



ANALISIS PERBANDINGAN KINERJA MODEL LONG SHORT-TERM MEMORY DAN RECURRENT NEURAL NETWORK DALAM PREDIKSI CUACA BERBASIS DATA CUACA REAL-TIME

Abdurrahman¹⁾, Suhirman²⁾

¹ Program Studi Informatika, Universitas Teknologi Yogyakarta

² Program Studi Magister Teknologi Informasi, Universitas Teknologi Yogyakarta

email: ¹ dursamiii19@gmail.com, ² suhirman@uty.ac.id

ARTICLE INFO

Article History:

Received : 20 Oktober 2025

Accepted : 3 November 2025

Published : 21 Desember 2025

Keywords:

Deep Learning

RNN

LSTM

Weather

Rainfall

IEEE style in citing this article:
Abdurrahman, Suherman, "Analisis Perbandingan Kinerja Model Long Short-Term Memory Dan Recurrent Neural Network Dalam Prediksi Cuaca Berbasis Data Cuaca Real-Time", *jurnalilmiah informatika*, vol. 10, no. 2, pp. 97-104, Des. 2025.

ABSTRACT

Unpredictable weather changes pose a major challenge in various sectors, including agriculture, transportation, and construction. Inaccurate rainfall predictions, especially on a local scale, often hamper community activities and decision-making that depend on weather conditions. This study aims to compare the performance of two artificial neural network models, namely Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN), in predicting rainfall based on hourly weather data collected in real-time using an ESP32 microcontroller equipped with BME280 and BH1750 sensors. The variables used include air temperature, humidity, rainfall, and light intensity. Both models were trained to predict weather conditions for the next few hours based on observation data that had been processed and normalized numerically. The evaluation was using three main metrics, namely Mean Squared Error (MSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). The results show that the LSTM model performed better with an MAE of 0.684, MSE of 0.7343, and R^2 of 0.2421, while the RNN model obtained an MAE of 0.2187, MSE of 0.3422, and R^2 of 0.8213. These findings prove that LSTM is more stable, efficient, and accurate in capturing the temporal patterns of weather data. This system has the potential to become the basis for developing local weather forecasts based on real-time data that are more adaptive to environmental changes.

Corresponding Author:

Abdurrahman

Universitas Teknologi
Yogyakarta

1. PENDAHULUAN

Cuaca yang tidak menentu sering kali menjadi kendala dalam berbagai aspek kehidupan, mulai dari aktivitas yang dilakukan sehari-hari hingga ke sektor ekonomi [1]. Masyarakat sering kali menghadapi kesulitan dalam merencanakan kegiatan yang berketergantungan terhadap cuaca seperti bepergian, menjemur pakaian, berolahraga, dan aktivitas luar ruangan lainnya [2]. Ketidakpastian cuaca secara signifikan berdampak juga terhadap sektor transportasi, konstruksi dan pertanian karena dapat mengganggu jadwal operasional maupun keamanan kerja [3]. Oleh karena itu, sistem yang mampu memprediksi cuaca secara akurat menjadi kebutuhan yang semakin penting, terutama dalam konteks adaptasi terhadap perubahan iklim global [4].

Beberapa penelitian terbaru seperti yang dilakukan oleh Keisler [5] dan Kaifeng Bi dkk. [6] telah menunjukkan potensi model berbasis *deep learning* yang mampu melampaui metode perkiraan numerik tradisional dalam memperkirakan cuaca global jangka pendek hingga menengah dengan menggunakan data historis. Penelitian-penelitian tersebut umumnya menggunakan data global dengan resolusi *spatio-temporal* yang sangat tinggi, sehingga mampu menghasilkan prediksi dengan cakupan ruang dan waktu yang luas [7]. Namun, metode serupa yang diterapkan pada skala lokal dan *real-time* masih menjadi tantangan tersendiri menurut Hewson dkk. [8] dan Chae dkk. [9]. Sebagian besar penelitian masih berfokus menggunakan infrastruktur komputasi berskala besar dan data beresolusi tinggi untuk memprediksi cuaca global, sementara itu kebutuhan masyarakat sebenarnya lebih spesifik pada skala lokal, seperti prediksi hujan

untuk beberapa jam ke depan dalam wilayah tertentu [10].

