AI

MDPI

Article

Machine Learning Models Informed by Connected Mixture Components for Short- and Medium-Term Time Series Forecasting

Andrey K. Gorshenin [1,*] and Anton L. Vilyaev [1,2]

[1] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, 119333 Moscow, Russia
[2] Moscow Center for Fundamental and Applied Mathematics, Lomonosov Moscow State University, 119991 Moscow, Russia
* Correspondence: agorshenin@frccsc.ru

Abstract: This paper presents a new approach in the field of probability-informed machine learning (ML). It implies improving the results of ML algorithms and neural networks (NNs) by using probability models as a source of additional features in situations where it is impossible to increase the training datasets for various reasons. We introduce connected mixture components as a source of additional information that can be extracted from a mathematical model. These components are formed using probability mixture models and a special algorithm for merging parameters in the sliding window mode. This approach has been proven effective when applied to real-world time series data for short- and medium-term forecasting. In all cases, the models informed by the connected mixture components showed better results than those that did not use them, although different informed models may be effective for various datasets. The fundamental novelty of the research lies both in a new mathematical approach to informing ML models and in the demonstrated increase in forecasting accuracy in various applications. For geophysical spatiotemporal data, the decrease in Root Mean Square Error (RMSE) was up to 27.7%, and the reduction in Mean Absolute Percentage Error (MAPE) was up to 45.7% compared with ML models without probability informing. The best metrics values were obtained by an informed ensemble architecture that fuses the results of a Long Short-Term Memory (LSTM) network and a transformer. The Mean Squared Error (MSE) for the electricity transformer oil temperature from the ETDataset had improved by up to 10.0% compared with vanilla methods. The best MSE value was obtained by informed random forest. The introduced probability-informed approach allows us to outperform the results of both transformer NN architectures and classical statistical and machine learning methods.

Keywords: probability-informed machine learning; finite normal mixtures; connected mixture components; forecasting; feature engineering

Citation: Gorshenin, A.K.; Vilyaev, A.L. Machine Learning Models Informed by Connected Mixture Components for Short- and Medium-Term Time Series Forecasting. *AI* **2024**, *5*, 1955–1976. https://doi.org/10.3390/ai5040097

Academic Editor: Mehdi Neshat

Received: 25 August 2024
Revised: 13 October 2024
Accepted: 17 October 2024
Published: 22 October 2024

1. Introduction

In practice, we often deal with incomplete and noisy data that is influenced by various random factors. To correctly describe patterns in such cases, the methods and approaches of probability theory, mathematical statistics, and random processes are traditionally used in mathematical modeling. Developing effective methods and algorithms for data analysis necessitates the creation of mathematical models that depict the complex systems as well as the statistical patterns of various processes within them. This task is relevant in the field of artificial intelligence (AI) [1]. For example, a few traditional solutions, such as Bayesian methods [2] and statistical learning [3,4], can be mentioned.

Training deep neural network (NN) architectures [5,6], including those for solving forecasting problems [7–10], requires a large amount of data or additional information about the structure of the data being investigated. Real-world datasets often have limited

volume [11,12] and contain errors and noise due to random factors. This can make the results of machine learning (ML) methods and neural networks worse. In particular, it is not always possible to use efficient, but computationally complex and demanding data and features, NN architectures like transformers [13] or Mamba [14]. To account for the influence of random factors and describe the heterogeneity and variability of the data, it is natural to use probabilistic and statistical models. This paper develops an approach to feature construction based on probability mixture models to improve NN and ML forecasting accuracy in time series. These models can improve the performance of algorithms of this type.

In machine learning, the term "informing" refers to the joint use of mathematical and ML models. It can imply feature construction based on various mathematical models. We develop an approach to probability informing that uses probability models and their characteristics as sources of additional information (features) for machine learning/neural network algorithms. Moreover, even within the framework of these models, parameters can be random. This allows us to flexibly account for the influence of external factors and improve the generalization ability of NN/ML models. The so-called physically-informed ML [15] should be mentioned in this context. Within its framework, ML models receive additional information based on the physical model of a process or phenomenon. However, in some cases, there is no physical model, but it is possible to create statistical approximations for the data. In this regard, our paper expands on the aforementioned approach by considering the influence of random factors on real data for various tasks. This leads to the development of probability-informed machine learning.

The fundamental novelty of our approach lies in two aspects. First, we suggest a new mathematical approach to informing ML models. Second, we demonstrate that models informed by so-called connected mixture components are significantly more accurate for short- and medium-term forecasting tasks on various real-time series. The basic mathematical model is mixtures of probability distributions. Their appearance can be attributed to the ability to describe the processes discussed in the paper using a specific type of stochastic differential equation. It is worth noting that the mixture parameters can also be random, which allows for a more flexible consideration of the impact of external factors. Therefore, it is necessary to provide both an analytical explanation for the type of distribution family and to estimate random parameters, which leads to semiparametric [16] statistical models.

The main contributions are as follows:

- A new method of probability informing of ML models is introduced. It involves creating additional features (connected mixture components) based on probability and stochastic models. This approach is suitable for various machine learning algorithms and deep neural networks.
- For the first time, we have shown that probability-informed ML models can also improve their accuracy in forecasting, not just neural networks like in previous research. When applying these models to geophysical time series, the Root Mean Square Error (RMSE) was reduced by 1.2–16.4% and the Mean Absolute Percentage Error (MAPE) was decreased by 3.2–24.3% compared with the results obtained with vanilla decision trees [17], random forests [18], and gradient boosting [19].
- For real-world time series, a significant increase in the ML and NN forecasting accuracy is demonstrated with various methods of probability informing by connected mixture components as well as forecast periods. Thus, for geophysical data, the RMSE was decreased by 0.8–27.7% and MAPE was by 0.1–45.7%, compared with models without informing. For Electricity Transformer Dataset (https://github.com/zhouhaoyi/ETDataset (accessed on 1 July 2024)) (ETDataset) [20], the Mean Squared Error (MSE) improvement was 1.2–10.0%.
- For any test datasets and algorithms, probability-informed models are better than vanilla ones. However, the best accuracy can be obtained by various algorithms. An informed ensemble of LSTM (Long Short-Term Memory) architecture [21] and vanilla transformer [22] provides the best results for geophysical data in short- and

medium-term forecasting. Alternatively, for medium-term forecasts, an informed random forest should be used for ETDataset.
- The introduced probability-informed approach allows us to outperform the results of both transformer NN architectures and classical statistical and machine learning methods.

The rest of the paper is organized as follows: Section 2 discusses known approaches to constructing features and informed machine learning. In Section 3, a methodology for constructing connected mixture components as well as corresponding probability-informing techniques are proposed. Section 4 shows the results of forecasting with different periods for real geophysical and electricity transformer time series. Section 5 briefly discusses the results obtained and further research directions in this area.

2. Related Works

Feature engineering is an essential tool in the field of machine learning. It allows us to obtain additional information from existing data without increasing its size. It can be used in classification [23], clustering, speech recognition, detection, and many others [24,25].

The modern solutions for time series forecasting are still often based on LSTM and transformer NN architectures [26,27]. To improve forecasting accuracy under limited dataset restrictions, additional information such as meta-data or multi-modal features can be used [28,29]. However, in real-world situations, time series often lack additional features, especially in experimental or historical datasets. In such cases, feature engineering approaches can be used.

The process of feature construction can be based on various transformations of raw data, such as algebraic operations. Additionally, it can involve extracting valuable information based on physical laws as well as mathematical models and approaches. This approach is promising in the field of physics-informed machine learning models [15]. It has already been successfully used in neural networks and deep learning models that are used to predict various physical processes [30–33] as well as time series analysis, such as predicting the movements of semi-submersible vehicles [34] taking into account wave fluctuations.

The physics-informed machine learning can use probabilistic, statistical, and stochastic models [35–39]. Moreover, the probabilistic characteristics of the data can be integrated into a specific physical model. For example, it can be helpful to create a model for errors that occur due to random noise in a dynamic system [40]. Another approach is to extend information using a physical model for probabilistic neural networks, such as diffusion one [41] or based on Bayesian inference [42].

Our research aims to develop an approach for informing based on the normal (Gaussian) mixture models. Previously, we have proposed the idea of using the first four moments of finite normal mixtures: expectation, variance, skewness, and kurtosis [43]. By adding only a small number of elements (features) to the inputs, we were able to improve the accuracy of short- and medium-term LSTM forecasts by an average of 11.4% for experimental data related to turbulent plasma and air-sea heat fluxes. A more complex method for creating features for recurrent neural networks (RNNs) was also introduced.It was used to select a trading strategy by an automatic trading system, which allowed us to get a yield increase of up to 23.3% for currency pairs, but, in some cases, the classic ML metrics used for forecasting (RMSE) were too large. For instance, it reached 0.3 on a few pairs.

In this paper, we propose a variety of methods for probability informing with additional features, based on connected mixture components both in a machine learning model and in neural networks focused on improving the quality of classical forecasting metrics RMSE, MAPE, and MSE.

3. Methodology of Probability Informing Based on Connected Mixture Components

This section describes approaches to feature construction based on probability models for data, as well as merging initial inputs and additional features for ML algorithms and deep neural network architectures frequently used in various data forecasting problems.

3.1. Overall Framework

We consider the forecasting problem, that is, constructing a model f that best predicts the target variable Y based on a set of features X such that $f : X \to Y$. Methods for expanding the feature space X by forming new nontrivial features X^+ are focused on constructing a modified model $f^+ : (X \cup X^+) \to Y$ such that it is the optimal according to a certain criterion (i.e., accuracy metrics), while the modified f^+ model differs from the original f model only by inputs extended with the new features while maintaining its basic structure.

The general scheme (see Figure 1) begins with taking time series data as input for further processing. In order to enhance the forecasting capacity of our model, we extract and construct additional features from the input data using probability mixture models. These features are then used to inform the forecast model, increasing its ability to recognize complex relationships in the data. Finally, the model forecasts the target variable based on the informed model. The feature construction and connected mixture components are discussed in Section 3.2, whereas our probability-informed approach to ML models is introduced in Section 3.3.

