Prediksi Kemungkinan Hujan dengan menggunakan *Decision Tree*, *Logistic Regression*, dan LSTM

Hafizha Dini Giandra
Department of Informatics
Universitas Syiah Kuala
Banda Aceh, Aceh
giandra.23@mhs.usk.ac.id

Fadhilah Syafa
Department of Informatics
Universitas Syiah Kuala
Banda Aceh, Aceh
fadhilah.s@mhs.usk.ac.id

Muttaqin
Department of Informatics
Universitas Syiah Kuala
Banda Aceh, Aceh
muttaqin4@mhs.usk.ac.id

I. Introduction

Prediksi hujan masih menjadi tantangan besar bagi para ahli iklim. Hujan adalah komponen terpenting dari sistem iklim. Proses memprediksi terjadinya hujan di suatu wilayah memperhitungkan keakuratan prediksi, kesalahan dalam prediksi, dan estimasi volume curah hujan serta kemungkinan curah hujan di wilayah tertentu [1], sehingga untuk mengetahui hal-hal tersebut perlu mengumpulkan, menganalisis, dan melakukan penelitian terhadap berbagai data dan parameter meteorologi yang tersedia. Beberapa parameter dasar termasuk suhu (minimum dan maksimum), total curah hujan tahunan, dan kelembaban. Curah hujan dapat bervariasi dalam intensitas, variabilitas, dan frekuensi.

Kemungkinan terjadinya hujan sangat mempengaruhi alur kerja di berbagai bidang seperti perencanaan produksi pangan, pengelolaan sumber daya air, prediksi longsor dan/atau banjir, bahkan hal-hal trivial seperti perencanaan kegiatan sehari-hari terutama kegiatan *outdoor*. Prediksi hujan yang akurat dan tepat waktu diharapkan dapat memberikan tahap intervensi baru kepada sektor-sektor yang terkena dampak dari hujan. Namun, untuk perencanaan kegiatan sehari-hari, informasi apakah akan hujan atau tidak saja sudah cukup.

Permasalahan inilah yang menginspirasi kami untuk melakukan klasifikasi terkait kemungkinan hujan dengan menggunakan metode *machine learning*. *Machine learning* secara umum dapat didefinisikan sebagai metode komputasi yang menggunakan pengalaman untuk meningkatkan kinerja atau membuat prediksi yang akurat. Pengalaman pada *machine learning* mengacu pada informasi terdahulukan yang siap untuk dianalisis oleh sebuah metode. *Machine learning* memungkinkan komputer untuk menemukan solusi secara mandiri tanpa perlu diprogram secara eksplisit.

Penelitian ini dilakukan dengan menerapkan tiga metode *machine learning*, yaitu *Decision Tree*, *Logistic Regression*, dan *Long Short Term Memory* (LSTM) pada histori data cuaca dari dataset yang kami dapat. Performa diantara metode tersebut sangat bervariasi, sehingga memberikan ruang untuk peningkatan dengan memvariasikan rasio pelatihan dan pengujian [2]. Oleh karena itu, pemilihan metode *machine learning* yang sesuai dalam mengklasifikasikan hujan di suatu wilayah sangat penting.

Hasil dari penelitian ini adalah untuk mengetahui apakah dengan membandingkan metode *Decision Tree*, *Logistic Regression*, dan LSTM dapat digunakan untuk membantu memprediksi hujan.

II. Related Work

Pada penelitian sebelumnya yang dilakukan oleh Anwar et al. (2020), membandingkan kinerja empat algoritma dari *machine learning*, yaitu J48, *Random Forest*, *Naïve Bayes*, dan *Multilayer Perceptron* (MLP) untuk prediksi hujan. Data histori cuaca harian diperoleh dari situs BMKG untuk stasiun meteorologi di Tanjung Mas, Semarang, Indonesia, mulai tahun 2013 hingga 2019. Hasil dari penelitian ini menunjukkan bahwa algoritma MLP dan J48 dapat memberikan akurasi terbaik (hingga 78,4%) dibandingkan dengan algoritma lain, walaupun perbedaannya kecil [3].

