Time Series Forecasting using Diffusion Model in Case Water Quality of Chao Phraya River

Woraphon Lilakiatsakun
Innovative Education and Lifelong Learning Institute, Rajamangala University of Technology Tawan-ok
43 Moo 6, Bang Phra, Si Racha, Chonburi, Thailand 20110
Email: woraphon_li@rmutto.ac.th

Thanapol Phungtua-eng, Werachart Muttitanon
Department of Information Technology, Faculty of Business Administration an Information Technology
Rajamangala University of Technology Tawan-ok
58 Vibhavadi Rangsit Road, Ratchadaphisek, Din Daeng, Bangkok, Thailand 10400
Email: Thanapol_ph@rmutto.ac.th, werachart_mu@rmutto.ac.th

Saowakhon Nookhao
KMITL Business School, King Mongkut's Institute of Technology Ladkrab
1 Chalong Krung, 1 Alley, Lat Krabang, Bangkok 10520
Email: saowakhon.no@kmilt.ac.th

Manuscript Received June 27, 25025
Revised August 4, 2025
Accepted August 9, 2025

ABSTRACT

Time series forecasting is a fundamental task for predicting future values in order to understand underlying behavior. Numerous forecasting models rely on learning patterns from historical observations. However, these models face limitations when time series data exhibit inherent uncertainty, multi-modality (i.e., multiple plausible outcomes), abrupt regime changes, or complex noise. Recently, an alternative solution has emerged in which models aim to approximate the distribution of possible future outcomes. One such solution is the use of Denoising Diffusion Probabilistic Models (Diffusion Models), which offer a promising framework by learning a generative process that models complex data distributions in a probabilistic and iterative manner. This capability enables the model to effectively capture uncertainty and multi-modality in time series forecasting. In this study, we implement a Diffusion Model for time series forecasting. We conduct an evaluation on a real-world case study involving water quality in the Chao Phraya River. The results demonstrate that the Diffusion Model outperforms classical techniques such as Long Short-Term Memory (LSTM) in time series forecasting.

Keywords: *Diffusion Models, Time series forecasting, Probabilistic, Generative Models*

1. INTRODUCTION

Time series data are utilized to understand behavior and extract insights across various domains, such as weather forecasting, electricity monitoring, and resource management. However, real-world time series often suffer from issues such as missing values, which may arise due to measurement errors in IoT devices or communication losses. In addition, forecasting future values based on historical observations is crucial for applications like resource management and economic planning. This task requires sufficient training data and time series with few missing values. Those conditions are often insufficient for time series analysis in practice, that may deviate from the actual underlying phenomenon.

To mitigate these issues, researchers typically focus on two core tasks in time series analysis: imputation and forecasting. Time series imputation involves making assumptions about the underlying pattern of the series to replace missing values in a way that aligns with the actual pattern. In contrast, time series forecasting predicts future values based on historical observations.

Recently, many research have explored the use of generative AI models to synthesize complex patterns in time series data that closely resemble real-world series [8], [11]. Specifically, they can generate synthetic observations from historical records while preserving key statistical properties, particularly the distribution of the original data. This capability is commonly associated with the Denoising Diffusion Probabilistic Model (hereafter referred to as the Diffusion Model) [3], [13], [15]. Diffusion Models have been applied to both time series imputation and forecasting tasks. Moreover, Diffusion Models have demonstrated stronger performance compared to generative adversarial networks (GANs) and variational autoencoders (VAEs) [2], [10], [18].

Building on the benefits of Diffusion Models, we conduct a comprehensive study applying them to real-world case studies of water quality monitoring in Chao Phraya River, Bangkok[1]. This study highlights the challenges of time series imputation and forecasting under conditions of missing values and limited data in practical scenarios. The main contributions of this study are as follows:

- We demonstrate the application of Diffusion Models to a real-world case study involving forecasting tasks.
- We conduct an analysis of the influence of key hyperparameters in Diffusion Models to better understand their effectiveness for time series forecasting.
- We evaluate the performance of Diffusion Models in comparison with state-of-the-art methods for time series forecasting

2. PRELIMINARIES

2.1.DIFFUSION MODELS

The Diffusion Model was first proposed by J. Sohl Dickstein et al. [15], and later improved by J. Ho et al. through the use of variational inference [3]. A Diffusion Model consists of two main phases. First, the forward process aims to systematically and gradually destroy the original data into pure noise through an iterative noising procedure. In contrast, the reverse process involves training a model to iteratively restore the noisy data back to its original form. The Figure 1 shows both forward and reverse process of the Diffusion Model. The mathematical formulations of both the forward and reverse processes are defined as follows.