Perkembangan terkini pada teknologi *Internet of Things* (IoT) membuka peluang untuk pengumpulan data lokal secara langsung menggunakan perangkat mikrokontroler seperti ESP32 yang mampu diintegrasikan dengan sensor-sensor yang dapat menangkap kondisi cuaca dengan kombinasi sensor seperti BME280 yang dapat menangkap suhu dan kelembaban, dan BH1750 yang berfungsi untuk menangkap intensitas cahaya [11]. Serta dengan bantuan sensor *tipping bucket rain gauge* yang berfungsi untuk pengukuran curah hujan [12]. Dengan bantuan mikrokontroler dan sensor-sensor tersebut memungkinkan pengumpulan data cuaca secara *real-time* bersifat lokal, dimana data yang akan dikumpulkan merupakan data yang sesuai dengan lokasi pengumpulannya sehingga hasil prediksi akan lebih relevan dengan kondisi aktual di lapangan dibandingkan data umum dari penyedia layanan cuaca daring [13]. Namun kualitas hasil prediksi tentunya tergantung pada algoritma *machine learning* atau pembelajaran mesin yang digunakan pada pemrosesan data tersebut [14].

Dalam penelitian lain yang dilakukan oleh Syahreza dkk. [15] juga membuktikan bahwa algoritma *machine learning* seperti *Support Vector Regression* (SVR), *Random Forest*, dan *Extreme Gradient Boosting* (XGBoost) dapat memberikan hasil cukup baik dalam prediksi beberapa parameter cuaca. Model-model tersebut efektif dalam mengolah data cuaca seperti kelembaban, suhu, intensitas cahaya matahari, dan curah hujan. Namun kekurangan dari model-model tersebut ada pada keterbatasan dalam menangkap pola hubungan waktu antar data cuaca yang bersifat sekuensial, sehingga akurasi

prediksi pada data cuaca *real-time* sering kali belum optimal.

Keterbatasan tersebut dapat diatasi dengan pendekatan berbasis *deep learning* seperti menggunakan *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM) yang dapat menjadi alternatif yang menjanjikan. Model RNN memiliki kemampuan mengenali pola historis antar waktu, menjadikannya cocok untuk digunakan untuk memprediksi dalam basis data deret waktu [16]. Namun model ini masih memiliki masalah seperti *vanishing gradient* yaitu hilangnya informasi pada data yang berdurasi panjang, serta masalah *exploding gradient* yaitu meningkatnya *weight* atau bobot selama pelatihan secara berlebihan [17]. Model LSTM hadir sebagai penyempurnaan dari RNN dengan memanfaatkan *memory cell* yang memiliki kemampuan mempertahankan informasi lebih lama, sehingga akan lebih stabil dalam pemrosesan data cuaca yang bersifat fluktuatif [18].

Meskipun cukup banyak penelitian yang memanfaatkan model RNN dan LSTM untuk memprediksi cuaca, banyak yang masih menggunakan data harian atau data dari sumber eksternal seperti BMKG dan Google Weather. Pendekatan tersebut masih belum merepresentasikan kondisi cuaca pada suatu wilayah spesifik, terutama daerah yang memiliki karakteristik mikroklimat yang unik [19]. Oleh karena itu, penelitian ini dilakukan dengan data cuaca per jam yang dikumpulkan melalui mikrokontroler dan sensor-sensor pendukungnya untuk kemudian dibuat perbandingan performa untuk model RNN dan LSTM serta perbandingan hasil prediksinya dengan aplikasi cuaca yang umum digunakan seperti Google Weather.

