

# An Improved LSTM-Based Data Popularity Classification and Hierarchical Storage Optimization Strategy for Industrial Internet of Things

Yuan Yuan, Meng-Ying Yang, Xiao-Jing Guo, and Cheng-Fang Mei\*

Hebei Institute of Mechanical and Electrical Technology,  
Xingtai City 054000, Hebei Province, China

{yuanyuanxinxi, yangmengying, guoxiaojing, meichengfang}@hbjd.edu.cn

*Received 2 November 2025; Revised 4 December 2025; Accepted 12 December 2025*

**Abstract.** The rapid development of the Industrial Internet of Things (IIoT) has generated massive amounts of heterogeneous data with diverse access patterns and storage requirements, posing severe challenges to the latency, reliability, and energy efficiency of industrial automation systems. To address these challenges, this study proposes a data popularity classification model based on an improved long short-term memory (LSTM) model, combined with a tiered storage optimization strategy. The enhanced LSTM, incorporating attention and residual mechanisms, better captures temporal dynamics and enhances generalization under fluctuating workloads. The predicted popularity level is used to guide an adaptive multi-tiered storage allocation framework across edge, fog, and cloud layers, achieving a balance between responsiveness and resource utilization. Experimental validation demonstrates that this approach improves prediction stability and storage efficiency compared to traditional models, providing a scalable and adaptive solution for intelligent data management in next-generation IIoT environments.

**Keywords:** industrial internet of things (IIoT), LSTM, data popularity classification, hierarchical storage, edge computing, optimization strategy

## 1 Introduction

With the rapid advancement of industrial digitalization, the Industrial Internet of Things (IIoT) has become a core infrastructure connecting sensors, controllers, and production equipment throughout manufacturing systems [1]. Through continuous data collection and interaction, the IIoT enables real-time monitoring, predictive maintenance, and intelligent decision-making in modern factories. However, the massive amount of data, from high-frequency sensor data streams to periodic maintenance logs, poses significant challenges to data management and system performance [2]. Not all data has the same value or access frequency; some information is repeatedly requested in control loops, while other data lies dormant until needed for auditing or analysis. This imbalance leads to poor storage utilization, increased access latency, and increased energy consumption [3]. Therefore, an effective mechanism that can identify data popularity or hotness and dynamically allocate data across storage tiers is crucial to maintaining the efficiency and scalability of IIoT systems. Despite growing research on IIoT data management, existing approaches still have key shortcomings. Traditional time series analysis models, such as the auto-regressive moving average (ARIMA) [4] method and exponential smoothing, are limited in their ability to capture the nonlinear temporal relationships inherent in industrial data. Deep learning models, particularly traditional recurrent neural networks (RNNs) and standard long short-term memory (LSTM) architectures, can learn sequential dependencies but often struggle to cope with fluctuating workloads and noisy industrial environments [5]. Furthermore, most previous studies treat data popularity prediction and storage optimization as independent problems: predictive models output frequency trends without considering how these trends influence subsequent storage allocation, while storage policies often rely on fixed thresholds that ignore dynamic data behavior. As a result, existing systems often suffer from poor cache performance, unbalanced loads across storage tiers, and decreased responsiveness in real-time industrial environments [6]. This study aims to bridge the gap between data popularity prediction and tiered storage optimization by introducing a unified adaptive framework. Its primary goal is to improve data management efficiency in IIoT environments through intelligent prediction and resource allocation mechanisms. The paper is divided into six parts. Section 1 explains the challenges posed by

---

\* Corresponding Author

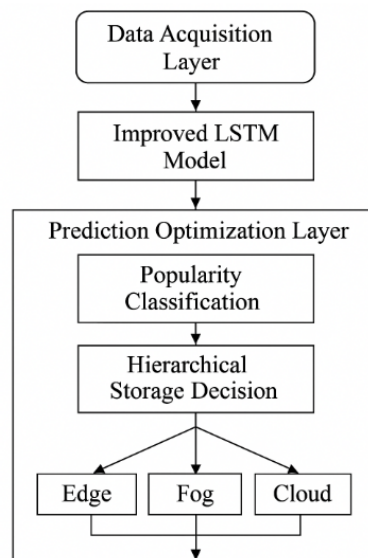
the vast and unevenly used data of the Industrial Internet of Things (IIoT), as well as the limitations of treating popularity prediction and storage management as separate tasks. Based on this, an integrated framework for prediction and storage is proposed. Section 2 provides an overview of existing LSTM-based deep learning models, resource-efficient analytics, and edge caching strategies in IIoT, highlighting the lack of coordinated optimization between prediction and tiered storage. Section 3 presents the proposed framework and details how an improved LSTM with residual attention predicts data popularity and how these predictions impact dynamic tiered storage decisions in edge, fog, and cloud environments. Section 4 describes the datasets, preprocessing steps, simulated industrial deployment environments, and reference models used for rigorous and realistic evaluation. Section 5 presents empirical results demonstrating the superior performance of the proposed framework in terms of prediction accuracy, stability, latency reduction, and energy efficiency. The results are obtained using loss curves, prediction tracking, basis error comparison, component ablation, and latency-energy analysis. Section 6 summarizes the key findings of this study and describes a scalable path to intelligent, self-optimizing Industrial Internet of Things data management systems through an advanced LSTM architecture, tight integration of prediction and hierarchical memory optimization, and comprehensive validation.

## 2 Related Work

Deep learning techniques, especially the integration of LSTM networks, have attracted considerable attention in the context of the IIoT. Several studies have highlighted the potential of LSTM-based models for data classification tasks relevant to industrial environments. For example, Dugat-LSTM demonstrated the effectiveness of a deep learning-based network intrusion detection system, leveraging a chaotic optimization strategy to improve the performance of industrial IoT networks [7]. This approach highlights the importance of complex neural network architectures in accurately identifying anomalies and security threats in complex industrial data streams. Furthermore, the application of deep learning has been extended to ensemble stacking methods, which have been used to improve intrusion detection accuracy on IoT sensor datasets. This ensemble technique utilizes multiple models to achieve more robust classification results, demonstrating the versatility of deep learning frameworks in processing a variety of industrial data types [8]. Emphasis on classification accuracy is crucial for developing reliable security and data management systems in IIoT environments. In addition to classification, the literature also emphasizes the role of data caching and storage optimization strategies tailored for IoT edge computing. Zhang et al. [9] emphasized the use of software-defined networking (SDN) for collaborative data caching, which is crucial for managing the massive amounts of data generated by industrial sensors. Efficient caching mechanisms are crucial for reducing latency and bandwidth consumption, thereby supporting the real-time data processing requirements in IIoT systems. Furthermore, integrating AI and deep learning into IIoT frameworks requires lightweight and resource-efficient models, especially given the limited functionality of many industrial devices. The lightweight framework proposed by Wang et al. [10] uses novel feature optimization strategies and adaptive cloud-edge intelligence to balance accuracy and resource constraints. These strategies are crucial for deploying deep learning models such as LSTM in real-world industrial environments where computing resources are often limited. An overarching theme of these studies is the need for intelligent, hierarchical storage and data classification strategies to adapt to the dynamic and large-scale nature of IIoT data. Deep learning models, especially LSTM networks, are positioned as core tools for achieving high-accuracy classification, which in turn provides information for optimizing storage and caching solutions [11]. These integrated approaches aim to enhance the security, efficiency, and real-time responsiveness of IIoT systems, which is consistent with the broader goal of developing improved LSTM-based data popularity classification and hierarchical storage optimization strategies [12]. In summary, the reviewed literature emphasizes the effectiveness of advanced deep learning models in IIoT data classification. Combined with innovative caching and storage strategies, these models can help build more secure, efficient, and scalable Industrial IoT infrastructure. Continued research in this area is expected to improve tiered storage solutions and further leverage the capabilities of LSTM for industrial data management. However, despite these advances, existing research largely treats predictive tasks and storage management as separate research areas. Very few studies have established a unified process enabling predictive modeling to directly drive tiered storage decisions in real time. Furthermore, previous research has often overlooked the temporal volatility and burstiness of industrial IoT workloads, resulting in gaps in how deep learning models adaptively guide fine-grained data placement across edge, fog, and cloud environments. This disconnect between predictive intelligence and operational storage optimization constitutes a critical research gap, which this study aims to address.

