ECOLE

Experience-based Computation:
Learning to Optimise

Project Number: 766186
Project Acronym: ECOLE
Project title: Experienced-based Computation: Learning to Optimize

Deliverable D3.2
Deep structured learning and model space learning for Engineering and ICT data

Authors:
Sibghat Ullah, Thomas Bäck – Universiteit Leiden
Stephen Friess, Xin Yao – University of Birmingham
Zhao Xu – NEC Laboratories Europe

Project Coordinator: Professor Xin Yao, University of Birmingham
Beneficiaries: Universiteit Leiden, Honda Research Institute Europe, NEC Laboratories Europe

H2020 MSCA-ITN
Date of the report: 31.03.2021

Contents

Executive Summary

This document provides a concise report on the research invested and the scientific contributions made regarding the work package 3.2 in ECOLE. This work package deals with the issue of big data analytics through deep structured models and representation learning in the context of Engineering & ICT applications. In ECOLE, the idea of learning high-level representation with the help of supplementary domain information was proposed. A systematic empirical investigation was conducted to validate the theory. The findings suggest that the deep structured models trained with the help of supplementary domain information were 1.73 % better on average than the state-of-the-art models for multi-step-ahead time series forecasting.

Major Achievements

Major scientific achievements regarding the work package 3.2 are presented. In particular, short answers to some of the most important research questions are described:

Research Questions	Discussion
Can supplementary domain information be utilized to find useful representation in the data?	Our findings indicate the practical applicability of supplementary domain information for learning high-level abstractions in the temporal ICT data in the health sector. (cf. Fig. 4).
How effective are the high-level representations for better modeling of the temporal data?	Models trained with the help of these representations (which are learned through supplementary domain information) are 1.73 % better on average than the state-of-the-art deep generative models (cf. Fig. 5 and Table III).
Can the results be generalized to other domain-rich application, e.g., financial and economic signal-processing?	Similar results can be expected. The reason for this is that supplementary domain information links the temporal data in the form of a graph, e.g., patients with similar disease diagnostic are likely to have similar temporal features in their ICU stay, thereby connected to each other. Then, the temporal features of one patient can be utilized to help predict the temporal features of similar patient(s) (cf. Tables IV and Fig. 5).

1. Introduction

ECOLE aims at shortening the product-development cycle, reducing the resource consumption during the complete process, and creating more balanced and innovative products. One of the most important challenges ECOLE undertakes is *Mining & Learning Temporal Data* in the context of Engineering & ICT Applications. To model the temporal data in these domains, ECOLE proposes to learn high-level abstractions in data which can provide more robust and compact representations than the data space. This is intuitive for two reasons:

1) Temporal data-sets in these domains are marked by irregular, highly-sporadic and strongly-complex structures, and are consequently difficult to model by traditional state-space models. Therefore, learning high-level abstractions in data can provide more robust representations which can be employed for downstream learning tasks, e.g., forecasting, classification.
2) Temporal data-sets in these domains are characterized by loads of supplementary domain information. This supplementary information can be useful for learning high-level encodings and latent correlations in the data.

This report is to reflect the work and research invested in the work package 3.2 which embroils the issue of big data analytics in ICT and Engineering applications with the help of deep structured models and representation learning. In this report, we therefore summarize the following scientific findings and research outcomes pertaining to the work package 3.2:

- Processing of temporal data with the help of deep structured models such as Recurrent Neural Networks, Variational Autoencoders, and Variational Recurrent Neural Networks (Sec. 2). We concisely describe these models and their working mechanism.
- We provide a case study related to the ICT data in health sector. The case study augments the Variational Recurrent Neural Networks by learning high-level abstractions in the data with the help of supplementary domain information. We demonstrate that this augmentation improves the forecasting accuracy of temporal data on average by 1.73 %.

The following publication in the ECOLE is contributing to this report:

S. Ullah, Z. Xu, H. Wang, S. Menzel, B. Sendhoff and T. Bäck, " Exploring Clinical Time Series Forecasting with Meta-Features in Variational Recurrent Models," in International Joint Conference on Neural Networks (*IJCNN*), Glasgow, 2020, IEEE.

2. Processing of Temporal Data

In the past, processing of temporal data relied heavily on state-space models which were typically linear and were suited for univariate time series, although multivariate non-linear extensions of such models exist [1]. These methods required specifications of trends, seasonality, cyclical effects and shocks in time-series processing. As a result, these methods had higher interpretability. However, this interpretability (usually) came at the cost of the model accuracy since such models lacked the dynamic and complex nature of the multivariate time-series extracted from modern ubiquitous systems such as economic transaction processing systems and electronic health record (EHRs) systems. In this report, we discuss processing of temporal data in the context of deep structured models. This is since with the advent of deep learning [2], many methodologies have been proposed to employ deep structured models, e.g., RNNs, for time-series processing. Hybrid approaches [3] to combine state-space models and deep structured models have also been proposed. Note however that vanilla deep learning models have deterministic hidden states and lack the intrinsic stochasticity found in the latent variable models such as Hidden Markov Models (HMMs) and Kalman Filters. Recent studies [4] [5] have argued to incorporate some stochasticity in deep learning models while modeling complex sequences which can improve the generalization capability of these models.

On the other hand, variational autoencoders (VAEs) [6] have been proposed to capture high-variability in complex data-sets. VAEs are a class of deep latent-variable models which learn the complex intractable posterior over the data space by employing the variational inference (VI) and the reparameterization trick. However, vanilla VAEs are suited for non-sequential data-sets only. Recently in [5], the authors extend the variational autoencoders (VAEs) for highly-variable sequential data which is named variational recurrent neural network (VRNN).

A variational recurrent neural network (VRNN) contains a variational autoencoder (VAE) at each time-step t which is conditioned on the previous hidden state h_{t-1} of an RNN, thus, modelling the sequential structure in the data. In the same paper, the significance of this model is demonstrated on various sequential data-sets. VRNNs, however, have rarely been adopted for time series forecasting tasks. Recently in [7], the authors evaluate VRNNs for time series forecasting on various synthetic and one real benchmark data-sets against several neural baselines including recurrent neural network with extended Kalman filters (RNN-EKFs) [8], robust echo state state networks (RESNs) [9] and co-evolutionary multi-task learning (CMTL) [10], and conclude that VRNNs outperform all the baselines on most data-sets.