Penelitian ini bertujuan menganalisis dan membuat perbandingan performa

model RNN dan LSTM dalam prediksi cuaca yang berbasis data resolusi per-jam, serta membandingkannya dengan hasil perkiraan cuaca dari layanan eksternal guna menentukan model yang paling optimal untuk pengembangan sistem perkiraan cuaca lokal.

2. METODE PENELITIAN

Tahapan-tahapan utama dalam penelitian ini meliputi: pengumpulan data, *preprocessing* data, perancangan model RNN dan LSTM, serta evaluasi model.

2.1 Pengumpulan Data

Pengumpulan data dilakukan dengan perangkat ESP32 yang terhubung dengan sensor lingkungan yang memiliki peran masing-masing. Sensor BME280 digunakan untuk menangkap variabel suhu dan kelembaban, sensor BH1750 untuk menangkap intensitas cahaya dan sensor *tipping bucket* untuk mengukur curah hujan. Selain itu sebagai evaluasi dan perbandingan, hasil prediksi dari model akan dibandingkan dengan data perkiraan cuaca dari layanan eksternal *Google Weather* pada lokasi dan rentang waktu yang sama. Perbandingan ini bertujuan menilai sejauh mana model berbasis data dari sensor lokal mampu memberikan hasil prediksi yang sebanding dengan layanan perkiraan cuaca komersial yang telah banyak digunakan.

Tabel 1. Variabel Data

Variabel	Tipe Data	Jumlah Data
Notasi	Nama	
X1	Waktu	Datetime
X2	Suhu	Numerik
X3	Kelembaban	Numerik
X4	Intensitas Cahaya	Numerik
Y	Curah Hujan	Numerik

Penelitian ini menggunakan data yang berhasil didapat dari perangkat ESP32 dengan mengambil hasil

pembacaan sensornya setiap detik dan dirata-ratakan dalam bentuk per-jam. Tabel 1 menunjukkan hasil dari pembacaan sensor yang dilakukan dari awal Juli 2025 hingga akhir September 2025 dengan hasil yang terdiri dari waktu, suhu, kelembaban, intensitas cahaya, dan curah hujan. Adapun sampel data yang didapat dari aplikasi *Google Weather* pada 25 September 2025 dengan lokasi yang sama ditunjukkan pada Tabel 2.

Tabel 2. Sampel data *Google Weather*

Jam	Suhu	Cuaca
14.00	32	Cerah
15.00	30	Cerah
16.00	28	Cerah
17.00	27	Cerah
18.00	26	Hujan
19.00	25	Hujan
20.00	24	Hujan
21.00	24	Hujan
22.00	24	Hujan
23.00	23	Cerah
00.00	23	Cerah

2.2 Preprocessing Data

Selanjutnya tahap *preprocessing* dilakukan untuk menyiapkan data yang dihasilkan dari sensor untuk siap digunakan dalam pelatihan model. Proses *preprocessing* meliputi pembersihan data dari missing value dan data duplikat, transformasi kolom waktu menjadi format numerik berupa jam 0 hingga 23, normalisasi nilai, dan pembagian data untuk pelatihan dan pengujian model. Normalisasi nilai menggunakan metode *Min-Max Scaling*, yaitu mengubah nilai pada data numerik menjadi rentang nilai 0 hingga 1 untuk menghindari dominasi nilai pada fitur tertentu yang memiliki skala besar yang akan berpengaruh terhadap pelatihan model. Pada tabel 3 merupakan contoh data sebelum *preprocessing* yang mana kolom tanggal dan jam masih satu. Sedangkan pada tabel 4 merupakan hasil *preprocessing* dengan kolom tanggal dan jam dipisah dan hanya diambil kolom jam saja, kemudian dilakukan normalisasi untuk setiap nilai pada semua kolom.