[1] https://gdcatalog.go.th/organization/4388d502-250d-48b3-bdcc-43c0d6f0287b

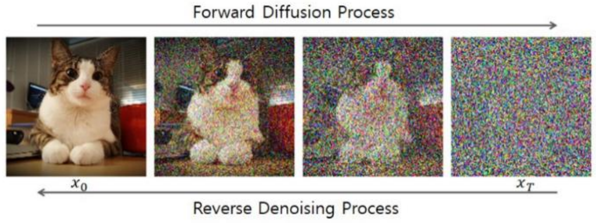


Fig. 1 *Illustration of the forward and reverse processes in the Diffusion Model [11].*

1) *Forward Process*: Let $X^0 \in R^{M\times N}$ denote the original input matrix, where the superscript 0 indicates the initial state of the Diffusion Model before any noise has been injected. Consequently, the sequence of noisy inputs at each iteration of the forward process is denoted by $X^1, X^2, \ldots, X^K$. The forward process of the Diffusion Model is defined as:

$$q(X^k \mid X^{k-1}) = \mathcal{N}\left(X^k; \sqrt{\alpha_k}X^{k-1}, (1-\alpha_k)I\right) \quad (1)$$

where $\alpha_k \in (0,1)\ for\ k = 1,2,\ldots,K$ are hyperparameters controlling the noise level at each iteration, such that X^K becomes a sample drawn from a standard Gaussian distribution $\mathcal{N}(0, I)$.

Eq. 1 can be reformulated to express the cumulative effect of the forward process from X^0 to X^k. As shown in [15], the resulting closed-form distribution is given by:

$$q(X^k \mid X^0) = \mathcal{N}\left(X^k; \sqrt{\widetilde{\alpha_k}}X^0, (1-\widetilde{\alpha_k})I\right) \quad (2)$$

Here, $\widetilde{\alpha_k} \coloneqq \prod_{i=1}^{k} \alpha_i$ represents the cumulative product of the noise schedule up to step k.

2) *Reverse Process*: In the reverse process, the Diffusion Model learns to reconstruct X^0 from the noisy input X^K. The reverse process is defined as a learned distribution parameterized by neural network parameters θ:

$$p_\theta(X^{k-1} \mid X^k) = \mathcal{N}\left(X^{k-1}; \mu_\theta(X^k, k), \Sigma_\theta(X^k, k)\right) \quad (3)$$

Here, θ denotes the learnable parameters of the neural network. The objective function introduced by J. Ho *et al.* [3] simplifies the variational bound into a denoising score-matching loss, where the model learns to predict the added noise. Specifically, a noisy sample can be generated using reparameterization from Eq. 2 as follows:

$$X^k = \sqrt{\widetilde{\alpha_k}}X^0 + \sqrt{1-\widetilde{\alpha_k}}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \quad (4)$$

The loss function by J. Ho *et al.* [3] is defined as:

$$\mathcal{L}_{\text{simple}}(\theta) = E_{X^0,\epsilon,k}\left[||\epsilon - \epsilon_\theta\left(X^k, k\right)||^2\right] \quad (5)$$

where $\epsilon_\theta\left(X^k, k\right)$ denotes the predicted noise at timestep k using a neural network (typically a U-Net or Transformer). From the loss function in Eq. 5, the model is trained to minimize the difference between the actual Gaussian noise ϵ and the predicted noise $\epsilon_\theta\left(X^k, k\right)$. This encourages the model to produce noise estimates that follow the true Gaussian distribution used during the forward process. After training, the predicted noise $\epsilon_\theta(X^k, k)$. can be used to reconstruct the original data X^0 from the noisy input X^k. Based on this concept, the Diffusion Model learns to approximate the original data distribution by modeling the reverse noise process. Once trained, it can generate synthetic data by sampling from the learned reverse distribution starting from Gaussian noise. Therefore, the trained Diffusion Model can be used to generate future values or fill in missing values in the context of time series forecasting and imputation [1], [11].

3. RELATED WORKS

Time series forecasting involves learning patterns from historical observations to predict future values of the time series [6], [19], [22]. One well-known machine learning method used for this task is the *Long Short-Term Memory* (LSTM) network [15]. However, LSTM models often struggle to achieve high accuracy in long-horizon forecasting, where the goal is to predict values beyond 48 future time steps —a task known to be particularly challenging [22].

Many long-horizon time series forecasting models rely on Transformer based architectures, which are designed to capture long range dependencies through attention mechanisms. Models such as Informer [22], Autoformer [20], and FEDformer [23] have shown strong performance by leveraging hierarchical or frequency-aware attention. However, despite their predictive accuracy, these models typically focus on point estimates rather than modeling the underlying data distribution.

Diffusion Models offer a generative perspective by learning the underlying data distribution, which enables them to generate diverse outputs that closely resemble real data [8], [10]. In this paper, we compare the performance of LSTM and Diffusion Models for long-horizon time series forecasting, highlighting the advantages of generative approaches in capturing uncertainty and structural patterns.

