 \rightarrow x_3 \ldots x_n$, dengan $\underset{ht}{\rightarrow}$ *hidden state* arah maju, x_t input pada waktu ke-*t*

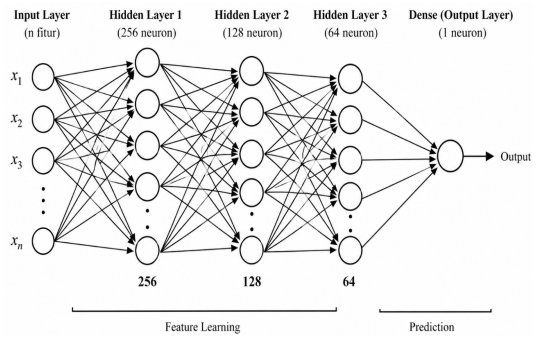
$$\underset{ht}{\rightarrow} = LSTM(x_t, \underset{ht-1}{\longrightarrow}) \quad (4)$$

Persamaan (5) adalah langkah mundur (*backward* LSTM) memproses data dari urutan $x_n \leftarrow x_{n-1}$, $\leftarrow x_{n-2}$, $\leftarrow x_{n-3}$ dengan $\underset{ht}{\leftarrow}$ *hidden state* arah mundur dan x_t input pada waktu ke-*t*

$$\underset{ht}{\leftarrow} = LSTM(x_t, \underset{ht+1}{\longleftarrow}) \quad (5)$$

2.5. *Hidden Layer* pada Model

Struktur model DL yang digunakan (LSTM, GRU, dan Bi-LSTM) dalam penelitian ini terdapat tiga *hidden layer* sebelum lapisan *dense* atau output. Setiap model mengikuti struktur serupa dengan tiga lapisan *rekuren* yang berfungsi sebagai *hidden layer* dengan neuron 256, 128, dan 64 untuk mengektraksi fitur dari data sekuensial sebelum diteruskan ke lapisan *dense* terakhir yang bertindak sebagai lapisan output. Struktur *hidden layer* pada jaringan saraf tiruan dapat dilihat pada gambar 6.



Gambar 6. Struktur *Hidden Layer*

Berikut contoh *listing* program pada model LSTM yang menunjukkan definisi *hidden layer*.

Hidden Layer LSTM

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Input,
LSTM, Dense

LOOKBACK_WINDOW = 14
model_lstm = Sequential([
  Input(shape=(LOOKBACK_WINDOW,X_train.shape[
  2])),
  LSTM(256,return_sequences=True,activation='
  tanh')
  LSTM(128,return_sequences=True,activation='
  tanh')
  LSTM(64, activation='tanh')
  Dense(1, activation='sigmoid')
])
Model_lstm.summary()
```

Pada listing program LSTM diatas, terdiri dari tiga *hidden layer* dengan satu *dense*. Jumlah lapisan *neuron* 256, 128, dan 64 memungkinkan model untuk mengektraksi fitur secara hierarki. Lapisan awal mempelajari representasi fitur tingkat rendah atau pola dasar dari data cuaca, sementara lapisan yang lebih dalam mengidentifikasi pola yang lebih kompleks atau abstraksi tingkat tinggi dengan menggabungkan informasi dari berbagai variabel cuaca dan periode waktu, kemampuan ini sangat penting untuk memahami dinamika cuaca kompleks yang mendasari pembentukan hujan [26]. Untuk mengoptimalkan kemampuan arsitektur LSTM, GRU, dan Bi-LSTM dalam mengekploitasi dependensi temporal, pemodelan menerapkan metode *sliding window* dengan *lookback period* selama 14 hari. Dengan demikian, representasi data input matrik berdimensi tiga (*batch_size,* 14, 24). Fungsi LOOKBACK WINDOW untuk menetapkan jumlah waktu (*time step*) ke belakang yang digunakan model memprediksi nilai pada langkah waktu berikutnya. Pada output digunakan fungsi aktivasi *sigmoid* karena dalam penelitian menggunakan klasifikasi biner, sehingga output model berupa probabilitas 0 dan 1.

2.6. Evaluasi Model

Tahap evaluasi dilakukan untuk menilai kinerja model, proses evaluasi menggunakan *Confusion Matrix*. Matriks yang digunakan untuk menggambarkan kinerja model klasifikasi berdasarkan perbandingan antara hasil prediksi dan aktual. Melalui metode ini diperoleh beberapa metrik evaluasi seperti akurasi dengan persamaan (6), presisi dengan persamaan (7), *recall* persamaan (8), dan F1-score dengan persamaan (9).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (6)$$

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$F1 = \frac{2\ x\ (Recall\ x\ Precision)}{Recall\ x\ Precision} \quad (9)$$

Confusion Matrix sebagai indikator kinerja model memberikan wawasan tentang *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Analisis evaluasi ini penting dalam konteks prediksi hujan untuk memahami bagaimana setiap model menangani kedua kelas tidak hujan (*Class 0*) dan kelas hujan (*Class 1*).

Penelitian ini juga melakukan evaluasi performa model dari segi sumber daya komputasi yaitu waktu komputasi pelatihan, dan penggunaan RAM puncak pada masing-masing model. Perangkat komputer yang digunakan untuk eksplorasi dalam penelitian ini dengan spesifikasi prosesor 13th Gen Intel(R) Core (TM) i5-1340P berkecepatan 1,90 GHZ, memori RAM sebesar 16 GB, serta sistem operasi Windows 11 Home Single Language. Perangkat lunak memanfaatkan platform komputasi awan Google Colab untuk mendukung proses pengembangan dan eksekusi kode berbaris notebook secara efisien dan fleksibel, terutama dalam menjalankan eksperimen DL dan pengolahan data berskala besar.

