

HYPERPARAMETER TUNING LSTM SEBAGAI ESTIMATOR SENSOR RELATIVE HUMIDITY PADA AUTOMATIC WEATHER STATION BERBASIS SIMULATED ANNEALING HYPERPARAMETER TUNING LSTM AS RELATIVE HUMIDITY SENSOR ESTIMATOR ON AUTOMATIC WEATHER STATION BASED ON SIMULATED ANNEALING

**Naufal Ananda^{1,5}, Haryas Subyantara Wicaksana^{2,5}, Yusuf Giri
Wijaya^{3,5}, Rhakamerta Hijazi^{4,5}**

¹Balai Besar MKG Wilayah II Ciputat BMKG

²Pusat Instrumentasi Kalibrasi dan Reayasa BMKG

³Pusat Riset Teknologi Penerbangan, BRIN

⁴Pusat Riset Teknologi Deteksi Radiasi dan Analisis Nuklir, BRIN

⁵Instrumentasi dan Kontrol, Fakultas Teknologi Industri, Institut Teknologi Bandung
naufal.ananda@gmail.com ; haryas.wicaksana@bmkgo.id ; yusuf.giri.wijaya@brin.go.id
; rhakamerta@gmail.com

ABSTRAK

Pengukuran kelembapan udara relatif (RH) sebagai salah satu besaran cuaca dilakukan di lapisan permukaan menggunakan Automatic Weather Station (AWS). Pada tahun 2020, sensor RH AWS masih memiliki 7% tingkat unavailability karena kerusakan pencatu daya, kerusakan sensor serta gangguan jaringan komunikasi. Pada penelitian ini, dirancang estimasi nilai sensor RH AWS yang dapat dijadikan alternatif terhadap unavailability data RH AWS. Optimasi performa algoritma LSTM sebagai estimator RH dapat dilakukan melalui hyperparameter tuning berbasis simulated annealing (SA). Data diambil dari output sensor RH AWS Pemalang, Jawa Tengah. Model LSTM terbaru (LSTM-SA) selanjutnya digunakan sebagai estimator data sensor RH AWS Pemalang. Hasil estimasi sensor RH AWS Pemalang menggunakan model LSTM-SA kemudian dikomparasi terhadap model LSTM tanpa SA. Jumlah neuron optimal berdasarkan algoritma simulated annealing yaitu 48 neuron per hidden layer. Batch size optimal berdasarkan algoritma simulated annealing yaitu 21. Nilai RMSE 0,015% lebih rendah dibanding nilai RMSE awal tanpa hyperparameter tuning terhadap batch size. Algoritma LSTM-SA mampu mengoptimasi hyperparameter algoritma LSTM dengan menurunnya nilai RMSE. Nilai error yang dihasilkan masih kurang dari 3 %RH sesuai ketentuan dokumen World Meteorological Organization (WMO) No.8.

Kata kunci : Hyperparameter tuning, Simulated Annealing, Relative Humidity.

ABSTRACT

Air relative humidity (RH) measurement as surface meteorological quantity is held by utilizing Automatic Weather Station (AWS). In 2020, RH sensor has 7% of unavailability level caused by power supply or sensor fault, and troubled data transmission. In this research, an estimation of AWS RH sensor values is designed that can be used as an alternative to the unavailability of AWS RH data. LSTM algorithm can be optimized by tuning its hyperparameter based on simulated annealing (SA) method. RH sensor of AWS Pemalang is used as model dataset. LSTM is combined with SA as an estimator, called LSTM-SA. This estimation result is compared to LSTM without SA. LSTM-SA defines 48 neurons per hidden layer and 21 of batch sizes. RMSE of LSTM-SA is 0,015 %RH lower than LSTM without SA. LSTM-SA is succeed to optimize LSTM algorithm for RH sensor estimation. The error value is less than 3 %RH based on WMO No.8 document.

