ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2.
Available from http://www.i6doc.com/en/.

LSTM encoder-decoder model for contextualized time series forecasting applied to the simulation of a digital patient's physiological variables.

J. Paris[1], C. Sinoquet[2], F. Taia-Alaoui[3] and C. Lejus-Bourdeau[4]

1 - CR2TI / UMR 1064 Inserm, Nantes University Hospital, France
2 - LS2N, UMR CNRS 6004, Nantes University, France
3 - GRICAD, Grenoble, France
4 - LE SiMU / Nantes University Hospital, Nantes University, France

Abstract. This paper explores utilizing an encoder-decoder neural architecture for unsupervised representation learning of mixed asynchronous data, presenting the JMETTS (Joint Modelling of Event Traces and Time Series) model. Our goal is to forecast short-term multivariate time series within event contexts. As a proof of concept, we examine a real-world case in digitally assisted training for anaesthesiology. JMETTS demonstrates high predictive performance, with a maximum prediction error percentage of approximately 5.5%, comparable to that of its only competitor published to date. The source code can be found at `https://github.com/jp3142/jmetts_models_and_pipeline`.

1 Introduction

Time series data are ubiquitous thanks to widely deployed sensors. This has prompted a notable increase in time series forecasting research, particularly for multivariate time series (MTS). However, events occurring in the studied dynamical systems or in their surrounding contexts often determine the evolution of the MTS variables observed. Nowadays, event traces (ETs) are collected along with MTSs in an increasing number of systems. This opens unparalleled prospects to explore new frameworks for event-contextualized MTS forecasting.

Over the last decade, there has been a surge in research on joint modelling of time series and *time-to-event data* in the very specific domain of survival analysis. Surpringly, despite the opportunities that technological sensor capabilities open up, and despite the considerable research attention given to MTS forecasting, we did not find any work in the realm of machine or deep learning, specifically targeting event-contextualized MTS prediction [1]. The autoregressive Markov chain-based solution of [5] is an exception to this rule. The present work introduces the JMETTS (Joint Modelling of Event Traces and Time Series) model, and thus investigates the viability of using a variant of the RNN (Recurrent Neural Network) encoder-decoder model for simulating physiological variables in a digital patient undergoing anaesthesia, in response to medical actions.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

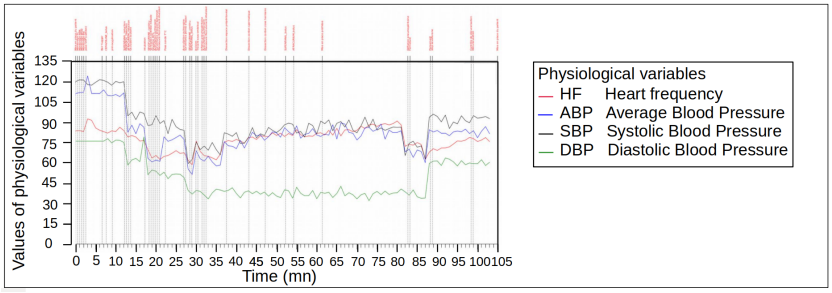

Fig. 1: An anaesthetic profile. Events are shown in red, in the top section.

2 Preliminaries

2.1 Mixed asynchronous data

The data available for our modelling problem consist of "anaesthetic profiles" $\{Z, E\}$, where $Z = [z_1,\ z_2,\ \cdots z_{\ell_Z}]$ is an MTS of dimension d_v and length ℓ_Z. The observations in Z are regularly sampled. We denote by $\tau_t \in \mathbb{R}^+$ the (continuous) timestamp of observation z_t at discrete time t. We assert that an history $E = [(e_1, t_1),\ \cdots (e_{\ell_E}, t_{\ell_E}]$, built on a finite set of n_e event categories $\mathscr{C}$, may impact the behavior of Z. Given t and t' $(1 \leq t < t' \leq \ell_Z)$, we denote by $Z_t^{t'}$ the vector $[z_t,\ \cdots z_{t'}]$ and by $E_t^{t'}$ the subsequence of E whose events e_i verify $\tau_t \leq t_i \leq \tau_{t'}$. Figure 1 displays such an anaesthetic profile.