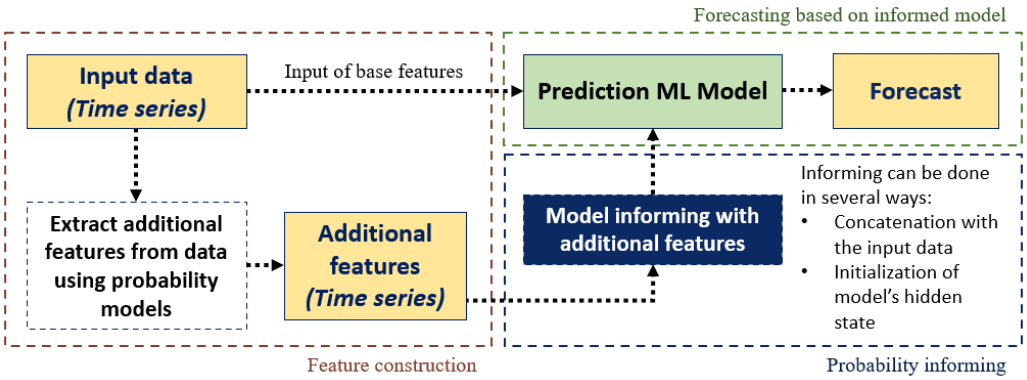

Figure 1. Scheme of probability feature construction and incorporation into machine learning models.

3.2. Feature Construction Based on Finite Normal Mixtures

Let us consider the procedure for constructing additional features, namely, connected mixture components, based on probabilistic and stochastic models. Indeed, many physical, financial [44], and other processes can be described using Itô stochastic differential equations [45] (SDEs) of the following form:

$$dX(t) = a(t)dt + b(t)dW(t), \tag{1}$$

where $X(t)$ is the stochastic process under consideration, $W(t)$ is the standard Wiener process, and the coefficients $a(t)$ (drift) and $b(t)$ (diffusion) are some random functions.

It is well-known [46] that, in the case of non-random drift and diffusion coefficients, under additional assumptions about measurability with respect to filtering and normality of the initial value distribution, the solution of SDE (1) is a certain Gaussian process with a given mean and covariance function. In such a situation, the increments of the process would also be Gaussian random variables. However, for random coefficients, the distributions take on the form of arbitrary location-scale normal mixtures. An arbitrary normal mixture can be approximated by finite ones. The estimation of unknown parameters can be based on various modifications of the EM algorithm [47–49]. The parameters of the approximating mixtures contain additional valuable information about the data structure derived from the mathematical model used for them [50]. Therefore, it is natural to consider them as additional features for ML methods and NN architectures. It is worth noting that the parameters of the family of probability distributions used to approximate observations can change significantly over time since real processes are usually not stationary. Therefore, at the approximation stage, the model is constructed not for the entire initial time series

but for some of its parts, windows, on which the process can be considered stationary in a certain sense.

Let us describe in more detail the methodology for constructing new features for data that can be described by SDE (1). Consider a subsample (window number t), which is a vector X with the following distribution function:

$$F(x, k(t), \boldsymbol{a}_t, \boldsymbol{\sigma}_t, \mathbf{p}_t) = \sum_{i=1}^{k(t)} p_i \Phi\left(\frac{x - a_i(t)}{\sigma_i(t)}\right), \tag{2}$$

where $x \in \mathbb{R}$, $\Phi(x) = \int_{-\infty}^{+\infty} e^{-x^2/2}\, dx$ and for all $j = 1, \ldots, k(t)$ standard conditions for parameters hold:

$$a_j(t) \in \mathbb{R}, \quad \sigma_j(t) \in \mathbb{R}, \quad \sigma_j(t) > 0, \quad \sum_{j=1}^{k(t)} p_i(t) = 1, \quad p_j(t) \geqslant 0.$$

Based on the estimation of the distribution parameters (2), there are the following non-trivial features, which contain information about the statistical behavior of the original series:

- $p_j(t)$ are weights of the corresponding components;
- $a_j(t)$ are expectations;
- $\sigma_i(t)$ are standard deviations.

It follows from the expression (2) that $3k(t) - 1$ new features are formed on each window with the number t (one weight value $p_j(t)$ can always be expressed in terms of the remaining ones). Then the window moves to the next time step, and the procedure repeats.

The process of evaluating mixture component parameters is based on the EM algorithm. On the one hand, it is a traditional method for estimating the parameters of mixture models [51–54]. On the other hand, this method is closely related to neural networks both in the framework of a well-known relationship [55] with the backpropagation, a traditional method of NN training, and in the form of implementation in various NNs [56].

Let us present a methodology for connecting parameters obtained on steps of the sliding window. On the first pass, an adjacency matrix $(a_j(t), \sigma_j(t))$ is constructed for consecutive steps t and $t+1$. Based on metrics of ℓ_p [57] spaces (for example, $p = 1, 2$), one can evaluate whether the parametric pair $(a_j(t+1), \sigma_j(t+1))$ at the $t+1$ step is a continuation of the pair $(a_j(t), \sigma_j(t))$ at the t-th step.

The estimates of the parameters $(a_j(t+1), \sigma_j(t+1))$ can vary significantly from those on the previous step, even if the total number of components $k(t+1)$ remains the same. In this case, a new component is formed that is not associated with any values in the previous step. It is worth noting that the parameters of the weights $p_j(t)$ are not taken into account. Indeed, the impact (i.e., weight) of the component on the overall finite normal mixture can change significantly at each step, but the expectation and variance do not vary much. In this case, it is considered to be the same component. One of the dimensions of the adjacency matrix corresponds to the number of steps, and the second, with the exception of completely zero vectors that occur during its initial initialization in the framework of software solutions, corresponds to the connected mixture components. At the second step in the two-dimensional space (a, σ), some clustering method is used with the number of components obtained in the previous step. This procedure finally forms connected mixture components. In general, their number does not coincide with the number of summands in the formula (2) used for approximation at each step.

The procedure presented below (see also Algorithm 1) forms connected mixture components:

1. Let $I^{(t)}$ be a set of indices (numbers) of components for step number t, that is, $I^{(t)} = \{1, 2, ..., k^{(t)}\}$, and $J^{(t+1)} = \{1, 2, ..., k^{(t+1)}\}$ is an analogous set for step. $t + 1$.
2. Let I_0 and J_0 be sets of indexes from the first and second sets, respectively, for which the nearest component was found. Initially, we assume $I_0 = \emptyset$, $J_0 = \emptyset$.
3. For each $J \in J^{(t+1)} \setminus J_0$, one should find the closest number I in the sense of solving an optimization problem:

$$I = \underset{i \in I^{(t)} \setminus I_0}{\operatorname{argmin}} \left(|a_i^{(t)} - a_J^{(t)}|^p + |\sigma_i^{(t)} - \sigma_J^{(t+1)}|^p \right)^{1/p}.$$

4. To correctly identify connected components, the following condition must be met:

$$\left(|a_I^{(t)} - a_J^{(t+1)}|^p + |\sigma_I^{(t)} - \sigma_J^{(t+1)}|^p \right)^{1/p} < \epsilon(\mathbf{a}, \sigma). \tag{3}$$

5. Steps 1–4 are repeated for each acceptable position of the sliding window, forming a parameter adjacency matrix.

Algorithm 1. Forming connected mixture components

```
function COUPLED_COMPONENTS(Data, options)
    // Estimation of mixture parameters
    Params ←EMs(Data, options.EM)
    // Initialization by the number of components chosen for all window positions
    Comps^(1) ← Params.k^(1);
    for n=t:LENGTH(Params)-1 do
        I_0 ← ∅,    J_0 ← ∅;        //Initialization of sets
        repeat
            // New or connected component
            [I, J] ←FIND_INDEX(Params,J^(t+1) \ J_0, I^(t) \ I_0);
            if I ≠ ∅ then        // The previous component for J is founded
                I_0 ← I_0 ⋃ I,    J_0 ← J_0 ⋃ J;
            else
                // Adding a new component
                J_0 ← J_0 ⋃ J;
                Comps^(n+1) ← ADD_NEW_COMP(Params, J);
        until (J^(t+1) \ J_0 ≠ ∅)
    // Labels for each set of parameters, clustering
    Labels ← CLUSTERING(Params, MAX_COMPS, options.ClustAlg);
    // Adjacency matrix
    AdjMatrix ← CONNECT(Params, Labels) :
    return AdjMatrix;
```

3.3. Probability-Informed Machine Learning Models

In this section, we consider approaches to merging inputs (i.e., data) with connected mixture components for ML models and NNs. Let the vector of estimated parameters of the mixture model be denoted as

$$\vec{c}(t) = (a_1(t), \cdots, a_N(t), \sigma_1(t), \cdots, \sigma_N(t), p_1(t), \cdots, p_{N-1}(t)). \tag{4}$$

The first possible approach involves directly combining the inputs with this vector. For each set of basic features associated with the position of the sliding window having an index of t, we add $3N - 1$ new features (separate parameters) to the input, which are based on probability models. It is important to note that the number N, i.e., the dimensionality of the vector $\vec{c}_t$, can vary across different windows. This depends on

the optimal number of components needed for approximating with the finite normal mixture (2). For easier data transfer to the model, we suggest using a fixed number N of components for all windows. This means that the set number of components will remain the same for each window.

The second approach is based on the assumption that the distribution in adjacent windows does not change significantly. This is because the windows differ by only one observation. Consequently, the time-varying parameters of the components $\vec{c}(t)$ can be combined into a single multidimensional time series. Figure 2 demonstrates the differences between these approaches. The extension of the feature space is based on blocks marked with green.

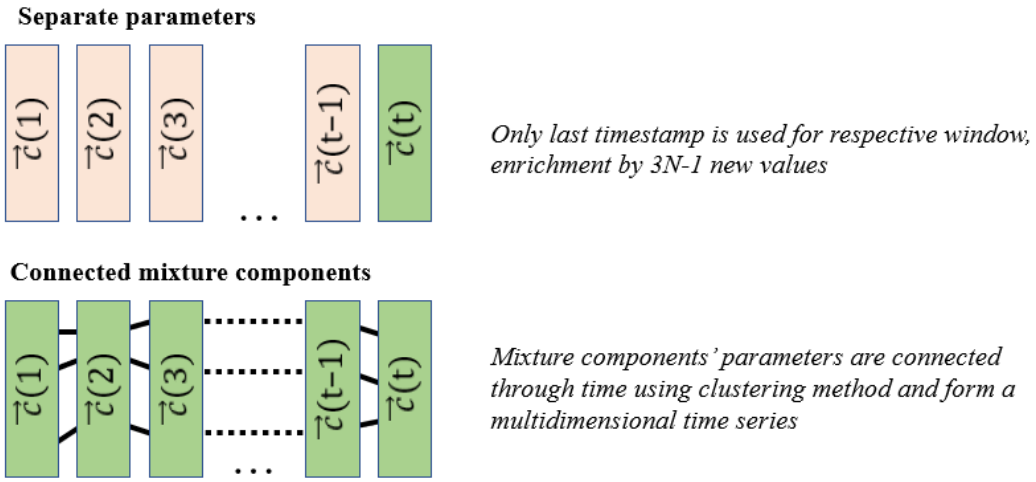

Figure 2. Separate parameters versus multidimensional time series.