2.1. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [2] [11] [12] are a family of neural networks (NNs) which are specialized to model the temporal correlations in the data. In particular, a recurrent neural network (RNN) receives a variable-length input sequence $x = (x_1, x_2, x_3, \dots, x_T)$; which it processes by computing the so-called hidden state h_t as a function of the current input x_t at time t and the previous hidden state h_{t-1} as:

$$h_t = f(x_t, h_{t-1};\ \theta)\ , \qquad (1)$$

where f is a non-linear activation function, and θ is the associated set of parameters to be optimized. The gated implementations of f result in networks known as long short-term memory (LSTM) [13] and gated recurrent unit (GRU) [14] which regulate the flow of information and prevent issues known as vanishing and exploding gradient problems [15]. RNNs model sequences by parameterizing a factorization of the joint sequence probability distribution as a product of the conditional probabilities such that:

$$p(x_1, x_2, x_3, \dots, x_T) = \prod_{t=1}^{T} p(x_t \mid \{x_i\}_{1\leq i<t}), \tag{2}$$

and

$$p(x_t \mid \{x_i\}_{1\leq i<t}) = g(h_{t-1}\,;\,\tau) \tag{3}$$

In Eqs. (2) and (3), T corresponds to the sequence length, $\{x_i\}_{1\leq i<t}$ denotes the set of inputs preceding x_t and g is a function mapping the hidden state h_{t-1} to the output conditional probability distribution parameterized by a set of parameters τ. Note that due to the space limitation, $\{x_i\}_{1\leq i<t}$, i.e., the set of the inputs preceding x_t and similar notations, e.g., $\{x_i\}_{1\leq i<T}$, are substituted with their compact representations, i.e., $x_{<t}$ in the remainder of the report.

2.2. Variational Autoencoders

A variational autoencoder (VAE) [6] is a deep latent-variable model to approximate the complex intractable posterior over the data space. A VAE uses a set of latent variables z designed to capture the high-variations in the data by encoding and reconstructing the data; thereby, learning the global properties of the data-space. More specifically, a VAE consists of two neural networks: an inference network (the encoder), and a generative network (the decoder) respectively. The encoder encodes the input x to the latent variable z, and the decoder maps this latent variable z back to reproduce x. The VAE treats the conditional probability distribution $p(x|z)$ as highly-flexible function approximation of x. However, the mapping from z to x cannot be implemented because of the intractable posterior $p(z|x)$ on the latent variable. The VAE thus introduces the variational approximation $q(z|x)$ of the intractable posterior $p(z|x)$. The approximate posterior $q(z|x)$ has highly-flexible form and its parameters are generated by the inference, i.e., encoder network. Lastly, the variational approximation $q(z|x)$ of $p(z|x)$ enables the use of Evidence Lower Bound (ELBO) (variational lower bound) as:

$$log\, p(x) \geq -KL(q(z|x)||\, p(z)) + \mathbb{E}[\, log\, p(x|z)], \tag{4}$$

where the expectation $\mathbb{E}[log\, p(x|z)]$ is with respect to $z \sim q(z|x)$, and $KL(Q||P)$ is the Kullback-Leibler divergence [16] between two probability distributions Q and P. In [6], the variational posterior $q(z|x)$ is modelled by a Gaussian $\mathcal{N}\left(\mu, diag(\sigma^2)\right)$ where the parameters μ and σ are the outputs of the inference network and $diag$ corresponds to the diagonal covariance structure of the Gaussian distribution. The prior $p(z)$ is assumed to be a standard Gaussian distribution. The training process focuses on maximizing ELBO in Eq. (4) which yields the optimal parameters for the inference and generative networks. A low variance estimator can be substituted with the help of the reparameterization trick $z = \mu + \sigma \odot \epsilon$; where $\epsilon \sim \mathcal{N}(0, I)$ is a vector of standard Gaussian variables and $\odot$ denotes the element-wise product:

$$\mathbb{E}[log\, p(x|z)] = \mathbb{E}[log\, p(x|z) = \mu + \sigma \odot \epsilon], \qquad (5)$$

where the expectation $\mathbb{E}[log\, p(x|z)]$ on the left-hand side is with respect to $z \sim q(z|x)$, and the expectation on the right-hand side is with respect to $\epsilon \sim \mathcal{N}(0, I)$.

2.3. Variational Recurrent Neural Networks

A variational recurrent neural network (VRNN) [5] is the extension of a standard VAE discussed above to the cases with sequential data. It is a combination of an RNN and a VAE as described in Eqs. (1) and (5) respectively. More specifically, a VRNN employs a VAE at each time-step t. However, the prior on the latent variable z_t of this VAE is assumed to be a multivariate Gaussian whose parameters are computed from the previous hidden state h_{t-1} of the RNN such that:

$$z_t \sim \mathcal{N}\left(\mu_{0,t}, diag\left(\sigma_{0,t}^2\right)\right), [\mu_{0,t},\ \sigma_{0,t}] = \varphi_\tau^{\text{prior}}(h_{t-1}), \qquad (6)$$

In Eq. (6), $\mu_{0,t}$ and $\sigma_{0,t}$ are the parameters of the prior $p(z_t)$, and $\varphi_\tau^{\text{prior}}$ refers to a non-linear function such as a feed-forward neural network (FFNN) [2] [11] parameterized by a set of parameters τ. The generating distribution in the decoder $p(x|z)$ is conditioned on both z_t and h_{t-1} such that:

$$x_t|\ z_t \sim \mathcal{N}\left(\mu_{x,t}, \text{diag}\left(\sigma_{x,t}^2\right)\right), \qquad (7)$$

where $[\mu_{x,t}, \sigma_{x,t}] = \varphi_\tau^{\text{dec}}(\varphi_\tau^z(z_t),\ h_{t-1})$, and $\mu_{x,t}$ and $\sigma_{x,t}$ are the parameters of the generating distribution. The hidden state h_t of the RNN is updated as:

$$h_t = f(\varphi_\tau^x(x_t), \varphi_\tau^z(z_t), h_{t-1};\ \theta)\,, \qquad (8)$$

