Predicting grid frequency short-term dynamics with Gaussian processes and sequence modeling

Bolin Liu*
bolin.liu@kit.edu
Karlsruhe Institute of Technology (KIT), Institute for Operations Research, Analytics and Statistics
Karlsruhe, Germany

Maximilian Coblenz
maximilian.coblenz@hwg-lu.de
Department of Services and Consulting, Ludwigshafen University of Business and Society
Ludwigshafen, Germany

Oliver Grothe
grothe@kit.edu
Karlsruhe Institute of Technology (KIT), Institute for Operations Research, Analytics and Statistics
Karlsruhe, Germany

ABSTRACT

Modeling and predicting grid frequency is an important task in power system control. While the consideration of external techno-economic features could improve the modeling of the short-term dynamics of grid frequency, such features are often only recorded on an hourly basis and require careful treatment. We present a purely data-driven approach to modeling grid frequency as an alternative to prediction models incorporating physical characteristics of power systems. Using sequence models such as gated recurrent units and transformers, we extract the necessary information and relationships from the static frequency vector to predict the process parameters for the short-term dynamics of frequency following a Gaussian process. Both for the evaluation measures (e.g. MSE, MAE, RMSE) for point estimators and for the measures for probalistic evaluations (e.g. Negative Log Likelihood Score, CPRS and Energy Score), our prediction performance is comparable to state-of-the-art models and outperforms various purely data-driven models such as daily profiles and k-nearest-neighbour profiles. Moreover, synthetic time series generated by our models can successfully reproduce the main statistical characteristics of the grid frequency.

CCS CONCEPTS

• **Applied computing → Forecasting**.

KEYWORDS

data driven modeling, electric power system, power-grid frequency

ACM Reference Format:
Bolin Liu, Maximilian Coblenz, and Oliver Grothe. 2024. Predicting grid frequency short-term dynamics with Gaussian processes and sequence modeling. In *The 15th ACM International Conference on Future and Sustainable Energy Systems (E-Energy '24), June 04–07, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3632775.3662160

1 INTRODUCTION

In our modern society, a stable power supply is crucial for our daily lives and guarantees economic activities. Power supply is linked to grid stability for which a quality factor is the grid frequency. If there is an imbalance between power generation and power consumption the grid frequency deviates from the reference frequency. Therefore, an accurate model of the grid frequency is extremely important for simulations and predictions related to grid stability. Yet, the increasing share of renewables in the energy supply makes grid frequency modeling even more challenging due to the volatility and unpredictability of renewable energy [21].

In the literature, an oscillation or motion equation motivated by the physical nature of the grid is generally assumed when considering the grid frequency [5, 23, 25]. In [14], the stochastic Ornstein-Uhlenbeck process is modified and a fractal noise statistic is proposed to realistically model the grid frequency in Great Britain, taking into account static properties such as fat tails and bimodality. In addition to random components, e.g., fluctuations, external influences such as technical and economic conditions must be modeled realistically [18]. In [7], a dynamic model is formulated whose parameters take into account the influences of the fundamental control systems, the market and noise. In order to develop an accurate model for grid frequency dynamics, even more features need to be taken into account. However, it is a major challenge to incorporate the technical and economic features into the modeling and prediction of the short-term development of the frequency deviation at the level of seconds, as the features are usually recorded hourly. The current study in [16] represents a special step towards solving this problem, in which a physically based machine learning model is presented whose physical model equations can take into account the influence of operating conditions in the form of techno-economic parameters on the short-term dynamics of the frequency control system.

Although physical models undoubtedly provide a solid basis for modeling the dynamics of grid frequency, the interesting question is whether a fully data-driven approach without detailed modeling of the physical principles can lead to comparable results. The aim of our paper is to provide exactly this kind of data-driven model. In the following, the frequency behaviour in Central Europe is considered as an example to demonstrate the methodology. Note that the methodology presented is not only applicable to large power operation systems, but also to micro or distributed energy sources. For example, characteristics can be extracted from the data of a smart meter and used to model or predict the local frequency

*Corresponding author

E-Energy '24, June 04–07, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0480-2/24/06
https://doi.org/10.1145/3632775.3662160

deviation. In this work, we want to explore the possibility of data-driven modeling of an energy system without precise physical models, which are not readily available for every energy system in reality.

A popular approach for modeling stochastic processes are Gaussian processes, which have proven to be a valuable tool in many areas due to their flexibility and strong modeling capabilities. Also, Gaussian processes become increasingly important in energy forecasting. For example, in [26] the hourly probability density of the electricity load is predicted using quantile regression of the Gaussian process. In addition to Gaussian processes, sequence models such as long short-term memory (LSTM), gated recurrent unit (GRU), or transformer have proven to be particularly useful for processing sequential data. These techniques were originally developed to understand and process natural language and have achieved great success in many areas [10, 27]. Furthermore, they become increasingly important for the prediction of time series. In [13], a CNN-LSTM model is successfully used to extract complex energy consumption features and predict the energy consumption of residential buildings. In [28], it is discussed how a transformer-encoder architecture can be used to learn multivariate time series representations.

Our paper presents a fully data-driven approach based on a combination of Gaussian processes and sequence models in order to model and to predict the short-term evolution of frequency deviations complementing models that take into account physical properties. For training, we use external feature values (e.g. day-ahead forecasts of load, generation, price and more, see Appendix C for details) recorded at the beginning of an hour, in conjunction with the corresponding time series data of grid frequency that were recorded within the same hour. The paper is structured as follows: Section 2 introduces the Gaussian process as a model for the frequency deviation and discusses solution approaches to account for correlations between time points. In Section 3, we show how the information in the techno-economic features can be extracted using GRU and transformer architectures to directly predict the Gaussian process without physical modeling. In particular, we show the possibilities to consider the fat tail behaviour by changing the marginal distributions of the stochastic processes. Section 4 contains an overview of the data used and models developed. In addition to the training details, we also present the baseline models and evaluation measures here. In Section 5, we present the evaluation results of our approaches to probabilistic forecasting and synthetic data generation in comparison to various baseline models. The paper ends with a conclusion. The source code of this work is available at [20].

2 GRID FREQUENCY MODEL BASED ON GAUSSIAN PROCESS

In this section, we derive a Gaussian process as a framework for modeling the short term evolution of grid frequency. We address the difficulty of modeling the correlation between time points and present solutions using covariance matrices with particular structures.

2.1 Model Setup

In the following, we denote the reference grid frequency by f_{ref}. For example, a reference frequency of 50 Hz is used in continental Europe. As the reference grid frequency remains constant over time, it is sufficient to model the deviation of the actual frequency from the reference, denoted as $\Delta f := f - f_{\text{ref}}$.

Let $(\Omega, \mathcal{A}, P)$ be a suitable probability space. We assume that the short-term dynamics of the grid frequency over a period of one hour, starting at time t_{start}, can be described by a Gaussian process $\Delta f : T \longrightarrow L^2(\Omega)$ with time interval $T = [t_{\text{start}}, t_{\text{end}}]$. The Gaussian process Δf is uniquely defined by a mean function $\mu(t) := \mathbb{E}[\Delta f_t]$ and a covariance function $\mathbf{C}(t, \tau) := \mathtt{Cov}[\Delta f(t), \Delta f(\tau)],\ t, \tau \in T$. L^2 is the space of square integrable functions and guarantees that variance $\sigma^2(t)$ and covariance $Cov(t, \tau), t \neq \tau$, are finite. In particular, the Gaussian process has the nice property that the random vector $(\Delta f(t_1), \cdots, \Delta f(t_n))^T$ follows an n-dimensional Gaussian distribution for any choice of $t_1, \ldots, t_n \in T$. A detailed discussion of Gaussian processes and their typical applications in machine learning can be found in [22].

The flexibility and power of the Gaussian process, as evidenced by its successful application in various fields [4, 8, 11], makes it a suitable candidate for modeling the complex dynamics of frequency deviation. Furthermore, since many physical models (such as the diffusion equation) ultimately lead to (multivariate) normal distributions under regularity conditions, it is justified in this sense to ask whether one can skip the intermediate step via (physical) model equations and, in our case, model the frequency deviation directly with a univariate Gaussian process. In particular, we do not use a specific physical model and associated stochastic differential equations, but try to learn the Gaussian process directly from the available day-ahead data. This purely data-driven model offers great flexibility in modeling.

For the model, we now consider an index set with discrete time points $I = \{t_0, \ldots, t_{N-1}\}$. For $N = 3600$, the underlying Gaussian process exactly reproduces the change in grid frequency per second. In this case, the Gaussian process can be represented by a multivariate Gaussian distribution with the N-dimensional mean vector $\mu \in \mathbb{R}^N$ and the covariance matrix $\mathbf{C} \in \mathbb{R}^{N \times N}$. In the following, we introduce approaches that identify the mean vector and the covariance matrix directly using advanced sequence models. Note that the discretization is not a strong constraint for Gaussian process with independent time points, as the mean and variance functions can be reconstructed using interpolation techniques such as splines.