Let us consider a few probability-informed machine learning methods. In the tasks under consideration, the use of decision trees (hereinafter referred to as model (I)) and their ensembles, random forests (model (II)), as well as gradient boosting on them (model (III)), holds significant interest, particularly due to their interpretability and efficiency. In models (I)–(III), the addition of connected mixture components can be implemented using the first method described in this section, namely, by adding separate parameters for each corresponding window. The additional variables are transferred to the forecasting model in the standard way, as shown in Figure 3.

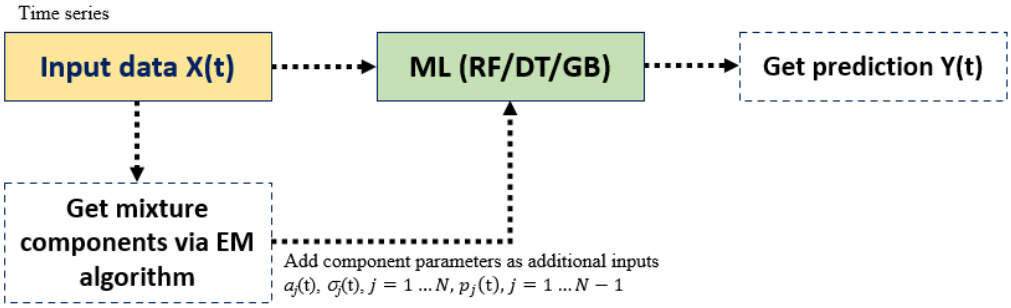

Figure 3. Informing ML algorithms (I)–(III) with parameters of probability models.

In our previous studies [44,58], we successfully tested the first approach (separate parameters) to informing LSTM architectures. This architecture remains basic for forecasting time series of various natures. In this paper, we improve the probability informing of neural network models, including for the ensemble architecture based on LSTM and transformer.

The neural network model (IV) (see Figure 4) consists of an LSTM layer, a dropout layer, and fully connected (FC) layers. The parameters of the connected mixture components can

be added as separate additional features for each corresponding window using the method proposed by Andrej Karpathy [59]:

- The additional features $\vec{c}$ are transformed using an affine transformation to a form that corresponds to the internal state of the recurrent neural network (RNN):

$$\vec{s} = W\vec{c} + \vec{b}, \tag{5}$$

  where W and $\vec{b}$ are trainable parameters of the model.
- Then, the hidden state of the RNN is initialized using the vector $\vec{s}$.

This approach allows additional features to be transferred into the model without affecting the input data of the LSTM layer, which represents a unified time series.

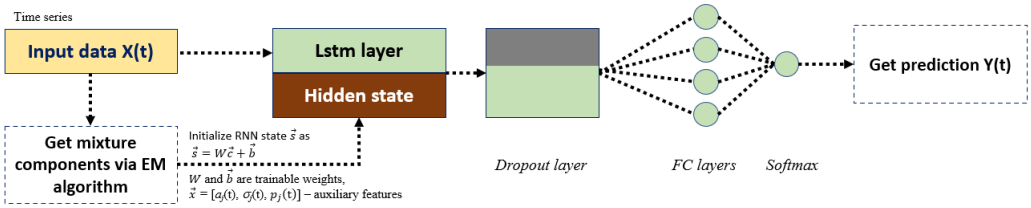

Figure 4. LSTM architecture (IV) informed by connected mixture components using hidden state initialization.

Probability informing of this architecture can also be organized in a different way, using the second method described at the beginning of this section. The model (V) consists of an LSTM layer, a dropout layer, and FC layers. Now, the statistical parameters are connected over time and transferred as a multivariate time series, merging with the original time series; see Figure 5.

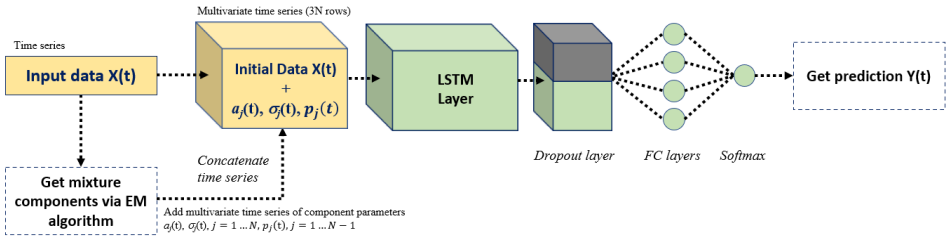

Figure 5. LSTM architecture (V) informed by multidimensional connected mixture components.

Finally, the model (VI) is an ensemble of LSTM and a vanilla transformer [22], see Figure 6. In this model, similar to model (V), the connected mixture components are used as a multivariate time series. It is fed in parallel to the input of the LSTM model with a dropout layer at the output and to the transformer block, consisting of several layers (the number of layers is a hyperparameter). Each transformer layer consists of attention heads, dropout, and Layernorm layers, as well as FC layers. Then, a feature fusion [60–63] is used: the predictions of both models are combined into a single vector, which after several FC layers leads to the desired forecast value.

The idea of combining the LSTM and transformer in one model is based on the following reasoning. It is well known that attention effectively models long-term dependencies, focusing on different parts of the sequences. Meanwhile, the LSTM memory captures short- and medium-term dependencies well but may forget important observations over time. Thus, ensemble allows us to consider all these dependencies, which could potentially enhance the overall generalization ability of the architecture.

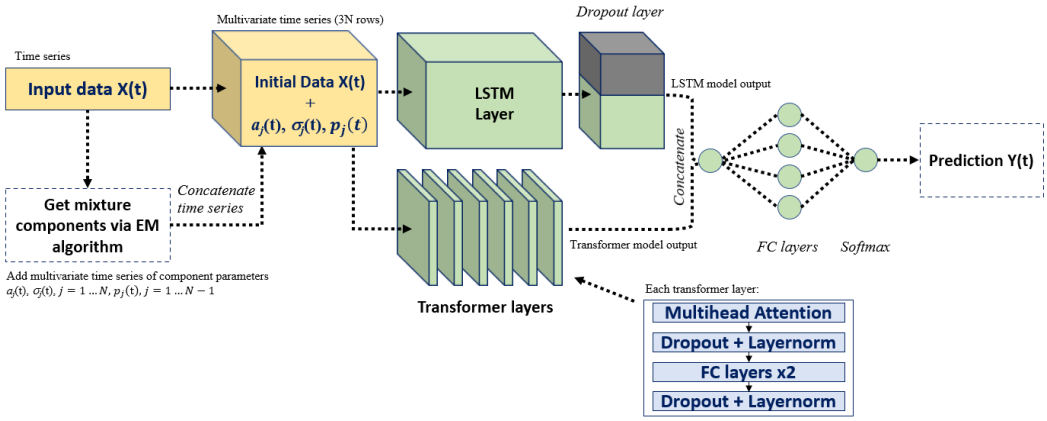

Figure 6. Ensemble of LSTM architecture and transformer (VI), informed by multidimensional connected mixture components.

4. Experimental Section

4.1. Test Data and Connected Mixture Components

The first type of experimental data is air-sea heat fluxes, which are measured at three geographic locations (the Gulf Stream, the Labrador Sea, and the Tropics). For each of them, values of so-called latent and sensible fluxes are available. These are 6 time series, each containing 14,612 observations from 2000 to 2010 (with a 6-h observation interval). For convenience, the following names are used for them further: Gulfstream-1, Gulfstream-2, Labrador-1, Labrador-2, Tropical-1, Tropical-2. The postfix 1 corresponds to latent fluxes $Q_e = L\rho C_e(q_s - q)V$, and 2 is sensible fluxes $Q_h = c_p\rho C_T(T_w - T_a)V$. Here, T_w and T_a are the water and air temperatures, respectively, V is the wind speed magnitude, q is the specific humidity of the near-surface air, q_s is the saturated specific humidity above the water surface, L is the latent heat of evaporation, c_p is the specific heat of air at constant pressure, and its density ρ, C_T, and C_e are the Stanton and Dalton numbers [64]. The correctness of modeling such data using SDE (1) was demonstrated, for example, see [65]. For the purpose of testing our approaches, researchers can also use reanalysis data from the ERA5 [66] or the RAS-NAAD [67] databases.

Figures 7 and 8 (both on the left) demonstrate fairly expected seasonal patterns in the geophysical data. Therefore, for the correctness of statistical methods, the increments of the time series should be used for modeling (see Figures 7 and 8, both on the right). A brief description of the characteristics of all considered geophysical time series is presented in Table 1.

The target variable is a corresponding flux value. The training set consists of about 12,000 observations from the first 8 years, and the test set is about 2000 observations from the last 16 months. Each original time series was divided into windows of 200 observations. For each window, the empirical distribution function was approximated by finite normal mixtures in a sliding window mode as described above in Section 3.2. A basic four-component finite normal mixture was used for the approximation, and the following number of structural components was identified for each series: 5 for the Labrador-1 and Tropical-2 series and 6 for the Gulfstream-1, Gulfstream-2, Labrador-2, and Tropical-1 series. Figure 9 shows an example of the connected mixture components for the Gulfstream-2.

Figure 9 presents the expectations of the connected mixture components with in-time evolution of the four of them marked in blue, orange, green, and red. A part of the time series of 150 values was taken for clarity. Figure 10 demonstrates the formation of a new component marked in purple with the disappearance of one of the previous components marked in green from observation with number 190 and further.

One more test dataset (see Figure 11), which is significantly different in physical nature from the geophysical time series, the Electricity transformer Dataset (ETDataset) [20] was used. It contains measurements of transformer oil temperature and transformer load readings in a region of China. A total of 17420 observations were collected from 2017 to 2018 (with a one-hour observation interval). A negative load refers to a situation where power flows in

the reverse direction through a transformer, i.e., from the secondary winding to the primary one [68,69]. The possibility of modeling electricity transformers using SDEs [70], as well as several more studies in this field [70–74], can be mentioned. Therefore, the approaches proposed in Section 3 can also be used to form an extended feature space in this case. Moreover, there are studies that use deep learning methods to forecast this type of data. In addition to the article [20], which introduces and analyzes ETDataset, there are other relevant papers, for example, see [75–78], in which various deep learning methods are used.

