Spatio-Temporal Neural Networks for Space-Time Series Forecasting and Relations Discovery

Ali Ziat*†, Edouard Delasalles†, Ludovic Denoyer, Patrick Gallinari
Sorbonne Universits, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris.
*Vedecom Institute, Eco-Mobility Department, 77000, Versailles, France
† Authors have equal contributions.
firstname.name@lip6.fr

***Abstract*—We introduce a dynamical spatio-temporal model formalized as a recurrent neural network for forecasting time series of spatial processes, i.e. series of observations sharing temporal and spatial dependencies. The model learns these dependencies through a structured latent dynamical component, while a decoder predicts the observations from the latent representations. We consider several variants of this model, corresponding to different prior hypothesis about the spatial relations between the series. The model is evaluated and compared to state-of-the-art baselines, on a variety of forecasting problems representative of different application areas: epidemiology, geo-spatial statistics and car-traffic prediction. Besides these evaluations, we also describe experiments showing the ability of this approach to extract relevant spatial relations.**

I. Introduction

Time series exhibiting spatial dependencies are present in many domains including ecology, meteorology, biology, medicine, economics, traffic, and vision. The observations can come from multiple sources e.g. GPS, satellite imagery, video cameras, etc. Two main difficulties when modeling spatio-temporal data come from their size - sensors can cover very large space and temporal lags - and from the complexity of the data generation process. Reducing the dimensionality and uncovering the underlying data generation process naturally leads to consider latent dynamic models. This has been exploited both in statistics [1] and in machine learning (ML) [2], [3].

Deep learning has developed a whole range of models for capturing relevant information representations for different tasks and modalities. For dynamic data, recurrent neural networks (RNN) handles complex sequences for tasks like classification, sequence to sequence prediction, sequence generation and many others [4]–[6].

These models are able to capture meaningful features of sequential data generation processes, but the spatial structure, essential in many applications, has been seldom considered in Deep Learning. Very recently, convolutional RNNs [7], [8] and video pixel networks [9] have been used to handle both spatiality and temporality, but for video applications only.

We explore a general class of deep spatio-temporal models by focusing on the problem of time series forecasting of spatial processes for different types of data. Although the target in this paper is forecasting, the model can easily be extended to cover related tasks like spatial forecasting (kriging [10]) or data imputation.

The model, denoted Spatio-Temporal Neural Network (STNN), has been designed to capture the dynamics and correlations in multiple series at the spatial and temporal levels. This is a dynamical system model with two components: one for capturing the spatio-temporal dynamics of the process into latent states, and one for decoding these latent sates into actual series observations.

The model is tested and compared to state of the art alternatives, including recent RNN approaches, on spatio-temporal series forecasting problems: disease prediction, traffic forecasting, meteorology and oceanography. Besides a comparative evaluation on forecasting tasks, the ability of the model to discover relevant spatial relations between series is also analyzed.

The paper is organized as follow: in section II we introduce the related work in machine learning and spatio-temporal statistics. The model is presented in section III with its different variants. The experiments are described in section V for both forecasting V-B and relations discovery V-C.

II. Related Work

The classical topic of time series modeling and forecasting has given rise to an extensive literature, both in statistics and machine learning. In statistics, classical linear models are based on auto-regressive and moving average components. Most assume linear and stationary time dependencies with a noise component [11]. In machine learning, non linear extensions of these models based on neural networks were proposed as early as the nineties, opening the way to many other non linear models developed both in statistics and ML, like kernel methods [12] for instance.

Dynamical state space models, such as recurrent neural networks, have been used for time series forecasting in different contexts since the early nineties [13]. Recently, these models have witnessed important successes in different areas of sequence modeling problems, leading to breakthrough in domains like speech [14], language generation [15], translation [16] and many others. A model closely related to our work is the dynamic factor graph model [17] designed for multiple series modeling. Like ours, it is a generative model with a latent component that captures the temporal dynamics and a decoder

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.80

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 13:58:07 UTC from IEEE Xplore. Restrictions apply.

for predicting the series. However, spatial dependencies are not considered in this model, and the learning and inference algorithms are different.

