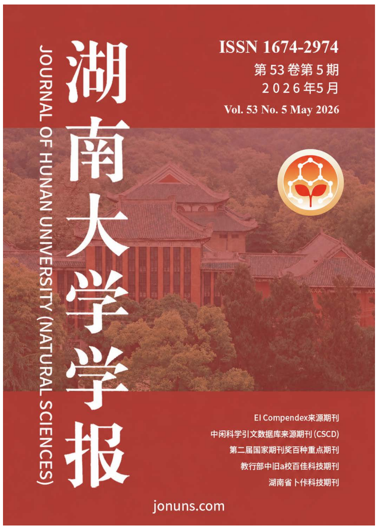



**Journal of Hunan University**

**(Natural Sciences)**

**Vol. 53 No. 5**

**May 2026**

**Available online at**

**https://jonuns.com**

Open Access Article https://doi.org/10.55463/issn.1674-2974.53.5.10

# Predicting Total Fish Production Using An LSTM-Based Linear Attention Model

**Nancy Jeane Tuturoong[1], Kawilarang Warouw Alex Masengi[1], Muhamad Dwisnanto Putro[1], Jimmy Reagen Robot[2], Lusia Manu[1]**

**[1] Postgraduate Program, Universitas Sam Ratulangi, Manado, Indonesia,**

**[2] Informatics Study Program, Sam Ratulangi University Manado, Indonesia,**

* Corresponding author: nancy.tuturoong@unsrat.ac.id

**Article History:**

*Received:* March 27, 2026
*Revised:* May 11, 2026
*Accepted:* May 20, 2026
*Published:* May 29, 2026

**Abstract:** Predicting fish production is important for supporting food security and sustainable fisheries management in Indonesia, particularly in the context of science-based policy making and catch quota regulation. However, accurate forecasting remains challenging due to inconsistencies in fisheries logbook data and the difficulty of capturing complex temporal patterns in production records. This study proposes a deep learning-based predictive framework for forecasting total fish production using a Trapezoid Long Short-Term Memory Linear Attention model. The model integrates an LSTM architecture with a linear attention mechanism to capture temporal dependencies and assign greater weight to relevant input features. The annual total production values were normalized using a global maximum value calculated across all fish species and years in the dataset. Model performance was evaluated using root mean squared error (RMSE) and mean absolute error (MAE). The experimental results produced an RMSE of 0.00465 and an MAE of 0.00244, indicating promising predictive performance after normalization. The model was trained by minimizing the mean squared error loss function. The predicted total production trends may provide useful scientific evidence for monitoring fishery production patterns, supporting precautionary stock management, and informing sustainable catch quota policies. Overall, the proposed model offers a data-driven approach for improving fisheries forecasting and strengthening evidence-based food security planning in Indonesia.

**Keywords:** fish production prediction; deep learning; LSTM; linear attention; sustainable fisheries management; food security; catch quota policy; Indonesia.

# 基于 LSTM 线性注意力模型的鱼类总产量预测

**摘要：** 鱼类产量预测对于支持印度尼西亚的粮食安全和可持续渔业管理具有重要意义，尤其是在科学决策和捕捞配额监管背景下。然而，由于渔业日志数据存在不一致性，以及生产记录中复杂时间模式难以捕捉，准确预测仍然面临挑战。本研究提出一种基于深度学习的预测框架，采用梯形长短期记忆线性注意力模型对鱼类总产量进行预测。该模型将 LSTM 架构与线性注意力机制相结合，以捕捉时间依赖关系，并对相关输入特征赋予更高权重。年度总产量值采用基于数据集中所有鱼类种类和所有年份计算得到的全局最大值进行归一化处理。模型性能通过均方根误差（RMSE）和平均绝对误差（MAE）进行评估。实验结果显示，模型的 RMSE 为 0.00465，MAE 为 0.00244，表明归一化后模型具有较好的预测性能。模型通过最小化均方误差损失函数进行训练。预测得到的总产量趋势可为监测渔业生产模式、支持预防性资源管理以及制定可持续捕捞配额政策提供有用的科学依据。总体而言，所提出的模型为改进渔业预测和加强印度尼西亚基于证据的粮食安全规划提供了一种数据驱动方法。

**关键词：** 鱼类产量预测；深度学习；LSTM；线性注意力；可持续渔业管理；粮食安全；捕捞配额政策；印度尼西亚。

## 1. Introduction

A marine country is reflected in its extensive territorial waters, abundant resources, and the critical role of the marine sector in its economy and society. The country possesses a maritime area of approximately 5.8 million square kilometers [1]. These areas encompass Indonesia’s territorial waters, exclusive economic zone, rivers, lakes, reservoirs, swamps, and other water bodies with significant potential for resource exploitation. The fisheries sector plays a crucial role in supporting both local communities and the national economy [2], [3], [4], [5]. However, the vast expanse of the marine domain necessitates intensive management and monitoring efforts, which frequently encounter various challenges [3], [5], [6], [7], [8], [9]. The fisheries sector is implemented through the national fish stock management system. This sector contributes substantially to national food security [2], [3], [7], [10], [11]. The primary benefit lies in ensuring the long-term availability of fish resources through sustainable management practices [6], [8], [9], [10], [11], [12]. A resilient and sustainable food system not only supports nutritional security but also generates broad economic benefits and strengthens the nation’s capacity [[3], [4], [7]. Monitoring and analysis of fishing quota utilization data, such as those obtained from Fishing Logbooks, serve as essential inputs for fisheries management and policy formulation [5], [6]. This process is expected to enhance the credibility of licensing data and support more effective fisheries planning. Accurate and objective fisheries data remain limited due to issues such as data manipulation and inconsistencies between logbook records and actual measurements at fish auction sites. Production data from small-scale fishers, who collectively contribute a significant portion of total catches, is often underreported or recorded only in aggregated form [5], [10], [11], [13]. In addition, fisheries data is scattered across multiple institutions and lacks real-time integration, which hampers scientific analysis and effective resource management.

Prediction technology plays a crucial role in modern data-driven systems. Prediction technology serves as a tool that utilizes historical data to support future planning. These technologies analyze large datasets to optimize operations. The deep learning method analyzes large datasets to predict future events, optimize resource allocation, and improve operational efficiency [14], [15], [16]. It learns hierarchical feature representations to model complex and nonlinear relationships within data [13], [17], [18], [19]. This method provides advanced analytical capabilities to enhance performance across various applications. The Long Short-Term Memory (LSTM) network processes sequential data to capture long-term dependencies and temporal patterns [20], [21], [22]. LSTM models utilize memory cells and gating mechanisms to improve prediction accuracy in time-series analysis. On the other hand, the availability and quality of data sources play a crucial role in enabling machine learning methods to achieve satisfactory accuracy [12], [23]. However, the limited availability of local fish catch datasets has motivated this study to generate a numerical dataset derived from annual fish catch records [14], [24].

The integration of recurrent learning models enhances predictive performance by adapting model parameters based on sequential feedback [15], [17], [25], [26]. This hybrid approach improves system efficiency by optimizing long-term decision-making in dynamic environments [15], [17], [25], [26], [27]. Machine learning models often struggle to identify the most important parts of the input required to produce accurate outputs. These models often treat all input features equally, which reduces their ability to capture critical dependencies within temporal data[9], [28]. The incorporation of attention mechanisms within the Long Short-Term Memory with Linear Attention (LSTM-LA) architecture addresses this limitation by enabling the model to focus selectively on features that contribute most to prediction accuracy, particularly in attention-based modeling [3], [4], [21], [26], [29]. As a result, the proposed framework enhances interpretability and predictive performance, particularly in complex time-series predictive modeling tasks [1 5] [28]. In summary, the attention module acts as an adaptive filtering and weighting mechanism that enables machine learning models to selectively process input features and significantly improve performance on complex tasks [9], [28]. Along with advances in deep learning, the attention module has emerged as a powerful approach that enables models to concentrate on critical input features and achieve better performance [9], [28]. Attention modules serve as an effective technique that enables models to dynamically assign importance to different parts of the input data during output generation [9], [28]. These modules allow models to focus on the most relevant information and disregard less significant features, thereby emulating human cognitive attention mechanisms [9], [28]. The attention module operates by computing a set of weights for each input element. Elements assigned higher weights represent the most relevant information for the current task, thereby guiding the model to focus more effectively on those critical inputs [9], [28]. This study proposes a Deep Learning-based predictive framework to produce accurate forecasts that support sustainable fisheries resource management [13], [18]. The framework integrates a Long Short-Term Memory with Linear Attention (LLA) network to enhance the model's capability in predicting total fish production [17], [26], [28].