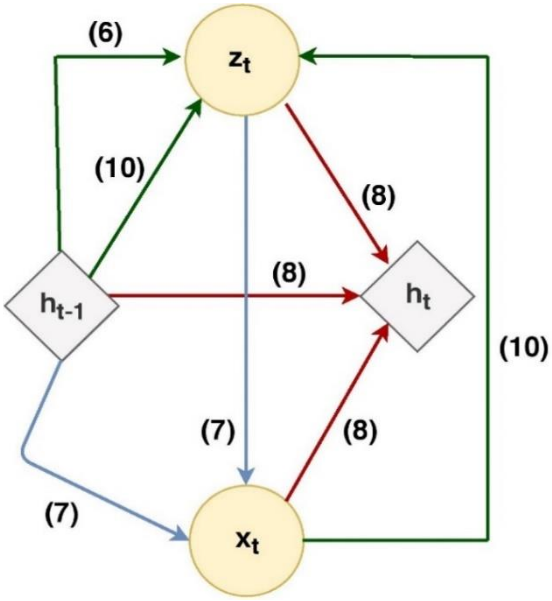

Figure 1. The schematic view of a VRNN. The green line connections correspond to the computations involving the (conditional) prior and posterior on z_t while the blue line connections show the computations involving the generative network, i.e., decoder. In addition, the computations for h_t are shown with red line connections. These connections depict the dependencies between the variables in Eqs. (6) – (10). Note that each connection/line is labelled according to the numbering of the equation it realizes.

where f is a non-linear activation function and φ_τ^x, φ_τ^z and $\varphi_\tau^{\text{dec}}$ in Eqs. (6) and (7) are the FFNNs similar to $\varphi_\tau^{\text{prior}}$. The hidden state h_t is a function of both $x_{\leq t}$ and $z_{\leq t}$. The joint probability distribution of x and z thus becomes:

$$p(x_{\leq T}, z_{\leq T}) = \prod_{t=1}^{T} p(x_t | z_{\leq t}, x_{<t}) p(z_t | x_{<t}, z_{<t}). \qquad (9)$$

We now discuss the inference, i.e., encoder network. Here, the approximate posterior $q(z_t | x_t)$ is a function of both x_t and h_{t-1} such as:

$$z_t |\, x_t \sim \mathcal{N}\left(\mu_{z,t}, \text{diag}\left(\sigma_{z,t}^2\right)\right), \qquad (10)$$

where $[\mu_{z,t}, \sigma_{z,t}] = \varphi_\tau^{\text{enc}}(\varphi_\tau^x(x_t),\ h_{t-1})$, and $\mu_{z,t}$ and $\sigma_{z,t}$ are the parameters of the approximate posterior and $\varphi_\tau^{\text{enc}}$ is a FFNN same as $\varphi_\tau^{\text{prior}}$, φ_τ^x, φ_τ^z and $\varphi_\tau^{\text{dec}}$. Conditioning on h_{t-1}, the posterior follows the factorization:

$$q(z_{\le T}|x_{\le T}) = \prod_{t=1}^{T} q(z_t|x_{\le t}, z_{<t}). \qquad (11)$$

The objective function to train both, inference and generative networks is to maximize ELBO based on the factorization in Eqs. (9) and (11); giving rise to the accumulative ELBO as:

$$\mathbb{E}\,[\sum_{t=1}^{T}(-KL(q(z_t|x_{\le t}, z_{<t})||p(z_t|x_{<t}, z_{<t}) + log\ p(x_t|z_{\le t}, x_{<t}))] \qquad (12)$$

where the expectation is with respect to $z_{\le T} \sim q(z_{\le T}|x_{\le T})$. The graphical representation of a standard VRNN is presented in Fig. 1.

3. Time Series Forecasting for Clinical Monitoring System

Due to great potential in improving the quality of healthcare service, the information and communication technologies (ICT) are widely used in the health sector. The clinical monitoring systems help the hospitals and doctors efficiently monitor health status of patients for a proactive, timely and effective diagnosis and treatment. Clinical time series generated from the monitoring systems are known for irregular, highly-sporadic and strongly-complex structures and are consequently difficult to model by traditional state-space models [1]. In this section of the report, we share some of the most important details from ECOLE [17], which tested the feasibility of learning high-level abstractions/meta-features from supplementary domain information in clinical settings. In ECOLE, we investigated the potential of applying VRNNs for forecasting clinical time-series extracted from electronic health records (EHRs) of patients. We have already established in Section 2 that VRNNs combine RNNs and VI, and are state-of-the-art methods to model highly-variable sequential data such as text, speech, time-series and multimedia signals in a generative fashion. In ECOLE, we proposed to incorporate multiple correlated time-series to improve the generalization capability, i.e., forecasting, of VRNNs. The selection of those correlated time-series was based on the similarity of the supplementary medical information, e.g., disease diagnostics, ethnicity and age, between the patients. We evaluated the effectiveness of utilizing such supplementary information with root mean square error (RMSE), on clinical benchmark data-set "Medical Information Mart for Intensive Care" (MIMIC III) [18] for multi-step-ahead prediction. We further performed subjective analysis to highlight the effects of the similarity of the supplementary medical information on individual temporal features, e.g., Systolic Blood Pressure (SBP), Heart Rate (HR), of the patients from the same data-set. Our results clearly showed that incorporating the correlated time-series based on the supplementary medical information could help improving the accuracy of the VRNNs for clinical time-series forecasting. Our conclusion was in line with the intuition of learning high-level abstractions/meta-features, and utilizing them for more robust training in deep structural models, and forms the mainstay of our research in work package 3.2. In the following of this section, we provide a sound theoretical understanding alongside practical details of our case study in ECOLE.

