Handling Massive Proportion of Missing Labels in Multivariate Long-Term Time Series Forecasting

Cristovão Iglesias Jr, Varun Mehta, Alina Venereo-Sanchez, Xingge Xu, Julien Robitaille, Robert Voyer, René Richard, Nabil Belacel, Amine Kamen, Miodrag Bolic

1 University of Ottawa, School of Electrical Engineering and Computer Science (EECS), Canada
2 National Research Council of Canada, Canada
3 Department of Bioengineering, McGill University, Canada

E-mail: cfrei096@uottawa.ca

Abstract.

Training Deep Learning (DL) models with missing labels is a challenge in diverse engineering applications. Missing value imputation methods have been proposed to try to address this problem, but their performance is affected with Massive Proportion of Missing Labels (MPML). This paper presents an approach for handling MPML in Multivariate Long-Term Time Series Forecasting. It is a two-step process where interpolation (using Gaussian Processes Regression (GPR) and domain knowledge from experts) and prediction model are separated to enable the integration of prior domain knowledge. First, a set of samples of the possible interpolation of the missing outputs are generated by the GPR based on the domain knowledge. Second, the observed input sensor data and interpolated labels from GPR are used to train the prediction model. We evaluated our approach with the development of a soft-sensor with one real dataset to forecast the biomass during recombinant adeno-associated virus (rAAV) production in bioreactors. Our experimental results demonstrate the potential of the approach through quantitative evaluation of the generated forecasts in a case that would be extremely difficult to train a DL model due to MPML.

1. Introduction

Multivariate Long-Term Time Series Forecasting (MLTTSF) task aims to predict the future values of a series based on the history of multiple series [1], and it is an important problem across several domains [2]. The Deep Learning (DL) techniques have an effective and important role in addressing MLTTSF task due to their ability to handle multiple input and output variables and handle complex nonlinear relationships. In addition, they do not require a scaled or stationary time series as an input [3, 1], and they have outperformed the traditional models, such as Autoregressive Integrated Moving Average (ARIMA) in several applications [4, 5, 6, 7, 8].

Despite the excellent performance of DL approaches, training them with missing labels in time series is still a challenge [9, 2, 10]. Here missing label (or missing data target) is defined as the data value that is not stored for a variable in the observation of interest [11]. There are three type mechanisms of missingness by which the data is missing: i) Missing Completely at Random (MCAR) when the fact that the data is missing is independent of the observed and unobserved data, ii) Missing at Random (MAR) when the probability of being missing is the...
same only within groups defined by the observed data and iii) Missing Not at Random (MNAR) if neither MCAR nor MAR holds \[12, 13, 14\]. DL models are Data-Driven Models (DDM), and they need data to learn through prediction error minimization. However, the learning process is inefficient when not enough data is available due to missing labels or low quality of data. The traditional methods formulate the optimization problem as a function of model parameters and solve it using classical gradient descent approaches. But here, due to too many missing labels, we cannot use the squared error function to optimize the model because it does not have enough values to be used in the comparison that allows minimizing the prediction error.

The most widely used methods for handling missing data fall into three main categories: i) Deletion methods (listwise deletion, i.e., complete-case analysis, pairwise deletion, i.e., available-case analysis), ii) Single Imputation Methods (mean/mode substitution, linear interpolation, Hot deck, and cold deck) and iii) Model-Based Methods (multiple imputation, k-nearest neighbors) \[12, 15, 16\]. In general, single imputation, multiple imputation and k-nearest neighbors are not recommended for the case of an extremely large proportion of the missing data \[15\]. These methods were not proposed and evaluated to deal with MPML. In addition, when the multivariate time series dataset has one univariate missing data pattern only in the target feature (the label), Semi-Supervised Learning (SSL) could be used. However, most of the SSL approaches have the performance affected by MPML \[15, 18\].