Sanie et al. (2020) menggunakan *ensemble learning* untuk meningkatkan efektivitas prediksi curah hujan. *Ensemble learning* adalah pendekatan yang menggabungkan beberapa pengklasifikasi *machine learning* untuk prediksi curah hujan, yang mencakup *Naïve Bayes, Decision Tree, Support Vector Machine, Random Forest, dan Neural Network* berdasarkan dataset yang diperoleh dari *Drainage and Irrigation Department* dan *Malaysian Meteorological Department*. Lebih khusus lagi, penelitian ini mengeksplorasi tiga penggabungan aljabar, yaitu probabilitas rata-rata (*average probability*), probabilitas maksimum (*maximum probability*), dan pemungutan suara mayoritas (*majority voting*). Hasil menunjukkan bahwa metode *ensemble* berdasarkan *majority voting* sangat efektif dalam meningkatkan performa prediksi curah hujan dibandingkan dengan klasifikasi individual [4]. Nilai dari *precision*, *recall*, dan F-*Measures* yang diperoleh dari *majority voting* adalah 71%, 77%, dan 76% dimana hasil tersebut lebih tinggi dibandingkan dengan penggabung aljabar lainnya.

Penelitian selanjutnya oleh Zhao et al. (2022) juga melakukan prediksi hujan berbasis *machine learning* menggunakan dataset Rain in Australia. Metode *machine learning* yang digunakan adalah *Decision Tree*, *Logistic Regression*, *Long Short Term Memory* (LSTM), *AdaBoost*, *Bagging Algorithm*, dan *K-Nearest Neighbor* (KNN) dengan K *value* dalam *range* 1 hingga 20. Penelitian tersebut memperoleh akurasi tertinggi yaitu 85% dengan metode *Logistic Regression* diikuti oleh metode *AdaBoost* dengan akurasi 82%, akan tetapi hasil tersebut tidak sesuai dengan ekspektasi. Terdapat beberapa hipotesis tentang alasan mengapa akurasi tersebut tidak dapat ditingkatkan lebih lanjut. Penggabungan beberapa model untuk isu tertentu merupakan cara yang efektif untuk mengklasifikasikan. Jadi salah satu alasan mengapa akurasi tidak dapat ditingkatkan adalah karena model terlalu lemah dan orisinal [6], sehingga sampai batas tertentu masih belum dapat memprediksi apakah akan terjadi hujan.

Hudnurkar and Rayavarapu (2022) mengeksplorasi penggunaan algoritma *Support Vector Machine* (SVM) dan *Artificial Neural Network* (ANN) untuk klasifikasi biner curah hujan. Data klasifikasi curah hujan harian yang digunakan dalam penelitian ini diperoleh dari *National Data Center* (NDC) IMD. Dataset tersebut diperoleh dari tahun 2000 hingga 2018 untuk stasiun Shivajinagar di Pune, Maharashtra,

India. Akurasi klasifikasi yang diperoleh 82,1% dan 82,8% dengan algoritma SVM dan ANN, untuk dataset yang tidak seimbang. Sementara parameter kinerja seperti *misclassification rate* dan F1-*score* menunjukkan bahwa hasil yang lebih baik dicapai dengan ANN, pemilihan parameter model untuk SVM kurang terlibat dibandingkan ANN [7]. Teknik adaptasi domain digunakan untuk klasifikasi curah hujan di dua stasiun lainnya di Nashik dan Chikalthana dengan jaringan yang dilatih untuk stasiun Shivajinagar. Hasil yang memuaskan kedua stasiun ini baru diperoleh setelah adanya perubahan metode *training* dari SVM dan ANN.

III. METHODOLOGY

Machine learning adalah salah satu teknologi terkini yang paling menarik. *Machine learning* telah diposisikan untuk mengatasi kekurangan kognisi manusia serta pemrosesan informasi, khususnya dalam menangani data yang besar, hubungannya, dan analisisnya. Secara umum, *machine learning* mempelajari penelitian dan konstruksi algoritma yang dapat dipelajari, dan memperoleh prediksi tentang data. Oleh karena itu, pendekatan *machine learning* dipilih untuk memprediksi hujan.