3.1. Medical Information Mart for Intensive Care (MIMIC)

Health care is one of the most exciting and challenging areas of information and communication technologies. The use of EHRs can significantly improve the existing health care systems since it can help identify early triage and risk evaluations [19] [20] [21] [22] for certain group of patients at very early stages of treatment. Most of the existing EHRs capture the temporal features for patients during their Intensive Care Unit (ICU) stay. Examples of such features include Heart Rate (HR), Oxygen Saturation Level (OSL), Body Temperature (BT) and Mean Blood Pressure (MBP) [23]. The set of these temporal signals can be used for further useful analysis such as phenotype classification [24] [25] [26], length-of-stay prediction [27] [28], risk-of-mortality prediction [27] and forecasting such signals for future time-steps. Unfortunately, these multivariate time-series are characterized by highly-irregular [29], sporadic and complex structures, and are consequently difficult to model by traditional methods. Note that irregular and sporadic multivariate time-series in this context refers to a time-series where the time intervals are not uniform and only a small subset of temporal features is observed at each time-step. Note also that the terms "temporal signals", "temporal features", "medical signals" and "time-series" have been used interchangeably throughout this report to refer to the same concept, i.e., time indexed variables of an individual patient which are observed in the ICU. In addition, terms “supplementary domain information”, “supplementary medical information”, “extra domain information” and “extra medical information” have also been used interchangeably throughout this report to refer to the non-temporal subjective information about the patients, which is observed when the patient is admitted to the hospital or ICU.

In ECOLE [17], we used “Medical Information Mart for Intensive Care” (MIMIC III) [18], which is publicly available and widely used benchmark data-set collected with a clinical monitoring system. MIMIC III is maintained in a relational database containing information of approximately 60,000 ICU admissions. It contains information about the demographics of the patients [27] [30], the laboratory tests, keynote events during the ICU stay, medications and the temporal signals in the ICU, e.g., MBP and BT. Since MIMIC III is a highly-complicated data-set involving millions of events; it is important to follow a standard approach to preprocess the data which can be used for the downstream learning tasks. To this end, we follow the procedure of [27] which provides the benchmark preprocessing for MIMIC III.

After following [27] for preprocessing; we are left with five different data-sets extracted from MIMIC III where each data-set corresponds to a specific learning task in [27] such as in-hospital-mortality-prediction, decompensation-prediction, length-of-stay-prediction, phenotype classification and multitask learning. In ECOLE [17], we proceeded with the in-hospital-mortality data-set extracted from MIMIC III since it filters most of the issues such as the missing ids and the length of stay. Some of the important attributes of the preprocessed in-hospital-mortality data-set are presented in Table I, in which the first four columns report the description of the data, the number of patients, the number of ICU stays and the number of observed temporal features respectively. The last two columns report the number of continuous and categorical temporal variables, i.e., features, respectively. The train and test data-sets are split in the preprocessing step with a ratio of 85% - 15%.

Table I. This table reports some of the most important attributes of the in-hospital-mortality data-set extracted from MIMIC III by following the preprocessing in **[27]**.

Type	**Patients**	**ICU Stays**	**Variables**	**Cont.’s Var.**	**Cat Var.**
Train	15331	17903	17	13	4
Test	2763	3236	17	13	4

The in-hospital-mortality data-set contains the timeline of the first 48 hours of each patient’s stay in the ICU. It is clear from Table I, that some patients have been admitted to the ICU more than once. We remove such duplicates from the records and make sure that each patient has exactly one ICU record. Furthermore, to handle the sporadic nature of the data; we re-sample the temporal features to have exactly one entry in one hour resulting in a total of 48 entries for each patient same as [27]. In the case there is more than one entry in an hour, we take the mean and substitute it as the only entry of the hour to make the data consistent. This results in each patient represented by a matrix of 48 × 17. At this point, 83% of the entries in a patient’s time-series matrix are missing on average. The overall missing rate for all 17 temporal features is presented in Figure 2. to further highlight the issue.

It can be observed from Fig. 2 that some features have extremely high missing rate and are consequently not fit for further analysis. As such, we remove them from the data and are left with only 6 temporal features, all of which are continuous with a missing rate of around 10 %. After this, we also remove those patients who have more than 10 % missing entries. Finally, we are left with 13400 patients in the training data-set and 2312 in the test data-set, and the missing rate is reduced to 10 %. The missing entries are then substituted by the column mean and thereupon we assume the complete information of each patient's time-series which is a matrix of size 48 × 6 where the six temporal features are Diastolic Blood Pressure (DBP), Heart Rate (HR), Mean Blood Pressure (MBP), Oxygen Saturation Level (OSL), Respiratory Rate (RR) and Systolic Blood Pressure (SBP) respectively. Apart from the temporal features, we also observe the disease diagnostics of each patient. This information is later used to compute high-level meta-features as discussed previously in the report. The histogram of the disease counts of all patients in the training and testing data-sets is presented in Fig. 3.