**Table 1. Proposed Dataset**

| Field | Description | Proposed Role |
|---|---|---|
| jenis_ikan_clean | Standardized fish name after text cleaning | Categorical feature / reference |
| jenis_ikan_encoded | Numeric encoding of fish type | Feature |
| Tahun | Production year | Feature |
| Januari–Desember | Monthly production values | Numeric features |
| Produksi_Tahun_Lalu | Previous year production | Lag feature |
| Jumlah | Raw annual total fish production | Main target |
| Total_Produksi_Tahunan | Normalized annual total production | Optional target only, not predictor |
| Growth_Rate | Annual growth relative to previous production | Drop from predictors |
| split | Train/Test label | New management field |
| anomaly_flag | Marks rows where monthly total does not match annual total | New quality-control field |

**Table 2. Proposed training and testing split**

| Split | Year range | Records | Percentages |
|---|---|---|---|
| Training | 2018–2023 | 245 | 65.2% |
| Testing | 2024–2025 | 131 | 34.8% |

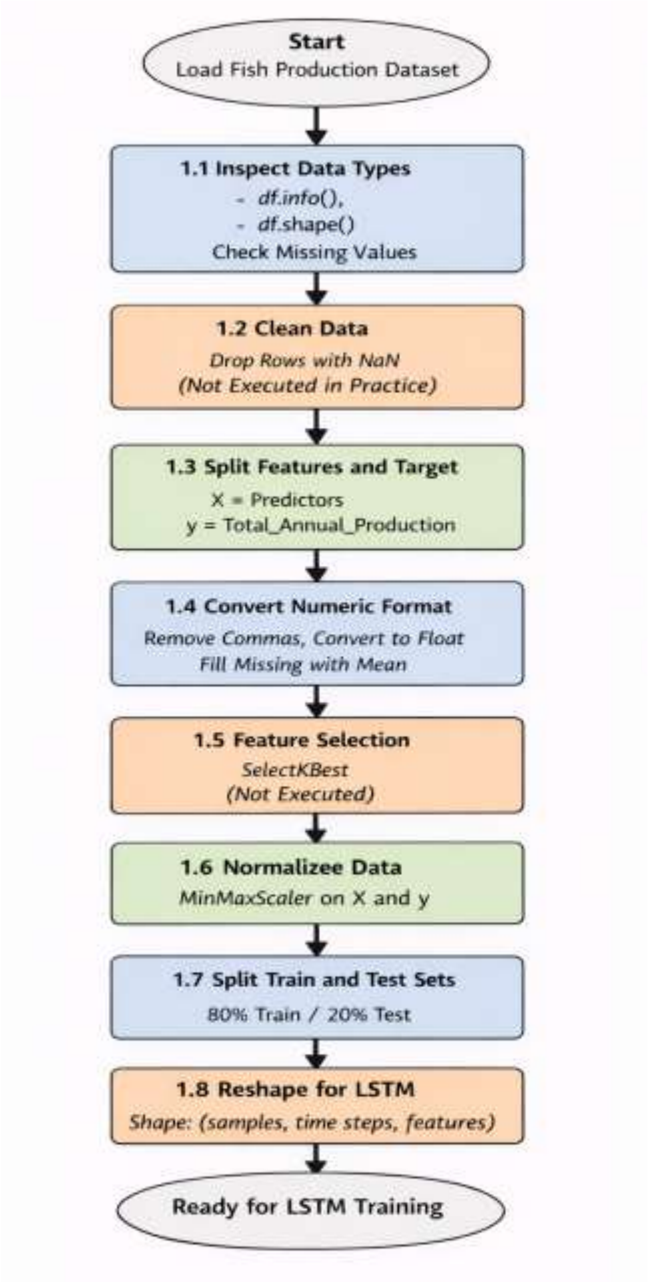



**Figure 1. Fish Production Dataset Preprocessing Flowchart**

The linear attention mechanism improves prediction performance by focusing on the most relevant features and reducing the influence of less significant data [9], [28]. The main contributions of this research are summarized as follows.

1. A new fish catch prediction model is proposed to estimate the total production of pelagic fish, supporting food security analysis in maritime regions [17], [26], [30].
2. A Linear Enhancement Block is introduced to improve the performance of the trapezoid network. The proposed LSTM with Linear Attention (LLA) offers a lightweight yet effective enhancement without relying on extensive neuron weights [9], [28].
3. To support the analysis of fish catch predictions, this study developed a dataset on pelagic fish production in the waters off Bitung for predictive modeling. This dataset is expected to support the development of an artificial intelligence-based fisheries prediction system [12], [13], [14], [18].
4. A comprehensive evaluation and comparison with other prediction methods and architectures is conducted, including an in-depth analysis of the proposed LSTM structure to assess the contribution of each component and module [17], [26], [28], [31].

The research gaps addressed in this study are as follows. First, fisheries production data in Indonesia remains inconsistent and unreliable due to logbook discrepancies and underreporting, with no cleaned species-level dataset available for deep learning prediction [5], [10], [11], [13]. Second, LSTM-based deep learning has not been applied to species-level total annual fish production prediction in Indonesian pelagic fisheries, despite its proven effectiveness in sequential data analysis [20], [21], [22]. Third, the integration of a linear attention mechanism into a compact LSTM architecture for fish production forecasting remains unexplored, despite demonstrated improvements in feature weighting for sequential prediction tasks [9], [17], [26], [28]. Fourth, no data-driven predictive model has been explicitly developed to support quota-based catch regulation and precautionary stock monitoring within Indonesia's national fisheries management framework [5], [6], [11], [12]. This paper novelty statement is a Trapezoid Long Short-Term Memory Linear Attention (LLA) network, enhanced with a linear attention mechanism, to predict total fish production

## 2. Methods and Materials

### 2.1. Proposed Dataset

The dataset for this study comprises fish production figures spanning the years 2018 to 2025. It contains 376 376 rows data fish species and 16 input features variables, with each rows detailing the production of a specific fish type in a particular year. The variables include the fish name (*jenis_ikan*), monthly production values from January to December, the year (*Tahun*), annual production totals, growth rate, previous-year production, and a fish-type encoding Table 1. While no empty cells were found, the raw data necessitate cleaning due to inconsistent numeric formats (comma-separated text values for production) and non-standardized categorical labels for fish names (including local, English, and coded labels).

As shown in Figure 1, the initial preprocessing must focus on standardizing the "*jenis_ikan*" column by removing extraneous spaces and consolidating variations of fish names for consistent representation. Subsequently, all monthly production columns and the "*Jumlah*" (raw annual total) column must be converted to a numeric format. A validation check should then be performed by comparing the raw annual total (*Jumlah*) with the sum of the monthly values. After numeric cleaning, 374 out of the 376 records show a match between the annual total and the sum of monthly production, indicating only minor discrepancies require manual review. It is crucial to note that Growth_Rate and "*Total_Produksi_Tahunan*" (a normalized annual total) are derived from the annual production total, and therefore, should be excluded as predictor variables for annual production modeling to prevent data leakage.

As shown in Table 2, the normalized dataset was partitioned using a random 80/20 split with a fixed random seed (random_state = 42), yielding 300 training samples and 76 testing samples. This partitioning strategy was selected based on the cross-sectional structure of the dataset, in which each rows represents an independent species-year unit rather than a continuous sequential record. This is vital for predictive modeling integrity. Unlike random sampling, this approach prevents future data leakage, ensuring the model learns only from historical data for a realistic assessment of future fish production. A more effective predictive strategy will use lag variables and partial-year monthly data instead of fields derived directly from the annual total, which is highly consistent with the sum of monthly figures. Model performance will be evaluated using MAE, RMSE, and Huber Loss.

