Applied Intelligence (2025) 55:483
https://doi.org/10.1007/s10489-024-06203-5

VAEneu: a new avenue for VAE application on probabilistic forecasting

Alireza Koochali[1,2,3] · Ensiye Tahaei[2] · Andreas Dengel[2,3] · Sheraz Ahmed[2]

Accepted: 13 December 2024 / Published online: 27 February 2025
© The Author(s) 2025

Abstract
This paper introduces VAEneu, a novel autoregressive method for multistep ahead univariate probabilistic time series forecasting, designed to address the challenges of generating sharp and well-calibrated probabilistic forecasts without assuming a specific parametric form for the predictive distribution. VAEneu leverages the Conditional VAE framework and optimizes the likelihood of the predictive distribution using the Continuous Ranked Probability Score (CRPS), a strictly proper scoring rule, as the loss function. This approach enables the model to learn flexible, sharp, and well-calibrated predictive distributions without the need for a tractable likelihood function. In a comprehensive empirical study, VAEneu is rigorously benchmarked against 12 baseline models across 12 datasets, demonstrating superior performance in both forecasting accuracy and uncertainty quantification. VAEneu provides a valuable tool for quantifying future uncertainties, and our extensive empirical study lays the foundation for future comparative studies for univariate multistep ahead probabilistic forecasting.

Keywords Probabilistic Forecasting · Generative models · Time series Analysis · Deep Learning

1 Introduction

Probabilistic forecasting is essential in decision-making, particularly in fields where accurately assessing risk is critical, such as healthcare [1–3], weather forecasting [4, 5], flood risk assessment [6], seismic hazard prediction [7, 8], renewable energy sector [9, 10], civil engineering [11, 12], and economic and financial risk management [13, 14]. These models provide insights into possible future outcomes and their likelihoods, aiding decision-makers in resource allocation, policy formulation, and strategic planning. The importance of these forecasts lies in their ability to model uncertainty about the future in the form of predictive distribution. The ultimate goal of a probabilistic forecaster is to output a predictive distribution with good calibration and sharpness. Calibration is concerned with the statistical consistency between predictive distribution and observation, while sharpness governs the confidence in the forecast itself. A sharp probabilistic forecaster outputs a narrow predictive distribution, which represents high confidence in the forecast. However, sharpness is desirable in conjunction with calibration to employ high confidence aligned with the true distribution [15].

The traditional probabilistic forecasting models like ARCH [16], GARCH [17–20], ensemble models [21] and Bayesian approaches [22, 23] often struggled with capturing complex, non-linear patterns and long-term dependencies in data. These models typically relied on strong assumptions about the underlying data distribution, such as linearity or stationarity, limiting their flexibility. The evolution of Deep Neural Networks (DNNs) has revolutionized probabilistic forecasting, introducing methods that efficiently leverage large datasets. This advancement has shifted the focus towards end-to-end learning, minimizing the need for manual feature engineering and opening new horizons in forecasting accuracy and application. Recurrent Neural Networks (RNNs) were the first neural network architectures that were designed

✉ Alireza Koochali
alireza.koochali@dfki.de

Ensiye Tahaei
seyede_ensiye.tahaei_shiyade@dfki.de

Andreas Dengel
andreas.dengel@dfki.de

Sheraz Ahmed
sheraz.ahmed@dfki.de

1 IAV GmbH, Trippstadterstr. 122, 67663 Kaiserslautern, Germany

2 DFKI GmbH, Trippstadterstr. 122, 67663 Kaiserslautern, Germany

3 Computer Science Department, University of Kaiserslautern-Landau, Erwin-Schrödingerstr 52, 67663 Kaiserslautern, Germany

especially to process sequential data. The Long-Short Term Memory (LSTM) [24] and Gated Recurrent Unit (GRU) [25] are two of the most well-established RNN variants in DNN-based forecasting models. Temporal Convolutional Neural Networks (TCNs) [26] have transformed the capabilities of CNNs to time series data. Seq2Seq models [27], later improved with Attention mechanism [28], further enhance time series modeling. Recently, Transformers [29] has added to the arsenal of time series modeling tools, which further empowered the richness of the probabilistic forecasting toolbox. While the DNN models have demonstrated remarkable performance for probabilistic forecasting, they mostly restrict themselves to the family of distribution with tractable likelihood regardless of true distribution (e.g. [30]). Another dominant approach is to model key properties of predictive distribution, such as quantile [31, 32] instead of modeling the predictive distribution itself. To circumvent the intractable likelihood, one newly emerging approach is to use GANs to model predictive distribution [33, 34]. In optimal scenarios, these models can generate samples from true data distribution. However, the instability of the adversarial objective function makes the training of these networks a daunting task.

In this work, we introduce VAEneu, an autoregressive probabilistic forecasting model that offers a novel approach to optimizing the likelihood of the predictive distribution without requiring a tractable likelihood function. Instead of assuming a parametric form for the predictive distribution, VAEneu minimizes a loss function based on the CRPS, which serves as a proxy for maximizing the likelihood. This approach grants VAEneu significant flexibility, as it can accurately learn any predictive distribution without being constrained by parametric assumptions. Additionally, since CRPS is a strictly proper scoring rule, optimizing it during training ensures that the model produces sharp and well-calibrated probabilistic forecasts. In summary, The main contributions of this paper are as follows:

- We propose **VAEneu**, a probabilistic forecasting model based on Conditional Variational Autoencoder (CVAE), incorporating CRPS as the loss function, which provides the following advantages over existing models:
  - **No parametric assumptions**: VAEneu does not require the predictive distribution to follow a specific parametric form, allowing it to capture complex data distributions flexibly.
  - **Optimizes sharpness and calibration**: By leveraging CRPS as the loss function, VAEneu ensures that the forecasts are both sharp and well-calibrated, providing more reliable probabilistic predictions.
- We demonstrate the superior performance of the proposed model through extensive empirical experiments using **12** baseline models and **12** univariate datasets.

The remainder of this paper is organized as follows. In Section 2, we review the relevant literature. Section 3 provides the foundational concepts upon which our work is built. In Section 4, we introduce our novel probabilistic forecasting model and detail our contributions to the forecasting field. Section 5 describes our benchmarking setup, including the datasets, baselines, and evaluation metrics used in our comparative study. The experimental results and their analysis are presented in Section 6. Finally, we summarize the key findings and conclude the paper in Section 7.

2 Related work

The rise of neural networks in recent years resulted in the emergence of powerful end-to-end probabilistic forecasting methods, which are capable of effectively learning from large datasets. Khosravi et al. [35] combined neural networks with the GARCH model [17] to provide prediction interval for time series forecasting. WaveNet [36] employed dilated convolutional neural networks (CNNs) to process long input time series data effectively and has shown outstanding performance in audio generation and time series forecasting. Long short-term memory(LSTM) networks have been used for parameterizing a linear State Space Model (SSM) to generate probabilistic forecast [37]. Rasp et al. [38] employed a neural network to post-process an ensemble weather prediction model. Prophet [39] is a modular regression model built upon the Generalised Additive Model [40], which can integrate domain knowledge into the modeling process. Wen et al. [32] suggested a multi-horizon quantile forecaster (MQ-R(C)NN) based on Sequence-to-Sequence RNN (Seq2Seq) [41] model. Gesthaus et al. [42] introduced another method for quantile forecasting based on the spline functions for representing output distributions. Salinas et al. [43] proposed an autoregressive forecaster for high-dimensional multivariate time series by utilizing a low-rank covariance matrix to model the output distribution. Also, they took a copula-based approach in conjunction with an RNN model to construct the forecaster. Wang et al. [44] uses a global model based on deep neural networks for capturing global non-linear patterns, while a probabilistic graphical model extracts the individual random effects locally. DeepTCN [45] is proposed as an encoder-decoder forecasting model based on dilated CNN models. This model can be utilized either as an explicit model or a quantile estimator. Rasul et al. [46] employed conditional normalizing flows to compose an autoregressive multivariate probabilis-

tic forecasting model. Normalizing Kalman Filter [47] has been proposed for probabilistic forecasting, which uses normalizing flows [48] to augment a linear Gaussian state space model. DeepAR [30] is suggested as a simple yet effective autoregressive explicit model based on RNNs. Gouttes et al. [49] suggested an autoregressive RNN model based on Implicit Quantile Networks [50] for quantile forecasting. Hasson et al. [51] introduced Level Set Forecaster (LSF). LSF organizes training data into groups, considering those that are sufficiently similar. Subsequently, the bins containing the actual values are employed to generate the predicted distributions. Lim et al. [31] proposed the Temporal Fusion Transformer(TFT). This attention-based model utilized a sequence-to-sequence model for the local process of the input window and a temporal self-attention decoder to model long-term dependencies. Denoising diffusion models have been used to develop TimeGrad [52], an autoregressive multivariate forecaster. Türkmen et al. [53] combined classical discrete-time conditional renewal processes [54] with a neural networks architecture to build a forecasting model suitable for intermittent demand forecasting. Xu et al. [55] expanded quantile RNN forecaster models to handle time series with mixed sample frequency. Kan et al. [56] proposed multivariate quantile functions, which are parameterized using gradients of "input convex neural networks" [57]. The model employs an RNN-based feature extractor to build a probabilistic forecaster. Multiple publications [33, 34, 58–60] have successfully applied GAN to transform samples from a prior distribution to samples from a predictive distribution.

The recent surge in interest in long sequence time series forecasting (LSTF) in the scientific community has led to notable developments in this domain. Pioneering contributions such as Informer [61], Autoformer [62], FEDformer [63], and TSMixer [64] have been significant. These models primarily focus on the challenge of substantially extending the forecast horizon, primarily within the context of modeling multivariate time series in a deterministic manner.

3 Background

In this section, we outline the key concepts underlying our proposed approach. We begin with an introduction to Variational Autoencoders (VAEs), explaining their generative and recognition models, training process, and the role of Kullback-Leibler divergence in regularizing the latent space. We then discuss Conditional VAEs (CVAEs), which extend VAEs by incorporating auxiliary information. Finally, we introduce the Continuous Ranked Probability Score (CRPS), a strictly proper scoring rule used to evaluate probabilistic forecasts, and explain its significance as a reconstruction loss function in our model.

3.1 Variational autoencoders

Variational Autoencoder (VAE) [65] is a member of explicit generative models and latent viable models. The model aims to generate artificial data x from the generative distribution $p_\theta(x|z)$ conditioned on latent variable $z \sim p_\theta(z)$. Due to intractable true posterior $p_\theta(z|x)$, the direct estimation of the parameter θ is not feasible. However, VAE defines a surrogate function $q_\phi(z|x)$, which is also known as the recognition model, to approximate the true posterior. With this initiative, the VAE can efficiently be trained via stochastic gradient decent using the variational lower bound of the log-likelihood:

$$\begin{aligned}\mathcal{L}_{\text{VAE}}(\mathbf{x};\theta,\phi) =& D_{KL}\left(q_\phi(\mathbf{z}\mid\mathbf{x})\|p_\theta(\mathbf{z})\right)\\ &- \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}\mid\mathbf{z})\right].\end{aligned} \tag{1}$$

The first term minimizes the Kullback-Leibler divergence (D_{KL}) between $q_\phi(z|x)$ and $p_\theta(z)$, encouraging the approximated posterior to be close to the prior distribution, resulting in a more consistent and interpretable latent space. The Kullback-Leibler divergence between two distributions $q(z)$ and $p(z)$ is expressed as:

$$D_{KL}(q(z) \parallel p(z)) = \int q(z) \log \frac{q(z)}{p(z)}\, dz$$

Assuming the prior over latent variable as an isotropic multivariate Gaussian $p_\theta(z) = \mathcal{N}(z; 0; \mathbf{I})$, the first term of the loss function can be integrated analytically. The second term of VAE loss is the negative log-likelihood of generative distribution, which measures how well the VAE can reconstruct the input from the latent space. This term, which is also known as reconstruction error, can be approximated by drawing multiple samples $z^{(l)}(l = 1, ..., L)$ from $q_\phi(z|x)$; hence the empirical loss function of VAE with Gaussian latent variable is:

$$\begin{aligned}\widetilde{\mathcal{L}}_{\text{VAE}}(\mathbf{x};\theta,\phi) =& D_{KL}\left(q_\phi(\mathbf{z}\mid\mathbf{x})\|\mathcal{N}(z;0;\mathbf{I})\right)\\ &- \frac{1}{L}\sum_{l=1}^{L}\log p_\theta\left(\mathbf{x}\mid\mathbf{z}^{(l)}\right)\end{aligned} \tag{2}$$

In practice, the $q_\phi(z|x)$ and $p_\theta(x|z)$ are approximated with neural networks, namely encoder and decoder, respectively. To enable backpropagation through the sampling process and decrease gradient variance, the encoder utilizes the reparameterization trick. In this setup, the encoder network of a VAE takes an input x and outputs the parameters of the latent distribution (mean μ and variance σ^2). This forms a Gaussian distribution:

$$q(z|x) = \mathcal{N}(z; \mu(x), \sigma^2(x))$$

Then, we can sample from this distribution using the following:

$$z = \mu(x) + \sigma(x) \odot \epsilon,$$

where ϵ is a random noise sampled from a standard normal distribution $\epsilon \sim \mathcal{N}(0, I)$.