The core idea of Diffusion Models is to train a model that gradually adds noise to the original data in a forward process, and then learns to reconstruct the original data from the noisy version through a reverse process. The reconstructed outputs are synthetic samples that closely resemble real data. Building on this concept, Diffusion Models have been successfully applied in various domains, including noise filtering, image generation, data imputation, and time series forecasting [8], [21]. In this paper, we focus specifically on their application to time series imputation and forecasting. There are several studies that utilize Diffusion Models for time series forecasting [1], [4], [5], [7], [13], as well as for imputation tasks [1], [4], [9], [16].

4. ADOPTED DIFFUSION MODEL ARCHITECTURE

To address the probabilistic forecasting task described above, we adopt a diffusion model as our generative forecaster. This model is designed to capture complex temporal dependencies and generate diverse future series based on historical observations. The implementation used in this work is publicly available at https://github.com/thanapol2/DDPMForecaster.

4.1 PROBLEM DEFINITION AND TERMINOLOGIES

Let $X \in R^{M\times N}$ denote a multivariate time series, where *M* is the number of variables (or sensors) and *N* is the total number of time steps. For each variable $m \in 1, \ldots, M$, the clean historical input sequence of length *L* is denoted as $X^0_{m,1:L} \in R^L$, where the superscript 0 indicates the original data without injected noise. The forecasting objective is to predict the future values over the next *H* time steps, represented as $\hat{X}_{m,L+1:L+H} \in R^H$.

The model is trained to learn a mapping from the input segment $X^0_{m,1:L}$ to the forecast segment $\hat{X}_{m,L+1:L+H}$ for all $m \in 1, \ldots, M$. In this study, we address this forecasting task using a generative approach based on Diffusion Models, which are designed to capture uncertainty and complex temporal patterns.

4.2 NEURAL NETWORK ARCHITECTURE

Our adopted Diffusion Model forecaster is implemented as a conditional neural network that operates on sliding windows of the multivariate time series. Given *M* variables (or sensors), an input context of length *L*, and a forecasting horizon of length *H*, the model utilizes three types of input. First, it receives a forecast segment that has been partially diffused with noise, representing a corrupted version of the true future values. Second, it incorporates the clean historical context, which provides recent observations for all variables and essential temporal information for forecasting. Third, the model is conditioned on the current diffusion timestep, which is normalized to promote stable learning across all steps.

These inputs are concatenated and passed through a feed-forward neural network comprising two fully connected layers with a ReLU activation in between. The first layer processes a flattened vector formed by combining the noised forecast, the context, and the normalized timestep, projecting it into a hidden representation. The second layer generates the predicted noise, which is subsequently reshaped to match the dimensions of the forecast window.

4.3 DIFFUSION AND REVERSE PROCESS

During training, Gaussian noise is injected into the clean forecast window at a randomly selected diffusion step *n*, following a predefined linear schedule for the noise variance. The model learns to predict the added noise, enabling it to perform denoising at each step. The forward (diffusion) process simulates gradually corrupting the target future with noise, while the reverse process iteratively denoises a random initial sample to generate diverse forecasts. Both processes follow the Diffusion Model framework, with the network conditioned on both the historical context and the current diffusion timestep.

The simplicity of this architecture makes it suitable for capturing temporal structure in multivariate time series while maintaining computational efficiency.

4.4 SAMPLING PROCEDURE

The reverse process in our adopted diffusion model generates sample forecasts by iteratively denoising from pure Gaussian noise. Starting from an initial random sequence, the model applies a series of denoising steps in reverse order, each step leveraging the learned noise predictor ϵ_θ and conditioned on the historical context as well as the current diffusion timestep.

At each reverse diffusion step k, the following update is performed:

$$X^{k-1} = \frac{1}{\sqrt{\alpha_k}}\left(X^k - \sqrt{1-\tilde{\alpha}_k}\,\epsilon_\theta(X^k,k)\right) + \sqrt{1-\tilde{\alpha}_k}\,z \quad (6)$$

where $z \sim \mathcal{N}(0,I)$ for $k > 1$, and $z = 0$ for $k = 1$. Here, α_k denotes the cumulative product of the noise coefficients at step k, and $\tilde{\alpha}_k$ is the corresponding cumulative value in the diffusion schedule. This iterative procedure transforms the initial noise into a coherent time series forecast by progressively reducing the uncertainty at each step.

4.5 FORECASTING WITH DIFFUSION MODELS

To generate probabilistic forecasts for a single time series, we leverage the generative capability of diffusion models by sampling multiple future trajectories. Given an observed historical context $X^0_{m,1:L}$ of length L for variable \$m\$, the model predicts the next H time steps as follows. For each forecast, a sample is drawn from a standard Gaussian distribution and iteratively denoised using the learned reverse diffusion process, conditioned on the observed context.