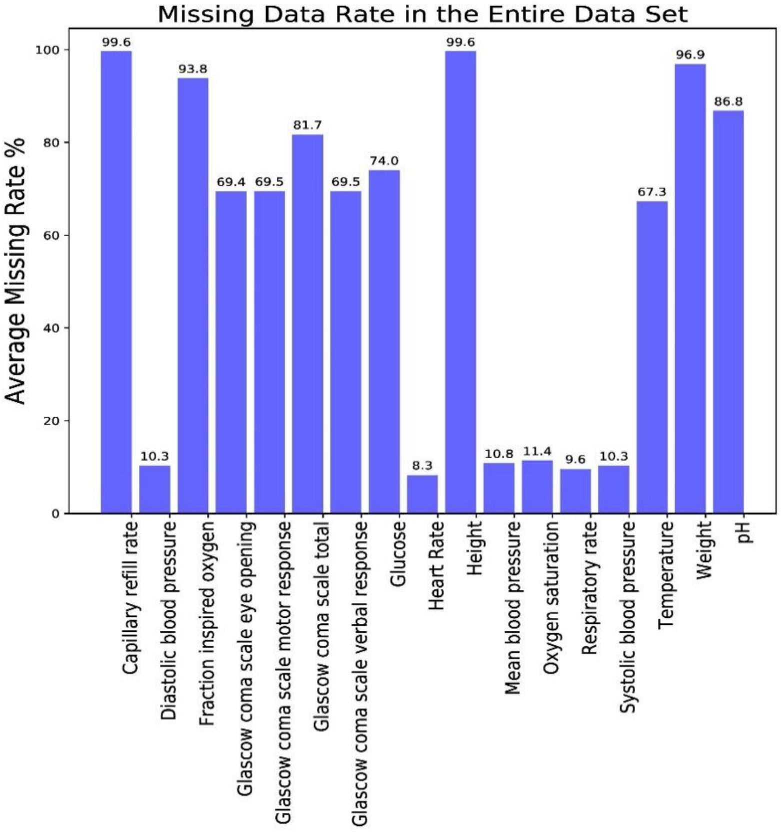

Figure 2. Average (i.e., train and test both) missing rate % for all 17 temporal features is presented. Capillary refill rate and Height are the channels with maximum missing rate (99.6) %, while Heart Rate has lowest missing ratio (8.3) %.

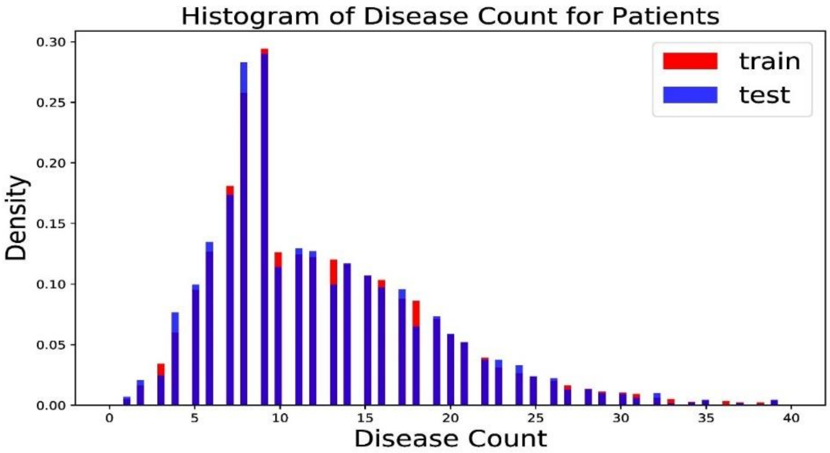

Figure 3. The histogram of disease counts for patients in the training and test data-sets is presented. The minimum and maximum number of disease(s) for an individual patient are 1 and 39 respectively; for both train and test data-set.

3.2. Learning of Meta-Features in Clinical Applications

Clinical data-sets are characterized by loads of supplementary information accompanying the primary data. Such supplementary information may contain details about the patients, the laboratory tests, and the working condition of the hospitals and ICUs. Some of this information may be useful for the clinical analysis, early triage, risk assessment, and a better understanding about the ongoing treatment. Thus, it is critical to incorporate such supplementary information for tasks such as temporal signal forecasting, risk assessment, mortality classification for critical patients, phenotype classification and length-of-stay-prediction. However, there is a lack of common algorithmic approaches to exploit such domain information to improve the outcome of the downstream learning tasks.

To conduct time-series forecasting for a particular patient; we propose to take a set of similar patients which is determined by some similarity criteria. Temporal signals extracted from these similar patients can be combined with the signals from the patient of interest to increase the robustness of the forecasting. This can improve the generalization ability of VRNNs for two reasons. Firstly, if the input time-series varies slightly; the model would be less prone to fail in reconstructing the time-series by including the correlated temporal signals of the similar patients. Secondly, the model utilizing the correlated temporal signals in the learning phase would be less likely to over-fit the data. For the similarity criterion, we choose the K-Nearest Neighbors (KNNs) [11] [12] with respect to the cosine similarity metric on disease diagnostics.

We denote the set of correlated temporal signals for a patient at time t with x_t^{rel}. The probability distributions for generative and inference networks are updated as:

$$p(x_{\leq T}, z_{\leq T}) = \prod_{t=1}^{T} p\left(x_t \middle| z_{\leq t}, x_{<t}, x_{<t}^{rel}\right) . p\left(z_t \middle| x_{<t}, z_{<t}, x_{<t}^{rel}\right) \quad (13)$$

$$q(z_{\le T}|x_{\le T}) = \prod\nolimits_{t=1}^{T} q\left(z_t|x_{\le t}, z_{<t}, x_{\le t}^{rel}\right) \tag{14}$$

where $x_{<t}^{rel}$ in Eq. (13) refers to the correlated temporal signals for a patient preceding time t. Similarly, $x_{\le t}^{rel}$ in Eq. (14) refers to the correlated temporal signals for a patient preceding and including time t. Note that in this way, all the expressions in Section 2.3 need to be updated by additionally conditioning on the multiple correlated temporal signals x_t^{rel}.

3.3. Improving Forecasting Accuracy with Meta-Features in MIMIC

The EHRs in the MIMIC III contain a variety of supplementary information, e.g., ethnicity, language, age and disease information, beyond the temporal features of the patients. However, most of such information is missing for the majority of the patients. Disease diagnostics is the only supplementary information present for each patient. As such, we only use the disease diagnostics as extra domain/supplementary information to compute the similarity between the patients. We convert each patient's disease information into a binary vector of size 6961 where 6961 is the size of the set of all unique diseases in the entire data-set. After this, we find the set of k most similar patients for each patient based on the cosine similarity of the disease vectors. We test the values of k for 2, 3, 4, and 5 and find out that $k = 3$ provides the best results. Thus, all the results mentioned in the following are achieved using k = 3 and $x_t^{rel} \in \mathbb{R}^{\mathrm{d}}$ where $d = 18$. Once we have x_t^{rel} available, we implement and evaluate the model.

3.4. Experimental Setup

In ECOLE, we considered the following variants of VRNN:

- Vanilla VRNN,
- VRNN-I (without the conditional prior in Eq. (6)),
- The proposed approaches: VRNN-S and VRNN-I-S ("S" stands for similarity), which implement the similar data $\boldsymbol{x_t^{rel}}$ into VRNN and VRNN-I respectively.