Conditional VAE (CVAE) [66] is an important extension of VAEs that enables us to train VAE conditioned on some auxiliary information c and generate artificial data from $p_\theta(x|c)$. To do so, the condition is presented to both encoder and decoder and is integrated into VAE's objective function:

$$\begin{aligned}\widetilde{\mathcal{L}}_{\text{CVAE}}(\mathbf{x}, \mathbf{c}; \theta, \phi) =& D_{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{c}) \| \mathcal{N}(z; 0; \mathbf{I})\right) \\ &- \frac{1}{L}\sum_{l=1}^{L} \log p_\theta\left(\mathbf{x} \mid \mathbf{z}^{(l)}, \mathbf{c}\right) \end{aligned} \tag{3}$$

3.2 Continuous ranked probability score

Scoring rules are functions that assign a numerical score to a probabilistic forecast, aiming to measure its accuracy with respect to the observed outcome. Let F be a probabilistic forecast for a random variable X, and let x be the actual observed value. A scoring rule $S(F, x)$ assigns a score to F, with lower scores indicating better performance.

A scoring rule is proper if the expected score is minimized when the forecast distribution F equals the true distribution G, that is:

$$\mathbb{E}_{x\sim G}[S(F, x)] \geq \mathbb{E}_{x\sim G}[S(G, x)].$$

This ensures that forecasters are incentivized to issue truthful predictions that reflect the true distribution. The scoring rule is strictly proper if the inequality holds strictly, i.e.,

$$\mathbb{E}_{x\sim G}[S(F, x)] > \mathbb{E}_{x\sim G}[S(G, x)] \quad \text{for all } F \neq G.$$

Strict properness is crucial in probabilistic forecasting because it ensures that the model's best strategy is to predict the true distribution G and that no other forecast can achieve the same score unless it matches G. This property discourages biased or misleading forecasts and drives the model to improve both the sharpness (narrowness of the predictive distribution) and calibration (statistical consistency with the observations).

The Continuous Ranked Probability Score (CRPS) is an example of a strictly proper scoring rule for evaluating probabilistic forecasts. For a forecast distribution F and an observed outcome x, the CRPS is defined as:

$$\text{CRPS}(F, x) = \int_{-\infty}^{\infty} (F(y) - \mathbf{1}(x \leq y))^2 \, dy, \tag{4}$$

where $F(y)$ is the cumulative distribution function (CDF) of the forecast, and $\mathbf{1}(x \leq y)$ is the indicator function. The CRPS rewards both sharpness and calibration, and because it is a strictly proper scoring rule, it ensures that the model learns a predictive distribution that closely aligns with the true distribution, leading to well-calibrated and accurate forecasts.

One of the key properties that make CRPS a versatile and flexible evaluation metric for probabilistic forecasting models is its ability to be approximated even when direct access to the CDF of the predictive distribution is unavailable. This flexibility allows CRPS to be applied to a wide range of probabilistic forecasting models.

For models that provide quantiles of the predictive distribution, CRPS can be approximated as the integrated pinball loss across all possible quantile levels α, ranging from 0 to 1 [42]. Let q_α denote the forecasted quantile at level α, and let x represent the observed value. The quantile-based CRPS can be expressed as:

$$\text{CRPS}(q_\alpha, x) = \int_0^1 2\Lambda_\alpha(q_\alpha, x)\, d\alpha, \tag{5}$$

where Λ_α is the pinball loss at quantile level α, defined as:

$$\Lambda_\alpha(q_\alpha, x) = (\alpha - \mathcal{I}_{[x<q_\alpha]})(x - q_\alpha), \tag{6}$$

with $\mathcal{I}_{[x<q_\alpha]}$ being the indicator function, which equals 1 if $x < q_\alpha$ and 0 otherwise.

In practical applications, the quantile-based CRPS is typically approximated by summing over a discrete set of available quantiles. The precision of this approximation is highly dependent on the number of quantile levels used.

For probabilistic forecasting models that generate sample forecasts from the predictive distribution (e.g., via Monte Carlo sampling), CRPS can be approximated using the following expression:

$$\text{CRPS}(F, x) \approx \text{CRPS}(\bar{X}, \bar{X}', x) = \mathbb{E}_F\left|\bar{X} - x\right| - \frac{1}{2}\mathbb{E}_F\left|\bar{X} - \bar{X}'\right|, \tag{7}$$

where $\bar{X}$ and $\bar{X}'$ are independent random samples from the predictive distribution F, and both have a finite first moment [15]. In practice, CRPS is computed by drawing two sets of samples from the predictive distribution to represent $\bar{X}$ and $\bar{X}'$. The accuracy of this approximation depends on the number of samples drawn from the predictive distribution.

These various methods of calculating CRPS, whether through quantiles or sample-based approaches, make it a crucial tool for evaluating the predictive performance of probabilistic forecasting models. CRPS provides a common evaluation metric that facilitates direct comparisons between different probabilistic forecasting methodologies, ensuring a

consistent and comprehensive assessment of both sharpness and calibration.

4 VAEneu

In this section, we present the specifics of our proposed methodology. We begin by formally defining the structure of probabilistic forecasting to set the foundation for the subsequent discussion. Next, we introduce VAEneu, a novel VAE-based probabilistic forecasting model, and explain its architectural design and training process.

4.1 Problem statement

A given dataset consists of a set of observations generated by the process of interest. Let $\mathcal{D}$ be a dataset containing univariate time series comprising T timesteps :

$$\mathcal{D} = \{x_0, x_1, ..., x_T\}. \tag{8}$$

In forecasting, our ultimate objective is to obtain a predictive distribution for future quantities based on historical information. To be more specific, we aim to model the conditional probability distribution as follows:

$$P(x_{t+1:t+h}|x_{0:t}), \tag{9}$$

where, h represents the forecast horizon. In this work, we aim to model the one-step ahead forecasting to simplify the training pipeline. During inference, we extend the resultant model into the multi-step forecasting model using auto-regression, i.e., feeding the model's forecast back to the model to forecast further in the future. Hence, the target predictive distribution is updated as:

$$P(x_{t+1}|x_{0:t}), \tag{10}$$

4.2 Proposed model

Our proposed model, VAEneu, employs a CVAE at its core, where the entire encoder-decoder pipeline is conditioned on the historical window input, i.e., $c = x_{0:t}$. The encoder of VAEneu takes the target value x_{t+1} and maps it to a latent space, while the decoder attempts to reconstruct the target value from this latent representation. In this framework, the encoder approximates the posterior distribution $q_\Phi(z|x_{t+1}, x_{0:t})$, acting as the recognition model. The generative model, approximated by the decoder, is defined as $p_\theta(x_{t+1}|z, x_{0:t})$, which aligns with the target predictive distribution described in Equation (10). Consequently, the VAEneu decoder can be used to generate samples from the predictive distribution for a given historical window $x_{0:t}$.

Since VAEneu generates samples from the predictive distribution, it is well-suited to optimize the sample-based approximation of CRPS, as defined in (7), for its reconstruction loss. Because CRPS is a strictly proper scoring rule, minimizing it during training leads toward a balanced predictive distribution, one that reconciles sharpness with a well-calibrated spread of forecast samples. This balance is crucial for producing accurate and reliable forecasts. The loss function used to train the VAEneu pipeline is defined as follows:

$$\begin{aligned}\widetilde{\mathcal{L}}_{\text{VAEneu}}(\mathbf{x_{t+1}}, \bar{\mathbf{X}}_{t+1}, \bar{\mathbf{X}}'_{t+1}, \mu, \sigma; \theta, \phi) =& D_{KL}\left(\mathcal{N}(\mu, \sigma)\|\mathcal{N}(\mathbf{0}, \mathbf{I})\right) \\ &+ \text{CRPS}(\bar{X}_{t+1}, \bar{X}'_{t+1}, x_{t+1})\end{aligned} \tag{11}$$

where x_{t+1} is the target value, $\bar{X}_{t+1}$ and $\bar{X}'_{t+1}$ are two independent sets of samples drawn from the decoder's output distribution, μ and σ are the mean and standard deviation of the Gaussian latent space outputted by the encoder, and ϕ and θ represent the encoder and decoder model parameters, respectively. The first term of the loss function, the Kullback-Leibler (KL) divergence, encourages the latent space of the autoencoder to converge towards a standard Gaussian distribution. The second term, the CRPS-based reconstruction loss, aligns the decoder's generative distribution with the target value x_{t+1}, ensuring the model produces sharp and well-calibrated predictive distributions. This combination allows VAEneu to capture the uncertainty in the forecasting task effectively while ensuring the generative model's outputs are as close as possible to the true distribution.

Algorithm 1 Algorithm for training VAEneu using one sample.

Input: Historical data $x_{0:t}$ as input sequence and x_{t+1} as target value, sample size L
Output: Updated model parameters θ and ϕ
$(\mu, \sigma) \leftarrow$ Encoder$(x_{0:t}, x_{t+1}; \phi)$
Initialize an empty list $\bar{X}_{t+1}$ to hold forecasts
for i in range(L) **do**
  $\epsilon \sim \mathcal{N}(0, 1)$
  $z \leftarrow \sigma \cdot \epsilon + \mu$
  $\hat{x}_{t+1} \leftarrow$ Decode$(z, x_{0:t}; \theta)$
  Append $\hat{x}_{t+1}$ to $\bar{X}_{t+1}$
end for
$\bar{X}'_{t+1} \leftarrow$ Shuffle$(\bar{X}_{t+1})$
$g \leftarrow \nabla_{\theta,\phi}\widetilde{\mathcal{L}}_{\text{VAEneu}}(\mathbf{x_{t+1}}, \bar{\mathbf{X}}_{t+1}, \bar{\mathbf{X}}'_{t+1}, \mu, \sigma; \theta, \phi)$
$\theta, \phi \leftarrow$ Update parameters using gradients g

The algorithm for training VAEneu is outlined in Algorithm 1. For a given input x_{t+1} and condition $x_{t:0}$, we sampled multiple instances z from the recognition distribution $q_\theta(z|x_{t:0}, x_{t+1})$ to obtain forecast samples for $\bar{X}_{t+1}$. The number of these samples, denoted as *sample size*, is a

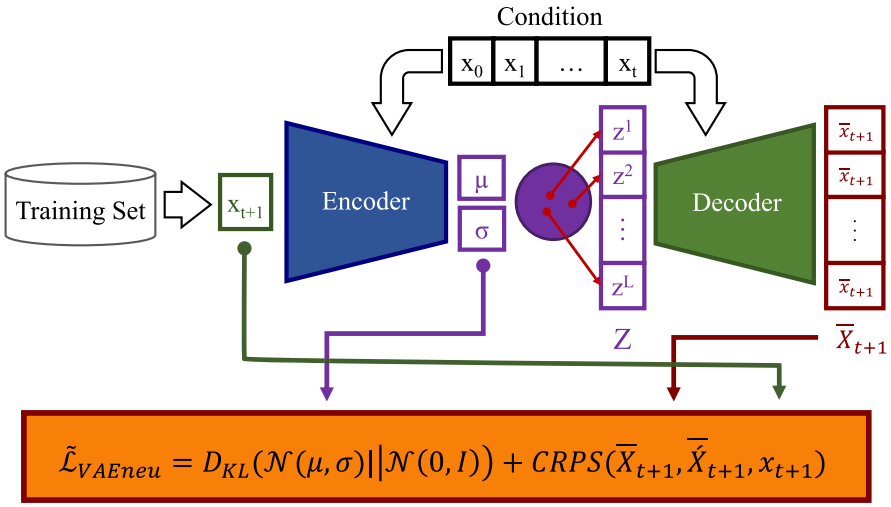


Fig. 1 VAEneu data processing pipeline during training

hyperparameter. To compute CRPS efficiently, we shuffled the forecast samples in $\bar{X}_{t+1}$ to create $\bar{X}'_{t+1}$, thereby circumventing the need for additional sampling from the predictive distribution.

4.3 Model architecture

Figure 1 illustrates how VAEneu processes input data, and Fig. 2 offers a detailed view of the VAEneu architecture. VAEneu is implemented in two distinct structural variants. VAEneu-TCN incorporates TCN in both its encoder and decoder, while VAEneu-RNN employs LSTM networks as its core components. The encoder process involves concatenating the historical data with the target forecast to form the network input $x_{0:t+1}$. This input is then transformed into a representation h_{t+1} using either LSTM or TCN. Subsequently, the representation passes through parallel fully connected layers to generate mean $\mu_\phi(h_{t+1})$ and standard deviation $\sigma_\phi(h_{t+1})$ outputs. Latent variables z are generated via reparameterization and then fed into the decoder.

The decoder's role is to learn the predictive distribution $p_\theta(x_{t+1}|z, x_{0:t})$. It begins by mapping the condition to its representation h_t and then concatenates this with the latent variable from the encoder. The final mapping to the next forecast is achieved through a fully connected layer.