Tabel 3. contoh data sebelum preprocessing

Datetime	Suhu	Kelembaban	Intensitas Cahaya	Curah Hujan
2025-09-15 14:00	27.4	68.0	676.9	0.3
2025-09-16 15:00	27.5	69.4	629.5	0.2
2025-09-17 16:00	27.8	69.0	629.5	0.3

Tabel 4. contoh data sesudah preprocessing

Jam	Suhu	Kelembaban	Intensitas Cahaya	Curah Hujan
0.4348	0.526	0.4887	0.6781	0.02
0.4784	0.5325	0.4959	0.6306	0.0133
0.5217	0.5519	0.49	0.7789	0.02

2.3 Arsitektur Model

Model *machine learning* yang digunakan untuk penelitian ini yaitu *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM) yang termasuk dalam jenis *deep learning*

berbasis *time series prediction*. Kedua model ini mampu memproses data urutan waktu dengan mempertimbangkan hubungan antar waktu sebelumnya, sehingga cocok digunakan untuk prediksi cuaca. Arsitektur model yang akan digunakan

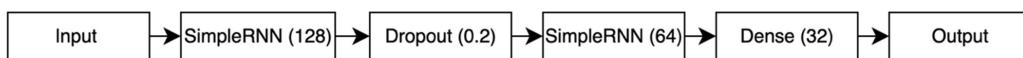
dalam penelitian ini direpresentasikan pada gambar 1 dan gambar 2.

Meskipun arsitektur model terlihat mirip, perbedaan utama keduanya terletak pada mekanisme pengolahan data sekuensial di dalam lapisan *looping* nya. Model RNN hanya menggunakan hidden state dari langkah sebelumnya untuk memperbarui informasi pada langkah saat ini, sehingga sangat memungkinkan untuk kehilangan informasi ketika urutan informasi terlalu

panjang, masalah ini disebut sebagai *vanishing gradient*. Sedangkan LSTM memiliki struktur internal sedikit lebih kompleks dengan adanya *cell* LSTM yang berisi tiga gerbang (*gate*), yaitu *input gate*, *forget gate*, dan *output gate* yang membuatnya mampu untuk menyimpan, memperbarui, atau menghapus informasi secara selektif sehingga mampu mempertahankan informasi jangka panjang.



Gambar 1. Arsitektur model LSTM



Gambar 2. Arsitektur model RNN

2.4 Evaluasi Model

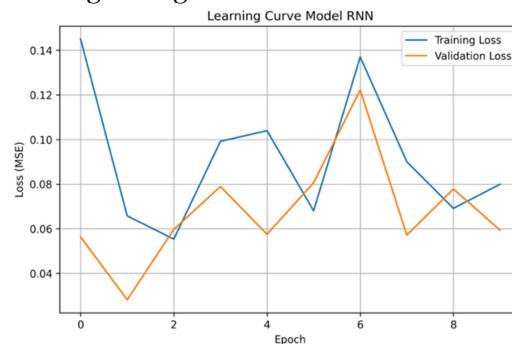
Metrik yang digunakan untuk evaluasi kedua model dalam penelitian ini, yaitu *Mean Square Error* (MSE) dan *Mean Absolute Error* (MAE) dengan penilaian semakin rendah maka semakin baik, serta nilai R^2 dengan penilaian semakin nilainya mendekati nilai 1 maka semakin baik sebuah model dalam memprediksi. Adapun sebagai perbandingan hasil prediksi model dengan aplikasi cuaca yang umum digunakan, peneliti menggunakan data dari *Google Weather* untuk melihat seberapa akurat hasil prediksi model.

