2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

DDN-GP: Estimating Regression Predictive Distributions With Missing Data

Chaoran Pang[1,2] , Hua Wang[3], Shikun Tian[1,2], Chen Chen[3], Wu Xu[1,*] and Lin Wang[2,1,*]

[1] Shandong Key Laboratory of Ubiquitous Intelligent Computing, University of Jinan, Jinan 250022, China
[2] QuanCheng Laboratory, Jinan 250103, China
[3] Shandong Institute for Product Quality Inspection, Jinan 250022, China
wuxvforever@foxmail.com
wangplanet@gmail.com

Abstract. Probability density estimation in time series often encounters missing values, which compromise data completeness and usability, making it difficult to accurately estimate distributions and leading to biased results. To address this challenge, we propose a novel probability density estimation method called DDN-GP, which introduces Gaussian Process(GP) to Deconvolutional Density Networks(DDN). This approach uses a nonlinear dimensionality reduction approach, employing GP in the latent space to handle missing input data, and takes advantage of DDN to estimate arbitrarily distributed times in time series even with missing output data, ultimately improving both prediction accuracy and model robustness. We validate DDN-GP on multiple datasets with missing data, and the experimental results demonstrate that our approach enhances predictive performance quantification compared to existing methods.

Keywords: Probability Density Estimation, Time Series, Missing Data, Gaussian Processes.

1 Introduction

A regression task involves predicting an output given an input x, it can be viewed as modeling the conditional probability distribution of the output random variable given the input, which is donated as $p(f(x)|x)$, where is the random variable $f(x)$ associated with each x, and $p(\cdot)$ denotes the corresponding probability distribution [1]. This finds applications in financial market forecasting, clinical monitoring, and cement hydration analysis.

However, in time series regression tasks, the data is commonly subject to missing values challenges due to data collection cost, such as sensor failures, data collection delays, or system storage anomalies [2], and the data is inherently sequential, meaning that variables are correlated with one another, making the presence of missing values more complicated to handle. Missing data reduces the overall completeness and con-

* Corresponding authors: L. Wang and X. Wu.

sistency of the time series, which can introduce significant biases in probability estimation and obscure the inherent relationships among variables, further complicating the task of modeling the underlying distributions.

In the context of time series, this missing data problem can arise in two primary forms: (1) Missing output data: where certain or all observations of the target variable are unrecorded. In this case, the model cannot be fully trained or properly evaluated, leading to potential biases in the estimation of the target variable's probability distribution. (2) Missing input data: where essential features or external explanatory variables are missing. Missing input data limits the model's ability to capture the full range of relationships among variables, which undermines the accuracy of estimating the conditional probability distribution and degrades predictive performance. Therefore, dealing with missing data in time series regression tasks requires strategies that can effectively address both types of missing information.

Time series can be understood as being influenced by multiple correlated data channels, each contributing unique information to the overall prediction. In the presence of missing data, several strategies can be employed to mitigate its adverse effects. One strategy involves leveraging inter-channel correlations: a common technique for handling missing inputs is to perform dimensionality reduction [3],[4] and then use the relationships among different channels in the reduced-dimensional space to impute or estimate the missing parts. By capturing latent coupling relationships in the reduced-dimensional space, one can more effectively tackle input data gaps while preserving a robust depiction of the system's dynamic evolution.

Another strategy is leveraging intra-channel correlations. When only a small number of samples are available and some are missing, one typically relies on assumptions of distributional smoothness to construct suitable priors or regularization strategies, thereby enhancing the robustness of distribution estimation. Furthermore, because real-world data often exhibit complex, multimodal characteristics [5], it is common to impose minimal constraints on the model and adopt a free-form probability distribution approach, which avoids the biases that could otherwise arise from single-mode assumptions.

Unfortunately, current methods often excel in only one particular aspect. Gaussian Processes [6] as a classical temporal statistical model, typically capture temporal dependence with kernel functions, yet it rely on complete data for kernel computation and parameter estimation, missing values disrupt their covariance structure and reduce prediction accuracy. Meanwhile, probability density estimation methods (e.g. Deconvolutional Density Network (DDN) [7]) achieve more robust distribution learning in settings with limited samples or localized regions through smooth priors, but it is difficult to construct sequential dependence.