After training VAEneu, during the inference phase, the encoder is discarded, and the decoder functions as the probabilistic forecaster. The decoder takes the historical window and samples from a standard Gaussian distribution, generating a sample from the predictive distribution for the next time step. This sample can then be fed back into the network to generate forecasts for subsequent time steps, extending

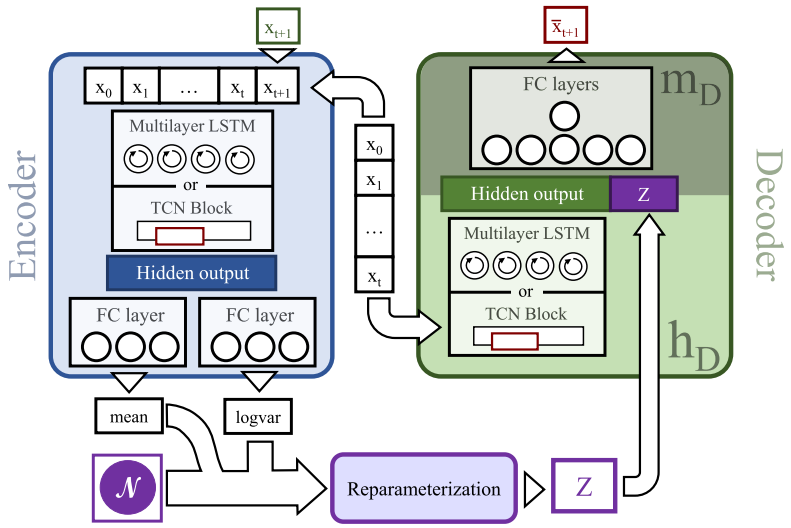
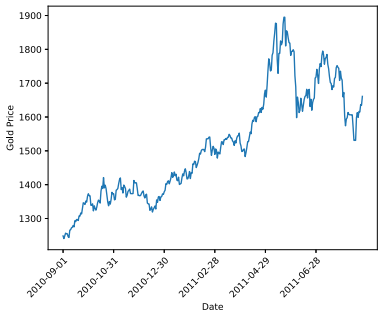
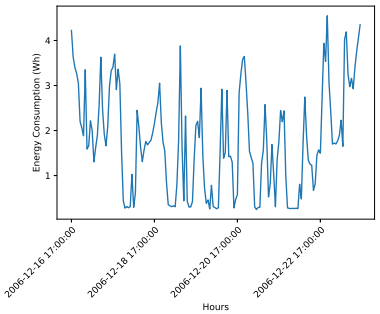


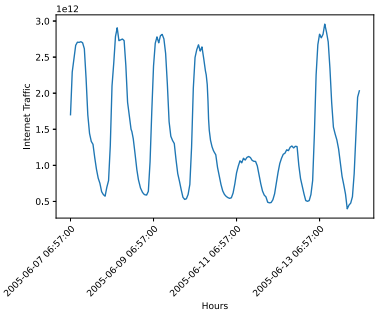
Fig. 2 The structure of VAEneu in detail



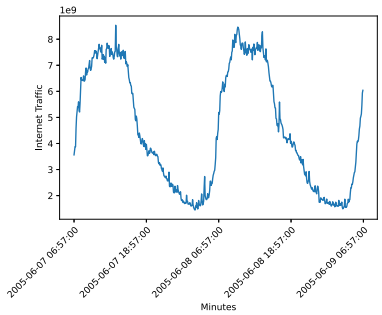
(a) Gold Price Dataset



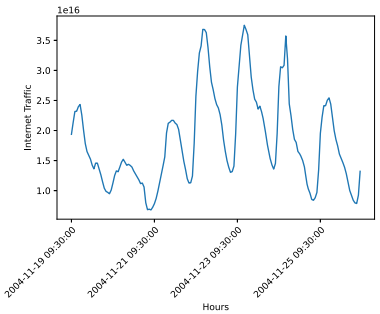
(b) HEPC Dataset



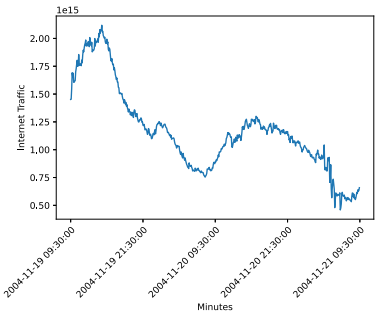
(c) Internet Traffic A1H Dataset



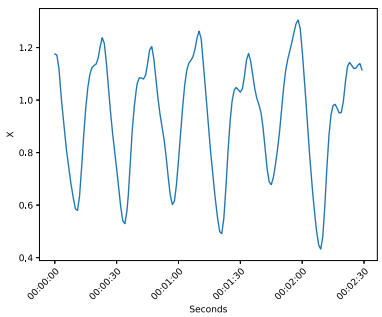
(d) Internet Traffic A1M Dataset



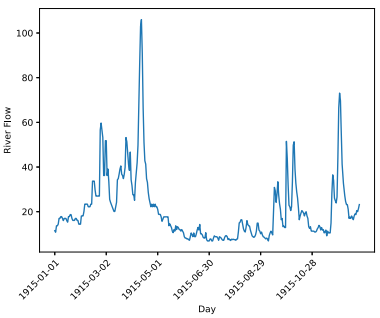
(e) Internet Traffic B1H Dataset



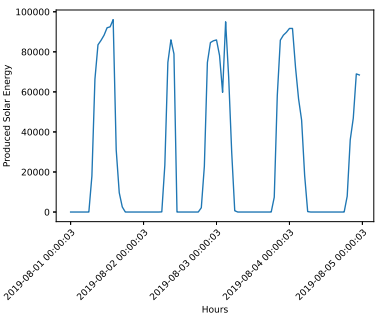
(f) Internet Traffic B5M Dataset



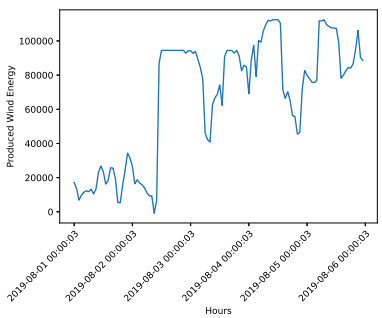
(g) Mackey Glass Dataset



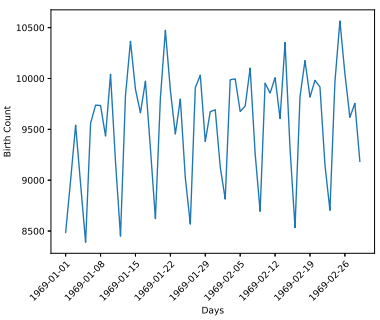
(h) Saugeenday River Dataset



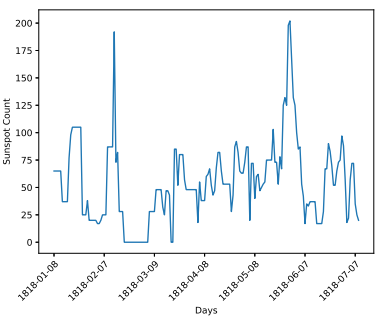
(i) Solar-4-seconds Dataset



(j) Wind-4-seconds Dataset



(k) US Births Dataset



(l) Sunspot Dataset

Fig. 3 Illustration of a slice of each univariate time series dataset

VAEneu's forecasting capabilities to multistep ahead forecasting through an autoregressive approach.

5 Experiment

5.1 Datasets

This section introduces an array of datasets utilized to evaluate VAEneu performance nuances under different data conditions. We employed 12 datasets sourced from public repositories. Table 1 lists these datasets and their main features, and Fig. 3 visualizes a segment from each to provide an overview of the diverse dynamics of these datasets. The rest of this section presents further details for each dataset.

5.1.1 Gold price dataset

Gold is a vital commodity with a significant history of global trading. The Gold Price dataset [67] provides daily pricing details, encompassing 2,487 data points from September 1st, 2010, to March 13th, 2020. This dataset facilitates analytical and predictive modeling of the gold market due to its comprehensive coverage during the specified period.

5.1.2 Household electric power consumption dataset

The Household Electric Power Consumption (HEPC) dataset [68] comprises electric power consumption readings from a singular household with a sampling rate of one minute. From December 16th, 2006, to November 26th, 2010, it encapsulates nearly four years of data. Notably, there are timestamps with missing values. An imputation approach, where missing values are replaced with the average of their adjacent observations, is adopted to ensure data integrity. Moreover, the dataset is resampled to aggregate the readings into hourly intervals, ensuring manageability while preserving inherent patterns.

Table 1 The properties of univariate datasets utilized for multi-step-ahead forecasting experiments

Dataset Name	Frequency	Length	History Size	Horizon
Gold Price	Daily	2487	60	30
HPEC	Hourly	34569	48	24
Internet Traffic A1H	Hourly	1231	48	24
Internet Traffic A5M	5 minutes	14772	24	12
Internet Traffic B1H	Hourly	1657	48	24
Internet Traffic B5M	5 minutes	19888	24	12
Mackey Glass	Seconds	20000	120	60
Saugeen River Flow	Daily	23741	60	30
Sunspot	Daily	73924	60	30
US Births	Daily	7305	60	30
Solar	Hourly	8219	48	24
Wind	Hourly	8219	48	24

5.1.3 Internet traffic datasets

Internet traffic datasets [69] offer data from two distinct ISPs, named A and B. The A dataset pertains to a private ISP with nodes across 11 European cities, capturing data on a transatlantic link from June 7the to July 29the, 2005. The B dataset is sourced from UKERNA1 and aggregates traffic across the UK's academic network from November 19th, 2004, to January 27th, 2005. The A dataset logs every 30 seconds, whereas B records every 5 minutes. Aggregated variants, A5M, A1H, B5M, and B1H, offer data at 5-minute and 1-hour resolutions respectively.

5.1.4 Macky-glass dataset

The time-delay differential equation proposed by Mackey and Glass [70] stands as a benchmark for generating chaotic time series. The equation is given by:

$$\dot{x} = \frac{a\, x(t-\tau)}{(1+10\cdot(t-\tau)) - b\, x(t)}\,. \tag{12}$$

Adhering to the parameters in [71], where $a = 0.1$, $b = 0.2$, and $\tau = 17$, a dataset of length 20,000 is generated using Equation (12). The dataset is accessible in [72].

5.1.5 Saugeen river flow dataset

This dataset [73] includes a univariate time series showcasing the daily mean flow of the Saugeen River at Walkerton measured in cubic meters per second ($\frac{m^3}{s}$). It captures data from January 1st, 1915, to December 31st, 1979. The dataset is archived in the Monash Time Series Forecasting Archive [74].

5.1.6 Solar-4-seconds and Wind-4-seconds datasets

These datasets capture solar and wind power production every 4 seconds from August 1st, 2019, for approximately one year. For the purpose of this research, the datasets are aggregated to represent hourly resolutions. Sourced from the Australian Energy Market Operator (AEMO) online platform, they are also part of the Monash Time Series Forecasting Archive [74].

5.1.7 Sunspot dataset

The dataset contains a time series depicting daily sunspot counts from January 8th, 1818, to May 31st, 2020. Any missing values within the dataset are addressed using the Last Observation Carried Forward (LOCF) method. The Solar Influences Data Analysis Center is the primary data source, with the dataset also being part of the Monash Time Series Forecasting Archive [74].

5.1.8 US birth dataset

This dataset illustrates the number of births in the US from January 1st, 1969, to December 31st, 1988. Originally sourced from the R mosaicData package [75], it's accessed for this research via the Monash Time Series Forecasting Archive [74].

5.2 Baseline

In this study, we compare the proposed model with 12 established probabilistic forecasting models, and this section is dedicated to a detailed examination of these models. These models represent the latest advancements in the field and have demonstrated their effectiveness and robustness across various research studies. The primary focus will be on elucidating the distinctive features, methodologies, and underlying principles of each model, highlighting their contributions to advancing probabilistic forecasting. In our experiments, we employed the implementation of these methods from GluonTS [76] package with their default hyperparameters, with the exception of ForGAN. For the ForGAN, we used the implementation provided by their author.

5.2.1 DeepAR

DeepAR, as presented by Salinas et al. [30], serves as a prominent explicit probabilistic forecasting model in the realm of time series prediction. At its core, DeepAR constructs a predictive distribution utilizing a Gaussian distribution, wherein the distribution's parameters, specifically the mean and standard deviation, are derived from the historical data. To facilitate this parameter extraction, DeepAR integrates a neural network structure. This architecture predominantly features a multi-layer RNN to map the time series history onto the Gaussian distribution parameters. The training regimen for the network adopts a maximum likelihood estimation paradigm. Within this paradigm, the probability density function of the Gaussian distribution acts as the likelihood function, ensuring optimal parameter approximation.

5.2.2 DeepState

Rangapuram et al. [37] proposed the DeepState model as a novel approach that combines state space models (SSMs) with deep learning for probabilistic time series forecasting. The paper proposes bridging the gap between SSMs and deep learning by parametrizing a linear SSM with a jointly-learned RNN. The deep RNN parametrizes the mapping from covariates to state space model parameters. The model is interpretable, as it allows inspection and modification of SSM parameters for each time series. It can automatically extract features and learn complex temporal patterns from raw time series and associated covariates. The model promises to contain the interpretability and data efficiency of SSMs and large data handling of deep neural networks.