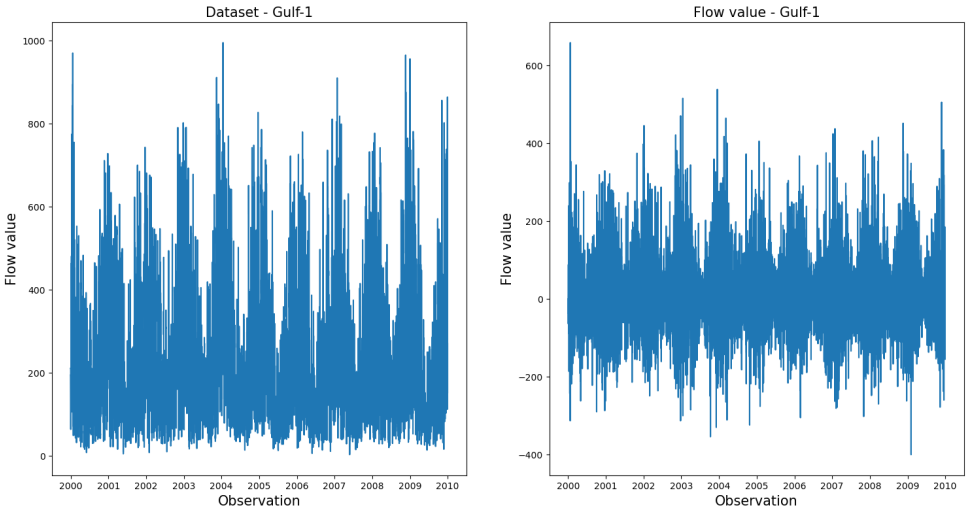

Figure 7. Gulfstream-1 (**left**) and its increments (**right**).

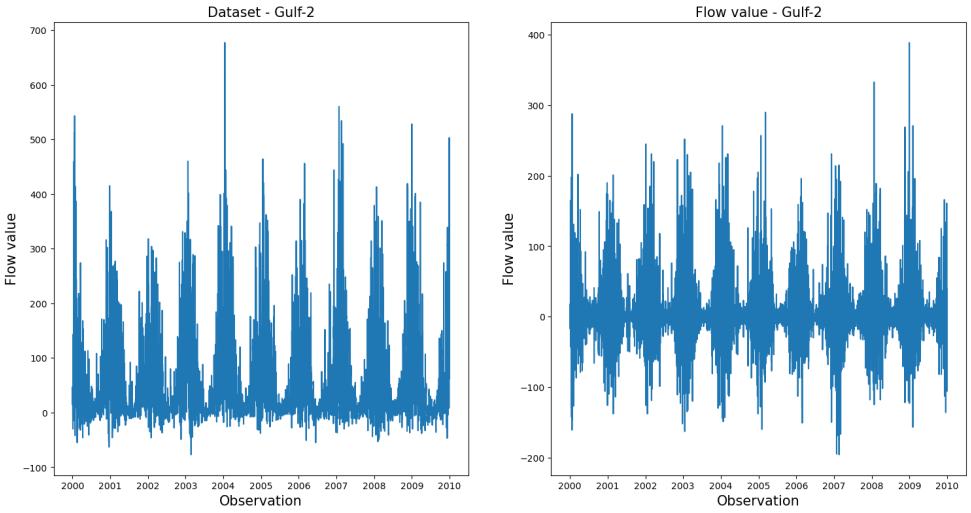

Figure 8. Gulfstream-2 (**left**) and its increments (**right**).

Table 1. Description of geophysical time series.

Characteristic	Gulfstream-1	Gulfstream-2	Labrador-1	Labrador-2	Tropical-1	Tropical-2
Number of observations	14,612	14,612	14,612	14,612	14,612	14,612
Minimum value	3	−77	−52	−143	9	−28
Maximum value	995	677	330	645	403	69
Mean value	227	52	60	65	136	10

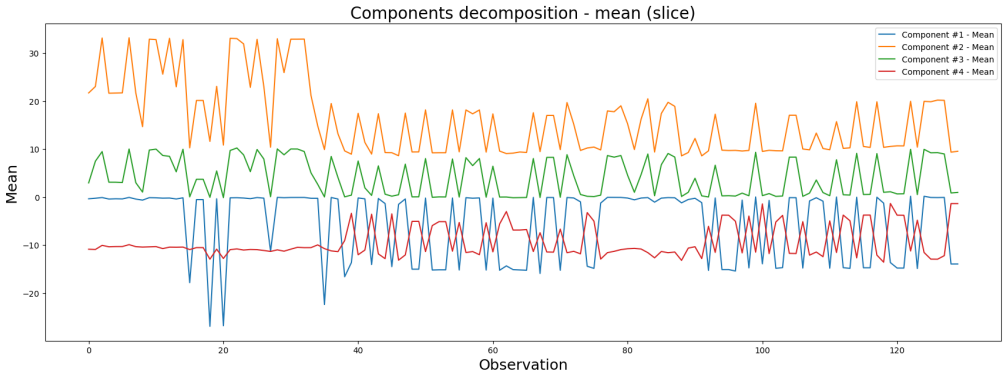

Figure 9. Example of expectations of the connected mixture components.

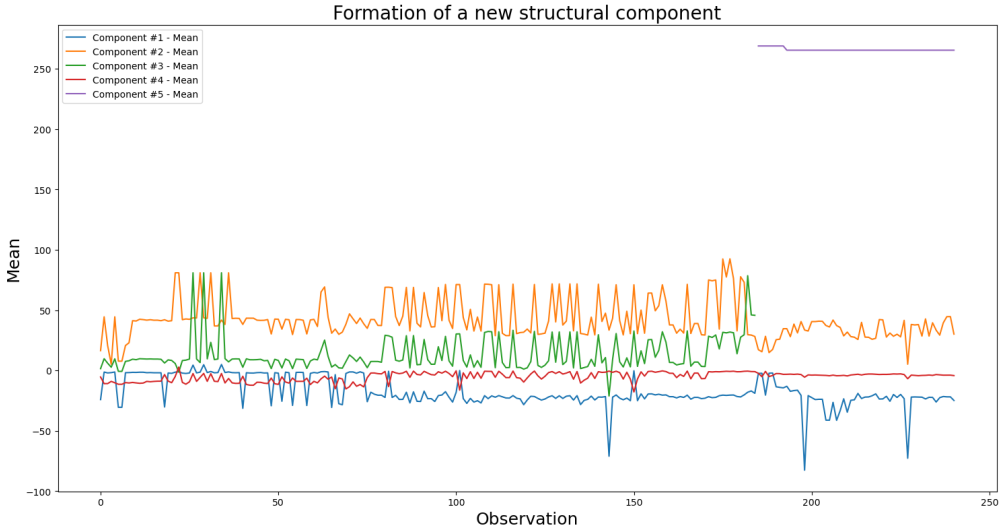

Figure 10. Example of the formation of a new connected mixture component.

We choose the transformer oil temperature (see Figure 11) as the target variable. The training set consists of about 10,000 observations from the first 12 months, and the test set consists of about 3500 observations from the last 4 months. Six features are used to measure the load: high useful load (HUFL), high useless load (HULL), middle useful load (MUFL), middle useless load (MULL), low useful load (LUFL), and low useless load (LULL). Table 2 presents their brief description.

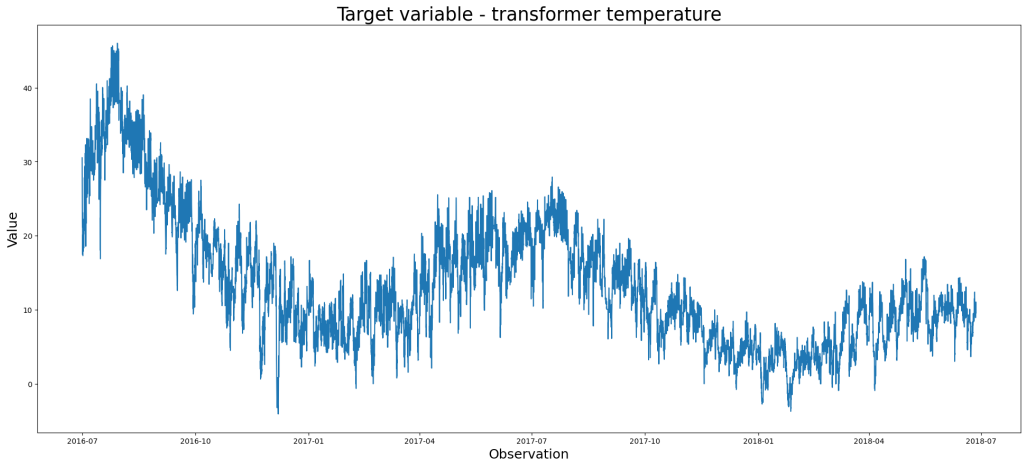

Figure 11. Example of the temperature of an electricity transformer.

Table 2. ETDataset data description.

Characteristic	HUFL	HULL	MUFL	MULL	LUFL	LULL
Number of observations	17,420	17,420	17,420	17,420	17,420	17,420
Minimum value	−22.7	−4.75	−25.0	−5.9	−1.4	−4.1
Maximum value	23.6	10.1	17.3	7.8	8.5	3.0
Mean value	7.4	2.2	4.3	0.9	3.1	0.9

Figures 12 and 13 demonstrate an example of the expectations and covariances of the connected mixture components for the ETDataset. For clarity, a subsample of 200 values is demonstrated, along with the temporal variation of the four components marked in blue, orange, green, and red. Unlike the geophysical data, the electricity transformer data has a more explicit separation of the components, especially on the expectations.