2.2 Task definition

In our application, we will update the evolution of an MTS each time a medical action occurs. Hence, our problem focuses on short-term prediction. Given a prediction horizon h, a parameter k defining the size of the past from which the prediction is made, and an anaesthetic profile $\{Z, E\}$, we wish to design a function to model the probability distribution $p(\hat{Z}_{t+1}^{t+h} | Z_{t-k+1}^{t},\ E_{t-k+1}^{t})$.

3 The JMETTS model

In many applications, we need to predict an output sequence Y as a complex function of an entire input sequence X. To achieve this goal, the RNN encoder-decoder (ED) architecture, initially proposed for natural language processing [2, 3], relies on two separate components: the encoder's role is to "encode" a variable-length source sequence X of size ℓ_X into a fixed-dimensional representation denoted as the context vector $C(X)$; then, the decoder initiates the "decoding" of $C(X)$ into the variable-length predicted sequence $\hat{Y}$, of size ℓ_Y (usually different from ℓ_X).

3.1 From mixed asynchronous data to merged data

The data fed to the JMETTS model is obtained by preprocessing anaesthetic profiles $\{Z, E\}$, following 4 steps. **Normalization** first rescales the d_v variables of Z to $[0, 1]$, using Min-Max scaling. We obtain Z^{norm}. Secondly, **synchronization** produces E^{sync}, a univariate categorical time series of same length as Z; it is built on $\mathscr{C} \cup \{\epsilon\}$, where ϵ denotes the "No Event" category. We assume that at

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

most one event can occur in time interval $[\tau_t, \tau_{t+1}[$. If it exists, this single event is aligned with t; otherwise, ϵ is aligned with t. Thirdly, **one-hot encoding** maps E^{sync} to the numerical MTS of dimension $n_e + 1$, denoted as E^{enco}, where each element is a vector filled with zeroes, except a 1 at the position specific to an event category. Finally, we move a sliding window of size $k + h$ (see Section 2.2) along Z^{norm} and E^{enco} in parallel, **to create the training pairs** $\{X, Y\}$:

$$X = \Lambda_{t-k+1}^{t}, \quad Y = \Lambda_{t+1}^{t+h}, \quad k \leq t \leq \ell_Z - h,$$
$$\text{with } \lambda_i = [z_{i,1}^{norm}, \cdots, z_{i,d_v}^{norm}, e_{i,1}^{enco}, \cdots, e_{i,n_e+1}^{enco}], \quad \lambda_i \in [0,1]^{d_v+n_e+1}.$$

This stated, we will simply refer to $X = [x_1, x_2, \cdots, x_k]$ and $Y = [y_1, y_2, \cdots, y_h]$.

3.2 JMETTS workflow

Both the encoder and decoder in JMETTS are stacked LSTMs where Long Short Term Memory units recurrently update two hidden states, c_t and h_t. In a stacked LSTM, each layer is fully connected to its adjacent layer: all the LSTM units in the same layer process the same inputs in parallel at each time step t.

During the training process of the ED model, the source sequence is the encoder's input sequence $X \in [0,1]^{k\times(d_v+n_e+1)}$, whereas the target sequence corresponds to the ground truth sequence $Y \in [0,1]^{h\times(d_v+n_e+1)}$ that we aim to predict ($\hat{Y}$). The encoder and the decoder are trained jointly to maximize the conditional probability $p(\hat{Y} \mid X)$.