To address the above limitations, this paper proposes DDN-GP, which integrates Gaussian Process(GP) with the existing free-form probability estimation method DDN. By leveraging nonlinear dimensionality reduction to project data with missing values into a latent space free of missing values, we incorporate the GP into DDN to capture the dynamic temporal pattern, thereby constructing temporal dependencies based on

DDN's ability to construct robust distributions and compensating for data incompleteness caused by missing values in the GP. Ultimately, this approach enables robust time series probability density estimation with missing data.

Our contributions are summarized as follows:

(1) We propose a framework DDN-GP for handling missing values in time series probability estimation. It maps incomplete data into a latent space, models temporal dynamics there, and then uses a density estimator to produce the final distribution, effectively addressing missing data issues.

(2) We propose a strategy that models temporal dynamics in probability density estimation tasks by employing a Gaussian Process (GP) in the latent space, effectively enhancing probability estimation performance in time series data.

(3) We evaluate the performance of DDN-GP on datasets covering diverse scenarios. Experimental results demonstrate that DDN-GP can effectively handles missing data and estimate and outperforms comparison methods.

2 Related Work

2.1 Conditional Density Estimation For Regression

In time series analysis, the challenge of missing data has always been a persistent issue. Early methods relied on heuristics, such as mean imputation [8], linear interpolation [9], or forward filling [10]. These methods are simple and interpretable, and they continue to be used in practice to this day.

2.2 Probability Density Estimation Methods

Regression imputation methods [11] use regression models to handle missing data, assuming that a relationship exists within the data. With the development of probabilistic models, probability density estimation methods have been increasingly applied due to their excellent ability to estimate uncertainty. Examples include Variational Autoencoders (VAEs) [12],[13] and Generative Adversarial Networks (GANs) [14],[15]. These models can estimate uncertainty effectively; however, challenges arise from the multi-modality and sparsity of the data. Meanwhile, Deconvolutional Density Networks (DDNs) [7] can propose free-form distributions even with sparse data, but they do not account for time dynamics.

2.3 Gaussian Processes

Gaussian processes (GPs) [16], as a classical Bayesian inference method, have been widely applied to tasks such as time series regression. GPs can be viewed as a collection of infinitely many random variables [17], and it directly estimates regression probabilities through its mean and covariance functions. By leveraging Bayesian inference, it can perform well even with limited data. However, the high computational complexity—mainly due to the need to invert and decompose large covariance matrices—restricts its direct application to large-scale data and complex problems. To overcome this, extended forms like sparse Gaussian processes [18] and deep Gaussian processes [19] have been developed. These methods approximate the original Gaussian process by introducing inducing points or multi-layer structures, significantly reducing computational demands while preserving the model's flexibility and expressive power.

2.4 Gaussian Processes For Probability Density Estimation

Gaussian processes (GPs) can be directly employed in probability density estimation to construct data dependence. The Gaussian Process Conditional Density Estimation (GP-CDE) [20] using GPs as the decoder to map the extended inputs into samples from the conditional distribution, enabling probability estimation in small-sample scenarios, albeit still under the Gaussian assumption. Meanwhile, Non-Gaussian Gaussian Processes (NGGP) [21] uses normalizing flows to model the Gaussian process posterior as an arbitrary non-Gaussian distribution, but its reliance on meta-learning, which requires large amounts of domain data, limits its broader applicability.

In addition, GPs employed in latent space are an important method. GPLVM [22] and its variations [23],[24],[25],[26] use Gaussian processes to construct a nonlinear latent space, providing an effective tool for dimensionality reduction and feature extraction in high-dimensional data. GP-VAE [27] also applies nonlinear dimensionality reduction and constructs GPs in the latent space to handle missing data, particularly in time series scenarios, but it is difficult to construct the distribution with strong generalization ability in sparse data situations.

3 Formulation

We aim to address the task of modeling regression probability distribution in time series scenarios with missing data. Existing normal statistical methods like Gaussian Processes, which catch temporal dependence and perform well when data is abundant, tend to degrade significantly in the presence of missing data, and probability density estimation models DDN can construct arbitrary distribution even with missing data, but it isn't concern time dynamic. To overcome these challenges, we leverage nonlinear dimensionality reduction techniques to map input data with missing values into a latent space. We then employ Gaussian Processes (GPs) to model the dynamics in this latent space, compensating for missing information and obtaining a smoother latent representation. Additionally, we incorporate a DDN decoder, which exploits inter-channel smoothness to construct a piecewise constant probability density estimate through multi-scale Deconvolutional operations. This enables accurate estimation of the target variable's distribution, even when data is severely missing. We train the DDN-GP using Cross-Entropy as the loss function and apply a variational inference method to train the latent space, ensuring both predictive accuracy and providing a measure of prediction uncertainty.