Metodologi penelitian yang digunakan dalam penelitian ini dibagi menjadi empat fase berbeda. Fase pertama adalah fase dataset, dimana mengidentifikasi data secara manual dengan menganalisis sumber, jumlah, dan detail lainnya. Selanjutnya, fase *pre-processing* mempersiapkan data untuk diproses lebih lanjut dengan membersihkan data (yaitu, mengatasi *missing value*) dan menormalkan data. Data yang telah diproses sebelumnya kemudian digunakan pada fase ketiga untuk dianalisis menggunakan dua metode *machine learning* guna mengidentifikasi metode terbaik dari dua pengklasifikasi *machine learning* tersebut. Fase keempat dan terakhir berfokus pada penilaian terhadap kinerja metode. Masing-masing dari keempat fase ini dijelaskan lebih lanjut pada subbagian berikut.

A. Dataset

Dataset yang digunakan pada penelitian ini berasal dari situs Kaggle, yaitu Rain in Australia dataset. Dataset ini berisi hasil observasi cuaca harian dari berbagai lokasi di Australia selama 10 tahun (2007-2017). Dataset tersebut memiliki 145.640 baris data.

Terdapat 23 fitur pada dataset ini seperti lokasi, suhu, kecepatan angin, kelembaban, dan lain sebagainya. Tujuh diantara fitur-fitur tersebut termasuk label kelas, adalah data nominal dan sisanya adalah data numerik.

B. Pre-processing

Sebelum proses klasifikasi dilakukan, data atau *record* dan atribut harus melalui beberapa tahap pengolahan awal data (*pre-processing*).

Data yang tidak lengkap atau hilang menimbulkan ketidakadilan atau bahkan kesalahan, oleh karena itu untuk memperoleh data yang berkualitas dilakukan beberapa tahapan *pre-processing* sebagai berikut.

- Semua atribut nominal, kecuali label kelas, disingkirkan karena tidak relevan dengan proses klasifikasi. Atribut-atribut tersebut adalah “Date”, tanggal observasi; “Location”, lokasi data diambil; “WindGustDir”, arah kecepatan angin tertinggi dalam periode 24 jam; “WindDir9am”, arah angin pada jam 9 pagi; “WindDir3pm”, arah angin pada jam 3 sore; dan “RainToday”, apakah hari ini hujan atau tidak.
- Pengambilan baris data tanpa nilai *null* atau NaN pada semua kolom. Dari total 145.460 baris dalam dataset mentah didapat 58.090 baris (~40% dari jumlah awal) yang sama sekali tidak memiliki nilai *null* atau NaN pada semua kolom.
- Dari hasil *pre-prosesing* sebelumnya kelas “*Yes*” memiliki sampel sebanyak 12.729 baris dan kelas “*No*” memiliki sampel sebanyak 45.361 baris. Seluruh baris dengan kelas “*Yes*” digunakan dan diambil sebanyak 12.729 baris dari kelas “*No*” untuk digunakan. Jumlah total dataset yang siap digunakan adalah 25.458 baris dengan 16 atribut numerik dan 1 atribut label kelas.
- Normalisasi dataset dilakukan dengan menggunakan metode *MinMaxScaler* dari *library Sklearn*. Fungsi *MinMaxScaler* mengubah seluruh nilai data menjadi dalam *range* antara 0 hingga 1.
- Pemisahan data *training* dan data *testing* dengan rasio 70:30 persen.

C. Decision Tree

Decision Tree (pohon keputusan) merupakan salah satu metode *machine learning* sederhana untuk klasifikasi dan regresi. Metode ini melakukan proses klasifikasi data berdasarkan fitur-fitur. Secara kasar proses pengklasifikasian dengan *decision tree* terlihat seperti kumpulan *syntax if-then*. Keuntungan metode ini adalah alur algoritmanya mudah dibaca dan dimengerti, serta kecepatan klasifikasi yang cepat.