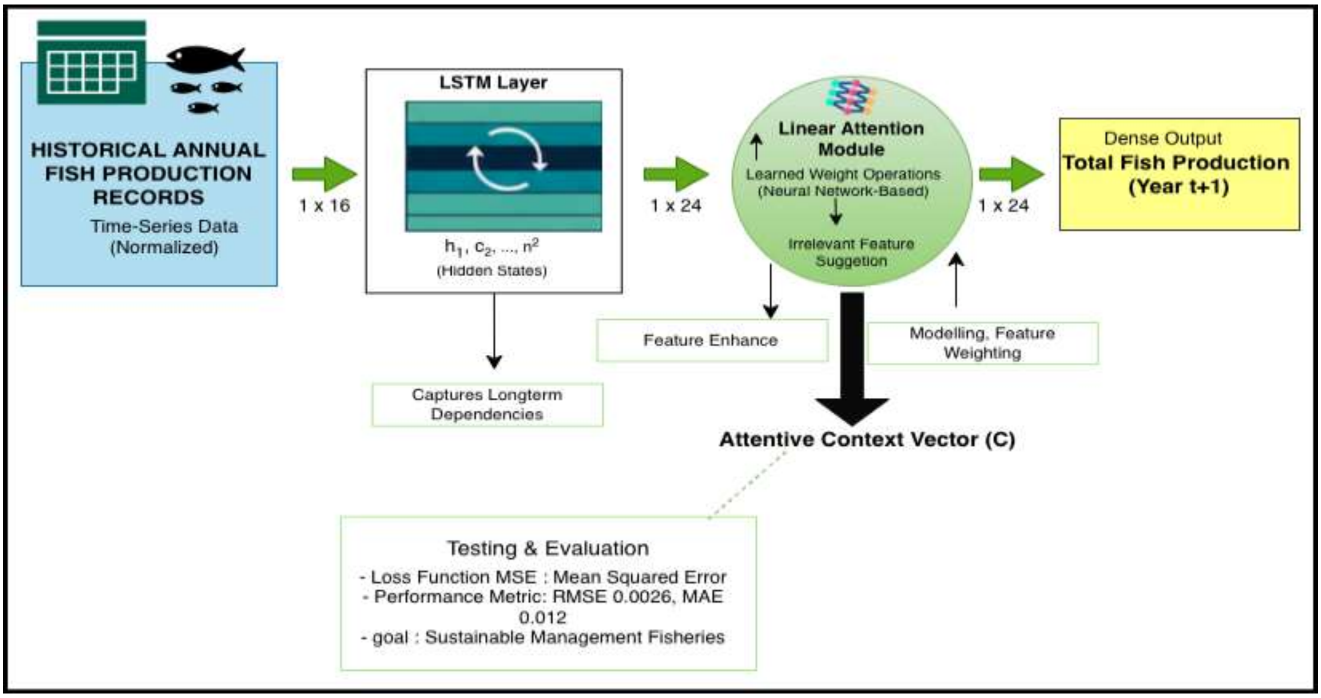



**Figure 2. The Proposed of LSTM Linear Attention Architecture**

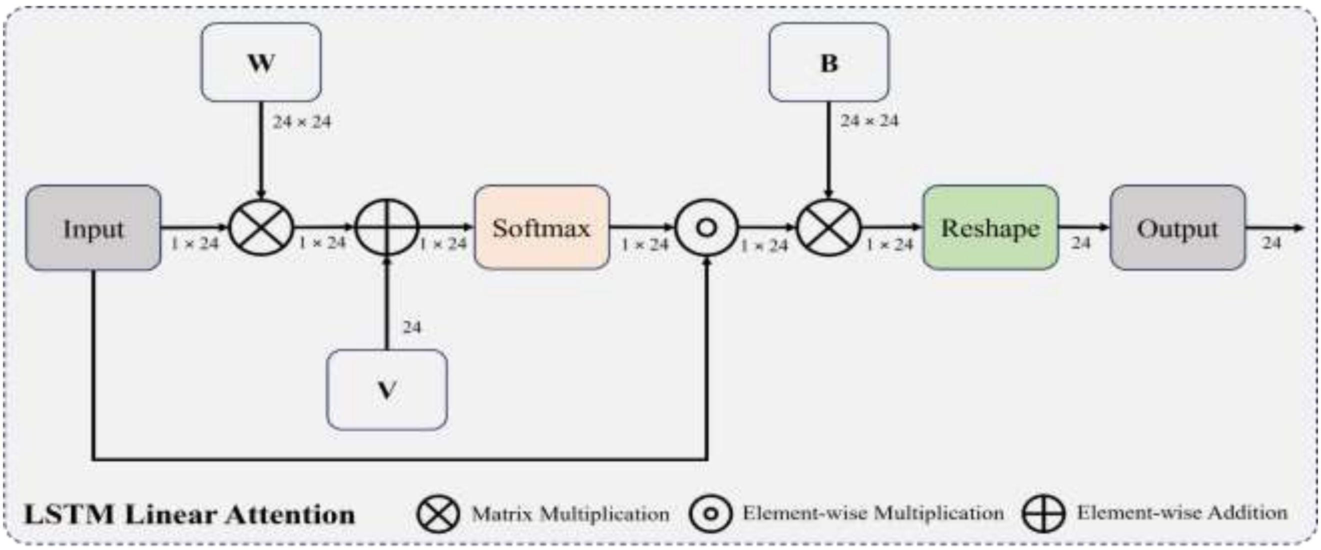



**Figure 3. LSTM Linear Attention (LLA) Module**

### 2.2. Proposed General Architecture

Based on Figure 2, the proposed architecture integrates Long Short-Term Memory (LSTM) units with residual connections and a multi-head attention mechanism to model long-range temporal dependencies and complex feature interactions in high-dimensional data. The process is illustrated using a trapezoidal representation to describe hierarchical feature transformation, where the lower base represents high-dimensional raw time-series data containing noise, variability, and detailed temporal fluctuations that serve as the foundation for feature extraction. As the data pass through the LSTM and attention layers, the representation is progressively refined and compressed, as indicated by the trapezoid's inclined sides, which reflect reduced dimensionality while preserving essential temporal patterns. During this stage, the model filters noise and extracts more meaningful features. At the top of the trapezoid, the representation becomes compact and highly abstract, retaining only the most relevant temporal information as a semantic representation optimized for predictive modeling. Overall, this architecture performs progressive refinement, noise reduction, and feature abstraction, leading to improved model stability and enhanced generalization in long-term predictive modeling tasks.

As shown in Table 3, the proposed architecture consists of five layers for modeling temporal dependencies and generating regression outputs. The input layer receives data of size N × 1 × 16, where N denotes the batch size and each sample contains 16 features at a single time step. This layer has no trainable parameters. An LSTM layer with 24 hidden units performs temporal feature extraction and produces an output of size N × 1 × 24 with 3,936 parameters. A Linear Attention (LLA) module is applied to transform the output into a context vector of size N × 24 by assigning adaptive weights to each feature. This module introduces 600 parameters. The context vector is processed by a fully connected layer with 24 neurons and 600 parameters for feature refinement. A second fully connected layer with one neuron generates the regression output of size N × 1, adding 25 parameters. The model contains a total of 5,161 trainable parameters, indicating a compact architecture suitable for efficient training and accurate prediction.

### 2.3. LSTM Linear Attention (LLA)

The proposed method in this research focuses on integrating a Linear Attention mechanism into the Long Short-Term Memory (LSTM) network to improve the accuracy of food stock and fish catch predictive modeling. Unlike standard attention mechanisms that possess quadratic complexity, the LLA architecture is designed to be computationally efficient while remaining capable of capturing crucial temporal dependencies. The effectiveness of LSTM-based architectures has been widely validated across various data analysis domains.

The proposed model structure involves processing an input $X$ with a dimension of 24 × 1, as shown in Figure 3. The process begins with a linear transformation, where the input $X$ is multiplied by a weight matrix ($W$) where $W \in \mathbb{R}^{24 \times 24}$ using matrix multiplication. To strengthen feature extraction, the model adds a weight vector ($V$), resulting in the intermediate matrix $M$ through the following equation:

$$M = (X \times W) + V. \tag{1}$$

**Table 3. The LSTM model summary**

| Layer | Output Dimension | Parameter |
|---|---|---|
| Input Layer | (Batch Size, 1, 16) | 0 |
| LSTM 1 | (Batch Size, 1, 24) | 3,936 |
| LLA | (Batch Size, 24) | 600 |
| Dense 1 | (Batch Size, 24) | 600 |
| Dense 2 | (Batch Size, 1) | 25 |
| Total number of parameters: 5,161 | | |


**Figure 4. Technical setup for proposed LSTM-LA**

The linear operation defined by the weight matrix $W$ enables the model to learn and discriminate relevant information from the input. This mechanism operates through matrix multiplication between the input $X$ and the weight matrix $W$, effectively transforming and filtering the input features. Subsequently, an attention map $M$ is generated via a linear transformation, followed by the application of the Softmax function to normalize the attention scores. This normalization allows the model to assign probabilistic weights to specific time steps or features that exhibit strong correlations with the prediction target.

Such a selective weighting mechanism is particularly important in forecasting tasks, where seasonal patterns and temporal dependencies play a significant role in determining future outcomes. Furthermore, the resulting attention weights are applied to the original input $X$ to produce a refined representation vector that emphasizes the most informative components of the data. The linear attention mechanism can be formally expressed as follows:

$$A = Softmax\ (M)\ *\ X. \qquad (2)$$

The Softmax activation function generates a probability-based attention map to recalibrate the input $X$, emphasizing the most important information while suppressing less relevant features. This mechanism enables the model to focus on significant components by increasing the contrast between informative and non-informative elements across the 24 feature dimensions. Subsequently, a linear transformation is applied to further enhance feature extraction using a weight matrix $(B)$ where $B \in \mathbb{R}^{24 \times 24}$, which is multiplied with the input through matrix multiplication. This operation aims to capture more discriminative feature representations. Furthermore, a reshape operation is performed to convert the feature map into a vector representation, ensuring compatibility with the fully connected layer. The overall process can be formulated as follows:

$$LLA = A\ \times\ B. \qquad (3)$$

The integration of this attention enables the LSTM unit to prioritize salient information from dynamic maritime and environmental data sequences while simultaneously reducing memory overhead. By incorporating real-time environmental parameters within an optimized architecture, the model achieves high operational efficiency for food stock management. The proposed mechanism enhances global contextual representation through efficient feature selection, allowing the network to identify relevant patterns during the prediction process. Probability maps are employed to selectively emphasize informative features while suppressing less relevant elements, thereby refining the overall feature representation.

### 2.4. Implementation Setup

The simulation of the proposed LSTM-LA model shown in Figure 4 was conducted in a Python-based environment using NumPy, Pandas, TensorFlow, scikit-learn, and Matplotlib. All experiments were executed on a server platform equipped with an Intel Xeon processor, 2 vCPU cores, 2.3 GHz clock speed, and 12 GB RAM. In the implementation pipeline, the dataset was first loaded and inspected to verify dimensional consistency and variable structure. The target variable was defined as "*Total_Produksi_Tahunan*", while the predictor matrix was constructed by excluding "*Total_Produksi_Tahunan*, *jenis_ikan*", and "*Jumlah*", and *Growth_Rate* from the input set, as the latter two are either direct components or derivatives of the target variable and would introduce data leakage if retained as predictors from the input set. To ensure numerical compatibility, the variables were transformed into machine-readable numeric format, followed by feature scaling for both the input matrix and the target output. This normalization procedure was employed to place all variables within a comparable range and to improve convergence stability during model training.

After preprocessing, the normalized dataset was divided into 80% training data and 20% testing data using a fix random seed (random_state = 42) to preserve reproducibility across experimental runs. The input matrix is then reshaped into a three-dimensional structure (samples, time steps, and features), as required by recurrent networks, with the final implementation using an input shape. The proposed LSTM-LLA architecture is implemented using the Keras functional API. The model is compiled with Mean Squared Error (MSE) as the loss function, the Adam optimizer with a learning rate of 0.001, and Mean Absolute Error (MAE) as the evaluation metric. Training is conducted for 200 epochs with a batch size of 32, and a callback mechanism is employed to save the best-performing model based on the minimum validation loss. For testing and performance validation, the saved best model was reloaded and applied to the unseen testing subset to generate predictions.

