Journal of Deep Learning, Computer Vision and Digital Image Processing
E-ISSN: 2986-8939 P-ISSN: 2986-8920
Vol. 4, No. 2, June 2026
DOI. https://doi.org/10.61255/decoding.v4i2.1715

Multivariate LSTM versus Simple Baselines for FFB Yield Forecasting at Marihat Plantation: A Feasibility Pilot

Rendi Zattul Tajemir
(Institut Teknologi Sawit Indonesia, Medan, Indonesia)

Raden Aris Sugianto✉
(Institut Teknologi Sawit Indonesia, Medan, Indonesia)

Sri Lestari Rahayu
(Institut Teknologi Sawit Indonesia, Medan, Indonesia)

OPEN ACCESS

ARTICLE HISTORY
Received: April 12, 2026
Revised: May 26, 2026
Accepted: June 30, 2026

KEYWORDS
Deep learning;
Fresh Fruit Bunches (FFB);
Long Short-Term Memory (LSTM);
Production prediction;
Time series.

ABSTRACT

Purpose – This study re-evaluates a multivariate LSTM for forecasting oil palm fresh fruit bunch (FFB) yield at Marihat Plantation after an audit identified a data-export bug in the original dataset. The data were rebuilt and assessed using a leakage-safe chronological design, naive and moving-average baselines, and seed-sensitivity checks.

Methods – Monthly FFB production, rainfall, and rainy-day records from plantation reports (2020–2023) were reconstructed and aggregated by planting-year cohort because block-level data were unavailable. Twelve-month windows across 11 cohorts yielded 240 sequences. With an 11-month purge gap, the chronological split comprised 10 training, 20 validation, and 10 test sequences. A two-layer LSTM (64 and 32 units; 20% dropout) was compared with five naive and moving-average baselines and retrained using 10 random seeds.

Findings – The LSTM achieved MAPE = 29.73% (MAE = 0.27; RMSE = 0.34 ton/ha) on the test set, outperforming all baselines (MAPE = 74–118%). Across 10 seeds, mean MAPE was 31.79% (SD = 4.59%; worst = 40.60%), remaining below every baseline. However, all test sequences represented December 2023 across 10 cohorts. The observed advantage therefore indicates feasibility, not forecasting superiority across time.

Research implications – The LSTM produced consistently lower errors than the baselines across cohorts for one target month, but the 10-sequence, single-month test set does not support general temporal claims. The prototype dashboard may support scenario exploration, although its 12-month forecasts depend on deterministic, unvalidated synthesized inputs.

Originality/value – Prompted by a data-quality audit documented in Supplementary S1, this study transparently rebuilds the evidence base and reframes the model as a feasibility pilot rather than a validated solution. Longer, gap-free records covering multiple test months and a validated multi-step procedure are required to establish a dependable advantage.

Correspondence Author: ✉ radenaris@itsi.ac.id
To cite this article : Tajemir, R. Z., Sugianto, R. A., & Rahayu, S. L. (2026). Multivariate LSTM versus simple baselines for FFB yield forecasting at Marihat Plantation: A feasibility pilot. Journal of Deep Learning, Computer Vision and Digital Image Processing, 4(2), 337–355. https://doi.org/10.61255/decoding.v4i2.1715

INTRODUCTION

Oil palm (Elaeis guineensis) is a strategic plantation commodity that contributes to national foreign exchange earnings and public welfare, particularly in major plantation regions such as North Sumatra [1]. The primary oil palm products, Crude Palm Oil (CPO) and Palm Kernel (PK), are derived from processing Fresh Fruit Bunches (FFB); consequently, the accuracy of FFB production estimates is a crucial factor in supply chain efficiency. FFB production tends to fluctuate due to climatic and agronomic factors such as rainfall, air temperature, number of rainy days, and crop growth conditions [2], resulting in periodically recorded production data that form time series characterized by trend, seasonal, and non-linear patterns, which require specialized analytical methods. To date, FFB production estimates at the Marihat Plantation have relied on the Company's Work Plan and Budget (RKAP), established as a static figure at the start of the year without accounting for changing actual field conditions, frequently leading to discrepancies between targets and realized output.

Setiawan and Zulkarnain [3] compared the LSTM and SARIMA methods on palm oil production data using Time Series Cross Validation and showed that the accuracy of both methods depends on the characteristics of the data and the model configuration. Deep learning models, especially LSTM, are reported to have good capabilities in capturing temporal patterns and nonlinear relationships compared to conventional methods, especially on seasonal and fluctuating data [4], [5]. LSTM is a development of the Recurrent Neural Network (RNN) that overcomes the vanishing-gradient problem through a cell-state mechanism together with forget, input, and output gates [6], [7], [8], allowing it to retain relevant information over longer sequences. Previous research in the palm oil sector has consistently reported low error rates for LSTM-based yield prediction: Ramadhan and Fachrie [9] and Putra and Harahap [10] reported that LSTM yields crop-productivity predictions with relatively low error rates, while Husaini et al. [11] and Syarovy et al. [12] also demonstrated strong LSTM performance in predicting oil palm production from historical data. Previous LSTM and hybrid applications in palm oil, FFB, and CPO forecasting vary in their input variables, model architectures, evaluation methods, and practical integration, including an LSTM–XGBoost comparison for CPO forecasting[13]. Overall, the literature identifies LSTM as a promising forecasting approach but also highlights the need for rigorous chronological evaluation and baseline comparison before its operational reliability can be established.

Existing studies report different and often non-interchangeable error metrics, so direct numerical comparisons require caution. Setiawan and Zulkarnain [3] reported an LSTM accuracy of 85.91% on univariate monthly CPO data; converting this figure to an equivalent MAPE would require the source to explicitly define accuracy as (100 − MAPE), which could not be confirmed, so no such conversion is made here. Ramadhan and Fachrie [9] and Putra and Harahap [10] reported MSE and normalized RMSE respectively rather than MAPE, which similarly precludes a direct percentage-based comparison. Husaini et al. [11] reported a considerably higher MAPE of 50.87% for a single-company production series. Against this backdrop, the present study's MAPE of 29.73% is lower than Husaini et al.'s figure, but unlike any of the five prior studies it is also the first among them to be benchmarked against simple naive and moving-average baselines, a comparison that in this case favors the LSTM, though on a single-month, 10-sequence pilot test set that limits how far this comparison can be generalized.

Literature reviews indicate that the application of machine learning in the palm oil industry has largely focused on classification tasks such as assessing fruit ripeness and detecting crop diseases, whereas predictive analysis based on historical production data remains comparatively limited [14]. Furthermore, the specific application of LSTM models to Fresh Fruit Bunch (FFB) production data at the Marihat Plantation, incorporating rainfall, rainy days, and crop age in a multivariate analysis while also accounting for climate lag effects, has rarely been undertaken; this study aims to help address this gap from both academic and practical perspectives. Accurately forecasting FFB production could help plantation management reduce the risk of discrepancies between production targets and actual output, and support more adaptive resource planning than static annual RKAP figures. This motivates a data-driven, historically informed prediction system that is accessible to

management through a web-based dashboard, so that prediction results can serve as a reusable planning aid rather than a manual, one-off calculation.

This study aims to (1) apply the LSTM method to predict FFB yield at the Marihat Plantation under a leakage-safe, chronologically evaluated design, using a dataset rebuilt directly from the plantation's original production reports after an export bug was found in an earlier dataset version; (2) evaluate the model's accuracy using MAPE, MAE, and RMSE and benchmark it against naive and moving-average baselines; and (3) develop a prototype prediction dashboard to illustrate how such a model could support production-planning discussions. Because genuinely gap-free monthly records only support modeling at the planting-year-cohort level over a 48-month window (2020–2023), yielding just 10 training and 10 test sequences, this study is best characterized as a data-quality-driven feasibility pilot rather than a definitive validation of LSTM's operational superiority over simpler methods.

METHOD

Research Design

This study employs a quantitative experimental approach using historical numerical Fresh Fruit Bunch (FFB) production data analyzed with a Long Short-Term Memory (LSTM) model**.** The experimental stages comprise problem identification, data collection, data preprocessing, LSTM model development, model evaluation (including comparison against naive and moving-average baselines), and web-based implementation, as illustrated in Figure 1.