4 Methodology

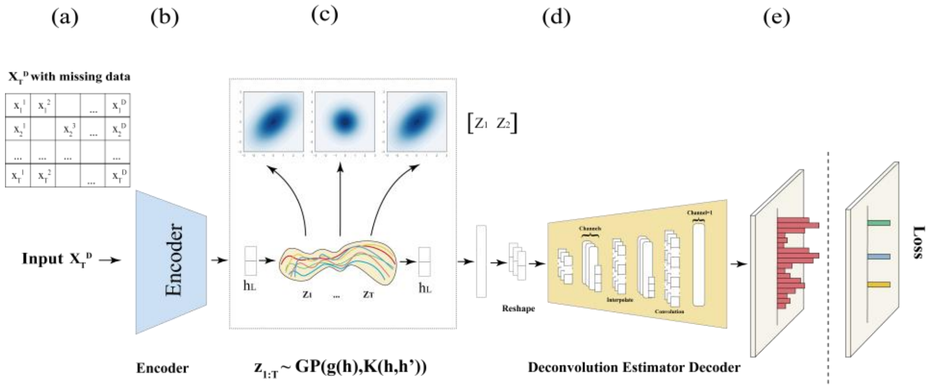

Fig. 1. The architecture of DDN-GP. (a) is the inputs with missing values. (b) is the Encoder of the model, which realize nonlinear reduction dimensions to map inputs with missing values into the latent space free of missing values. (c) is the Gaussian Processes Latent Space of the model, which simulates time dynamic. For the sake of presentation, we use a two-dimensional case $[z_1, z_2]$. (d) is the Deconvolutional Estimator in DDN. The estimator can effectively model free-form distributions with smooth prior. (e) is the Cross-Entropy loss of the network.

4.1 Problem Setting And Notation

Here we briefly describe the baseline model, and other components will be added into it progressively. Considering a dataset D={(X_t,y_t)|t=1,2...T}, where T is the time length of the sample. For the k-th sample, X_t^D epresents the multi-dimensional feature input, and y_t represents the corresponding label.

We assume that any subset of the data features X_t^d may be missing, and in extreme cases, the entire feature vector X_t could be absent. Consequently, y_t may also be missing. Our goal is to characterize the distribution $p(y_{\{1:T\}}|X_{\{1:T\}})$. In other words, for each time step t, we only have access to the observed portion X_t^o of X_t , which could even be empty, and we seek to model how the random variable y_t is distributed given this partial information, while leveraging time dependence.

The model is based on an Encoder-Estimator architecture. Next, we will provide a detailed overview of each component.

4.2 Encoder

Let $X_{\{1:T\}}^d$ be the input of the model, the vector X can represent it, and $y_{\{1:T\}}$ be the target samples of the input X, we use y to represent, where the feature of X_t^{d} may be missing, a straightforward approach is to fill in these missing values (for example, with zeros) and then apply a probability estimation model.

However, this can cause two samples with different missing-value patterns to appear very dissimilar, even though their ground truth might actually be quite similar. Therefore, we employ nonlinear dimensionality reduction methods to map the missing data space into a latent space. Specifically, we use an encoder with hidden states $h = f_{enc}(X)$ to obtain deterministic latent representations of the input, where the f_{enc} is a transition function implemented by LSTM [28] or Convolutions [29].

To overcome the missing data problem, it is reasonable to transform the deterministic latent representations into random variables. Because the features of the data are correlated, the latent representations must capture these correlations and use them to compensate for missing features. Accordingly, we map each h to a latent variable z by applying a stochastic mapping function $g(\sim)$ to model the dynamic.

$$p(z|h)=g(h)+\epsilon \tag{1}$$

Where $\epsilon\sim N(0,\sigma^2 I)$ is a Gaussian Noise, In addition, to reduce the complexity of the representation and mitigate potential overfitting, we should model the latent space in a smoother and more interpretable manner than the raw data space, this is represented as Fig 1(b).

Under this framework, the vanilla Deconvolutional Density Network (DDN) can be viewed a special case. In the vanilla DDN, the VL layer enforces that the latent variables are independent of each other, making it difficult to capture the correlation among features.