The predictive performance was then quantified using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Huber Loss, thereby allowing the model to be assessed from complementary error perspectives. This simulation and testing configuration demonstrates that the proposed LSTM-LA framework was implemented under a reproducible computational setup and evaluated through both direct testing and repeated validation to ensure reliable predictive performance. For full reproducibility, the following implementation parameters are fixed across all experiments: input feature = 16 (listed in the section 2.1, excluding Total_Produksi_Tahunan, jenis_ikan, Jumlah, and Growth_Rate); target variable = Total_Produksi_Tahunan(global_max normalized); input shape = (N, 1, 16); LSTM hidden units = 24; attention module parameters = 600, optimizer = adam (lr = 0.001); loss function = MSE; batch size = 32; maximum epocs = 200; random seed = 42; train-test split = 80/20; MinMaxScaler fitted on training set only; model checkpoint criterion = minimum validation loss (save_best_only= True); cross-validation = 5-fold on training set.

The predictive performance was then quantified using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Huber Loss, thereby allowing the model to be assessed from complementary error perspectives. This simulation and testing configuration demonstrates that the proposed LSTM-LA framework was implemented under a reproducible computational setup and evaluated through both direct testing and repeated validation to ensure reliable predictive performance.

# 3. Results and Discussions

## 3.1. Model Analysis

Machine learning architectures typically employ multiple operations to extract informative and discriminative features. These operations are organized into sequential layers that progressively refine representations toward accurate predictions. The proposed architecture consists of a trapezoidal baseline and a Linear Attention LSTM (LLA) module. The baseline acts as a foundational feature extractor, enabling the network to capture essential information, while the LLA module enhances key features through learned attention weights, thereby improving prediction performance.

In this work, we evaluate the performance of the proposed models through an ablation study to assess the impact of the LLA module on the baseline network. The trapezoidal model contains 4,561 trainable parameters and achieves loss values of 0.039, 0.0075, and 0.0008 for RMSE, MAE, and Huber loss, respectively. These results indicate competitive within the evaluated condition performance compared to other baseline models. Furthermore, the balanced number of extractor neurons at the input and output stages is designed to maintain feature consistency throughout the network. The LLA module has a positive impact on prediction performance, as shown in Table 4. Specifically, it reduces the loss values on the test data by 0.03435, 0.00506, and 0.00079 for RMSE, MAE, and Huber loss, respectively. This improvement is achieved without incurring substantial additional computational cost. Notably, the module introduces only 1,176 additional parameters, corresponding to a 25.78% increase in neuron weights. The proposed module employs a limited number of trainable parameters to identify relevant features, making it more efficient than conventional layers that require extensive operations. Furthermore, the generated probability weights regulate feature importance, effectively filtering out irrelevant information while preserving essential features.

## 3.2. Evaluation on Dataset

To establish a rigorous and transparent benchmark, all baseline models (LSTM-1 to LSTM-8) and architectures adopted from previous studies were fully re-implemented by coding them from scratch. To ensure absolute evaluation integrity, all compared methods were directly trained, validated, and tested using the exact same multi-dimensional fish production dataset (376 records) under identical operational constraints, including a 16-input feature space, MinMaxScaler normalization parameters, an 80/20 random partition strategy, and fixed global random seeds. Based on Table 5, a comparative evaluation is conducted between the proposed model and several LSTM-based baseline models to assess the performance improvement of the proposed approach. The results show that the baseline models exhibit varying performance depending on their architectural configurations and parameter sizes. The results indicate that LSTM-1 and LSTM-2 have identical parameter sizes of 9,553, implying comparable computational cost. However, LSTM-2 demonstrates competitive within the evaluated condition performance compared to LSTM-1. Specifically, LSTM-2 achieves a lower RMSE of 0.0103 compared to 0.0107 in LSTM-1, reducing the loss by 0.0004. Similarly, the MAE is reduced from 0.0035 to 0.0026, corresponding to a decrease of 0.0009. In contrast, both models exhibit identical Huber loss values of 0.0001, indicating comparable performance. These results suggest that LSTM-2 provides more accurate predictions than LSTM-1 without increasing computational cost. Subsequently, LSTM-3 utilizes a larger number of parameters, with 16,201. However, the results show that it produces significantly higher errors compared to LSTM-1 and LSTM-2, with an RMSE of 0.0764 and a Huber loss of 0.0029. In addition, the MAE is 0.0032, which is between the values of the previous LSTM models. In contrast, LSTM-4 uses 7,489 parameters and is more efficient. Despite this reduction in complexity, LSTM-4 achieves significantly better performance than LSTM-3, with an RMSE of 0.0104, reducing the error by 0.0660 from 0.0764. Similarly, the Huber loss decreased from 0.0029 to 0.0001, with a reduction of 0.0028. The MAE of LSTM-4 is 0.0030, showing a reduction of 0.0002 compared to LSTM-3.

**Table 4. Ablation study of the proposed model**

| Model | Number of Parameter | RMSE | MAE | Hubber Loss | K-Fold Cross Validation (5 folds) | | |
|---|---|---|---|---|---|---|---|
| | | | | | RMSE | MAE | Hubber Loss |
| Baseline Trapezoid [14], [15] | 4,561 | 0.039 | 0.0075 | 0.0008 | 0.0968 ± 0.0995 | 0.0301 ± 0.0259 | 0.0083 ± 0.0139 |
| **Trapezoid+ LLA** | **5,737** | **0.00465** | **0.00244** | **0.00001** | **0.01558 ± 0.01957** | **0.00357 ± 0.00407** | **0.00357 ± 0.00407** |

**Table 5. Performance comparison of the proposed baseline against representative feature extraction architectures**

| Model | Number of Parameter | RMSE | MAE | Hubber Loss | Configuration |
|---|---|---|---|---|---|
| LSTM-1 | 9,553 | 0.0107 | 0.0035 | 0.0001 | LSTM 32-LSTM 16-FC 1 |
| LSTM-2 | 9,553 | 0.0103 | 0.0026 | 0.0001 | LSTM 32-LSTM 16-FC 1 |
| LSTM-3 | 16,201 | 0.0764 | 0.0032 | 0.0029 | LSTM 50-LSTM 24-FC 1 |
| LSTM-4 | 7,489 | 0.0104 | 0.0030 | 0.0001 | LSTM 32-LSTM 24-FC 1 |
| LSTM-5 | 4,657 | 0.0137 | 0.0051 | 0.0001 | LSTM 24-LSTM 24-FC 1 |
| LSTM-6 | 15,073 | 0.0167 | 0.0039 | 0.0001 | LSTM 48-LSTM 48-FC 1 |
| LSTM-7 | 25,217 | 0.0630 | 0.0239 | 0.0020 | LSTM 64-LSTM 24-FC 1 |
| LSTM-8 | 4,657 | 0.0125 | 0.0052 | 0.0001 | LSTM 24-LSTM 24-FC 1 |
| **Trapezoid** | **4,561** | **0.0390** | **0.0075** | **0.0008** | **LSTM16-LSTM24-FC 1** |

**Table 6. Cross-study comparison of architectural complexity, computational efficiency, and reported performance metrics. RMSE, MAE and Huber Loss values for prior methods originate from different datasets and are presented as contextual reference, not direct benchmarks**

| Model | Number of Parameter | RMSE | MAE | Hubber Loss |
|---|---|---|---|---|
| Comesana, 2020 [18] | 7,505 | 0.03740 | 0.00590 | 0.00070 |
| Yucra, 2025 [13] | 31,673 | 0.02090 | 0.00650 | 0.00002 |
| Zhao, 2025 [28] | 7,041 | 0.02910 | 0.00600 | 0.00040 |
| Bilotta, 2022 [24] | 48,385 | 0.02210 | 0.00420 | 0.00020 |
| Moghar, 2020 [16] | 183,969 | 0.00290 | 0.00190 | 0.00001 |
| **Proposed** | **5,737** | **0.00465** | **0.00244** | **0.00001** |