Figure 2 shows how the JMETTS model operates, to generate $\hat{Y}$ conditionally on X. In the decoder, all LSTM units in first layer have their initial states ($h_{d,1}^{(0)}$ and $c_{d,1}^{(0)} \in \mathbb{R}^{d_h}$) initialized using the final states ($h_{e,2}^{(k)}$ and $c_{e,2}^{(k)}$) of encoder's last layer (shown with red arrows). In the learning and inference tasks, the information generated by the encoder's last layer at final time step k, that is the $y_{e,2}^{(k)}$ outputs, defines the context $C(X) \in \mathbb{R}^{d_h}$. The decoder waits for the encoder to process the entire sequence X_1^k before it starts decoding it using context vector $C(X)$. This summarized information about X is duplicated as many times as there are time steps in the decoder (h). Hence, at each time step in the decoder, the input of each of the LSTM units in the first layer will be fed with the context vector (shown with blue arrows). This represents an alternative approach to the data flow management described in the pioneering ED models [3, 2] where the decoder operates autoregressively using $\hat{y}_{t-1}$ as input (along with $C(X)$), to generate $\hat{y}_t$. The predicted sequence is $y_{d,2}^{(1)}, y_{d,2}^{(2)} \cdots y_{d,2}^{(h)}$.

3.3 Loss function

We devised a loss function that combines standard mean squared error (MSE) and standard categorical cross-entropy (CCE): $\Psi = \nu\ \text{MSE} + \text{CCE}$, where ν is used to take account of the difference in scale between MSE and CCE.

4 Experimental study

4.1 Experimental settings

Anaesthesia dataset Our dataset describes laparoscopic inguinal hernia surgeries for 1,000 men around thirty, without any prior medical history. This surgery

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

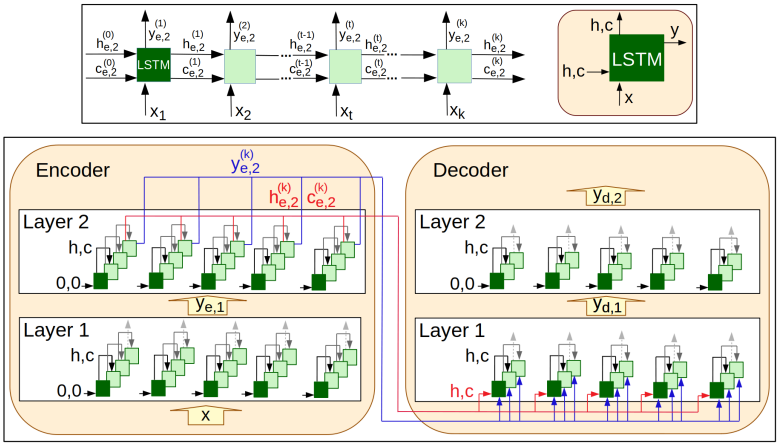

Fig. 2: Operational framework of the JMETTS model. Top section: LSTM unit (dark green) unfolded in time (light green). Bottom section: JMETTS model with 2 recurrent layers in both encoder and decoder. $k = 4$, $h = 3$, $d_h = 5$.

involves 35 possible medical acts. We considered 4 physiological variables: heart frequency (HF); systolic, average and diastolic blood pressures (SBP, ABP, DBP).

Preliminary analysis We wished to gather insights to guide the choice of architecture and hyperparameter setting for further analysis. We combined the 18 architectures described in Table 1 with 24 hyperparameter adjustments. We varied 4 hyperparameters: gradient descent optimization algorithm ∈ {Adam, Nadam, Adamax}; network weight initialization ∈ {Glorot Uniform, Glorot Normal}; batch size ∈ {32,64} and learning rate ∈ $\{10^{-4}, 10^{-3}\}$. We kept the values of 4 other hyperparameters constant: $k = 10$, $h = 10$; number of epochs: 40 and Dropout / Recurrent dropout: None. We drew a subset of 200 patients from the initial dataset and divided it into 5 equal folds. We then applied a 5-fold cross-validation scheme to each of the 432 model instances analyzed. For each instance, we computed the prediction performance using criterion Ψ (Section 3.3), averaging results across the 20 models trained within the cross-validation scheme. The hyperparameter ν (Section 3.3) was empirically calibrated to 100.