2.2 Treatment of Serial Dependency

The serial dependence structure of a Gaussian process is generally determined by its covariance function. For the discrete stochastic process Δf, we represent this dependence either directly by covariance matrices or by kernel representations. Handling a large and full covariance matrix and estimating all its parameters can be very challenging and computationally intensive. To reduce complexity and improve identifiability, special matrix structures can be adopted. In particular, we assume a block structure of the covariance matrix (with a diagonal matrix as the simplest candidate) and design suitable kernel specifications. Both are discussed in more detail below.

As a structural simplification for the covariance matrix, a band structure or a diagonal block structure can be assumed. A band structure covariance for the Gaussian process $\Delta f : T \to L^2(\Omega)$ means that the frequency values are correlated only within a certain rolling time window, while the block structure of a covariance matrix implies that the time points can be divided into groups (represented by the blocks) within which significant relationships (covariances) exist, while no or only weak relationships exist between the groups. A simple but important special case of the band structure (bandwidth equal to 1) and the diagonal block matrix is the diagonal covariance matrix. In this case, the time points are independent of each other and only the variance has to be modeled for each time point. It could also be useful to divide the time points into four groups with time points of 15 minutes each. For example, market trading takes place at discrete intervals (such as 15 minutes). At the start of a new 15 minute interval, power generation is rapidly adjusted to meet the new trading conditions and demands. This leads to regular jumps in frequency dynamics [17].

Another flexible yet powerful method for defining covariance functions for Gaussian processes is the use of special kernel functions that have a predefined functional form with a limited number of hyper parameters. When using predefined kernel matrices, only the hyper parameters need to be identified, which significantly reduces the complexity of determining the complete covariance matrix. A kernel frequently used in practice is the exponentiated quadratic kernel

$$k_{\mathrm{SE}}(t,t') = \sigma^2 \exp\left(-\frac{(t-t')^2}{2\ell^2}\right),$$

where σ is the variance amplitude and ℓ is the characteristic length scale. Rational quadratic kernels can model variations over multiple length scales with the additional parameter α

$$k_{\mathrm{RQ}}(t,t') = \sigma^2 \left(1 + \frac{(t-t')^2}{2\alpha\ell^2}\right)^{-\alpha}.$$

A more detailed discussion of the properties of kernel functions can be found in [22]. Appendix A shows different kernels and synthetic data for various hyper parameters. In particular, covariance matrices defined by kernel functions can have an approximate band structure (see Fig. 7 in Appendix A). Note that one advantage of using standard kernels is that certain kernel combinations also produce valid covariance matrices. For example, the addition or matrix multiplication of two covariance matrices again results in a covariance matrix, so that specific synthetic data can be generated if kernels with different patterns are suitably combined.

In this paper, we first assume that there is no correlation between the frequency deviations Δf at different points in time. In addition to the independence assumption, we also investigate to what extent the consideration of correlations by kernels can improve the results (see Section 5). Having discussed the basic model set-ups, now we turn to the next key step: identifying Gaussian process parameters using sequence models.

3 PROCESS LEARNING USING SEQUENCE MODELS

In this section, we first formally introduce the underlying learning task. From the requirements, we derive specific customised sequential models to efficiently and effectively process and extract the information from the techno-economic features, which are then used to predict distribution parameters of the Gaussian processes (cf. Fig. 1).

3.1 Learning Task and Loss functions

To complete our frequency model based on Gaussian processes, the process parameters (i.e. the mean vector μ and the covariance matrix $\mathbf{C}$) must be determined. As discussed in detail in [16, 18], the stochastic process of the grid frequency is influenced by external techno-economic properties. Therefore, we could design a method to predict the process parameters based on the external techno-economic features, see Fig. 1. Specifically for forecasting purposes, we focus below on constructing a model that processes the available day-ahead features. In particular, we use the following day ahead features: day ahead forecasts of the load, renewable generation, day ahead electricity prices, the planned generation, their respective increases and information on the hour of the day (for details see Appendix C).

In [16], the frequency value at the beginning of a period is used to initialize the learned stochastic processes and is not listed directly as a feature. Here we also use the frequency deviation of time begin as a feature value directly. For a time interval, we can synthesize a feature vector by combining the hourly resolved values of the external features described above and the initial value of the grid frequency at the beginning of this time interval. The learning task presented is a typical supervised learning task. For each input feature vector representing the economic-technical state at the beginning of an hour, we use the data set of N grid frequency values within the corresponding hour as the true values for training and testing.

Since we are modeling the grid frequency with Gaussian processes, we now need a model that predicts the parameters of the Gaussian process from the features. A major challenge is that the values of the available features are typically published at much lower frequencies, e.g. only at the beginning of an hour, resulting in an unbalanced size of the input and output dimensions. Therefore, models are needed that meaningfully transform the input data into higher dimensional data spaces, taking into account the communication possibilities between the values in the output sequences. This consideration leads to the use of a recurrent neural network structure, in particular a gated recurrent unit (GRU), which is an efficient and effective method for modeling time series data (see Section 3.2.1 for implementation details). Another idea to solve the problem described above is to use a transformer-like structure based on the attention mechanism. By using multi-attention headers, different aspects of the feature vector can be learned. After information processing by the GRU or transformer, the learned information about the relationships is further processed by fully connected layers to compute the process parameters. A simple model structure such as a dense neural network with fully connected layers would not be suitable since fully connected layers do not directly take into

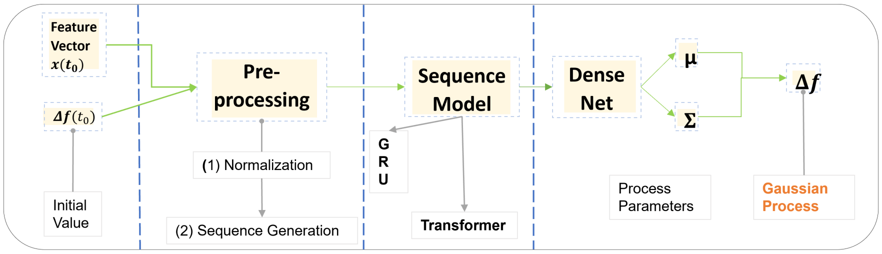


Figure 1: Overview of the entire model structure. The techno-economic characteristics and the value of the frequency deviation at the beginning of the hour are processed after a pre-processing step by sequence models such as GRU or Transformer to extract correlations, which are then used by a dense network to predict the mean vectors and the covariance matrix of the frequency deviations modeled as a Gaussian process within the hour.

account the latent structure of the data at either the input or the output, so learning the data representation in this way is inefficient.

Before presenting the adapted sequence models for predicting process parameters, we first introduce the following loss functions. Denoting the function learnt by the neural network as φ, we define the negative log likelihood loss function for X, a batch of input feature vector $\mathbf{x}_k$, as follows

$$\mathcal{L}(X) = \frac{1}{K}\sum_{k=1}^{K} -\log(p_{\varphi(\mathbf{x}_k)}(y_k)),$$

where $p_{\varphi(\mathbf{x}_k)}(\cdot)$ is the density function of the multivariate normal distribution with parameters encoded in the form of $\varphi(\mathbf{x}_k)$ and $y_k \in \mathbb{R}^N$ is the time series of frequency associated with the feature vector x_k.

Assuming that the frequency is uncorrelated between different points in time, the loss function simplifies to

$$\mathcal{L}(X) = \frac{1}{K}\frac{1}{N}\sum_{k=1}^{K}\sum_{n=0}^{N-1}\Big(\frac{1}{2}\log(2\pi) + \log(\sigma_n(x_k)) + \frac{1}{2\sigma_n(x_k)^2}(y_{k,n} - \mu_n(x_k))^2\Big).$$

If the correlation of the frequency is modeled by a kernel matrix $\mathbf{K}$, then the loss function is calculated as

$$\mathcal{L}(X) = \frac{1}{K}\sum_{k=1}^{K}\Big(\frac{N}{2}\log(2\pi) + \frac{1}{2}\log(|\mathbf{K}(x_k)|) + \frac{1}{2}(y_k - \mu(x_k))^T(\mathbf{K}(x_k))^{-1}(y_k - \mu(x_k))\Big).$$

As shown in [14, 16], large frequency deviations are more likely than a normal distribution would predict. Since in a Gaussian process the expected value function and covariance function do not need to be constant, the aggregate distribution of all time points of a Gaussian process can produce a different tail behaviour than a normal distribution. However, one could ask whether a relaxation of the Gaussian limits to fat-tail distributions could lead to an even more realistic representation of the tail behaviour. To this end, we also consider the Student-t distribution and the Cauchy distribution, which is a special case of a Student-t distribution, for each time point. The dynamics of the frequency deviation is modeled by a stochastic process where we have a fat tail distribution at each time point. In this case, we also assume that the time points are independent. In particular, we will learn a location-scale Student-t distribution to account for different mean positions and scattering behaviour at each time point. For stochastic process with marginal Cauchy distributions, we focus on learning the median and interquartile range. Details on the loss functions for fat-tail marginal distributions are provided in Appendix B. Here the flexibility of our approach of using sequence models (details see Section 3.2) for feature processing becomes apparent. In order to take different stochastic processes into account, we only need to exchange the loss function and and the rest of the model structure remains the same.