DL models need data to be trained, but when data is not available due to missing labels or the quality of the data is insufficient, then prior domain knowledge from experts can be incorporated to guide the learning process \[19, 17\]. Gaussian Process Regression (GPR) is one method that allows the use of domain knowledge from experts for modeling a univariate time series. The prior domain knowledge is expressed through the choice of the covariance function family and a set of initial hyper-parameters \[20, 21\]. In this paper, we present an initial study for handling MPML in multivariate multi-step time series forecasting. The approach is a two-step process to allow the integration of prior domain knowledge from experts. First, a set of \(n\) samples of the possible interpolation of the missing labels are generated by the GPR based on the domain knowledge. Second, the observed input sensor data and \(n\) interpolated labels from GPR are used to train the prediction model. The main contribution of this work is a novel approach for Handling Missing Labels based on domain knowledge from experts to sidestep the challenge of training DL models with MPML for MLTTSF. The approach was evaluated by developing a model to forecast the biomass of HEK293 cells in rAAV production. It was modeled using Long Short-Term Memory Encoder-Decoder Architecture and one real dataset obtained during rAAV production performed in 3L bioreactors. Our experimental results demonstrate the potential of the proposed approach through the experimental evaluation of the generated forecasts in a case that would be extremely difficult to train a DL model due to MPML. Furthermore, this approach is the first to be proposed and evaluated in a real case with massive missing values.

The remainder of this paper is organized as follows. In Section 2 we present the related work, and in Section 3 the DL for time series forecasting and GPR are introduced. The proposed approach is described in Section 4, followed by the detailed description of experimental evaluation in Section 5. In Section 6, the results regarding the forecasted biomass by the models developed with the proposed approach are provided. Finally, conclusions and discussion about future work are presented in Section 7.

2. Related Work

No one “perfect” method exists for filling in missing data. It is because there are many possible approaches to deal with missing data and they may lead to quite different errors \[22\]. In \[23\], the comparison of several methods was shown including Mode Imputation \[24\], Decision tree \[25\], and Class Center based Missing Values Imputation \[26\] for missing values estimation on multivariate dataset. They found that the Decision Tree can perform well when used in the
right situation as a missing values imputation method on a multivariate dataset. In [27], a novel model called forward and backward variable-sensitive LSTM (FBVS-LSTM) consisting of two decay mechanisms and some informative data was proposed. The model inputs are mainly the missing indicator, time intervals of missingness in both forward and backward direction, and the missing rate of each variable. They employed this information to address MNAR mechanism. In [28], the authors proposed a deep learning model based on Gated recurrent units (GRU), namely GRU-D, to effectively exploit two representations of informative missingness patterns (masking and time interval). However, their approach depends on robust assumptions suitable only for specific applications and does not consider the missing rate differences. All these approaches mentioned above were not proposed and evaluated to deal with a massive proportion of missing values (>95%) and do not allow for including the domain knowledge in the estimation of the missing data. Papers focusing on missing values handling evaluated their approaches using the datasets where the percentages of missing values ranged from 1% to 80% [29].

3. Background

3.1. Deep Learning for time series forecasting

The deep learning models often perform better for MLTTSF tasks compared to traditional methods when a labeled dataset $D_L$ is available with a large amount of data. The modern neural systems owe their success to the recent increase in data availability and computational capacity [10, 3] as well as their deep structure, which involves stacking multiple layers with a vast number of neurons [3]. These are powerful features for MLTTSF, particularly for problems with big data and complex-nonlinear dependencies [1], and to improve the prediction accuracy [9]. In addition, DL does not only improve the state of the art forecasting performance but also reduces the complexity of the forecasting pipeline and improving code maintainability [30].

A DL architecture that has showed promising results in MLTTSF is Long-Short Term Memory (LSTM) Encoder-Decoder [1, 31, 32, 33, 34, 35]. LSTM is based on Recurrent Neural Network (RNN), but it effectively overcomes the vanishing gradient issue in naively in RNNs models [36, 1, 37, 33]. The LSTM encoder-decoder architecture reads and generates a sequence of arbitrary lengths. In many real cases, we want to predict an output sequence given an input sequence of different lengths, without correspondence between each input and output. An Encoder-Decoder takes a sequence as input and generates the most probable next sequence as output. The architecture comprises two sub-models employing two LSTM networks called the encoder and decoder and an intermediate (encoder) vector connecting the encoder and decoder sub-models. More details can be found in [38, 1, 36].