Keywords : Hyperparameter tuning, Simulated Annealing, Relative Humidity.

1. Pendahuluan

Kelembapan udara relatif (RH) adalah rasio tekanan uap air udara terhadap tekanan uap air jenuh dalam bentuk persentase (%RH). Pengukuran RH cukup penting untuk analisis serta prakiraan cuaca. Pengukuran ini sangat korelatif terhadap perubahan state air pada kandungan atmosfer. Pengukuran RH sebagai salah satu besaran cuaca dilakukan di berbagai lapisan atmosfer, termasuk di lapisan permukaan tanah.

Automatic Weather Station (AWS) merupakan instrumen digital yang berfungsi mengukur parameter cuaca di lapisan permukaan secara otomatis. AWS terdiri dari tujuh sensor elektrik, salah satunya sensor RH [1]. Sensor RH berupa membran kapasitif dengan output tegangan analog. Perubahan tingkat kejenuhan uap air direspon oleh perubahan jarak antar membran dielektris. Perubahan jarak antar membran menyebabkan perubahan nilai kapasitansi membran. Selanjutnya, sinyal tersebut diamplifikasi serta dikonversi menjadi output tegangan oleh *signal conditioner* pada *body sensor*. Tegangan output sensor diinputkan ke digitizer dalam logger sehingga kembali menjadi data RH [2].

Badan Meteorologi Klimatologi dan Geofisika (BMKG) telah menginstal sejumlah AWS di di setiap wilayah zona musim di Indonesia. Zona musim adalah daerah yang pola hujan rata-ratanya memiliki perbedaan signifikan antara periode musim kemarau dan periode musim penghujan. Namun demikian, tingkat ketersediaan data AWS menjadi kendala dalam analisis cuaca permukaan. Pada tahun 2020, sensor RH AWS masih memiliki 7% tingkat *unavailability* karena

kerusakan pencatu daya, kerusakan sensor serta gangguan jaringan komunikasi. Adanya *gap* data dapat mengurangi tingkat akurasi analisis prakiraan cuaca.

Estimasi output sensor RH AWS menjadi alternatif solusi dalam menekan data *unavailability*. Estimasi kelembapan udara dapat dirancang menggunakan data sensor temperatur dan sensor RH AWS. Metode estimasi identik dengan model prediksi berdasarkan riwayat data suatu parameter. Khan, Omar dan Shamshad (2019) mendesain MLP sebagai model prediksi RH di Lahore, Pakistan [3]. Selanjutnya, Shad, Sharma dan Singh (2022) juga mendesain MLP sebagai model prediksi RH bulanan di New Delhi, India dengan performansi 3,42% MAE. Model MLP cukup baik diaplikasikan dalam estimasi jangka pendek [4]. Pada tahun yang sama, Parasyris dkk. (2022) mendesain LSTM sebagai model prediksi RH harian di Crete, Yunani dengan performansi 9,42% MAE. LSTM memiliki performa cukup baik untuk diimplementasikan dalam model estimasi jangka pendek maupun jangka panjang [5].

Penelitian ini mengadaptasi model LSTM yang disusun oleh Parasyris dkk. Nilai error performa LSTM memiliki potensi untuk diminimalisir lebih lanjut. Performa LSTM sangat dipengaruhi oleh penentuan *hyperparameter algoritma* itu sendiri. Optimasi performa algoritma LSTM sebagai estimator RH dapat dilakukan dengan mendesain ulang model arsitekturnya. Sebagian penentuan *hyperparameter* dalam model arsitektur LSTM dirancang hanya berdasarkan uji coba trial dan error.

Tsai dkk. (2020) mendesain *hyperparameter tuning* pada

algoritma *Deep Learning* berbasis *Simulated Annealing* untuk bidang transportasi [6]. Kalliola, Dzikiene dan Damasevicius (2021) telah melakukan *hyperparameter tuning* pada algoritma MLP berbasis Bayesian Search untuk prediksi harga real estate di Helsinki [8]. *Hyperparameter tuning* pada algoritma kecerdasan buatan belum banyak dilakukan dalam bidang instrumentasi meteorologi seperti AWS.