Refined analysis We examined 15 hyperparameter adjustements: learning rate ∈ $\{10^{-4}, 2.5 \times 10^{-4}, 5 \times 10^{-4}, 7.5 \times 10^{-4}, 10^{-3}\}$ and $k \in \{10, 20, 30\}$. We kept the values of the 6 other hyperparameters constant: optimization: Adamax;

$E_m D_n$	Numbers of nodes in encoder (in bold characters) and decoder layers		
E_2D_2	[**440**,**440**,440,440]	[**440**,**220**,220,440]	[**880**,**440**,440,440]
E_3D_2	[**440**,**440**,**440**,440,440]	[**440**,**220**,**110**,110,440]	[**880**,**440**,**220**,220,440]
E_3D_3	[**440**,**440**,**440**,440,440,440]	[**440**,**220**,**110**,110,220,440]	[**880**,**440**,**220**,220,440,440]
E_4D_2	[**440**,**440**,**440**,**440**,440,440,440]	[**440**,**220**,**110**,**55**,55,440]	[**880**,**440**,**220**,**110**,110,440]
E_4D_3	[**440**,**440**,**440**,**440**,440,440,440]	[**440**,**220**,**110**,**55**,55,220,440]	[**880**,**440**,**220**,**110**,110,220,440]
E_4D_4	[**440**,**440**,**440**,**440**,440,440,440,440]	[**440**,**220**,**110**,**55**,55,220,440]	[**880**,**440**,**220**,**110**,110,220,440]

Table 1: The 18 architectures considered in the preliminary study. m: number of layers in encoder. n: number of layers in decoder.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

weight initialization: Glorot Uniform; batch size: 32; $h = 10$; number of epochs: 40 and Dropout / Recurrent dropout: None. We used the same cross-validation scheme as previously, starting this time from the dataset of 1,000 patients.

Comparison with baselines Our first baseline is NHMC-AR [5]. Alternatively, the PHMC-MLAR model [4] can be used to suit our objective, if we map its partially observed states to events [5]. We used the standard Mean Absolute Percentage Error (MAPE) to compare the models' MTS predictive performances.

We calculated an average MAPE for the JMETTS model, using the 5-fold cross-validation scheme applied to the dataset of 1,000 patients. The MAPE values relative to PHMC-MLAR and NHMC-AR were obtained from prior experiments [5] performed on the anaesthesia dataset partitioned into training, validation, and test sets, comprising 500, 200, and 300 patients, respectively. Model preselection based on BIC score was applied to the training set, and model selection based on MAPE criterion was performed on the validation set.

4.2 Results

Impact of architecture and four hyperparameters On the anaesthesia dataset, increasing the number of layers decreases the predictive performance of JMETTS. In constrast, we found no discernible impact of the number of nodes on the performance of JMETTS. On the other hand, we observed that the six best $\mathrm{E}_m\mathrm{D}_n$ model instances share the same hyperparameter setting by an overwhelming majority. These results recommend choosing the simplest configuration for the refined study, that is the parsimonious architecture E_2D_2, with *440* LSTM *units in each layer*, *Adamax optimization*, *uniform Glorot initialization*, and a *batch size of 32*. This instance is referred to as JMETTS* hereafter.

Impact of hyperparameter k and learning rate Increasing the value of k clearly enhances the predictive performance of JMETTS* by a ratio of at least 1.5 between $k = 10$ and $k = 30$. Among the 15 hyperparameter settings examined, the performances with $k = 30$ ranked 1st to 4th and 9th, while those obtained with $k = 20$ ranked 5th to 8th and 10th. In contrast, fine-tuning the learning rate is not influential. Finally, we have retained the model trained with $k = 30$ and a *learning rate of* 10^{-3}, leaving other hyperparameters unchanged, as the best JMETTS model for comparison with the baselines.

Comparative analysis The two hyperparameters of the NHMC-AR and PHMC-MLAR models are the number of states q in the Markov process and the autoregressive order r in the time series. NHMC-AR($q = 4$, $r = 5$) and PHMC-MLAR($q = 6$, $r = 2$) obtain the lowest MAPE values on the anaesthesia dataset.