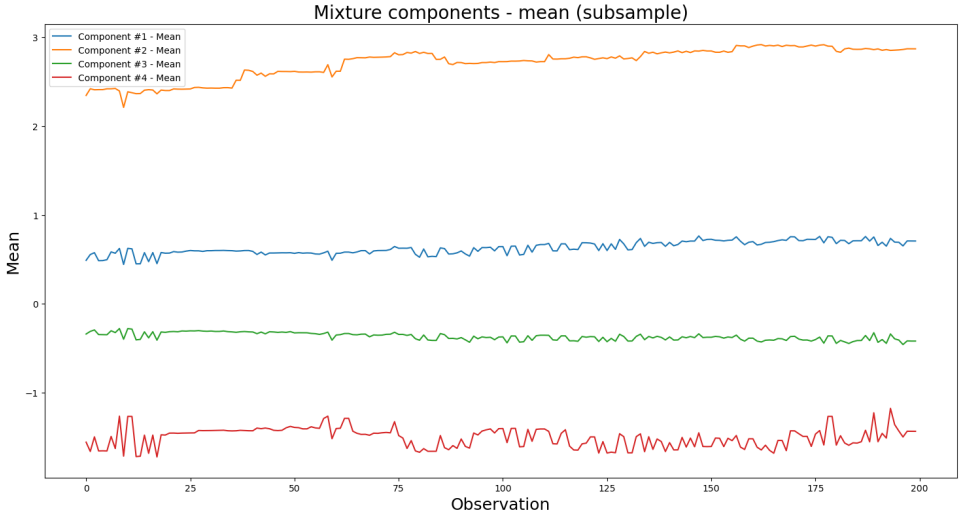

Figure 12. Example of expectations of components (ETDataset).

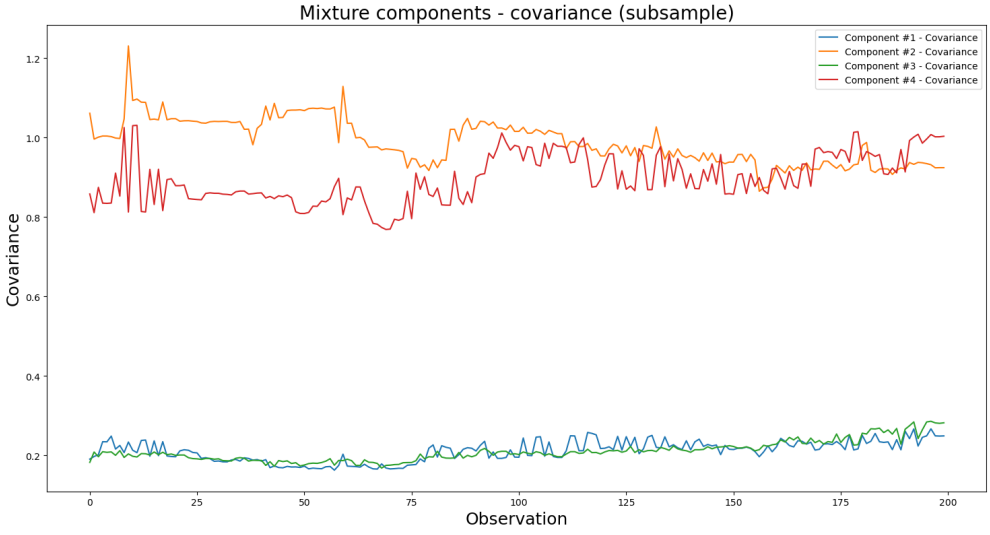

Figure 13. Example of covariances of components (ETDataset).

4.2. Accuracy Metrics, Hyperparameters, and Typical Training Times

This section describes the accuracy metrics and hyperparameters used for the models described in Section 3.3.

Geophysical data are standardized using min–max normalization:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Forecasting accuracy is measured by the RMSE and MAPE:

$$RMSE = \sqrt{\frac{1}{n}\sum(y_i - \hat{y}_i)^2}, \quad MAPE = \frac{1}{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \cdot 100\%,$$

where y_i is the target variable, $\hat{y}_i$ is the prediction.

For ease of comparison with known results [20], the ETDataset is standardized using the following normalization:

$$X_{norm} = \frac{X - X_{mean}}{\sigma^2},$$

where X_{mean} is the mean value and σ is the standard deviation. For these data, the MSE is used to determine prediction accuracy:

$$MSE = \frac{1}{n}\sum(y_i - \hat{y}_i)^2$$

Throughout the paper, the comparison of experimental results using MAPE, MSE, and RMSE is based on the relative error formula.

Tables 3–5 show the ranges of hyperparameter values for models (I)–(III). Table 6 presents the hyperparameters for architectures (IV) and (V), as well as Table 7 contains them for the ensemble architecture (VI).

The Optuna framework [79] was used for hyperparameter optimization, automating the process of finding the optimal values for ML models. It is a Sequential Model-Based Optimization method that uses information from previous hyperparameter evaluations to find the next optimal values. This framework significantly reduces the computational complexity of the tuning stage. Instead of training with 4 million configurations (as in a full grid search method), we used only 200 iterations to find the best configuration for each model. It is worth noting that the information does not change the hyperparameter space.

Table 3. Hyperparameters for model (I)—decision tree.

Hyperparameter	Value Range	Description
max_depth	5–50	Maximum tree depth
min_samples_split	2–50	Minimum number of samples required to split an internal node
min_samples_leaf	1–20	Minimum number of samples in a leaf
max_features	Sqrt, $\log_2$	Function for the maximum number of features considered when splitting

Table 4. Hyperparameters for model (II)—random forest.

Hyperparameter	Value Range	Description
n_estimators	30–300	Number of trees
max_depth	5–40	Maximum tree depth
min_samples_split	2–20	Minimum number of samples required to split an internal node
min_samples_leaf	1–10	Minimum number of samples in a leaf
max_features	Sqrt, $\log_2$	Function for the maximum number of features considered when splitting

Table 5. Hyperparameters for model (III)—gradient boosting.

Hyperparameter	Value Range	Description
n_estimators	30–300	Number of boosting steps
learning_rate	0.005–0.5	Model learning rate
max_depth	3–15	Maximum tree depth
Subsample	0.5–1.0	Part of the training data for each iteration

Table 6. Hyperparameters for architectures (IV) and (V)—LSTM.

Hyperparameter	Value Range	Description
Units_LSTM1	24–256	Number of neurons in the LSTM layer
Units_FC1	24–256	Number of neurons in the fully connected layer
Learning_rate	$5 \cdot 10^{-5}$–10^{-3}	Model learning rate
Dropout_rate	0–50%	Dropout layer parameter
L_1 Regularization	10^{-5}–10^{-3}	L_1 regularization parameter
L_2 Regularization	10^{-6}–10^{-4}	L_2 regularization parameter
Epochs	50–900	50 epochs are for all, while the best models are trained for another 850 epochs

Table 7. Hyperparameters for architecture (VI)—LSTM and transformer.

Hyperparameter	Value Range	Description
Units_LSTM1	24–256	Number of neurons in the LSTM layer
Units_FC1	24–256	Number of neurons in the fully connected layer
Num_heads	2–32	Number of transformer attention heads
Num_layers	1–10	Number of transformer layers
Hidden_size	128–4096	transformer hidden layer size
Units_final	4–128	Number of neurons in the feature fusion layer
Learning_rate	$5 \cdot 10^{-5}$–10^{-3}	Model learning rate
Dropout_rate	0–50%	Dropout layer parameter
L_1 Regularization	10^{-5}–10^{-3}	L_1 regularization parameter
L_2 Regularization	10^{-6}–10^{-4}	L_2 regularization parameter
Epochs	50–900	50 epochs are for all, while the best models are trained for another 850 epochs

A high-performance computing cluster based on NVIDIA V100 GPU 32 GB was used for training the models. The training time of a single NN model was as follows:

- Seven minutes for architecture (IV) on geophysical data;
- Fifteen minutes for architecture (V) on geophysical data;
- Forty minutes for architecture (V) on ETDataset (due to the larger input space);
- Three hours for an ensemble architecture (VI) on geophysical data;
- Five hours for architecture (VI) on ETDataset.

4.3. Geophysical Data Forecasting

For each of the models (I)–(VI) from Section 3.3, the geophysical data (see Section 4.1) was used to compare forecasting results with and without informing by connected mixture components. We consider the short-term (18 h, i.e., 3 observations) and the medium-term periods (60 h, i.e., 10 observations). Tables 8–11 present the corresponding forecasting results:

- Short-term forecasts: Table 8 (RMSE) and Table 9 (MAPE);
- Medium-term forecasts: Table 10 (RMSE) and Table 11 (MAPE).

Minimum values in Tables 8–11 are marked in bold.

Based on the results of experiments with various configurations of hyperparameters, it was established that for a forecast on 3 observations, informing with connected mixture components leads to the following ranges of the RMSE decrease (see Table 8):

- From 1.4% to 3.7% for model (I);
- From 0.8% to 2.7% for model (II);
- From 1.0% to 2.7% for model (III);
- From 1.4% to 15.9% for architecture (IV);
- From 9.4% to 27.7% for architecture (V);
- From 4.8% to 26.7% for architecture (VI).

The best accuracy values were obtained using the informed ensemble architecture of LSTM and transformer (VI).

The ranges of the MAPE decrease are as follows (see Table 9):

- From 1.4% to 5.0% for model (I);
- From 1.5% to 5.0% for model (II);
- From 0.8% to 3.9% for model (III);
- From 7.5% to 22.5% for architecture (IV);
- From 20.5% to 45.7% for architecture (V);
- From 5.5% to 26.2% for architecture (VI).

The best accuracy values in most cases were obtained using the informed ensemble architecture of LSTM and transformer (VI), although for Labrador-1 and Tropical-2, the informed LSTM (V) and random forest (II) performed slightly better. From the comparison of results for architecture (IV) and architecture (V), it is evident that merging with a multivariate set of components yields a greater accuracy increase than providing individual values.

Table 8. Forecasts for 3 observations, RMSE.

Model	Gulfstream-1	Gulfstream-2	Labrador-1	Labrador-2	Tropical-1	Tropical-2
Decision tree	0.178	0.103	0.141	0.111	0.151	0.083
Informed decision tree (I)	0.174	0.100	0.139	0.109	0.147	0.080
Random forest	0.125	0.074	0.095	0.076	0.108	0.059
Informed random forest (II)	0.122	0.072	0.094	0.074	0.107	**0.058**
Gradient boosting	0.134	0.080	0.103	0.081	0.115	0.062
Informed gradient boosting (III)	0.131	0.079	0.102	0.079	0.112	0.061
LSTM	0.078	0.076	0.080	0.073	0.083	0.070
Informed LSTM (IV)	0.069	0.071	0.069	0.072	0.075	0.061
Informed LSTM (V)	0.065	0.061	0.067	0.066	0.065	**0.058**
LSTM + transformer	0.074	0.076	0.078	0.065	0.069	0.063
Informed LSTM + transformer (VI)	**0.060**	**0.060**	**0.066**	**0.062**	**0.061**	**0.058**

Table 9. Forecasts for 3 observations, MAPE.