Recently, the development of non parametric generative models has become a very popular research direction in Deep Learning, leading to different families of innovative and promising models. For example, the Stochastic Gradient Variational Bayes algorithm (SGVB) [18] provides a framework for learning stochastic latent variables with deep neural networks, and has recently been used by some authors to model time series [19]–[21]. In our context, which requires to model explicitly both spatial and temporal dependencies between multiple time series, variational inference as proposed by such models is still intractable, especially when the number of series grows, as in our experiments.

Spatio-temporal statistics have already a long history [1], [22]. The traditional methods rely on a descriptive approach using the first and second-order moments of the process for modeling the spatio-temporal dependencies. More recently, dynamical state space models, where the current state is conditioned on the past have been explored [23]. For these models, time and space can be either continuous or discrete. The usual way is to consider discrete time, leading to the modeling of time series of spatial processes. When space is continuous, models are generally expressed by linear integro-difference equations, which is out of the scope of our work. With discrete time and space, models come down to general vectorial autoregressive formulations. These models face a curse of dimensionality in the case of a large number of sources. Different strategies have been adopted to solve this problem, such as embedding or parameter reduction. This leads to model families that are close to the ones used in machine learning for modeling dynamical phenomena, and incorporate a spatial components. An interesting feature of these approaches is the incorporation of prior knowledge inspired from physical models of space-time processes. This consists in taking inspiration from prior background of physical phenomena, e.g. diffusion laws in physics, and using this knowledge as guidelines for designing dependencies in statistical models. In climatology, models taking into account both temporal and geographical components have also been used such as Gaussian Markov Random Fields [24]

In the machine learning domain, spatio-temporal modeling has been seldom considered, even though some spatio-temporal models have been proposed [25]. [2] introduce a tensor model for kriging and forecasting. [3] use conditional random fields for detecting activity in video, where time is discretized at the frame level and one of the tasks is the prediction of future activity. Brain Computer Interface (BCI) is another domain for spatio-temporal data analysis with some work focused on learning spatio-temporal filters [26], [27], but this is a very specific and different topic.

III. The STNN Model

A. Notations and Task

Let us consider a set of n temporal series, m is the dimensionality of each series and T their length [1]. $m = 1$ means that we consider n univariate series, while $m > 1$ correspond to n multivariate series each with m components. We will denote X the values of all the series between time 1 and time T. X is then a $\mathbb{R}^{T\times n\times m}$ 3-dimensional tensor, such that $X_{t,i,j}$ is the value of the j-th component of series i at time t. X_t will denote a slice of X at time t such that $X_t \in \mathbb{R}^{n\times m}$ denotes the values of all the series at time t.

For simplicity, we first present our model in a mono-relational setting. An extension to multi-relational series where different relations between series are observed is described in section III-D. We consider that the spatial organization of the sources is captured through a matrix $W \in \mathbb{R}^{n\times n}$. Ideally, W would indicate the mutual influence between sources, given as a prior information. In practice, it might be a proximity or similarity matrix between the sources: for geo-spatial problems, this might correspond to the inverse of a physical distance - e.g. geodesic - between sources. For other applications, this might be provided through local connections between sources using a graph structure (e.g. adjacency matrix for connected roads in a traffic prediction application or graph kernel on the web). In a first step, we make the hypothesis that W is provided as a prior on the spatial relations between the series. An extension where weights on these relations are learned is presented in section IV.

We consider in the following the problem of spatial time series forecasting i.e predicting the future of the series, knowing their past. We want to learn a model $f : \mathbb{R}^{T\times n\times m} \times \mathbb{R}^{n\times n} \rightarrow \mathbb{R}^{\tau\times n\times m}$ able to predict the future at τ time-steps of the series based on X and on their spatial dependency.