3. HASIL DAN PEMBAHASAN

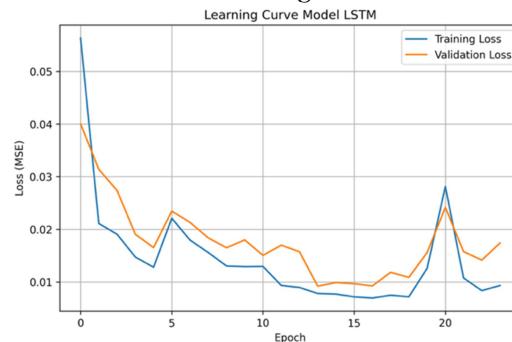
3.1 Hasil

Hasil dari pelatihan model RNN dan LSTM ditunjukkan dalam bentuk grafik *learning curve* pada gambar 3 dan gambar 4. Sumbu horizontal menggambarkan jumlah *epoch* selama proses pelatihan, sedangkan sumbu vertikal menunjukkan nilai *loss*. Nilai *loss* pada data pelatihan untuk model RNN

menunjukkan pola yang tidak stabil dan cenderung tinggi. Hal ini dikarenakan RNN lebih rentan terhadap *vanishing gradient* dan kehilangan informasi jangka panjang pada data yang memiliki ketergantungan antar waktu.



Gambar 3. Learning Curve RNN



Gambar 4. Learning Curve LSTM

Sementara itu, model LSTM menunjukkan penurunan *loss* yang lebih konsisten dan stabil seiring bertambahnya *epoch*. Baik training loss maupun validation loss terlihat menurun secara bertahap meskipun terdapat sedikit fluktuasi di akhir pelatihan. Ini membuktikan bahwa model LSTM mampu melakukan generalisasi dengan baik terhadap data yang diberikan. Dengan demikian, berdasarkan pola *learning curve*, dapat disimpulkan bahwa model LSTM memiliki performa pembelajaran yang lebih baik dibandingkan dengan model RNN dalam memproses data deret waktu untuk prediksi cuaca.

Tabel 4. Performa hasil evaluasi

Model	Metriks		
	MSE	MAE	R ²
RNN	0.7343	0.684	0.2421
LSTM	0.3422	0.2187	0.8213

Hasil evaluasi menggunakan data uji ditunjukkan pada tabel 4, diamana model LSTM memiliki performa jauh lebih baik dibandingkan dengan model RNN dengan hasil nilai R² 0.8213 untuk model LSTM dan 0.2421 untuk model RNN. Sedangkan evaluasi perbandingan hasil prediksi dapat dilihat pada tabel 5, yang menunjukkan perbandingan cuaca berdasarkan curah hujan yang berhasil diprediksi oleh model RNN dan LSTM dengan menetapkan angka dengan curah hujan diatas 0 sebagai hujan, dan 0 sebagai cerah.

Tabel 5. Perbandingan hasil prediksi

Jam	Aktual		RNN		LSTM		Google Weather
	curah hujan	cuaca	curah hujan	cuaca	curah hujan	cuaca	cuaca
14.00	0.0	cerah	0.3	hujan	0.0	cerah	cerah
15.00	0.0	cerah	0.7	hujan	0.0	cerah	cerah
16.00	0.5	hujan	0.0	cerah	0.0	cerah	cerah
17.00	0.7	hujan	0.0	cerah	0.1	hujan	cerah
18.00	0.6	hujan	0.0	cerah	0.3	hujan	hujan
19.00	0.1	hujan	0.3	hujan	0.5	hujan	hujan
20.00	0.0	cerah	0.0	cerah	0.0	hujan	hujan
21.00	0.0	cerah	1.2	hujan	0.0	cerah	hujan
22.00	0.0	cerah	1.5	hujan	0.0	cerah	hujan
23.00	0.2	hujan	0.0	cerah	0.6	hujan	cerah
00.00	0.4	hujan	0.0	cerah	0.3	hujan	cerah

3.1 Pembahasan

Berdasarkan hasil pengujian, model LSTM menunjukkan performa yang lebih baik dibandingkan dengan RNN dalam memprediksi kondisi cuaca, khususnya curah hujan. Perbandingan antara data hasil prediksi, data aktual, dan data dari *Google Weather* memperkuat temuan bahwa LSTM lebih

akurat dalam mengenali pola cuaca jangka pendek, sedangkan RNN masih menunjukkan tingkat kesalahan yang cukup tinggi dan belum mampu menangkap pola temporal secara efektif.