Penelitian ini bertujuan melakukan optimasi dalam *hyperparameter tuning* algoritma LSTM sebagai estimator RH. Penelitian ini menggunakan metode *simulated annealing* karena estimasi sinyal RH AWS bersifat nonlinier. Data diambil dari output sensor RH AWS Pemalang, Jawa Tengah. Output sensor RH memiliki interval per 10 menit.

2. Metode

2.1. Simulated Annealing

Simulated annealing (SA) merupakan sebuah metode optimasi modern yang mengadaptasi simulasi pendinginan termal pada logam yang dipanaskan. Partikel atom logam akan bergerak bebas ketika dipanaskan pada temperatur tinggi dalam tingkat energi tertentu. Pergerakan atom menjadi semakin terbatas ketika temperatur logam didinginkan secara perlahan, sehingga atom logam mencapai stagnasi formasi tertentu dengan tingkat energi yang rendah. Kecepatan proses pendinginan mempengaruhi pembentukan formasi tersebut. *Simulated annealing* akan mencapai nilai minimum atau maksimum suatu fungsi objektif sebagaimana simulasi pendinginan logam membentuk formasi akhir dalam tingkat energi minimum. Optimasi fungsi berbasis *simulated*

annealing mengasumsikan parameter fungsi objektif sebagai temperatur. Parameter ini dikendalikan menggunakan konsep distribusi probabilitas Boltzmann. Konsep ini secara matematis dinyatakan pada Persamaan (1) sebagai berikut [7]:

$$P(E) = e^{-E/kT} \dots(1)$$

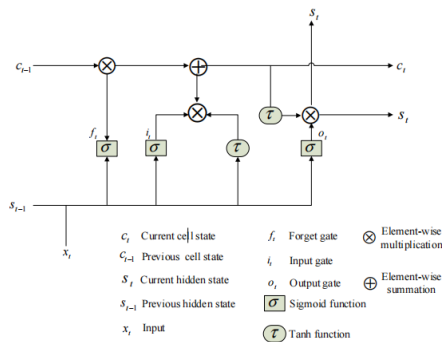
P(E) adalah probabilitas atom mencapai tingkat energi E, sementara k adalah konstanta Boltzmann sebesar $1,38 \times 10^{-23}$ J/K, dan T adalah nilai temperatur. Distribusi probabilitas Boltzmann mengacu pada sistem termodinamika, dengan nilai T yang semakin kecil, maka probabilitas atom mencapai tingkat energi tinggi akan semakin rendah, sehingga diperoleh nilai minimum fungsi objektif. Tabel 1 menunjukkan *pseudo code* algoritma *simulated annealing* [6].

Tabel 1. Algoritma *Simulated Annealing*

Algoritma :	Simulated Annealing
Input :	Fungsi objektif f(s), temperatur awal (T ₀), <i>cooling factor</i> (α), jumlah iterasi n
Output :	Solusi optimum fungsi objektif s
1	Inisialisasi s ₀ dan T ₀
2	Untuk i=0,.....,n-1 lakukan:
3	Ambil nilai s _{baru} dari tetangga terdekat s _i
4	ΔE = f(s _{baru}) - f(s _i)
5	P(E) = e ^{-E/kT}
6	Jika bilangan acak (0,1) < P(E), maka s _{i+1} = s _{baru} Jika tidak s _{i+1} = s _i akhiri fungsi if
7	T = α.T
8	Akhiri iterasi

2.2. Long Short Term Memory

Long Short Term Memory (LSTM) adalah algoritma jaringan syaraf tiruan yang mampu memprediksi suatu parameter *time series* dalam jangka panjang. LSTM merupakan pengembangan algoritma Recurrent Neural Network (RNN) yang memiliki sel memori di setiap neuronnya. Gambar 1 menunjukkan struktur internal memori sel (neuron) LSTM [5].