3.2. Gaussian Process Regression

Gaussian Processes Regression (GPR) is a powerful tool for nonlinear regression that combines prior process and sampled training set to infer posteriori process as interpolation results. This is useful for estimating the missing data in the response variable.

Assume that the predictor variables $X = \{x_i\}_{i=1}^{N} \in \mathbb{R}^{N \times l}$ is an N-attribute time series each of which has l observations and response variables $y = \{y_i \in \mathbb{R}\}_{i=1}^{N}$. If the noises are assumed to be Gaussian and white, the measured records can be expressed as $y = f(x) + \epsilon$ where $f$ is a mapping function. Treating $f$ as a random function, we assume that the distribution over $f$ is a Gaussian process,

$$f(x) = \mathcal{GP}(m(x), k(x, x')),$$

where $x$ and $x'$ are indices that are analogous to the location indices of multidimensional Gaussian variables, and mean function $m(x)$ and the kernel (covariance) function $k(x, x')$ of a real process $f(x)$ are defined as
\[ m(x) = \mathbb{E}[f(x)], \] (2)

\[ k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]. \] (3)

The covariance function specifies the covariance between pairs of random variables and the specification of the covariance function implies a distribution over functions. Some of commonly-used covariance function are Squared Exponential Kernel, Periodic Kernel, Polynomial Kernel, Linear Kernel and Rational Quadratic Kernel. This last one was used in this paper and it is defined as follow:

\[ k_{RQ}(x, x') = \sigma_f^2 (1 + \frac{(x, x')}{\alpha l^2} - \alpha)\sigma_n^2 \] (4)

where \( l \) is length-scale, the \( \sigma_f^2 \) is the signal variance, the \( \sigma_n^2 \) is the noise variance. The parameter \( \alpha \) (scale mixture rate) determines the relative weighting of large-scale and small-scale variations. It controls how sharp or wide the kernel shape is and \( \delta_{ij} = 1 \) only when \( i = j \), otherwise \( \delta_{ij} = 0 \). The domain knowledge can be used to determine the value of the hyper-parameters set and the most representative hyper-parameters are \( l, \sigma_f^2 \) and \( \sigma_n^2 \). They specify the precise shape of the covariance function and can be viewed as specifying a distribution over function parameters, instead of being parameters which specify a function directly.

After defining the hyper-parameters and fitting the GPR model to the data, the predicted mean vector, \( \mu(X^*) \), and covariance matrix, \( \Sigma(X^*, X^*) \), can be calculated at unlabelled points \( \{X^* = x_{unlabel}, y^* = y_{mis}\} \), conditional on the observed data points \( \{X, y = y_{obs}\} \). This is done using the key predictive equations for Gaussian process regression

\[ \mu(X^*) = k(X^*, X)(k(X, X) + \sigma_n^2 I)^{-1}y \] (5)

\[ \Sigma(X^*, X^*) = k(X^*, X^*) - k(X^*, X)(k(X, X) + \sigma_n^2 I)^{-1}k(X, X^*). \] (6)

After computing the mean vector and covariance matrix (using equations 5 and 6) \( n \) samples can be obtained from the predictive equations according to the method described in [39] (section A.2). These samples are the estimation for the missing labels, \( y_{mis} \).

4. Handling MPML
The proposed approach is a two-step process that combining interpolation with the prediction model. The interpolation process has five steps, and it starts with an extremely unlabelled dataset with the missing labels interpolated by GPR.

4.1. Interpolation
Three conditions should be met before starting the interpolation procedure: 1) the MCAR assumption should be plausible, 2) labels represent the univariate time series with arbitrary missing data patterns, and 3) the prior domain knowledge should be available. Here the interpolation is performed by replacing missing labels with GPR estimation at unobserved points. This procedure is constituted of 5 steps:

- **Step 1** - Select all observed data with the pair input feature and output (label). For example, the input data that does not have the correspondent output data (label) should not be selected.
• **Step II** - Select kernel, mean function, and hyper-parameters of the GPR model based on the prior knowledge from experts. The kernel could be selected from expert knowledge about the possible characteristic of the feature (variable) with missing data. The other hyper-parameters should be selected based on the uncertainty regards the regions with missing data. Domain knowledge executes a vital role in enhancing the learning performance by helping to design better training data sets [40, 41]. Expert knowledge is a source of knowledge that is held by a particular group of experts [42]. Here, the expert knowledge should be used to get some information about the possible characteristic of the target feature (missing label), such as trends, patterns, constantness, smoothness, roughness variation, etc.