4.3 Gaussian Process Latent Variables

The learning of $g(\sim)$ is the essential of the model, that need to satisify the follow points: (1) It should be capable of incorporating variations in feature correlations, thereby accurately capturing the dynamic dependencies among features and effectively compensating for missing data. (2) The latent space must be smooth and interpretable, ensuring that transitions within this space are gradual and the model's behavior remains explainable.

In this work, we propose to learn the stochastic function $g(\sim)$ using a nonlinear approach with a functional prior defined by Gaussian Processes. As shown in Fig (1), we sample $g(h)$ from a GP prior, which not only allows for the independent filling of missing values but also effectively captures the correlations among features. Moreover, the latent space induced by the Gaussian Processes consists of correlated variables, meaning that a change in one variable (e.g., z_2). This property is particularly beneficial for completing features in the latent space when handling missing data. We define the stochastic function $g(h)$ according to a GP prior:

$$g(h)\sim GP(m(h),K(h,h')) \tag{2}$$

with the mean function $m(h)$ and the covariance function $k(h, h')$ as:

$$m(h) = h \tag{3}$$

$$K(h,h') = v^2 \frac{||h-h\prime||^2}{2r^2} \tag{4}$$

where h denotes a deterministic latent state and h' represents another latent state, The parameter υ controls the average distance of the $g(\sim)$, and r regulates the correlation among the random variables, increasing r strengthens the correlation between z and z'.

In this work, we adopt a semi-parametric Gaussian Process prior for the latent space, where $m(h)$ is generated from a parametric function $g(h)$ and the covariance function $k(h,h')$ is non-parametric. This approach not only introduces correlations among the random variables to address missing features but also preserves the inherent characteristics of the original data space, this is represented as Fig 1 (c).

4.4 Decoder with Deconvolutional Estimator

In this section, we introduce the generative process for the conditional probability distribution $p(y_t|z_t)$ derived from the latent variables $z_{\{1:T\}}$. Due to the inherent sparsity and incompleteness of the observed y values, it is crucial to obtain accurate estimates even when sample sizes are limited.

An effective strategy for enforcing smoothness in the estimated distribution is to leverage the correlations between outputs. Specifically, we capture the inherent relationships among outputs through their multi-scale features and hierarchical structure, allowing us to extract consistent data patterns rather than merely focusing on random occurrences, the Deconvolutional estimator in DDN remains a powerful tool for free-form modeling. This estimator assumes that the domain D of variable y is finite and partitions it into a series of uniform bins B. By constructing a piecewise-constant probability density estimation using these bins and employing multi-channel, multi-scale deconvolution during the modeling process, we ensure that the resulting probability distribution is both smooth and precise.

The Decoder can be represented using a $f_{dec}(\sim)$, which divides the output into N Bins. The $p(y_t|z_t)$ can be represented as:

$$\mathrm{p}(\mathrm{y}_t|z_t) = \begin{cases} \cdots \\ \frac{f_{dec}(y_t|z_t)}{\triangle B} \quad if\ y_t\ \in B_i \\ \cdots \end{cases} \tag{5}$$

where B_i is the i-th Bin of N bins that pariationing the output space.

Specifically, the latent vector z_t is first mapped to a fully connected layer, which is then reshaped into a multi-channel initial feature map. This reconstructed feature map is subsequently fed into a sequence of Deconvolutional layers constructed using upsampling and convolution. Upsampling expands the feature map by an integer scale factor, while convolution transforms it by applying shared weights to introduce spatial correlations. After these Deconvolutional layers, an unnormalized logit vector is obtained, which is finally passed through a SoftMax layer to produce a discretized probability vector, this is represented as Fig 1 (d).

4.5 Training DDN-GP

The likelihood of $P(y_t|X)$ is computed by the output of the Deconvolutional estimator and the target y_t, the estimator receives the input X. Additionally, the variational loss L_{KL} is computed within the latent space of Gaussian processes.

$$-\frac{1}{T}\sum_{t=0}^{T} E\left[\frac{1}{J}\sum_{j=0}^{N}\log\left(g(y_t|z_t)\right)\cdot 1(y_t)\right] - \beta L_{KL} \tag{6}$$

where T denote the number of samples and N the number of bins, while β is a hyperparameter that balances the reconstruction and regularization terms. Inspired by previous research, we employ variational inference [30] for latent space training. Specifically, we approximate the true posterior $p(z|h)$ with the variational posterior $q_\phi(z|h)$ by maximizing the evidence lower bound (ELBO) of the marginal log-likelihood.