Gambar 1. Struktur Internal Memori Sel LSTM.

Model LSTM terdiri dari koneksi antar sel dalam tiga *gate*: *input gate*, *forget gate* dan *output gate*. *Forget gate* berfungsi mengatur arus informasi, sehingga LSTM mempunyai manajemen terhadap memori sel. Tabel 2 menunjukkan *pseudo code* algoritma LSTM [9].

Tabel 2. Algoritma LSTM.

Algoritma :	Long Short Term Memory
Input :	Input (x_t), <i>previous cell state</i> (c_{t-1}), dan <i>previous hidden state</i> (s_{t-1})
Output :	<i>Current cell state</i> (c_t) dan <i>current hidden state</i> (s_t)
1	Inisialisasi $x, c_{t-1}, s_{t-1} \rightarrow \text{ReLU}(x) = \max(0, 1)$
2	$n = \text{epoch}$
3	Untuk $i = 0, 1, \dots, n$, lakukan:
4	$f_t = \sigma(W_f \cdot [s_{t-1}, x_t] + b_f)$
5	$i_t = \sigma(W_i \cdot [s_{t-1}, x_t] + b_i)$

6	$c_t = \tanh(W_c \cdot [s_{t-1}, x_t] + b_c)$
7	$c_t = f_t * c_{t-1} + i_t * c_t$
8	$o_t = \sigma(W_o \cdot [s_{t-1}, x_t] + b_o)$
9	$s_t = o_t * \tanh(c_t)$
10	Hitung RMSE antara c_t dan c_{aktual}
11	Akhiri iterasi

2.3. Hyperparameter Tuning pada Algoritma LSTM

Hyperparameter merupakan spesifikasi parameter desain suatu algoritma kecerdasan buatan. Parameter algoritma LSTM meliputi jumlah neuron per layer, jumlah epoch dan ukuran *batch*. *Hyperparameter tuning* adalah proses menentukan parameter desain optimal untuk suatu algoritma dalam memecahkan permasalahan tertentu, salah satunya menggunakan metode *simulated annealing* [9]. Tabel 3 menunjukkan model arsitektur LSTM awal yang digunakan untuk mengestimasi output sensor RH AWS [6].

Tabel 3. Model Arsitektur LSTM.

Jumlah <i>hidden layer</i>	2
Jumlah <i>neuron per hidden layer</i>	40
<i>Batch size</i>	30
Jumlah <i>epoch</i>	50
Optimizer	Adam
Fungsi aktivasi	Rectifier Linear Unit (ReLU)

Model LSTM disimulasikan terhadap data output sensor RH AWS Pemalang pada periode Januari - Maret 2021. Output sensor RH merupakan data kelembapan udara per 10 menit. Data dibagi menjadi 80% data training dan 20% data testing. Input model LSTM sebagai estimator data RH menggunakan riwayat data temperatur dan RH AWS Pemalang. Temperatur memiliki korelasi tinggi terhadap kelembapan relatif. Selain itu, sensor RH terinstalasi satu *enclosure* dengan

sensor temperatur AWS. Tabel 4 menunjukkan input model LSTM tersebut. Parameter delay input diadaptasi dari penelitian Parasyris (2022) [5].

Tabel 4. Parameter Input Model LSTM.

Parameter Input	Delay
T - Temperatur Udara (°C)	(t - 3), (t - 2), (t - 1)
R - Kelembapan Relatif Udara (%RH)	(t - 3), (t - 2), (t - 1)

Dalam penelitian ini, tujuan *simulated annealing* adalah sebatas mengoptimasi parameter jumlah neuron per layer, batch size dan jumlah epoch. Rentang pencarian nilai parameter tersebut dinyatakan dalam Tabel 5. Rentang ini bersifat simetris dengan parameter awal sebagai titik tengah pencarian. Rentang pencarian juga mempertimbangkan waktu pemrosesan komputasi sesuai kemampuan PC.