Table 2 shows that HF is the best predicted variable by all 3 models. There is a noticeable discrepancy between JMETTS and NHMC-AR, on one side, and PHMC-MLAR, on the other. The MAPE values reach up to around 7 to 9%, and even reach 11% (DBP) for PHMC-MLAR. In contrast, JMETTS and NHMC-AR exhibit high and comparable performances, which remain consistently stable across all horizons for both models, predominantly falling between 4.5 and 5.5%. Moreover, the performance of JMETTS is consistently slightly higher than that

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Variable	Model	Prediction horizons									
		1	2	3	4	5	6	7	8	9	10
HF	PHMC-MLAR	3.5	4.6	5.0	5.5	6.0	6.1	6.2	6.5	6.7	6.9
	NHMC-AR	3.6	3.8	4.2	4.5	4.6	4.7	4.9	5.0	5.1	5.1
	JMETTS	2.4	2.7	2.9	3.1	3.3	3.4	3.6	3.8	3.9	4.1
SBP	PHMC-MLAR	5.1	6.6	7.4	7.9	8.1	8.3	8.5	8.8	8.8	8.8
	NHMC-AR	4.4	4.9	5.1	5.2	5.3	5.4	5.5	5.6	5.7	5.8
	JMETTS	3.7	4.0	4.3	4.6	4.8	4.9	5.1	5.3	5.4	5.6
ABP	PHMC-MLAR	6.3	7.9	8.4	8.9	8.9	9.0	9.1	9.1	8.9	8.2
	NHMC-AR	4.9	5.4	5.6	5.7	5.7	5.6	5.6	5.4	5.1	5.0
	JMETTS	3.7	4.2	4.6	4.9	5.1	5.3	5.5	5.7	6.0	6.2
DBP	PHMC-MLAR	5.5	7.0	8.1	8.3	8.4	9.4	9.8	11.1	11.1	11.0
	NHMC-AR	4.4	4.8	5.0	5.0	5.0	5.0	4.7	5.1	4.7	4.6
	JMETTS	4.3	4.5	4.7	4.8	4.9	4.9	4.9	5.0	5.0	5.0

Table 2: Comparison of the MAPE values of JMETTS, NHMC-AR and PHMC-MLAR obtained for the anaesthesia dataset. MAPE values are expressed in percentages.

of NHMC-AR (except for ABP at horizons 8 to 10), and the difference is relatively marked for HF, across all horizons (ranging from 1.0 to 1.4%).

The positive outcomes from JMETTS allow us to validate both the dataflow management specified in our ED framework and the design of the composite loss tailored for multimodal data, on the anaesthesia dataset. One limitation may arise from the need to calibrate the hyperparameter ν associated with this loss function. Further investigations are required to relax the restriction of observing at most one event occurrence between two consecutive time steps.

5 Conclusion and future work

The proof-of-concept study outlined in this paper demonstrates the applicability of JMETTS for time series prediction dedicated to data-driven simulation of a digital patient's vital signs. Future work will explore whether integrating attention mechanisms into JMETTS, inspired by transformer neural networks, can sustain good predictive performance as the number of variables and events increases. We will assess JMETTS and its future enhancements across various surgeries.

References

[1] K. Benidis, S. Rangapuram, V. Flunkert *et al.*, Deep learning for time series forecasting: tutorial and literature survey, *ACM Computing Surveys*, 55(6):121:1–121:36, 2023.

[2] I. Sutskever, O. Vinyals, and Q. Le. Sequence to sequence learning with neural networks, In *proceedings of Advances in Neural Information Processing Systems 27: 28th Annual Conference on Neural Information Processing Systems* (NeurIPS), pages 3104–3112, 2014.

[3] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau *et al.*, Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP), pages 1724–1734, 2014.

[4] F. Dama and C. Sinoquet, Partially Hidden Markov Chain Multivariate Linear Autoregressive model: inference and forecasting - application to machine health prognostics, *Machine Learning*, 112(1):45–97, 2023.

[5] F. Dama, C. Sinoquet, and C. Lejus-Bourdeau, A hidden Markov model with Hawkes process-derived contextual variables to improve time series prediction. Case study in medical simulation. In *proceedings of the 31st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (ESANN), pages 519–524, 2023.