P-ISSN : 2089-3353
E-ISSN : 2808-9162

3. Hasil dan Pembahasan

3.1. Pra-pemrosesan Data

Tahapan pra-pemrosesan data merupakan proses penting untuk memastikan kualitas, konsistensi, dan relevansi dari data mentah sebelum digunakan dalam pemodelan. Berdasarkan pemeriksaan pada variabel atau kolom pada dataset terdapat nilai hilang (*missing value*). Penanganan nilai hilang pada semua kolom numerik, diimputasi menggunakan median untuk meminimalkan distorsi akibat *outlier* yang sering terjadi pada data cuaca. Tabel 3 menunjukkan jumlah nilai hilang dan presentase pada tiap variabel sebelum dan sesudah dilakukan penanganan nilai hilang.

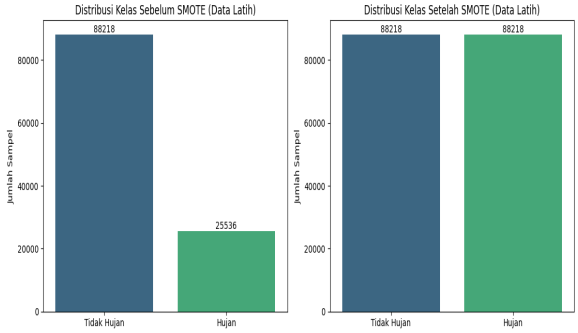
Tabel 3. Penanganan Nilai Hilang

Variabel	Nilai Hilang (sebe-lum)	Presen-tase (%)	Nilai Hilang (sesu-dah)	Presen-tase (%)
MinTemp	637	0,45	0	0
MaxTemp	322	0,23	0	0
Rainfall	1.406	0,90	0	0
Evaporation	60.843	42,79	0	0
Sunshine	67.816	47,69	0	0
WindGustSpeed	9.270	6,25	0	0
WindSpeed9am	1.348	0,95	0	0
WindSpeed3pm	2.630	1,85	0	0
Humidity9am	1.774	1,25	0	0
Humidity3pm	3.610	2,54	0	0
Pressure9am	14.014	9,86	0	0
Pressure3pm	13.981	9,83	0	0
Cloud9am	53.657	37,74	0	0
Cloud3pm	57.094	40,15	0	0
Temp9am	904	0,64	0	0
Temp3pm	2.726	1,92	0	0
RainToday	1.406	0,99	0	0

Langkah *encoding* di terapkan pada fitur kategorikal non-biner (*Location, WindGustDir, WindDir9am*, dan *WindDir3pm*) diubah menjadi representasi numerik. Langkah krusial selanjutnya adalah pencegahan data *leakage* dengan menghilangkan fitur *RISK_MM*. Secara definisi *RISK_MM* merupakan jumlah curah hujan aktual yang tercatat pada hari berikutnya. Karena fitur target *RainTomorrow* diturunkan langsung dari nilai ini (jika *RISK_MM* > 1 maka *RainTomorrow* = Ya). Menurut pedoman pemodelan berbasis ML, menyertakan atribut yang diukur setelah event terjadi di masa depan sebagai prediktor akan menyebabkan target *leakage,* yaitu kondisi dimana model dilatih menggunakan data yang mengandung informasi tentang target prediksi yang seharusnya tidak boleh diketahui pada saat pelatihan [27]. Target *leakage* jika tidak dihapus, model akan mendapatkan akurasi artifisial yang sangat tinggi selama fase pelatihan, namun gagal generalisasi pada saat diimplementasikan pada kondisi nyata [28].

Pemisahan fitur dan target juga dilakukan, fitur *RainTomorrow* ditetapkan sebagai target atau fitur dependen sedangkan fitur lainnya sebagai fitur independen. Data fitur independen selanjutnya dibagi menjadi data pelatihan (*training set*) dan pengujian (*testing set*) dengan rasio 80:20. Normalisasi dengan *MinMaxScaler* untuk menskalakan semua fitur ke rentang 0-1, normalisasi memastikan semua fitur berkontribusi secara proporsional. Untuk mengatasi ketidakseimbangan kelas yang signifikan pada target diterapakan teknik SMOTE hanya pada data pelatihan dan berhasil menyeimbangkan jumlah sampel antar kelas. Data pengujian dibiarkan dalam kondisi aslinya untuk merepresentasikan kondisi nyata sebenarnya saat dilakukan evaluasi model. Selanjutnya data di *reshape* menjadi format input yang sesuai dan memungkinkan model LSTM, GRU, Bi-LSTM dalam memproses data sebagai deret waktu.

3.2. Hasil Implementasi Teknik SMOTE



Gambar 7. Implementasi Teknik SMOTE

Pada gambar 7, menunjukkan hasil penerapan SMOTE pada data pelatihan untuk menangani ketidakseimbangan kelas (*class imbalance*). Grafik sebelah kiri menggambarkan kondisi distribusi kelas sebelum penerapan SMOTE, dimana terjadi ketidakseimbangan yang signifikan antara kelas tidak hujan (*Class 0*) dan kelas hujan (*Class 1*). Kelas tidak hujan memiliki jumlah sampel yang jauh lebih besar yaitu sebanyak 88.218 dibandingkan dengan kelas hujan yang hanya berjumlah 25.536 sampel. Ketidakseimbangan ini berpotensi menyebabkan model klasifikasi cenderung bias terhadap kelas mayoritas, sehingga menurunkan kemampuan model dalam mendeteksi kelas minoritas secara akurat. Sementara itu, grafik sebelah kanan menunjukkan distribusi kelas setelah penerapan SMOTE, dimana jumlah sampel pada kedua kelas telah berhasil di seimbangkan. Proses *oversampling* menghasilkan jumlah sampel yang sama untuk kelas tidak hujan dan hujan, masing-masing sebanyak 88.218 data. Dengan demikian, permasalahan ketidakseimbangan kelas dapat diatasi, sehingga model memiliki kesempatan untuk mempelajari pola dari kedua kelas secara lebih proporsional. Hal ini diharapkan dapat meningkatkan performa klasifikasi serta kemampuan generalisasi model dalam memprediksi kejadian hujan.