$$L_{KL} = [q_\phi(z|h)||p(z|h)] \tag{7}$$

Thereforce, we have the ultimate optimization goal:

$$argmin\{-\frac{1}{T}\sum_{t=0}^{T} E\left[\frac{1}{J}\sum_{j=0}^{N}\log\left(g(y_t|z_t)\right)\cdot 1(y_t)\right] - \beta[q_\phi(z|h)||p(z|h)]\} \tag{8}$$

5 Experiments

5.1 Datasets With Missing Data

Toy And Real Regression Datasets To validate the effectiveness of DDN-GP, we employed four regression datasets: one toy dataset and two real-world regression datasets and four time series datasets, we set different data missing rate [10%,30%,50%] to evaluate performance using the negative log-likelihood (NLL) metric.

Testing across these datasets allows for a comprehensive evaluation of the performance of DDN-GP in handling various data types and missing value scenarios. We used Adam optimizer with the default confguration. The learning rate was set to 1e-3, and the batch size is 32.

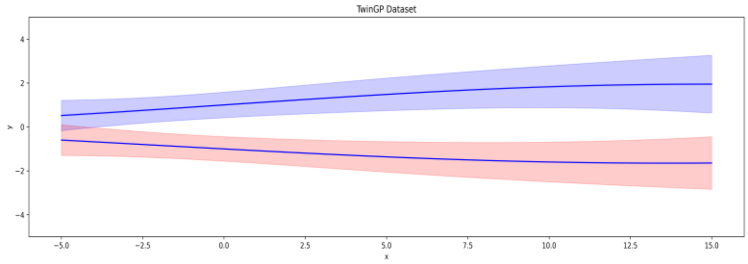

Fig. 2. TwinGP Dataset Probability Distribution. The above and the bottom distributions are composed by Gaussian Process with RBF kernel, the blue lines represents the mean of Gaussian Processes, and the regions are confidence interval.

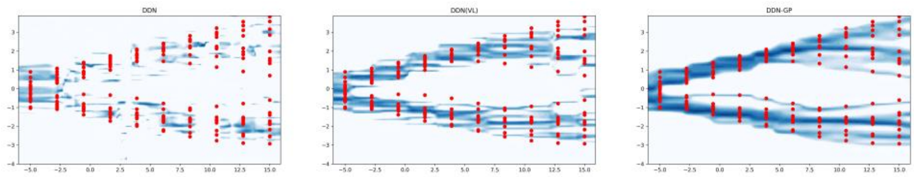

Fig. 3. The experimental results of DDN-GP comprised with DDN, DDN(VL) in TwinGP Dataset. The left is DDN, the middle is DDN(VL), and the right is DDN-GP. The red points in the figures represents the training data, and the blue regions are the probability estimation results of models. The hyperparameter of β is [0,0.3,0.3].

5.2 Effectiveness Verification

Models In all experiments, because the part of input data is missing, we set mask for missing features. The models of all experiments begin with input layer and two hidden layers, the num of hidden neurons is [32,64], and we use BatchNorm for every hidden layer for batch-normalization. For Gaussian Process in the latent space, we use RBF kernel [31] and set the hyperparameter $\ell = 1$, $\sigma = 1$. The estimator of DDN contained two Upsample-Conv-BatchNorm-LeakyReLU blocks and one Upsample-Conv block in each target dimension. We confgured the blocks so that the output space was partitioned into N = 68 bins. We used 16 latent codes in latent variables. In addition, the target range of y is important, For the toy TwinGP dataset , we set [-4,4], and for real world datasets, we set [y_{min}−1,y_{max}+1].

Table 1. Experimental Performance (NLL), β=0.3, Missing rate = 30%.