3.2 Information Extraction using Sequence-to-Sequence Models

In this section, we present the GRU and transformer-like neural network structure specially adapted for modeling the short-term dynamics of grid frequency. To solve the problem that the values of the available features are recorded hourly, but the frequency values are recorded every second, we use custom GRU and Transformer-Structure. Whereas in GRU the techno-economical feature vector is artificially repeated for a number of points in time and then processed by an "autoregressive" type of network structure, the transformer-like structure attempts to learn different aspects from a static feature vector by multi-head attention.

3.2.1 GRU-Structure. We assume the same macro-techno-economic state for a hour with time index set I. In addition, we assume a hidden state $h_t, t \in I$ for each time point, which determines the (distribution of the) frequency deviation, e.g., the mean and the variance at that time point. Following an autoregressive modeling approach, one would calculate the hidden state h_{t+1} for time $t+1$ as a function of h_t and the global techno-economic state x. This modeling principle is implemented below using a GRU structure [1]. For this purpose, we consider the following process in Fig. 2. For each time step, we assume the same techno-economic feature vector $x_t = x$ as input. This can be achieved by repeating the input feature vector x for a number of prediction time points, $x_0 = \cdots = x_{N-1} = x$.

To make a prediction for the time t, a preliminary hidden state h'_t is first estimated. To do this, the feature vector $x_t = x$ is processed with fully connected layers to calculate a value $r_t \in [0, 1]$. r_t

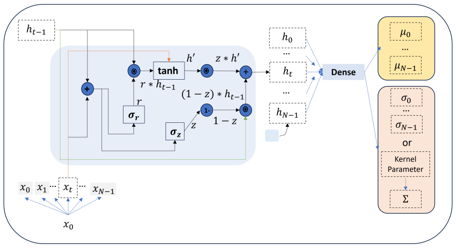


Figure 2: Information processing using GRU to predict frequency deviation. The feature vector is processed at each time point with a hidden state from the last time point to get the current hidden state. The hidden states at all time points are then passed through a dense net to determine the mean and covariance matrices.

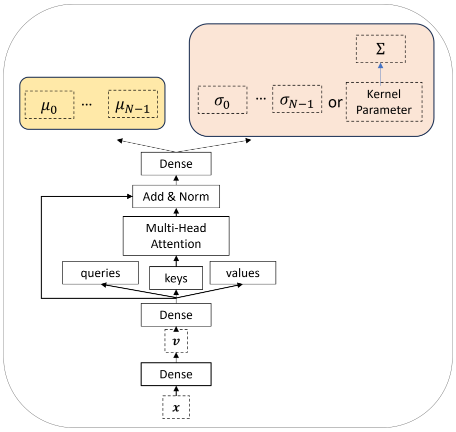


Figure 3: Extraction of relationship information between features using a transformer encoder structure. The static feature vector is used to generate embedding vectors, to which the multi-head attention is then applied to learn the different aspects of the feature relationships, which are passed through a dense network to predict the process parameters of the Gaussian process. Parts of the illustration are based on the original illustration of the transformer structure in [24].

represents the proportion of information from h_{t-1} that propagates from time t-1 to t. If r_t is close to 0, this means that the sub-neural network has reset the information from the previous state. It is therefore also referred to as a reset gate. A value of $r_t = 1$ means that the influence of the past is strongly taken into account, which can be interpreted as a complete propagation of the previous state component.

The preliminary hidden state h'_t can then be calculated from $r_t * h_{t-1}$ and x_t. To get a final estimate for the hidden state, the preliminary hidden state h'_t can be weighted with the previous hidden state h'_t with

$$h_t = z_t \cdot h'_t + (1 - z_t) \cdot h_{t-1}.$$

z_t is calculated similarly to r_t. The reset gate and the update gate, which are controlled by different fully connected layers, are trained to dynamically trade off remembering previous information and recognizing new information [1, 2].

After obtaining the hidden states $h_t, t = 1, \ldots, N$ for the frequency dynamics for all time points in the period considered, we can use this learned information to compute, for example, the means and variances of $\Delta f(t_i), i = 0, \ldots, N-1$, by inserting a dense network between the outputs of the GRU and the outputs of the entire model. Depending on whether we take the correlation into account or not, we can then use different loss functions (cf. Section 3.1) to fit the model using a stochastic optimisation procedure such as ADAM.

3.2.2 Transformer-Structure. In contrast to GRU, we do not artificially duplicate the static feature vector to create pseudo time series as inputs, but instead try to generate different aspects of the data from the static feature vector that encode the dependencies of the individual feature values, which can then be used directly to predict the distribution parameters of the Gaussian processes Δf.

In particular, from a static feature vector of length L, we generate a sequence of L embedding vectors, which are then processed by multi-head attention as presented in [24].

Since all features in the feature vector are captured at the same time, we do not use positional encoding, unlike in typical transformer setups. The embedding vector is transformed by three different dense nets to obtain three different representations: queries, keys, and values. From queries and keys, weights are computed by computing scalar products, which weight values and give an aggregated vector of value elements. The scaled scalar products are called attention scores, and the attention computation unit is called a single head. To look at different aspects of our techno-economic features simultaneously and thus learn complex patterns and relationships more effectively, multiple heads can then be used simultaneously. The results of all the heads are combined by concatenation and reprojection to produce the final output of the multi-head attention layer. As in the original transformer in [24], we also use residual connections and layer normalization to stabilize the training process. Fig. 3 shows the entire model structure.

This aggregated information about the relationships between feature values can then be used to build another dense network to predict, for example, the mean vector or covariance matrix parameters. As with the GRU structure, different loss functions from Section 3.1 can be used here, depending on the purpose.

4 STUDY SETUP

In the following section, we present the setup of our study, in which different models based on sequence models are built and evaluated. First, we present the data used to train the models. Then we give an insight into the models created and the details of the training. Finally, we briefly present the different baseline models and the evaluation methodology.

4.1 Data Set Description, Model Input and Output

We use the same database as in [16] for reasons of comparability. For details on the data, see Appendix C. The complete dataset consists of 26 859 data points. Each data point belongs to a time interval and consists of a feature vector and a time series of the

Marginal Distribution	Serial Dependency	Network Principle
Gaussian	independent	GRU
Gaussian	independent	Transformer
Gaussian	exponential quadratic kernel	GRU
Gaussian	rational quadratic kernel	GRU
Gaussian	exponential quadratic kernel	Transformer
Gaussian	rational quadratic kernel	Transformer
t-marginal-distribution	independent	Transformer
Cauchy-margnal distribution	independent	Transformer

Table 1: Overview of the trained models and their setups.

grid frequency. The feature vector consists of values of external economic-technical variables at the beginning of the hour, values that coordinate the temporal information and the initial value of the grid frequency at the beginning of the hour. Specifically, we considered eleven day-ahead features containing price, generation and load information and their ramp values (for details see Tab. 5). With two time features and one initial value of the grid frequency, we obtain then a 14-dimensional static feature vector, resolved in seconds.

While all trained models based on the Gaussian process use the above-mentioned 14-dimensional static feature vector as input, the output of the models is different. With independent Gaussian process models, the outputs of the models are the mean values and the standard deviations of the grid frequency deviation for each point in time (e.g. every second). To limit the computational complexity of training with covariance matrices, models for Gaussian processes with correlations are trained to compute predictions for every 15th second instead of every second of an hour. The outputs of the models are the mean of the grid frequency deviation at every 15th second and the kernel parameters of the kernel matrix of all predicted time points. To generate the training data for this, every 15th element in each frequency sequence is selected. The feature data remains the same.

4.2 Training Details

The network structures are implemented as described in Section 3.2. Tab. 1 provides an overview of the models developed and their assumptions. More details on the model structures can be found in the Appendix D.

We use the negative log likelihood functions as the loss functions. To reduce the computational effort for GRU-based models, we assume that the time points can be divided into consecutive groups of time points and the dependency information of each group can be encoded by a separate latent state. Our preliminary experiments show that 180 latent states are sufficient to achieve good results. Therefore, for performance evaluation in GRU-based models, we implement 180 latent states h_i for 3600 time points, each of which decodes the dependency information of 20 time points. By subsequently applying a dense network of suitable dimension (e.g. 3600 if we can learn a mean and variance at each time point), we again obtain the outputs for each time point. For transformer-based models we always use one attention block with 4 heads. We trained models considering gaussian marginals with both the transformer and the GRU structure. The models with fat-tail marginal distributions were trained with the transformer structure. An implementation with the GRU is also possible. However, our preliminary experiments show that the transformer-based structures can be trained faster compared to the GRU structure.