Model	Gulfstream-1	Gulfstream-2	Labrador-1	Labrador-2	Tropical-1	Tropical-2
Decision tree	34.8%	21.1%	25.5%	19.9%	25.3%	13.7%
Informed decision tree (I)	33.7%	20.1%	24.7%	19.1%	24.8%	13.5%
Random forest	25.4%	18.3%	18.6%	13.5%	20.0%	10.8%
Informed random forest (II)	25.2%	17.7%	17.9%	13.3%	19.7%	**10.4%**
Gradient boosting	29.3%	18.5%	22.3%	18.4%	21.2%	14.5%
Informed gradient boosting (III)	27.9%	18.2%	22.0%	17.7%	20.9%	14.3%
LSTM	16.8%	17.2%	21.8%	15.3%	13.4%	16.9%
Informed LSTM (IV)	14.4%	16.0%	18.5%	13.6%	11.2%	13.8%
Informed LSTM (V)	13.3%	13.1%	**15.7%**	12.7%	10.3%	11.6%
LSTM + transformer	14.3%	15.8%	17.4%	13.0%	10.7%	12.4%
Informed LSTM + transformer (VI)	**12.8%**	**12.9%**	16.5%	**10.3%**	**8.8%**	10.7%

Table 10. Forecasts for 10 observations, RMSE.

Model	Gulfstream-1	Gulfstream-2	Labrador-1	Labrador-2	Tropical-1	Tropical-2
Decision tree	0.201	0.126	0.158	0.130	0.192	0.103
Informed decision tree (I)	0.196	0.124	0.154	0.125	0.187	0.101
Random forest	0.145	0.093	0.115	0.100	0.138	0.075
Informed random forest (II)	0.139	0.092	0.109	0.095	0.133	0.073
Gradient boosting	0.155	0.099	0.124	0.106	0.149	0.080
Informed gradient boosting (III)	0.152	0.093	0.122	0.103	0.145	0.078
LSTM	0.092	0.090	0.096	0.089	0.101	0.085
Informed LSTM (IV)	0.083	0.086	0.088	0.084	0.092	0.075
Informed LSTM (V)	0.082	0.084	0.081	0.081	0.083	0.071
LSTM + transformer	0.085	0.089	0.092	0.079	0.084	0.076
Informed LSTM + transformer (VI)	**0.074**	**0.076**	**0.077**	**0.075**	**0.071**	**0.070**

Table 11. Forecasting errors in the MAPE metric—forecast for 10 observations.

Model	Gulfstream-1	Gulfstream-2	Labrador-1	Labrador-2	Tropical-1	Tropical-2
Decision tree	38.3%	25.9%	26.7%	24.4%	32.2%	17.0%
Informed decision tree (I)	38.0%	24.9%	26.4%	21.9%	31.5%	17.0%
Random forest	29.5%	23.0%	22.5%	17.8%	25.6%	13.7%
Informed random forest (II)	28.8%	22.5%	20.6%	17.2%	24.4%	13.1%
Gradient boosting	33.2%	22.6%	26.4%	24.2%	27.4%	18.8%
Informed gradient boosting (III)	32.4%	21.9%	26.2%	23.1%	26.5%	18.4%
LSTM	18.7%	19.7%	23.5%	16.5%	16.0%	19.7%
Informed LSTM (IV)	17.3%	18.1%	22.0%	15.4%	14.7%	17.3%
Informed LSTM (V)	16.8%	18.0%	20.1%	15.0%	13.2%	15.2%
LSTM + transformer	16.6%	18.3%	20.0%	15.4%	13.0%	14.8%
Informed LSTM + transformer (VI)	**15.7%**	**16.4%**	**19.2%**	**12.6%**	**11.3%**	**12.8%**

In the experiments for forecasts on 10 observations, similar results were obtained. The ranges of the RMSE decrease are as follows (see Table 10):

- From 1.6% to 4.0% for model (I);
- From 1.1% to 5.5% for model (II);
- From 1.6% to 6.4% for model (III);
- From 4.6% to 13.3% for architecture (IV);
- From 7.1% to 21.7% for architecture (V);
- From 5.3% to 19.5% for architecture (VI).

The best accuracy values were obtained using the informed ensemble architecture of LSTM and transformer (VI).

The ranges of the MAPE decrease are as follows (see Table 11):

- From 0.1% to 11.4% for model (I);
- From 2.2% to 9.2% for model (II);
- From 0.7% to 4.8% for model (III);
- From 6.8% to 13.8% for architecture (IV);
- From 9.4% to 29.6% for architecture (V);
- From 4.2% to 22.2% for architecture (VI).

The best accuracy values in most cases were obtained using the informed ensemble architecture of LSTM and transformer (VI).

Figure 14 shows examples of short-term forecasts for the best model (VI). Each graph represents a part of the test set and includes actual data and the forecast for the same geographical location at the corresponding time. The forecast for future steps was built using actual observations, with no use of prior forecasts. In the upper left corner of Figure 14, the forecast for a sample of 150 observations for the Gulfstream-1 (RMSE = 0.060) is presented. Similarly, in the upper right corner, the forecast for the Gulfstream-2 (RMSE = 0.060) is shown; the forecast for the Labrador-1 (RMSE = 0.066) is in the middle left; the forecast for the Labrador-2 series (RMSE = 0.062) is in the middle right; the forecast for the Tropical-1 (RMSE = 0.061) is in the bottom left; the forecast for the Tropical-2 (RMSE =0.058) is in the bottom right.

From the presented results, it is evident that merging with a multinomial set of connected mixture components is more preferable in terms of accuracy compared with schemes that add individual values. In particular, greater accuracy was achieved for at least half of the test series (Gulfstream-1, Tropical-1, Tropical-2) for geophysical data compared with the values presented in the paper [58]. For the remaining series, the results are comparable, and the differences can be explained by computational errors. This confirms the conclusion that this type of probability informing can be effectively used in various machine learning models to improve forecast quality.

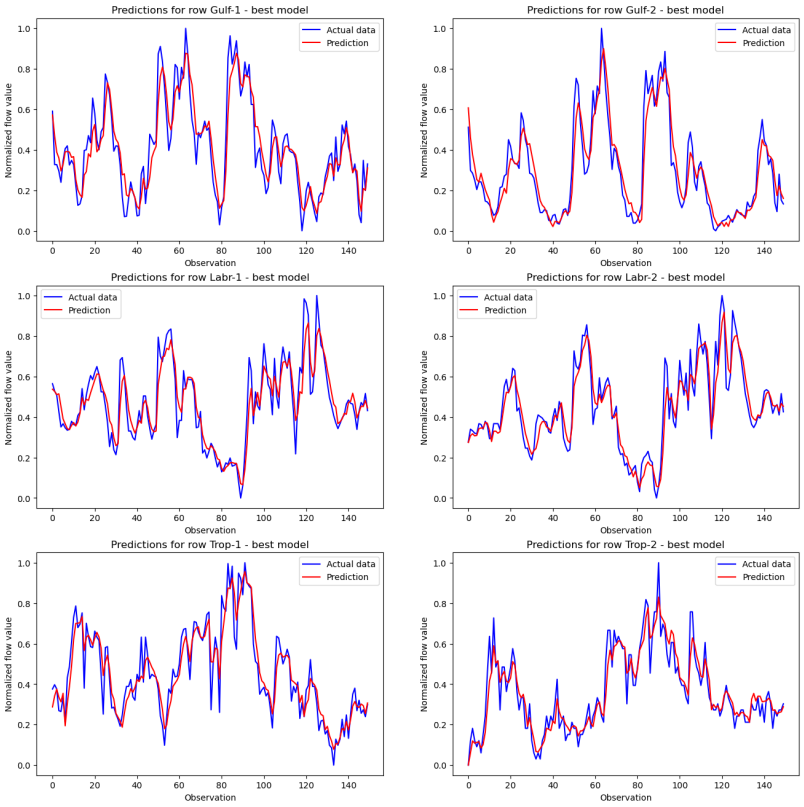

Figure 14. Example of geophysical data short-term forecasts.

4.4. ETDatset Forecasting

For each model (I)-(III) and (V), (VI) from Section 3.3 concerning electricity transformers, forecasting results with and without informing by connected mixture components are compared based on MSE. There is a medium-term forecast: 48 h, which corresponds to 48 observations in the time series. Architecture (IV) is not used because the input data already represent a multivariate series, that is, architecture (V). Table 12 presents the corresponding results. Minimum values are marked in bold.

Table 12. Forecasting errors for ETDataset.

Model	MSE	RMSE
Decision tree	0.255	0.505
Informed decision tree (I)	0.248	0.498
Random forest	0.178	0.422
Informed random forest (II)	**0.166**	**0.407**
Gradient boosting	0.175	0.418
Informed gradient boosting (III)	0.173	0.416
LSTM	0.232	0.482
Informed LSTM (V)	0.211	0.459
LSTM + transformer	0.224	0.473
Informed LSTM + transformer (VI)	0.207	0.455

The experimental results for various hyperparameter configurations show that the informed models with connected mixture components have improvements in MSE accuracy. They are higher by the following amounts:

- A total of 2.8% for model (I);
- A total of 7.2% for model (II);
- A total of 1.2% for model (III);
- A total of 10.0% for architecture (V);
- A total of 8.2% for architecture (VI).

The minimal MSE of 0.166 was demonstrated by the informed random forests. Figure 15 shows an example of forecasting the test dataset using the model (II).

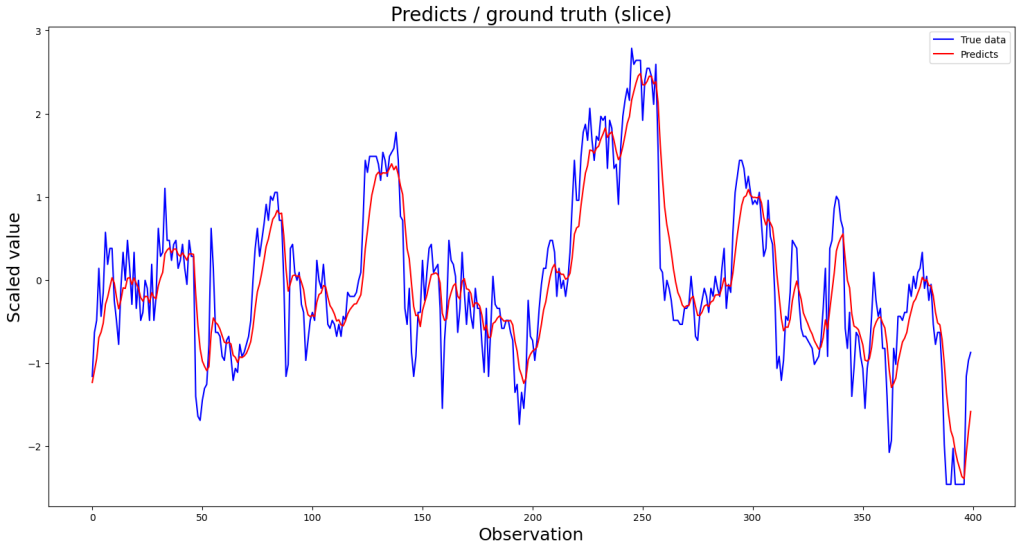

Figure 15. Example of forecasts for electricity transformer data.