3.3. Struktur Model

Konfigurasi pelatihan dibuat seragam dengan tujuan agar proses evaluasi dan perbandingan performa model dapat dilakukan lebih objektif dan seimbang. Ketiga

P-ISSN : 2089-3353
E-ISSN : 2808-9162

model dilatih menggunakan *batch size* 32, *epoch* 100, *optimizer* Adam, dan *callback* berupa *early stopping* untuk mencegah *overfitting*.

Tabel 4. Kontruksi LSTM

Layer	*Shape Output*	Parameter
LSTM	(none,14, 256)	287.744
LSTM_ 1	(none,14, 128)	197.120
LSTM_ 2	(none, 64)	49.408
Dense	(none,1)	65
Jumlah		534.337

Tabel 5. Kontruksi GRU

Layer	*Shape Output*	Parameter
GRU	(none,14, 256)	216.576
GRU_ 1	(none,14, 128)	148.224
GRU_ 2	(none, 64)	37.248
Dense	(none,1)	65
Jumlah		402.113

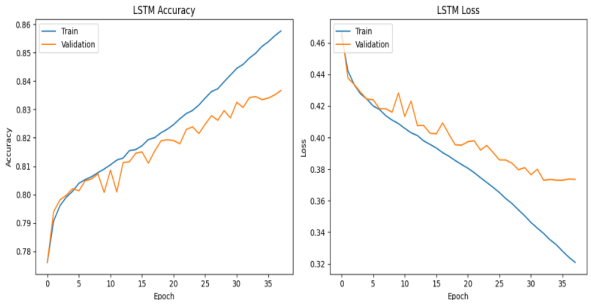
Tabel 6. Kontruksi Bi- LSTM

Layer	*Shape Output*	Parameter
Bi-LSTM	(none,14, 512)	575.488
Bi-LSTM_ 1	(none,14, 256)	656.384
Bi-LSTM_ 2	(none, 128)	164.352
Dense_2	(none,1)	129
Jumlah		1.396.353

Analisis struktur model meliputi *hidden layer, shape output,* serta jumlah parameter. Perbandingan pada tabel 4, tabel 5, dan tabel 6 dapat dijelaskan berdasarkan jumlah parameter, model Bi-LSTM memiliki jumlah parameter terbanyak (1.396.353), diikuti LSTM (534.337), dan GRU (402.113). Jumlah parameter yang lebih tinggi seringkali mengindikasikan model lebih kompleks, berpotensi lebih kuat dalam menangkap pola, tetapi juga rentan terhadap overfitting dan membutuhkan waktu pelatihan lebih lama. Berdasarkan *shape output* pada model LSTM, dan GRU adalah (none, 64), sedangkan Bi-LSTM adalah (none, 128). Hal ini menunjukkan bahwa model Bi-LSTM menghasilkan representasi fitur yang lebih kaya dua kali lipat dari lapisan terakhirnya, karena menggabungkan konteks pemrosesan dari dua arah (*forward* dan *backward*). Berdasarkan perhitungan jumlah parameter, perbedaan utama dalam perhitungan parameter anatara LSTM, GRU, dan Bi-LSTM terletak pada jumlah *gate* dan arah pemrosesan. LSTM menggunakan tiga *gate*, GRU menggunakan dua *gate*, dan Bi-LSTM menggandakan jumlah parameter karena melatih dua model untuk arah maju dan arah mundur.

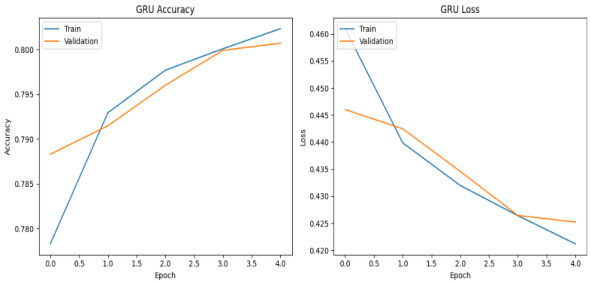
3.4 Analisis Perbandingan Grafik Akurasi dan *Loss*

Grafik memberikan informasi dinamika pembelajaran dari ketiga model DL selama proses pelatihan, yang divisualisasikan dengan kurva akurasi dan kurva *loss* untuk data pelatihan (*train*) dan validasi (*validation*) terhadap jumlah *epoch.* Grafik akurasi dan *loss* model LSTM dapat dilihat pada gambar 8. Grafik sebelah kiri adalah kurva akurasi pelatihan, (garis biru) menunjukkan peningkatan yang stabil dan konsisten dari *epoch* awal hingga mencapai puncaknya.



Gambar 8. Grafik Akurasi dan Loss Model LSTM

Akurasi validasi (garis orange) meningkat secara paralel, menjelaskan kemampuan generalisasi model LSTM yang baik. Kesenjangan antara akurasi pelatihan dan validasi cenderung kecil dan stabil di akhir pelatihan, mengindikasikan bahwa model tidak mengalami *overfitting* yang signifikan. Grafik sebelah kanan adalah kurva *loss* pelatihan, menunjukkan penurunan di awal dan kemudian melandai mendekati nilai minimum. Kurva *loss* validasi juga mengikuti pola penurunan yang serupa dekat dengan *loss* pelatihan. Konvergensi terjadi pada *epoch* yang relatif awal karena penerapan *early stop* yaitu untuk mencegah model untuk terus belajar pada pola spesifik data pelatihan yang mungkin tidak bergeneralisasi dengan baik. Dari sisi stabilitas model LSTM menunjukkan kurva yang relatif halus dan stabil, mengindikasikan proses pembelajaran yang konsisten.