Tabel 5. Rentang Pencarian *Hyperparameter Tuning* LSTM.

Ne - Rentang jumlah neuron	[20, 60]
Bs - <i>Batch size</i>	[20, 40]
Ep - Jumlah epoch	[30, 70]

Kriteria *hyperparameter tuning* menggunakan nilai *root mean square error (RMSE)* pada setiap iterasi. Nilai RMSE berupa komparasi antara nilai output RH aktual terhadap output RH prediksi LSTM. Persamaan (2) merupakan parameter RMSE yang secara matematis dinyatakan sebagai berikut [10]:

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (RH_{aktual} - RH_{prediksi})^2} \dots (2)$$

Nilai output RMSE model LSTM setiap iterasi menjadi fungsi objektif dalam metode optimasi *simulated*

annealing. *Hyperparameter tuning* bertujuan menemukan parameter LSTM dengan nilai RMSE paling minimum [10]. Tabel 6 menunjukkan *pseudocode hyperparameter tuning* algoritma LSTM dalam estimasi output sensor RH [9].

Tabel 6. Pseudocode *Hyperparameter Tuning* LSTM.

Algoritma	<i>Hyperparameter Tuning</i>
Input :	D = { T(t-3), T(t-2), T(t-1), R(t-3), R(t-2), R(t-1) }
Output :	s ^{LSTM} = { Ne, Bs, Ep }
1	Inisialisasi s ₀ dan T ₀
2	Ketika terminasi kriteria tidak memenuhi:
3	v = <i>NeighborSelection</i> (s)
4	Atur jumlah neuron per <i>hidden layer</i> sesuai v
5	Atur <i>batch size</i> per <i>hidden layer</i> sesuai v
6	Atur jumlah <i>epoch</i> sesuai v
7	Inisialisasi bobot (w) dan bias (b) awal
8	Ketika terminasi kriteria tidak memenuhi:
9	s ^{LSTM} = <i>Modelling</i> (w, b)
10	e = <i>Evaluation</i> (w, b)
11	Akhiri iterasi
12	f = RMSE (D, s ^{LSTM})
13	Jika f memenuhi kriteria keberterimaan probabilitas:
14	s = v
15	Perbaharui nilai T
16	Akhiri iterasi
17	Output s ^{LSTM}

3. Hasil dan Pembahasan

Hyperparameter awal LSTM sebagai estimator sensor RH memiliki 40 neuron per *hidden layer* dengan *batch size* 30 dan jumlah epoch 50. Ketika LSTM disimulasikan sebagai estimator sensor RH AWS Pemalang, *hyperparameter* ini menghasilkan nilai RMSE 2,23382 %RH. *Hyperparameter tuning* tahap pertama dilakukan terhadap jumlah neuron per *hidden layer* LSTM. Tabel 7 menunjukkan hasil *hyperparameter tuning* terhadap jumlah neuron per

hidden layer menggunakan algoritma *simulated annealing*.

Tabel 7. *Hyperparameter Tuning* Jumlah Neuron LSTM.

Iterasi	Jumlah Neuron	RMSE (%RH)
1	46	2,22092
7	47	2,20162
18	48	2,20022

Jumlah neuron optimal berdasarkan algoritma *simulated annealing* yaitu 48 neuron per *hidden layer*. Nilai RMSE 0,03% lebih rendah dibanding nilai RMSE awal tanpa *hyperparameter tuning*. Selanjutnya, tahap kedua *hyperparameter tuning* dilakukan terhadap *batch size* LSTM menggunakan 48 neuron per *hidden layer*. Tabel 8 menunjukkan hasil *hyperparameter tuning* terhadap *batch size* menggunakan algoritma *simulated annealing*.