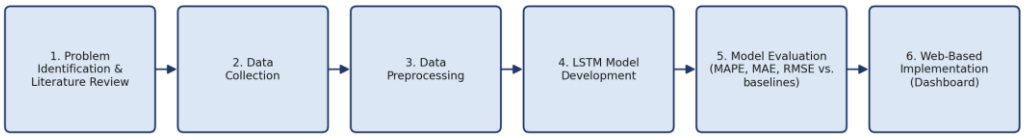

Figure 1. Research stages of the proposed system

Population and Sampling

The study initially used historical monthly FFB production data recorded under 503 unique Entity_ID codes at the Marihat Plantation, each intended to represent a specific Afdeling–Block combination, covering January 2020 to October 2025, compiled in Master_LSTM_Final_Large.csv. However, auditing this file revealed that after removing 831 exact-duplicate rows, most remaining Entity_ID–month combinations still contained multiple conflicting rows with different Production_Kg values but identical Area_Ha, Palm_Age, Rainfall_CH, and RainyDays_HH an unresolved data-export artifact, not genuine repeated measurements (a detailed audit of this pattern, including the affected Entity_ID–month combinations and a worked example, is provided in Supplementary S1). Because the original sequence-building step counted rows rather than calendar months, this artifact allowed sequences labeled as 12-month windows to silently compress as few as 7 real calendar months. The dataset was therefore rebuilt from the plantation's original Excel production reports (the "Kg TBS" monthly table in each year's MAT-LM76 report, cross-checked against a 2010–2025 rainfall record), aggregated per planting-year (Tahun Tanam) cohort across the whole plantation, since block-level monthly breakdowns were not available in any source document. Genuinely gap-free monthly production could only be confirmed for 2020–2023; 2024 records could not be located in any available source and were excluded, along with partial 2025 records for consistency.

A 12-month sliding window was then applied within each planting-year cohort to construct multivariate sequences, each containing 12 input features (Table 1; a 24-month lag feature pair was dropped because the available 48-month window could not support it) with Yield (Ton/Ha) as the prediction target. This window strictly requires 13 calendar-consecutive months (12 input months

plus 1 target month) per cohort; only 11 of the cohorts with recoverable monthly history met this requirement, and only 9 of those spanned the full 2020–2023 window. This produced 240 candidate sequences in total. Sequence generation was performed independently within each cohort and explicitly checked for calendar consecutiveness (not merely row count), so that no sequence silently spans a data gap.

Because adjacent sequences generated by a 12-month sliding window with stride 1 share up to 11 of their 12 input months, a random split of these sequences (as used in an earlier version of this analysis) allows near-duplicate, heavily overlapping windows to fall on both sides of the train/test boundary a form of temporal leakage that inflates apparent accuracy [15]. To avoid this, sequences were split chronologically by target month, and an 11-month purge gap was enforced at each partition boundary the minimum gap that guarantees zero window overlap for a 12-month sliding window so that no training-set window and no validation- or test-set window share a single calendar month. This purge-gapped design unavoidably discards sequences that fall inside the gap: of the 240 candidate sequences, 10 (4.2%) were retained for training, 20 (8.3%) for validation, and 10 (4.2%) for testing, while the remaining 200 sequences (83.3%) fell within a purge gap and were excluded from all three partitions (Table 1). This is reported here transparently as the central limitation of this study rather than a minor implementation detail: with only 48 genuinely gap-free calendar months available, a strict, leakage-safe chronological design leaves very little data for training and testing. Notably, because the available calendar range is so short, all 10 test sequences share the same target month (December 2023) across 10 different cohorts the split is chronologically valid, but the test set is a single-month cross-section rather than a temporal sequence of test months.

Table 1. Chronological, Purge-Gapped Train/Validation/Test Split of the 240 Candidate Sequences

Partition	Sequences	% of 240 candidates	Approx. target-month range	Purpose
Training	10	4.2%	Jan 2022 only	Model fitting; scaler fitting
Purge gap 1	(discarded)	—	≈ Feb 2022 – Nov 2022	12-month buffer, no overlap with training
Validation	20	8.3%	Dec 2022 – Jan 2023	Early stopping / LR scheduling
Purge gap 2	(discarded)	—	≈ Feb 2023 – Nov 2023	12-month buffer, no overlap with validation
Testing	10	4.2%	Dec 2023 only	Held-out performance evaluation

Instrument

The research instrument was a Long Short-Term Memory (LSTM) model implemented using the TensorFlow/Keras Functional API [16]. The proposed architecture comprised two LSTM layers with 64 and 32 hidden units respectively, each followed by a 20% Dropout layer, and a Dense output layer (Figure 2). In the standard LSTM architecture, the cell state employs the hyperbolic tangent (tanh) activation function, whereas the forget, input, and output gates use the sigmoid activation function to regulate information flow through the network [8], [17].

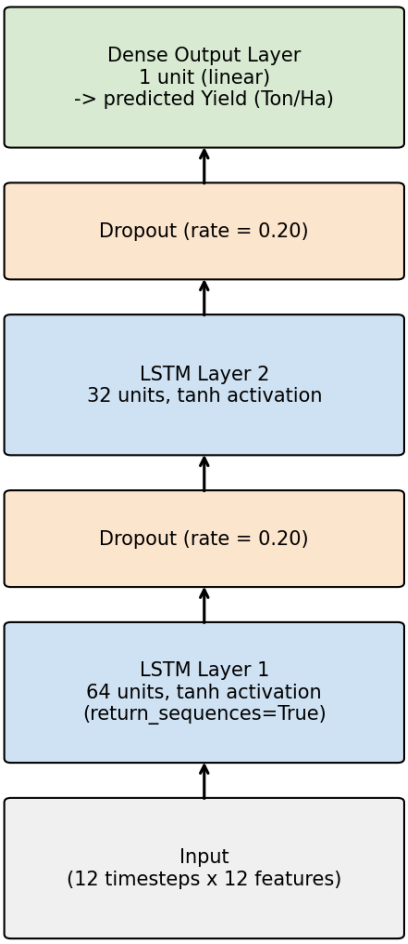

Figure 2. Implemented two-layer LSTM architecture (64 and 32 units) with 20% Dropout after each layer and a Dense output layer predicting FFB yield (Ton/Ha).

The model was optimized using the Adam optimizer (learning rate 0.001) with the Mean Squared Error (MSE) loss function, a batch size of 8, and a maximum of 300 training epochs. ReduceLROnPlateau and EarlyStopping callbacks were used to improve convergence and reduce overfitting [5]. The number of LSTM units, dropout rate, learning rate, and batch size were manually predetermined based on common practice reported in prior LSTM-based time-series forecasting work [5], [17], [18], rather than derived from a systematic hyperparameter search: no grid or random search, validation-based tuning criterion, or sensitivity analysis was performed, in contrast to approaches such as [19]. These settings should therefore be read as a reasonable, literature-informed starting configuration rather than an optimized one. The complete model contained 32,161 trainable parameters. All experiments used Python 3.11.9, TensorFlow/Keras 2.21.0, pandas 3.0.5, and NumPy 2.4.6; the random seed for each run is reported alongside its result throughout this paper (see the Sensitivity to Random Initialization and Ablation subsections below). Table 2 summarizes the 12 input features used in the model.