• **Step III** - Fit the GPR model to the observed data selected in Step I and hyper-parameters of Step II.

• **Step IV** - Calculate the predicted mean and covariance matrix of the missing label (unobserved points), conditional on the observed data (distribution of posterior) using the Equations 5 and 6.

• **Step V** - Select \( n \) samples (interpolations) from the posterior predictive distribution using the results of Step IV and the method described in [39] (section A.2).

### 4.2. Prediction Model

This procedure creates the model to perform the multivariate multi-step time series forecasting task based on DL architecture and in the multivariate dataset with the missing label replaced.

• **Step VI** - Combine the \( n \) possible interpolated labels generated by GPR with the input features of the unlabelled multivariate time-series dataset used in Step I. This step generates a new multivariate time-series dataset with the input features pair with \( n \) interpolated labels.

• **Step VII** - Convert the new multivariate time-series dataset generated in Step VI in a supervised time-series dataset. Time series forecasting can be framed as a supervised learning problem. Since, given a sequence of numbers for a multivariate time series dataset, it is possible to restructure the data to look like a supervised learning problem. It can be done by using previous time steps as input variables and using the next step as the output variable. More details could be viewed in [1].

• **Step VIII** - Select one DL Architecture for multivariate multi-step Time Series Forecasting task to be trained with the new multivariate time series dataset (from Step VII). Good options are Encoder-Decoder Model and Attention mechanism. After the model is trained and tested, this step requires analysis of prediction with domain specialist to check if the multivariate multi-step time series forecasting makes sense. If it does not make sense, step II should be considered again.

### 5. Experiment

Recombinant adeno-associated virus (rAAV) vectors have been preferred in a number of gene therapy applications, and to produce rAAV, transfecting HEK293 cells with three plasmids is the most widely used platform [43, 44]. During its production in bioreactor, critical quality attributes, like viable cell density (VCD), need to be monitored for quality assurance purpose. Conventional way of the cell density measurement is by taking samples from bioreactor and counting with a cell counter, which has a long sampling interval, usually per hours, and has a delay between samplings and test. It is considered the offline measurement. To overcome this, dielectric spectroscopy probe has been introduced to monitor the biomass, an indicator of VCD, online by reading the capacitance of the cell culture and estimate the total volume of the cells [43, 44]. The reading interval can be set as small as per second to provide a real time growth.
profile of the cells. However, when a certain probe is not available in some laboratory setups or when target some other critical quality attributes that are unable to be detected with current online sensory techniques, it is a real case where an extremely unlabelled multivariate time-series dataset can be generated with only the off-line measurement, which we explored to evaluate the proposed approach. Using two different models, an experiment was performed to evaluate the proposed approach to forecast the biomass of HEK293 cell culture one hour ahead based on the last one hour of the input features. The objective of this experiment is to try to evaluate the proposed approach comparing the performance of the prediction model generated (from the proposed approach using an extremely unlabelled dataset) with a model trained with a dataset containing only observed data (without missing labels) to check if they have similar performance to forecast the biomass. To achieve this objective, the LSTM Encoder-Decoder architecture was used to create the models and an aperiodic multivariate time-series dataset with three input features without missing labels was used as the ground truth allowing the comparison between the models through the prediction and generalization test. The code used to train the models can be downloaded from [https://github.com/CARG-uOttawa/handlingMPML](https://github.com/CARG-uOttawa/handlingMPML).