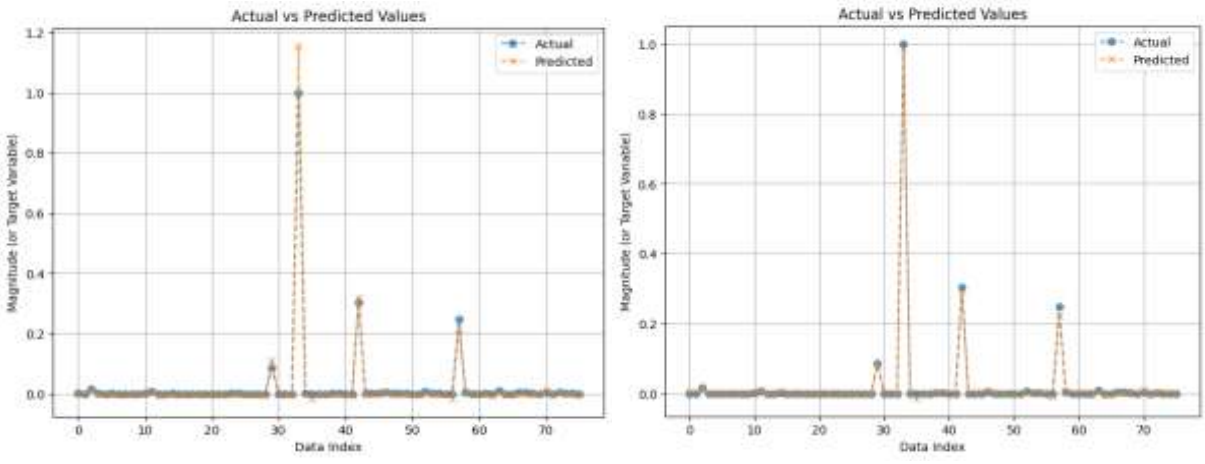
Furthermore, LSTM-5 uses fewer parameters, with only 4,657 parameters, making it more efficient. However, this reduction in complexity leads to higher errors. The RMSE is 0.0137, increasing by 0.0033 from LSTM-4. Similarly, the MAE increases to 0.0051, with a difference of 0.0021, while the Huber loss achieves 0.0001. On the other hand, LSTM-6 uses 15,073 parameters and produces a higher loss. The RMSE is 0.0167, increasing by 0.0030 from LSTM-5. The MAE is 0.0039, while the Huber loss is 0.0001. Similarly, LSTM-7 uses a large parameter of 25,217 and also produces higher errors. The RMSE is 0.0630, increasing by 0.0463 from LSTM-6. The MAE increases to 0.0239, with a difference of 0.02, while the Huber loss rises to 0.0020, increasing by 0.0019.

In addition, LSTM-8 uses a similar number of parameters as LSTM-5, with 4,657. The LSTM-8 achieves an RMSE loss of 0.0125, which is a reduction of 0.0012 from LSTM-5. However, the MAE increases to 0.0052, with a difference of 0.0001 from LSTM-5. Subsequently, the Huber loss achieves 0.0001. Based on the results, the Trapezoid model uses 4,561 parameters, which is lower than all baseline LSTM models. Furthermore, the Trapezoid model achieves an RMSE of 0.0390, an MAE of 0.0075, and a Huber loss of 0.0008. These results indicate higher errors compared to several LSTM models with lower error values. The Trapezoid model reduces the RMSE by 0.0374 and the Huber loss by 0.0021, while the MAE increases by 0.0043, compared to LSTM-3. In addition, the Trapezoid model shows better performance than LSTM-7, with lower errors and fewer parameters. Despite using significantly fewer parameters, the proposed model demonstrates competitive performance. This efficiency enables further enhancement of feature extraction by incorporating attention modules while keeping the number of parameters low.

To contextualize the proposed model within the broader LSTM-based deep learning literature, Table 6 provides a two-dimensional comparison againts five representatives prior methods that share the same LSTM architectural family. The first_dimension concerns architectural efficiency: parameter count, FLOPs, and inference time are architecture-level properties that remain valid for cross-study comparison regardless of dataset used. The second_dimension concerns reported performance metrics: the RMSE, MAE and Huber Loss values for methods [13], [16], [18], [24], and [28] are source from their original publication, where they were evaluated on different datasets and application domains. These values are therefore provided as contextual reference rather than as direct performance benchmarks. The primary controlled performance comparison, in which all models are evaluated on the identical fish production dataset under identical condition, is presented in Table 5.

The results show that the method in [18] achieves an RMSE of 0.03740, MAE of 0.00590, and Huber loss of 0.00070. It uses 7,505 parameters, representing a moderate model size compared to other methods in the table. On the other hand, the method in [13] uses a significantly larger number of parameters (31,673). However, the performance improvement is limited. The RMSE is reduced to 0.02090, a decrease of 0.0165 compared to [18]. In addition, the MAE increases to 0.00650, indicating a slight degradation in accuracy. In contrast, the Huber loss decreases to 0.00002, showing a reduction of 0.00068 compared to [18]. These results indicate that the method in [13] is not optimal, as the substantial increase in model complexity does not lead to consistent or significant performance improvements.

The model in [28] has 7,041 parameters, which is smaller than the models in [18] and [13]. It also performs better than [18], with an RMSE of 0.02910, meaning the error is lower by 0.0083. In addition, the MAE is 0.00600, which is slightly higher than [18] but 0.00050 lower than [9], despite using fewer parameters. Furthermore, compared to the work in [18], the Huber loss is 0.00040, which shows a reduction of 0.00030. Compared to [18], the method in [28] performs better by achieving lower RMSE and Huber loss with 464 fewer parameters, making it a more accurate and efficient model. In addition, the method in [24] achieves an RMSE of 0.02210, which is 0.007 lower than that of the model in [28]. The Huber loss also improves to 0.00020, reducing the error by 0.00020 compared to [28]. The MAE is 0.00420, indicating better accuracy. However, this performance comes at the cost of significantly increased model complexity, as the method uses 48,385 parameters. This large number of parameters reduces the overall efficiency, highlighting a tradeoff between performance and model complexity. Furthermore, the method in [16] achieves the lowest error among the previous methods, with an RMSE of 0.00290, MAE of 0.00190, and Huber loss of 0.00001. However, it uses 183,969 parameters, which is the largest among all methods. It indicates that the method in [16] requires significantly higher model complexity to achieve its competitive within the evaluated condition.




(a) (b)

**Figure 5. Comparison of actual and predicted values: (a) Baseline model showing higher deviation at peak regions; (b) The proposed mode demonstrating improved prediction alignment with reduced error**

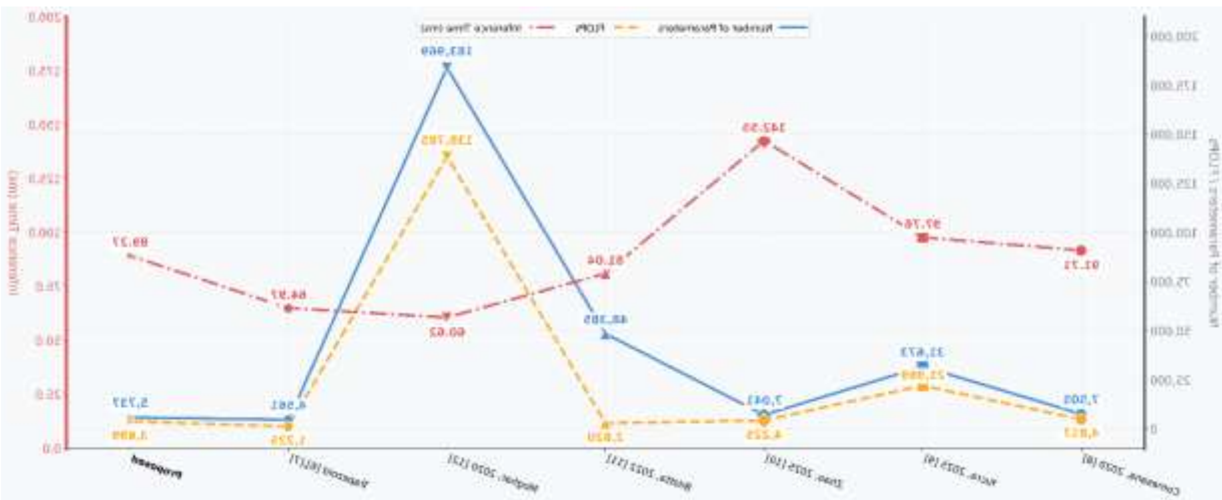


**Figure 6. Efficiency comparison of the proposed model against other methods**

Based on Table 6, the proposed model demonstrates competitive performance compared to the method in [16], while using only 5,737 parameters, the lowest among all methods. However, the RMSE is 0.00465, representing a slight increase of 0.00175 compared to [16]. Furthermore, the MAE is 0.00244, which is 0.00054 higher than [16], while the Huber loss remains the same at 0.00001. These results indicate that the proposed model achieves a better trade-off between performance and efficiency, maintaining comparable accuracy with significantly reduced model complexity. Notably, although the proposed model uses far fewer parameters, its error values remain on the same order of $10^{-3}$, similar to [16], indicating that the performance degradation is minimal.

Furthermore, Figure 5 illustrates the comparison between actual and predicted values for the two models across all data indices. In Figure 5(a), the baseline model captures the overall trend of the data. However, noticeable deviations occur at several peak points, indicating limited accuracy in modeling high-magnitude values. In contrast, Figure 5(b) shows that the improved model achieves closer alignment with the ground truth, particularly at critical peaks, with reduced overestimation and more consistent predictions across the sequence. Overall, these results demonstrate that the proposed model provides competitive within the evaluated condition predictive performance and stability compared to the baseline approach.