Saat proses *training*, metode ini menganalisis data *training* untuk membuat model dengan menggunakan prinsip meminimalkan fungsi kerugian (*loss function*). Proses *training* metode ini secara umum dilakukan dalam tiga tahap, yaitu pemilihan fitur, pembuatan *decision tree*, dan *pruning* (pemangkasan). Berikut alur kerja proses pembelajaran metode *decision tree*.

1) Inisialisasi *node root* (akar).
2) Melakukan partisi *node* menggunakan atribut dengan nilai *information gain* tertinggi.
3) Lakukan proses (2) secara berulang hingga:
   - Semua sampel pada sub-*node* tersebut memiliki label kelas yang sama.
   - Mencapai kondisi yang telah ditetapkan sebelumnya, seperti maksimum kedalaman, atau batasan minimum sampel yang diperlukan untuk proses partisi.
   - Semua atribut telah digunakan untuk proses partisi.

D. Logistic Regression

Logistic regression adalah metode statistik yang biasanya digunakan untuk klasifikasi biner yaitu ketika variabel bersifat kategorikal atau diskrit dan hanya memiliki dua label kelas, seperti *Yes* atau *No*, 0 atau 1, dan *true* atau *false*.

Metode ini menggunakan fungsi logistik (juga dikenal sebagai fungsi sigmoid) untuk mengubah hasil perhitungan linear dari fitur masukan menjadi nilai antara 0 atau nilai 1. Nilai ambang batas (*threshold*) yang digunakan biasanya adalah 0,5. Jika nilai prediksi lebih besar atau sama dengan 0,5 maka data tersebut akan diklasifikasikan menjadi kelas 1, jika tidak maka akan diklasifikasikan menjadi kelas 0.

Kelebihan dari *logistic regression* adalah metode implementasinya sederhana dibandingkan dengan metode lain dan waktu pelatihannya lebih sedikit, meluas ke beberapa prediksi yang disebut *multinominal regression*, jika datanya dapat dipisahkan secara linier sehingga memberikan akurasi yang baik.

E. Long Short Term Memory

Dataset yang digunakan pada penelitian ini yaitu Rain in Australia, tidak hanya disusun berdasarkan lokasi di Australia tetapi juga berdasarkan tanggal. Hal ini memungkinkan bahwa prediksi cuaca mungkin bergantung pada cuaca sebelumnya dalam seminggu. *Long Short Term Memory* (LSTM) merupakan algoritma *deep learning* yang populer dan cocok digunakan untuk membuat prediksi dan klasifikasi yang berhubungan dengan waktu.

Algoritma ini merupakan pengembangan dari algoritma RNN (*Recurrent Neural Network*). Dalam algoritma RNN, *output* dari langkah terakhir diumpankan kembali sebagai *input* pada langkah yang sedang aktif. Namun, algoritma RNN memiliki kesulitan mempelajari dan menyimpan informasi dalam rangkaian yang panjang karena masalah gradien hilang, yang mana gradien kesalahan terhadap bobot menjadi sangat kecil, menyebabkan model kesulitan mempelajari informasi jauh di masa lalu. Algoritma LSTM mampu menyimpan informasi untuk jangka waktu yang lama. Hal ini kemudian dapat digunakan untuk memproses, memprediksi, dan mengklasifikasikan informasi berdasarkan data deret waktu.

Struktur algoritma LSTM terdiri atas *neural network* dan beberapa blok memori yang berbeda. Blok memori ini disebut sebagai *cell*. *State* dari *cell* dan *hidden state* akan diteruskan ke *cell* berikutnya. Informasi yang dikumpulkan oleh algoritma LSTM kemudian akan disimpan oleh *cell* dan manipulasi memori dilakukan oleh komponen yang disebut dengan *gate*. Ada tiga jenis *gate* pada algoritma LSTM, di antaranya *forget gate, input gate, dan output gate.*

F. Evaluation Metrics

Untuk tujuan mengevaluasi metode yang diusulkan, *information retrieval metrics* digunakan. Evaluasi dilakukan dengan menggunakan beberapa pengukuran yaitu. *accuracy*, *precision*, *recall*, dan F1-*score*. Semua pengukuran ini didasarkan pada *false positive* (FP), *false negative* (FN), *true positive* (TP), dan *true negative* (TN). *Precision* dan *recall* adalah pengukuran yang diperlukan untuk menunjukkan performa model untuk kelas tertentu (yang sangat berguna dalam dataset dengan kelas tidak seimbang, yang tidak dapat diketahui dengan *accuracy*.