Table 2. Description of the 12 Input Features Used to Construct Each 12-Month Sequence

Feature	Description	Type / Source
Production_Kg	Total monthly FFB production for the cohort (Kg)	Raw agronomic record
Area_Ha	Planted area of the cohort (Ha)	Raw agronomic record

Feature	Description	Type / Source
Palm_Age	Economic age of the oil palm stand (years)	Raw agronomic record
Rainfall_CH	Mean monthly rainfall (mm)	Raw climatic record
RainyDays_HH	Number of rainy days in the month	Raw climatic record
CH_Lag_6	Rainfall, 6 months prior	Engineered lag feature
HH_Lag_6	Rainy days, 6 months prior	Engineered lag feature
CH_Lag_12	Rainfall, 12 months prior	Engineered lag feature
HH_Lag_12	Rainy days, 12 months prior	Engineered lag feature
Month_Sin	Sine cyclical encoding of the calendar month	Engineered seasonal feature
Month_Cos	Cosine cyclical encoding of the calendar month	Engineered seasonal feature
Yield_Ton_Ha	Historical yield (Ton/Ha); also the prediction target — included within each 12-month input window as an autoregressive feature	Raw record / target

Procedures

The research was conducted from November 2025 to August 2026 and proceeded through problem identification and literature review, data collection, data preprocessing, model development, model evaluation (including baseline comparison), and system implementation. Data preprocessing included duplicate removal, MinMaxScaler normalization (fit on the training partition only), generation of rainfall and rainy-day lag features at 6- and 12-month horizons (a 24-month horizon was dropped because the available 48-month window could not support it), and cyclical seasonal encoding (Month_Sin, Month_Cos) [20]. All random seeds were fixed (seed = 42), and the software environment used TensorFlow/Keras 3 with scikit-learn for preprocessing. The trained LSTM model was deployed through a Streamlit-based web dashboard integrated with a MySQL database (prediksi_sawit) to store prediction history and support interactive single-month and 12-month forecasting for plantation management.

Analysis plan

Model performance was evaluated using the Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). MAPE expresses prediction error as a percentage and is widely used in time-series forecasting for its straightforward interpretation [21], [22]; following commonly cited forecasting-performance thresholds, MAPE below 10% is considered highly accurate, 10–20% good, 20–50% reasonable, and above 50% poor [23]. Because MAPE alone can be unstable near zero-valued targets and does not by itself indicate whether a model outperforms trivial forecasting rules, three additional baselines were computed on the same chronological test

partition: (1) a 1-month persistence forecast (predicting the most recently observed month's yield), (2) moving-average forecasts using the trailing 3-, 6-, and 12-month means of the input window, and (3) a 12-month seasonal-naive forecast (predicting the value observed exactly 12 months earlier). No zero-valued yield observations occurred in the test partition, so the MAPE denominator was well-defined throughout.

RESULTS AND DISCUSSION

Results

The proposed LSTM model was implemented to forecast FFB production at Marihat Plantation using a multivariate time-series approach, evaluated under the leakage-safe, chronologically purge-gapped design described above.

Dataset Preparation

The initial dataset comprised 2,781 monthly observations purportedly across 503 Entity_ID (Afdeling–Block) codes; after removing 831 exact-duplicate records, an audit found that most remaining Entity_ID–month combinations still contained unresolved duplicate rows with conflicting production values (392 Entity_ID–month combinations, 965 rows; see Supplementary S1 for the full audit and a worked example). The dataset was therefore rebuilt directly from the plantation's original Excel production reports at the planting-year-cohort level, yielding 686 clean monthly observations across all 20 reconstructed cohorts, each internally gap-free but varying in length (Table 4). Lag features for rainfall and rainy days (6 and 12 months; a 24-month lag was dropped as unsupportable by this shorter window) and cyclical seasonal features were then generated. Using a 12-month sliding window with an explicit calendar-consecutiveness check, the rebuilt dataset yielded 240 candidate sequences from the 11 cohorts with sufficiently long records; after the chronological, 11-month-purge-gapped split, 10 sequences were used for training, 20 for validation, and 10 for testing (Table 3).

The reconstructed production records covered 20 distinct planting-year cohorts in total; 11 were retained for modeling and 9 were excluded. Table 4 summarizes the exclusion criteria and reasons. Excluding a cohort was a purely mechanical consequence of sequence construction requiring both a full 12-month lag history before the input window and a valid target month after it not a judgment call about data quality; no cohort with sufficient calendar coverage for at least one valid sequence was dropped.

Table 3. Planting-Year Cohort Inclusion and Exclusion Summary (20 Cohorts in the Reconstructed Records)

Planting year(s)	Consecutive months available	Status	Reason
2000, 2001, 2004, 2005, 2007, 2009, 2010, 2011, 2016 (9 cohorts)	48 (2020–2023)	Included	—
1999, 2017 (2 cohorts)	36	Included	—
1991, 1992, 1993, 1994, 1995, 1996, 1998 (7 cohorts)	24 (Jan 2020–Dec 2021 only)	Excluded	No sequence position has both a valid 12-month lag history and a valid target month simultaneously — the available window is exactly 1 month short of what both requirements need together, even though 24 months exceeds the 13-month bare minimum. Consistent with these blocks having been replanted around 2022.
2019 (1 cohort)	12 (2023 only)	Excluded	Below the 13-month minimum needed for any valid sequence.
2020 (1 cohort)	2 (2023 only)	Excluded	Far too short; likely a newly-recorded planting not yet established in the monthly reporting system.

Notably, the seven cohorts with 24 consecutive months (planting years 1991–1996 and 1998) were excluded despite comfortably exceeding the 13-month bare minimum for a single sequence: their records ran exactly from January 2020 through December 2021 with nothing before or after, so no 12-month input window in that range simultaneously has a full 12-month lag history available before it and an observed target month available after it the available span is exactly one month short of what both requirements need at once. This pattern is consistent with these blocks having been replanted around 2022, after which their planting-year identity changed and their pre-2022 records were not continued under the same cohort. The included and excluded cohorts also differ systematically in composition, not only in data availability. Table 4 compares mean planted area, palm age, and yield across the three groups. The 11 modeled cohorts average 14.4 years of age and 2.13 Ton/Ha mean yield, while the 7 cohorts excluded for insufficient lag/target margin average 26.4 years and only 0.92 Ton/Ha less than half the mean yield of the modeled cohorts consistent with the declining productivity typically seen in aging oil palm stands and with the hypothesis that these blocks were replanted around 2022.

Table 4. Mean Area, Palm Age, and Yield by Cohort Inclusion Group

Group	Mean area (Ha)	Mean palm age (years)	Mean Yield (Ton/Ha)
Included (11 cohorts)	211.0	14.4	2.13
Excluded, insufficient lag/target margin (7 cohorts)	140.0	26.4	0.92
Excluded, too short (2 cohorts)	146.6	3.9	1.10

This is an important limitation for generalizability, not merely a data-availability footnote: because only genuinely gap-free 2020–2023 records could support valid sequence construction, and those records happen to belong to younger, higher-yielding cohorts, the model was necessarily trained and evaluated on a systematically younger, higher-yielding subset of the plantation. Its performance on older, lower-yielding blocks which include exactly the kind of aging stands likely to be prioritized for replanting decisions remains untested and should not be assumed to match the results reported here.

Model Training Performance

The LSTM model was trained while monitoring MSE loss on the training and validation partitions (Figure 3). Training loss decreased steadily and converged to a low, stable value within a few epochs unsurprising given only 10 training sequences. Validation loss (n = 20) remained substantially higher and noisier throughout, moving between roughly 0.6 and 0.9 without a clear stable trend, and EarlyStopping restored the model weights from epoch 3, the point of lowest observed validation loss. This pattern is more consistent with a model fitting a very small training partition than with a well-generalized fit; the small partition sizes make it difficult to distinguish genuine learning from partition-specific noise, and this training curve should be read with that caveat.