Datasets	N	D_{in}	D_{out}	DDN-GP	DDN(VL)	DDN
TwinGP	2000	1	1	**-0.7987±0.0021**	-0.7813±0.0017	-0.7791±0.0010
license_plates_acution_data	190	3	1	**-1.0137±0.0013**	-0.8843±0.0010	-0.3256±0.0003
daily_gold_rate	10115	5	1	**-1.0148±0.0011**	-0.9987±0.0012	-0.3125±0.0002
Concrete_Data Dataset	1031	8	1	**-3.2946±0.0023**	-3.1196±0.0009	-3.0123±0.0004
Air Quality Dataset	9358	12	1	**-2.9120±0.0006**	-2.8978±0.0003	-0.1299±0.0002
Boston	507	13	1	**-3.7294±0.0027**	-3.5484±0.0011	-2.8390±0.0008
Financial Distress	3673	83	1	**-1.0377±0.0019**	-0.9812±0.0014	-0.2197±0.0006

Ablation Experiments To assess the capability of Gaussian Processes in the latent space of DDN, we conducted ablation experiments by comparing DDN, DDN(VL), and DDN-GP on datasets with missing data. DDN is the version without a probabilistic latent space; we directly set the latent mapping using MLPs. DDN(VL) is the DDN augmented with a Variational Layer [7], which can be interpreted as a Gaussian Process with an independent kernel that does not model relationships between variables. DDN-GP, on the other hand, incorporates Gaussian Process latent variables. Both DDN(VL) and DDN-GP were trained using variational inference methods. In the variational bottleneck [30], we set the hyperparameter β = 0.3. We conducted experiments on the

Twin Dataset (see Fig. 2). A small portion of the data was retained as observed values, while the remaining data was treated as missing, We test each dataset 5 times to take the negative log-likelihood(NLL) mean as the evaluation metric.

Table 2. Experimental Performance In Different Missing Rate(NLL)

Datasets	DDN-GP	GP-VAE	GP
TwinGP(10%)	**-0.7977±0.0025**	-0.7812±0.0020	5.0016±0.0002
TwinGP(30%)	**-0.7987±0.0021**	-0.7951±0.0017	4.9543±0.0002
TwinGP(50%)	**-0.7614±0.0032**	-0.7587±0.0028	4.9709±0.0002
TwinGP(70%)	**-0.7215±0.0036**	-0.7911±0.0031	4.9543±0.0001
license_plates_acution_data(10%)	**-1.1293±0.0009**	-1.1136±0.0014	2.2761±0.0003
license_plates_acution_data(30%)	**-1.0137±0.0013**	-0.8843±0.0010	2.3256±0.0003
license_plates_acution_data(50%)	**-2.0981±0.0035**	-2.0978±0.0023	2.9813±0.0002
daily_gold_rate(10%)	-1.1431±0.0008	**-1.2081±0.0010**	2.3121±0.0003
daily_gold_rate(30%)	**-1.0148±0.0011**	-0.9987±0.0012	2.3125±0.0002
daily_gold_rate(50%)	**-0.9341±0.0015**	-0.9562±0.0008	2.3341±0.0002
Concrete_Data(10%)	-3.1132±0.0016	**-3.2511±0.0013**	2.9124±0.0003
Concrete_Data(30%)	**-3.2946±0.0023**	-3.1196±0.0009	3.0019±0.0004
Concrete_Data(50%)	**-2.9813±0.0031**	-2.8611±0.0027	3.2371±0.0002
Air Quality Dataset(10%)	-3.0125±0.0013	**-3.1266±0.0010**	3.5126±0.0002
Air Quality Dataset(30%)	**-2.9120±0.0016**	-2.8978±0.0013	4.1299±0.0002
Air Quality Dataset(50%)	**-2.0981±0.0025**	-2.0978±0.0019	4.9813±0.0003
Boston(10%)	**-3.6809±0.0023**	-3.5612±0.0021	2.3156±0.0003
Boston(30%)	**-3.7294±0.0027**	-3.5484±0.0011	2.1352±0.0008
Boston(50%)	**-3.6146±0.0030**	-3.5781±0.0027	2.4376±0.0002
Financial Distress(10%)	**-1.2091±0.0015**	-1.1392±0.0006	-0.2027±0.0004
Financial Distress(30%)	**-1.0377±0.0019**	-0.9812±0.0014	-0.2197±0.0006
Financial Distress(50%)	**-0.7612±0.0031**	-0.7601±0.0010	-0.1892±0.0003