Model yang didapat akan memprediksi 2 kelas, yaitu "*Yes*" yang berarti besok akan hujan dan "*No*." yang berarti besok tidak hujan. R2, SSE, dan MSE lebih baik untuk nilai kontinu, sedangkan model dalam penelitian ini tidak memprediksi keluaran seperti itu.

Accuracy didefinisikan sebagai total dari hasil yang diklasifikasikan dengan benar (TP dan TN) dibagi dengan semua hasil pengujian. *Accuracy* dapat dihitung menggunakan (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision mengevaluasi hasil *true positive* (TP) terhadap total hasil yang diprediksi positif (TP dan FP). *False positive* (FP) merupakan entitas yang diklasifikasikan secara salah, yang dapat dihitung menggunakan (2).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall digunakan dalam menilai hasil positif yang diklasifikasikan dengan benar (TP) terhadap total hasil yang positif pada kenyataannya (TP dan FN). Evaluasi ini dihitung seperti yang ditunjukkan pada (3).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Terakhir, F1-*score* dapat dilihat secara sederhana sebagai rata-rata dari *recall* dan *precision* yang menunjukkan performa model secara keseluruhan, dapat dihitung seperti yang ditunjukkan pada (4).

$$F1\text{-}score = \frac{2 \times precision \times recall}{precision + recall} \quad (4)$$

G. Potensi Penambahan Metode

Kami berencana melakukan penambahan metode *Random Forest* sehingga menggunakan dan melakukan perbandingan empat metode *machine learning* untuk pengklasifikasian hujan.

Random forest adalah metode pengembangan *decision* tree yang banyak digunakan untuk berbagai tujuan, termasuk regresi, klasifikasi, dan prediksi. *Random forest* merupakan kumpulan *decision tree* yang dibangun dengan nilai parameter, seleksi fitur, dan jumlah sampel yang acak dan independen antar pohon. Pembangunan pohon dengan cara ini mengecilkan kemungkinan terjadinya *overfitting* terhadap model yang dihasilkan. Seringnya terjadi *overfitting* merupakan salah satu kelemahan dari metode *decision tree*.

REFERENCES

[1] X. Zhang, S. N. Mohanty, A. K. Parida, S. K. Pani, B. Dong, and X. Cheng, "Annual and Non-Monsoon Rainfall Prediction Modelling Using SVR-MLP: An Empirical Study From Odisha," *IEEE Access*, vol. 8, pp. 30223-30233, February 2020.

[2] N. K. A. Appiah-Badu, Y. M. Missah, L. K. Amekudzi, N. Ussiph, T. Frimpong, and E. Ahene, "Rainfall Prediction Using Machine Learning Algorithms for the Various Ecological Zones of Ghana," *IEEE Access*, vol. 10, pp. 5069-5082, December 2021.

[3] M. T. Anwar, W. Hadikurniawati, E. Winarno, and W. Widiyatmoko, "Performance Comparison of Data Mining Techniques for Rain Prediction Models in Indonesia," *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 83-88, December 2020.

[4] N. S. Sani, A. H. A. Rahman, A. Adam, I. Shlash, and M. Aliff, "Ensemble Learning for Rainfall Prediction," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 11, November 2020.

[5] Y. Zhao, H. Shi Y. Ma, M. He, H. Deng, and Z. Tong, "Rain Prediction Based on Machine Learning," *Proceedings of the 2022 8th International Conference on Humanities and Social Science Research (ICHSSR 2022)*, vol. 664, pp. 2957-2970, Juni 2022.

[6] S. Hudnurkar and N. Rayavarapu, "Binary classification of rainfall time-series using machine learning algorithms," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 1945-1954, April 2022.