5.1. Dataset
The dataset used to train and test the LSTM Encoder-Decoder models was generated by the Viral Vectors and Vaccines Bioprocessing Group of McGill University. The dataset was obtained with the 3L bioreactor, and it has three different rAAV productions (runs). This dataset includes online measurements of the following parameters (or signals): i) biomass (cells/ml) (label), ii) Cumulative O2 - total oxygen supplied to bioreactor (ml), iii) Dissolved Oxygen (%) - oxygen concentration comparing to saturated oxygen in medium, and iv) Cumulative Dissolved Oxygen - dissolved oxygen integrated over time. The three oxygen-related parameters were selected because oxygen demanding of the system is a key indicator for cell growth. The runs were performed in different conditions of agitation speed (rpm) and pH. The runs 1 and 2 (Figure 1 plots a and b) were used to train the models to forecast the biomass and the run 3 (Figure 2.a.) to perform the generalization test. It has the objective of comparing the forecasted biomass (obtained with the models after being trained) with the observed biomass data (ground truth) of run 3. This dataset has data from a real AAV production and in the runs is possible to view the time of transfection. That is the process of delivering plasmid DNA into HEK293 cells that will produce vector particles.

5.2. Models
The two models were developed to forecast the biomass one hour ahead based on the last one hour of three input features (input of the model) using multivariate multi-step LSTM Encoder-Decoder architecture. The input features include: i) Cumulative O2 (ml) ii) Dissolved Oxygen (%), and iii) Cumulative Dissolved Oxygen. In the following will be described how the models were developed. Figure 1 summarize how the two models were trained.

- Model M1 - trained with all observed biomass values from the union of the run 1 (5378 values pairs) and run 2 (6143 values pairs), see the Figure 1 plots a, b and c.
- Model M2 - this model was trained using the proposed approach. Aiming to simulate the situation of use an extremely unlabelled dataset to train a DL model, five observed biomass values were selected from the run-1 and run-2 to represent offline measurement and used with input features (Cumulative O2 (ml), Dissolved Oxygen (%) and Cumulative Dissolved Oxygen). This resulted in two extremely unlabelled training sets with 99.9% of missing labels in both cases (run 1 with 5/5378 and run 2 with 5/6143) which is a MPML. In the Figure 1 plots d and f, we can see these five biomass data points (blue points) selected from the runs, and in plots e and g, the input features from the runs. After
selecting these five observed biomass values, the GPR interpolation was performed, and ten interpolation samples (for each run) were obtained, see red lines in plots d and f. The GPR was performed using Rational Quadratic Kernel, aiming to model the exponential phase of rAAV production. It represents the integration of the domain knowledge from the experts in the interpolation. In the literature, it is reported that the biomass values show an exponential behavior before the time of transfection. The ten obtained samples were combined with the observed input features of runs 1 and 2 (plots e and g), and this generated a unique training set (plot h) used to train the model M2.

All models have two encoders and two decoders as an LSTM hidden layer with 150 units. They were trained with Adam optimizer and Huber loss as the loss function with 50 epochs and batch size of 32.

6. Results

After the models (M1 and M2) were trained, we performed the prediction test of biomass with all runs 1, 2, and 3 (Figure 2 - a, b and c). The predict test with run-3 is the generalization test because all models do not have any information about the run-3. Runs 1 and 2 were used to train the model M1. However, the model M2 does not have all information about the observed biomass of run 1 and 2. Since the training process of M2 was performed with a training set
Figure 2. Results of experimental evaluation. The Prediction test of models M1 and M2 has the objective of forecasting the observed biomass of run-1, run 2 and run-3 (generalization test) plots (a, b and c). In d.1, d.2, d.3, e.1, e.2 and e.3 we have the results that were compared with the observed biomass (ground truth). The model M1 show the best performance in the prediction test and the model M2 showed the best performance in generalization test and a similar performance to M1 in the prediction test.
composed of interpolated biomass (using only five observed biomass values selected from runs 1 and 2, see Figure 1 and the input features of the runs 1 and 2. we can see the forecasting results of the biomass in the Figure plots d.1, d.2, d.3, e.1, e.2 and e.3.

The model M1 performed the prediction (plots d.2 and d.3) with the predicted biomass (red lines) very close to the observed data (blue lines), and it had the lowest Normalized Root Mean Square Error (NRMSE) values in the prediction test. These results were expected because it was trained with only observed data and had more information about the real behavior of the biomass data (the ground truth). It was able to approximate and reproduce the time of transfection in both cases. It is the moment that the biomass is increasing, decrease abruptly and come back to increase, see Figure plots d.2 and d.3. In the generalization test, the M1 showed the prediction farther from observed values.