Tabel 8. *Hyperparameter Tuning Batch Size* LSTM.

Iterasi	Batch Size	RMSE (%RH)
1	20	2,22847
2	19	2,19511
8	21	2,18452

Batch size optimal berdasarkan algoritma *simulated annealing* yaitu 21. Nilai RMSE 0,015% lebih rendah dibanding nilai RMSE awal tanpa *hyperparameter tuning* terhadap *batch size*. Selanjutnya, tahap ketiga *hyperparameter tuning* dilakukan terhadap jumlah *epoch* LSTM menggunakan 48 neuron per *hidden layer* dan *batch size* 21. Tabel 9 menunjukkan hasil *hyperparameter tuning* terhadap jumlah *epoch* menggunakan algoritma *simulated annealing*.

Tabel 9. *Hyperparameter Tuning* Jumlah Epoch LSTM.

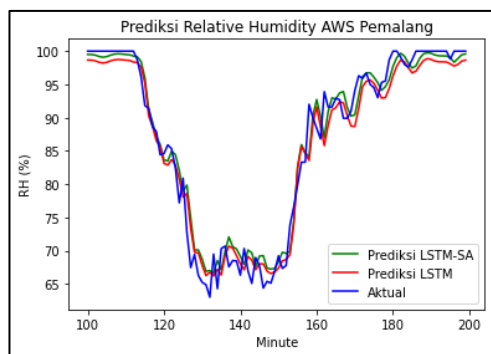
Iterasi	Jumlah Epoch	RMSE (%RH)
1	63	2,23822
8	61	2,18946
16	59	2,18653

Hasil *hyperparameter tuning* jumlah *epoch* LSTM menunjukkan nilai RMSE lebih tinggi dibanding tanpa *hyperparameter tuning*. Adanya *setting* temperatur, *cooling rate* serta *space search range* yang kurang tepat dapat menyebabkan penurunan performa *simulated annealing*. Selain itu, tingkat sensitivitas jumlah *epoch* belum dianalisis terhadap output fungsi objektif. Penambahan jumlah *epoch* kemungkinan tidak berpengaruh signifikan terhadap pemodelan LSTM. Berdasarkan hasil tersebut, jumlah *epoch* 50 tetap dipertahankan dalam model. Tabel 10 merupakan model arsitektur LSTM terbaru berdasarkan hasil *hyperparameter tuning*.

Tabel 10. Model LSTM Hasil *Hyperparameter Tuning*.

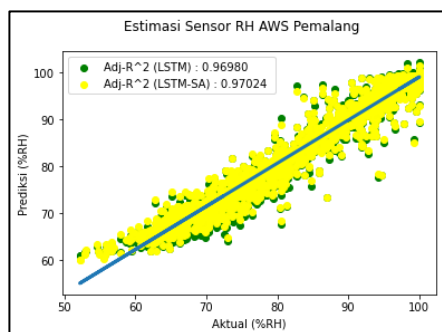
Jumlah <i>hidden layer</i>	2
Jumlah neuron per <i>hidden layer</i>	48
<i>Batch size</i>	21
Jumlah <i>epoch</i>	50
Optimizer	Adam
Fungsi aktivasi	Rectifier Linear Unit (ReLU)

Model LSTM terbaru (LSTM-SA) selanjutnya digunakan sebagai estimator data sensor RH AWS Pemalang. Hasil estimasi sensor RH AWS Pemalang menggunakan model LSTM-SA kemudian dikomparasi terhadap model LSTM tanpa SA.



Gambar 2. Plot Estimasi LSTM vs LSTM-SA.

Gambar 2 menunjukkan hasil plot estimasi RH AWS Pemalang dari kedua algoritma tersebut. Hasil estimasi LSTM-SA berhimpit lebih dekat terhadap aktual dibandingkan hasil estimasi LSTM terhadap aktual. Kriteria performansi *Adjusted-R²* dilakukan lebih lanjut terhadap kedua hasil estimasi tersebut. Kriteria ini bertujuan meninjau korelasi secara numerik antara hasil estimasi terhadap data aktual.