5.2.3 DeepFactor

The DeepFactor [44] employs both classical and deep neural time series forecasting models to construct probabilistic forecaster. DeepFactor is a global-local method where each time series or its latent function is represented as a combination of a global time series and a corresponding local model. The global factors are modeled by RNNs. These latent global deep factors can be thought of as dynamic principal components driving the underlying dynamics of all time series. Local models can include various choices like white noise processes, linear dynamical systems (LDS), or Gaussian processes (GPs). This stochastic component captures individual random effects for each time series. DeepFactor systematically combines of deep neural networks and probabilistic models in a global-local framework and develops an efficient and scalable inference algorithm for non-Gaussian likelihoods.

5.2.4 Deep renewal processes

Deep Renewal Processes [53] is a novel framework for probabilistic forecasting of intermittent and sparse time series. The method focuses on forecasting demand data that appears sporadically, posing unique challenges due to long periods of zero demand followed by non-zero demand. The framework is based on discrete-time renewal processes, which are adept at handling patterns like aging, clustering, and quasi-periodicity in demand arrivals. The model innovatively incorporates neural networks, specifically RNNs, replacing exponential smoothing with a more flexible neural approach. Moreover, the framework also extends to model continuous-time demand arrivals, enhancing flexibility and applicability.

5.2.5 GPForecaster

For the Gaussian Process (GP), we employed a model implemented in GluonTS [76]. This model defines a Gaussian Process with Radial Basis Function (RBF) as the kernel for each time step. GP with an RBF kernel can capture both the trends and nuances of time series data, while its probabilistic nature provides a measure of uncertainty or confidence in its predictions. This combination is especially valuable in scenarios where understanding the uncertainty of future events is as crucial as the predictions themselves.

5.2.6 Multi-horizon quantile recurrent forecaster (MQ-RNN)

The Multi-Horizon Quantile Recurrent Forecaster, or in short MQ-RNN [32], is a direct quantile regression method that aims to forecast the quantile of predictive distribution for multiple horizons directly. The model employs Seq2Seq architecture for quantile regression. Besides MQ-RNN, this paper also utilized MQ-CNN, which is the same model with CNN components as the main components of Seq2Seq architecture instead of RNN modules.

5.2.7 Multi-horizon quantile convolutional forecaster (MQ-CNN)

MQ-CNN is a similar method to MQ-RNN, with CNN components as the main components of Seq2Seq architecture instead of RNN modules.

5.2.8 Prophet

The prophet [39] is a robust and scalable solution for forecasting time series data, particularly in business contexts where handling specific scenarios such as multiple strong seasonalities, trend changes, outliers, and holiday effects becomes paramount. The prophet decomposes time series into three components. The first component, trend, Captures non-periodic changes in the data. The second component, seasonality, represents periodic changes, like weekly yearly cycles. finally, the last component, holidays, accounts for holidays and events that occur on irregular schedules. Each component is modeled separately and fits using Bayesian infeProbabilistic Stream Flow Forecasting for Reservoir Operatiorence to estimate parameters. Furthermore, the model is designed to be intuitive, allowing analysts to make informed adjustments based on domain knowledge.

5.2.9 WaveNet

The Wavenet [36] is a powerful generative model originally proposed for voice synthesis. Wavenet forecasts sequentially in an autoregressive fashion using previously generated samples as input to predict the next.WaveNet employs a deep neural network consisting of stacks of dilated causal convolutions, enabling it to capture a wide range of audio frequencies and temporal dependencies. Wavenet defines predictive distribution for each time step in the horizon by quantifying the time series and using a softmax at the output layer of the model. In other words, the wavenet quantifies the time series to make the problem similar to a classification problem and provides the output of the softmax layer as the predictive distribution.

5.2.10 Transformer

The transformer [29] is a novel sequence-to-sequence model that represents a departure from previous methods by relying entirely on attention mechanisms, dispensing with recurrent and convolutional neural networks. The Transformer follows the typical encoder-decoder structure, but both the encoder and the decoder are composed of a stack of identical layers, each with a novel self-attention mechanism. The Self-Attention mechanism allows the model to weigh the importance of different time steps in the input sequence, enabling it to capture context more effectively. At the core of the model is the scaled dot-product attention, a mechanism that calculates the attention scores based on the scaled dot products of queries and keys. Unlike recurrent models, the Transformer allows for much more parallelization, making training faster. Moreover, the self-attention mechanism enables the model to learn long-range dependencies more effectively.

5.2.11 Temporal fusion transformers

Temporal Fusion Transformers (TFT) [31] is an encoder-decoder-based quantile regression model for multi-horizon forecasting. It has many novelties under the hood that make its performance excel in many scenarios, including:

- **Static Covariate Encoders**: These encode static context vectors, which are crucial for conditioning the temporal dynamics within the network.
- **Gating Mechanisms**: They provide adaptive depth and complexity to the network, allowing it to handle a wide range of datasets and scenarios.
- **Variable Selection Networks**: These networks select relevant input variables at each time step, enhancing interpretability and model performance by focusing on the most salient features.
- **Temporal Processing**: The model employs a sequence-to-sequence layer for local processing and interpretable multi-head attention mechanisms to capture long-term dependencies.

The TFT demonstrates superior performance across various real-world datasets, outperforming existing benchmarks in multi-horizon forecasting. The model facilitates interpretability, allowing users to identify globally important variables, persistent temporal patterns, and significant events.

5.2.12 ForGAN

ForGAN [33] is a probabilistic forecaster that employs a conditional GAN to learn a mapping from a prior distribution to the predictive distribution conditioned on an input history window. It incorporates RNN modules in both the Generator and Discriminator components. The ForGAN was proposed for one-step ahead univariate time series forecasting. In this paper, we extend the model to encompass multi-step ahead forecasting by incorporating an auto-regressive approach.

5.3 Evaluation metrics

To comprehensively evaluate the performance of our proposed model, we employed several rigorous assessment metrics and methodologies. The primary metric for performance evaluation is CRPS, which is particularly suited for probabilistic forecasting models. This section elaborates on the various evaluation criteria used, including sharpness and calibration, which are crucial for determining the accuracy and reliability of the predictive distributions. Additionally, we include Mean Squared Error (MSE) to provide a multifaceted view of the model's performance. Through these assessments, we aim to validate the effectiveness and robustness of VAEneu across diverse datasets and comparative baselines.

5.3.1 CRPS

In this study, we utilize the CRPS to assess the performance of various probabilistic forecasters. For models that yield samples from the predictive distribution as their forecast output, we used 1,000 samples per time step in the forecast horizon from predictive distribution to approximate CRPS using Equation (7). For models which provides quantiles of predictive distribution as forecast, we used 99 equally distance quantile levels to approximate CRPS using Equation (5).

5.3.2 Sharpness and calibration

To assess the sharpness and calibration of a forecasting model, we quantify these properties using two key percentiles of the predictive distribution: the 5th percentile (p_5) and the 95th percentile (p_{95}).

Sharpness Sharpness measures the concentration of the predictive distribution. We quantify sharpness using the ratio of p_{95} to p_5, denoted as:

$$\frac{p_{95}}{p_5}.$$

A sharp predictive distribution yields a ratio close to 1, indicating that the distribution is narrow and concentrated. Larger values of this ratio indicate a wider predictive distribution, implying lower sharpness. If the ratio value falls below 1, it indicates a quantile crossing error, which invalidates the results and suggests significant issues with the model.

Calibration Calibration measures how well the predicted intervals of the model match the actual outcomes. We assess calibration using the 90% coverage probability (CP_{90}), which evaluates the proportion of true values that fall within the prediction interval defined by p_5 and p_{95}. The coverage probability is calculated as follows:

$$CP_{90} = \frac{1}{N}\sum_{i=1}^{N} I(p_5^i \le x^i \le p_{95}^i),$$

where N is the number sample in the test set, x is the observed value, and $I(\cdot)$ is the indicator function that returns 1 if the condition is true and 0 otherwise.

For a well-calibrated model, CP_{90} should be close to the nominal coverage level of 0.90. If CP_{90} is significantly lower than 0.90, the model is under-calibrated, indicating that the prediction intervals are too narrow and often miss the true values. Conversely, if CP_{90} is significantly higher than 0.90, the model is over-calibrated, indicating that the prediction intervals are too wide and often include the true values unnecessarily.

5.3.3 Mean squared error (MSE)

The Mean Squared Error (MSE) is a standard metric used to assess the accuracy of deterministic forecasting models. It measures the average squared difference between the predicted values and the actual observed values. The formula for MSE is given by:

$$MSE = (x - \hat{x})^2,$$

where x is the observed value t, and $\hat{x}$ is the predicted value.

When applied to probabilistic models, MSE considers only how well the first moment (mean) of the predictive distribution matches the ground truth. It does not account for the uncertainty or spread of the predictive distribution. Therefore, while MSE can provide insights into the accuracy of the point predictions derived from the mean of the predictive distribution, it may not fully capture the model's

performance in probabilistic forecasting, where the entire predictive distribution is of interest. In this context, it is important to complement MSE with other metrics that consider the distributional aspects of the forecasts.

5.4 Experiment setup

Our experimental setup involves partitioning each dataset, reserving a portion equivalent to five forecast horizons for testing while the remainder is utilized for model training. The proposed models undergo training using RMSProp optimizer [77] for a maximum of 100,000 steps. However, training is ceased prematurely if no improvement in model performance is observed over 5,000 consecutive training steps.

The models detailed in this study have been implemented leveraging the Pytorch framework [78] and trained on a machine equipped with an Nvidia RTX 6000 GPU. We used eight samples to estimate CRPS during training (sample size = 8). The rest of the hyperparameters for VAEnue models are either fixed values or defined solely based on history windows size (HWS), i.e., the condition. Table 2 lists the VAEneu models' hyperparameters. Conducting experiments with these hyperparameters guarantees fair comparison with baseline models, which were used with their default hyperparameters. Furthermore, with this setup, we can also get an insight into the robustness of the VAEneu with regard to model structure hyperparameters.

To circumvent any anomalies and ensure robustness, each model is subjected to three independent training iterations on every dataset. The reported performance metric is an arithmetic mean of these three runs, which is indicated as $\overline{CRPS}$. The coefficient of variation (CV) is also presented in parentheses to shed light on the stability and reproducibility of each technique. CV provides a standardized measure of dispersion of a probability distribution or frequency distribution and is defined as:

$$CV = \frac{\sigma}{\mu} \times 100\%, \tag{13}$$

where μ and σ represent the mean and standard deviation of three CRPSs from three runs. A lower CV indicates less variability relative to the mean.

Table 2 Hyperparameters of VAEneu models

Hyperparameter	Value
VAEneu-RNN	
Sample size	8
Number of LSTM layers	1
LSTM hidden size	$\frac{HWS}{2}$
Latent variable "z" size	$\frac{HWS}{2}$
VAEneu-TCN	
Sample size	8
TCN block kernel size	5
TCN block number of layers	$\text{Log}(\frac{HWS}{8})$
TCN block hidden size	2×Log(HWS)
Latent variable "z" size	2×Log(HWS)

6 Results and discussion

In this section, we present a comprehensive evaluation of the proposed VAEneu model using four key metrics: Continuous Ranked Probability Score (CRPS), Mean Squared Error (MSE), a 90% coverage probability (CP_{90}) for calibration, and a 95% to 5% ratio (P95/P5) for sharpness. These metrics collectively provide a holistic view of the model's performance, capturing both the accuracy of point predictions and the quality of the probabilistic forecasts. The following subsections detail the results and discussion for each metric, providing insights into the strengths and weaknesses of the VAEneu model compared to the baseline models.

6.1 Forecast accuracy evaluation

The evaluation of VAEneu models and 12 baselines across 12 diverse datasets is encapsulated based on CRPS in Table 3. The extensive data presented in these tables can initially seem overwhelming. However, a color-coded scheme aids in interpreting the relative performance of the models. The color denotes the relative score of a model in comparison to the best-performing model on that dataset and it is defined as:

$$\Delta_{CRPS} = \frac{\overline{CRPS} - \overline{CRPS}^*}{\overline{CRPS}^*} \times 100\%, \tag{14}$$

wherein $\overline{CRPS}^*$ represents the $\overline{CRPS}$ of the top-performing model for a given dataset. With a glance at the color scheme of Table 3, we can perceive that VAEneu models are the best-performing model on five datasets, and on the rest of the datasets, they are always following the best-performing model closely, with the expectation of one dataset, US Birth dataset.