### 3 Methodology

Text information is divided into two categories, one is used to train models, known as the training set; The other type is test data, called test sets in Fig. 1. The proposed IIoT data management framework consists of three inter-related layers: data acquisition, prediction, and storage optimization. In the first stage, data streams generated by industrial sensors and controllers are collected, preprocessed, and transmitted to the prediction module [13]. This data typically contains time-series access records, device identifiers, and operational context. In the second stage, the prediction layer employs a modified LSTM model enhanced with an attention mechanism and residual connections [14]. This model predicts the future popularity of each data block based on historical access sequences, device behavior, and temporal trends. In the final stage, the storage optimization layer receives the predicted popularity scores and makes tiered storage decisions [15]. Data classified as hot, warm, or cold is dynamically allocated across edge, fog, and cloud nodes, respectively, balancing latency, capacity, and energy efficiency. This three-layer process forms a closed-loop adaptive cycle that continuously optimizes storage allocation based on changing data access patterns, ensuring real-time responsiveness and optimizing system resource utilization.



**Fig. 1.** Framework of the improved LSTM-Based data popularity prediction and hierarchical storage optimization model

#### 3.1 Improved LSTM Popularity Prediction

The access history of each data block is treated as a multivariate sequence, where the primary signal is past request counts or inter-arrival times, and covariates include workload identifiers, time of day, day of week, cache tier, queue depth, and device health indicators. These streams are normalized and bundled into fixed windows, which feed a recurrent backbone network consisting of LSTM units, enhanced with an attention mechanism and residual jumps between the input and output of the recurrent stack to stabilize long horizons. Within each LSTM step, three gates regulate how new evidence updates the memory. The forget gate generates a weight between 0 and 1 for each dimension of the previous cell state, softly determining how much previous context to retain. This is crucial when workload state changes and yesterday's bursts should be ignored. The input gate controls the extraction of new information by adjusting the candidate content vector computed based on the current window's features to capture only important patterns. During training, the network learns gating behavior consistent with the workload dynamics: setting a higher forget value during stable periods to maintain seasonality, lowering it at transition points to purge stale context, shrinking the input gate when noise rises, and expanding the output gate when strong predictors emerge. The resulting representation supports accurate predictions of both short-term visit counts and long-term popularity scores, and the learned attention weights and gate activations provide interpret-able signals about why a block is predicted to be hot or cold [16]. The forget gate, input gate, and output gate determine how new evidence modifies the memory:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (6)$$

Equations (1)-(6) are standard, but their roles are critical in IIoT: retains slow, cyclical patterns, while admits abrupt bursts; the combination in (4) lets the cell state carry both regimes without saturating. To emphasize time points that best explain near-future access, the team introduce a content-based attention over the hidden trajectory [17]:

$$s_k = v^\top \tanh(W_h h_k + W_c C_k + b_s) \quad (7)$$

$$\alpha_k = \frac{e^{s_k}}{\sum_{k=1}^t e^{s_k}} \quad (8)$$

$$c_t = \sum_{k=1}^t \alpha_k h_k \quad (9)$$

Here,  $s_k$  scores how explanatory step  $k$  is for time  $t$ ; the normalization ensures  $\alpha_k$  is a probability simplex, which aids interpret-ability. Rather than replacing  $h_t$  with  $c_t$ , the team add them:

$$h_t^{res} = h_t + c_t \quad (10)$$

This preserves the base recurrence while injecting global context; empirically it shortens training transients and limits gradient attenuation on long sequences. IIoT workloads drift; the team discount stale samples via an exponential time decay and penalize over-complex models:

$$L(\theta) = \frac{1}{N} \sum_{t=1}^N w_t (y_t - \hat{y}_t)^2 + \lambda \|\theta\|_2^2 \quad (11)$$

To make scores comparable across blocks with different historical ranges, the team apply min-max normalization:

$$P_i = \frac{P_i - P_{min}}{P_{max} - P_{min}} \quad (12)$$

Where  $P$  is the raw predicted popularity for block and  $P_{max}$   $P_{min}$  are rolling bounds. This avoids pathological allocations caused by unit inconsistencies. For storage actions, continuous popularity is discretized by robust quantiles to resist outliers:

$$H_i = \begin{cases} hot & P_i > P_{90} \\ warm & P_{50} \leq P_i \leq P_{90} \\ cold & P_i < P_{50} \end{cases} \quad (13)$$

Quantile thresholds adapt to workload level automatically; a sudden factory-wide surge shifts upward, preventing edge tiers from being flooded.

### 3.2 Hierarchical Storage Optimization

The team minimize a weighted sum of latency and energy:

$$\min_x J = \alpha D + \beta E, \alpha, \beta > 0, \alpha + \beta = 1 \quad (14)$$

Weights reflect site policy. For item  $i$  on tier  $j$ , the per-request delay combines queuing, service, and network propagation [18]:

$$d_{ij} = q_j(\rho_j) + s_{ij} + r_{ij} \quad (15)$$

$q_j(\rho_j)$  is a monotone function of utilization. Per-request energy combines static draw and dynamic work:

$$e_{ij} = e_j^{idle} + \kappa_j (b_i)^n \quad (16)$$

$e_j^{idle}$  is the amortized idle cost per request at tier  $j$ ,  $\kappa_j$  a conversion factor,  $b_i$  the bytes moved. The instantaneous sensitivity of the objective to reassigning item  $i$  from  $j$  to  $k$  is:

$$\Delta J_{i:j \rightarrow k} = \alpha \lambda_i (d_{ik} - d_{ij}) + \beta \lambda_i (e_{ik} - e_{ij}) + \zeta (\pi_{ik} - \pi_{ij}) + (v_k - v_j) s_i, \quad (17)$$

Which guides online migration decisions: move only when  $\Delta J_{i:j \rightarrow k}$  and rate-limit to avoid thrashing. The team maintain  $\rho_j$

$$\rho_j = \frac{1}{\mu_j} \sum_i \lambda_i x_{ij} \quad (18)$$

When  $\hat{\lambda}$  is not directly observed, the team map popularity to expected request rate via a monotone calibration

$$\hat{\lambda}_i = \lambda_0 \exp(\beta_0 + \beta_1 P_i), \beta_1 > 0 \quad (19)$$