Contrast Experiments A similar model to DDN-GP is GP-VAE [27], which maps missing data to a latent space using an Encoder–Decoder architecture and leverages Gaussian processes to capture sequence dependencies. It is particularly well-suited for time-series tasks. We set 2Conv-layers-Encoder, the Gaussian Processes latent dim=2, and use RBF kernel, 2MLPs-layers-Decoder [7]. To compare the performance of DDN-GP and GP-VAE, we selected four different time series datasets, including Air Quality Dataset from UCI, daily_gold_rate, license_plates_acution_data, Financial Distress, these records multitemporal measurements across various variables and exhibits clear temporal dynamics. The quantitative comparison in Table 2 shows that although both models map missing data into a Gaussian Process latent space and perform comparably on temporal tasks, DDN-GP can construct arbitrary-form distributions with enhanced generalization capabilities using a Deconvolutional estimator. Furthermore, DDN-GP adapts to unknown data patterns and, as the missing rate increases, demonstrates robust adaptability to arbitrary probability distribution compared to GP-VAE. Gaussian Process(GP) [16] is a classical statistical method that estimates time series probability distributions and is particularly well-suited for sparse datasets. To validate the performance of DDN-GP and GP, we conducted experiments comparing DDN-GP with a standard GP model (using an RBF kernel) and employed 100 inducing points [18]. The results indicate that the performance of GP is lower than that of DDN-GP. This is because, under conditions of missing data and strict distributional constraints, the standard GP struggles to accurately model the underlying data distribution. In contrast, DDN-GP, with its ability to estimate arbitrary probability distributions, proves to be more robust and adaptable in such scenarios.

6 Conclusion

This paper addresses the challenges of missing data in time series regression tasks and the difficulty of accuracy with existing probability estimation methods. To this end, we propose a novel regression predictive distribution estimation model called DDN-GP. This model innovatively integrates nonlinear dimensionality reduction, Gaussian Process (GP) dynamic modeling, and Deconvolutional Density Network (DDN)-based free-form probability distribution estimation. DDN-GP successfully maps high-dimensional input data with missing values into a complete, missing-value-free latent space. In this latent space, it captures data dynamic using GP and ultimately constructs an arbitrary probability distribution using DDN.

Experimental results demonstrate that the DDN-GP model effectively handles missing input and output data, significantly reducing probability estimation bias. By leveraging correlations between feature channels and smoothness within channels, the model enhances predictive accuracy. Specifically, nonlinear dimensionality reduction allows the model to effectively impute missing values, avoiding the difficulties of directly dealing with incomplete data. The use of GP ensures predictive accuracy in data-rich regions, while DDN provides flexible probability distribution modeling capabilities, enabling it to adapt to complex data distributions.

In summary, the proposed DDN-GP model offers a novel approach and solution for regression predictive distribution estimation in the presence of missing data. It improves prediction accuracy and robustness, thereby providing a more reliable basis for decision-making. Future research directions may include further optimizing the nonlinear dimensionality reduction algorithm, exploring more effective smoothness prior strategies, and applying the model to a wider range of real-world applications, such as sensor data imputation and time series forecasting.

Acknowledgments. This study was supported by National Natural Science Foundation of China under Grant No. 61872419, No. 62072213. Shandong Provincial Natural Science Foundation No. ZR2022JQ30, No. ZR2022ZD01, No. ZR2023LZH015. Taishan Scholars Program of Shandong Province, China, under Grant No. tsqn201812077. The

"New 20 Rules for University" Program of Jinan City under Grant No. 2021GXRC077. Key Research Project of Quancheng Laboratory, China under Grant No. QCLZD202303. Research Project of Provincial Laboratory of Shandong, China under Grant No. SYS202201. University of Jinan Disciplinary Cross-Convergence Construction Project 2023 No. XKJC-202303, University of Jinan Disciplinary Cross-Convergence Construction Project 2024 No. XKJC-202402. National "Challenge-Based" Major Science and Technology Project in the Building Materials Industry Project No. 2023JBGS11-03. Key Research and Development Program of Shandong Province Grant No. 2024CXPT084. High-performance Computing Platform at University of Jinan.