B. Modeling Time Series with Continuous Latent Factors

Let us first introduce the model in the simpler case of multiple time series prediction, without considering spatial relations. The model has two components.

The first one captures the dynamic of the process and is expressed in a latent space. Let Z_t be the latent representation, or latent factors, of the series at time t. The dynamical component writes $Z_{t+1} = g(Z_t)$. The second component is a decoder which maps latent factors Z_t onto a prediction of the actual series values at t: $\tilde{X}_t = d(Z_t)$, $\tilde{X}_t$ being the prediction computed at time t. In this model, both the representations Z_t and the parameters of the dynamical and decoder components are learned. Note that this model is different from the classical RNN formulations [16], [28]. The state space component of a RNN with self loops on the hidden cells writes $Z_{t+1} = g(Z_t, X'_t)$, where X'_t is the ground truth X_t during training, and the predicted value $\tilde{X}_t$ during inference. In our approach, latent factors Z_t are learned during training

[1]We assume that all the series have the same dimensionality and length. This is often the case for spatio-temporal problems otherwise this restriction can be easily removed.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 13:58:07 UTC from IEEE Xplore. Restrictions apply.

and are not an explicit function of past inputs as in RNNs: the dynamics of the series are then captured entirely in the latent space.

This formal definition makes the model more flexible than RNNs since not only the dynamic transition function $g(.)$, but also the state representations Z_t are learned from data. A similar argument is developed in [17]. It is similar in spirit to Hidden Markov models or Kalman filters.

a) Learning problem: Our objective is to learn the two mapping functions d and g together with the latent factors Z_t, directly from the observed series. We formalize this learning problem with a bi-objective loss function that captures the dynamics of the series in the latent space and the mapping from this latent space to the observations. Let $\mathcal{L}(g, d, Z)$ be this objective function:

$$\begin{aligned} \mathcal{L}(d, g, Z) = \frac{1}{T}\sum_t \Delta(d(Z_t), X_t) \text{ (i)} \\ +\lambda\frac{1}{T}\sum_{t=1}^{T-1} ||Z_{t+1} - g(Z_t)||^2 \qquad \text{(ii)} \end{aligned} \tag{1}$$

The first term (i) measures the ability of the model to reconstruct the observed values X_t from the latent factor Z_t. It is based on loss function Δ which measures the discrepancy between predictions $d(Z_t)$ and ground truth X_t. The second term (ii) aims at capturing the dynamicity of the series in the latent space. This term forces the system to learn latent factors Z_{t+1} that are as close as possible to $g(Z_t)$. Note that in the ideal case, the model converges to a solution where $Z_{t+1} = g(Z_t)$, which is the classical assumption made when using RNNs. The hyper-parameter λ is used here to balance this constraint and is fixed by cross-validation. The solution d^*, g^*, Z^* to this problem is computed by minimizing $\mathcal{L}(d, g, Z)$:

$$d^*, g^*, Z^* = \arg\min_{d,g,Z} \mathcal{L}(d, g, Z) \tag{2}$$

b) Learning algorithm: In our setting, functions d and g, described in the next section, are differentiable parametric functions. Hence, the learning problem can be solved end-to-end with Stochastic Gradient Descent (SGD) techniques[2] directly from (9). At each iteration, a pair (Z_t, Z_{t+1}) is sampled, and Z_t, Z_{t+1}, g and d are updated according to the gradient of (1). Training can also be performed via mini-batch, meaning that for each iteration several pairs (Z_t, Z_{t+1}) are sampled, instead of a single pair. This results in a high learning speed-up when using GPUs which are the classical configuration for running such methods.