Performa LSTM yang lebih unggul ini disebabkan oleh kemampuannya dalam menangani *long-term dependency* pada data deret waktu (time

series). Arsitektur LSTM yang dilengkapi dengan *cell state* dan *gating mechanism* memungkinkan model untuk mempertahankan informasi penting dari periode sebelumnya, sehingga pola musiman atau tren curah hujan dapat dipelajari dengan lebih baik. Sebaliknya, RNN cenderung mengalami *vanishing gradient problem* ketika memproses urutan data yang panjang, menyebabkan model sulit mempertahankan informasi historis yang relevan.

Selama proses pelatihan, kurva *loss* pada model LSTM menunjukkan penurunan yang lebih stabil dan konsisten dibandingkan RNN. LSTM cenderung mencapai nilai *loss* yang lebih rendah pada *epoch* akhir, menandakan pembelajaran yang lebih efisien dalam menyesuaikan bobot terhadap pola data. Sebaliknya, RNN memperlihatkan fluktuasi nilai *loss* yang tinggi dan cenderung stuck di titik tertentu, yang menunjukkan kesulitan dalam mengoptimalkan parameter terhadap data yang bersifat non-linear.

Jika dilihat dari hasil evaluasi menggunakan metrik *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), dan koefisien determinasi (R^2), model LSTM memperoleh nilai R^2 sebesar 0.8213, yang menunjukkan bahwa sekitar 82% variasi data cuaca dapat dijelaskan oleh model ini. Nilai R^2 di atas 0.8 umumnya dikategorikan sebagai "baik" atau "kuat" dalam konteks prediksi data lingkungan, sementara nilai di bawah 0.5 dianggap lemah atau kurang representatif berdasarkan pada penelitian oleh Tatachar [20]. Sebagai perbandingan, model RNN hanya mencapai R^2 sebesar 0.2421, yang menandakan bahwa model tersebut belum mampu menjelaskan hubungan antar variabel dengan baik.

Dengan nilai kesalahan yang lebih rendah pada LSTM (MAE dan MSE yang kecil) dan R^2 yang tinggi, dapat dikatakan bahwa model ini memiliki kemampuan generalisasi yang lebih baik terhadap data baru. Artinya, prediksi yang dihasilkan oleh LSTM tidak hanya akurat terhadap data pelatihan, tetapi juga tetap konsisten ketika diuji dengan data yang belum dilihat sebelumnya oleh model.

Hasil ini juga selaras dengan penelitian sebelumnya oleh Syahreza dkk. [15] yang membandingkan model SVR, XGBoost, dan Random Forest, di mana model berbasis pohon seperti XGBoost lebih unggul dibandingkan SVR karena kemampuannya menangani hubungan non-linear antar variabel. Namun, penelitian ini menunjukkan bahwa model berbasis jaringan syaraf berulang seperti LSTM mampu melangkah lebih jauh lagi, terutama dalam memahami pola waktu dan fluktuasi antar jam yang sulit ditangkap oleh model konvensional.

Sementara itu, hasil penelitian ini menunjukkan bahwa model LSTM memiliki kemampuan yang lebih baik dibandingkan ketiga model tersebut, terutama dalam mengenali pola temporal dan hubungan jangka panjang pada data deret waktu. Mekanisme *gating* pada LSTM memungkinkan model mempertahankan informasi penting dari periode sebelumnya, yang membuatnya lebih efisien dalam memprediksi fenomena cuaca seperti curah hujan. Dengan demikian, penelitian ini memperkuat temuan sebelumnya sekaligus memberikan pembaruan dengan menunjukkan bahwa pendekatan berbasis jaringan syaraf berulang khususnya LSTM mampu menghasilkan performa prediksi yang lebih akurat dan stabil

pada data meteorologi yang bersifat dinamis.