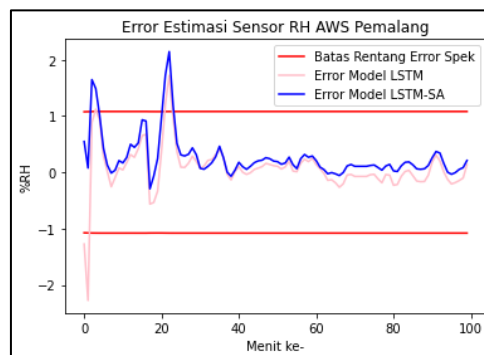
Gambar 3. Plot Sebaran Data Estimasi LSTM vs LSTM-SA.

Gambar 3 menunjukkan plot sebaran data hasil estimasi LSTM dan LSTM-SA terhadap data aktual. Plot ini juga menampilkan nilai *Adjusted-R²* kedua algoritma. Plot sebaran LSTM-SA sedikit lebih rapat ke arah garis interpolasi linier dibanding plot sebaran LSTM. Algoritma LSTM-SA memiliki nilai *Adjusted-R²* lebih tinggi dibanding LSTM. Hal ini berarti terdapat peningkatan nilai korelasi antara data hasil estimasi terhadap

aktual setelah dilakukan *hyperparameter tuning*. Selanjutnya, performa estimasi model divalidasi terhadap performa spesifikasi sensor RH AWS Pemalang. Sensor RH AWS Pemalang adalah Vaisala Hunicap 180R tipe HMP155A. Sensor ini memiliki spesifikasi batas *error* yang dinyatakan pada Persamaan (3) sebagai berikut:

$$\text{error} = 1 + (0.008 \times \text{RH}_{\text{sensor}}) \dots (3)$$

Nilai batas *error* tersebut pada kondisi temperatur $-20^{\circ}\text{C} \sim +40^{\circ}\text{C}$. Batas ini bervariasi sesuai hasil pembacaan sensor terhadap parameter RH. Batas ini menunjukkan spesifikasi sensor berdasarkan pengolahan non-linearitas, histeresis dan *repeatability* sensor. Nilai *error* model diperoleh dari nilai selisih estimasi model terhadap nilai aktual pengukuran sensor RH.



Gambar 4. Plot Error Estimasi Sensor RH AWS Pemalang.

Gambar 4 menunjukkan plot error model LSTM-SA dan LSTM terhadap batas error sesuai spesifikasi sensor. Hasil *plot error* model menunjukkan adanya *spike* yang melampaui batas spesifikasi sensor, baik algoritma LSTM-SA maupun LSTM. Namun demikian, sebagian besar *plot error* LSTM-SA lebih mendekati angka 0 dibanding LSTM. Penyimpangan terhadap batas *error* spesifikasi mengindikasikan perlu dilakukan

hyperparameter tuning lebih lanjut terhadap algoritma LSTM. Selain disebabkan oleh nilai bias algoritma LSTM, performa estimasi sensor juga dipengaruhi oleh *drift* akibat masa penggunaan sensor elektronik di lapangan.

Hasil estimasi perlu dikomparasi lebih lanjut terhadap hasil *quality control data* serta kalibrasi sensor di laboratorium. Terlepas dari semua itu, algoritma LSTM-SA mampu mengoptimasi *hyperparameter* algoritma LSTM dengan menurunnya nilai RMSE serta meningkatnya nilai *Adjusted-R²*. Meskipun terdapat *spike* melampaui batas spesifikasi sensor, hasil *plot error* model LSTM-SA masih memenuhi persyaratan *World Meteorological Organization* (WMO) terkait operasional AWS. Nilai *error* yang dihasilkan masih kurang dari 3 %RH sesuai ketentuan dokumen WMO No.8.