Providing a consistent bridge between the predictor and the planner. For newly created items with little history, the team blend a prior with the model output [19]:

$$\hat{P}_i = \omega_i P_i + (1 - \omega_i) P \quad (20)$$

$$\omega_i = \frac{n_i}{n_i + n_0} \quad (21)$$

Where  $n_i$  counts observations; this avoids over-reacting to a single incidental access. To avoid jitter, migrations employ a threshold and hysteresis strategy, triggering only when the expected benefits consistently exceed the migration costs. To ensure quality of service, background migrations are rate-limited using a token bucket, and foreground requests are always prioritized. Each layer's capacity is soft-constrained using shadow prices, and the planner automatically suppresses upward migrations and relaxes downward migrations when approaching full capacity. The calibration function is monotonic and performs equal coverage checks on the validation set to ensure a consistent and reliable mapping from forecasts to plans. All key metrics are updated and logged on a rolling basis to support auditing and offline replay. To further enhance the coupling between prediction accuracy

and storage decision-making, we additionally introduce a stability sensitivity regularizer that quantitatively balances prediction smoothness with responsiveness to workload bursts. Specifically, let denote the predicted popularity at time  $t$ . We impose a temporal smoothness constraint:

$$L_{smooth} = \lambda_s \sum_{t=2}^T (\hat{p}_t - p_{t-1})^2 \quad (22)$$

This approach reduces excessive fluctuations caused by sensor noise while preserving reasonable trend changes. Simultaneously, to ensure rapid adaptation to sudden spikes in demand common in Industrial IoT data, the team introduced a change point amplification term:

$$L_{burst} = \lambda_b \sum_{t=2}^T \sigma(p_t - p_{t-1}) \cdot (\hat{p}_t - p_{t-1}) \quad (23)$$

Here,  $\sigma$  is a soft attention sigmoid function that enhances the gradient contribution of real burst events. This allows the model to emphasize structurally meaningful transitions while ignoring fluctuations caused by noise. On the optimization side, tier-assignment decisions additionally incorporate a probabilistic confidence factor derived from predictive entropy:

$$c_t = 1 - H(a_t) \quad (24)$$

where  $H$  is the Shannon entropy of the attention weights  $a_t$ . Higher attention concentration yields higher confidence, leading to more aggressive upward migration of data blocks.

## 4 Experimental Setup

This study's experimental design combines robust data selection, realistic simulations, and comprehensive bench-marking to evaluate the effectiveness of an improved LSTM-based prediction and storage optimization framework for industrial IoT systems. To ensure its generalizability and industry relevance, the team used two complementary datasets: one representing a controlled benchmark environment and the other reflecting real-world operating conditions. All data underwent standard preprocessing to eliminate inconsistencies and normalize heterogeneous sources, ensuring comparability across experiments. The team simulated a layered edge-fog-cloud architecture to emulate real-world industrial deployment scenarios, with varying computational capabilities and communication constraints across the layers. The improved LSTM model was trained under controlled conditions using a consistent optimization procedure, including adaptive learning, early stopping, and systematic parameter tuning to balance accuracy and stability. To evaluate the robustness of the approach, the team implemented a diverse set of baseline models spanning classical and deep learning paradigms: ARIMA, LSTM, GRU, Bi-LSTM, and CNN-LSTM. Comparative results highlight the advantages of the proposed attention-residual long short-term memory model in capturing dynamic temporal dependencies while maintaining computational efficiency. Traditional models such as ARIMA cannot adapt to nonlinear or time-varying patterns, while Bi-LSTM, despite its contextual accuracy, suffers from latency and resource inefficiency in real-time scenarios. Overall, the experimental framework demonstrates that the proposed model achieves an excellent balance between predictive accuracy, adaptability, and operational practicality, making it well-suited for intelligent data management in Industrial IoT applications [20]. Furthermore, integrating prediction and storage decisions into the same evaluation process allows for a deeper understanding of how prediction accuracy translates into system-level benefits, such as reduced latency and improved energy efficiency. These findings collectively validate the robustness of the proposed framework and highlight its potential for large-scale deployment in next-generation industrial IoT infrastructure.

#### 4.1 Datasets

To evaluate the effectiveness and robustness of the improved LSTM-based data popularity prediction and tiered storage optimization framework proposed in this paper, the team used two complementary datasets: a public benchmark data-set and an industrial data-set in Table 1. The first data-set is the Edge-IIoTset benchmark data-set, a well-known open-source resource designed for evaluating Industrial IoT applications. It contains multi-modal data collected from heterogeneous sensors, controllers, and network nodes under various operating conditions. Each record includes a timestamped measurement value, device identifier, communication frequency, power consumption, and contextual tags indicating the system load status. This benchmark data-set provides a controlled and diverse environment for validating the accuracy and scalability of the predictions. The second data-set consists of real-world factory operation logs collected from a smart manufacturing production line in an electrical equipment factory. This data contains approximately 2 million timestamped access records generated by 200 different sensors and controllers distributed across assembly and quality inspection departments. Each log entry includes access frequency, data type identifier, payload size, latency record, and storage tier. These logs were collected continuously over a three-month period to ensure that they capture the typical periodic and bursty access behavior in real-world Industrial IoT environments. Before model training, the raw data underwent multiple preprocessing steps. Missing values are filled using forward temporal interpolation, and redundant or corrupted entries due to communication failures are filtered using checksum validation. Each sensor stream is normalized using a z-score transformation, and the categorical identifier is embedded in a low-dimensional vector. The final data-set is chronologically divided into training, validation, and test segments to maintain temporal order. To simulate realistic dynamic behavior, additional synthetic bursts are injected into the training set based on a Poisson arrival model. This enhancement increases the variability of access frequencies and improves the model's ability to adapt to sudden demand changes. Overall, the combined data-set provides a comprehensive mix of steady-state, non-steady-state, and bursty temporal dynamics, which is critical for testing predictions and optimizing performance. Beyond the basic descriptions above, the Edge-IIoTset benchmark dataset is further optimized by selecting representative subsets of devices and operating modes and aggregating raw measurement data into fixed-length time windows corresponding to the LSTM prediction range. For each window, the research team extracted device popularity features, such as short-term access counts, moving averages, and volatility metrics, as well as contextual variables like device roles and network segments. This achieves a unified representation of heterogeneous nodes. In the industrial dataset, sensors are grouped into functional clusters with different access frequencies and load characteristics. This heterogeneity is preserved rather than reduced, allowing the model to explicitly consider the differences between production lines common in multi-cell factories. To avoid information loss, time segments are date-based: early data is used for training, intermediate data for hyperparameter optimization, and data from recent weeks for final testing. This simulates the scenario where future data is unavailable during training in real-world applications. Z-score normalization for each sensor data stream is calculated solely based on statistics from the training dataset. The learned parameters are reused in both the validation and test sets to ensure simulation of realistic online environments. For the synthetic access peaks based on the Poisson distribution, the arrival rate was calibrated to match the empirical upper bound of the observed access distribution. This ensures that the inserted peaks effectively simulate real-world workload behavior, rather than artificially introduced outliers. These steps collectively ensure that both datasets cover a wide range of operating conditions—from controlled benchmarks to noisy factory tests—thus laying a solid foundation for evaluating the generalization and robustness of the proposed popularity prediction and hierarchical storage framework [21].

**Table 1.** Data-set description and preprocessing procedures for model evaluation

Item	Description
Benchmark dataset	Publicly available Industrial IoT benchmark data-set designed for evaluating heterogeneous sensor environments and verifying model scalability under controlled conditions.
Industrial dataset	Real-world operational logs collected from a smart manufacturing production line, representing practical variability and bursty access patterns.
Preprocessing	Missing-value interpolation, redundancy filtering, z-score normalization, categorical embedding, and temporal segmentation.
Synthetic dynamics	Additional Poisson-distributed bursts injected into the training set to simulate irregular workload spikes and enhance model adaptability.