4. KESIMPULAN

Dalam penelitian ini berhasil didapatkan simpulan setelah dilakukannya analisis berdasarkan hasil pelatihan dan pengujian, dapat disimpulkan bahwa model *Long Short-Term Memory* (LSTM) menghasilkan nilai dengan performa yang lebih baik dibandingkan *Recurrent Neural Network* (RNN) dalam memprediksi cuaca berbasis data deret waktu. Model LSTM menunjukkan nilai evaluasi yang lebih unggul dengan R^2 sebesar 0.8213 dan kesalahan yang lebih rendah dibandingkan RNN (R^2 sebesar 0.2421), serta memperlihatkan *loss curve* yang stabil dan menurun secara konsisten selama pelatihan. Keunggulan ini disebabkan oleh arsitektur LSTM yang mampu menangani *long-term dependency* melalui mekanisme *gating*, sehingga model dapat mengenali pola temporal dan musiman dengan lebih akurat. Hasil perbandingan dengan data aktual dan *Google Weather* juga menunjukkan bahwa LSTM mampu menghasilkan prediksi yang lebih mendekati kondisi cuaca sebenarnya. Dengan demikian, penelitian ini menegaskan bahwa pendekatan berbasis jaringan syaraf LSTM lebih efektif dan dapat lebih diandalkan dibandingkan RNN dalam konteks prediksi cuaca, sekaligus memperkuat temuan dari penelitian sebelumnya yang menyatakan bahwa model *deep learning* memiliki kemampuan yang lebih unggul dalam penanganan data cuaca yang kompleks dan dinamis.

5. REFERENSI

- [1] N. A. Putri, D. B. Ramadha, T. D. P. A. Kusuma, and H. Prayitno, "Pengaruh Faktor Cuaca Terhadap Perencanaan
- Penerbangan Dan Dampaknya Pada Keterlambatan Penerbangan," *J. Penelit.*, vol. 10, no. 1, pp. 19–27, 2025, doi: <https://doi.org/10.46491/jp.v10i1.1873>
- [2] H. Hajrina, T. Tahir, L. O. Amaluddin, and A. Andrias, "Dampak Perubahan Cuaca Pada Pendapatan Masyarakat Nelayan di Pantai Nelayan," *Penelit. Pendidik. Geogr.*, vol. 10, no. 3, pp. 298–307, 2025, doi: 10.36709/jppg.v10i3.330
- [3] J. Andrew, B. Koes, P. Cantik, A. Makmur, and M. S. Dharsono, "Jurnal Ilmiah Mahasiswa Teknik dan Desain Analisis Risiko Hujan terhadap Penyelesaian Pekerjaan Struktur pada Proyek Konstruksi," vol. 01, no. 01, 2025, doi: 10.51170/jimtd.v1i1.82.
- [4] P. A. G. Watson, "Machine learning applications for weather and climate need greater focus on extremes," *Environ. Res. Lett.*, vol. 17, no. 11, 2022, doi: 10.1088/1748-9326/ac9d4e.
- [5] R. Keisler, "Forecasting Global Weather with Graph Neural Networks," pp. 1–16, 2022, doi: 10.48550/arXiv.2202.07575.
- [6] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, "Accurate medium-range global weather forecasting with 3D neural networks," *Nature*, vol. 619, no. 7970, pp. 533–538, 2023, doi: 10.1038/s41586-023-06185-3.
- [7] J. Liu, L. Wu, T. Zhang, J. Huang, X. Wang, and F. Tian, "STFM: Accurate Spatio-Temporal Fusion Model for Weather Forecasting," *Atmosphere (Basel.)*, vol. 15, no. 10, pp. 1–17, 2024, doi: 10.3390/atmos15101176.
- [8] T. D. Hewson and F. M. Pillosu, "A low-cost post-processing technique improves weather forecasts around the world," *Commun. Earth Environ.*,