### 3.3. Efficiency Analysis

The efficiency comparison in Figure 7 evaluates the proposed model against several existing methods in terms of the number of parameters, FLOPs, and inference time. The work in [18] utilizes 7,505 parameters and requires 4,817 FLOPs, achieving an inference time of 91.71 ms. In addition, the work in [13] shows a significant increase in model complexity, with 31,673 parameters and 21,969 FLOPs, resulting in a slightly higher inference time of 97.76 ms. On the other hand, the work in [28] uses relatively fewer parameters, with 7,041 parameters and 4,225 FLOPs, yet records the highest inference time of 142.55 ms, indicating inefficiency in runtime performance. Furthermore, the work in [24] presents a different trade-off, with 48,385 parameters with only 2,820 FLOPs, achieving an inference time of 81.04 ms. The work in [16] has the largest model complexity, with 183,969 parameters and 138,785 FLOPs, yet achieves the fastest inference time of 60.62 ms, suggesting optimization at the hardware or implementation level. In contrast, the Trapezoid model in [15] and [14] demonstrates strong efficiency with only 4,561 parameters and 1,225 FLOPs, achieving an inference time of 64.97 ms. The proposed model maintains a relatively low complexity with 5,737 parameters and 3,699 FLOPs, while achieving an inference time of 89.27 ms. Compared to the Trapezoid model, the proposed model uses 1,176 additional parameters and 2,474 more FLOPs, with an increase in inference time of 24.30 ms. Compared to other methods, the proposed model offers a balanced trade-off between computational cost and runtime performance. It significantly reduces parameter size and FLOPs compared to the work in [13] and [16], while maintaining competitive inference speed. These results indicate that the proposed model achieves an efficient design, making it suitable for deployment in resource-constrained environments without significantly compromising performance.

### 3.4. Discussion

As shown in Figure 7, the LSTM-LA model is well suited to be implemented as a decision-support application, not only as a standalone prediction model. Because LSTM can learn temporal patterns and linear attention can focus on the most relevant signals, the model is appropriate for sequential marine data such as fish production, catch volume, sea surface temperature, chlorophyll, salinity, rainfall, tides, market demand, and vessel activity. In practice, the application can be designed as a predictive modeling and monitoring platform where users such as fisheries offices, aquaculture managers, harbor operators, and coastal planners upload historical data and receive future production forecasts, anomaly alerts, and seasonal trend analysis. Based on your reported efficiency profile, the model is compact enough to be deployed in a lightweight server-based application for routine prediction.

A practical implementation can be divided into three layers. The first is the data layer, which collects historical fisheries production records, water-quality sensor data, satellite-derived oceanographic variables, weather data, and operational records. The second is the model layer, where preprocessing, normalization, feature engineering, model training, and inference are performed using the trained LSTM-LA model. The third is the application layer, which delivers the results through a web dashboard or mobile interface. Based on Figure 7, users should be able to select location, species, and time range, then view outputs such as predicted production, expected high-risk periods, trend graphs, and warning notifications. For fisheries management, this can support production predictive modeling, fishing season prediction, early warning for low-yield periods, water-quality risk assessment, and resource planning. For aquaculture, the same application can be adapted to predict growth performance, feed demand, mortality risk, and harvest timing. From a software perspective, the most feasible implementation is to deploy the LSTM-LA model as a backend prediction service using FastAPI or Flask, while the frontend can be built with Streamlit, Dash, or a full web framework such as React. The backend receives new input data, applies the same preprocessing pipeline used during training, runs inference, and returns prediction results to the interface.

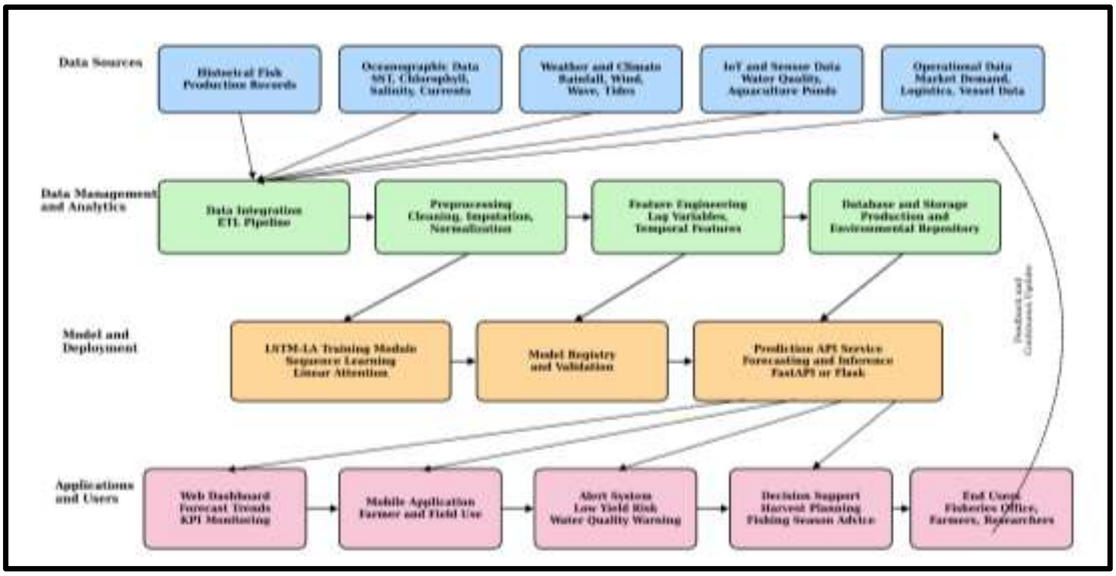



**Figure 7. Application Framework of LSTM-LA for Fisheries and Marine Data Science**

A database such as PostgreSQL can store historical production and environmental data, while scheduled jobs can update predictions daily, weekly, or monthly. For marine applications with geospatial components, the dashboard should include maps, zonation overlays, and temporal charts to help users interpret the forecast spatially and seasonally. The strongest application idea for your study would be a smart fisheries forecasting system with modules for production prediction, environmental trend monitoring, model performance tracking, and management recommendation output. This makes the model scientifically meaningful and operationally useful.

## 4. Conclusion

This study proposes an attention-based model to predict total fish production, supporting sustainable fisheries management and food security in Indonesia. By combining the temporal learning capability of LSTM with a linear attention mechanism, the proposed framework effectively captures sequential dependencies and improves feature weighting in fish production time-series data. Experimental results confirmed the predictive performance of the model, achieving an RMSE of 0.00465 and MAE of 0.00244, which indicate high predictive accuracy and strong agreement between predicted and observed values.

Beyond accuracy, the proposed model demonstrated a favorable trade-off between predictive performance and computational efficiency. With only 5,737 parameters, it remained substantially lighter than several comparative approaches while maintaining competitive results, highlighting its suitability for practical implementation in fisheries forecasting systems and resource-constrained environments. Overall, the findings show that integrating linear attention into an LSTM architecture provides an effective and efficient solution for total fish production forecasting. The proposed model offers a scientifically grounded basis for supporting evidence-based fisheries policies, quota regulation, and precautionary monitoring of fishery resources. Future studies should incorporate broader environmental and operational variables, extend validation across wider regions and species, and evaluate the framework in real-world decision-support applications to improve model generalizability and decision-support applicability.

## Declarations

*Author Contributions*

Conceptualization, Nancy Jeane Tuturoong and Muhamad Dwisnanto Putro.; methodology, Nancy Jeane Tuturoong.; software, Lusia Manu.; validation, Nancy Jeane Tuturoong., Muhamad Dwisnanto Putro. and Kawilarang Warouw Alex Masengi.; formal analysis, Nancy Jeane Tuturoong.; investigation, Nancy Jeane Tuturoong.; resources, Lusia Manu.; data curation, Muhammad Dwisnanto Putro.; writing—original draft preparation, Nancy Jeane Tuturoong.; writing—review and editing, Jimmy Reagen Robot.; visualization, Nancy Jeane Tuturoong.; supervision, Kawilarang Warouw Alex Masengi.; project administration, Nancy Jeane Tuturoong. All authors have read and agreed to the published version of the manuscript.

*Data Availability Statement*

The datasets analyzed during the current study are available from the corresponding author upon reasonable request.

*Funding*
Funding information is not available.

*Acknowledgements*
The authors would like to thank all parties who contributed to this research, either directly or indirectly, especially in data provision, technical support, and valuable discussions during the study.

*Institutional Review Board Statement*
Not applicable because this study did not involve human participants or animals and was conducted using secondary data and computational modellings.

*Informed Consent Statement*
*Not Applicable* for studies not involving humans.

*Conflicts of Interest*
The author declares that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