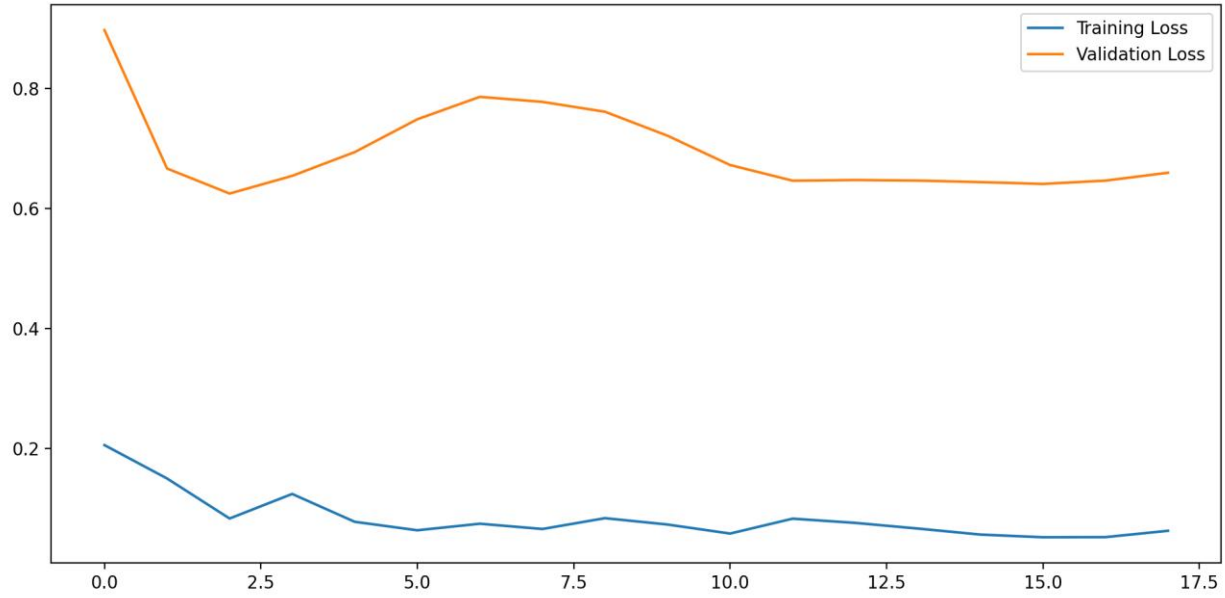

Figure 3. Training and validation loss (MSE) across epochs. X-axis: training epoch (EarlyStopping triggered at epoch 18; best weights restored from epoch 3); Y-axis: MSE loss on min-max-scaled yield. Validation loss remains substantially higher than training loss throughout, consistent with a 10-sequence training set.

Model Evaluation

The trained model was evaluated using 10 chronologically held-out test sequences representing different planting-year cohorts. All sequences shared December 2023 as their target month, allowing performance to be examined across cohorts under the same forecasting period. MAPE was used to assess relative error, while MAE and RMSE quantified the magnitude of prediction errors in ton/ha. Therefore, the results primarily indicate cross-cohort performance for a single target month rather than temporal generalization across multiple periods. Table 5 summarizes the three evaluation metrics and their interpretation.

Table 5. LSTM Model Evaluation Results on the Chronological, Purge-Gapped Test Set (n = 10)

Evaluation Metric	Value	Notes
Mean Absolute Percentage Error (MAPE)	29.73%	"Reasonable" category per the adopted thresholds [23], though from an extremely small, single-month test set
Mean Absolute Error (MAE)	0.27 Ton/Ha	Test-set mean yield = 1.27 Ton/Ha, range 0.55–1.87 Ton/Ha
Root Mean Squared Error (RMSE)	0.34 Ton/Ha	—

The MAPE of 29.73% indicates that predictions differed from actual yields by approximately 29.73% on average. This falls within the "reasonable" category under the adopted thresholds [21]. The MAE of 0.27 ton/ha represents the average absolute prediction error, while the higher RMSE of 0.34 ton/ha suggests that several sequences produced comparatively larger errors. Because actual yields ranged from 0.55 to 1.87 ton/ha, MAPE may be particularly sensitive to errors in cohorts with lower yields. Moreover, with only 10 sequences targeting the same month, one or two large deviations could substantially influence all three metrics. Therefore, the threshold-based classification alone does not establish satisfactory forecasting performance; comparison with simpler methods is necessary. Table 6 and Figure 4 present this baseline comparison.

Table 6. LSTM vs. Naive and Moving-Average Baselines on the Same Chronological Test Set (n = 10)

Method	MAPE	MAE (Ton/Ha)	RMSE (Ton/Ha)
Seasonal-naive (value 12 months prior)	118.21%	1.26	1.33
Moving average, 12-month window	74.31%	0.81	0.84
Proposed LSTM	29.73%	0.27	0.34
Moving average, 6-month window	80.52%	0.92	0.94
Moving average, 3-month window	77.28%	0.93	0.96
Persistence (previous month's value, no model)	92.23%	1.14	1.18

The LSTM outperforms all five naive and moving-average baselines evaluated on this test set, including the 1-month persistence forecast that requires no model at all a reversal of what an earlier, data-error-affected version of this analysis had reported. Inspection of the test-partition target series (Figure 4) helps explain why: because the clean, gap-free calendar range available after rebuilding the dataset spans only 48 months (2020–2023), every one of the 10 test sequences targets the same calendar month, December 2023, across 10 different planting-year cohorts. December 2023 yields differ substantially from each cohort's own recent months, so naive baselines that extrapolate from recent history perform poorly by construction, while the LSTM having learned cross-cohort and seasonal patterns during training generalizes better to this single, atypical month. This is an encouraging result, but it should not be read as a general claim that the LSTM outperforms simple forecasting: the comparison covers one calendar month and 10 sequences, which is too small a sample to establish forecasting skill across time. It represents a genuine, data-constrained finding to be reported transparently not evidence of validated superiority and further testing across additional months, once more clean historical records become available, is needed before a general claim can be made.

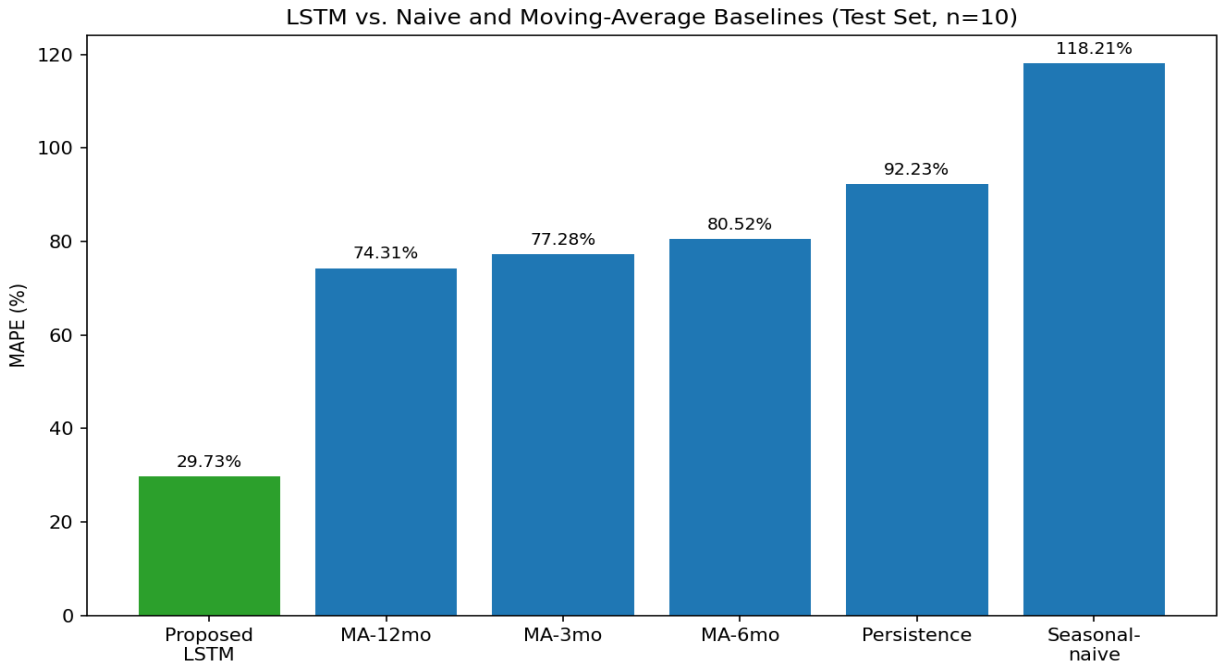

Figure 4. MAPE of the proposed LSTM against naive and moving-average baselines on the chronological test set.

Sensitivity to Random Initialization

Because the training partition is extremely small (n = 10), the model's reported performance could plausibly reflect a favorable random weight initialization rather than a stable, reproducible result. To check this, the model was retrained from scratch 10 times under the identical chronological, purge-gapped split, varying only the random seed governing weight initialization and batch ordering. MAPE across the 10 runs averaged 31.79% (SD = 4.59%; range 24.59–40.60%), MAE averaged 0.29 Ton/Ha (SD = 0.03), and RMSE averaged 0.37 Ton/Ha (SD = 0.04). The single run reported above (MAPE = 29.73%) falls within this range and is not an outlier. Critically, even the worst-performing seed (MAPE = 40.60%) remained far below the best-performing baseline (12-month moving average, MAPE = 74.31%), so the finding that the LSTM outperforms all five naive and moving-average baselines on this test set holds across every seed tested, not just the specific initialization reported in Table 6. This does not remove the sample-size limitation all 10 seeds share the same 10-sequence test partition and single target month but it does rule out favorable initialization as an alternative explanation for the model's advantage over the baselines.