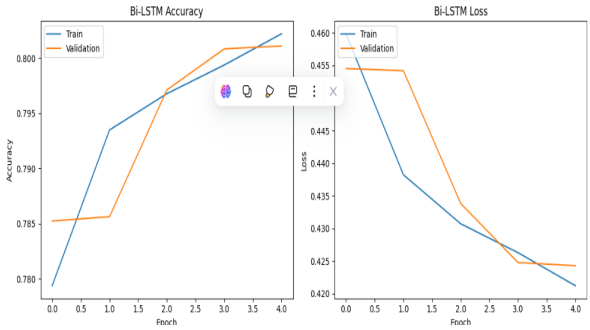


Gambar 9. Grafik Akurasi dan Loss Model GRU

Visualisasi grafik akurasi dan *loss* pada model GRU dapat dilihat pada gambar 9. Grafik sebelah kiri adalah kurva akurasi pelatihan (garis biru) menunjukkan peningkatan yang cepat dan mencapai level yang baik. Akurasi validasi (garis orange) juga meningkat, tetapi sedikit fluktuatif dimana proses pembelajaran stagnan (*plateu*) lebih awal dibandingkan LSTM. Kesenjangan antara akurasi pelatihan dan validasi sedikit terlihat, namun dengan *early stoping* dapat membantu mencegah *overfitting* yang lebih parah. Grafik sebelah kanan adalah kurva *loss* pelatihan dan validasi menunjukkan pola penurunan yang mirip dengan model LSTM. Tetapi *loss* validasi sedikit lebih tinggi atau kurang stabil yang mengindikasikan bahwa model GRU kurang *robust* dalam generalisasi atau memerlukan *tuning* lebih lanjut. Dari sisi stabilitas, meskipun GRU dikenal lebih sederhana, kurva metriknya menunjukkan sedikit fluktuasi yang dapat

P-ISSN : 2089-3353
E-ISSN : 2808-9162

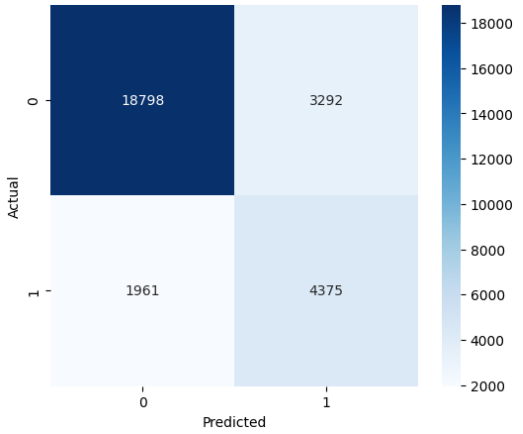
disebabkan oleh kurangnya kemampuan untuk menyimpan memori jangka panjang.



Gambar 10. Grafik Akurasi dan Loss Model Bi-LSTM

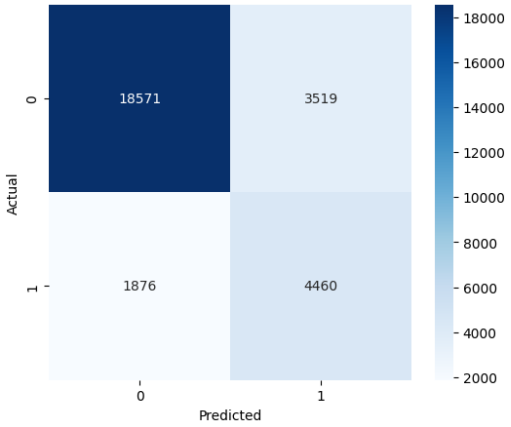
Pada gambar 10, menunjukkan visualisasi grafik akurasi dan *loss* model Bi-LSTM. Grafik sebelah kiri (garis biru) adalah kurva akurasi pelatihan meningkat dengan cepat, secara paralel akurasi validasi (garis orange) juga menunjukkan peningkatan. Model Bi-LSTM menangkap konteks dari dua arah, yang dapat menghasilkan kinerja validasi kuat. Dalam kasus ini, akurasi validasi menunjukkan peningkatan yang solid, dan mencapai level yang sebanding atau sedikit dibawah model LSTM. Grafik sebelah kanan adalah kurva *loss* pelatihan dan validasi keduanya menunjukkan penurunan. *Loss* validasi tetap terkendali dan tidak menunjukkan *overfitting* yang signifikan. Dari sisi stabilitas, Bi-LSTM cenderung menunjukkan stabilitas yang baik, menggabungkan kemampuan LSTM untuk memori jangka panjang dengan manfaat pemrosesan data pada dua arah.

3.5. Analisis *Confusion Matrix*



Gambar 11. *Confusion Matrix* Model LSTM

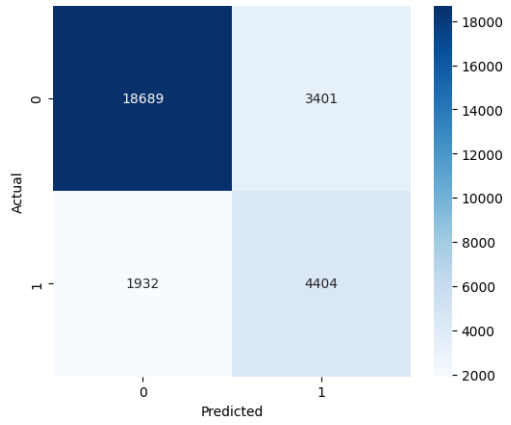
Analisis *Confusion Matrix* model LSTM pada gambar 11. TN berjumlah 18.798 berarti kasus tidak hujan (*class* 0) diprediksi dengan benar tidak hujan. TP berjumlah 4.375 berarti kasus hujan (*class* 1) berhasil diprediksi dengan benar sebagai hujan. FP berjumlah 3.292 hal ini berarti 3.292 kasus tidak hujan yang salah diprediksi sebagai hujan. FN berjumlah 1.961 hal ini berarti kasus 1.961 hujan yang salah diprediksi sebagai tidak hujan. Keseimbangan antara TP dan FN, bersama dengan FP, perlu dipertimbangkan dalam konteks aplikasi peringatan dini, di mana FN memiliki konsekuensi yang lebih serius karena kasus hujan terlewatkan tanpa peringatan. Model LSTM mencapai akurasi keseluruhan sebesar 0,81 dengan F1-Score untuk kelas hujan sebesar 0,62 menunjukkan peningkatan *recall* kelas minoritas.