We do not include the other neural baselines such as recurrent neural network with extended Kalman filters (RNNEKFs) [8], robust echo state networks (RESNs) [9] and co-evolutionary multi-task learning (CMTL) [10] since we are fundamentally interested in robust and improved forecasting of VRNNs [5] by attempting to learn the local variations in the data. Table II reports the implementation details of all four models. In Table II, the first three columns show the model, the dimensions of x_t and z_t respectively. The fourth and fifth column describe the number of hidden layers and the size of each hidden layer accordingly. The last two columns report the batch size and the number of epochs respectively. The implementations of all four models are with GRUs and all temporal features are re-scaled between −1 and 1. The choice of the batch size is 100 based on [6]. For the choice of the number of hidden layers and their size, we try a variety of combinations including the previous settings in [27] [5] [16]. Our final choice is 50 for the hidden with in total 2 layers trained for 5 epochs. We found that at this setting all four models performed the best. Notably, this is different from any of the settings used in [27] [5] [16].

We are interested in evaluating our models for multi-step-ahead forecasting. We evaluate the models on one to ten-step ahead forecasting. For one-step-ahead forecasting, we train all the models on 47 time-steps and predict the last time-step. For two to five-step-ahead forecasting, we train all the models on 43 time-steps and predict the next two, three, four and five steps respectively. For six to ten-step-ahead forecasting, we train all the models on 38 time-steps and predict the next six, seven, eight, nine and ten steps. We evaluate all the models on Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum\nolimits_{i=1}^{M}(y_l - \hat{y}_l)^2} \tag{15}$$

where y_l and $\hat{y}_l$ in Eq. (15) are the vectors representing the true and predicted values of all six temporal features for the patient l and M denotes the size of the test data-set. We now discuss the results obtained from the above experimental setup.

3.5. Results

In this section, we first report the average, i.e., for all the temporal variables, Root Mean Square Error (RMSE) on the test data-set for multi-step-ahead forecasting in Table II. In this table, the first column displays the step size for forecasting. The next four columns present the RMSE with rounded standard deviations using VRNN, VRNN-I (i.e., without the conditional prior in Eq. (6)), VRNN-S (i.e., VRNN employing x_t^{rel}), and VRNN-I-S (i.e., without the conditional prior and employing x_t^{rel}). The last two columns share the p values resulting from the Mann-Whitney U test. These tests have the alternative hypotheses RMSE (VRNN-S) < RMSE (VRNN) and RMSE (VRNN-I-S) < RMSE (VRNN-I) respectively. These tests find if VRNNs utilizing x_t^{rel} (also labelled M3 and M4 in the table) are significantly better than the respective baselines (which are labelled M1 and M2 respectively in the table). From Table II, it can be observed that VRNN-I-S achieves the lowest values of RMSE in all the ten cases. Furthermore, VRNN-S achieves the second lowest error in all the ten cases. Lastly, the rounded standard deviations in Table II are analogous for all four models. From the last two columns in Table II, we find out that in 6/10 cases; at-least one of VRNN-S and VRNN-I-S performs significantly better than the respective baseline as indicated by the p values.

We further perform a simple qualitative analysis to highlight the importance of x_t^{rel} in robust and improved forecasting of VRNNs. We select three patients in the test data-set where VRNN-S and VRNN-I-S both achieve the lowest RMSE. For each of these patients, we select the ten most similar patients based on disease diagnostics and plot the corresponding cosine similarity values in the form of a heat map in Figure 4. This heat map verifies that our choice of $k = 3$ in previous section is plausible since in all three cases, high similarity values are observed for the first few (i.e., two, three) related patients only. Moving forward with $k = 3$; we report the information about the set of common diseases between our selected patients and their corresponding most similar patients in Table III. In this table, the first column shows the identity of each of the three selected patients. The second column reports the number of common diseases between that patient and its k most similar patients. The third column shares the International Classification of

Diseases, Ninth Revision (ICD9) codes for the corresponding diseases. The last column categorizes the respective ICD9 codes to the most appropriate disease family (i.e., Heart, Blood Pressure, Kidney, Respiratory) for better interpretation and analysis.

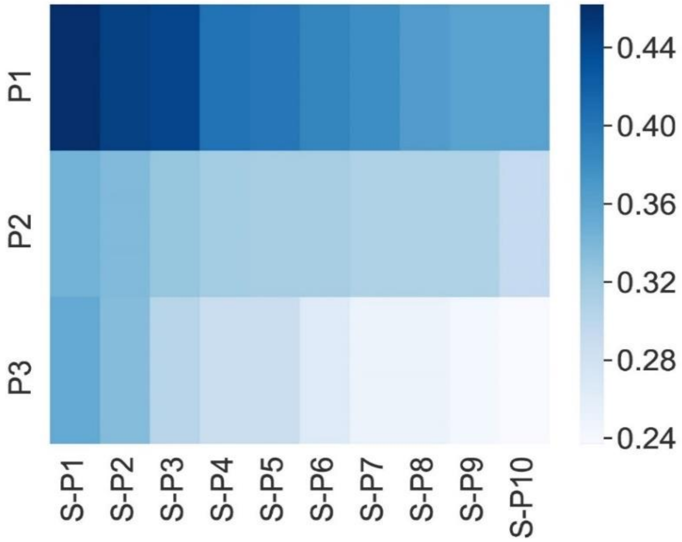

Figure 4. This heat map visualizes the cosine similarity values between our patients of interest (P1, P2, and P3) and their corresponding ten most similar patients (S-P) based on disease diagnostics.*

Table II. Average RMSE on all ten-steps-ahead forecasting tasks on test data is presented. The first column shows the step size, the next four columns share the RMSE for all four models. Given the alternative hypotheses $\boldsymbol{H_a}$: $\mathrm{M3} < \mathrm{M1}$ and $\boldsymbol{H_a}$: $\mathrm{M4} < \mathrm{M2}$ where M1, M2, M3 and M4 correspond to the models in columns 2-5 respectively; two Mann-Whitney U tests are performed to find if the error differences are significant using standard $\boldsymbol{\alpha = 0.05}$ in both tests.