All models were trained for a maximum of 100 epochs, with a batch size of 128. The validation loss was monitored during each training. In particular, we used early stopping and learning rate reduction techniques to avoid overfitting and improve model performance. Training was canceled if it did not improve over 5 epochs. The learning rate was multiplied by a factor of 0.1 if no improvement was observed after 3 epochs. This helps e.g. to fine-tune the model by taking smaller steps when a learning plateau seems to be reached. This reduction in the learning rate continues until a lower limit is reached and training is terminated by early stopping. Data from 2015 to 2018 was used for training and validation, while data from 2019 was used for evaluation.

4.3 Baseline Models and Evaluation Measures

To evaluate the performance of the models, we compare our models with other base models. We consider models that make probabilistic predictions as the Gaussian processes above do and also models that make point predictions. In addition to the day-ahead and ex-post models in [16], we also consider other data-driven models such as daily profile or constant profile, for which a Gaussian distribution with global mean and standard deviation of the frequency data between 2015 and 2018 is assumed, as in [16]. For the baseline models for the point forecast, we use all mean estimators of the probabilistic models as baseline models. We also included simple point estimators, such as the stepwise constant profile, which assumes that the frequency deviations are equal to the frequency deviation at the beginning of the hour. In addition, a simple nearest neighbour model was developed that calculates a weighted sum of the frequency sequences of the nearest feature vectors in the historical data. Tab. 8 in Appendix E provides an overview and detailed information of all the comparative models considered.

Measures such as Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are used to compare the point forecasts. We evaluate the probabilistic forecasts based on normal distributions with independent time points using negative log-likelihood and Continuous Ranked Probability Score [9]. Histograms of the realised quantiles are also created. In addition, we use energy scores [6] to compare the predictions of our model

Model	MAE	MSE	RMSE
constant zero	0.1255	0.0262	0.1543
global mean	0.1233	0.0254	0.1524
begin value	0.1772	0.0499	0.2199
mean daily profile	0.0939	0.0145	0.1190
KNN profile	0.0847	0.0116	0.1072
mean PIML day-ahead	0.0882	0.0126	0.1117
mean PIML ex-post	0.0881	0.0125	0.1110
mean indepedent gaussian GRU	**0.0814**	**0.0106**	**0.1027**
mean independent gaussian transformer	0.0821	0.0109	0.1038
model cauchy transformer	0.0828	0.0111	0.1048
model student transformer	0.0819	0.0108	0.1036

Table 2: Evaluation results of performance for point predictions.

Model	Neg. Loglikelihood	Energy Score
Independent	-236.87	2.05
Gaussian GRU with Rational Quadratic	**-352.21**	**1.91**
Gaussian Transformer with Rational Quadratic	-347.79	1.96
Gaussian GRU with Exponentiated Quadratic	-318.33	2.07
Gaussian Transformer with Exponentiated Quadratic	-312.83	2.15

Table 4: Comparison of performance for models with diagonal covariance matrix and kernel using negative log likelihood and energy score.

Model	negative log likelihood	CRPS
constant profile	-531.8881	0.0882
daily profile	-765.5269	0.0663
PIML day-ahead	-824.7810	0.0625
PIML ex-post	-825.234	0.0623
indepedent gaussian GRU	-871.4940	**0.0575**
independent gaussian transformer	**-872.9515**	0.0581

Table 3: Evaluation of performance for probabilistic predictions using median of negative log likelihood and CRPS.

with kernels. Details of the measures used can be found in the Appendix F.

5 RESULTS

In this section, we evaluate the results of the various models presented. First, we compare these models, which are based on the assumption of independent time points, with different baseline models. We then illustrate the performance of the Gaussian sequence models using kernels. We consider both the performance of the point prediction and that of the probability prediction. Finally, we examine the properties of synthetically generated data and compare them with the properties of the real frequency deviation data. In particular, we investigate whether replacing the marginal distributions with fat tails leads to better results. To present the results in a standardised way, we always compare the angular frequency deviation $\omega = 2\pi \cdot \Delta f$.

5.1 Prediction Performance

We start with the evaluation of the point forecasts using the measures MAE, MSE and RMSE. Since the models in [16] were evaluated with a 15-minute time interval to achieve their best performance, we compare the measures for a 15-minute time interval here. As Tab. 2 shows, the mean lines of all models based on sequence modeling outperform other data-based models and the two PIML models. In particular, the GRU-based Gaussian process model shows the best result for all measures.

We calculate the negative log-likelihood loss and the continuous ranked probability score to evaluate the performance of the probabilistic forecasts. Again, both our GRU-based and transformer-based models are slightly better than day-ahead and ex-post models in [16] and clearly outperform the daily profile and the constant profile (see Tab. 3). In the Appendix H, we also provide the evaluation results of the measures on one-hour intervals and histograms of the realised quantiles.

To compare specific prediction examples, we choose the same time windows as in [16] (see Fig. 4). Since the time points are independent of each other, we can use the standard deviation function to draw the enveloping lines around the mean value function. Our results when using day-ahead features are comparable to those of [16], which use day-ahead and ex-post features, for both the good cases (see Fig. 4 (a), (c)) and the bad ones (see Fig. 4 (b), (d)). This means, first, that our data-driven methods can learn the complex nature of stochastic differential equations under the assumption of independent Gaussian processes, and second, that they confirm indirectly the correctness of the choice of model equations in [16].

Taking into account the correlations between time points, the exponential quadratic kernel and rational quadratic kernel models outperform independent Gaussian processes in terms of negative log-likelihood (see Tab. 4). Again, the GRU-based models are slightly better than the transformer-based models. For example, the energy scores of the exponential quadratic kernel models are worse than the energy scores of the independent Gaussian process models. However, in terms of energy scores, the results with rational quadratic kernels are still the best. In Appendix G, we provide conditional predictions using the correlations learned for the bad cases above that could not be predicted well under the assumption of independent time points.

5.2 Synthetic Data Generation

Our models can generate realistic synthetic data from techno-economic features, e.g., for optimization or simulation purposes. For a given techno-economic vector, our models first learn the distribution parameters (depending on whether correlations are taken into account or not). Synthetic time series can then be generated from the multivariate Gaussian distribution. To illustrate this, we generated frequency data for the period January 2019 with external

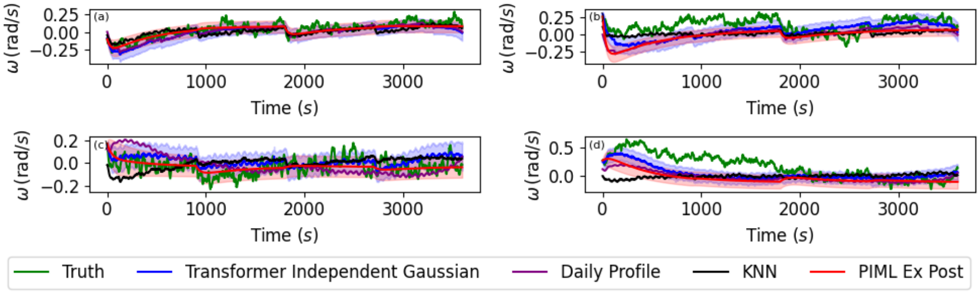


Figure 4: Good and bad prediction examples. The Transformer Independent Gaussian model shows similar performance to the PIML ex-post model. Both models can outperform the daily profile and the KNN model in good scenarios ((a) and (c)) and are similar to the daily profile in bad scenarios ((b) and (d)).

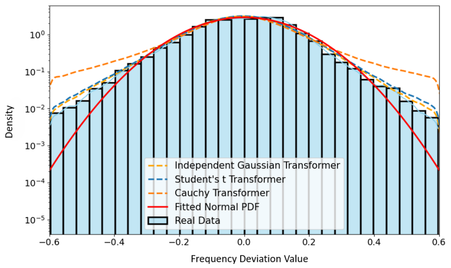


Figure 5: Comparisons of estimated distributions of real data and synthetic data generated using models based on gaussian, transformed student and cauchy marginal distributions.

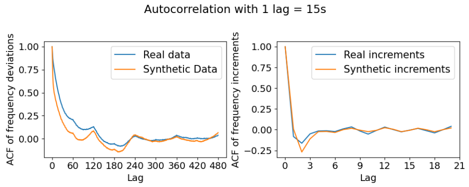


Figure 6: Comparison of the ACF for frequency deviation and its increments between real data and synthetic data, generated by a GRU-based Gaussian process model with rational quadratic kernels. Good agreement can be observed.

features and checked whether the synthetic data correspond to the typical features of frequency data. In particular, we compare the estimated probability density function (PDF) of the frequency deviation of the real and synthetic data.