The model M2 showed the results close to the observed data in the prediction test, but it could not reproduce the time of transfection, plots e.2 and e.3. This is because it did not have this information in the training set generated from the 5 biomass data points selected. However, the M2 showed the lowest NRMSE in the generalization test (NRMSE: 1.118 - plot e.1) compared to M1 (NRMSE: 1.191 - plot d.1).

7. Conclusion and Future works
In this paper, an approach for handling MPML for Multivariate Multi-Step Time Series Forecasting was proposed. The main contribution of this work is to enable training DL models in the case of an unlabelled dataset with a massive proportion of missing labels. So far, to the best of our knowledge, this is the first approach proposed and evaluated to deal with a proportion of missing labels higher than 95%. The viability of the proposed approach was demonstrated through an experimental evaluation performed with the development of conceptual soft-sensors with one real dataset. The experimental evaluation showed that it was possible to train one model (M2) from a dataset with MPML using the proposed approach and get results similar to the model (M1) trained with a dataset composed of only observed data. This approach is an initial study. The future works are: i) include uncertainty in the prediction, ii) evaluate the approach with a real extremely unlabelled multivariate time series dataset, and iii) developing DL-based ensemble model.

References
[1] Brownlee J 2018 Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python (Machine Learning Mastery)
[2] Torres J F, Hadjout D, Sebaa A, Martinez-Alvarez F and Troncoso A 2021 Deep learning for time series forecasting: A survey vol 9 (Mary Ann Liebert, Inc., publishers 140 Huguenot Street, 3rd Floor New . . . ) pp 3–21
[3] Mahmoud A and Mohammed A 2021 A survey on deep learning for time-series forecasting Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges (Springer) pp 365–392
[4] Sezer O B, Gudelek M U and Ozbayoglu A M 2020 Financial time series forecasting with deep learning: A systematic literature review: 2005–2019 vol 90 (Elsevier) p 106181
[5] Lara-Benítez P, Carranza-García M and Riquelme J C 2021 An experimental review on deep learning architectures for time series forecasting
[6] Mohammadi Farsani R and Paki R 2021 A transformer self-attention model for time series forecasting vol 9 (Shahid Rajaee Teacher Training University) pp 1–10
[7] Abbasimehr H and Paki R 2021 Improving time series forecasting using lstm and attention models (Springer) pp 1–19
[8] Thapa S, Zhao Z, Li B, Lu L, Fu D, Shi X, Tang B and Qi H 2020 Snowmelt-driven streamflow prediction using machine learning techniques (lstm, narx, gpr, and svr) vol 12 (Multidisciplinary Digital Publishing Institute) p 1734
[9] Wan R, Mei S, Wang J, Liu M and Yang F 2019 Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting vol 8 (Multidisciplinary Digital Publishing Institute) p 876
[10] Lim B and Zohren S 2021 Time-series forecasting with deep learning: a survey vol 379 (The Royal Society Publishing) p 20200290
[11] Kang H 2013 The prevention and handling of the missing data vol 64 (Korean Society of Anesthesiologists) p 402
[12] Salgado C M, Azevedo C, Proença H and Vieira S M 2016 Missing data (Springer) pp 143–162
[13] Van Buuren S 2018 Flexible imputation of missing data (CRC press)
[14] Leke C A and Marwala T 2019 Missing data estimation using invasive weed optimization algorithm Deep Learning and Missing Data in Engineering Systems (Springer) pp 115–128
[15] Jakobsen J C, Ghud C, Weterslev J and Winkel P 2017 When and how should multiple imputation be used for handling missing data in randomised clinical trials—a practical guide with flowcharts vol 17 (BioMed Central) pp 1–10
[16] Hyndman R J and Athanasopoulos G 2018 Forecasting: principles and practice (OTexts)
[17] Yang X, Song Z, King I and Xu Z 2021 A survey on deep semi-supervised learning
[18] Van Engelen J E and Hoos H H 2020 A survey on semi-supervised learning vol 109 (Springer) pp 373–440