Step Size	**VRNN (M1)**	**VRNN-I (M2)**	**VRNN-S (M3)**	**VRNN-I-S (M4)**	**H_a (M3<M1)**	**H_a (M4<M2)**
1	0.01152	0.01209	0.01040	**0.01034**	**3.5e-28**	**4.6e-64**
2	0.01047	0.01047	0.01042	**0.01039**	0.26	0.078
3	0.01058	0.01059	0.01053	**0.01050**	0.23	**0.045**
4	0.01062	0.01062	0.01057	**0.01054**	0.21	**0.036**
5	0.01062	0.01063	0.01058	**0.01055**	0.22	**0.021**
6	0.01071	0.01064	0.01062	**0.01060**	0.074	0.14
7	0.01071	0.01064	0.01063	**0.01060**	0.056	0.12
8	0.01073	0.01066	0.01064	**0.01062**	**0.046**	0.10
9	0.01074	0.01066	0.01065	**0.01062**	**0.042**	0.09
10	0.01073	0.01066	0.01065	**0.01062**	0.051	0.074

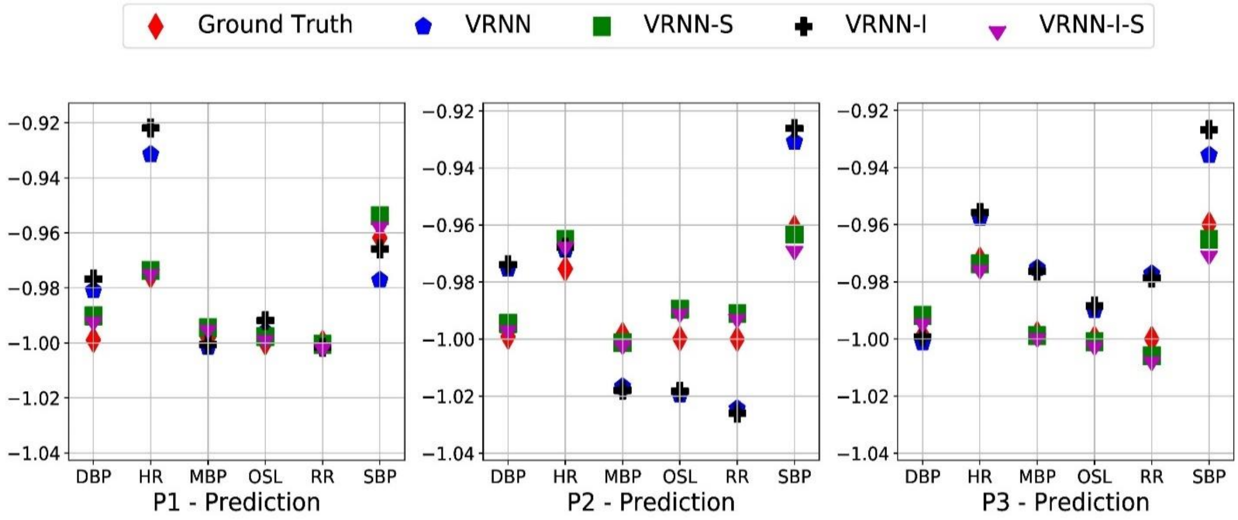

Figure 5. One-step-ahead prediction on all six temporal features of the selected patients are presented. The six temporal features are Diastolic Blood Pressure (DBP), Heart Rate (HR), Mean Blood Pressure (MBP), Oxygen Saturation Level (OSL), Respiratory Rate (RR) and Systolic Blood Pressure (SBP) respectively.

Table III. This table shares the information of the common diseases found between our selected patients and their k most similar patients.

ID	Diseases	ICD9	Category
P1	4	414(.01, .9), 427.31, 428.0	Heart, Blood Pressure
P2	3	785.52, 995.92, 584.9	High Blood Pressure, Kidney
P3	2	507.0, 518.81	Respiratory, Blood Pressure

After reporting the information about the common diseases, we plot the predictions of all four models on our patients of interest in Figure 5. This figure shares the one-step-ahead predicted values (re-scaled) for all six temporal variables for these patients. Considering the first patient (P1) in Figure 5; we observe that VRNN-S and VRNN-I-S outperform the baselines on Heart Rate (HR) which is related to the category of the most common diseases for that Patient in Table IV. Similarly analyzing the second patient (P2); we observe that VRNNS and VRNN-I-S outperform the baselines on Systolic Blood Pressure (SBP) which is strongly related to high blood pressure related diseases. Finally, the same analysis is performed for third patient (P3) where VRNN-S and VRNN-I-S achieve superior predictions on Respiratory Rate (RR) and Systolic Blood Pressure (SBP). From Figure 5, we verify that the set of correlated temporal features x_t^{rel} indeed help improving the forecasting accuracy of the VRNNs for clinical signals. This is especially true for the temporal features which are related to the set of the common diseases between the patients. We now move on to discuss the summary of our report along-side the future research line.

4. Summary & Future Work

This deliverable report focused on the research achievements regarding the task 3.2 about deep structured learning and model space learning for engineering and ICT data. The importance of learning high-level abstractions from data-sets can be hardly overstated, since these high-level meta-features can be utilized to foster the training process of a variety of downstream learning tasks. In this task, we investigated state-of-the-art approaches of temporal data analysis, and provided an overview of recent literature on time-series processing using deep structured models. In particular, we introduced a solid theoretical background for RNNs, VAEs and VRNNs from which we stemmed the leading edge research in the task 3.2.To meet the challenges in time series forecasting, we proposed a novel deep learning framework, which incorporates the high-level meta-features in training VRNNs to explore correlations among time series. The work has been published at IJCNN 2020. This research is an important contribution with practical impact, since the advantage and outperformance of the proposed approach in the domain-rich applications, i.e., applications where we have loads of supplementary data/information accompanying the primary data, is significant and demanded for many engineering and ICT use cases.