In the original paper [20], a slightly better result (MSE = 0.158) was obtained using a significantly more complex modern transformer NN architecture, Informer, for forecasting

long-term dependencies. This accuracy value is lower than the corresponding one for the informed random forest by only 0.008. Moreover, our probability-informed approach allows us to outperform the previously obtained values [20] by both the transfomer NN architecture known as Reformer [80] (MSE = 0.284) and the statistical method Prophet [81] (MSE = 0.168).

5. Conclusions and Discussion

The paper suggests an approach to using connected mixture components as additional features for improving the efficiency of machine learning models. We consider six schemes that are suitable for both standard ML methods and deep neural network architectures.

Based on the results of experiments, we show an improvement in forecasting accuracy measured by RMSE from 0.8% to 27.7% and by MAPE from 0.1% to 45.7% due to the expansion of the feature space for geophysical data. An improvement measured by MSE is from 1.2% to 10.0% for electricity transformers. It is demonstrated that informing based on multidimensional time series gives a more significant improvement in model forecasts than a separate set of values: RMSE is reduced by 1.2–16.4%, MAPE is by 0.5–22.2%. Based on the results obtained, we can conclude that a probability-informed approach is effective not only for neural networks, including ensemble ones, but also for classical ML models (decision tree, random forest, gradient boosting).

Further research directions involve exploring more complex data representations, such as those based on various types of autoencoders, including variational ones. This approach aims to address issues related to the dynamically changing number of mixture components, which are additional features for ML models. Additionally, it can lead to the development of more specialized deep probability-informed NN architectures to enhance the ability of such models to process real data in a more universal manner. For example, it is of great interest to apply these approaches in a wide range of applications, including telecommunications traffic [82]. In this area, significant progress has been made in solving problems related to forecasting and detecting anomalous observations using probabilistic and statistical models within machine learning [83], primarily deep Gaussian models [84]. The probability informing based on connected mixture components in this research field is promising for the future.

Author Contributions: Conceptualization, A.K.G.; formal analysis, A.K.G. and A.L.V.; funding acquisition, A.K.G.; investigation, A.K.G. and A.L.V.; methodology, A.K.G.; project administration, A.K.G.; resources, A.K.G.; supervision, A.K.G.; validation, A.K.G. and A.L.V.; visualization, A.K.G. and A.L.V.; writing–original draft, A.K.G. and A.L.V.; writing–review and editing, A.K.G. and A.L.V. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the Ministry of Science and Higher Education of the Russian Federation, project No. 075-15-2024-544.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The geophysical data used was confidential: spatio-temporal time series were kindly provided by a world-renowned researcher in the field of the ocean and atmospheric sciences, Corresponding member of the Russian Academy of Sciences, Professor S. K. Gulev (Russia). The Electricity Transformer Dataset is available online: https://github.com/zhouhaoyi/ETDataset (accessed on 1 July 2024).

Acknowledgments: The research was carried out using the infrastructure of the Shared Research Facilities "High Performance Computing and Big Data" (CKP "Informatics") of the Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Bzdok, D.; Altman, N.; Krzywinski, M. Statistics versus Machine Learning. *Nat. Methods* **2018**, *15*, 233–234. [CrossRef] [PubMed]
2. Korb, K.; Nicholson, A. *Bayesian Artificial Intelligence*; Chapman and Hall/CRC: London, UK, 2011.
3. Murphy, K. *Probabilistic Machine Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2022.

4. James, G.; Daniela, W.; Trevor, H.; Robert, T. *An Introduction to Statistical Learning: With Applications in R*; Springer: Berlin/Heidelberg, Germany, 2023.
5. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
6. Dong, S.; Wang, P.; Abbas, K. A Survey on Deep Learning and its Applications. *Computer Sci. Rev.* **2021**, *40*, 100379. [CrossRef]
7. Lim, B.; Zohren, S. Time-series Forecasting with Deep Learning: A Survey. *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.* **2021**, *379*, 20170179. [CrossRef] [PubMed]
8. Torres, J.F.; Hadjout, D.; Sebaa, A.; Martínez-Álvarez, F.; Troncoso, A. Deep Learning for Time Series Forecasting: A Survey. *Big Data* **2021**, *9*, 3–21. [CrossRef]
9. Benidis, K.; Rangapuram, S.S.; Flunkert, V.; Wang, Y.; Maddix, D.; Turkmen, C.; Gasthaus, J.; Bohlke-Schneider, M.; Salinas, D.; Stella, L.; et al. Deep Learning for Time Series Forecasting: Tutorial and Literature Survey. *ACM Comput. Surv.* **2022**, *55*, 1–36. [CrossRef]
10. Chen, Z.; Ma, M.; Li, T.; Wang, H.; Li, C. Long sequence time-series forecasting with deep learning: A survey. *Inf. Fusion* **2023**, *97*, 101819. [CrossRef]
11. Safonova, A.; Ghazaryan, G.; Stiller, S.; Main-Knorn, M.; Nendel, C.; Ryo, M. Ten Deep Learning Techniques to Address Small Data Problems with Remote Sensing. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *125*, 103569. [CrossRef]
12. Xu, P.; Ji, X.; Li, M.; Lu, W. Small Data Machine Learning in Materials Science. *NPJ Comput. Mater.* **2023**, *9*, 42. [CrossRef]
13. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]
14. Alkilane, K.; He, Y.; Lee, D.H. MixMamba: Time series modeling with adaptive expertise. *Inf. Fusion* **2024**, *112*, 102589. [CrossRef]
15. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-Informed Machine Learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [CrossRef]
16. Härdle, W.; Werwatz, A.; Müller, M.; Sperlich, S. *Nonparametric and Semiparametric Models*; Springer Series in Statistics; Springer: Berlin/Heidelberg, Germany, 2004. [CrossRef]
17. Safavian, S.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [CrossRef]
18. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
19. Schapire, R.E. The Boosting Approach to Machine Learning: An Overview. In *Lecture Notes in Statistics*; Springer: New York, NY, USA, 2003; pp. 149–171. [CrossRef]
20. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 11106–11115. [CrossRef]
21. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; NIPS'17, pp. 6000–6010.
23. Scott, S.; Matwin, S. Feature engineering for text classification. In Proceedings of the ICML, Bled, Slovenia, 27–30 June 1999; Volume 99, pp. 379–388.
24. Mutlag, W.K.; Ali, S.K.; Aydam, Z.M.; Taher, B.H. Feature Extraction Methods: A Review. *J. Phys. Conf. Ser.* **2020**, *1591*, 012028. [CrossRef]
25. Fernandes, S.V.; Ullah, M.S. A Comprehensive Review on Features Extraction and Features Matching Techniques for Deception Detection. *IEEE Access* **2022**, *10*, 28233–28246. [CrossRef]
26. Zhou, H.; Li, J.; Zhang, S.; Zhang, S.; Yan, M.; Xiong, H. Expanding the Prediction Capacity in Long Sequence Time-Series Forecasting. *Artif. Intell.* **2023**, *318*, 103886. [CrossRef]
27. Jia, B.; Wu, H.; Guo, K. Chaos Theory Meets Deep Learning: A New Approach to Time Series Forecasting. *Expert Syst. Appl.* **2024**, *255*, 124533. [CrossRef]
28. Cruz, L.F.S.A.; Silva, D.F. Financial Time Series Forecasting Enriched with Textual Information. In Proceedings of the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Virtual, 13–15 December 2021; pp. 385–390. [CrossRef]
29. Plutenko, I.; Papkov, M.; Palo, K.; Parts, L.; Fishman, D. Metadata Improves Segmentation Through Multitasking Elicitation. In Proceedings of the Domain Adaptation and Representation Transfer, Vancouver, BC, Canada, 12 October 2024; pp. 147–155.
30. Raissi, M.; Perdikaris, P.; Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
31. Mao, Z.; Jagtap, A.D.; Karniadakis, G.E. Physics-informed neural networks for high-speed flows. *Comput. Methods Appl. Mech. Eng.* **2020**, *360*, 112789. [CrossRef]
32. Cai, S.; Mao, Z.; Wang, Z.; Yin, M.; Karniadakis, G.E. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mech. Sin.* **2021**, *37*, 1727–1738. [CrossRef]
33. Jin, X.; Cai, S.; Li, H.; Karniadakis, G.E. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *J. Comput. Phys.* **2021**, *426*, 109951. [CrossRef]