References

1. Chen, Yunlu, and Nan Zhang.: Density regression with conditional support points. Technometrics 64(3) : 396-408(2022).
2. N. Mohammadi Foumani, L. Miller, C. W. Tan, G. I. Webb, G. Forestier, and M. Salehi.: Deep learning for time series classification and extrinsic regression: A current survey. ACM Computing Surveys 56(9), 1-45 (2024).
3. Ainsworth, Samuel K., Nicholas J. Foti, and Emily B. Fox.: Disentangled VAE representations for multi-aspect and missing data. arXiv preprint arXiv:1806.09060 (2018).
4. Ma, Chao, et al.: Eddi: Efficient dynamic discovery of high-value information with partial vae. arXiv preprint arXiv:1809.11142 (2018).
5. Harakeh, Ali, et al.: Estimating regression predictive distributions with sample networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, (2023).
6. Roberts, Stephen, et al.: Gaussian processes for time-series modelling. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 371(1984), (2013).
7. Chen, Bing, et al.: Deconvolutional density network: Modeling free-form conditional distributions. In: Proceedings of the AAAI Conference on Artificial Intelligence, (2022).
8. Patrician P A.: Multiple imputation for missing data. Research in nursing & health 25(1), 76-84(2002).
9. Kokaram A C, Morris R D, Fitzgerald W J, et al.: Interpolation of missing data in image sequences. IEEE Transactions on Image Processing 4(11): 1509-1519 (1995).
10. Lipton Z C, Kale D C, Wetzel R.: Modeling missing data in clinical time series with rnns. Machine Learning for Healthcare 56(56), 253-270 (2016).
11. Templ M, Kowarik A, Filzmoser P.: Iterative stepwise regression imputation using standard and robust methods. Computational Statistics & Data Analysis 55(10), 2793-2806 (2011).
12. Dalca A V, Guttag J, Sabuncu M R.: Unsupervised data imputation via variational inference of deep subspaces. arXiv preprint arXiv:1903.03503 (2019).
13. Ma C, Tschiatschek S, Palla K, et al.: Eddi: Efficient dynamic discovery of high-value information with partial vae. arXiv preprint arXiv:1809.11142 (2018).
14. Li S C X, Jiang B, Marlin B.: Learning from incomplete data with generative adversarial networks 1902, (2019).
15. Yoon, Jinsung, James Jordon, and Mihaela Schaar.: Gain: Missing data imputation using generative adversarial nets. In: International conference on machine learning, PMLR (2018).
16. Aigrain S, Foreman-Mackey D.: Gaussian process regression for astronomical time series. Annual Review of Astronomy and Astrophysics 61(1), 329-371 (2023).

17. Corani, Giorgio, Alessio Benavoli, and Marco Zaffalon.: Time series forecasting with gaussian processes needs priors. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham (2021).
18. Yang W, Feng Y, Wan J, et al.: Sparse gaussian process regression for landslide displacement time-series forecasting. Frontiers in Earth Science 10: 944301 (2022).
19. Wang, Yali, et al.: Sequential inference for deep Gaussian process. Artificial Intelligence and Statistics, PMLR (2016).
20. Dutordoir V, Salimbeni H, Hensman J, et al.: Gaussian process conditional density estimation. Advances in neural information processing systems 31, (2018).
21. Sendera, Marcin, et al.: Non-gaussian gaussian processes for few-shot regression. Advances in Neural Information Processing Systems 34, 10285-10298 (2021).
22. Lawrence, Neil.: Gaussian process latent variable models for visualisation of high dimensional data. Advances in neural information processing systems 16, (2003).
23. Titsias, Michalis, and Neil D. Lawrence.: Bayesian Gaussian process latent variable model. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, (2010).
24. Märtens, Kaspar, Kieran Campbell, and Christopher Yau.: Decomposing feature-level variation with covariate Gaussian process latent variable models. In: International Conference on Machine Learning, (2019).
25. Lalchand, Vidhi, Aditya Ravuri, and Neil D. Lawrence.: Gaussian Process Latent Variable Flows for Massively Missing Data. Third Symposium on Advances in Approximate Bayesian Inference.
26. Beckers, Thomas, and Sandra Hirche.: Prediction with approximated Gaussian process dynamical models. IEEE Transactions on Automatic Control 67(12), 6460-6473 (2021).
27. Fortuin V, Baranchuk D, Rätsch G, et al.: GP-VAE: Deep probabilistic multivariate time series imputation. In: Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, 1651-1661 (2020).
28. DiPietro, Robert, and Gregory D. Hager.: Deep learning: RNNs and LSTM. Handbook of medical image computing and computer assisted intervention. Academic Press, 503-519 (2020).
29. LeCun, Yann, et al.: Backpropagation applied to handwritten zip code recognition. Neural computation 1(4), 541-551 (1989).
30. Ranganath R, Gerrish S, Blei D.: Black box variational inference. In: Artificial intelligence and statistics. pp. 814-822. PMLR (2014).
31. Kanagawa, Motonobu, et al.: Gaussian processes and kernel methods: A review on connections and equivalences. arXiv preprint arXiv:1807.02582 (2018).