The tail behaviour of frequency fluctuations is well reproduced by the synthetic time series of an independent Gaussian process with transformer structure and outperforms a simple estimate of a normal distribution. The Cauchy marginal distribution overestimates the presence of fat tails, while the Student-t marginal distribution also reproduces the tail behaviour very accurately due to its flexibility (see Fig. 5). Fig. 6 shows the auto correlation functions (ACF) of the frequency deviation and the frequency increments $\Delta\omega(t) = \Delta\omega(t + 1s) - \Delta\omega(t)$, which were calculated from the synthetic data of a Gaussian process model with transformer structure and rational quadratic kernel. An exact mapping of the autocorrelation of increments is generally a difficult task. For example, [16] is not well able to take this aspect into account, since an analytical solution of the Flokker-Planck equation requires the assumption of incorrectness of the fluctuation [16]. Here, the ACF of the frequency and the frequency increments are well represented by the synthetic data. For more details on the autocorrelations of other models with different kernels, see Appendix H.3.

6 CONCLUSION

This paper presents a fully data-driven approach to modeling and predicting the short-term evolution of frequency deviations. This fully data-driven approach is based on a combination of Gaussian processes and sequential models such as GRU and Transformer. Different models have been trained with different sequential models and kernels and evaluated with different measures for both point predictions and probabilistic predictions using negative log-likelihood, CRPS and energy scores. Although we do not accurately model the physical properties, we achieve results that are comparable to physically informed machine learning models and show slightly better results on a range of evaluation measures. Our models outperform simple data-driven models. The GRU structure performs slightly better than the transformer-based process models but transformer process models are faster and easier to train. The synthetic data of our models with Gaussian and Student's marginal distributions fulfil the typical stochastic properties of frequency data, such as fat-tail behaviour. The behaviour of the autocorrelation functions of the frequency deviation and its increment are also well represented by the synthetic data generated by a correlated Gaussian process model. Compared to simple data-driven methods such as the k-nearest neighbours algorithm, our probabilistic models presented here offer great simulation capabilities to generate realistic data sets for different scenarios or models that can be used virtually for testing purposes.

ACKNOWLEDGMENTS

Maximilian Coblenz and Bolin Liu gratefully acknowledge support from Forschungsstart project P2022-13-009 of the Carl-Zeiss-Stiftung.

REFERENCES

[1] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[3] Anirban DasGupta. 2010. *Continuous Random Variables.* Springer New York, New York, NY, 153, 165. https://doi.org/10.1007/978-1-4419-5780-1_7

[4] Peter J Diggle, Jonathan A Tawn, and Rana A Moyeed. 1998. Model-based geostatistics. *Journal of the Royal Statistical Society Series C: Applied Statistics* 47, 3 (1998), 299–350.

[5] Giovanni Filatrella, Arne Hejde Nielsen, and Niels Falsig Pedersen. 2008. Analysis of a power grid using a Kuramoto-like model. *The European Physical Journal B* 61 (2008), 485–491.

[6] Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association* 102, 477 (2007), 359–378.

[7] Leonardo Rydin Gorjão, Mehrnaz Anvari, Holger Kantz, Christian Beck, Dirk Witthaut, Marc Timme, and Benjamin Schäfer. 2020. Data-driven model of the power-grid frequency dynamics. *IEEE access* 8 (2020), 43082–43097.

[8] James Hensman, Nicolo Fusi, and Neil D Lawrence. 2013. Gaussian Processes for Big Data. In *Uncertainty in Artificial Intelligence.* Citeseer, 282.

[9] Hans Hersbach. 2000. Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting* 15, 5 (2000), 559–570.

[10] Saidul Islam, Hanae Elmekki, Ahmed Elsebai, Jamal Bentahar, Nagat Drawel, Gaith Rjoub, and Witold Pedrycz. 2023. A comprehensive survey on applications of transformers for deep learning tasks. *Expert Systems with Applications* (2023), 122666.

[11] Achin Jain, Truong Nghiem, Manfred Morari, and Rahul Mangharam. 2018. Learning and control using Gaussian processes. In *2018 ACM/IEEE 9th international conference on cyber-physical systems (ICCPS).* IEEE, 140–149.

[12] Alexander Jordan, Fabian Krüger, and Sebastian Lerch. [n. d.]. Evaluating Probabilistic Forecasts with scoringRules. ([n. d.]).

[13] Tae-Young Kim and Sung-Bae Cho. 2019. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* 182 (2019), 72–81.

[14] David Kraljic. 2022. Towards realistic statistical models of the grid frequency. *IEEE Transactions on Power Systems* 38, 1 (2022), 256–266.

[15] Johannes Kruse. 2023. johkruse/PIML-for-grid-frequency-modelling: v0.2.0. https://doi.org/10.5281/zenodo.8014136

[16] Johannes Kruse, Eike Cramer, Benjamin Schäfer, and Dirk Witthaut. 2023. Physics-Informed Machine Learning for Power Grid Frequency Modeling. *PRX energy* 2, 4 (2023), 043003.

[17] Johannes Kruse, Benjamin Schäfer, and Dirk Witthaut. 2020. Predictability of power grid frequency. *IEEE access* 8 (2020), 149435–149446.

[18] Johannes Kruse, Benjamin Schäfer, and Dirk Witthaut. 2021. Revealing drivers and risks for power grid frequency stability with explainable AI. *Patterns* 2, 11 (2021).

[19] J. Kruse, D. Witthaut, and B. Schäfer. 2021. Supplementary data: "Revealing drivers and risks for power grid frequency stability with explainable AI". https://doi.org/10.5281/zenodo.5118352

[20] Bolin Liu, Maximilian Coblenz, and Oliver Grothe. 2024. Supplementary Code for the Paper "Predicting grid frequency short-term dynamics with Gaussian processes and sequence modeling". https://github.com/bolin-liu/sequence-model-and-gaussian-process-for-frequency-prediction.

[21] Meriem Ourahou, Wiam Ayrir, B EL Hassouni, and Ali Haddi. 2020. Review on smart grid control and reliability in presence of renewable energies: Challenges and prospects. *Mathematics and computers in simulation* 167 (2020), 19–31.

[22] Carl Edward Rasmussen, Christopher KI Williams, et al. 2006. *Gaussian processes for machine learning.* Vol. 1. Springer.

[23] Benjamin Schäfer, Moritz Matthiae, Xiaozhu Zhang, Martin Rohden, Marc Timme, and Dirk Witthaut. 2017. Escape routes, weak links, and desynchronization in fluctuation-driven networks. *Physical Review E* 95, 6 (2017), 060203.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[25] Allen J Wood, Bruce F Wollenberg, and Gerald B Sheblé. 2013. *Power generation, operation, and control.* John Wiley & Sons.

[26] Yandong Yang, Shufang Li, Wenqi Li, and Meijun Qu. 2018. Power load probability density forecasting using Gaussian process quantile regression. *Applied Energy* 213 (2018), 499–509.

[27] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation* 31, 7 (2019), 1235–1270.

[28] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining.* 2114–2124.

A KERNELS

We show here four different kernels and two generated synthetic time series in each case (see Fig. 7). The amplitude and the length scale are set to one for all kernels shown. For the periodic kernel, the period parameter is set to 15.

B LOSS FUNCTIONS FOR FAT-TAIL MARGINAL DISTRIBUTIONS

The Student-t distribution with v degrees of freedom has the density

$$f(t) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{v\pi}\,\Gamma\left(\frac{v}{2}\right)}\left(1+\frac{t^2}{v}\right)^{-\frac{v+1}{2}},$$

where Γ is the gamma function. For v greater than 30, the Student-t distribution can be approximated by the normal distribution. This means with the assumption of the student-t distribution, the model is still capable of learning a Gaussian process-like process because v is learnable. In particular, We learn for each time point a location-scale Student-t-distribution $a_i \cdot \mathcal{T} + b_i$ to account for different mean positions and scattering behaviour at each time point t.

The Cauchy distribution is a special case of the Student-t distribution with $v = 1$. Note that the Cauchy distribution has neither an expected value nor a variance [3]. Therefore, for a stochastic process with marginal Cauchy distributions, we focus on learning the median and interquartile range.

For the true value $y \in \mathbb{R}^N$, The loss function using the likelihood function can then be calculated for the process with Student-t distributions

$$\begin{aligned}\mathcal{L}_{\text{t}}(y,a,b,v) = -\frac{1}{N}\sum_{i=1}^{N-1}\Big(&\log\Gamma\left(\frac{v_i+1}{2}\right) - \log\Gamma\left(\frac{v_i}{2}\right) - \frac{1}{2}\log(v_i\pi) \\ &- \mid \log(a_i) \mid - \frac{v_i+1}{2}\log\left(1+\frac{(y_i-b_i)^2}{v_i a_i^2}\right)\Big),\end{aligned}$$

and for a process with Cauchy-distributions

$$\mathcal{L}_{\text{Cauchy}(y,m_t,\gamma_t)} = \frac{1}{N}\sum_{t=0}^{N-1}\left[\ln(\pi\gamma_t) + \ln\left(1+\left(\frac{x_t-m_t}{\gamma_t}\right)^2\right)\right],$$

where the median m_t is the position parameter and γ_t is the half-width.