Prediction Performance

The distribution of absolute prediction errors is summarized in Table 8. A total of 9 of 10 predictions (90.0%) had an absolute error below 0.5 Ton/Ha, and no test-set prediction exceeded 1.0 Ton/Ha of absolute error, indicating that the LSTM's errors were consistently bounded on this single-month test set consistent with it also outperforming every baseline evaluated (Table 7).

Table7. Distribution of Absolute Prediction Errors on the Chronological Test Set (n = 10)

Absolute Error Range (Ton/Ha)	Number of Predictions	Proportion (%)
0.0–0.3	6	60.00
0.3–0.5	3	30.00
0.5–1.0	1	10.00
> 1.0	0	0.00
Total	10	100.00

Figure 5 shows that the LSTM's predictions cluster in a comparatively narrow band (roughly 1.2–1.4 Ton/Ha) while the actual December 2023 yields span a much wider range across cohorts (0.55–1.87 Ton/Ha). This same under-reactivity that limited the model in an earlier, block-level version of this analysis is still visible here, but on this test set it works in the LSTM's favor: because naive baselines simply copy or average each cohort's own recent, differently-scaled history, they are pulled far from the shared, atypical December 2023 level, whereas the LSTM's smoothed, centrally-anchored predictions happen to sit closer to most cohorts' actual December values. This is a favorable coincidence of this particular month's data, not evidence that under-reactivity is generally desirable.

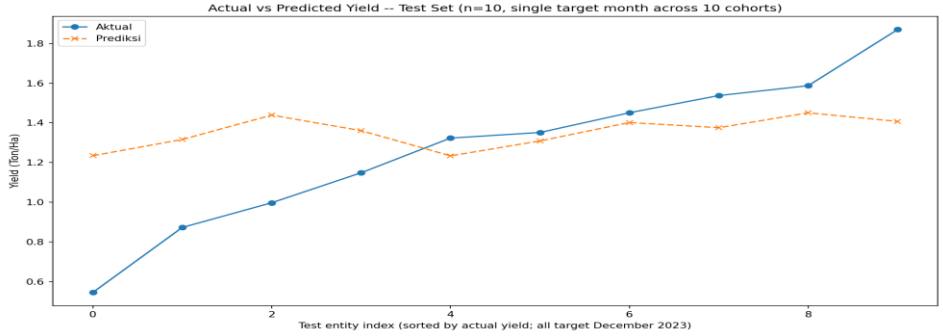

Figure 5. Actual versus LSTM-predicted FFB yield on the chronological test set (n = 10), sorted by actual yield; all 10 sequences share December 2023 as their target month, across 10 different planting-year cohorts.

Statistical Significance and Per-Cohort Error Breakdown

Because the test partition already contains exactly one sequence per cohort (Table 1), the per-sequence errors reported here are simultaneously per-cohort errors no cohort contributes more than one test point, and none can dominate the aggregate MAPE. Table 8 reports the absolute percentage error (APE) for the LSTM and the two strongest baselines (persistence, 12-month moving average) for each of the 10 cohorts.

Table 8. Per-Cohort Prediction Errors on the Chronological Test Set (n = 10)

Cohort	Actual (Ton/Ha)	LSTM Pred.	LSTM APE (%)	Persistence APE (%)	12-mo MA APE (%)
0	0.55	1.23	126.44	137.13	193.32
1	0.87	1.32	50.85	59.60	86.74
2	1.00	1.44	44.30	100.94	84.88
3	1.45	1.40	3.40	96.63	80.02
4	1.59	1.45	8.61	106.55	69.01
5	1.32	1.23	6.77	84.72	45.21
6	1.87	1.41	24.76	77.01	42.16
7	1.35	1.31	3.15	80.51	53.08
8	1.15	1.36	18.50	105.38	62.29
9	1.54	1.38	10.53	73.79	26.41

Cohort 0, with the lowest actual yield in the test set (0.55 Ton/Ha), is the hardest case for every method (LSTM APE = 126.44%), but the LSTM's error there is still smaller than persistence (137.13%) and substantially smaller than the 12-month moving average (193.32%). For the remaining nine cohorts the LSTM's APE is below 51% and often in the single digits, while both baselines remain above 59% throughout. To formally test whether the LSTM's error differs from each baseline, paired bootstrap resampling (10,000 resamples over the 10 paired APE differences) was applied to the four baselines spanning the widest range of the comparison persistence, the 3- and 12-month moving averages, and the seasonal-naive forecast; the 6-month moving average (MAPE = 80.52%, between the 3- and 12-month results) was included in the descriptive comparison in Table 7 but not separately bootstrap-tested.

The LSTM's mean APE was significantly lower than each of the four baselines tested: persistence (mean difference = −62.49 percentage points, 95% CI [−79.97, −42.62]), 3-month moving average (−47.55, CI [−61.00, −32.53]), 12-month moving average (−44.58, CI [−56.12, −33.48]), and seasonal-naive (−88.48, CI [−129.51, −58.88]); none of these four 95% confidence intervals include zero, so the LSTM's advantage over each tested baseline on this test set is unlikely to be due to chance alone. This complements, rather than replaces, the caveats already noted in Limitations: the test set remains small (n = 10) and confined to one calendar month, so this result establishes internal statistical robustness on the available data, not generalization to other months.

Ablation: Value of Autoregressive versus Climate-Only Features

Production_Kg, Area_Ha, and historical Yield_Ton_Ha are all present in the 12-month input window alongside climate and time features, and are mathematically related to one another and to the target (see Instrument section); to test whether they carry genuine information beyond simple autoregression on the target, or are largely redundant, three feature scenarios were compared under the identical chronological split and 10-seed retraining protocol used in the sensitivity analysis above: (A) the full 12-feature set used throughout this study; (B) the same set with Production_Kg and Area_Ha removed, retaining historical Yield_Ton_Ha alongside all climate/time features; and (C) climate-and-time features only, with Production_Kg, Area_Ha, and historical Yield_Ton_Ha all removed.

Table 9. Ablation of Production, Area, and Historical Yield Features (10-Seed Mean MAPE)

Feature scenario	Mean MAPE (%)	SD (%)	Δ vs. full set (pp)
A: Full feature set (12 features)	31.79	4.59	—
B: Without Production_Kg, Area_Ha (10 features)	35.04	5.75	+3.25
C: Climate/time only, no autoregressive features (9 features)	36.47	7.79	+4.68 vs. A

Mean MAPE degraded monotonically as these features were removed, and its variability across seeds increased in step (Table 9). Removing Production_Kg and Area_Ha alone cost 3.25 percentage points on average; removing historical Yield_Ton_Ha as well cost a further 1.43 points. This indicates that Production_Kg, Area_Ha, and historical Yield_Ton_Ha are not simply redundant restatements of one another or of the target each appears to contribute distinct, useful information to the model. At the same time, the climate-and-time-only model (scenario C) still reached a mean MAPE of 36.47%, comfortably below every naive baseline in Table 7 (74–118%), indicating that climate and time features also carry standalone predictive signal rather than functioning only as noise alongside the autoregressive features. As with the sensitivity analysis above, these comparisons are based on the same 10-sequence test partition and should be read as a preliminary robustness check, not a definitive feature-importance ranking.

Architecture Size Sensitivity

With only 10 training sequences fitting a two-layer LSTM of 32,161 trainable parameters, it is unclear whether this level of model complexity is justified rather than simply tolerated. To check this, three smaller architectures were compared against the original two-layer 64+32-unit design under the identical data, split, and 10-seed retraining protocol used throughout: a half-sized two-layer model (32+16 units), a quarter-sized two-layer model (16+8 units), and a minimal single-layer model (8 units, no second LSTM layer).