- vol. 2, no. 1, pp. 1–10, 2021, doi: 10.1038/s43247-021-00185-9.
- [9] S. Chae, J. Shin, S. Kwon, S. Lee, S. Kang, and D. Lee, “PM10 and PM2.5 real-time prediction models using an interpolated convolutional neural network,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–9, 2021, doi: 10.1038/s41598-021-91253-9.
- [10] L. Chen *et al.*, “FuXi: a cascade machine learning forecasting system for 15-day global weather forecast,” *npj Clim. Atmos. Sci.*, vol. 6, no. 1, pp. 1–11, 2023, doi: 10.1038/s41612-023-00512-1.
- [11] Y. K. Prayoga, A. Z. Arfianto, D. P. Riananda, Muhammad Khoirul Hasin, Adianto, and Ryan Yudha Adhitya, “Integrasi Sistem ESP32 dengan Pemantauan Cuaca Menggunakan Sensor Meteorologi,” *J. Appl. Smart Electr. Netw. Syst.*, vol. 6, no. 01, pp. 23–31, 2025, doi: 10.52158/jasens.v6i01.1149.
- [12] D. A. Segovia-Cardozo, C. Bernal-Basurco, and L. Rodríguez-Sinobas, “Tipping Bucket Rain Gauges in Hydrological Research: Summary on Measurement Uncertainties, Calibration, and Error Reduction Strategies,” *Sensors*, vol. 23, no. 12, 2023, doi: 10.3390/s23125385.
- [13] L. Huang *et al.*, “Combining Random Forest and XGBoost Methods in Detecting Early and Mid-Term Winter Wheat Stripe Rust Using Canopy Level Hyperspectral Measurements,” *Agric.*, vol. 12, no. 1, pp. 1–16, 2022, doi: 10.3390/agriculture12010074.
- [14] R. S. Nurhalizah, R. Ardianto, and P. Purwono, “Analisis Supervised dan Unsupervised Learning pada Machine Learning: Systematic Literature Review,” *J. Ilmu Komput. dan Inform.*, vol. 4, no. 1, pp. 61–72, 2024, doi: 10.54082/jiki.168.
- [15] A. Syahreza, N. K. Ningrum, and M. A. Syahrazy, “Perbandingan Kinerja Model Prediksi Cuaca: Random Forest, Support Vector Regression, dan XGBoost,” *Edumatic J. Pendidik. Inform.*, vol. 8, no. 2, pp. 526–534, 2024, doi: 10.29408/edumatic.v8i2.27640.
- [16] S. Nosouhian, F. Nosouhian, and A. K. Khoshouei, “A review of recurrent neural network architecture for sequence learning: Comparison between LSTM and GRU,” *Preprints.org*, no. July, pp. 1–7, 2021, doi: 10.20944/preprints202107.0252.v1
- [17] N. Zucchetti and A. Orvieto, “Recurrent neural networks: vanishing and exploding gradients are not the end of the story,” *Adv. Neural Inf. Process. Syst.*, vol. 37, no. Section 5, 2024, doi: 10.48550/arXiv.2405.21064.
- [18] C. B. Vennerød, A. Kjærnan, and E. S. Bugge, “Long Short-term Memory RNN,” 2021, doi: 10.48550/arXiv.2105.06756.
- [19] J. Oui, “Commodifying a ‘Good’ Weather Data: Commercial Meteorology, Low-cost Stations, and the Global Scientific Infrastructure,” *Sci. Technol. Hum. Values*, vol. 47, no. 1, pp. 29–52, 2022, doi: 10.1177/0162243921995889.
- [20] A. V. Tatachar, “Comparative Assessment of Regression Models Based On Model Evaluation Metrics,” *Int. Res. J. Eng. Technol.*, vol. 8, no. 9, pp. 853–860, 2021, [online] available: <https://www.irjet.net/archives/V8/i9/IRJET-V8I9127.pdf>