To provide a concise summary of each model's performance across all datasets, we present the average model ranks. The average model rank is calculated by first ranking the models on each dataset based on their performance according to the mean CRPS ($\overline{CRPS}$). For each model, we

Table 3 The figure displays the results of the multi-step-ahead forecasting experiment

	Gold price	HEPC	Internet traffic A1H	Internet traffic A5M	Internet traffic B1H	Internet traffic B5M
Prophet	1.1e+2 (0.3%)	3.5e-1 (2.1%)	4.9e+11 (1.0%)	1.3e+9 (1.5%)	1.9e+15 (1.7%)	1.4e+14 (1.2%)
MQ-RNN	3.2e+2 (9.4%)	5.3e-1 (17.0%)	1.4e+12 (0.0%)	5.1e+9 (0.0%)	1.6e+16 (0.0%)	1.2e+15 (0.0%)
MQ-CNN	4.6e+1 (15.5%)	7.3e-1 (27.7%)	4.7e+11 (3.8%)	9.3e+8 (1.6%)	2.4e+15 (8.1%)	2.4e+14 (3.1%)
TFT	4.7e+1 (18.4%)	2.9e-1 (24.5%)	2.5e+11 (42.6%)	1.7e+8 (5.3%)	8.1e+14 (3.0%)	**3.0e+13 (10.1%)**
Transformer	7.8e+1 (55.8%)	2.8e-1 (7.8%)	1.0e+11 (37.4%)	1.8e+8 (32.5%)	8.6e+14 (19.2%)	5.1e+13 (31.7%)
DeepFactor	1.1e+2 (77.4%)	1.1e+0 (9.9%)	1.5e+12 (55.5%)	3.4e+9 (2.2%)	5.8e+15 (24.3%)	4.9e+14 (4.1%)
DeepAR	6.5e+1 (48.2%)	**2.6e-1 (4.2%)**	9.9e+10 (11.4%)	1.0e+8 (13.4%)	6.5e+14 (29.3%)	3.5e+13 (5.5%)
DRP	1.3e+3 (0.2%)	5.6e-1 (2.4%)	1.4e+12 (0.0%)	5.2e+9 (0.0%)	1.6e+16 (0.0%)	1.2e+15 (0.0%)
Wavenet	3.3e+1 (23.2%)	2.9e-1 (6.7%)	1.8e+11 (23.9%)	1.3e+8 (9.3%)	1.5e+15 (14.2%)	3.3e+13 (22.6%)
GPForecaster	3.9e+1 (2.2%)	6.0e-1 (1.5%)	1.4e+12 (0.0%)	5.1e+9 (0.0%)	1.6e+16 (0.0%)	1.2e+15 (0.0%)
DeepState	4.5e+1 (10.3%)	7.2e-1 (12.4%)	7.9e+11 (5.2%)	1.2e+9 (25.1%)	2.8e+15 (8.6%)	2.7e+14 (19.7%)
ForGAN	1.9e+1 (4.2%)	3.0e-1 (5.7%)	8.9e+10 (4.6%)	**9.6e+7 (4.6%)**	1.1e+15 (35.1%)	4.0e+13 (6.0%)
VAEneu-RNN	**1.9e+1 (1.3%)**	3.1e-1 (8.4%)	**5.3e+10 (5.8%)**	1.1e+8 (6.1%)	5.8e+14 (20.7%)	4.2e+13 (12.6%)
VAEneu-TCN	2.1e+1 (2.7%)	2.9e-1 (3.7%)	5.9e+10 (4.0%)	1.1e+8 (12.7%)	**5.3e+14 (11.4%)**	3.5e+13 (7.0%)
	Mackey Glass	Saugeen river	Solar 4 seconds	Sunspot	US births	Wind 4 seconds
Prophet	1.3e-1 (0.7%)	1.2e+1 (2.3%)	7.5e+3 (1.4%)	2.2e+1 (1.5%)	3.6e+2 (0.9%)	1.2e+4 (2.4%)
MQ-RNN	1.4e-1 (5.2%)	2.1e+1 (3.4%)	5.4e+3 (8.1%)	3.4e+0 (4.8%)	9.5e+2 (2.4%)	1.4e+4 (1.3%)
MQ-CNN	1.2e-1 (7.4%)	1.4e+1 (1.2%)	4.8e+3 (11.9%)	3.5e+0 (14.6%)	5.0e+2 (14.2%)	1.3e+4 (1.5%)
TFT	5.9e-3 (17.8%)	8.9e+0 (7.0%)	3.3e+3 (18.2%)	3.0e+0 (3.0%)	**1.3e+2 (1.8%)**	9.3e+3 (14.0%)
Transformer	1.2e-1 (19.1%)	1.2e+1 (14.9%)	1.9e+3 (10.4%)	2.6e+0 (0.7%)	3.9e+2 (58.9%)	8.9e+3 (23.1%)
DeepFactor	2.5e-1 (1.9%)	2.2e+1 (1.0%)	3.0e+4 (2.4%)	2.8e+0 (12.3%)	3.1e+3 (53.2%)	1.2e+4 (3.9%)
DeepAR	7.6e-2 (8.1%)	1.3e+1 (2.4%)	**1.6e+3 (13.3%)**	2.6e+0 (2.3%)	1.8e+2 (7.8%)	9.7e+3 (12.0%)
DRP	1.9e-1 (0.0%)	1.0e+1 (2.4%)	3.0e+4 (0.0%)	3.4e+1 (23.4%)	1.1e+4 (0.1%)	1.7e+4 (0.0%)
Wavenet	7.1e-3 (1.8%)	9.4e+0 (4.7%)	4.2e+3 (31.7%)	**2.6e+0 (12.0%)**	3.2e+2 (10.6%)	1.3e+4 (22.0%)
GPForecaster	3.3e-1 (44.3%)	1.1e+1 (1.8%)	2.1e+4 (1.6%)	4.0e+0 (16.5%)	4.2e+2 (0.2%)	1.2e+4 (3.1%)
DeepState	2.8e-1 (1.1%)	1.5e+1 (13.4%)	7.4e+3 (18.3%)	9.9e+0 (13.7%)	5.4e+2 (4.8%)	2.8e+4 (28.3%)
ForGAN	4.9e-3 (31.2%)	**8.3e+0 (0.5%)**	3.1e+3 (6.4%)	3.6e+0 (3.4%)	2.8e+2 (6.8%)	6.9e+3 (3.6%)
VAEneu-RNN	8.6e-4 (10.8%)	8.4e+0 (0.2%)	2.8e+3 (3.7%)	3.4e+0 (2.4%)	2.9e+2 (3.4%)	**6.9e+3 (0.3%)**
VAEneu-TCN	**4.5e-4 (24.9%)**	8.4e+0 (1.1%)	1.8e+3 (7.5%)	2.7e+0 (1.5%)	2.9e+2 (7.1%)	7.4e+3 (0.5%)

Each model underwent three separate training iterations on each dataset. The performance is quantified using the average CRPS (lower is better), with the Coefficient of Variation (CV) indicated within parentheses. The color-coding of the cells illustrates the model's performance relative to the best-performing model on each dataset: green signifies the top model, blue indicates a deviation of up to 10% from the best model, black for up to 50% deviation, orange for up to 100% deviation, and red for deviations exceeding 100%. Notably, the VAEneu models demonstrate exemplary performance, securing the position of the best model in five datasets and, in most cases, deviating less than 50% from the leading model's results

then compute the average of its ranks across all datasets, as shown in Table 4. The table clearly indicates that VAEneu-TCN achieved the lowest average rank, demonstrating its consistently accurate forecasts across all datasets. VAEneu-RNN follows closely in second place, highlighting that the TCN variant of VAEneu outperforms the RNN version, although both surpass other baselines on average.

Moreover, the table reveals a clear clustering of the models into two groups based on their average rank. The top 7 models have closely aligned average ranks, forming the "top-tier models." In contrast, there is a notable gap between these and the ranks of the bottom 7 models. Consequently, we will refer to these top 7 models as the "top-tier models" hereafter.

Next, we delved deeper into the performance analysis of top-tier models. Figure 4 illustrates the distribution of relative scores Δ_{CRPS} among the top-tier models across all datasets. VAEneu-TCN separates itself from the rest of the models with a stable performance that is perfect or near perfect across all datasets. VAEneu-RNN follows the VAEnet-TCN alongside DeepAR and ForGAN. DeepAR's remarkable performance was somewhat unexpected, considering that DeepAR assumes explicitly that predictive distribution is Gaussian distribution, which renders the model less flexible.

Table 4 The average rank of studied models across all datasets based on their performance according to the $\overline{CRPS}$

Model	Average rank	
VAEneu-TCN	2.00	Top tier models
VAEneu-RNN	2.75	
DeepAR	3.16	
ForGAN	3.25	
TFT	3.25	
Wavenet	4.08	
Transformer	4.33	
MQ-CNN	8.58	Low tier models
Prophet	8.75	
GPForecaster	10.16	
DeepState	10.33	
MQ-RNN	10.75	
DeepFactor	10.83	
Deep Renewal Peocesses	12.25	

The performance of DeepAR suggests that Gaussian distributions can aptly approximate uncertainty in many real-world scenarios.

Figure 5 exhibits VAEneu models forecast on the Internet Traffic B1H dataset and provides qualitative comparison ground against DeepAR, the best-performing baseline model on this dataset. The DeepAR model generates a sharp and relatively low-calibrated predictive distribution. In contrast, the VAEneu models demonstrate an effective balance between sharpness and calibration within their predictive distributions. Notably, the TCN-based model has slightly better sharpness in comparison to the RNN variant.

To analyze the model performance consistency across three runs, we investigate the distribution of CV for top-tier models and present the result in Fig. 6. Here, we can observe that VAEneu, alongside ForGAN and DeepAR models, can deliver good performance at a consistent rate. Figures 8, 9, and 10 present a sample forecast for all datasets from VAEneu models and the best baselines for each dataset to provide a qualitative insight into the models' performances as well.

Finally, to shed light on baseline and proposed models performances from a different perspective, figure 7 illustrates relative MSE for top-tier models. These results are consistent with relative CRPS from Fig. 4, illustrating the distinct performance of VAEneu-TCN, which captured the first moment of predictive distribution precisely. The RNN-based VAEneu model has on paar performance with other baseline models, with the exception of Wavenet, which distinctively has worse performance compared to other top-tier models considering MSE across all datasets.

6.2 Calibration and sharpness evaluation

The main goal of modeling a probabilistic forecaster is to reach a sharp and well-calibrated predictive distribution. These two qualities should be analyzed and considered together. A sharp predictive distribution with low calibration would output the wrong predictive distribution with high confidence. In contrast, a well-calibrated model with low sharpness would give us the probabilities that match with the observed frequencies well, however, with high uncertainty in the provided forecast.

Figure 11 illustrates the P95/P5 ratio and CP_{90} for all baseline models and proposed models to reflect the sharpness and calibration of their predictive distribution, respectively. The VAEneu models, ForGAN and Prophet, generated the

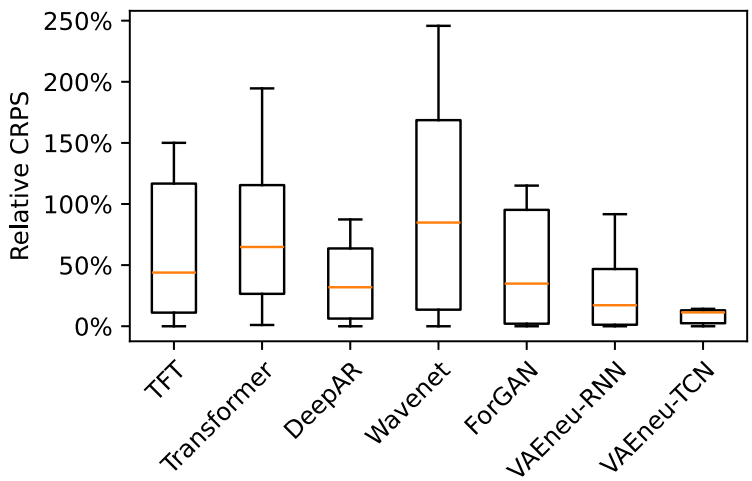


Fig. 4 The distribution of relative CRPS for top tier models across all datasets revealing remarkable performance of VAEneu-TCN model which is constantly secured best results or performed close to the best model on all datasets

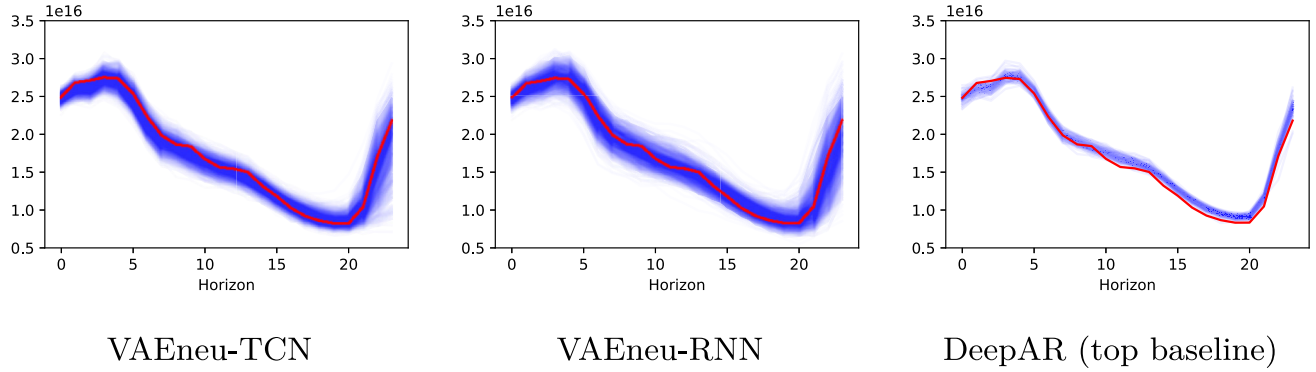


Fig. 5 The qualitative results of VAEneu models alongside the top baseline model, DeepAR, on the Internet Traffic B1h dataset demonstrate that the proposed models' calibrations are relatively superior in comparison to DeepAR predictive distribution