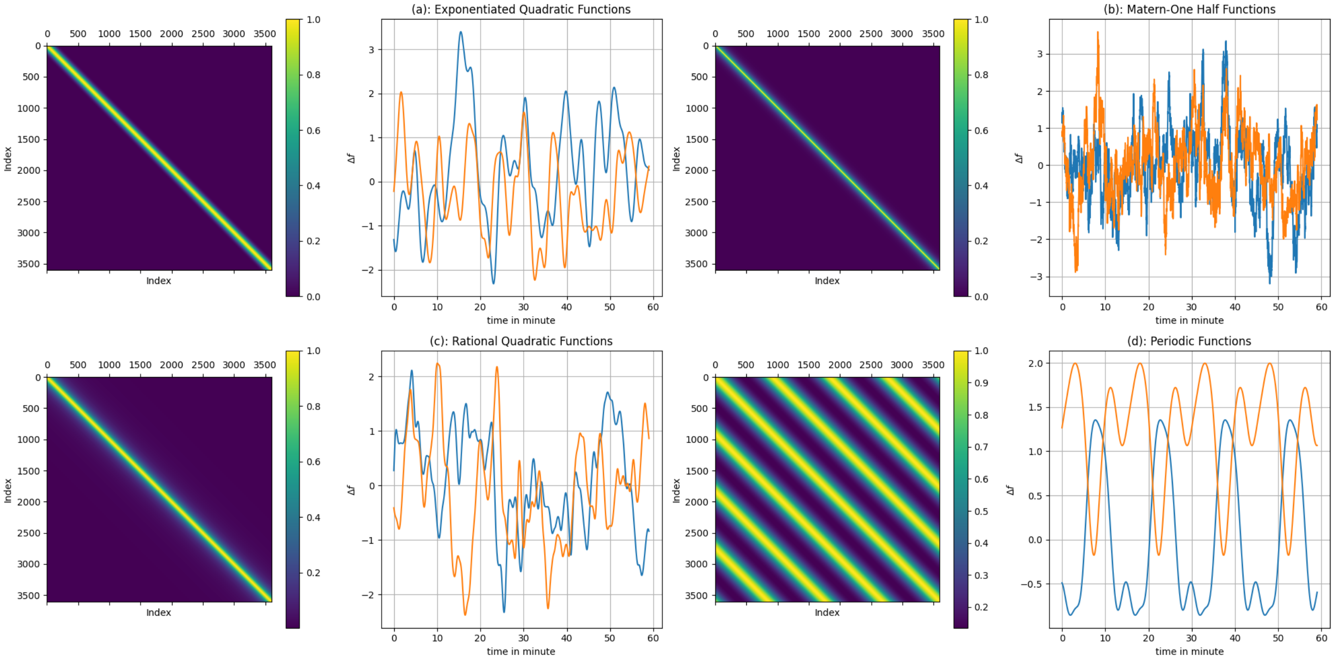


Figure 7: Kernels and two generated synthetic time series in each case.

C DATA

Type	Feature	Unit
Extern	Load day-ahead	MW
Extern	Solar day-ahead	MW
Extern	Offshore wind day-ahead	MW
Extern	Onshore wind day-ahead	MW
Extern	Load ramp day-ahead	MW/h
Extern	Generation ramp day-ahead	MW/h
Extern	Solar ramp day-ahead	MW/h
Extern	Offshore wind ramp day-ahead	MW/h
Extern	Onshore wind ramp day-ahead	
Extern	Price day-ahead	euro/MWh
Extern	Price ramp day-ahead	euro/MWh/h
Time	$cos(\pi/12hour)$	-
Time	$sin(\pi/12hour)$	-
Initial value	Grid Frequency value at the beginning of the hour to be forecast	1/s

Table 5: Overview of Features.

For the models, we consider hourly recorded external techno-economic features and the associated grid frequency data in a temporal resolution of seconds in Central Europe between 2015-2019. An overview of the features is shown in Tab. 5. In particular, we used the cleansed data in [18, 19]. The raw feature data of the cleansed data are from the ENTSO-E Transparency Platform and the raw frequency data are from TransnetBW GmbH [19]. In addition, we used the code in [15] to generate training and test data to then ensure comparability with results from the physics-informed machine learning models in [16].

D DETAILS ON MODEL STRUCTURES

Layer (type)	Output Shape
Input Layer	[(None, 14)]
Repeat Vector	(None, 180, 14)
GRU	(None, 128)
Dropout	(None, 128)
6 fully connected layers	(None, 128)
Dropout	(None, 128)
Output Layer	(None, 7200)

Table 6: Model structure of a GRU-based Mode for independnt gaussian process.

The model structure for models with gaussian process and GRU is shown in Tab. 6. The structure of the models based on transformer can be found in Tab. 7. We always use the same structures for different processes (whether gaussian or fat tail marginal distribution, or whether the time points are dependent) and only exchange the loss functions for training and learning the concrete parameters of the models (and thus the output shape). This also shows the flexibility of our approach of using sequence models. For hidden layers, we use the Relu-function as the activation function. For output layers, we usually use the linear activation function. To take into account the properties of variances or the scaling parameters of kernels, we also use the softplus function in the loss function to guarantee their positivity.

Layer (type)	Output Shape
Input Layer	[(None, 14)]
Initial Dense with 3 layers	(None, 448)
Reshape	(None, 14,32)
Three dense nets for query, key and value, each with three hidden layers of 64 neurons	3 times (None, 128)
multi head attention	(None, 14,32)
Add	(None, 14, 32)
LayerNormalization	(None, 14, 32)
Flatten	(None, 48)
6 fully connected layers	(None. 128)
Dropout	(None,128)
Output Layer	(None, 480)

Table 7: Model structure of a Transformer-based Model for a correlated gaussian process (with time step = 15 s).

E BASELINE MODELS

An overview and descriptions of the baseline models used can be found in Tab. 8. In particular, when implementing the KNN profile, we search the feature space of the training data for the features that have the smallest distance to the feature of the current time interval to be predicted and then calculate the prediction as a weighted sum of the frequency series for the given feature vectors. To optimize the hyper parameters, cross-validation is performed to determine the number of neighbours k. A KNN regressor is instantiated and trained with the optimal k value. For the implementation we use the "KNeighborsRegressor" from the "sklearn.neighbours" library. For the data generation of test data using the PIML dayahead model, PIML ex-post model, constant profile and daily profile, we use the code provided in [15]. In addition, we include all means of all probabilistic models as point predictors.

F EVALUATION MEASURES

A point predictor $\bar{y}$ of the true value y for n predictions could be evaluated with MAE (mean absolute error), MSE (mean squared error) and RMSE (root mean squared error)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \bar{y}_i|$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \bar{y}_i)^2$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}.$$

For a one dimensional probabilistic predictor $\hat{Y}$ for Y with the distribution function F, density function f and a realized value y, one could use the negative log likelihood and CRPS (Continuous Ranked Probability Score) [6, 12]

$$\log \mathcal{L}(y) = -\log f(y)$$

$$CRPS(F, y) = \mathbb{E}_F\,|Y_1 - y| - \frac{1}{2}\mathbb{E}\,|Y_1 - Y_2|\,,$$

where Y1 and Y2 are iid-distributed with distribution F.

There is an explicit formula for a normal distribution [6]

$$CRPS(N(\mu, \sigma^2), x) = \sigma\left[\frac{1}{\sqrt{\pi}} - 2\phi\left(\frac{x-\mu}{\sigma}\right) - \frac{x-\mu}{\sigma}\left(2\Phi\left(\frac{x-\mu}{\sigma}\right) - 1\right)\right],$$

where ϕ and Φ denote the probability density function and the cumulative distribution function of a standard Gaussian variable. We used the explicit formula to implement CRPS for our study. In the case of a multidimensional probalistic predictor $\hat{Y}$, the negative log likelihood could be used, too. In addition, the energy score can be used as a generalised form of CRPS to evaluate the performance of $\hat{Y}$

$$ES(F, \hat{y}) = \mathbb{E}\|Y - y\| - \frac{1}{2}\mathbb{E}\|Y - Y'\|,$$

where y and y' are independent identically distributed random variables from the distribution F, and $\|\cdot\|$ represents the Euclidean norm.

Like [12], we also use the Monte Carlo method to approximate the energy score by sampling for Y and Y' and then taking the empirical mean from the above expression.

For our study, we calculate the above measures for each data point in the test data and then use the median of the values as the final evaluation measure.

G CONDITIONAL FORECASTING

In the following, we provide conditional forecasting examples. Since the time points are now dependent, we cannot simply draw the mean function and the fill lines to calculate prediction examples. Instead, we draw the next unknown time point with realized time points using the learned correlations and draw the enveloping line using the conditional standard deviation. We repeat this process for the worst cases in Fig. 4 (b) and (d), where the independent assumptions are obviously not sufficient.