This sampling procedure is repeated N times, with each run producing a plausible realization of the future segment $\hat{X}_{m,L+1:L+H}$. The final forecast is computed as the average across all sampled trajectories:

$$\hat{X}_{m,L+1:L+H} = \frac{1}{N}\sum_{n=1}^{N}\hat{X}^{(n)}_{m,L+1:L+H} \quad (7)$$

where $\hat{X}^{(n)}_{m,L+1:L+H}$ denotes the *n*-th generated trajectory for variable m. This Monte Carlo approach enables the model to capture both uncertainty and possible multi-modal behaviors in the predicted future values. In our implementation, this process is carried out by the $sample$ method, which aggregates results from multiple runs to provide point forecasts and uncertainty estimates, following the reverse denoising procedure proposed in [3].

5. DATASET

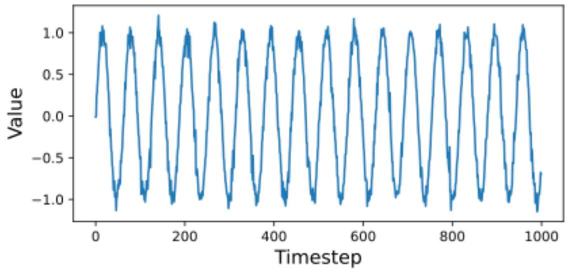
5.1 SYNTHETIC DATASETS

To evaluate the effectiveness of Diffusion Models in forecasting under challenging conditions, we construct two synthetic time series datasets that simulate non-ideal scenarios often encountered in real-world applications:

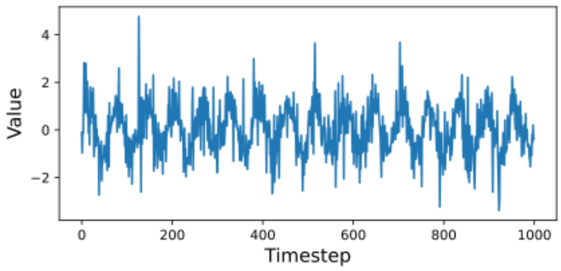
- **Non-Gaussian Heavy-Tailed Noise (Laplace)**: This dataset simulates time series data corrupted by Laplace-distributed noise, characterized by sharp peaks and heavy tails. It is designed to assess the model's robustness to non-Gaussian disturbances and outliers.
- **Non-linear Heteroskedastic Series**: This dataset exhibits non-linear temporal dynamics and heteroskedastic variance patterns (i.e., time-dependent variance). It aims to evaluate the model's ability to adapt to complex, non-stationary patterns with varying levels of uncertainty over time.

Two datasets were used to control and vary the noise levels, each consisting of 10,000 time steps. Examples of high and low noise levels for the Laplace and

Heteroskedastic datasets, using the first 1,000 time steps, are shown in Figs. 2 and 3, respectively. Note that only one time series ($X^0_{m,1:1000}$) is shown for clarity of visualization.

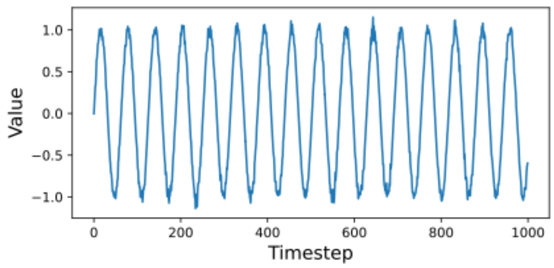


(a) low-level noise

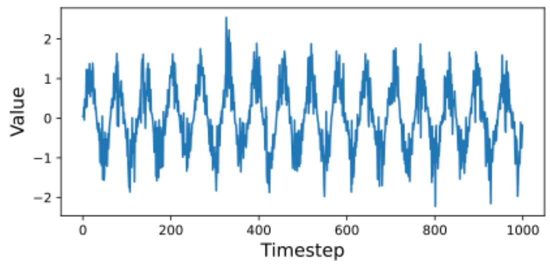


(b) high-level noise

***Fig**. **2** Non-Gaussian heavy-tailed noise (Laplace) at varying noise levels.*



(a) low-level noise



(b) high-level noise

Fig. 3 *Non-linear heteroskedastic series* at varying noise levels.

5.2 REAL-WORLD DATASETS

We utilized real-world water quality datasets provided by the Metropolitan Waterworks Authority in Bangkok[2]. The data were collected from several monitoring stations along the Chao Phraya River, including Phra Nang Klao Bridge, Phra Phuttha Yodfa Bridge, Khlong Lat Pho, and the Port Authority of Thailand. Each dataset consists of multivariate time series with measurements recorded every 10 minutes. The recorded variables include temperature, conductivity, total dissolved solids (TDS), pH, and chlorophyll concentration.

During preprocessing, we observed that the dataset contained missing values, likely due to sensor errors or data transmission issues. To this issue, we aggregated the data from 10-minute intervals to hourly intervals before using it for training and forecasting in our experiments. These datasets are publicly available on the Thai government open data portal.