# References

[1] A. Baheramsyah, B. Cahyono, and Suganda, “Slurry Ice as a Cooling System on 30 GT Fishing Vessel,” International Journal of Marine Engineering Innovation and Research, vol. 1, no. 3, pp. 136–142, 2017, doi: 10.12962/J25481479.V1I3.
[2] F. T. C. Barreto, M. B. L. da Silva, M. C. de O. Costa, K. C. Lacerda, and C. L. da S. Junior, “Convolutional long short-term memory neural network for spatiotemporal forecasting of surface currents from HF-radar,” Ocean Model. (Oxf)., vol. 201, p. 102711, Apr. 2026, doi: 10.1016/J.OCEMOD.2026.102711.
[3] T. Marwa, Muizzuddin, A. Bashir, S. Andaiyani, and A. Cahyadi, “Determinants of the Blue Economy Growth in the Era of Sustainability: A Case Study of Indonesia,” Economies 2024, Vol. 12, Page 299, vol. 12, no. 11, p. 299, Nov. 2024, doi: 10.3390/ECONOMIES12110299.
[4] H. Setiyowati, M. Nugroho, and A. Halik, “Developing a Blue Economy in Depok West Java, Indonesia: Opportunities and Challenges of Neon Tetra Fish Cultivation,” Sustainability (Switzerland), vol. 14, no. 20, Oct. 2022, doi: 10.3390/SU142013028.
[5] J. P. Kritzer, F. Sun, D. Willard, J. Mimikakis, and J. Virdin, “Fisheries,” pp. 33–44, 2025, doi: 10.1007/978-981-19-4790-2_3.
[6] J. Kelautan, P. Terapan, E. Khusus, # Korespondensi, S. Arsitektur, and S. W. Trenggono, “PENANGKAPAN IKAN TERUKUR BERBASIS KUOTA UNTUK KEBERLANJUTAN SUMBER DAYA PERIKANAN DI INDONESIA,” Jurnal Kelautan dan Perikanan Terapan (JKPT), vol. 1, no. 0, pp. 1–8, Jan. 2023, doi: 10.15578/JKPT.V1I0.12057.
[7] M. M. H. Mozumder and P. Schneider, “Advancing sustainability through the circular economy in small-scale fisheries: A global review of practices, challenges, and policy innovations,” Mar. Policy, vol. 185, p. 107001, Mar. 2026, doi: 10.1016/J.MARPOL.2025.107001.
[8] I. Van-Deste, J. C. Lopes, and R. P. Lopes, “Memory vs. Attention: Investigating LSTM and Transformer Models in Human Action Recognition,” Procedia Comput. Sci., vol. 278, pp. 1299–1306, Jan. 2026, doi: 10.1016/J.PROCS.2026.03.113.
[9] J. Huang, W. Chen, and T. Zhang, “A linear-attention-combined convolutional neural network for EEG-based visual stimulus recognition,” Biocybern. Biomed. Eng., vol. 44, no. 2, pp. 369–379, Apr. 2024, doi: 10.1016/J.BBE.2024.05.001.
[10] N. Stacey et al., “Developing sustainable small-scale fisheries livelihoods in Indonesia: Trends, enabling and constraining factors, and future opportunities,” Mar. Policy, vol. 132, p. 104654, Oct. 2021, doi: 10.1016/J.MARPOL.2021.104654.
[11] A. Suherman, Y. Hernuryadin, P. Suadela, U. A. Furkon, and T. Amboro, “Transformation of Indonesian capture fisheries governance: Review and prospects,” Mar. Policy, vol. 174, p. 106619, Apr. 2025, doi: 10.1016/J.MARPOL.2025.106619.
[12] K. Wang, C. Zhang, Y. Ji, B. Xu, Y. Xue, and Y. Ren, “Navigating life-history parameter uncertainty in data-limited fisheries: a comparative analysis of length-based stock assessment methods,” Reviews in Fish Biology and Fisheries 2025 35:3, vol. 35, no. 3, pp. 1711–1733, Jul. 2025, doi: 10.1007/S11160-025-09979-Y.
[13] J. P. P. Merma Yucra, D. J. Cerezo Quina, G. A. Echaiz Espinoza, M. A. Valderrama Solis, D. D. Yanyachi Aco Cardenas, and A. Ortiz Salazar, “Design and Implementation of an LSTM Model with Embeddings on MCUs for Prediction of Meteorological Variables,” Sensors, vol. 25, no. 12, Jun. 2025, doi: 10.3390/s25123601.
[14] M. Alazab, S. Khan, S. S. R. Krishnan, Q. V. Pham, M. P. K. Reddy, and T. R. Gadekallu, “A Multidirectional LSTM Model for Predicting the Stability of a Smart Grid,” IEEE Access, vol. 8, pp. 85454–85463, 2020, doi: 10.1109/ACCESS.2020.2991067.
[15] D. G. Lui, G. Tartaglione, F. Conti, G. De Tommasi, and S. Santini, “Long Short-Term Memory-Based Neural Networks for Missile Maneuvers Trajectories Prediction∗,” IEEE Access, vol. 11, pp. 30819–30831, 2023, doi: 10.1109/ACCESS.2023.3262023.
[16] A. Moghar and M. Hamiche, “Stock Market Prediction Using LSTM Recurrent Neural Network,” in

Procedia Computer Science, Elsevier B.V., 2020, pp. 1168–1173. doi: 10.1016/j.procs.2020.03.049.
[17] Y. P. Huang and S. P. Khabusi, "Artificial Intelligence of Things (AIoT) Advances in Aquaculture: A Review," Processes 2025, Vol. 13, Page 73, vol. 13, no. 1, p. 73, Jan. 2025, doi: 10.3390/PR13010073.
[18] M. M. Comesaña, L. Febrero-Garrido, F. Troncoso-Pastoriza, and J. Martínez-Torres, "Prediction of building's thermal performance using LSTM and MLP neural networks," Applied Sciences (Switzerland), vol. 10, no. 21, pp. 1–16, Nov. 2020, doi: 10.3390/app10217439.
[19] A. Q. Wu et al., "Deep Learning for Sustainable Aquaculture: Opportunities and Challenges," Sustainability 2025, Vol. 17, Page 5084, vol. 17, no. 11, p. 5084, Jun. 2025, doi: 10.3390/SU17115084.
[20] Y. He and Q. Chen, "Construction and Application of LSTM-Based Prediction Model for Tunnel Surrounding Rock Deformation," Sustainability (Switzerland), vol. 15, no. 8, Apr. 2023, doi: 10.3390/su15086877.
[21] H. Xu, L. Song, Y. Li, T. Zhang, and J. Shen, "Comparison of LSTM and QR models for predicting CPUE of albacore tuna in waters near the Cook Islands," vol. 91, pp. 259–274, 2025, doi: 10.1007/s12562-025-01851-z.
[22] H. Xu et al., "Effects of Different Spatial Resolutions on Prediction Accuracy of Thunnus alalunga Fishing Ground in Waters Near the Cook Islands Based on Long Short-Term Memory (LSTM) Neural Network Model," Journal of Ocean University of China 2023 22:5, vol. 22, no. 5, pp. 1427–1438, Sep. 2023, doi: 10.1007/S11802-023-5525-5.
[23] H. Liu and L. Yang, "A Comparative Study of CNN-sLSTM-Attention-Based Time Series Forecasting: Performance Evaluation on Data with Symmetry and Asymmetry Phenomena," Symmetry (Basel)., vol. 17, no. 11, Nov. 2025, doi: 10.3390/sym17111846.
[24] S. Bilotta, E. Collini, P. Nesi, and G. Pantaleo, "Short-Term Prediction of City Traffic Flow via Convolutional Deep Learning," IEEE Access, vol. 10, pp. 113086–113099, 2022, doi: 10.1109/ACCESS.2022.3217240.
[25] N. González-Cancelas, J. Vaca-Cabrero, and A. Camarero-Orive, "The Role of the Fishing Sector in the Blue Economy: Prioritization, Environmental Challenges, and Sustainable Strategies in Europe, with a Focus on Spain," Journal of Marine Science and Engineering 2025, Vol. 13, Page 621, vol. 13, no. 3, p. 621, Mar. 2025, doi: 10.3390/JMSE13030621.
[26] W. Taparhudee et al., "An Attention-Based Hybrid CNN–Bidirectional LSTM Model for Classifying Chlorophyll-a Concentration in Coastal Waters," Water 2026, Vol. 18, Page 33, vol. 18, no. 1, p. 33, Dec. 2025, doi: 10.3390/W18010033.
[27] Y. Zhang, M. Yamamoto, G. Suzuki, and H. Shioya, "Collaborative Forecasting and Analysis of Fish Catch in Hokkaido From Multiple Scales by Using Neural Network and ARIMA Model," IEEE Access, vol. 10, pp. 7823–7833, 2022, doi: 10.1109/ACCESS.2022.3141767.
[28] Y. Zhao, C. S. Lim, F. Xue, C. Long, and A. H. P. Tan, "Lightweight LSTM and GRU Design for Data-Driven Rotor Position Error Estimation in IPMSM Drives," IEEE Open Journal of the Industrial Electronics Society, vol. 6, pp. 851–867, 2025, doi: 10.1109/OJIES.2025.3571204.
[29] H. Xu et al., "Effects of Different Spatial Resolutions on Prediction Accuracy of Thunnus alalunga Fishing Ground in Waters Near the Cook Islands Based on Long Short-Term Memory (LSTM) Neural Network Model," Journal of Ocean University of China, vol. 22, no. 5, pp. 1427–1438, Oct. 2023, doi: 10.1007/s11802-023-5525-5.
[30] W. Wirata et al., "Development of a Deep Learning Based Price Prediction Model for Katsuwonus pelamis at Nizam Zachman Fishing Port Jakarta Indonesia," Thalassas: An International Journal of Marine Sciences 2025 41:2, vol. 41, no. 2, pp. 69-, Mar. 2025, doi: 10.1007/S41208-025-00822-6.
[31] S. Bilotta, E. Collini, P. Nesi, and G. Pantaleo, "Short-Term Prediction of City Traffic Flow via Convolutional Deep Learning," IEEE Access, vol. 10, pp. 113086–113099, 2022, doi: 10.1109/ACCESS.2022.3217240.