Gambar 12. *Confusion Matrix* Model GRU

Analisis *Confusion Matrix* model GRU pada gambar 12, menunjukkan jumlah FP sedikit lebih tinggi dibandingkan model LSTM, mengindikasikan kecenderungan yang sedikit lebih besar terhadapa FP. Model GRU mengidentifikasi TP 4.460 kasus hujan dengan benar, menunjukkan kinerja yang kompetitif dalam deteksi positip. Sementara itu FN 1.876 kasus hujan gagal di deteksi sebagai hujan. Jumlah FN ini sedikit lebih rendah dari LSTM menunjukkan GRU memiliki sensivitas yang sedikit lebih baik terhadap kelas hujan.

Model GRU mencapai akurasi keseluruhan 0,81 dengan *F1-Score* untuk kelas hujan (*class* 1) sebesar 0,62 yang serupa dengan LSTM.



Gambar 13. *Confusion Matrix* Model Bi-LSTM

Analisis *Confusion Matrix* model Bi-LSTM pada gambar 13, model ini mencatat FP 3.401 berada di antara LSTM dan GRU dan FN 1.932 sedikit lebih tinggi dari GRU yang berarti lebih sering melewatkan kejadian hujan yang sebenarnya. Performa model Bi-LSTM sedikit kurang konsisten dibandingkan model LSTM, terutama dalam mendeteksi kejadian hujan.

P-ISSN : 2089-3353
E-ISSN : 2808-9162

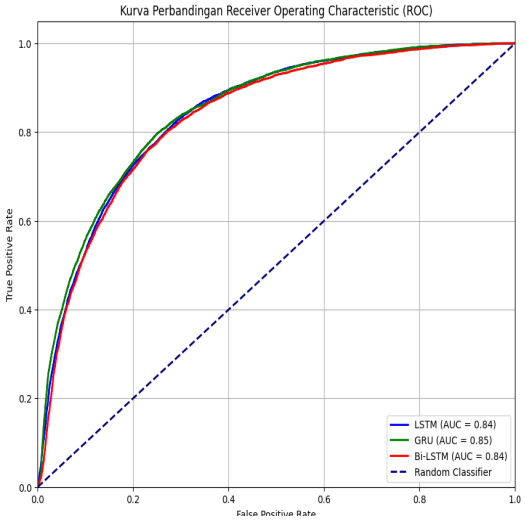
Akurasi keseluruhan model Bi-LSTM adalah 0,82 dengan *F1-Score* untuk kelas hujan (*class* 1) sebesar 0,59. Performa yang sangat dekat antara GRU dan Bi-LSTM dalam metrik *confusion matrix* dan F1-Score ini menunjukkan bahwa arah bidireksional pada LSTM tidak memberikan keuntungan yang signifikan dibandingkan GRU.

3.6. Evaluasi Model

Tabel 7. Perbandingan Evaluasi Model

Matrik Evaluasi		LSTM	GRU	Bi-LSTM
Overall Accuracy		0.81	0.81	0.82
Precision	Class 0	0.90	0.91	0.91
	Calss 1	0.57	0.55	0.60
Recall	Class 0	0.85	0.84	0.85
	Calss 1	0.67	0.70	0.57
F1-Score	Class 0	0.88	0.87	0.88
	Class 1	0.62	0.62	0.59
AUC-ROC		0.84	0.85	0.84

Berdasarkan metrik evaluasi pada tabel 7, evaluasi komprehensif terhadap ketiga model menunjukkan kinerja yang kompetitif. Matrik evaluasi *overall accuracy* menunjukkan Bi-LSTM lebih unggul 0,82 dikuti LSTM 0,81 dan GRU 0,81. Namun, perincian metrik pada kelas hujan (*class* 1) mengungkap perbedaan signifikan. Analisis *recall* model GRU tertinggi 0,70 diikuti LSTM 0,67 dan Bi-LSTM memperoleh 0,57. Model dengan *recall* tertinggi pada kelas hujan (*class* 1) mengindikasikan kemampuan yang lebih baik dalam mengidentifikasi kejadian hujan sebenarnya, sehingga meminimalkan FN. Sebaliknya, analisis *precission* untuk kelas hujan Bi-LSTM unggul mencapi 0,60 diikuti LSTM 0,57 dan GRU 0,55. Model dengan *precission* tertinggi menunjukkan kecenderungan lebih rendah terhadap FP.



Gambar 14. Kurva ROC

Analisis *Area Under the Receiver Operating Characteristic Curve* (AUC-ROC) pada gambar 14 memberikan visualisasi komprehensif menunjukkan kemampuan membedakan antara kelas positif dan negatif. Pada ketiga model yang dievaluasi, GRU memperoleh nilai lebih tinggi 0,85 dibandingkan LSTM dan Bi-LSTM masing-masing 0,84. Nilai yang hampir setara ini mengindikasikan bahwa model memiliki kapabilitas yang baik dalam membedakan kelas positif (hujan) dan kelas negatif (tidak hujan). Hasil pencapaian ketiga model ini secara signifikan berhasil melampaui performa SVM pada penelitian terdahulu [10]. Model memberikan peringkat lebih tinggi pada sampel positif acak dibandingkan dengan dengan sampel negatif acak. Secara visual kurva ROC untuk ketiga model ini yang memplot *True Positive Rate* (TPR) terhadap *False Positive Rate* (FPR) menunjukkan jalur yang berdekatan dan cenderung bergerak kearah sudut kiri atas grafik. Nilai AUC-ROC menunjukkan kemampuan membedakan kelas dengan memberikan kinerja sepadan.

Analisis hasil uji *McNemar Test* dilakukan untuk memvalidasi signifikansi statistik (*p-value*) perbedaan kinerja setara pada ketiga model secara berpasangan (*pairwise*). Uji ini mengevaluasi apakah ada perbedaan yang signifikan dalam tingkat kesalahan dua pengklasifikasian pada data set yang sama. Tabel 8 menyajikan hasil uji *McNemar Test*.