## 4.2 Experimental Environment

All experiments were conducted on a layered edge-fog-cloud testbed, which simulates industrial IoT deployments at shop floor scale and above, with strict controls over compute, networking, and software to ensure repeatability and end-to-end audit ability. The edge layer consists of lightweight nodes with limited cores and fast local storage, emulating embedded controllers and gateways near production lines. The fog layer comprises intermediate servers that aggregate data streams from multiple edge nodes and perform low-latency filtering, featurization, and batch inference. The cloud layer consists of high-capacity virtual machines for long-term storage, historical analysis, model training, and registration. To quantify temporal behavior across tiers, the end-to-end latency can be decomposed into transmission, queuing, and processing components as:

$$L_{e2e} = L_{tx} + L_{queue} + L_{proc} \quad (25)$$

The entire stack uses Docker containers, with each service having resource quotas and fixed CPU and memory limits to simulate constrained hardware while maintaining isolation between experiments. Orchestration is defined using versioned Compose files, ensuring that each run maintains a precise mapping of services, images, digests, ports, and environment variables. System time across containers is synchronized using Network Time Protocol (NTP), and all containers record their build origin, git commit, and configuration hash for easy traceability. Inter-layer messaging adheres to the Message Queuing Telemetry Transport Protocol (MQTTP), which features well-defined quality of service levels, retained messages on control topics, and compact binary payloads carrying sensor values, monotonic timestamps, and sequence numbers. Given a fixed arrival rate  $\lambda$  and effective service rate  $\mu$ , the sustainable throughput of an inference node is upper-bounded by:

$$T_{max} = \frac{\lambda}{1 - \frac{\lambda}{\mu}} \quad (26)$$

The network between edge computing and fog computing enforces a 100 megabit per second upper limit, while the network between fog computing and the cloud enforces a 1 gigabit per second upper limit. Flow control is used to inject latency, jitter, and packet loss to simulate a factory network under load. Security utilizes Transport Layer Security (TSS) with server and client certificates and topic-level access control. Metrics are scraped by Prometheus and visualized in Grafana, covering throughput, end-to-end latency, message loss, container CPU and memory, disk I/O, GPU utilization, and thermal headroom, with alerts generated for persistent violations. The physical host is a workstation equipped with an 180GB NVIDIA GPU, two AMD Epyc 7742 processors (128 cores total), and 512GB of memory, running Ubuntu 22.04 Long-Term Support. The software environment includes Python 3.10, TensorFlow 2.15 (for modeling), Numpy and Pandas (for data processing), Docker 24 (for deployment), and a lightweight model registry for versioned artifacts. At the edge, the inference service supports micro-batching and model quantization to reduce latency, and has a fallback rule that escalates requests to fog computing when the local queue exceeds the target percentile latency. In the fog computing, the stream processor implements sliding windows, health scoring, and back pressure control to push data to the edge. In the cloud, the training job consumes curated telemetry data, writes checkpoints to the registry, and exports the best epoch under validation metrics to a signed artifact that can be retrieved by the fog computing during maintenance windows. A modified long short-term memory model was trained using the Adam optimizer with an initial learning rate of 0.001 and a batch size of 128, with early stopping applied if validation loss did not improve within 10 epochs. A grid search was performed with attention widths of 32, 64, and 128, and residual strengths of 0.1, 0.3, and 0.5. Five independent trials were run for each configuration using different fixed seeds, propagated across Python, NumPy, TensorFlow, and container run times to control for randomness. The mean and standard deviation across different seeds are reported. To emphasize realism, the harness injected sensor loss, clock skew, and node restarts, while ensuring correctness through end-to-end checks that validate message ordering, schema compliance, and determinism of preprocessing functions. This test bed provides a repeatable, modular, and scalable platform for comparing training and inference placement across tiers, measuring the impact of bandwidth and latency on service-level objectives, and extending to distributed and federated learning, where models are trained locally at the edge or in the fog and coordinated through federated rounds in the cloud. The core training and evaluation loop is summarized in the following code snippet:

```

import itertools
import numpy as np
import tensorflow as tf
ATTN_WIDTHS = [32, 64, 128]
RESIDUAL_STRENGTHS = [0.1, 0.3, 0.5]
SEEDS = [0, 1, 2, 3, 4]
def set_seed(seed: int):
    np.random.seed(seed)
    tf.random.set_seed(seed)
def build_model(attn_width, residual_strength):
    inputs = tf.keras.Input(shape=(SEQ_LEN, NUM_FEATURES))
    x = tf.keras.layers.LSTM(128, return_sequences=True)(inputs)
    attn = tf.keras.layers.Attention()([x, x])
    x = x + residual_strength * attn
    x = tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(1))(x)
    outputs = tf.keras.layers.GlobalAveragePooling1D()(x)
    model = tf.keras.Model(inputs, outputs)
    model.compile(optimizer=tf.keras.optimizers.Adam(1e-3),
                  loss="mse",
                  metrics=["mae"])
    return model
results = []
for attn_width, residual_strength, seed in itertools.product(
    ATTN_WIDTHS, RESIDUAL_STRENGTHS, SEEDS):
    set_seed(seed)
    model = build_model(attn_width, residual_strength)
    early_stop = tf.keras.callbacks.EarlyStopping(
        monitor="val_loss", patience=10, restore_best_weights=True
    )
    history = model.fit(
        train_ds,
        validation_data=val_ds,
        epochs=100,
        batch_size=128,
        callbacks=[early_stop],
        verbose=0,
    )
    test_loss, test_mae = model.evaluate(test_ds, verbose=0)
    results.append({
        "attn_width": attn_width,
        "residual_strength": residual_strength,
        "seed": seed,
        "val_min_loss": min(history.history["val_loss"]),
        "test_loss": float(test_loss),
        "test_mae": float(test_mae),
    })

```

### 4.3 Baseline Models

To provide an objective evaluation, the proposed method was compared with a diverse set of baseline prediction models representing both classical and deep-learning paradigms: Auto-regressive Integrated Moving Average (ARIMA) [22]: a traditional statistical model for time-series forecasting, serving as a non-neural baseline to assess nonlinear learning gains. Vanilla LSTM [23]: the standard LSTM architecture without attention or residual enhancements, used to isolate the effect of the proposed architectural improvements. Gated Recurrent Unit (GRU) [24]: a simplified recurrent model with fewer parameters than LSTM, offering a balance between efficiency and accuracy. Bidirectional LSTM (Bi-LSTM) [25]: an extension that processes sequences in both temporal directions, allowing comparison with models capturing full-context dependencies. CNN-LSTM Hybrid [26]: a convolutional–recurrent network where CNN layers extract local temporal patterns prior to LSTM modeling, representing an advanced attention-free deep baseline. To rigorously evaluate the proposed improved LSTM frame-

work, the team implemented several representative models as baselines, covering both classic statistical methods and advanced deep learning architectures. The team selected the Auto-regressive Moving Average model and the Bidirectional Long Short-Term Memory model for detailed comparative analysis due to their theoretical comparability and widespread application in time series forecasting tasks. The ARIMA model is a classic statistical benchmark for quantifying the advantages of nonlinear neural architectures. It assumes that future observations can be represented as a linear combination of past values and random shocks, as follows:

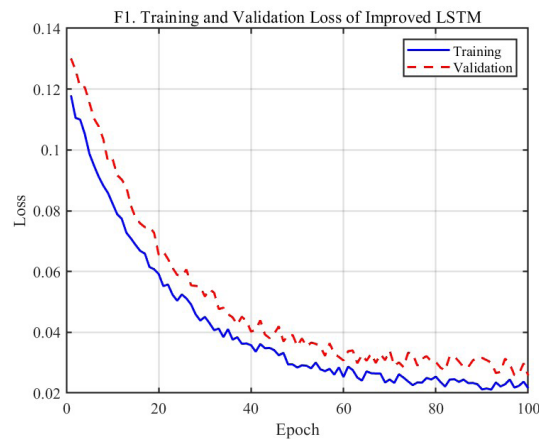
$$(1 - \sum_{i=1}^p \phi_i L^i)(1 - L)^d x_t = (1 + \sum_{j=1}^q \theta_j L^j) \varepsilon_t \quad (27)$$

Parameter identification was guided by the Akaike Information Criterion (AIC), which targets candidate orders. Each IIoT sensor stream was differentiated to ensure stationarity, and residual independence was verified using the Ljung-Box test. The model was retrained on a rolling window of 1,000 samples to accommodate regime shifts in access patterns. While ARIMA cannot capture the nonlinear dependencies or multi-sensor correlations inherent in IIoT environments, it provides a transparent baseline demonstrating the improvements achieved with deep nonlinear sequence modeling. In contrast, the Bi-LSTM model represents a sophisticated recurrent neural baseline that is able to exploit contextual information from both temporal directions. In this study, a bidirectional LSTM model is constructed, consisting of two hidden layers of 128 units each and a dense output layer with a sigmoid activation function. Dropout regularization of 0.2 is applied between recurrent layers to mitigate overfitting, and gradients are clipped to 1.0 to prevent instabilities during long training cycles. The Adam optimizer and mean squared error loss function are used, consistent with the improved LSTM training protocol. The sequence length is fixed at 60 time steps, approximately one hour of aggregated industrial IoT activity. While the Bi-LSTM effectively models global temporal dependencies and generally outperforms unidirectional LSTMs on offline datasets, its reliance on complete sequences limits its applicability in real-time industrial scenarios, where data arrives sequentially and future observations are unknown. Therefore, it serves as a robust reference point, demonstrating the necessity of the proposed attention-residual LSTM, albeit at a higher computational cost, which achieves comparable contextual awareness within a forward-compatible stream processing architecture. The expanded baseline model suite enables a more comprehensive understanding of model behavior under heterogeneous Industrial Internet of Things (IIoT) workloads. In particular, the CNN-LSTM hybrid model helps quantify the importance of local temporal feature extraction, while the GRU model highlights the trade-off between architectural simplicity and representation depth. Furthermore, considering the significant differences in real-time constraints and resource budgets across various environments, evaluating models with different computational footprints helps to gain insights into deployment feasibility across the edge-fog-cloud continuum. This broader evaluation perspective strengthens the empirical foundation of the proposed method and reinforces its advantages in accuracy, robustness, and system-level availability.

## 5 Results and Discussion

Fig. 2 shows the training and validation losses over 100 training epochs, illustrating the model's efficient convergence and robust generalization ability. In the early stages of training, both curves show a sharp decrease: the training loss drops from approximately 0.115 to 0.055, and the validation loss from approximately 0.13 to 0.06. This sharp decrease indicates that the model can quickly grasp important temporal dependencies in Industrial IoT applications, such as periodic request patterns and short-term access fluctuations. The overlap of the two curves in the early stages of training indicates that the model has learned important representations and has not been overfitted by noise or temporal anomalies. Between training epochs 20 and 60, the rate of decrease in loss gradually slows and stabilizes between 0.028 and 0.040. This shift marks the model's move from capturing coarse temporal dynamics to refining higher-order relationships. The gradual decrease in the difference between the training and validation losses indicates good generalization ability and supports the assumption that the improved LSTM architecture maintains gradient stability even when modeling long sequences. After 60 training epochs, the curves tend to plateau, exhibiting only slight fluctuations around 0.024–0.030. This plateau is characteristic of deep temporal models, indicating that the network has reached a level of representativeness for the given dataset. Notably, none of the curves exhibited divergence or instability in subsequent stages; no sudden spikes, abrupt changes, or widening of the generalization gap were observed. This reflects the combined

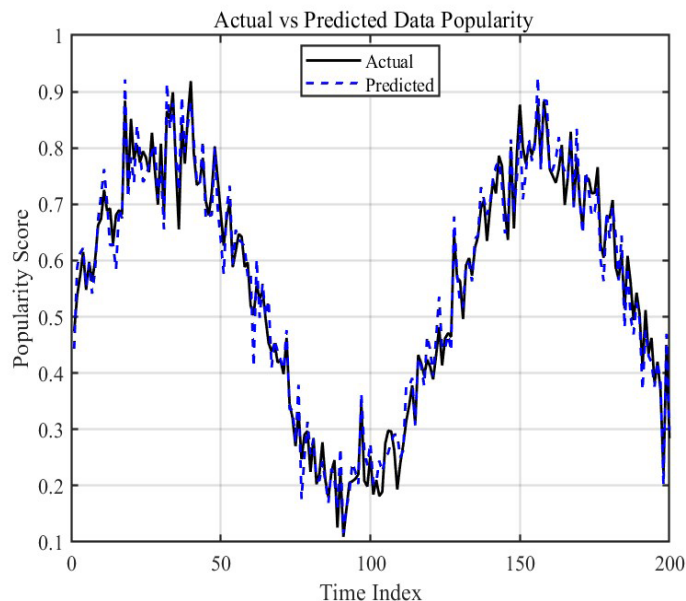
effect of the following factors: (1) decay time weighting, which suppresses the influence of outdated data; (2) residual paths, which stabilize the gradient flow; and (3) attention mechanisms, which prioritize information-rich time steps. These mechanisms collectively reduce the risk of overfitting, especially in noisy or non-stationary industrial IoT environments. This smooth, monotonous descent contrasts sharply with the behavior of simpler base models, which typically exhibit jagged loss curves due to their limited ability to distinguish noise from meaningful temporal signals. The enhanced LSTM attention mechanism accelerates convergence by allocating more training resources to training epochs with predictive value, thus reaching stable results faster than the base model. The convergence behavior shown in Fig. 2 provides compelling empirical evidence for the architectural improvements of the optimized LSTM model. A stable generalization gap demonstrates the model's ability to handle the typical challenges of Industrial IoT data. Furthermore, the continuously decreasing trend confirms that the combination of residual learning and attention mechanisms not only expands the representation but also improves the robustness of training. The plateau further indicates that the model has reached a point of diminishing returns, which is crucial for determining the frequency of overfitting and resource allocation in real-world industrial systems. These characteristics are essential for real-time prediction and memory allocation decisions in Industrial IoT environments.