## 参考文:

[1] A. Baheramsyah、B. Cahyono 和 Suganda，"30 GT 渔船上作为冷却系统的浆冰"，International Journal of Marine Engineering Innovation and Research，第 1 卷，第 3 期，第 136–142 页，2017，doi: 10.12962/J25481479.V1I3。
[2] F. T. C. Barreto、M. B. L. da Silva、M. C. de O. Costa、K. C. Lacerda 和 C. L. da S. Junior，"用于基于 HF 雷达的表层海流时空预测的卷积长短期记忆神经网络"，Ocean Model. (Oxf).，第 201 卷，第 102711 页，2026 年 4 月，doi: 10.1016/J.OCEMOD.2026.102711。
[3] T. Marwa、Muizzuddin、A. Bashir、S. Andaiyani 和 A. Cahyadi，"可持续发展时代蓝色经济增长的决定因素：印度尼西亚案例研究"，Economies 2024, Vol. 12, Page 299，第 12 卷，第 11 期，第 299 页，2024 年 11 月，doi: 10.3390/ECONOMIES12110299。
[4] H. Setiyowati、M. Nugroho 和 A. Halik，"印度尼西亚西爪哇德波蓝色经济的发展：霓虹灯鱼养殖的

机遇与挑战”，Sustainability (Switzerland)，第 14 卷，第 20 期，2022 年 10 月，doi: 10.3390/SU142013028。
[5] J. P. Kritzer、F. Sun、D. Willard、J. Mimikakis 和 J. Virdin，“渔业”，第 33–44 页，2025，doi: 10.1007/978-981-19-4790-2_3。
[6] J. Kelautan、P. Terapan、E. Khusus、# Korespondensi、S. Arsitektur 和 S. W. Trenggono，“基于配额的可衡量捕捞以实现印度尼西亚渔业资源可持续性”，Jurnal Kelautan dan Perikanan Terapan (JKPT)，第 1 卷，第 0 期，第 1–8 页，2023 年 1 月，doi: 10.15578/JKPT.V1I0.12057。
[7] M. M. H. Mozumder 和 P. Schneider，“通过小规模渔业中的循环经济推进可持续性：实践、挑战与政策创新的全球综述”，Mar. Policy，第 185 卷，第 107001 页，2026 年 3 月，doi: 10.1016/J.MARPOL.2025.107001。
[8] I. Van-Deste、J. C. Lopes 和 R. P. Lopes，“记忆与注意力：在人类动作识别中研究 LSTM 和 Transformer 模型”，Procedia Comput. Sci.，第 278 卷，第 1299–1306 页，2026 年 1 月，doi: 10.1016/J.PROCS.2026.03.113。
[9] J. Huang、W. Chen 和 T. Zhang，“一种结合线性注意力的卷积神经网络用于基于 EEG 的视觉刺激识别”，Biocybern. Biomed. Eng.，第 44 卷，第 2 期，第 369–379 页，2024 年 4 月，doi: 10.1016/J.BBE.2024.05.001。
[10] N. Stacey 等，“发展印度尼西亚可持续小规模渔业生计：趋势、促进与制约因素及未来机遇”，Mar. Policy，第 132 卷，第 104654 页，2021 年 10 月，doi: 10.1016/J.MARPOL.2021.104654。
[11] A. Suherman、Y. Hernuryadin、P. Suadela、U. A. Furkon 和 T. Amboro，“印度尼西亚捕捞渔业治理转型：回顾与展望”，Mar. Policy，第 174 卷，第 106619 页，2025 年 4 月，doi: 10.1016/J.MARPOL.2025.106619。
[12] K. Wang、C. Zhang、Y. Ji、B. Xu、Y. Xue 和 Y. Ren，“在数据有限渔业中应对生活史参数不确定性：基于长度的资源评估方法比较分析”，Reviews in Fish Biology and Fisheries 2025 35:3，第 35 卷，第 3 期，第 1711–1733 页，2025 年 7 月，doi: 10.1007/S11160-025-09979-Y。
[13] J. P. P. Merma Yucra、D. J. Cerezo Quina、G. A. Echaiz Espinoza、M. A. Valderrama Solis、D. D. Yanyachi Aco Cardenas 和 A. Ortiz Salazar，“在微控制器上设计并实现带嵌入的 LSTM 模型用于气象变量预测”，Sensors，第 25 卷，第 12 期，2025 年 6 月，doi: 10.3390/s25123601。
[14] M. Alazab、S. Khan、S. S. R. Krishnan、Q. V. Pham、M. P. K. Reddy 和 T. R. Gadekallu，“用于预测智能电网稳定性的多方向 LSTM 模型”，IEEE Access，第 8 卷，第 85454–85463 页，2020，doi: 10.1109/ACCESS.2020.2991067。
[15] D. G. Lui、G. Tartaglione、F. Conti、G. De Tommasi 和 S. Santini，“基于长短期记忆的神经网络用于导弹机动轨迹预测”，IEEE Access，第 11 卷，第 30819–30831 页，2023，doi: 10.1109/ACCESS.2023.3262023。
[16] A. Moghar 和 M. Hamiche，“使用 LSTM 循环神经网络进行股票市场预测”，载于 Procedia Computer Science，Elsevier B.V.，2020，第 1168–1173 页，doi: 10.1016/j.procs.2020.03.049。
[17] Y. P. Huang 和 S. P. Khabusi，“水产养殖中人工智能物联网（AIoT）的进展：综述”，Processes 2025, Vol. 13, Page 73，第 13 卷，第 1 期，第 73 页，2025 年 1 月，doi: 10.3390/PR13010073。
[18] M. M. Comesaña、L. Febrero-Garrido、F. Troncoso-Pastoriza 和 J. Martínez-Torres，“使用 LSTM 和 MLP 神经网络预测建筑热性能”，Applied Sciences (Switzerland)，第 10 卷，第 21 期，第 1–16 页，2020 年 11 月，doi: 10.3390/app10217439。
[19] A. Q. Wu 等，“可持续水产养殖中的深度学习：机遇与挑战”，Sustainability 2025, Vol. 17, Page 5084，第 17 卷，第 11 期，第 5084 页，2025 年 6 月，doi: 10.3390/SU17115084。
[20] Y. He 和 Q. Chen，“基于 LSTM 的隧道围岩变形预测模型构建与应用”，Sustainability (Switzerland)，第 15 卷，第 8 期，2023 年 4 月，doi: 10.3390/su15086877。
[21] H. Xu、L. Song、Y. Li、T. Zhang 和 J. Shen，“用于预测库克群岛附近海域长鳍金枪鱼 CPUE 的 LSTM 与 QR 模型比较”，第 91 卷，第 259–274 页，2025，doi: 10.1007/s12562-025-01851-z。
[22] H. Xu 等，“基于长短期记忆（LSTM）神经网

络模型的不同空间分辨率对库克群岛附近海域长鳍金枪鱼渔场预测精度的影响”，Journal of Ocean University of China 2023 22:5，第 22 卷，第 5 期，第 1427–1438 页，2023 年 9 月，doi: 10.1007/S11802-023-5525-5。

[23] H. Liu 和 L. Yang，“基于 CNN-sLSTM-Attention 的时间序列预测比较研究：对具有对称与非对称现象数据的性能评估”，Symmetry (Basel).，第 17 卷，第 11 期，2025 年 11 月，doi: 10.3390/sym17111846。

[24] S. Bilotta、E. Collini、P. Nesi 和 G. Pantaleo，“基于卷积深度学习的城市交通流短期预测”，IEEE Access，第 10 卷，第 113086–113099 页，2022，doi: 10.1109/ACCESS.2022.3217240。

[25] N. González-Cancelas、J. Vaca-Cabrero 和 A. Camarero-Orive，“渔业部门在蓝色经济中的作用：欧洲的优先排序、环境挑战与可持续战略——以西班牙为重点”，Journal of Marine Science and Engineering 2025, Vol. 13, Page 621，第 13 卷，第 3 期，第 621 页，2025 年 3 月，doi: 10.3390/JMSE13030621。

[26] W. Taparhudee 等，“一种基于注意力机制的混合 CNN–双向 LSTM 模型用于近岸水域叶绿素-a 浓度分类”，Water 2026, Vol. 18, Page 33，第 18 卷，第 1 期，第 33 页，2025 年 12 月，doi: 10.3390/W18010033。

[27] Y. Zhang、M. Yamamoto、G. Suzuki 和 H. Shioya，“利用神经网络和 ARIMA 模型从多尺度进行北海道鱼类捕捞量协同预测与分析”，IEEE Access，第 10 卷，第 7823–7833 页，2022，doi: 10.1109/ACCESS.2022.3141767。

[28] Y. Zhao、C. S. Lim、F. Xue、C. Long 和 A. H. P. Tan，“用于 IPMSM 驱动中数据驱动转子位置误差估计的轻量级 LSTM 和 GRU 设计”，IEEE Open Journal of the Industrial Electronics Society，第 6 卷，第 851–867 页，2025，doi: 10.1109/OJIES.2025.3571204。

[29] H. Xu 等，“基于长短期记忆（LSTM）神经网络模型的不同空间分辨率对库克群岛附近海域长鳍金枪鱼渔场预测精度的影响”，Journal of Ocean University of China，第 22 卷，第 5 期，第 1427–1438 页，2023 年 10 月，doi: 10.1007/s11802-023-5525-5。

[30] W. Wirata 等，“印度尼西亚雅加达 Nizam Zachman 渔港鲣鱼深度学习价格预测模型的开发”，Thalassas: An International Journal of Marine Sciences 2025 41:2，第 41 卷，第 2 期，第 69 页起，2025 年 3 月，doi: 10.1007/S41208-025-00822-6。

[31] S. Bilotta、E. Collini、P. Nesi 和 G. Pantaleo，“基于卷积深度学习的城市交通流短期预测”，IEEE Access，第 10 卷，第 113086–113099 页，2022，doi: 10.1109/ACCESS.2022.3217240。

**Manuscript Information**
Word count: 10,135 words (excluding references).

**Peer-Review Record**
Fast-track status: Not fast-tracked.
First-round reviews received: 3 reports.
Revision cycles completed: 3 rounds.
Final version submitted: May 20, 2026