Tabel 8. Hasil uji *McNemar Test*

Uji McNemar	p < 0.05	Kesimpulan
LSTM-GRU	0.00	Signifikan
LSTM-Bi-LSTM	0.00	Signifikan
GRU-Bi-LSTM	0.00	Signifikan

Hasil uji *McNemar Test* menunjukkan terdapat perbedaan signifikan secara statistik dengan nilai *p-value* untuk semua perbandingan dibawah ambang batas signifikansi 0,05. Hal ini mengindikasikan bahwa perbedaan kinerja yang diamati diantara ketiga arsitektur, bukan disebabkan oleh variasi acak melainkan merefleksikan perbedaan kemampuan prediktif masing-masing model. Berdasarkan perbandingan evaluasi model secara keseluruhan, model GRU menawarkan keseimbangan kinerja terbaik di semua metrik, menjadikannya pilihan yang paling *robust* untuk prediksi hujan.

3.7. Analisis Perbandingan Komputasi Model

Efisiensi komputasi yang diukur melalui waktu pelatihan dan sumber daya merupakan aspek penting dalam pemilihan model. Analisis efisiensi komputasi dari ketiga model menunjukkan perbedaan signifikan dalam waktu pelatihan dan penggunaan RAM puncak. perbandingan komputasi dapat dilihat pada tabel 9.

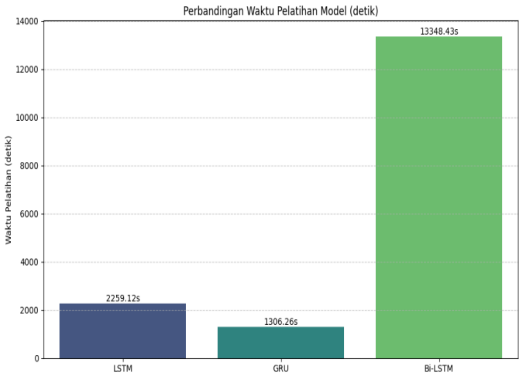
Tabel 9. Perbandingan Komputasi Model

Evaluasi	LSTM	GRU	Bi-LSTM
Waktu Pelatihan (detik)	2.259,12	1.306,26	13.348,43
Penggunaan RAM Puncak (MB)	1.971,47	2.053,77	2.242,60

Model GRU menunjukkan efisiensi tertinggi dengan waktu pelatihan tercepat 1.306,26 detik. Kinerja ini sebagai keunggulan GRU dalam hal kecepatan

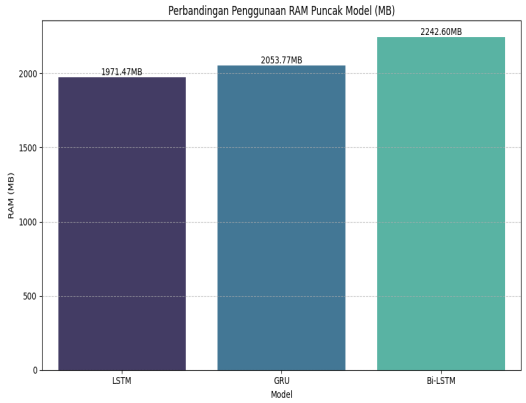
P-ISSN : 2089-3353
E-ISSN : 2808-9162

komputasi karena arsitekturnya yang lebih sederhana dibandingkan LSTM sehingga mengurangi jumlah parameter yang dihitung selama proses *backpropagation*. Model LSTM memerlukan waktu pelatihan sedikit lebih lama 2.259,12 detik. Meskipun lebih lambat dari GRU, waktu ini masih berada dalam kisaran yang dapat diterima mengingat kompleksitas internalnya yang lebih tinggi untuk mengatasi masalah *vanishing gradient.* Model Bi-LSTM paling lambat dengan waktu pelatihan 13.348,43 detik. Peningkatan waktu pelatihan pada Bi-LSTM sebagai konsekuensi dari sifat bidireksional yang melibatkan pelatihan dua lapisan (*forward* dan *backward*). Hal ini secara langsung menggandakan beban komputasi per *timestep* sehingga memperpanjang total durasi pelatihan. Perbandingan ini menunjukkan adanya *trade-off* yang jelas antara kompleksitas arsitektur model dan efisiensi waktu pelatihan, dimana model yang lebih kompleks seperti Bi-LSTM memberikan kemampuan pemodelan konseptual yang lebih kaya namun dengan biaya komputasi yang lebih tinggi. Grafik perbandingan waktu komputasi ketiga model dapat dilihat pada gambar 15.



Gambar 15. Perbandingan Waktu Komputasi

Analisis penggunaan RAM puncak selama pelatihan merupakan metrik penting untuk mengevaluasi efisiensi komputasi model terutama dalam lingkungan dengan sumber daya terbatas. Model LSTM menggunakan RAM puncak terendah yaitu 1.971,47 MB. Hal ini mengindikasikan bahwa arsitektur LSTM memiliki jejak memori yang efisien. Model GRU menggunakan RAM puncak sedikit lebih tinggi 2.053,77 MB. Meskipun sedikit lebih tinggi, angka ini masih mencerminkan efisiensi yang baik konsisten dengan desain GRU yang lebih sederhana dibandingkan LSTM. Sebaliknya, Bi-LSTM menggunakan RAM puncak tertinggi 2.242,60 MB. Penggunaan memori yang tertinggi ini karena sifat bidireksionalnya yang melatih dua lapisan LSTM terpisah untuk setiap *timestep*, sehingga membutuhkan sumber daya memori lebih besar. Perbedaan dalam penggunaan RAM ini menunjukkan *trade-off* antara kompleksitas model dan efisiensi sumber daya, dimana model yang lebih kompleks cenderung memerlukan alokasi memori yang lebih besar. Grafik perbandingan penggunaan RAM puncak pelatihan dapat dilihat pada gambar 16.



Gambar 16. Perbandingan Penggunaan RAM

Relevansi memahami karakteristik model dari segi waktu komputasi dan penggunaan sumber daya RAM sangat penting untuk *deployment* pada perangkat *edge* atau sistem *real time* dengan keterbatasan memori, serta untuk mengoptimalkan biaya infrastruktur komputasi.