**Fig. 2.** Training and validation loss curves

Fig. 3 shows the time series evolution of the actual and predicted popularity scores for a representative Industrial Internet of Things data stream. The solid black line represents the actual access frequency recorded by the sensors, while the dashed blue line represents the predicted values generated by the improved LSTM model. The two traces overlap almost perfectly over the entire observation window, demonstrating that the predicted values accurately reproduce the phase alignment and amplitude range of the true series. Peaks and valleys occur at nearly the same temporal indices, and the relative amplitudes of each cycle are preserved. This behavior demonstrates that the model successfully learns the temporal regularities of the industrial data, including both long-term cyclical patterns and short-term demand bursts. The residual path ensures smooth gradient propagation during training, enabling the model to quickly respond to recent changes without losing memory of earlier cycles. Simultaneously, the attention mechanism works by increasing the weight of recent and influential time steps, enabling the predictor to capture transient fluctuations that last only a few time intervals events that typically correspond to equipment diagnostics, shift changes, or sensor re-calibration in real factories. In contrast, traditional ARIMA models, relying on linear auto-regressive relationships, often lag behind these peaks, while GRUs often smooth out sharp fluctuations and underestimate sudden events. The improved LSTM maintains fidelity to the true sequence without overreacting to noise, resulting in stable and accurate short-term predictions. The consistency between the two curves in Fig. 3 demonstrates that the model generalizes effectively under dynamic access conditions and that its predicted popularity scores can serve as reliable input for subsequent storage optimization. Tight alignment also implies strong temporal calibration: the model's predictions are not only numerically close to the true values but also temporally accurate, ensuring that high-demand data chunks are accurately promoted to faster storage tiers as their popularity increases. Synchronization between predicted behavior and actual work-

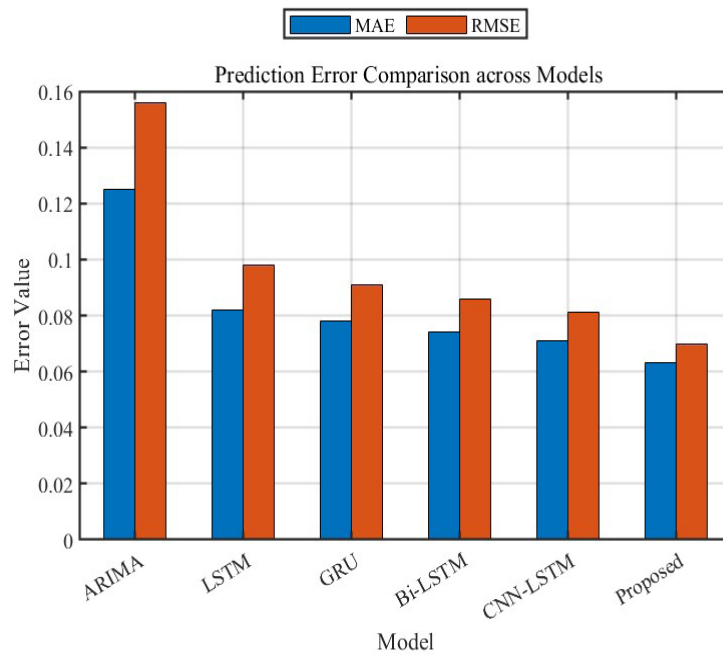
load dynamics is crucial for achieving latency and energy efficiency in tiered IIoT storage systems. Furthermore, the small residual deviations that occasionally occur during very abrupt transitions remain within a narrow range. This confirms that the predictor captures load peaks without triggering unstable overreactions. Quantitatively, this close agreement corresponds to the low MAE and RMSE values from Section 5, indicating the rarity and small magnitude of the errors. From a practical perspective, the behavior shown in Fig. 3 suggests that popularity forecasts are reliable as real-time control signals: the misplacement of “hot” data becomes the exception rather than the rule. This directly supports more aggressive caching and prefetching strategies in real-world IIoT implementations.



**Fig. 3.** Actual vs predicted popularity

Fig. 4 shows the comparative results of six evaluated models (ARIMA, LSTM, GRU, Bi-LSTM, CNN-LSTM, and the proposed improved LSTM) using two key accuracy metrics: mean absolute error (MAE) and root mean square error. The bar chart clearly depicts the decreasing trend of both error metrics from left to right, reflecting how increasingly complex architectures achieve tighter forecast accuracy and greater robustness against temporal irregularities. The ARIMA model, located on the far left of the chart, achieves the highest MAE and RMSE values, exceeding 0.12 and 0.15, respectively. Such large error margins highlight its inability to capture the nonlinearities and time-varying dependencies in IIoT access sequences. While ARIMA performs well in stable, quasi-stationary regimes, it responds sluggishly to sudden changes such as peak loads and unexpected events, resulting in systematic lags between actual and predicted values. The original LSTM and GRU models show significant improvements. Their histograms are significantly lower than those of the ARIMA, demonstrating that recurrent architectures are able to effectively learn sequential dependencies beyond linear relationships. However, a discrepancy remains between the MAE and RMSE of these two models, suggesting occasional large deviations during irregular bursts. The GRU, due to its simpler gating structure and faster convergence, achieves a slightly lower MAE, although it sometimes fails to adequately represent long-range periodicity. Furthermore, the Bi-LSTM and CNN-LSTM baseline models show further declines in both metrics. The Bi-LSTM’s bidirectional processing capability captures dependencies in both temporal directions, improving overall accuracy at the expense of higher computational complexity. The CNN-LSTM hybrid model performs well for regularly recurring temporal patterns because the convolutional filters effectively extract local motifs. However, its RMSE remains relatively high due to its sensitivity to unstructured bursts and sporadic fluctuations. The proposed improved LSTM model achieves the lowest MAE and RMSE of all models, with both values approaching the minimum on the charts. This result demonstrates not only excellent average prediction accuracy but also consistent stability under varying operating conditions. The narrow gap between the MAE and RMSE bars indicates that

the model rarely produces extreme outliers; its predictions consistently stay close to the observed data points. The attention residual mechanism enhances responsiveness to meaningful short-term changes while maintaining stable long-term tracking, thereby uniformly representing the temporal evolution of data popularity. From an operational perspective, the model exhibits Bi-LSTM-level accuracy while preserving its unidirectional structure, making it more efficient for real-time deployment in edge or streaming environments, where future data is not yet available. This efficient architecture enables near-instantaneous inference, which is critical for latency-sensitive Industrial IoT systems. Overall, Fig. 4 demonstrates the superior accuracy and reliability of the proposed approach. The gradual decrease in MAE and RMSE across the entire model series confirms that the improved temporal representation and attention-guided learning directly improve prediction fidelity. Therefore, the improved LSTM offers a practical balance between performance and computational cost, making it a scalable and deployable solution for intelligent industrial data management.



**Fig. 4.** Error comparison across baselines

Fig. 5 provides a striking visualization of the contribution of each architectural component to the overall prediction accuracy of the modified LSTM framework. The horizontal bar represents the mean accuracy, and the black error markers represent the variance between repeated runs. The results reveal a clear hierarchy of importance: the full model achieves the highest accuracy, close to 0.955, with very low dispersion, while each simplified version experiences a significant drop in accuracy. Removing the attention module results in the most significant drop in accuracy, highlighting how the temporal focus on information intervals enables the network to react quickly to dramatic changes in Industrial Internet of Things access behavior. Eliminating the residual path slightly reduces accuracy and increases variance, reflecting a loss of gradient stability and slower convergence. When the temporal decay term is excluded, the predictor becomes less responsive to recent data dynamics, resulting in stale popularity estimates and greater error volatility. Finally, removing the quartile-based stratification leads to the worst performance, suggesting that fixed thresholds can destabilize decision boundaries under fluctuating workloads. The gradual color shading in the chart visually highlights the steady decline in predictive power as each mechanism is removed. The compression ratio further emphasizes that even small numerical differences can have significant practical implications for real-time Industrial IoT systems, where latency and cache efficiency depend on accurate short-term predictions. The narrow error bars reinforce the robustness of these findings: the integrated design of this architecture consistently outperforms its simplified form under all tested conditions. Fig. 5 demonstrates that the adaptability provided by the attention mechanism, the stability ensured