6. EVALUATION SETTING

All experiments were implemented using Python 3.11 and executed on a system equipped with an NVIDIA RTX 2060 GPU (8GB RAM).

6.1 BASELINE LSTM CONFIGURATION FOR COMPARISON

To evaluate the performance of our proposed method for time series forecasting, we compare it with a standard LSTM model. The LSTM is trained using the same input and output window settings as our method to ensure a fair comparison. It is optimized using the mean squared error (MSE) loss function and implemented with a single hidden layer and a fixed number of hidden units, which are tuned through validation.

To apply LSTM in a generative manner for time series forecasting, we adopt a Monte Carlo approach using dropout at inference time, as in Eq. 7. This allows us to estimate predictive uncertainty and ensures a fair comparison between the LSTM baseline and our proposed diffusion-based method.

In this study, we implement an LSTM-based forecasting model using the standard method of the PyTorch library. The model employs a single LSTM layer with 64 hidden units, consistent with the hidden size used in our Diffusion Model. A dropout layer with a probability of 0.5 is applied to prevent overfitting. For LSTM training, we use the Adam optimizer with a learning rate of 1×10^{-3}, and the MSE is used as a loss

[2] https://gdcatalog.go.th/organization/4388d502-250d-48b3-bdcc-43c0d6f0287b

function. LSTM model is available in our GitHub response[3].

6.2 EVALUATION METRICS

To evaluate the performance of the forecasting models, we use both deterministic and probabilistic metrics. The deterministic metrics assess the average magnitude of error between the predictions and ground truth values, while the probabilistic metric evaluates the quality of the predictive distribution.

All metrics are computed in the context of multivariate time series forecasting, where $X^0_{m,t}$ denotes the clean ground truth value for variable *m* at time *t*, and $\hat{X}_{m,t}$ denotes the predicted value.

Note: For simplicity, we omit the superscript and refer to $X_{m,t}$ as the ground truth in the remainder of this section.

Mean Squared Error (MSE) measures the average squared difference between predicted and true values, heavily penalizing larger errors:

$$\text{MSE} = \frac{1}{MH}\sum_{m=1}^{M}\sum_{t=1}^{H}\left(\hat{X}_{m,t} - X_{m,t}\right)^2 \quad (8)$$

Mean Absolute Error (MAE) computes the average absolute difference between predictions and actual values:

$$\text{MAE} = \frac{1}{MH}\sum_{m=1}^{M}\sum_{t=1}^{H}\left|\hat{X}_{m,t} - X_{m,t}\right| \quad (9)$$

Mean Absolute Percentage Error (MAPE) expresses the error as a percentage. It is commonly used for interpretability but may be undefined when any $X_{m,t} = 0$:

$$\text{MAPE} = \frac{100}{MT}\sum_{m=1}^{M}\sum_{t=1}^{H}\left|\frac{\hat{X}_{m,t} - X_{m,t}}{y_{i,t}}\right| \quad (10)$$

Symmetric Mean Absolute Percentage Error (sMAPE) addresses MAPE's limitations by normalizing the error with the sum of absolute values:

$$\text{sMAPE} = \frac{100}{MT}\sum_{m=1}^{M}\sum_{t=1}^{T}\frac{|\hat{X}_{m,t} - X_{m,t}|}{(|\hat{X}_{m,t}| + |X_{m,t}|)/2} \quad (11)$$

Continuous Ranked Probability Score (CRPS) measures the distance between the predicted cumulative distribution function (CDF) *F* and the observed value $X_{m,t}$, generalizing MAE to probabilistic forecasts [12]:

$$\text{CRPS}(\text{F}, X_{m,t}) = \int_{-\infty}^{\infty}\left(\text{F}(z) - 1\{z \geq X_{m,t}\}\right)^2 dz \quad (12)$$

Lower values for all metrics indicate better predictive performance.

7. EXPERIMENTAL RESULTS

7.1 INFLUENCE OF NOISE LEVEL AND FORECAST LENGTH ON LOSS IN THE DIFFUSION MODEL

We begin by evaluating how the forecast length and noise level affect the training dynamics of the diffusion model. In this experiment, we focus on the denoising performance by computing the MSE between the predicted and true noise at each diffusion step. This analysis helps reveal how different forecast configurations and noise intensities influence the model's ability to learn an accurate denoising function across the forward diffusion iterations. The plot of influence of noise level and forecast length on loss results are show as Figures 4 and 5, respectively. Note that Laplace and Hetero refer to the Non-Gaussian Heavy-Tailed Noise and Non-linear Heteroskedastic Series datasets, respectively.