Dengan demikian, pada konteks prediksi hujan jika prioritas utama adalah deteksi kejadian hujan yang efektif dengan *recall* tinggi dan efisiensi sumber daya yang optimal, maka model GRU merupakan pilihan yang paling unggul. LSTM menjadi alternatif yang kuat dengan kinerja kompetitif dan penggunaan RAM yang efisien. Sedangkan Bi-LSTM kurang optimal karena membutuhkan komputasi yang tinggi dan *recall* relatif rendah.

4. Kesimpulan

Penelitian ini berhasil membangun model prediksi hujan harian dan mengevaluasi dengan membandingkan kinerja serta efisiensi komputasi dari *Deep Learning* model LSTM, GRU, dan Bi-LSTM dengan fokus pada penanganan ketidakseimbangan kelas menggunakan SMOTE. Hasil penelitian menjawab RQ1, bahwa penerapan SMOTE berhasil menyeimbangkan distribusi kelas dan memungkinkan model untuk belajar dari representasi data yang lebih proporsional. Selain itu secara signifikan meningkatkan kemampuan model dalam mengidentifikasi kejadian hujan sebenarnya ditandai dengan peningkatan *recall* GRU mencapai (0,70), LSTM (0,67), dan Bi-LSTM (0,57) hal ini menunjukkan bahwa model menjadi lebih sensitif terhadap kelas minoritas dan meminimalisir risiko *false negative* atau hujan terlewat. Evaluasi model menggunakan *Area Under Curve* (AUC-ROC) menunjukkan kemampuan diskriminatif yang baik GRU memperoleh nilai (0,85), LSTM (0,84), dan Bi-LSTM sebesar (0,84). Analisis perbandingan untuk menjawab RQ2, menunjukkan GRU unggul dengan *recall* tertinggi di dukung dengan waktu pelatihan tercepat (1.306,26 detik) dan penggunaan RAM yang efisien (2.053,77 MB). Model LSTM menunjukkan kinerja sangat kompetitif dengan *overall accuracy* dan *F1-score* yang serupa dengan GRU, serta penggunaan

P-ISSN : 2089-3353
E-ISSN : 2808-9162

RAM terendah (1.971,47 MB) menjadikannya alternatif yang kuat meskipun dengan waktu pelatihan sedikit lebih lama (2.259,12 detik). Sementara itu model Bi-LSTM meskipun mencapai *overall accuracy* tertinggi mengalami *trade-off* yang signifikan pada efisiensi komputasi dengan waktu pelatihan terlama (13.348,43 detik) dan penggunaan RAM tertinggi (2.242,60 MB) serta *recall* terendah.

Berdasarkan temuan ini performa model GRU paling unggul dan direkomendasikan dalam aplikasi prediksi hujan yang memprioritaskan deteksi efektif kejadian hujan dan efisiensi sumber daya komputasi. LSTM sebagai alternatif dengan performa cukup stabil berada sedikit dibawah GRU, dan Bi-LSTM performa kurang optimal dari kinerja dan efisiensi komputasi.

Penelitian selanjutnya, disarankan untuk mengekplorasi arsitektur hibrida atau teknik *ensemble learning* yang lebih canggih untuk menangkap pola temporal dan spasial yang lebih kompleks.

Daftar Rujukan

[1] M. Maharani and R. Rajakumari, "Optimizing Rainfall Prediction: A Machine Learning Model Comparison," in *2025 Third International Conference on Emerging Applications of Material Science and Technology (ICEAMST)*, 2025, pp. 1141–1147. https://doi.org/10.1109/ICEAMST67459.2025.11335660

[2] A. Yadav, J. Singh, D. Joshi, A. K. Pradhan, and others, "Hybrid Artificial Intelligence-Based Approaches for Rainfall Forecasting," in *2025 International Conference on Artificial intelligence and Emerging Technologies (ICAIET)*, 2025, pp. 1–6. https://doi.org/10.1109/ICAIET65052.2025.11210929

[3] M. J. Alam and A. Majumder, "The statistical analysis of rainfall trend and its variability (1901--2020) in Kolkata, India," *Bull. Geogr. Phys. Geogr. Ser.*, no. 23, pp. 5–16, 2022. https://doi.org/10.12775/bgeo-2022-0006

[4] F. Furizal, A. B. Fawait, H. Maghfiroh, A. Ma'arif, A. A. Firdaus, and I. Suwarno, "Long short-term memory vs gated recurrent unit: a literature review on the performance of deep learning methods in temperature time series forecasting," *Int. J. Robot. Control Syst.*, vol. 4, no. 3, pp. 1506–1526, 2024. https://doi.org/10.31763/ijrcs.v4i3.1546

[5] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Comput. Sci.*, vol. 2, no. 3, pp. 1–21, 2021. https://doi.org/10.1007/s42979-021-00592-x

[6] A. Casolaro, V. Capone, G. Iannuzzo, and F. Camastra, "Deep learning for time series forecasting: Advances and open problems," *Information*, vol. 14, no. 11, p. 598, 2023. https://doi.org/10.3390/info14110598

[7] W. Fang, Q. Xue, L. Shen, and V. S. Sheng, "Survey on the application of deep learning in extreme weather prediction," *Atmosphere (Basel).*, vol. 12, no. 6, p. 661, 2021. https://doi.org/10.3390/atmos12060661

[8] I. Ayus, N. Natarajan, and D. Gupta, "Comparison of machine learning and deep learning techniques for the prediction of air pollution: a case study from China," *Asian J. Atmos. Environ.*, vol. 17, no. 1, p. 4, 2023. https://doi.org/10.1007/s44273-023-00005-w

[9] J. Patil and N. Baloorkar, "Weather Predictions using Machine Learning Techniques," in *2025 3rd International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, 2025, pp. 743–749. https://doi.org/10.1109/ICSCDS65426.2025.11166990