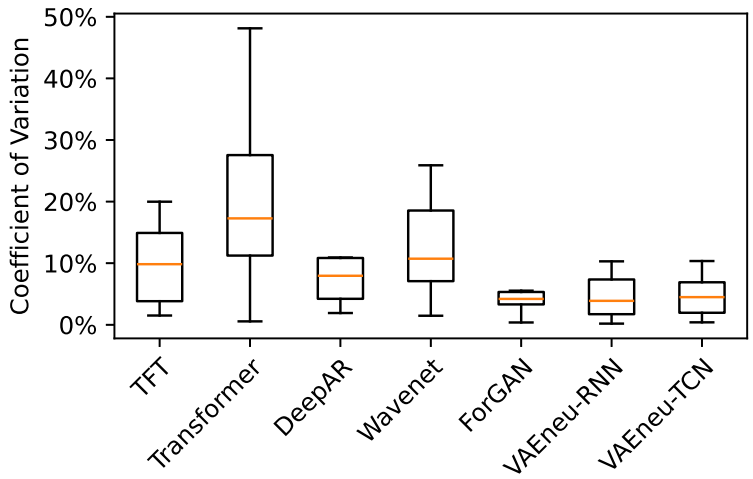


Fig. 6 The coefficient of variation distribution for top tier models across all datasets indicates the consistent performance of VAEneu models across three training iteration overall datasets and highlights stable performance for the ForGAN and DeepAR models as well

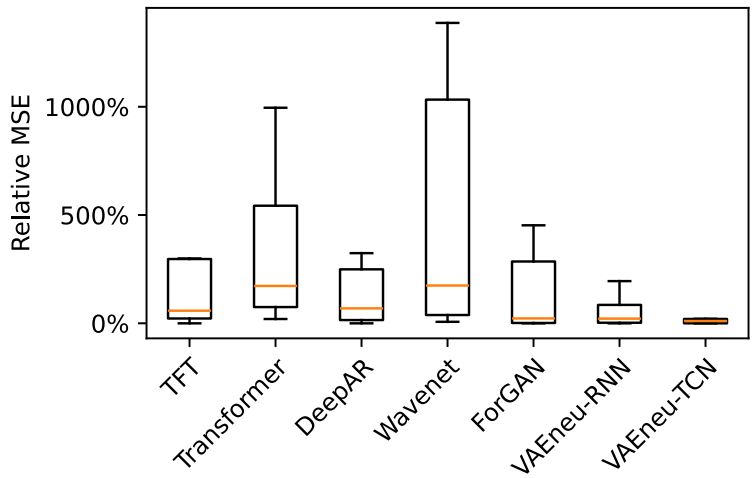


Fig. 7 The distribution of relative MSE for top-tier models across all datasets endorsing the outstanding performance of the VAEneu-TCN model, which is consistent with previous results based on CRPS

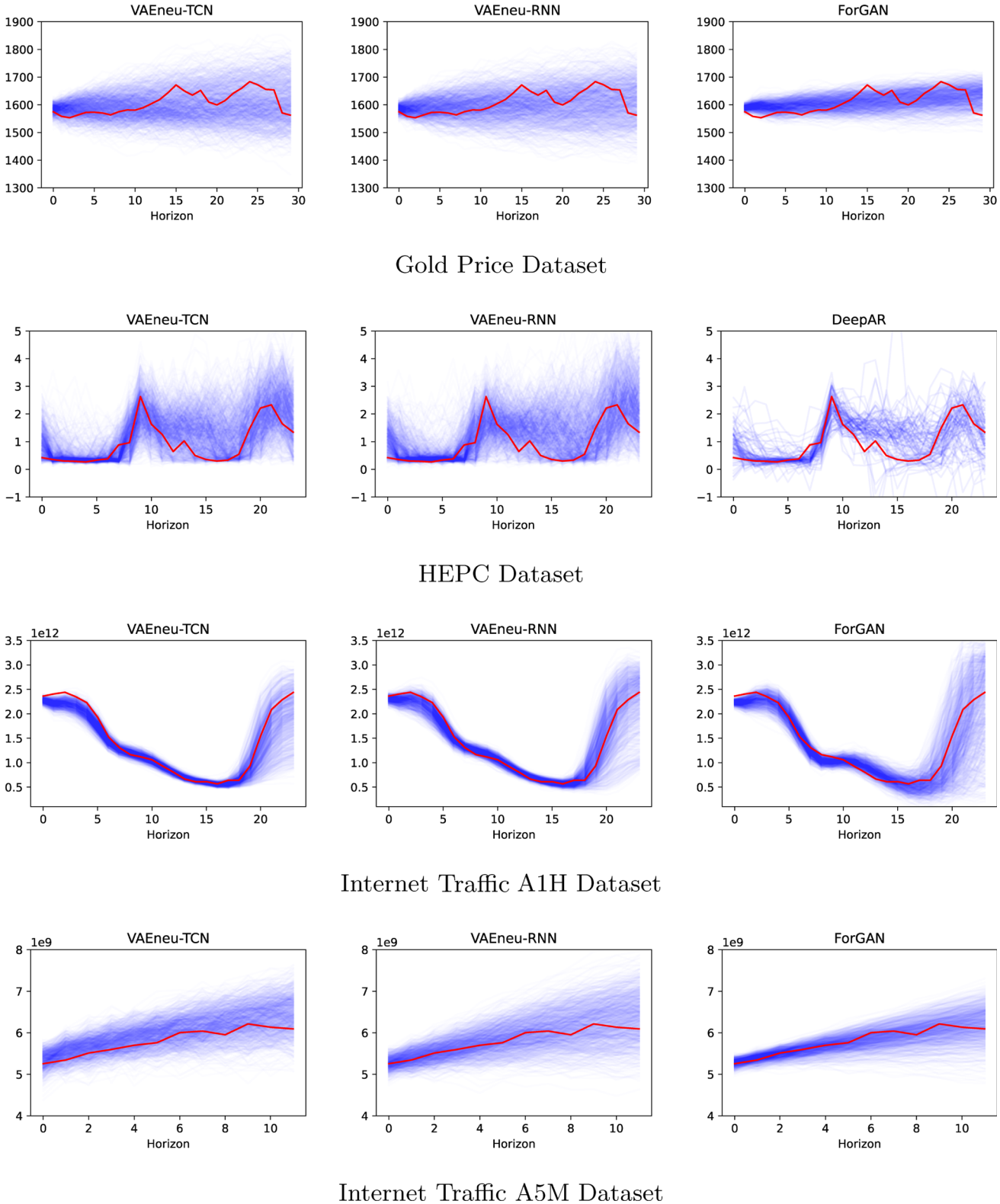


Fig. 8 Sample of the forecast for VAEneu models alongside the best baseline model

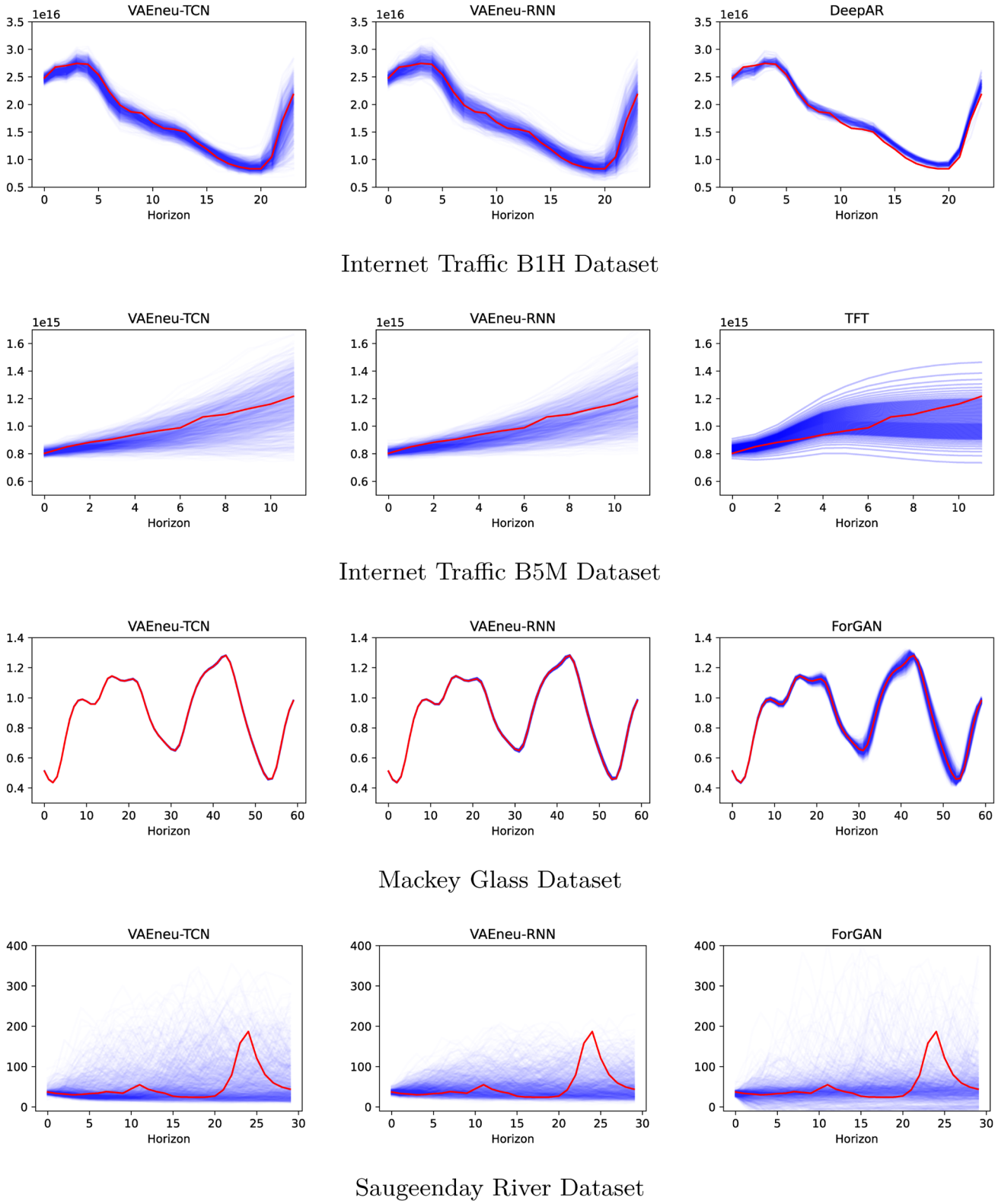


Fig. 9 Sample of the forecast for VAEneu models alongside the best baseline model