As shown in the training loss results in Figure 4 we observed that neither the noise level nor the dataset type had a significant impact on training loss across epochs. All settings follow a similar trend, and the loss stabilizes around epoch 50. This demonstrates the robustness of the Diffusion Model with respect to noise levels. Based on these results, we set the number of training epochs to 50 for subsequent evaluations. We then evaluated the influence of forecast length, which emphasizes the challenge of long-horizon time series forecasting, as discussed in [22].

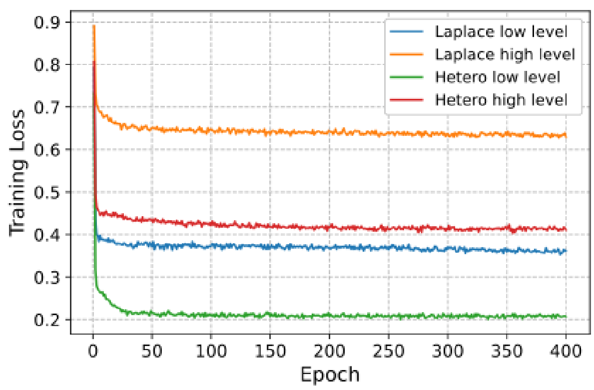


Fig. 4 *Training loss over epochs across different datasets.*

[3] https://github.com/thanapol2/DDPMForecaster

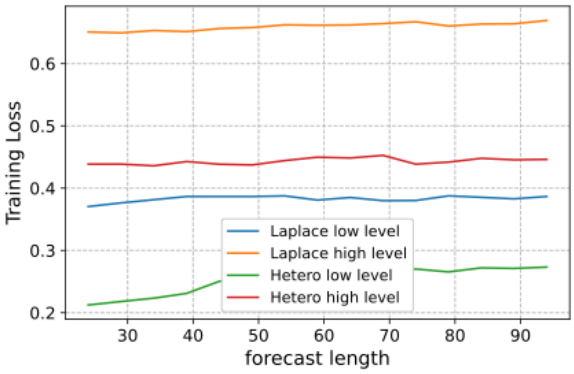


Fig. 5 *Influence forecast length on training loss across different datasets.*

The results in Figure 5 highlight the stability and robustness of our Diffusion Model when varying the forecast horizon, especially beyond 48-time steps. These findings confirm that the Diffusion Model is resilient to both noise levels and forecast length, indicating that neither factor substantially influences its forecasting performance.

7.2 FORECASTING EVALUATION ON SYNTHETIC DATASETS

We evaluated the forecasting task using all metrics described in Section 6.2. For this evaluation, each synthetic dataset was split into a training set and a testing set with a 70:30 ratio: $X_{m,1:7000}$ was used for training and $X_{m,7001:10000}$ for testing. The time series was further divided into input sequences of length 24 and forecast windows of length 24. Note that LSTM does not specifically address the concept of long-horizon forecasting, typically defined as forecasting more than 48 future time steps [22]. Therefore, we set the forecast horizon to 24 steps to ensure a fair comparison between LSTM and our proposed method. The results are shown in Table 1.

As a result, LSTM performs better on deterministic metrics, such as MAE and MSE, across most datasets, indicating a strong average point-wise accuracy. However, our Diffusion Model outperforms LSTM on distribution-aware metrics, such as MAPE, sMAPE, and CRPS, which better capture relative error and the quality of probabilistic forecasts.

For the Laplace low-level noise dataset, LSTM achieves the lowest values across all metrics except CRPS. This is because of the relatively low noise intensity, which allows LSTM to fit the data well without being significantly affected by outliers. In such low-noise conditions, the benefits of explicitly modeling uncertainty, as done by Diffusion Models, are less significant.

In contrast, for the Laplace high-level noise dataset and both low- and high-level heteroskedastic datasets, while LSTM still performs well on MSE and MAE, it struggles with percentage-based and probabilistic metrics. On the other hand, our Diffusion Model, shows greater robustness in handling heavy-tailed noise and time-varying variance. Lower MAPE, sMAPE, and CRPS scores demonstrate its strength in capturing uncertainty and generating reliable forecasts under complex patterns or non-Gaussian conditions.

In summary, LSTM can provide robust forecasting in low-noise, deterministic settings, while the diffusion model provides superior performance in environments characterized by high uncertainty, outliers, or non-stationary behavior.

7.3 FORECASTING EVALUATION ON REAL-WORLD DATASETS

We evaluated the forecasting task on the real-world water quality datasets using the same evaluation procedure described in Section 7.2, ensuring consistency with the synthetic dataset experiments by splitting each time series into a 70:30 ratio for training and testing. However, for the real-world data, each sequence was segmented into input windows of length 24 (representing one day) and forecast horizons of length 24 (also one day). The results are shown in Table 2.

Notably, we selected four stations from the original files because their time series contained long, uninterrupted measurement periods without missing values.