Table 10. Mean MAPE Across 10 Seeds for Four LSTM Architecture Sizes

Architecture	Trainable params	Mean MAPE (%)	SD (%)
XL (original): 2 layers, 64+32 units	32,161	31.79	4.59
L: 2 layers, 32+16 units	8,913	30.57	2.62
M: 2 layers, 16+8 units	2,665	30.42	1.92
S: 1 layer, 8 units	681	33.58	5.43

Table 10 shows that the original 32,161-parameter architecture was not the best-performing option: both the half-sized (8,913 parameters) and quarter-sized (2,665 parameters) models achieved slightly lower mean MAPE (30.57% and 30.42% respectively, versus 31.79% for the original) with substantially lower run-to-run variability (SD 2.62% and 1.92%, versus 4.59%). Only the most aggressively reduced single-layer model (681 parameters) performed worse on average (33.58%) and with the highest variability (SD 5.43%), suggesting that some representational capacity beyond a single small layer is still useful, but that the specific 64+32-unit configuration used throughout this study is not clearly justified by this 10-sequence training set: a model with roughly one-twelfth as many parameters (the quarter-sized configuration) matched or slightly exceeded it in accuracy while being noticeably more stable across random initializations. Given the small and clustered test set, this should be read as evidence that the reported results are not an artifact of excess model capacity, rather than as a definitive optimal architecture recommendation.

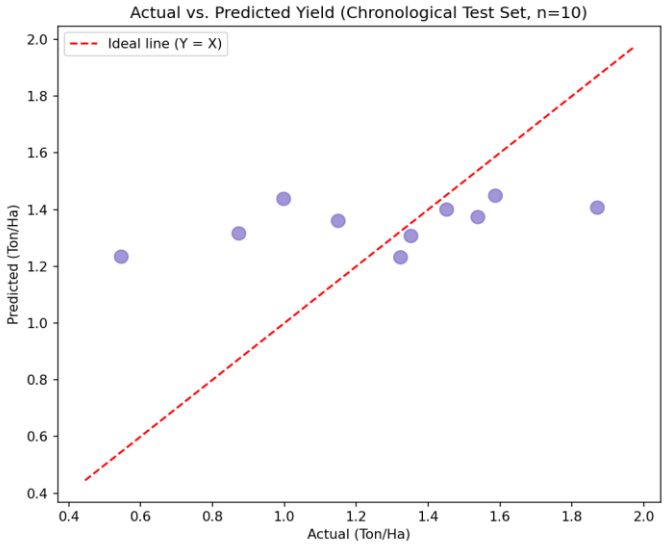

Figure 6. Scatter plot of actual versus LSTM-predicted FFB production (Ton/Ha) on the chronological test set (n = 10); the dashed line indicates perfect agreement.

The scatter plot in Figure 6 shows the same pattern from a different angle: predictions cluster in a narrower vertical band than the actual values span, consistent with the model regressing toward the mean of the training distribution rather than fully tracking cohort-to-cohort variation within this single target month.

System Implementation

The trained LSTM model was deployed through a Streamlit-based web dashboard (Figure 7), which allows plantation managers to run single-month prediction simulations, generate a 12-month production forecast, visualize results, and access historical prediction records stored in the integrated MySQL database.

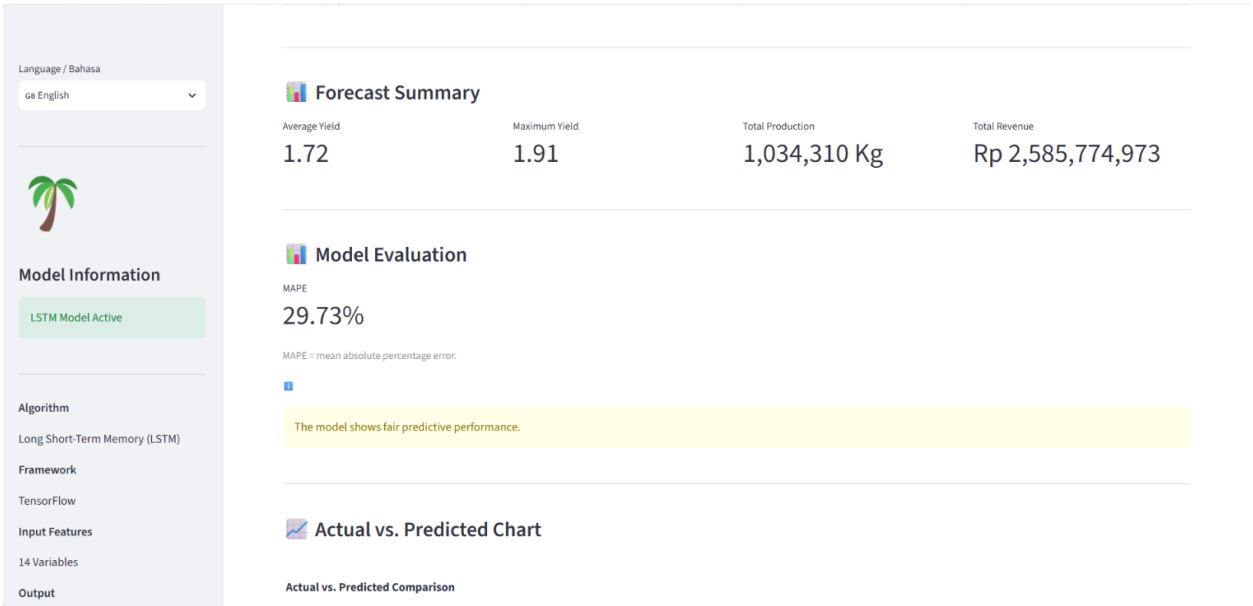

Figure 7. Streamlit-based web dashboard prototype for production forecasting, integrated with a MySQL database. Note: this screenshot reflects the corrected dashboard build, which reports MAPE directly rather than the derived "Akurasi (Informal)" = (100 − MAPE)

The dashboard's 12-month forecast is generated autoregressively, but from a user-specified planting scenario rather than from a selected cohort's actual historical record. The user enters four values planted area, current production, crop age, and a categorical rainfall condition and the system

synthesizes an initial 12-month input window from these values: each of the 12 months is assigned the entered production and rainfall figures directly, with no random perturbation so identical inputs always produce an identical initial window and, consequently, an identical 12-month forecast — and the three lag-feature columns (6-, 12-, and 24-month lagged rainfall and rainy days) are derived as fixed ratios of the entered rainfall value rather than drawn from the cohort's real historical series. The model then predicts month t+1 from this synthetic window, appends the prediction in place of the oldest month, and repeats this twelve times. Because both the initial window and the future exogenous features are synthesized rather than observed, the deployed dashboard functions as a what-if scenario simulator for exploring how a hypothetical planting profile might yield under the trained model, not as a tool for forecasting any specific planting-year cohort's real trajectory forward from its actual historical data; this is a materially more limited claim than the cohort-level historical evaluation reported in Results, and the two should not be conflated. Even within this scenario-based design, real future rainfall and rainy-day counts for any given planting profile would generally differ from the synthesized values used here, particularly across a seasonal transition, so multi-step forecast accuracy is expected to degrade with horizon in a way this study did not separately measure [24]. The dashboard should therefore be described, and is described throughout this manuscript, as a functioning prototype for scenario-based exploration rather than as a validated operational decision-support system tied to real cohort-level data; one specific functional property reproducibility was verified: identical inputs were confirmed to produce identical headline predictions and 12-month forecasts across repeated submissions, after the scenario-generation procedure was changed to remove the random perturbation described above. Broader usability or forecast-consistency testing of the dashboard was not performed, and connecting its forecasting logic to a selected cohort's actual historical record remains unimplemented.