c) Inference: Once the model is learned, it can be used to predict future values of the series. The inference method is the following: the latent factors of any future state of the series is computed using the g function, and the corresponding observations is predicted by using d on these factors. Formally, let us denote $\tilde{Z}_\tau$ the predicted latent factors at time $T+\tau$. The forecasting process computes $\tilde{Z}_\tau$ by successively applying the g function τ times on the learned vector Z_T:

$$\tilde{Z}_\tau = g \circ g \circ ... \circ g(Z_T) \tag{3}$$

and then computes the predicted outputs $\tilde{X}_\tau$:

$$\tilde{X}_\tau = d(\tilde{Z}_\tau) \tag{4}$$

C. Modeling Spatio-Temporal Series

Let us now introduce a spatial component in the model. We consider that each series has its own latent representation at each time step. Z_t is thus a $n \times N$ matrix such that $Z_{t,i} \in \mathbb{R}^N$ is the latent factor of series i at time t, N being the dimension of the latent space. This is different from approaches like [17] or RNNs for multiple series prediction, where Z_t would be a single vector common to all the series. The decoding and dynamic functions d and g are respectively mapping $\mathbb{R}^{n\times N}$ to $\mathbb{R}^{n\times m}$ and $\mathbb{R}^{n\times N}$ to $\mathbb{R}^{n\times N}$.

The spatial information is integrated in the dynamic component of our model through a matrix $W \in \mathbb{R}_+^{n\times n}$. In a first step, we consider that W is provided as prior information on the series' mutual influences. In IV, we remove this restriction, and show how it is possible to learn the weights of the relations, and even the spatial relations themselves, directly from the observed data. The latent representation of any series at time $t+1$ depends on its own latent representation at time t (intra-dependency) and on the representations of the other series at t (inter-dependency). Intra-dependency will be captured through a linear mapping denoted $\Theta^{(0)} \in \mathbb{R}^{N\times N}$ and inter-dependency will be captured by averaging the latent vector representations of the neighboring series using matrix W, and computing a linear combination denoted $\Theta^{(1)} \in \mathbb{R}^{N\times N}$ of this average. Formally, the dynamic model $g(Z_t)$ is designed as follow:

$$Z_{t+1} = h(Z_t\Theta^{(0)} + WZ_t\Theta^{(1)}) \tag{5}$$

Here, h is a non-linear function. In the experiments we set $h = tanh$ but h could also be a more complex parametrized function like a multi-layer perceptron (MLPs) for example – see section V. The resulting optimization problem over d, Z, $\Theta^{(0)}$ and $\Theta^{(1)}$ writes:

$$\begin{aligned} d^*, Z^*, \Theta^{(0)*}, \Theta^{(1)*} = \\ \arg\min_{d,Z,\Theta^{(0)},\Theta^{(1)}} \frac{1}{T}\sum_t \Delta(d(Z_t), X_t) + \\ \lambda\frac{1}{T}\sum_{t=1}^{T-1} ||Z_{t+1} - h(Z_t\Theta^{(0)} + WZ_t\Theta^{(1)})||^2 \\ \text{with } Z_t \in \mathbb{R}^{n\times N} \end{aligned} \tag{6}$$

D. Modeling different types of relations

The model in section III-C considers that all the spatial relations are of the same type (e.g. source proximity). For many problems, we will have to consider different types of relations. For instance, when sensors correspond to physical locations and the target is some meteorological variable, the relative orientation or position of two sources may imply a different type

[2]In the experiments, we used the Nesterov's Accelerated Gradient (NAG) method [29].

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 13:58:07 UTC from IEEE Xplore. Restrictions apply.