As shown in Table 2, the forecasting performance for the temperature variable closely resembles the trends observed in the evaluation on synthetic datasets. However, we observe a notable increase in error across all metrics for both the LSTM and the diffusion model for the TDS variable. This performance degradation is due to the lack of clear periodic patterns in the TDS time series, features that are typically crucial for accurate forecasting. In the absence of such regularities, both models struggle to capture meaningful temporal dependencies. To better understand this scenario, we plan to conduct a more in-depth investigation of TDS forecasting. Moreover, we will collaborate with domain experts familiar with the characteristics of this dataset.

Table 1. *Forecasting performance comparison on synthetic datasets. The best-performing values for each metric and dataset are both bolded and underlined.*

		Metrics				
Dataset Name	**Methods**	**MSE**	**MAE**	**MAPE**	**sMAPE**	**CRPS**
Laplace Low-level	LSTM	**<u>0.094</u>**	**<u>0.223</u>**	**<u>431.736</u>**	**<u>50.760</u>**	0.382
	Our Diffusion Model	0.104	0.268	461.119	67.873	**<u>0.268</u>**
Laplace High-level	LSTM	**<u>0.552</u>**	**<u>0.538</u>**	592.060	288.716	0.893
	Our Diffusion Model	0.825	0.737	**<u>512.220</u>**	**<u>125.711</u>**	**<u>0.737</u>**
Hetero Low-level	LSTM	0.024	0.117	3613.506	**<u>26.215</u>**	0.186
	Our Diffusion Model	**<u>0.021</u>**	**<u>0.116</u>**	**<u>131.861</u>**	32.448	**<u>0.116</u>**
Hetero High-level	LSTM	**<u>0.137</u>**	**<u>0.274</u>**	322.293	74.816	0.426
	Our Diffusion Model	0.156	0.329	**<u>175.297</u>**	**<u>50.106</u>**	**<u>0.341</u>**

Table 2. *Forecasting performance comparison on the water quality dataset from the Chao Phraya River. The best-performing values for each metric and dataset are both bolded and underlined.*

		Metrics				
Dataset Name	**Methods**	**MSE**	**MAE**	**MAPE**	**sMAPE**	**CRPS**
Port Authority of Thailand (Temp)	LSTM	**<u>1.829</u>**	**<u>1.119</u>**	3.646	3.730	1.077
	Our Diffusion Model	2.400	3.295	**<u>3.533</u>**	**<u>3.247</u>**	**<u>1.067</u>**
Port Authority of Thailand (TSD)	LSTM	**<u>41167.231</u>**	**<u>185.934</u>**	**<u>60.144</u>**	**<u>87.602</u>**	**<u>178.321</u>**
	Our Diffusion Model	43563.840	243.917	68.953	96.413	225.644
South Bangkok Power Plant (Temp)	LSTM	**<u>1.641</u>**	**<u>1.073</u>**	**<u>3.472</u>**	3.554	**<u>1.070</u>**
	Our Diffusion Model	2.496	2.852	5.138	**<u>3.112</u>**	1.667
South Bangkok Power Plant (TSD)	LSTM	**<u>12292.982</u>**	**<u>101.482</u>**	50.031	67.366	94.929
	Our Diffusion Model	28298.888	256.433	**<u>23.740</u>**	**<u>39.883</u>**	**<u>56.384</u>**
Khlong Lat Pho (Temp)	LSTM	**<u>1.959</u>**	**<u>1.069</u>**	**<u>3.364</u>**	**<u>3.460</u>**	**<u>1.129</u>**
	Our Diffusion Model	4.432	9.335	5.347	5.871	1.959
Khlong Lat Pho (TSD)	LSTM	**<u>16867.505</u>**	**<u>129.609</u>**	62.056	90.028	124.178
	Our Diffusion Model	40286.720	456.583	**<u>55.106</u>**	**<u>33.515</u>**	**<u>90.513</u>**
Phra Phuttha Yodfa Bridge (Temp)	LSTM	2.383	1.332	4.247	4.371	1.185
	Our Diffusion Model	**<u>2.070</u>**	**<u>1.666</u>**	**<u>3.507</u>**	**<u>3.386</u>**	**<u>1.133</u>**
Phra Phuttha Yodfa Bridge (TSD)	LSTM	**<u>3889.306</u>**	**<u>61.969</u>**	36.222	44.312	54.384
	Our Diffusion Model	19529.363	95.215	**<u>35.697</u>**	**<u>22.099</u>**	**<u>51.333</u>**

8. DISCUSSION AND CONCLUSION

In this paper, we adopted a diffusion model for time series forecasting and provided a detailed evaluation in the context of water quality monitoring in the Chao Phraya River. Overall, the diffusion model offers a promising alternative by learning a generative process that captures complex data distributions in a probabilistic and flexible manner.