[19] Dirks M 2018 Learning with prior domain knowledge and insufficient annotated data Canadian Conference on Artificial Intelligence (Springer) pp 360–363
[20] Azman K 2005 Incorporating prior knowledge into gaussian process models Proceedings of 6th International PhD Workshop on Systems and Control—A Young Generation Viewpoint, Volume A pp 253–256
[21] Roberts S, Osborne M, Elden M, Reece S, Gibson N and Aigrain S 2013 Gaussian processes for time-series modelling vol 371 (The Royal Society Publishing) p 20110550
[22] Ipsen N, Mattei P A and Frellsen J 2020 How to deal with missing data in supervised deep learning? ICML Workshop on the Art of Learning with Missing Values (Artemiss)
[23] Pristyanto Y and Pratama I 2019 Missing values estimation on multivariate dataset: Comparison of three type methods approach 2019 International Conference on Information and Communications Technology (ICOICT) (IEEE) pp 342–347
[24] Little R J and Rubin D B 2019 Statistical analysis with missing data vol 793 (John Wiley & Sons)
[25] Han J, Kamber M and Pei J 2011 Data mining concepts and techniques third edition vol 5 pp 83–124
[26] Tsi C F, Li M L and Lin W C 2018 A class center based approach for missing value imputation vol 151 (Elsevier) pp 124–135
[27] Fouladgar N and Fränling K 2020 A novel lstm for multivariate time series with massive missingness vol 20 (Multidisciplinary Digital Publishing Institute) p 2832
[28] Che Z, Purushotham S, Cho K, Sontag D and Liu Y 2018 Recurrent neural networks for multivariate time series with missing values vol 8 (Nature Publishing Group) pp 1–12
[29] Pratama I, Pernanasar A E, Ardiyanto I and Indrayani R 2016 A review of missing values handling methods on time-series data 2016 International Conference on Information Systems and Innovation (ICITSI) (IEEE) pp 1–6
[30] Faloutsos C, Gasthaus J, Januschowski T and Wang Y 2018 Forecasting big time series: old and new vol 11 (VLDB Endowment) pp 2102–2105
[31] Du S, Li T and Horng S J 2018 Time series forecasting using sequence-to-sequence deep learning framework 2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP) (IEEE) pp 171–176
[32] Liu F, Lu Y and Cai M 2020 A hybrid method with adaptive sub-series clustering and attention-based stacked residual lstms for multivariate time series forecasting vol 8 (IEEE) pp 62423–62438
[33] Du S, Li T, Yang Y, Gong X and Horng S J 2019 An lstm based encoder-decoder model for multistep traffic flow prediction 2019 International Joint Conference on Neural Networks (IJCNN) (IEEE) pp 1–8
[34] Ismail N H, Du M, Martinez D and He Z 2019 Multivariate multi-step deep learning time series approach in forecasting parkinson’s disease future severity progression Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics pp 383–389
[35] Du S, Li T, Yang Y and Horng S J 2020 Multivariate time series forecasting via attention-based encoder–decoder framework vol 388 (Elsevier) pp 269–279
[36] Park S H, Kim B, Kang C M, Chung C C and Choi J W 2018 Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture 2018 IEEE Intelligent Vehicles Symposium (IV) (IEEE) pp 1672–1678
[37] Petneházi G 2019 Recurrent neural networks for time series forecasting
[38] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H and Bengio Y 2014 Learning phrase representations using rnn encoder-decoder for statistical machine translation
[39] Rasmussen C E and Williams C K I 2006 Gaussian processes for machine learning. Adaptive computation and machine learning (MIT Press) ISBN 026218253X
[40] Deng C, Ji X, Rainey C, Zhang J and Lu W 2020 Integrating machine learning with human knowledge
[41] Brownlee J 2020 *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python* (Machine Learning Mastery)

[42] von Rueden L, Mayer S, Beckh K, Georgiev B, Giesselbach S, Heese R, Kirsch B, Pfrommer J, Pick A, Ramamurthy R *et al.* 2019 *Informed machine learning–a taxonomy and survey of integrating knowledge into learning systems*

[43] Naso M F, Tomkowicz B, Perry W L and Strohl W R 2017 *BioDrugs* **31** 317–334

[44] Petiot E, Ansorge S, Rosa-Calatrava M and Kamen A 2017 *Journal of biotechnology* **242** 19–29