Dataset	Task	n	m	nb relations	time-step	total length	training length	#folds
Google Flu	Flu trends	29	1	1 to 3	weeks	$\approx$ 10 years	2 years	50
GHO (25 datasets)	Number of deaths	91	1	1 to 3	years	45 years	35 years	5
Wind	Wind speed and orientation	500	2	1 to 3	hours	30 days	10 days	20
PST	Temperature	2520	1	8	months	$\approx$ 33 years	10 years	15
Bejing	Traffic Prediction	5000	1	1 to 3	15 min	1 week	2 days	20

TABLE I: Datasets statistics. n is the number of series, m is the dimension of each series, $timestep$ corresponds to the duration of one time-step and *#folds* corresponds to the number of temporal folds used for validation. For each fold, evaluation has been made on the next 5 values at $T+1, T+2, ..., T+5$. The relation columns specifies the number of different relation types used in the experiments i.e the number of $W^{(r)}$ matrices used in each dataset. 1 to 3 means that the best among 1 to 3 relations was selected using cross validation

of dependency between the sources. In the experimental section, we consider problems with relations based on the relative position of sources, $north, south, west, east, ...$ The multi-relational framework generalizes the previous formulation of the model, and allows us to incorporate more abstract relations, like different measures of proximity or similarity between sources. For instance, when sources are spatiality organized in a graph, it is possible to define different graph kernels, each one of them modeling a specific similarity. The following multi-relational formulation is based on adjacency matrices, and can directly incorporate different graph kernels.

Each possible relation type is denoted *r* and is associated to a matrix $W^{(r)} \in \mathbb{R}_+^{n\times n}$. For now, and as before, we consider that the $W^{(r)}$ are provided as prior knowledge. Each type of relation *r* is associated to a transition matrix $\Theta^{(r)}$. This learned matrix captures the spatio-temporal relationship between the series for this particular type of relation. The model dynamics writes:

$$Z_{t+1} = h(Z_t\Theta^{(0)} + \sum_{r\in\mathcal{R}} W^{(r)} Z_t \Theta^{(r)}) \tag{7}$$

where $\mathcal{R}$ is the set of all possible types of relations. The learning problem is similar to equation (6) with Z_{t+1} replaced by the expression in (7). The corresponding model is illustrated in figure 1. This dynamic model aggregates the latent representations of the series for each type of relation, and then applies $\Theta^{(r)}$ on this aggregate. Each $\Theta^{(r)}$ is able to capture the dynamics specific to relation (r).

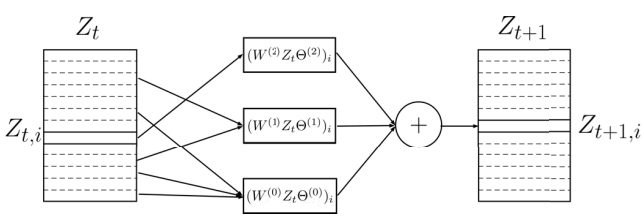

Fig. 1: Architecture of the STNN model as described in Section III-D

.

IV. Learning the relation weights and capturing spatio-temporal correlations

In the previous sections, we made the hypothesis that the spatial relational structure and the strength of influence between series were provided to the model through the $W^{(r)}$ matrices. We introduce below an extension of the model where weights on these relations are learned. This model is denoted STNN-R(efining). We further show that, with a slight modification, this model can be extended to learn both the relations and their weights directly from the data, without any prior. This extension is denoted STNN-D(iscovering).

We will first introduce the STNN-R extension. Let $\Gamma^{(r)} \in \mathbb{R}^{n\times n}$ be a matrix of weights such that $\Gamma^{(r)}_{i,j}$ is the strength of the relation between series i and j in the relation r. Let us extend the formulation in Equation (7) as follows:

$$Z_{t+1} = h(Z_t\Theta^{(0)} + \sum_{r\in\mathcal{R}} (W^{(r)} \odot \Gamma^{(r)}) Z_t \Theta^{(r)}) \tag{8}$$

where $\Gamma^{(r)}$ is a matrix to be learned, $W^{(r)}$ is a prior i.e a set of observed relations, and $\odot$ is the element-wise multiplication between two matrices. The learning problem can be now be written as:

$$\begin{aligned} d^*, Z^*, \Theta^*, \Gamma^* = \\ \arg\min_{d,Z,\Gamma} \frac{1}{T}\sum_t \Delta(d(Z_t), X_t) + \gamma|\Gamma| \\ +\lambda\frac{1}{T}\sum_{t=1}^{T-1} ||Z_{t+1} - h(\sum_{r\in(R)} (W^{(r)} \odot \Gamma^{(r)}) Z_t\Theta^{(r)})||^2 \end{aligned} \tag{9}$$

where $|\Gamma^{(r)}|$ is a l_1 regularizing term that aims at sparsifying $\Gamma^{(r)}$. We thus add an hyper-parameter γ to tune this regularization factor.