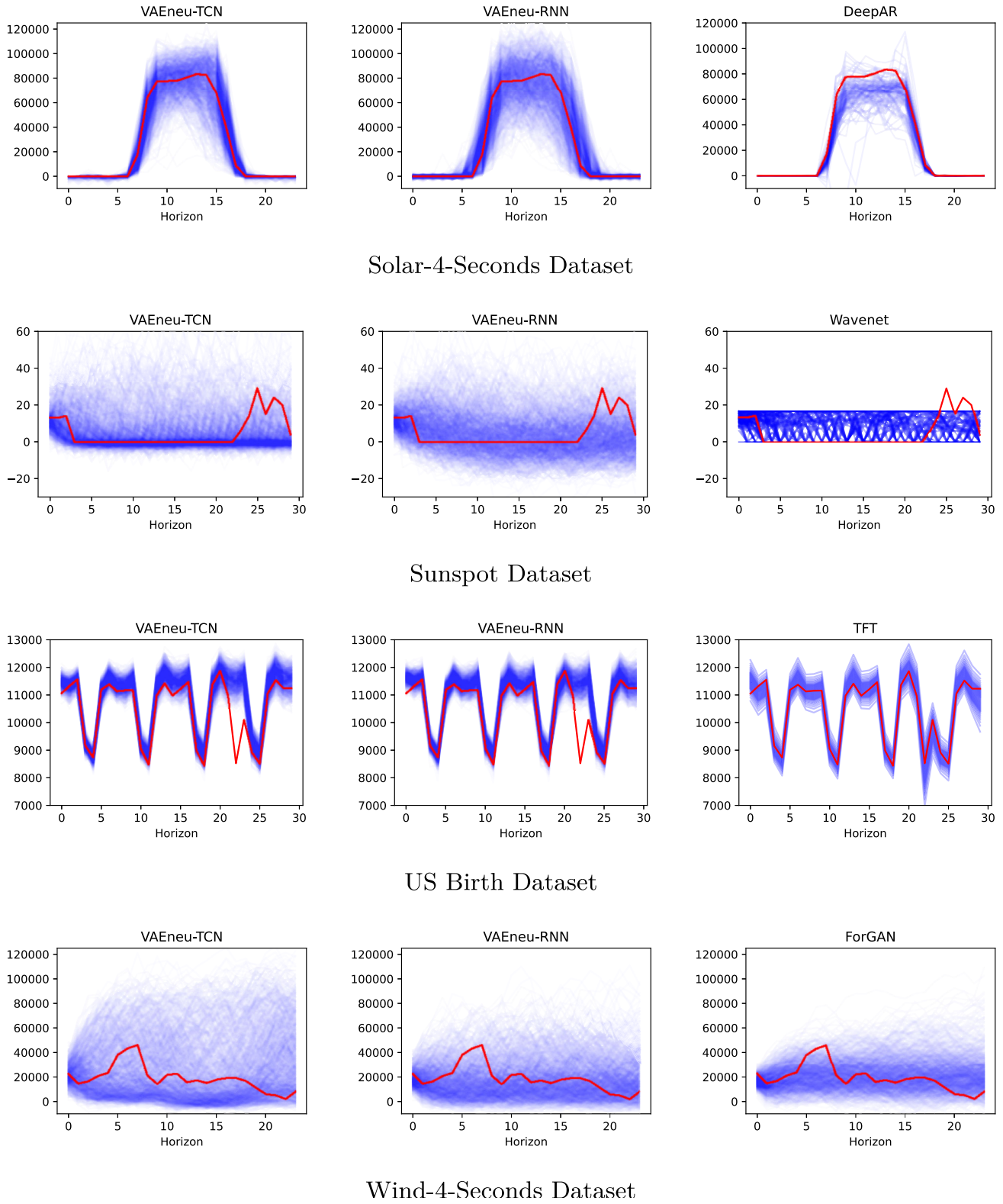
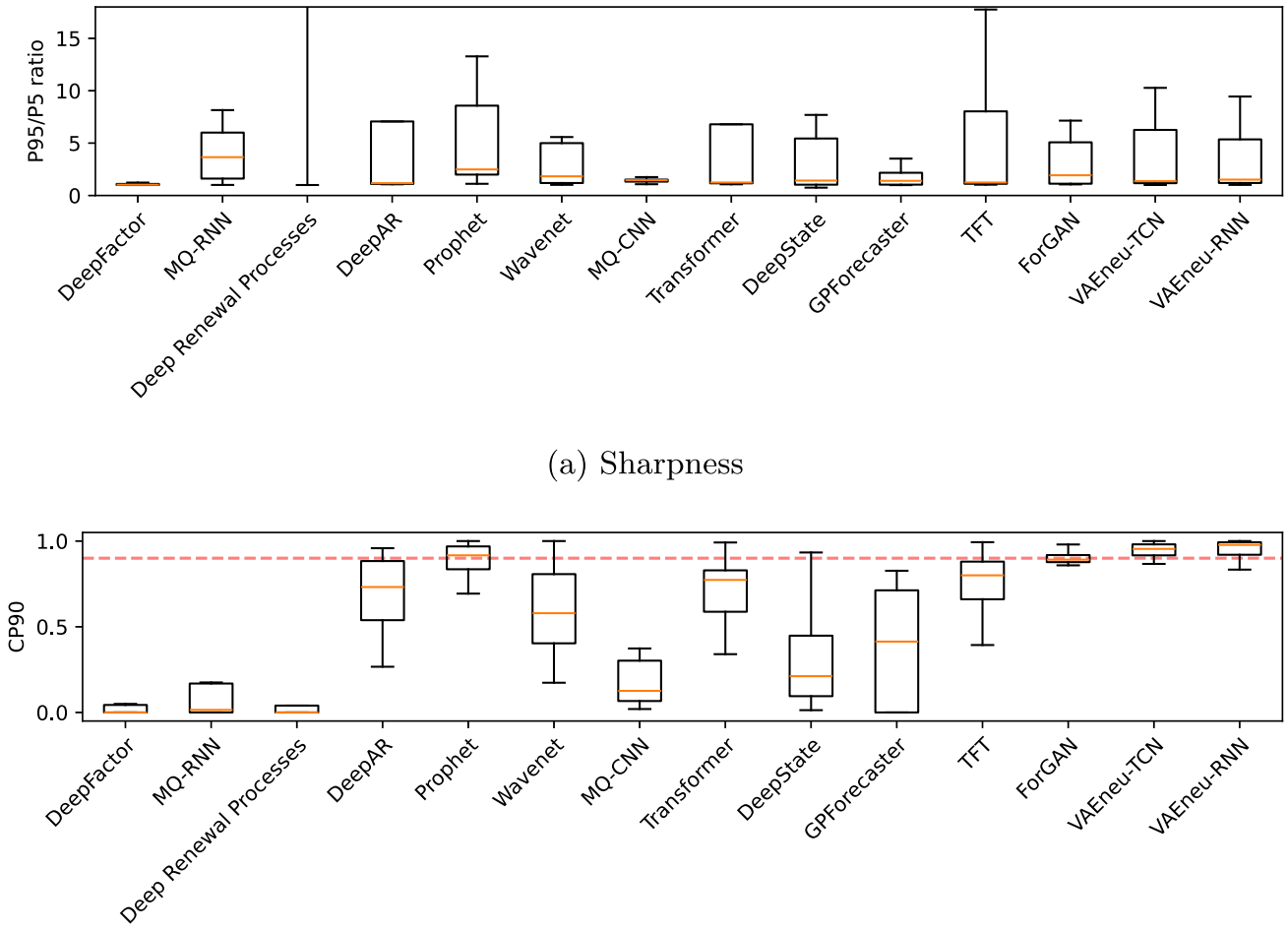


Fig. 10 Sample of the forecast for VAEneu models alongside the best baseline model



(a) Sharpness

(b) Calibration

Fig. 11 Quantification of Sharpness and Calibration of studied models using P95/P5 ratio and CP_{90} respectively. The dashed red line in (b) indicates the optimal value for a well-calibrated model. The models that fall below this line are under-calibrated, while the models that lay above this line are over-calibrated

most well-calibrated forecasts with relatively low sharpness. These results confirm that optimizing CRPS as a loss function results in well-calibrated and sharp predictive distribution. It is also worth noting that there are models with very low sharpness, such as DeepFactor, MQ-CNN, and GPForecaster, that were produced under calibrated predictive distribution.

6.3 Analyzing the impact of the sample size on training

The *sample size* designates the number of samples for each input used to approximate the CRPS during training. A larger sample size enhances the accuracy of CRPS estimation. This, in turn, should refine the quality of gradients obtained from the CRPS loss, facilitating the weight update process within the network. Nevertheless, the influence of the sample size on the model's training trajectory remains an area yet to be fully explored. In this section, we aim to clarify the consequence of varying sample size values on both the CRPS of the resultant optimal model and the average time taken to complete a training step.

To assess the interplay of different sample sizes, we trained VAEneu architectures across three distinct datasets. Sample sizes were chosen following the relation:

$$\text{sample size} = 2^i, \text{ where } i \in \{0, 1, 2, \ldots, 7\}. \tag{15}$$

Figure 12 encapsulates our findings. An evident trend emerges: as the sample size escalates, models tend to converge to a configuration with a more favorable CRPS.

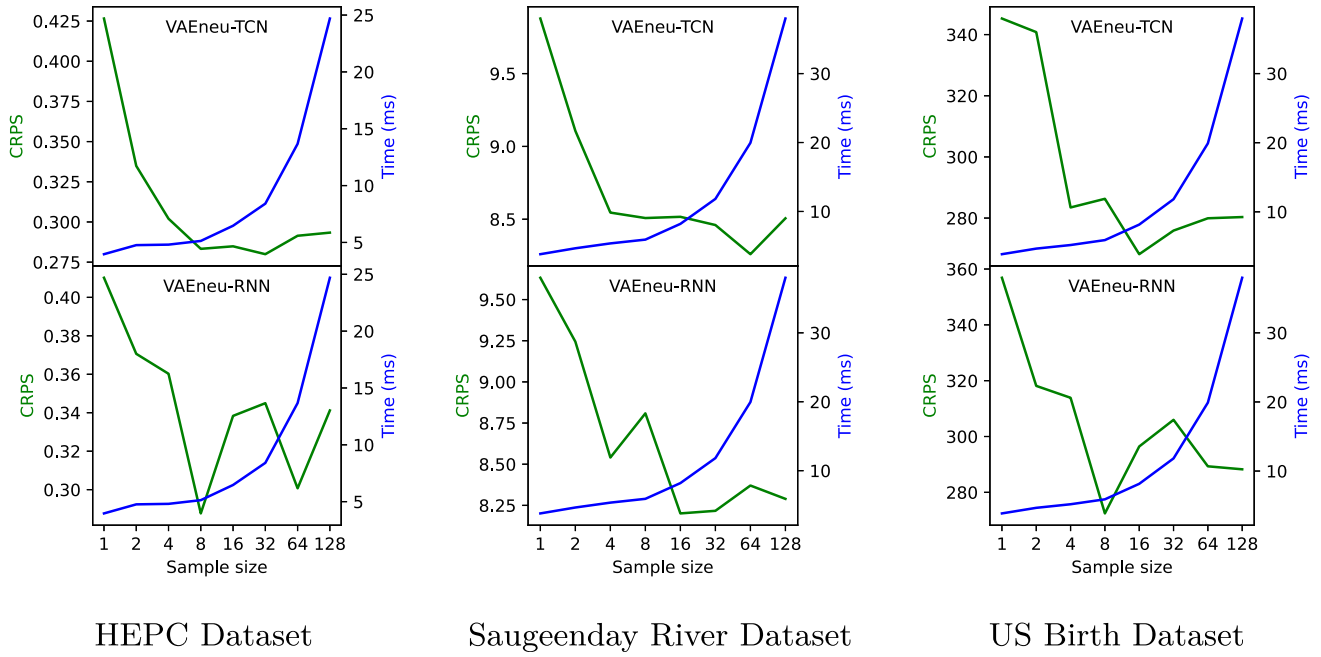


Fig. 12 The effect of sample size on the CRPS of final model and training step processing time on three datasets. A larger sample size would result in a better model in exchange for a longer training time; however, with sample size $\in$ [4, 32], a model can reach a low CRPS without increasing the batch processing time significantly

This improvement, however, comes at the cost of an increased batch processing duration. In the case where the sample size $=$ 1, the CRPS loss turns into MAE loss. From this figure, we can observe that moving from MAE loss toward CRPS loss improves the model performance significantly, underlying the effectiveness of the proposed methodology. Moreover, an interesting observation to note is that by choosing a sample size within the interval [4, 32], one can achieve decent CRPS values without significantly extending the training step processing time.

7 Conclusion

This study introduces VAEneu, a pioneering probabilistic forecasting approach utilizing CVAEs. The methodology centers on optimizing the CRPS as the loss function during training, allowing the CVAEs to learn predictive distributions effectively without imposing restrictive assumptions. Our extensive empirical evaluations demonstrate remarkable and consistent performance, and this approach emerges as a significant and innovative contribution to the field of probabilistic forecasting, offering a robust tool for decision-makers to navigate the realm of uncertainty with confidence.

Author Contributions Conceptualization: Ensiye Tahaei, Alireza Koochali, Sheraz Ahmed; Methodology: Alireza Koochali; Formal analysis and investigation: Alireza Koochali, Sheraz Ahmed; Writing - original draft preparation: Alireza Koochali; Writing - review and editing: Sheraz Ahmed; Supervision: Sheraz Ahmed, Andreas Dengel.

Funding No funding was received for conducting this study.

Data Availability All data used in this study are available publicly, and they are properly cited in Section 5.1

Declarations

Conflict of Interest/Competing Interests: The authors have no relevant financial or non-financial interests to disclose.

Ethics Approval and Consent to Participate: Not applicable

Consent for Publication: All authors agreed on the content of the paper for publication