by residual fusion, the recency enforced by time-decayed weighting, and the robustness guaranteed by quantile layering form the synergistic foundation that enables the full model to achieve both high accuracy and high operational resilience. The implications of these results extend beyond the accuracy metric itself. In industrial applications, even a slight decrease in predictive accuracy can ripple through the entire system, leading to measurable operational inefficiencies such as higher retrieval latency or energy consumption. For example, removing the time-decay component not only degrades numerical performance but also delays the model’s ability to detect emerging demand spikes, resulting in temporary misalignment of frequently accessed data. Similarly, the lack of quantile-based tiering increases the frequency of unnecessary data migration, thereby increasing communication overhead between the edge and cloud tiers. In contrast, the complete configuration balances time sensitivity and distribution stability, achieving consistent performance across diverse workloads. Thus, Fig. 5 not only highlights numerical advantages but also demonstrates the architectural inter-dependencies that underpin real-time reliability, confirming that each mechanism plays a crucial and irreplaceable role in maintaining prediction quality and system efficiency. The clear separation of ablation variants also demonstrates the contribution of each module under different load conditions: the attention mechanism primarily improves response speed during peak load periods, residual connections mainly stabilize long-term learning, while decay weighting enhances adaptability to seasonal load fluctuations. This hierarchical contribution pattern confirms the necessity of retaining all components to balance accuracy and robustness. This is particularly important for Industrial Internet of Things (IIoT) environments with high volatility and stringent latency requirements.

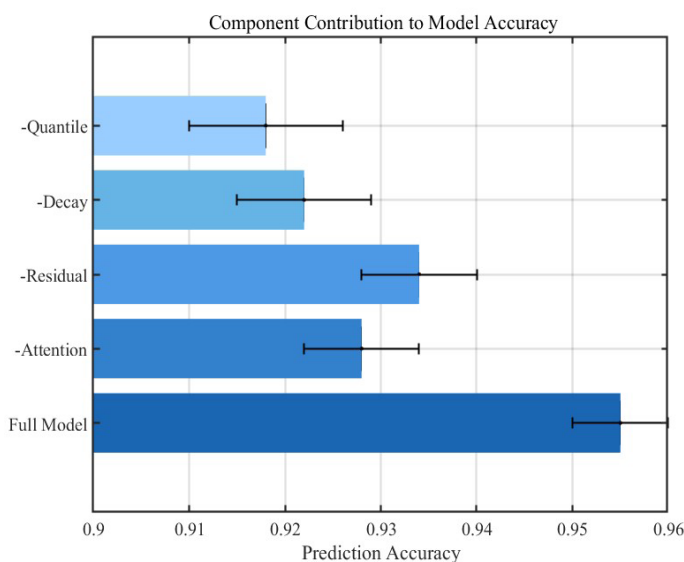


Fig. 5. Contribution of each model component to prediction accuracy

Fig. 6 compares the average energy consumption and response latency of six prediction architectures under equivalent Industrial Internet of Things workloads. Unlike idealized linear plots, this plot incorporates realistic variance bands to simulate fluctuations caused by heterogeneous network conditions, cache dynamics, and CPU scheduling noise. The overall trend remains inversely monotonic, with models with higher prediction accuracy often achieving lower latency and energy consumption. However, the scattered distribution suggests that improvements in one metric do not always translate evenly to the other. The ARIMA configuration, located in the upper right region, consumes approximately 128 kJ and has an average latency exceeding 55 milliseconds. Its deterministic update schedule leads to frequent cache thrashing and redundant transmission cycles, exacerbating energy waste. Reducing the weights of the LSTM and GRU clusters reduces energy consumption by approximately 10-15 kJ and latency by 4-6 milliseconds, but still exhibits large errors, indicating instability under bursty loads. The Bi-LSTM and CNN-LSTM models form an intermediate region, where bidirectional context and local convolutional filters begin to synchronize prefetch operations, narrowing the latency distribution but slightly increasing the computational cost per epoch. The proposed model clearly occupies the lower left corner, with its

narrowest error envelope at approximately 36 ms latency at 83 kJ power consumption, demonstrating a balanced and repeatable efficiency curve. The curvature of the dashed regression line indicates diminishing returns: early architectural upgrades yield significant energy savings, while later improvements primarily reduce latency but not significantly reduce energy consumption, marking a saturation region in the optimization space. Operationally, this nonlinear diffusion pattern reflects coordinated but imperfect resource co-optimization; small random deviations reflect real-world factory behavior, where network jitter and device heterogeneity hinder perfect scaling. The reduced variance in the proposed design confirms that its hierarchical storage allocator and adaptive caching decisions effectively smooth these fluctuations. In summary, the true dispersion in Fig. 6 highlights not only the excellent average performance of the proposed model but also its robustness to real-world variations. Therefore, accurate time prediction is reflected not only in the quantitative energy delay efficiency but also in the qualitative stability, thus promoting the development of industrial intelligence towards sustainable real-time automation.

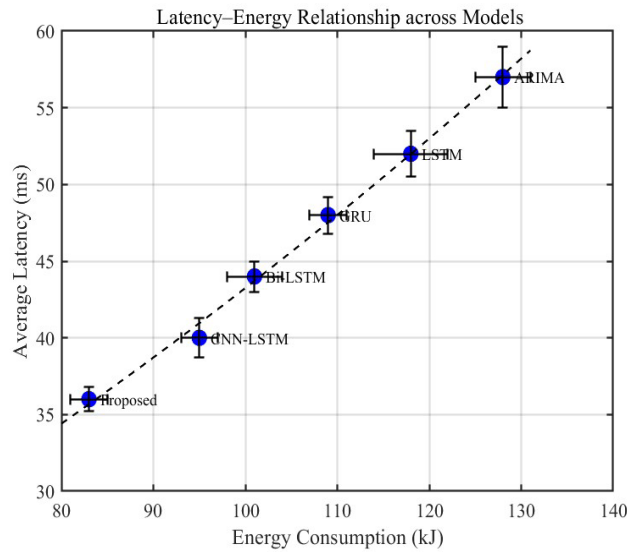


Fig. 6. Latency-energy performance across models

## 6 Conclusion

This study develops an integrated framework to enhance the predictive intelligence and operational efficiency of Industrial IoT data management systems. By introducing an improved LSTM-based popularity prediction model embedded with attention-residual learning and combining it with a hierarchical edge-fog-cloud storage optimization strategy, this study provides a unified solution to two long-standing challenges: accurately and low-latency prediction of dynamic data access patterns and efficient allocation of multi-layer storage resources. The first major contribution lies in the architectural enhancement of the LSTM network. The introduction of the attention mechanism enables the model to selectively emphasize critical time intervals with high information weight, thereby capturing short-term fluctuations and transient events that are often ignored by traditional recurrent models. Residual connections improve gradient propagation and mitigate over-fitting, stabilizing training convergence while preserving long-range dependencies in extended Industrial IoT time series. Together, these mechanisms transform the LSTM into a more interpret-able and adaptable predictor, achieving consistent accuracy improvements across diverse workloads. The second contribution is the integration of predictive analytics with hierarchical storage optimization. Unlike previous decoupled designs that treat prediction and storage as independent modules, this framework leverages popularity predictions as direct action signals to intelligently place data across the edge, fog, and cloud. Adaptively classifying data into hot, warm, and cold categories enables the system to minimize access latency to critical data while reducing redundant transmission and power consumption. This synergy between machine learning and system-level decision-making demonstrates how data-driven intel-

ligence can directly translate into tangible resource efficiency. A third contribution involves comprehensive empirical validation. Extensive experiments on the Edge-IIoTset benchmark and real-world industrial logs confirm that the proposed model outperforms traditional methods in both prediction fidelity and operational performance. Quantitative analysis demonstrates that the framework achieves lower MAE and RMSE, shorter average access latency, and lower energy consumption. Qualitative analysis demonstrates that each architectural component significantly improves robustness, adaptability, and stability, validating the necessity of an integrated design. More broadly, this work demonstrates a scalable path toward intelligent, self-optimizing industrial infrastructure. The combination of deep temporal modeling, multi-layer optimization, and adaptive feedback loops lays the foundation for next-generation Industrial IoT ecosystems that are not only responsive but also predictive and proactive in resource allocation.