On synthetic datasets, we were able to generate sequences with high noise levels, allowing us to evaluate the robustness of the model under challenging conditions. The results show that the diffusion model achieves higher CRPS scores compared to LSTM, indicating that it can more effectively model the uncertainty and generate forecasts that are closer to the true distribution.

However, when evaluating using deterministic metrics such as MSE and MAE, the LSTM baseline outperformed the diffusion model. This is likely because LSTM is optimized specifically for point predictions, whereas the diffusion model is designed to generate a distribution over possible future values, which may not always yield the lowest average error in pointwise comparisons.

These findings highlight the trade-off between probabilistic accuracy and pointwise prediction accuracy. While LSTM excels in minimizing direct prediction error, diffusion models provide richer forecasts that better reflect the underlying uncertainty and variability in time series data.

9. ACKNOWLEDGEMENT

We would like to thank the Metropolitan Waterworks Authority (MWA) of Thailand for providing access to valuable open data, which has been instrumental to the success of this research.

REFERENCES

[1] J. L. Alcaraz and N. Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learn-ing Research*, 2022.
[2] M. Chen, S. Mei, J. Fan, and M.Wang. An overview of diffusion models: Applications, guided genera-tion, statistical rates and optimization. *arXiv pre-print arXiv:2404.07771*, 2024.
[3] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.
[4] M. Kollovieh, A. F. Ansari, M. Bohlke-Schneider, J. Zschiegner, H. Wang, and Y. B. Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 28341–28364, 2023.
[5] Y. Li, X. Lu, Y. Wang, and D. Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. In *Advances in Neural Information Processing Systems*, pp. 23009–23022, 2022.
[6] Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, and Q. Wen. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6555–6565, 2024.
[7] H. Lim, M. Kim, S. Park, and N. Park. Regular time-series generation using SGM. *arXiv preprint arXiv:2301.08518*, 2023.
[8] L. Lin, Z. Li, R. Li, X. Li, and J. Gao. Diffusion models for time-series applications: a survey. *Frontiers of Information Technology & Electronic Engineering*, 25(1):19–41, 2024.
[9] Y. Liu, T. Ajanthan, H. Husain, and V. Nguyen. Self-supervision improves diffusion models for tabular data imputation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 1513–1522, 2024.
[10] C. Luo. Understanding diffusion models: A unified perspective.*arXiv preprint arXiv:2208.11970*, 2022.
[11] C. Meijer and L. Y. Chen. The rise of diffusion models in time-series forecasting, *arXiv preprint arXiv:2401.03006*, 2024.
[12] A. H. Murphy and R. L. Winkler. A general framework for forecast verification. *Monthly Weather Review*, 115(7):1330 – 1338, 1987.
[13] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8162–8171, 2021.
[14] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pp. 8857–8868, 2021.
[15] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 2256–2265, 2015.
[16] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing systems*, volume 27, pp. 3104 – 3112, 2014.
[17] Y. Tashiro, J. Song, Y. Song, and S. Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24804–24816, 2021.
[18] V. T. Truong, L. B. Dang, and L. B. Le. Attacks and defenses for generative diffusion models: A comprehensive survey. *ACM Comput. Surv*., 57(8), 2025.
[19] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun. Transformers in time series: a survey. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, 2023.
[20] H. Wu, J. Xu, J. Wang, and M. Long. Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing systems*, pp. 22419 – 22430, 2021.
[21] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surv.*, 56(4), 2023.
[22] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, 2021.
[23] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 27268–27286, 2022.

Woraphon Lilakiatsakun is currently the Director of the Center of Excellence in Security at Raja-mangala University of Technology Tawan-ok (2023–present). He earned his Ph.D. in Electrical and Tele-communication Engineering from the University of New South Wales, Australia, in 2005. Dr. Lilakiatsakun's research interests include artificial intelligence, telecommunication, and security technologies .

Thanapol Phungtua-eng received his Ph.D. in Informatics from Shizuoka University, Japan, in 2024. He is currently an Assistant Professor at the Faculty of Business Administration and Information Technology, Rajamangala University of Technology Tawan-Ok, Chakra-bongse Bhuvanarth Campus, Bangkok. His main research interests include data mining and machine learning approaches, particularly for anomaly detection, forecasting, and time series decomposition. He is also involved in transient pattern analysis in time-domain astrophysics.

Werachart Muttitanon is currently a lecturer in Information Technology at Rajamangala University of Tech-nology Tawan-ok (2013–present). He earned his Ph.D. in Information Technology Management from Mahidol University, Thailand, in 2019. Dr. Werachart Muttitanon's research interests include computer networks, information technology management, and information security.

Saowakhon Nookhao holds a Ph.D. in Information Technology Manage-ment from Mahidol University. She is currently a lecturer in the Digital Transformation and Technology Management Program at the KMITL Business School, King Mongkut's Institute of Technology Ladkrabang. Her acade-mic and research interests focus on Management Information Systems, Business Intelligent, and Information Technology Management.