34. Li, Y.; Xiao, L.; Wei, H.; Kou, Y.; Yang, L.; Li, D. A Time-Frequency Physics-Informed Model for Real-Time Motion Prediction of Semi-Submersibles. *Ocean. Eng.* **2024**, *299*, 117379. [CrossRef]
35. Saito, N.; Coifman, R.R.; Geshwind, F.B.; Warner, F. Discriminant feature extraction using empirical probability density estimation and a local basis library. *Pattern Recognit.* **2002**, *35*, 2841–2852. [CrossRef]
36. Gorodetsky, V.; Samoylov, V. Feature Extraction for Machine Learning: Logic-Probabilistic Approach. In Proceedings of the Fourth International Workshop on Feature Selection in Data Mining, Hyderabad, India, 21 June 2010; Volume 10, pp. 55–65.
37. Le, T.; Schuff, N. A Probability-Based Approach for Multi-scale Image Feature Extraction. In Proceedings of the 2014 11th International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 7–9 April 2014; pp. 143–148. [CrossRef]
38. Ma, Y.; Huang, B. Bayesian Learning for Dynamic Feature Extraction With Application in Soft Sensing. *IEEE Trans. Ind. Electron.* **2017**, *64*, 7171–7180. [CrossRef]
39. Yan, H.; He, L.; Song, X.; Yao, W.; Li, C.; Zhou, Q. Bidirectional Statistical Feature Extraction Based on Time Window for Tor Flow Classification. *Symmetry* **2022**, *14*, 2002. [CrossRef]
40. Subramanian, A.; Mahadevan, S. Probabilistic Physics-Informed Machine Learning for Dynamic Systems. *Reliab. Eng. Syst. Saf.* **2023**, *230*, 108899. [CrossRef]
41. Fuhg, J.N.; Bouklas, N. On Physics-Informed Data-Driven Isotropic and Anisotropic Constitutive Models Through Probabilistic Machine Learning and Space-Filling Sampling. *Comput. Methods Appl. Mech. Eng.* **2022**, *394*, 114915. [CrossRef]
42. Zhou, T.; Jiang, S.; Han, T.; Zhu, S.P.; Cai, Y. A Physically Consistent Framework for Fatigue Life Prediction Using Probabilistic Physics-Informed Neural Network. *Int. J. Fatigue* **2023**, *166*, 107234. [CrossRef]
43. Gorshenin, A.; Kuzmin, V. Method for improving accuracy of neural network forecasts based on probability mixture models and its implementation as a digital service. *Inform. Primen.* **2021**, *15*, 63–74. [CrossRef]
44. Gorshenin, A.K.; Vilyaev, A.L. Finite Normal Mixture Models for the Ensemble Learning of Recurrent Neural Networks with Applications to Currency Pairs. *Pattern Recognit. Image Anal.* **2022**, *32*, 780–792. [CrossRef]
45. Itô, K. *On Stochastic Differential Equations*; Number 4; American Mathematical Society: Washington, DC, USA, 1951.
46. Gikhman, I.; Skorokhod, A.V. *The Theory of Stochastic Processes II*; Springer: Berlin/Heidelberg, Germany, 2004.
47. Wu, X.; Kumar, V.; Quinlan, J.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.J.; Ng, A.; Yu, P.S.; Zhou, Z.-H.; et al. Top 10 Algorithms in Data Mining. *Knowl. Inf. Syst.* **2008**, *14*, 1–37. [CrossRef]
48. Gorshenin, A.K. On Implementation of EM-type Algorithms in the Stochastic Models for a Matrix Computing on GPU. *AIP Conf. Proc.* **2015**, *1648*, 250008. [CrossRef]
49. Belyaev, K.P.; Gorshenin, A.K.; Korolev, V.Y.; Osipova, A.A. Comparison of Statistical Approaches for Reconstructing Random Coefficients in the Problem of Stochastic Modeling of Air–Sea Heat Flux Increments. *Mathematics* **2024**, *12*, 288. [CrossRef]
50. Gorshenin, A.; Korolev, V.; Shcherbinina, A. Statistical estimation of distributions of random coefficients in the Langevin stochastic differential equation. *Inform. Primen.* **2020**, *14*, 3–12. [CrossRef]
51. Liu, C.; Li, H.; Fu, K.; Zhang, F.; Datcu, M.; Emery, W. A Robust EM Clustering Algorithm for Gaussian Mixture Models. *Pattern Recognit.* **2012**, *45*, 3950–3961. [CrossRef]
52. Wu, D.; Ma, J. An Effective EM Algorithm for Mixtures of Gaussian Processes via the MCMC Sampling and Approximation. *Neurocomputing* **2019**, *331*, 366–374. [CrossRef]
53. Zeller, C.B.; Cabral, C.R.B.; Lachos, V.H.; Benites, L. Finite mixture of regression models for censored data based on scale mixtures of normal distributions. *Adv. Data Anal. Classif.* **2018**, *13*, 89–116. [CrossRef]
54. Abid, S.; Quaez, U.; Contreras-Reyes, J. An Information-Theoretic Approach for Multivariate Skew-t Distributions and Applications. *Mathematics* **2021**, *9*, 146. [CrossRef]
55. Audhkhasi, K.; Osoba, O.; Kosko, B. Noise-Enhanced Convolutional Neural Networks. *Neural Netw.* **2016**, *78*, 15–23. [CrossRef] [PubMed]
56. Greff, K.; van Steenkiste, S.; Schmidhuber, J. Neural Expectation Maximization. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6694–6704.
57. Kolmogorov, A.; Fomin, S. *Elements of the Theory of Functions and Functional Analysis*; FIZMATLIT: Moscow, Russia, 2004.
58. Gorshenin, A.K.; Kuzmin, V.Y. Statistical Feature Construction for Forecasting Accuracy Increase and its Applications in Neural Network Based Analysis. *Mathematics* **2022**, *10*, 589. [CrossRef]
59. Karpathy, A.; Li, F.-F. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 3128–3137.
60. Wang, B.; Jiang, T.; Zhou, X.; Ma, B.; Zhao, F.; Wang, Y. Time-Series Classification Based on Fusion Features of Sequence and Visualization. *Appl. Sci.* **2020**, *10*, 4124. [CrossRef]
61. Chang, J.; Jin, L. Gating Mechanism Based Feature Fusion Networks for Time Series Classification. In Proceedings of the 2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Wuhan, China, 22–24 April 2022; pp. 147–150. [CrossRef]
62. Wang, T.; Liu, Z.; Zhang, T.; Hussain, S.F.; Waqas, M.; Li, Y. Adaptive feature fusion for time series classification. *Knowl.-Based Syst.* **2022**, *243*, 108459. [CrossRef]
63. Park, S.H.; Syazwany, N.S.; Lee, S.C. Meta-Feature Fusion for Few-Shot Time Series Classification. *IEEE Access* **2023**, *11*, 41400–41414. [CrossRef]

64. Perry, A.; Walker, J. *The Ocean-Atmosphere System*; Longman: London, UK, 1977.
65. Gorshenin, A.K.; Osipova, A.A.; Belyaev, K.P. Stochastic analysis of air–sea heat fluxes variability in the North Atlantic in 1979–2022 based on reanalysis data. *Comput. Geosci.* **2023**, *181*, 105461. [CrossRef]
66. Hersbach, H.; Bell, B.; Berrisford, P.; Hirahara, S.; Horányi, A.; Muñoz-Sabater, J.; Nicolas, J.; Peubey, C.; Radu, R.; Schepers, D.; et al. The ERA5 global reanalysis. *Q. J. R. Meteorol. Soc.* **2020**, *146*, 1999–2049. [CrossRef]
67. Gavrikov, A.; Gulev, S.K.; Markina, M.; Tilinina, N.; Verezemskaya, P.; Barnier, B.; Dufour, A.; Zolina, O.; Zyulyaeva, Y.; Krinitskiy, M.; et al. RAS-NAAD: 40-yr high-resolution north atlantic atmospheric hindcast for multipurpose applications (new dataset for the regional mesoscale studies in the atmosphere and the ocean). *J. Appl. Meteorol. Climatol.* **2020**, *59*, 793–817. [CrossRef]
68. Grainger, J.J.; Stevenson, W.D. *Power System Analysis*; McGraw Hill: New York, NY, USA, 1994.
69. Weedy, B.; Cory, B.; Jenkins, N.; Ekanayake, J.; Strbac, G. *Electric Power Systems*; Wiley: Hoboken, NJ, USA, 2012.
70. Banchuin, R.; Chaisricharoen, R. An SDE based Stochastic Analysis of Transformer. In Proceedings of the 2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON), Barcelona, Spain, 30 January 2019; pp. 310–313. [CrossRef]
71. Schein, O.; Denk, G. Numerical solution of stochastic differential-algebraic equations with applications to transient noise simulation of microelectronic circuits. *J. Comput. Appl. Math.* **1998**, *100*, 77–92. [CrossRef]
72. Römisch, W.; Winkler, R. Stochastic DAEs in Circuit Simulation. In Proceedings of the Modeling, Simulation, and Optimization of Integrated Circuits, Basel, Switzerland, 23 October 2003; pp. 303–318. [CrossRef]
73. Kolarova, E. Modelling RL Electrical Circuits by Stochastic Diferential Equations. In Proceedings of the EUROCON 2005—The International Conference on "Computer as a Tool", Belgrade, Serbia, 21–24 November 2005; Volume 2, pp. 1236–1238. [CrossRef]
74. Patil, N.S.; Sharma, S.N. On a non-linear stochastic dynamic circuit using Stratonovich differential. *J. Frankl. Inst.* **2015**, *352*, 2999–3013. [CrossRef]
75. Huy, P.C.; Minh, N.Q.; Tien, N.D.; Anh, T.T.Q. Short-Term Electricity Load Forecasting Based on Temporal Fusion Transformer Model. *IEEE Access* **2022**, *10*, 106296–106304. [CrossRef]
76. Torres, J.; Martí'nez-Álvarez, F.; Troncoso, A. A Deep LSTM Network for the Spanish Electricity Consumption Forecasting. *Neural Comput. Appl.* **2022**, *34*, 10533–10545. [CrossRef] [PubMed]
77. Wang, C.; Wang, Y.; Ding, Z.; Zheng, T.; Hu, J.; Zhang, K. A Transformer-Based Method of Multienergy Load Forecasting in Integrated Energy System. *IEEE Trans. Smart Grid* **2022**, *13*, 2703–2714. [CrossRef]
78. Cui, Y.; Li, Z.; Wang, Y.; Dong, D.; Gu, C.; Lou, X.; Zhang, P. Informer Model with Season-Aware Block for Efficient Long-Term Power Time Series Forecasting. *Comput. Electr. Eng.* **2024**, *119*, 109492. [CrossRef]
79. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 4–9 August 2019; KDD '19, pp. 2623–2631. [CrossRef]
80. Kitaev, N.; Kaiser, L.; Levskaya, A. Reformer: The Efficient Transformer. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26 April–1 May 2020.
81. Taylor, S.J.; Letham, B. Forecasting at Scale. *Am. Stat.* **2018**, *72*, 37–45. [CrossRef]
82. Kochetkova, I.; Kushchazli, A.; Burtseva, S.; Gorshenin, A. Short-Term Mobile Network Traffic Forecasting Using Seasonal ARIMA and Holt-Winters Models. *Future Internet* **2023**, *15*, 290. [CrossRef]
83. Gorshenin, A.; Kozlovskaya, A.; Gorbunov, S.; Kochetkova, I. Mobile network traffic analysis based on probability-informed machine learning approach. *Comput. Netw.* **2024**, *247*, 110433. [CrossRef]
84. Viroli, C.; McLachlan, G. Deep Gaussian Mixture Models. *Stat. Comput.* **2019**, *29*, 43–51. [CrossRef]