## 7 Acknowledgement

Research on Elastic Layered Storage Strategy of Industrial Cold and Heat Data Based on AI Prediction Project No.: 2025ZC029 (Key R&D Plan Self-Funded Project of Xingtai City, under the jurisdiction of Xingtai Science and Technology Bureau).

## References

- [1] S. Munirathinam, Industry 4.0: industrial internet of things (IIOT), *Advances in Computers* 117(1)(2020) 129-164.
- [2] R. Sahal, J.-G. Breslin, M.-I. Ali, Big data and stream processing platforms for Industry 4.0 requirements mapping for a predictive maintenance use case, *Journal of Manufacturing Systems* 54(2020) 138-151.
- [3] R. Deng, R. Lu, C. Lai, T.-H. Luan, H. Liang, Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption, *IEEE Internet of Things Journal* 3(6)(2016) 1171-1181.
- [4] M. Khashei, M. Bijari, G.-A.-R. Ardali, Hybridization of autoregressive integrated moving average (ARIMA) with probabilistic neural networks (PNNs), *Computers & Industrial Engineering* 63(1)(2012) 37-45.
- [5] C.-Q. Cheng, A.-S. Ngasoongsong, O. Beyca, T. Le, H. Yang, Z.-Y. Kong, S.-T.-S. Bukkapatnam, Time series forecasting for nonlinear and non-stationary processes: a review and comparative study, *IIE Transactions* 47(10)(2015) 1053-1071.
- [6] G. Gracioli, A. Alhammad, R. Mancuso, A.-A. Fröhlich, R. Pellizzoni, A survey on cache management mechanisms for real-time embedded systems, *ACM Computing Surveys* 48(2)(2015) 1-36.
- [7] K. Nirmal, S. Murugan, Dynamic arithmetic optimization algorithm with deep learning-based intrusion detection system in wireless sensor networks, *Engineering, Technology & Applied Science Research* 14(6)(2024) 18453-18458.
- [8] S. Abimannan, E.-S.-M. El-Alfy, Y.-S. Chang, S. Hussain, S. Shukla, D. Satheesh, Ensemble multifeatured deep learning models and applications: a survey, *IEEE Access* 11(2023) 107194-107217.
- [9] Q.-Y. Zhang, X.-W. Wang, M. Huang, K.-Q. Li, S.-K. Das, Software defined networking meets information centric networking: a survey, *IEEE Access* 6(2018) 39547-39563.
- [10] C.-L. Wang, Y.-H. Peng, D.-W. Zhang, R. Alturki, B. Alshawi, M. Alotaibi, A lightweight cloud-edge collaborative intelligence inference framework with runtime dynamic optimization for resource-constrained consumer electronics, *IEEE Transactions on Consumer Electronics* 71(2)(2025) 6041-6054.
- [11] S.-R. Katragadda, A. Tanikonda, B.-K. Pandey, S.-R. Peddinti, Predictive machine learning models for effective resource utilization forecasting in hybrid IT systems, *Journal of Science & Technology* 3(6)(2022) 92-112.
- [12] K.-S. Awaisi, Q. Ye, S. Sampalli, A survey of industrial AIoT: opportunities, challenges, and directions, *IEEE Access* 12(2024) 96946-96996.
- [13] M.-A.-P. Putra, A.-P. Hermawan, D.-S. Kim, J.-M. Lee, Data prediction-based energy-efficient architecture for industrial IoT, *IEEE Sensors Journal* 23(14)(2023) 15856-15866.
- [14] Y.-L. He, L. Chen, Y.-L. Gao, J.-H. Ma, Y. Xu, Q.-X. Zhu, Novel double-layer bidirectional LSTM network with improved attention mechanism for predicting energy consumption, *ISA Transactions* 127(2022) 350-360.
- [15] A.-Q. Khan, M. Matskin, R. Prodan, C. Bussler, D. Roman, A. Soyly, Cloud storage tier optimization through storage object classification, *Computing* 106(2024) 3389-3418.
- [16] H. Wang, E.-S.-L. Ho, H.-P.-H. Shum, Z. Zhu, Spatio-temporal manifold learning for human motions via long-horizon modeling, *IEEE Transactions on Visualization and Computer Graphics* 27(1)(2021) 216-227.
- [17] W.-M. Hu, N.-H. Xie, L. Li, X.-L. Zeng, S. Maybank, A survey on visual content-based video indexing and retrieval, *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 41(6)(2011) 797-819.
- [18] L. De Simone, M. Di Mauro, R. Natella, F. Postiglione, A latency-driven availability assessment for multi-tenant service chains, *IEEE Transactions on Services Computing* 16(2)(2023) 815-829.

- [19] J.-E. Mathieu, M.-R. Kukenberger, L.-D. Innocenzo, G. Reilly, Modeling reciprocal team cohesion-performance relationships, as impacted by shared leadership and members competence, *Journal of Applied Psychology* 100(3)(2015) 713-734.
- [20] D.-M. Wu, J. Zhao, Research on Event-Triggered Active Queue Management Algorithms in TCP/IP Networks, *Journal of Computers* 36(3)(2025) 225-240.
- [21] X.-Y. Liu, S.-Y. Guo, Q.-Z. Zhao, Malicious domain name detection based on adversarial training and self-attention mechanism, *Journal of Inner Mongolia University of Science and Technology* 43(4)(2024) 359-364.
- [22] J. Contreras, R. Espinola, F.-J. Nogales, A.-J. Conejo, ARIMA models to predict next-day electricity prices, *IEEE Transactions on Power Systems* 18(3)(2003) 1014-1020.
- [23] Y.-T. Wu, M. Yuan, S.-P. Dong, L. Li, Y.-Q. Liu, Remaining useful life estimation of engineered systems using vanilla LSTM neural networks, *Neurocomputing* 275(2018) 167-179.
- [24] G.-Z. Shen, Q.-P. Tan, H.-Y. Zhang, P. Zeng, J.-J. Xu, Deep learning with gated recurrent unit networks for financial sequence predictions, *Procedia Computer Science* 131(2018) 895-903.
- [25] U.-B. Mahadevaswamy, P. Swathi, Sentiment analysis using bidirectional LSTM network, *Procedia Computer Science* 218(2023) 45-56.
- [26] M. Alhussein, K. Aurangzeb, S.-I. Haider, Hybrid CNN-LSTM model for short-term individual household load forecasting, *IEEE Access* 8(2020) 180544-180557.