We performed extensive experiments to demonstrate the performance of the proposed methods. In particular, we evaluated the effectiveness of utilizing multiple correlated time series in time-series forecasting. Such correlated time-series are pervasive in the ICT domain, such as the medical sensor data monitoring the clinical status of a set of similar patients; where the similarity can be computed on the basis of the supplementary domain information such as disease diagnostics, age and ethnicity etc. As our baselines, we chose VRNN and its variant which are state-of-the-art deep generative models for sequential data-sets. Based on the empirical analysis reported in the deliverable, we demonstrated that the performance of VRNNs can be improved by integrating the correlated temporal signals. Additionally, one can find from our experiments that incorporation of multiple correlated time series helps recovering the temporal features related to the common diseases between the patients.

It is nonetheless important to state that the simple similarity criteria used in the experiments needs to be further enhanced to capture more complex relationships between the patients such as learning vector representations of graphs in an unsupervised fashion. These vector representations can then be included in the training to learn more robust relationships between the patients. We aim to focus on such enhanced similarity computations and other information-rich application areas, e.g., industry, IoT and communication network time-series forecasting, in future. On the basis of the points discussed above, we believe that discarding such supplementary domain information while analyzing clinical data-sets may not be an optimal strategy since such information may be used to improve the generalization. Lastly, we believe there is a dire need of additional clinical benchmark data-sets to improve upon the state-of-the-art in this area.

Bibliography

[1] H. Lütkepohl, New introduction to multiple time series analysis, Springer Science & Business Media, 2005.

[2] I. Goodfellow, A. Courville and Y. Bengio, Deep learning. Vol. 1. No. 2., Cambridge: MIT press, 2016.

[3] S. S. Rangapuram, M. Seeger, J. Gasthaus, L. Stella, Y. Wang and T. Januschowski, "Deep state space models for time series forecasting," in *Advances in Neural Information Processing Systems*, 2018.

[4] J. Bayer and C. Osendorfer, "Learning stochastic recurrent networks," arXiv preprint arXiv:1411.7610, 2014.

[5] C. Junyoung, K. Kastner, L. Dinh, K. Goel, A. Courville and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in neural information processing systems 28,* 2015.

[6] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR),* 2013.

[7] D. Hsu, "Multi-period time series modeling with sparsity via Bayesian variational inference," arXiv preprint arXiv:1707.00666, 2017.

[8] G. V. Puskorius and L. A. Feldkamp, "Neurocontrol of nonlinear dynamical systems with kalman filter trained recurrent networks," *IEEE Transactions on neural networks,* vol. 5, no. 2, pp. 279-297, 1994.

[9] D. Li, H. Min and J. Wang, "“Chaotic time series prediction based on a novel robust echo state network," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 23, no. 5, pp. 787-799, 2012.

[10] R. Chandra, O. Yew-Soon and G. Chi-Keong, "Co-evolutionary multi-task learning with predictive recurrence for multi-step chaotic time series prediction," *Neurocomputing,* pp. 21-34, 2017.

[11] C. M. Bishop, Pattern recognition and machine learning, springer, 2006.

[12] T. Hastie, R. Tibshirani and J. Friedman, The elements of statistical learning: data mining, inference, and prediction, pringer Science & Business Media, 2009.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[14] K. Cho, B. V. Merriënboer, G. Caglar, B. Dzmitry, B. Fethi, S. Holger and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[15] R. Pascanu, T. Mikolov and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning,* 2013.

[16] D. P. Kingma, D. J. Rezende, S. Mohamed and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in neural information processing systems,* 2014.

[17] S. Ullah, Z. Xu, H. Wang, S. Menzel, B. Sendhoff and T. Bäck, "Exploring Clinical Time Series Forecasting with Meta-Features in Variational Recurrent Models," in *2020 International Joint Conference on Neural Networks (IJCNN*, 2020.

[18] M. III. [Online]. Available: https://www.nature.com/articles/sdata201635.

[19] D. W. Bates, S. Suchi, O.-M. Lucila, S. Anand and E. Gabriel, "Big data in health care: using analytics to identify and manage high-risk and high-cost patients," *Health Affairs,* vol. 33, no. 7, pp. 1123-1131, 2014.

[20] M. Chen, H. Yixue, H. Kai, W. Lu and W. Lin, "Disease prediction by machine learning over big data from healthcare communities," *Ieee Access,* vol. 5, pp. 8869-8879, 2017.

[21] H. Kaur and K. W. Siri, "Empirical study on applications of data mining techniques in healthcare," *Journal of Computer science,* vol. 2, no. 2, pp. 94-200, 2006.

[22] H. C. Koh and T. Gerald, "Data mining applications in healthcare," *Journal of healthcare information management,* vol. 19, no. 2, p. 65, 2011.

[23] G. D. Clifford, S. Daniel J and V. Mauricio, "User guide and documentation for the MIMIC II database," MIMIC-II database version 2, no. 95, 2009.

[24] Z. Che, K. David, L. Wenzhe, B. Mohammad Taha and L. Yan, "Deep computational phenotyping," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[25] E. Choi, B. Mohammad Taha, S. Andy, S. Walter F and S. Jimeng, "Doctor ai: Predicting clinical events via recurrent neural networks," in *Machine Learning for Healthcare Conference*, 2016.

[26] T. A. Lasko, D. Joshua C and L. Mia A, "Correction: computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data," *PLoS one,* vol. 8, no. 8, 2013.

[27] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg and A. Galstyan, "Multitask learning and benchmarking with clinical time series," *Scientific data,* vol. 6, no. 1, pp. 1-18, 2019.

[28] A. Rajkomar, E. Oren, K. Chen, A. Dai, N. Hajaj, M. Hardt, P. Liu, X. Liu, J. Marcus, M. Sun and P. Sundberg, "Scalable and accurate deep learning with electronic health records," *NPJ Digital Medicine,* vol. 1, no. 1, p. 18, 2018.

[29] A. Oren, E. Hazan and A. Zeevi, "Online time series prediction with missing," in *International Conference on Machine Learning,* 2015.

[30] T. Gentimis, A. Ala'J, A. Durante, K. Cook and R. Steele, "Predicting hospital length of stay using neural networks on mimic iii data," in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech*, 2017.