Discussion

The rebuilt-data LSTM model achieved a MAPE of 29.73% under a chronological, purge-gapped evaluation designed specifically to prevent the temporal leakage identified in an earlier, randomly split version of this analysis [15]. This number reflects a fully rebuilt dataset: after an audit found that the original 503-entity, block-level dataset contained unresolved duplicate-month rows that let sequences labeled as 12-month windows silently compress as few as 7 real calendar months, the dataset was rebuilt directly from the plantation's original monthly production reports, at the coarser planting-year-cohort level, over a genuinely gap-free 48-month window. Benchmarking against simple baselines — a step an earlier version of this study omitted, and which the review process specifically requested — shows that on this rebuilt data — a single-month, 10-sequence test set — the model actually outperforms all five evaluated baselines (Table 7), a reversal of the previous, data-error-affected finding. Both the earlier under-performance and this pilot's over-performance are reported transparently, since neither is a large enough or clean enough evaluation to be treated as conclusive on its own.

Compared with the univariate LSTM studies summarized in Table 1, the present study addresses a more complex, multivariate forecasting problem (12 input features across 11 planting-year cohorts) and is the only one of the six studies compared to benchmark its LSTM against non-model baselines on the same evaluation partition. Husaini et al. [11], evaluated on a single company's univariate series, reported a MAPE of 50.87%, higher than the 29.73% obtained here; however, that comparison should be read cautiously given the differences in data source, entity structure, and evaluation design between the two studies. A numeric comparison against Setiawan and Zulkarnain [3] is not attempted here because their reported metric (accuracy) could not be confirmed as directly convertible to MAPE.

An important contribution of this study beyond the forecasting exercise itself is the demonstration that a trained LSTM model can be deployed within a web dashboard integrated with a relational database, enabling interactive forecasting and historical record-keeping rather than one-off offline evaluation. This remains a genuine, if modest, practical contribution: the dashboard's model-serving and database integration function as intended. However, the specific claim that the dashboard's 12-month forecasts are operationally reliable is not supported by the evidence collected in this study,

both because the underlying model's advantage over naive baselines was not established and because the autoregressive multi-step forecasting logic itself was not separately validated.

Several limitations should be highlighted. First, and most importantly, the model's apparent out-performance relative to naive baselines may be an artifact of the single, atypical test month (December 2023) rather than a general property of the model; a longer chronological test period, once more clean historical data become available, is needed to resolve this. Second, after the underlying dataset was rebuilt to fix a data-export bug, the usable entity pool shrank to 11 planting-year cohorts (from an original 503 purported block-level Entity_ID codes) and the resulting test-set size (n = 10) is very small, sharply limiting statistical confidence in any of the reported metrics. Third, the model tends to under-react to short-term variation between cohorts, smoothing predictions toward the training distribution's central tendency (Figures 4–5); architectures or training objectives explicitly penalizing under-reactivity, or residual-correction approaches such as the corrector-LSTM framework of Baghoussi et al. [25], may be a useful direction for future refinement. Fourth, the dashboard's recursive 12-month forecast runs on a synthesized, user-specified planting scenario rather than a selected cohort's real historical record, so its output is a what-if projection rather than a forecast tied to any specific cohort's actual trajectory or to future climate. Finally, this evaluation was not benchmarked against the plantation's own RKAP planning figures at all: RKAP records are tied to physical blocks, whereas the rebuilt dataset models planting-year cohorts aggregated across the whole plantation, so the block-level RKAP comparison attempted in an earlier version of this analysis is no longer methodologically applicable and has been removed rather than reused. A cohort- or plantation-level RKAP comparison, conducted on the same calendar months as a future test set, remains an important task, since RKAP — not naive statistical baselines — is the actual planning practice the proposed system is meant to improve upon.

Implications

These findings suggest that a multivariate LSTM approach can generalize across planting-year cohorts for a given calendar month better than naive baselines extrapolating from each cohort's own recent history, and that a database-integrated forecasting dashboard is technically achievable for plantation-scale deployment. However, because this advantage was observed on a single calendar month with only 10 test sequences, it should not yet be read as a general property of the model, and its readiness for operational use in production planning should be considered provisional rather than established. Any near-term use of the dashboard's forecasts for planning decisions should be accompanied by a simple persistence or moving-average forecast for comparison, until evaluation across additional chronological test months and a cohort- or plantation-level RKAP benchmark are available.

Limitations

This study has several limitations: (1) after an audit revealed a data-export bug in the original 503-entity, block-level dataset, genuinely clean monthly records could only be confirmed at the coarser planting-year-cohort level for 11 cohorts over a 48-month window (2020–2023), yielding a training partition of just 10 sequences and a test partition of just 10 sequences, all sharing a single target month (December 2023); this sample size is too small to support any claim of statistical reliability, and the reported metrics should be treated as a feasibility pilot; (2) under this design, the LSTM outperformed all five naive baselines evaluated, but because every test sequence shares one calendar month, this result may reflect that month's atypical conditions relative to each cohort's recent history rather than a general forecasting advantage, and cannot be extrapolated to other months without further testing; (3) retraining across 10 random seeds confirmed that the LSTM's advantage over all baselines is not an artifact of a favorable weight initialization (MAPE mean = 31.79%, SD = 4.59%, range 24.59–40.60%, still well below every baseline), but all 10 seeds share the same 10-sequence test partition and single target month, so this stability check does not substitute for evaluation across additional months; (4) block-level production, which was central to the original research design and to the dashboard's per-block framing, could not be reconstructed from any available source with genuine monthly granularity, so all findings here describe whole-plantation, cohort-aggregated behavior rather than individual-block behavior; (5) the dashboard's 12-month forecasting logic

operates on a synthesized, user-specified planting scenario rather than a selected cohort's real historical record — so its output should be read as an illustrative what-if projection, not as an operational forecast tied to any specific cohort's actual trajectory; this scenario-generation procedure was subsequently made fully deterministic (the random ±5% input perturbation used in earlier versions was removed), so identical user inputs now reproduce identical forecasts, but the logic has not itself been separately validated against real outcomes; (6) this evaluation was not benchmarked against RKAP planning figures, since RKAP records are tied to physical blocks and are not currently reconcilable with the cohort-level entities used here; and (7) the system is not yet integrated with the company's ERP platform, and model retraining is currently a manual process.

CONCLUSION

This study applied a multivariate Long Short-Term Memory (LSTM) model to forecast Fresh Fruit Bunch (FFB) production at Marihat Plantation using historical production, rainfall, rainy days, crop age, and planted area as input variables, under a chronological, purge-gapped evaluation design intended to prevent the temporal leakage present in an earlier, randomly split version of this analysis. After an audit uncovered a data-export bug in the original block-level dataset, the dataset was rebuilt directly from the plantation's original production reports at the planting-year-cohort level, over a genuinely gap-free 48-month window (2020–2023). On the resulting 10-sequence test set all sharing December 2023 as their target month across 10 cohorts the two-layer LSTM architecture achieved a MAPE of 29.73%, within the "reasonable" range under the thresholds adopted here, yet it outperformed all five evaluated baselines, including 1-month persistence. Given the extremely small, single-month test set, this result should be read as an encouraging feasibility finding rather than as evidence of validated forecasting superiority.

Beyond its predictive performance, this study contributes a working prototype that integrates a trained LSTM model into a web-based dashboard connected to a MySQL database, enabling forecasting, visualization, and historical record-keeping. This dashboard is best characterized as a technically functioning prototype rather than a validated decision-support system: its 12-month autoregressive forecasting logic operates on a synthesized, user-specified planting scenario rather than a selected cohort's real historical record; this scenario-generation procedure is deterministic (identical inputs reproduce identical forecasts), but the forecasting logic has not itself been validated against real outcomes, and no RKAP comparison was possible with the cohort-level data available in this study, since RKAP records are tied to physical blocks rather than planting-year cohorts.

Future research should prioritize five directions: (1) extending the chronological evaluation as more historical months accumulate, to test whether the LSTM's advantage over naive baselines observed on this single month generalizes across a longer and more varied test period, or was specific to December 2023; (2) conducting a cohort- or plantation-level RKAP benchmark, since RKAP not statistical baselines is the planning practice this system is intended to improve upon, and no such comparison was attempted in this rebuilt-data analysis; (3) exploring alternative architectures (GRU, Bidirectional LSTM, CNN-LSTM hybrids) and residual-correction or fuzzy-inference approaches such as fuzzy LSTM [26] that may better track short-term cohort-level variation; (4) connecting the dashboard's forecasting logic to a selected cohort's real historical record rather than the current synthesized, user-specified scenario so that its 12-month projections can be evaluated as genuine cohort-level forecasts; and (5) validating or replacing the dashboard's recursive multi-step forecasting logic, for example by coupling it with an independent climate/rainfall forecast rather than a last-observed-value assumption [24], and by integrating the system with the plantation's ERP platform and an automated retraining mechanism.