Since the entire distribution of all time points is subject to a Gaussian process, this conditional distribution $\Delta f_{i+1}|\Delta f_i, \ldots, \Delta f_0$ is also a normal distribution $\mathcal{N}(\mu_{i+1|0:i}, \sigma^2_{i+1|0:i})$ with

$$\mu_{i+1|0:i} = \mu_{i+1} + \sum_{j=0}^{i} \Sigma_{i+1,j}\Sigma^{-1}_{j,j}(x_j - \mu_j)$$

and

$$\sigma^2_{i+1|0:i} = \Sigma_{i+1,i+1} - \sum_{j=0}^{i}\sum_{k=0}^{i} \Sigma_{i+1,j}\Sigma^{-1}_{j,k}\Sigma_{k,i+1}$$

.

That is, with realized values $x_0, x_1, \ldots, x_i$, the conditional distribution of x_{i+1} can then be predicted as a Gaussian distribution $\mathcal{N}(\mu_{i+1|0:i}, \sigma^2_{i+1|0:i})$.

The prediction samples in Fig. 8 illustrate that our models, taking into account the correlations, can describe the evolution of the frequency deviation accurately.

Type	Name	Description
point	constant zero	constant frequency deviation with 0 Hz
point	global mean	global mean of the whole frequency data
point	KNN profile	prediction based on the feature distance of historical data
point	begin value profile	predictions for frequency values of a one-hour interval is the frequency value at beginning of the time interval
probabilistic	constant profile	gaussian distribution with the global mean and variance of the whole frequency data
probabilistic	daily profile	the prediction for a time of day t is equal to a normal distribution with the mean and standard deviation of all training data at this time of day
probabilistic	PIML dayahead	Physical informed Machine Learning Model using day ahead features in [16]
probabilistic	PIML ex-post	Physical informed Machine Learning Model using ex-post features in [16]
point	mean daily profile	mean of daily profile
point	mean PIML dayahead	mean of PIML dayahead model
point	mean PIML ex-post	mean of PIML ex-post model

Table 8: Overview of baseline models.

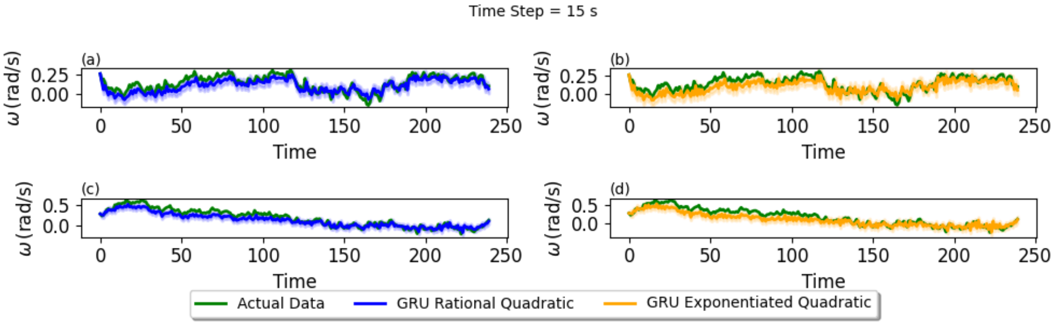


Figure 8: Conditional prediction with time step = 15s. We use here the GRU Structure. The distribution of the next point in time is predicted based on the realised values. The predictions can map the trend of the dynamic development of the frequency deviation very well.

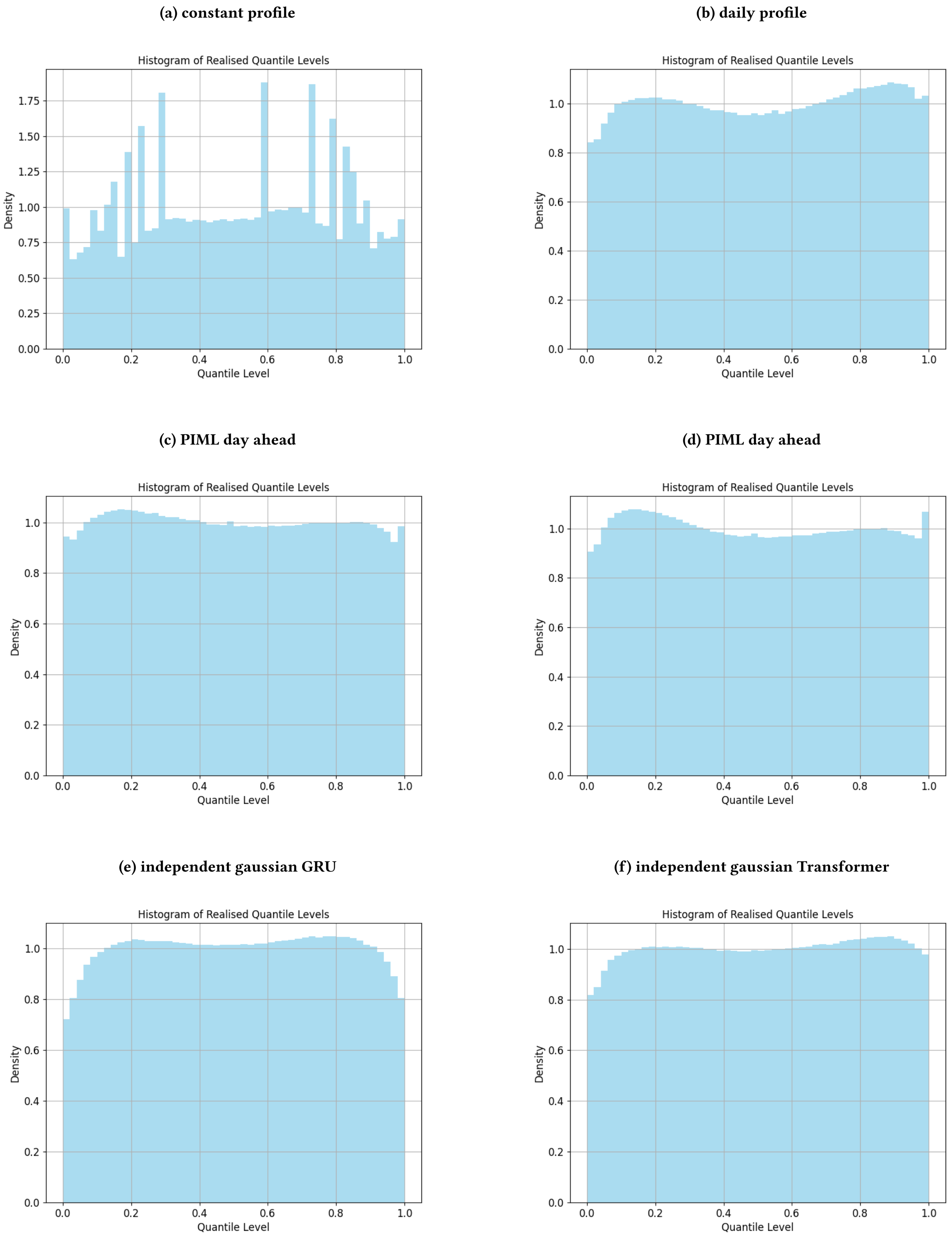
H FURTHER RESULTS ON PREDICTION PERFORMANCE AND SYNTHETIC DATA

H.1 Prediction Performance on One-hour Intervals

The models for point and probabilistic forecasts are also for hourly intervals (3600 seconds) evaluated (see Tab. 9 and 10). Again, we can observe that the sequence model based models are better than the other baseline models.

H.2 Histogram of Realised Quantiles

The Fig. 9 shows a histogram of quantile levels for each probability model based on an independent Gaussian distribution. The quantile levels are calculated from the cumulative distribution function of the normal distribution for each true value using the estimated means and standard deviations for each time point. Note that the quantile level of a true value indicates the percentage of the distribution below which that value falls. Here, we have independent time points. In an ideally calibrated model, due to the nature of the probability integral transformation, the histogram of realised quantile levels should have an approximately uniform distribution. The Fig. 9 show that all models clearly outperform the constant profile and that the calibration quality of our sequence-model-based Gaussian processes is comparable to the two PIML models.



(a) constant profile

(b) daily profile

(c) PIML day ahead

(d) PIML day ahead

(e) independent gaussian GRU

(f) independent gaussian Transformer

Figure 9: Histograms of realised quantiles for all models based on independent normal distributions.