[10] N. Hayaty, H. Kurniawan, M. R. Rathomi, F. Chahyadi, and M. Bettiza, "Rainfall prediction with support vector machines: a case study in Tanjungpinang City, Indonesia," in *Bio web of conferences*, 2023, p. 1003. https://doi.org/10.1051/bioconf/20237001003

[11] K. Gupta, R. K. Gupta, and A. Gupta, "Daily, Weekly and Monthly Rain Forecasting using Deep Learning," in *2024 3rd International Conference for Innovation in Technology (INOCON)*, 2024, pp. 1–6. https://doi.org/10.1109/INOCON60754.2024.10511700

[12] G. D. Aryoso, B. Kanata, and M. S. Yadnya, "Rainfall Prediction Using Gate Recurrent Unit (Gru) for The Mataram City Area," *J. Penelit. Pendidik. IPA*, vol. 11, no. 2, pp. 1114–1119, 2025. https://doi.org/10.29303/jppipa.v11i2.9874

[13] M. L. Hawali and W. Walid, "Pemilihan Neuron LSTM dan LSTM Bayesian Optimization Untuk Prediksi Curah Hujan Bulanan Berbasis Iklim," *J. Fasilkom*, vol. 15, no. 2, pp. 376–382, 2025. https://doi.org/10.37859/jf.v15i2.9251

[14] S. H. Setiyani and Y. Rahma, "Penerapan Dekomposisi Matriks untuk Reduksi Kompleksitas Komputasi pada Algoritma Machine Learning," *J. FASILKOM*, vol. 16, no. 1, pp. 79–85, 2026. https://doi.org/10.37859/jf.v16i1.11166

[15] D. Leni and others, "Analisis Heatmap Korelasi dan Scatterplot untuk Mengidentifikasi Faktor-Faktor yang Mempengaruhi Pelabelan AC efisiensi Energi," *J. Rekayasa Mater. Manufaktur dan Energi*, vol. 6, no. 1, pp. 41–47, 2023. https://doi.org/10.30596/rmme.v6i1.13133

[16] P. S. Sai, T. M. Thiyagu, and P. G. Soumya, "Rainfall Prediction using Machine Learning Algorithms," in *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 2024, pp. 677–681. https://doi.org/10.1109/ICAAIC60222.2024.10575126

[17] I. Chahid, A. K. Elmiad, and M. Badaoui, "Data preprocessing for machine learning applications in healthcare: a review," in *2023 14th international conference on intelligent systems: theories and applications (SITA)*, 2023, pp. 1–6. https://doi.org/10.1109/SITA60746.2023.10373591

[18] S. C\ualin, "Handling Imbalanced Data: the SMOTE Technique," in *2025 17th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2025, pp. 1–5. https://doi.org/10.1109/ECAI65401.2025.11095450

[19] R. Mohammed and others, "FADA-SMOTE-Ms: Fuzzy adaptative smote-based methods," *IEEE Access*, vol. 12, pp. 158742–158765, 2024. https://doi.org/10.1109/ACCESS.2024.3480848

[20] H. K. Mohajan, "Deep Learning: A Brief Study on Its Architectures and Applications," *Art Soc.*, vol. 4, no. 8, pp. 41–49, 2025. https://doi.org/10.63593/AS.2709-9830.2025.09.003

[21] V. Sharmila, S. Kannadhasan, A. R. Kannan, P. Sivakumar, and V. Vennila, *Challenges in Information, Communication and Computing Technology: Proceedings of the 2nd International Conference on Challenges in Information, Communication, and Computing Technology (ICCICCT 2024), April 26th \& 27th, 2024, Namakkal, Tamil Nadu, India*. CRC Press, 2024. https://doi.org/10.1201/9781003559092

[22] M. Krichen and A. Mihoub, "Long short-term memory networks: A comprehensive survey," *AI*, vol. 6, no. 9, p. 215, 2025. https://doi.org/10.3390/ai6090215

[23] U. Rawat and C. S. Rai, "Enhancing CNN-Attention-LSTM with Novel LSTM Gate Mechanism: Evaluation on Benchmark and Brain Tumor Imaging Datasets," 2024. https://doi.org/10.21203/rs.3.rs-5392434/v1

[24] H. Asrawi, A. Sunyoto, and B. Setiaji, "Implementation of Bidirectional Gated Recurrent Units for Text Classification," in *2023 6th International Conference on Information and Communications Technology (ICOIACT)*, 2023, pp. 464–469. https://doi.org/10.1109/ICOIACT59844.2023.10455822

[25] M. L. Kurniawan, A. P. Sari, and E. Y. Puspaningrum, "Air Quality Prediction using a BiLSTM-Based Approach for Sustainable Environmental Management," *bit-Tech*, vol. 8, no. 2, pp. 2555–2564, 2025. https://doi.org/10.32877/bt.v8i2.3286

[26] H. Zhao, G. Zhang, M. Du, and X. Wang, "Improving Global Precipitation in Numerical Weather Prediction Systems based on Deep Learning Techniques," in *2023 IEEE International Conference on High Performance Computing \& Communications, Data Science \& Systems, Smart City \& Dependability in Sensor, Cloud \& Big Data Systems \&*

P-ISSN : 2089-3353
E-ISSN : 2808-9162

Application (HPCC/DSS/SmartCity/DependSys), 2023, pp. 333–339. https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys60770.2023.00053

[27] S. Albelali and M. Ahmed, "Hidden Leaks in Time Series Forecasting: How Data Leakage Affects LSTM Evaluation Across Configurations and Validation Strategies," *arXiv Prepr. arXiv2512.06932*, 2025. https://doi.org/10.48550/arXiv.2512.06932

[28] A. Ichwani, R. I. Kesuma, A. Setiawan, I. E. Wicaksono, and R. Hanifah, "Preventing Data Leakage in Classification via Integrated Machine Learning Pipelines: Preprocessing, Feature Transformation, and Hyperparameter Tuning," *J. Tek. Inform.*, vol. 7, no. 1, pp. 391–410, 2026. https://doi.org/10.52436/1.jutif.2026.7.1.5490.