ACKNOWLEDGMENT

The authors would like to express their deepest gratitude to the Faculty of Tarbiyah and Teacher The authors would like to express gratitude to the management of the Marihat Plantation for provid-ing access to the production data used in this study, as well as to the Indonesian Palm Oil Institute and the academic advisor for their guidance and support throughout this research process.

AUTHOR CONTRIBUTION STATEMENT

RZT conceptualized the study, curated and reconstructed the dataset, developed the LSTM model and baseline experiments, conducted the formal analysis, created the visualizations, developed the Streamlit–MySQL dashboard, and drafted the manuscript. RAS and SLR provided methodological guidance, supervision, validation, interpretation of the findings, and critical revision of the manuscript. All authors reviewed and approved the final manuscript for submission.

AI DISCLOSURE STATEMENT

The authors used an AI assistant to support the drafting of selected passages, text summarization, and language refinement. The AI assistant was not used to generate or modify the research data, train or evaluate the forecasting models, perform the analysis, or determine the study's conclusions. All AI-assisted content was critically reviewed and edited by the authors, who take full responsibility for the accuracy, integrity, and content of this publication.

REFERENCES

[1] T. F. Suardi, L. Sulistyowati, T. I. Noor, and I. Setiawan, "Analysis of the Sustainability Level of Smallholder Oil Palm Agribusiness in Labuhanbatu Regency, North Sumatra," *Agric.*, vol. 12, no. 9, 2022, doi: 10.3390/agriculture12091469.

[2] N. Khan *et al.*, "Prediction of Oil Palm Yield Using Machine Learning in the Perspective of Fluctuating Weather and Soil Moisture Conditions: Evaluation of a Generic Workflow," *Plants*, vol. 11, no. 13, 2022, doi: 10.3390/plants11131697.

[3] N. H. Setiawan and Zulkarnain, "Forecasting Palm Oil Production Using Long Short-Term Memory (Lstm) With Time Series Cross Validation (Tscv)," *Int. J. Soc. Serv. Res.*, vol. 4, no. 05, pp. 1237–1251, 2024, doi: 10.46799/ijssr.v4i05.780.

[4] A. Casolaro, V. Capone, G. Iannuzzo, and F. Camastra, "Deep Learning for Time Series Forecasting: Advances and Open Problems," *Inf.*, vol. 14, no. 11, 2023, doi: 10.3390/info14110598.

[5] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philos. Trans. R. Soc. A*, vol. 379, no. 2194, p. 20200209, 2021, doi: 10.1098/rsta.2020.0209.

[6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[7] M. Krichen and A. Mihoub, "Long Short-Term Memory Networks: A Comprehensive Survey," *AI*, vol. 6, no. 9, pp. 1–21, 2025, doi: 10.3390/ai6090215.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/

[9] F. Ramadhan and M. Fachrie, "Implementasi Algoritma Long Short-Term Memory Pada Sistem Prediksi Hasil Panen Sawit," *J. Inform. Teknol. dan Sains*, vol. 6, no. 4, pp. 937–944, Nov. 2024, doi: 10.51401/jinteks.v6i4.4876.

[10] A. P. Putra and A. M. Harahap, "Implementasi Time Series Forecasting dengan Algoritma LSTM untuk Pemantauan dan Prediksi Produktivitas Kelapa Sawit Berdasarkan Hasil Panen," *JURIKOM (Jurnal Ris. Komputer)*, vol. 12, no. 2, pp. 46–56, Apr. 2025, doi: 10.30865/jurikom.v12i2.8495.

[11] F. Husaini, I. Permana, M. Afdal, and F. N. Salisah, "Penerapan Algoritma Long Short-Term Memory untuk Prediksi Produksi Kelapa Sawit," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 2, pp. 366–374, Feb. 2024, doi: 10.57152/malcom.v4i2.1187.

[12] M. Syarovy *et al.*, *Prediction of Oil Palm Production Using Recurrent Neural Network Long Short-Term Memory (RNN-LSTM)*, vol. 1. Atlantis Press International BV, 2023. doi: 10.2991/978-94-6463-122-7_6.

[13] K. Aqbar and R. A. Supomo, "Performance Analysis of LSTM and XGBoost Models Optimization in Forecasting Crude Palm Oil (CPO) Production at Palm Oil Mill (POM)," *Int. J. Comput. Appl.*, vol. 185, no. 17, pp. 37–44, Jun. 2023, doi: 10.5120/ijca2023922890.

[14] N. Khan, M. A. Kamaruddin, U. U. Sheikh, Y. Yusup, and M. P. Bakht, "Oil Palm and Machine Learning: Reviewing One Decade of Ideas, Innovations, Applications, and Gaps," *Agriculture*, vol. 11, no. 9, p. 832, Aug. 2021, doi: 10.3390/agriculture11090832.

[15] H. Hewamalage, K. Ackermann, and C. Bergmeir, "Forecast evaluation for data scientists: common pitfalls and best practices," *Data Min. Knowl. Discov.*, vol. 37, no. 2, pp. 788–832, Mar. 2023, doi: 10.1007/s10618-022-00894-5.

[16] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd Edition)*, O'Reilly Media, 2022. [Online]. Available: https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/

[17] N. I. A. Kader, U. K. Yusof, M. N. A. Khalid, and N. R. N. Husain, "A Review of Long Short-Term Memory Approach for Time Series Analysis and Forecasting," in *Proceedings of the 2nd International Conference on Emerging Technologies and Intelligent Systems (ICETIS 2022)*, M. A. Al-Sharafi, M. Al-Emran, M. N. Al-Kabi, and K. Shaalan, Eds., Cham: Springer, 2023, pp. 12–21. doi: 10.1007/978-3-031-20429-6_2.

[18] J. Brownlee, *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2018. [Online]. Available: https://machinelearningmastery.com/deep-learning-for-time-series-forecasting/

[19] H. Dhake, Y. Kashyap, and P. Kosmopoulos, "Algorithms for Hyperparameter Tuning of LSTMs for Time Series Forecasting," *Remote Sens.*, vol. 15, no. 8, p. 2076, Apr. 2023, doi: 10.3390/rs15082076.

[20] A. Pranolo *et al.*, "Enhanced Multivariate Time Series Analysis Using LSTM: A Comparative Study of Min-Max and Z-Score Normalization Techniques," *Ilk. J. Ilm.*, vol. 16, no. 2, pp. 210–220, Aug. 2024, doi: 10.33096/ilkom.v16i2.2333.210-220.

[21] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Monash University, 2021. [Online]. Available: https://otexts.com/fpp3/

[22] D. Koutsandreas, E. Spiliotis, F. Petropoulos, and V. Assimakopoulos, "On the selection of forecasting accuracy measures," *J. Oper. Res. Soc.*, vol. 73, no. 5, pp. 937–954, May 2022, doi: 10.1080/01605682.2021.1892464.

[23] C. D. Lewis, *Industrial and Business Forecasting Methods: A Practical Guide to Exponential Smoothing and Curve Fitting*. London: Butterworth Scientific, 1982.

[24] E. Strøm and O. E. Gundersen, "Performance metrics for multi-step forecasting measuring win-loss, seasonal variance and forecast stability: an empirical study," *Appl. Intell.*, vol. 54, no. 21, pp. 10490–10515, Nov. 2024, doi: 10.1007/s10489-024-05715-4.

[25] Y. Baghoussi, C. Soares, and J. Mendes-Moreira, "Corrector LSTM: built-in training data correction for improved time-series forecasting," *Neural Comput. Appl.*, vol. 36, no. 26, pp. 16213–16231, Sep. 2024, doi: 10.1007/s00521-024-09962-x.

[26] W. Wang, J. Shao, and H. Jumahong, "Fuzzy inference-based LSTM for long-term time series prediction," *Sci. Rep.*, vol. 13, no. 1, p. 20359, Nov. 2023, doi: 10.1038/s41598-023-47812-3.