Model	MAE	MSE	RMSE
constant zero	0.100163	0.016221	0.123233
global mean	0.100212	0.016226	0.123266
begin value	0.151541	0.036620	0.189428
mean daily profile	0.081699	0.010686	0.102458
KNN profile	0.078300	0.009728	0.098325
mean PIML day-ahead	0.079684	0.010182	0.100159
mean PIML ex-post	0.080332	0.010224	0.100493
mean indepedent gaussian GRU	0.075965	0.009120	0.095231
mean independent gaussian transformer	0.076791	0.009349	0.096402
model cauchy transformer	0.077286	0.009513	0.097261
model student transformer	0.076565	0.009304	0.096163

Table 9: Evaluation results of performance for point predictions for one-hour intervals.

Model	negative log likelihood	CRPS
constant profile	-2659.6437	0.0708
daily profile	-3411.7385	0.0575
PIML day-ahead	-3440.4902	0.0563
PIML ex-post	-3417.1639	0.0567
indepedent gaussian GRU	-3575.8352	0.0536
independent gaussian transformer	-3549.5811	0.0542

Table 10: Evaluation of performance for probabilistic predictions for one-hour intervals.

H.3 Autocorrelation of Synthetic Data

Here, we show the ACFs of the synthetically generated data using different models (see Fig. 10, 11 and 12). While good agreement can generally be observed for both variables for all models, there is a clear difference in the acf value of the increments at 2 lags (30s) between the models with exponentiated and rational quadratic kernels.

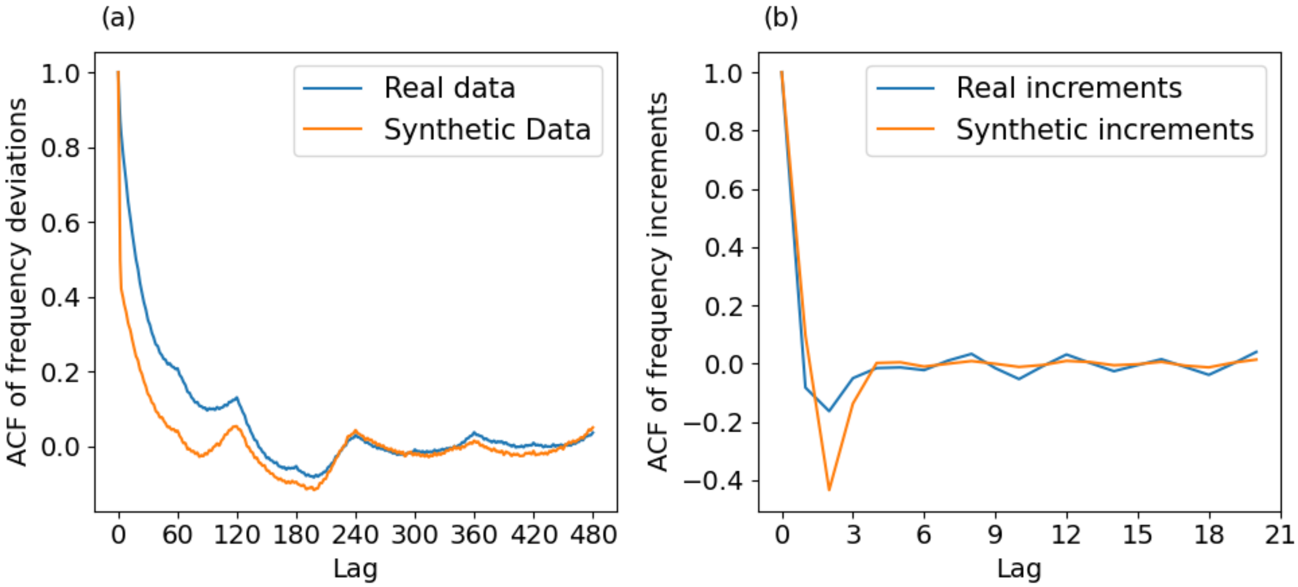


Figure 10: Auto correlation functions for frequency deviation and its increments generated by a GRU Gaussian process model with exponentiated quadratic kernel.

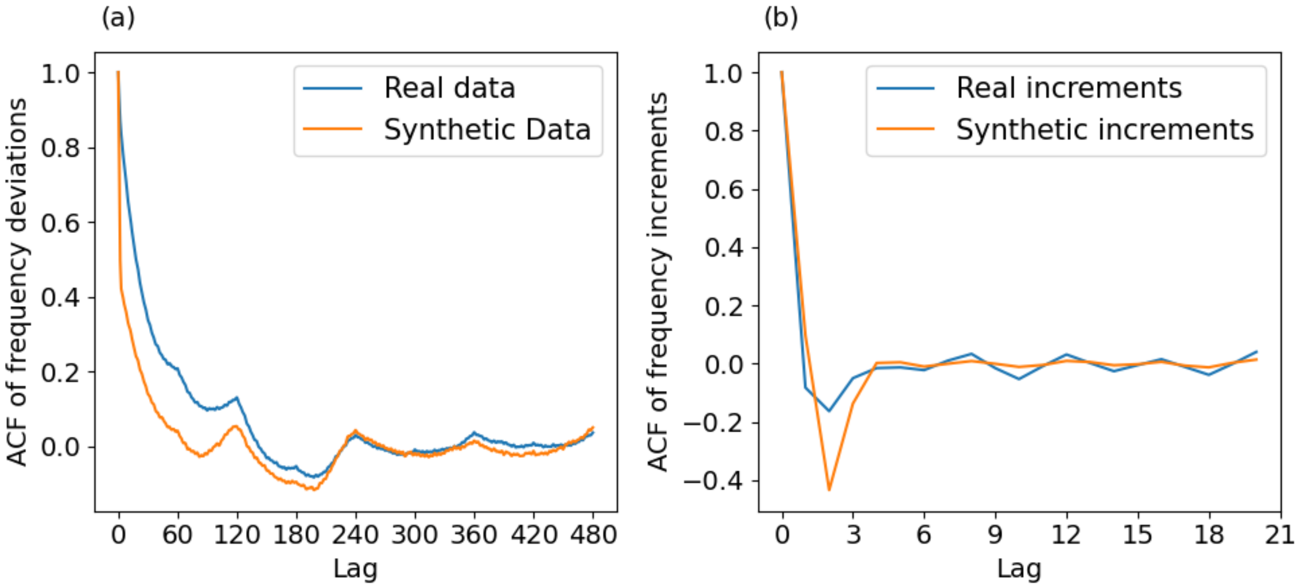


Figure 11: Auto correlation functions for frequency deviation and its increments generated by a transformer Gaussian process model with exponentiated quadratic kernel.

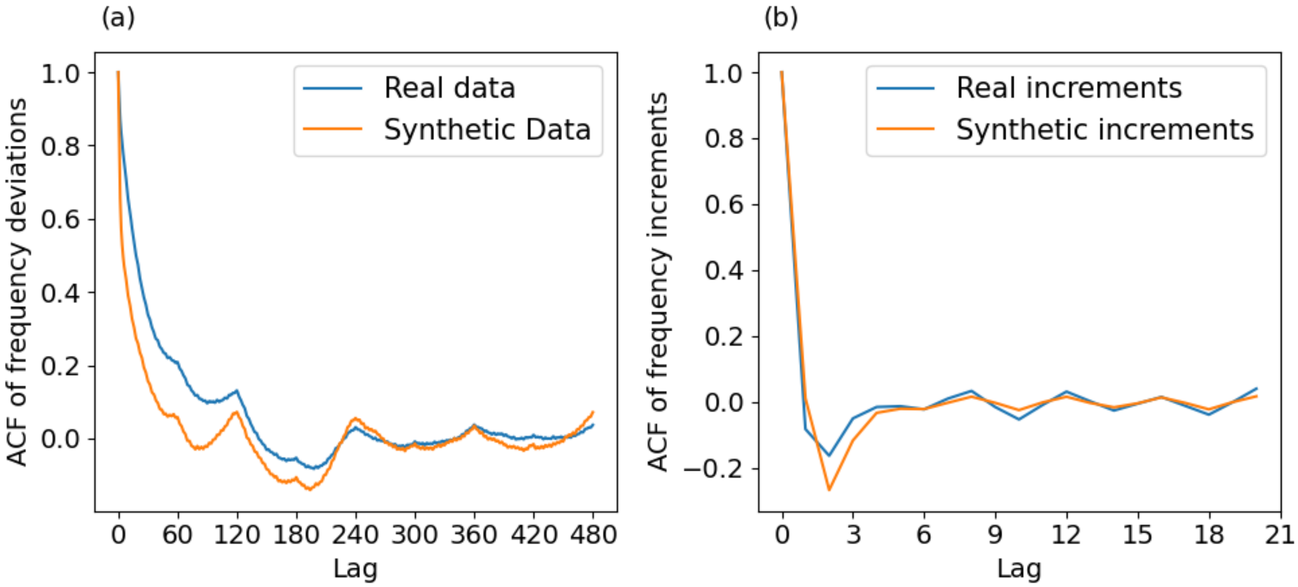


Figure 12: Auto correlation functions for frequency deviation and its increments generated by a transformer Gaussian process model with rational quadratic kernel.