Materials Availability: Not applicable

Code Availability: Not applicable

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

References

1. Avati A, Pfohl S, Lin C, Nguyen T, Zhang M, Hwang P, Wetstone J, Jung K, Ng AY, Shah NH (2018) Predicting inpatient discharge prioritization with electronic health records. CoRR. arXiv:1812.00371
2. Janssoone T, Bic C Kanoun D, Hornus P, Rinder P (2018) Machine learning on electronic health records: models and features usages to predict medication non-adherence. 1811–12234 arXiv e-prints arXiv:1811.12234
3. Avati A, Jung K, Harman S, Downing L, Ng A, Shah NH (2018) Improving palliative care with deep learning. BMC Med Inf Decis Mak 18(4):55–64
4. Racah E, Beckham C, Maharaj T, Kahou SE, Prabhat M, Pal C (2017) ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In: Advances in Neural Information Processing Systems, pp 3402–3413
5. Rodrigues ER, Oliveira I, Cunha R, Netto M (2018) DeepDownscale: a deep learning strategy for high-resolution weather forecast. In: 2018 IEEE 14th International Conference on E-Science (e-Science), pp 415–422. IEEE, Amsterdam, the Netherlands
6. Nevo S, Anisimov V, Elidan G, El-Yaniv R, Giencke P, Gigi Y, Hassidim A, Moshe Z, Schlesinger M, Shalev G et al (2019) ML for Flood Forecasting at Scale. arXiv preprint arXiv:1901.09583
7. Mousavi SM, Zhu W, Sheng Y, Beroza GC (2018) CRED: A Deep Residual Network of Convolutional and Recurrent Units for Earthquake Signal Detection. 1810–01965. arXiv e-prints arXiv:1810.01965
8. Ross ZE, Yue Y, Meier MA, Hauksson E, Heaton TH (2019) PhaseLink: A Deep Learning Approach to Seismic Phase Association. J Geophys Res Solid Earth 124(1):856–869. https://doi.org/10.1029/2018JB016674, https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018JB016674
9. Gensler A, Sick B (2018) A Multi-Scheme Ensemble Using Coopetitive Soft-Gating With Application to Power Forecasting for Renewable Energy Generation. arXiv preprint arXiv:1803.06344
10. Balal AT, Jafarabadi YPT, Demir AT, Igene MT, Giesselmann MT, Bayne ST (2023) Forecasting solar power generation utilizing machine learning models in lubbock
11. Al-Abayechi YF, Al-Khafaji AS (2023) Forecasting the impact of the environmental and energy factor to improve urban sustainability by using (sem). Civil Eng J 9(10):2554–2567
12. Mohamed HH, Ibrahim AH, Hagras OA (2023) Forecasting the real estate housing prices using a novel deep learning machine model. Civil Eng J 9:46–64
13. Huang WR, Perez MA (2018) Accurate, Data-efficient learning from noisy, Choice-based labels for inherent risk scoring. CoRR. arXiv:1811.10791
14. Zhang Q, Luo R, Yang Y, Liu Y (2018) Benchmarking deep sequential models on volatility predictions for financial time series. 1811–03711. arXiv e-prints arXiv:1811.03711
15. Gneiting T, Raftery AE (2007) Strictly proper scoring rules, prediction, and estimation. J Am Stat Assoc 102(477):359–378
16. Engle RF (1982) Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica J Econ Soc987–1007
17. Bollerslev T (1986) Generalized autoregressive conditional heteroskedasticity. J Econ 31(3):307–327
18. Nelson DB (1991) Conditional Heteroskedasticity in Asset Returns: A New Approach. Econ 59(2):347–370. https://doi.org/10.2307/2938260
19. Zakoian JM (1994) Threshold heteroskedastic models. J Econ Dyn Control 18(5):931–955. https://doi.org/10.1016/0165-1889(94)90039-6
20. Glosten LR, Jagannathan R, Runkle DE (1993) On the relation between the expected value and the volatility of the nominal excess return on stocks. J Finan 48(5):1779–1801. https://doi.org/10.1111/j.1540-6261.1993.tb05128.x
21. Gneiting T, Raftery AE, Westveld AH, Goldman T (2005) Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. Mon Weather Rev 133(5):1098–1118. https://doi.org/10.1175/MWR2904.1
22. Raftery AE, Gneiting T, Balabdaoui F, Polakowski M (2005) Using bayesian model averaging to calibrate forecast ensembles. Mon Weather Rev 133(5):1155–1174. https://doi.org/10.1175/MWR2906.1
23. Frazier DT, Maneesoonthorn W, Martin GM, McCabe BPM (2019) Approximate bayesian forecasting. Int J Forecast 35(2):521–539. https://doi.org/10.1016/j.ijforecast.2018.08.003
24. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735. Accessed: 26 Jan 2024
25. Cho K, Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv arXiv:1406.1078 [cs, stat]. https://doi.org/10.48550/arXiv.1406.1078. Accessed 26 Jan 2024
26. Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv arXiv:1803.01271 [cs]. https://doi.org/10.48550/arXiv.1803.01271. Accessed 26 Jan 2024
27. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: advances in neural information processing systems, vol 27. Curran Associates, Inc., Montréal,Canada
28. Bahdanau D, Cho K, Bengio Y (2016) Neural machine translation by jointly learning to align and translate. arXiv. arXiv:1409.0473 [cs, stat]. https://doi.org/10.48550/arXiv.1409.0473. Accessed 26 Jan 2024
29. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: advances in neural information processing systems, vol 30. Curran Associates, Inc., California, USA
30. Salinas D, Flunkert V, Gasthaus J, Januschowski T (2020) DeepAR: Probabilistic forecasting with autoregressive recurrent networks. Int J Forecast 36(3):1181–1191
31. Lim B, Arik SO, Loeff N, Pfister T (2021) Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. Int J Forecast 37(4):1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012
32. Wen R, Torkkola K, Narayanaswamy B, Madeka D (2018) A Multi-Horizon Quantile Recurrent Forecaster. arXiv . https://doi.org/10.48550/arXiv.1711.11053
33. Koochali A, Schichtel P, Dengel A, Ahmed S (2019) Probabilistic forecasting of sensory data with generative adversarial networks – ForGAN. IEEE Access 7:63868–63880. https://doi.org/10.1109/ACCESS.2019.2915544
34. Zhou X, Pan Z, Hu G, Tang S, Zhao C (2018) Stock market prediction on high-frequency data using generative adversarial nets. Math Probl Eng 2018:4907423. https://doi.org/10.1155/2018/4907423
35. Khosravi A, Nahavandi S, Creighton D (2013) A neural network-GARCH-based method for construction of Prediction Intervals. Electr Power Syst Res 96:185–193. https://doi.org/10.1016/j.epsr.2012.11.007
36. Oord Avd, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499
37. Rangapuram SS, Seeger MW, Gasthaus J, Stella L, Wang Y, Januschowski T (2018) Deep State Space Models for Time Series Forecasting. In: Bengio S, Wallach H, Larochelle H, Grauman K,

Cesa-Bianchi N, Garnett R (eds) Adv Neural Inf Process Syst, vol 31. Curran Associates Inc, Montréal, Canada
38. Rasp S, Lerch S (2018) Neural Networks for Postprocessing Ensemble Weather Forecasts. Mon Weather Rev 146(11):3885–3900. https://doi.org/10.1175/MWR-D-18-0187.1
39. Taylor SJ, Letham B (2018) Forecasting at scale. Am Stat 72(1):37–45. https://doi.org/10.1080/00031305.2017.1380080
40. Hastie TJ, Tibshirani RJ (1990) Generalized additive models. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, Oxfordshire, United Kingdom
41. Graves A (2014) Generating sequences with recurrent neural networks. arXiv.https://doi.org/10.48550/arXiv.1308.0850
42. Gasthaus J, Benidis K, Wang Y, Rangapuram SS, Salinas D, Flunkert V, Januschowski T (2019) Probabilistic forecasting with spline quantile function RNNs. In: Proceedings of the 22nd international conference on artificial intelligence and statistics, pp 1901–1910. PMLR, Okinawa, Japan
43. Salinas D, Bohlke-Schneider M, Callot L, Medico R, Gasthaus J (2019) High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes. In: advances in neural information processing systems, vol 32. Curran Associates, Inc., Vancouver Canada
44. Wang Y, Smola A, Maddix D, Gasthaus J, Foster D, Januschowski T (2019) Deep factors for forecasting. In: Proceedings of the 36th international conference on machine learning, pp 6607–6617. PMLR, California,USA
45. Chen Y, Kang Y, Chen Y, Wang Z (2020) Probabilistic forecasting with temporal convolutional neural network. Neurocomput 399:491–501. https://doi.org/10.1016/j.neucom.2020.03.011
46. Rasul K, Sheikh AS, Schuster I, Bergmann U, Vollgraf R (2020) Multi-variate probabilistic time series forecasting via conditioned normalizing flows. arXiv preprint arXiv:2002.06103
47. Bézenac E, Rangapuram SS, Benidis K, Bohlke-Schneider M, Kurle R, Stella L, Hasson H, Gallinari P, Januschowski T (2020) Normalizing kalman filters for multivariate time series analysis. In:advances in neural information processing systems, vol 33, pp 2995–3007. Curran Associates, Inc., Virtual
48. Rezende D, Mohamed S (2015) Variational inference with normalizing flows. In: Proceedings of the 32nd international conference on machine learning, pp 1530–1538. PMLR, Lille, France
49. Gouttes A, Rasul K, Koren M, Stephan J, Naghibi T (2021) Probabilistic time series forecasting with implicit quantile networks. arXiv. https://doi.org/10.48550/arXiv.2107.03743
50. Dabney W, Ostrovski G, Silver D, Munos R (2018) Implicit quantile networks for distributional reinforcement learning. In: Proceedings of the 35th international conference on machine learning, pp 1096–1105. PMLR, Stockholm, Sweden
51. Hasson H, Wang B, Januschowski T, Gasthaus J (2021) Probabilistic forecasting: A level-set approach. Adv Neural Inf Process Syst 34:6404–6416
52. Rasul K, Seward C, Schuster I, Vollgraf R (2021) Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In: Proceedings of the 38th international conference on machine learning, pp 8857–8868. PMLR, Virtual
53. Türkmen AC, Januschowski T, Wang Y, Cemgil AT (2021) Forecasting intermittent and sparse time series: A unified probabilistic framework via deep renewal processes. PLOS ONE. 16(11):0259764. https://doi.org/10.1371/journal.pone.0259764
54. Croston JD (1972) Forecasting and Stock Control for Intermittent Demands. J Oper Res Soc 23(3):289–303. https://doi.org/10.1057/jors.1972.50
55. Xu Q, Liu S, Jiang C, Zhuo X (2021) QRNN-MIDAS: A novel quantile regression neural network for mixed sampling frequency data. Neurocomput 457:84–105. https://doi.org/10.1016/j.neucom.2021.06.006
56. Kan K, Aubet FX, Januschowski T, Park Y, Benidis K, Ruthotto L, Gasthaus J (2022) Multivariate quantile function forecaster. arXiv. https://doi.org/10.48550/arXiv.2202.11316
57. Amos B, Xu L, Kolter JZ (2017) Input convex neural networks. In: Proceedings of the 34th international conference on machine learning, pp 146–155. PMLR, Sydney, Australia
58. Koochali A, Dengel A, Ahmed S (2021) If you like it, GAN It—Probabilistic multivariate times series forecast with GAN. Eng Proc 5(1). https://doi.org/10.3390/engproc2021005040
59. Zhang K, Zhong G, Dong J, Wang S, Wang Y (2019) Stock market prediction based on generative adversarial network. Procedia Comput Sci 147:400–406
60. Lin Y, Dai X, Li L, Wang F-Y (2019) Pattern Sensitive Prediction of Traffic Flow Based on Generative Adversarial Framework. IEEE Trans Intell Transp Syst 20(6):2395–2400. https://doi.org/10.1109/TITS.2018.2857224
61. Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, Zhang W (2021) Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. Proc AAAI Conf Artif Intell 35(12):11106–11115. https://doi.org/10.1609/aaai.v35i12.17325. Number: 12. Accessed 31 Jan 2024
62. Wu H, Xu J, Wang JLong M (2022) Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. arXiv. arXiv:2106.13008 [cs]. https://doi.org/10.48550/arXiv.2106.13008
63. Zhou T, Ma Z, Wen Q, Wang X, Sun L, Jin R (2022) FEDformer: frequency enhanced decomposed transformer for long-term series forecasting. arXiv. arXiv:2201.12740 [cs, stat]. https://doi.org/10.48550/arXiv.2201.12740
64. Chen SA, Li CL, Yoder N, Arik SO, Pfister T (2023) TSMixer: An all-MLP architecture for time series forecasting. arXiv. arXiv:2303.06053 [cs]. https://doi.org/10.48550/arXiv.2303.06053
65. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114
66. Sohn K, Lee H, Yan X (2015) Learning structured output representation using deep conditional generative models. In: advances in neural information processing systems, vol 28. Curran Associates, Inc., Montréal, Canada
67. Gold Price Dataset. www.kaggle.com/datasets/arashnic/learn-timeseries-forecasting-from-gold-price
68. Hebrail G, Berard A (2012) Individual household electric power consumption. UCI Machine Learning Repository
69. Cortez P, Rio M, Rocha M, Sousa P (2012) Multi-scale internet traffic forecasting using neural networks and time series methods. Expert Syst 29(2):143–155. Publisher: Wiley Online Library
70. Mackey MC, Glass L (1977) Oscillation and chaos in physiological control systems. Sci 197(4300):287–289. Publisher: American Association for the Advancement of Science
71. Méndez E, Lugo O, Melin P (2017) A competitive modular neural network for long-term time series forecasting. Nat-inspired Des Hybrid Intell Syst. Springer, Cham, pp 243–254
72. Macky-Glass Dataset. https://git.opendfki.de/koochali/forgan/-/tree/master/datasets/mg
73. Ogie RI, Shukla N, Sedlar F, Holderness T (2017) Optimal placement of water-level sensors to facilitate data-driven management of hydrological infrastructure assets in coastal mega-cities of developing nations. Sustain Cities Soc 35:85–395. https://doi.org/10.1016/j.scs.2017.08.019
74. Godahewa R, Bergmeir C, Webb GI, Hyndman RJ, Montero-Manso P (2021) Monash time series forecasting archive. In: neural information processing systems track on datasets and benchmarks
75. Pruim R, Kaplan D, Horton N (2022) mosaicData: Project MOSAIC data sets

76. Alexandrov A, Benidis K, Bohlke-Schneider M, Flunkert V, Gasthaus J, Januschowski T, Maddix DC, Rangapuram S, Salinas D, Schulz J, Stella L, Türkmen AC, Wang Y (2020) GluonTS: probabilistic and neural time series modeling in python. J Mach Learn Res 21(116):1–6
77. Tieleman T, Hinton G et al (2012) Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw Mach Learn 4(2):26–31
78. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) PyTorch: an imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems 32. Curran Associates Inc, Vancouver, Canada, pp 8024–8035

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.