4. Kesimpulan

Dari hasil penelitian ini dapat disimpulkan sebagai berikut :

1. Jumlah neuron optimal berdasarkan algoritma *simulated annealing* yaitu 48 neuron per hidden layer. Nilai RMSE 0,03% lebih rendah dibanding nilai RMSE awal tanpa *hyperparameter tuning*.
2. *Batch size* optimal berdasarkan algoritma *simulated annealing* yaitu 21. Nilai RMSE 0,015% lebih rendah dibanding nilai RMSE awal tanpa *hyperparameter tuning* terhadap *batch size*.
3. Penambahan jumlah *epoch* kemungkinan tidak berpengaruh signifikan terhadap pemodelan LSTM.
4. Algoritma LSTM-SA memiliki nilai *Adjusted-R²* lebih tinggi dibanding LSTM.
5. Meskipun terdapat *spike* melampaui batas spesifikasi sensor, hasil *plot error* model

LSTM-SA masih memenuhi persyaratan *World Meteorological Organization* (WMO) terkait operasional AWS. Nilai *error* yang dihasilkan masih kurang dari 3 %RH sesuai ketentuan dokumen WMO No.8.

Daftar Pustaka

- [1] BMKG (2014) : Peraturan Kepala Badan Meteorologi Klimatologi dan Geofisika Nomor 7 Tahun 2014 tentang Standar Teknis dan Operasional Pemeliharaan Peralatan Pengamatan MKG.
- [2] World Meteorological Organization (2018) : *WMO No.8 Guide to Instruments and Methods of Observation*.
- [3] Dr Bushra Shamshad, M Zubair Khan, Zara Omar, "Modeling and Forecasting Weather Parameters using ANN-MLP, ARIMA and ETS model: A case study for Lahore, Pakistan", *International Journal of Scientific & Engineering Research*, Volume 10, Issue 4, 2019.
- [4] Mohammad Shad, Y. D. Sharma Abhishek Singh, "Forecasting og Monthly Relative Humidity in Delhi, India, using SARIMA and ANN models", *Modeling Earth Systems and Environment*, Springer, 2022.
- [5] Antonios Parasyris, George Alexandrakis, Georgios V. Kozyrakis, Katerina Spanoudaki dan Nikolaos A. Kampanis, "Predicting Meteorological Variables on Local Level with SARIMA, LSTM and Hybrid Techniques", *Atmosphere*, 2022,

<https://doi.org/10.3390/atmos13060878>.

- [6] Chun-Wei Tsai, Chien-Hui Hsia, Shuang-Jie Yang, Shih-Jui Liu, Zhi-Yan Fang, “*Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing*”, Applied Soft Computing Journal 88, 2020.
- [7] Sergio Ledesma, Miguel Torres, Donato Hernández, Gabriel Aviña, and Guadalupe García, “*Temperature cycling on simulated annealing for neural network learning*”, Advances in Artificial Intelligence, hal. 161–171, 2007.
- [8] Jussi Kalliola, Jurgita Kapočiūtė-Dzikienė, dan Robertas Damaševičius, “*Neural network hyperparameter optimization for prediction of real estate prices in Helsinki*”, PeerJ Comput. Sci. 7, 2021.
- [9] Nikolaos Gorgolis, Ioannis Hatzilygeroudis, Zoltan Istenes, Lazlo – Grad Gyenne, “*Hyperparameter Optimization of LSTM Network Models through Genetic Algorithm*”, Conference Paper, 2019.
- [10] Bahareh Nakisaa, Mohammad Naim Rastgoo, Andry Rakotonirainy, Frederic Mairea, Vinod Chandran, “*Long Short Term Memory Hyperparameter Optimization for a Neural Network Based Emotion Recognition Framework*”, IEEE Access, 2022.