If no prior is available, then simply removing the $W^{(r)}s$ from equation (8) leads to the following model:

$$Z_{t+1} = h(Z_t\Theta^{(0)} + \sum_{r\in\mathcal{R}} \Gamma^{(r)} Z_t \Theta^{(r)}) \tag{10}$$

where $\Gamma^{(r)}$ is no more constrained by the prior $W^{(r)}$ so that it will represent both the relational structure and the relation weights. Both models are learned with SGD, in the same way as described in III-B. The only difference is that a gradient step on the $\Gamma^{(r)}s$ is added.

V. Experiments

Experiments are performed on a series of spatio-temporal forecasting problems representative of different domains. We consider predictions within a $+5$ horizon i.e. given a training series of size T, the evaluation of the quality of the model

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 13:58:07 UTC from IEEE Xplore. Restrictions apply.

Models	Disease		Car Traffic	Geographical		
	Google Flu	GHO[3]	Beijing	Speed	Direction	PST
MEAN	.175	.335	.201	0.191	0.225	.258
AR	.101±.004	.299 ± .008	.075 ± .003	.082 ± .005	0.098±.016	.15 ± .002
VAR-MLP	.095 ± .004	.291 ± .004	.07 ± .002	.071 ± .005	0.111±0.14	.132 ± .003
DFG	.095 ± .008	.288 ± .002	.068 ± .005	.07 ± .004	.092 ± .006	.99 ± .019
RNN-tanh	.082 ± .008	.287 ± .011	.075 ± .006	.064 ± .003	.09 ± .005	.141 ± .01
RNN-GRU	.074 ± .007	.268 ± .07	.074 ± .002	.059 ± .009	.083 ± .005	.104 ± .008
STNN	.066 ± .006	**.261** ± .009	.056 ± .003	**.047** ± .008	**.061** ± .008	.095 ± .008
STNN-R	**.061** ± .008	**.261** ± .01	**.055** ± .004	**.047** ± .008	**.061** ± .008	**.08** ± .014
STNN-D	.073 ± .007	.288 ± .09	.069 ± .01	.059 ± .008	.073 ± .008	.109 ± .015

TABLE II: Average RMSE for the different datasets computed for T+1, T+2,...,T+5. Standard deviation was computed by re-training the models on different seeds.

Disease / Model	AR	VAR-MLP	RNN-GRU	Mean	DFG	STNN-R
All causes	0.237	0.228	0.199	0.35	0.291	**0.197**
Tuberculosis	0.407	0.418	**0.37**	0.395	0.421	0.377
Congenital syphilis	0.432	0.443	0.417	0.459	0.422	**0.409**
Diphtheria	0.406	0.396	0.387	0.404	0.419	**0.385**
Malignant neoplasm of esophagus	0.355	**0.341**	**0.341**	0.363	0.372	0.345
Malignant neoplasm of stomach	0.44	0.434	0.431	0.455	0.452	**0.43**
	0.267	0.254	0.282	0.303	0.301	**0.253**
Malignant neoplasm of intestine	0.281	0.29	0.278	0.314	0.305	**0.275**
Malignant neoplasm of rectum	0.501	0.499	**0.481**	0.504	0.509	0.498
Malignant neoplasm of larynx	0.321	0.313	0.32	0.314	0.329	**0.310**
Malignant neoplasm of breast	0.375	0.375	0.382	0.394	0.38	**0.36**
Malignant neoplasm of prostate	0.111	0.113	**0.109**	0.184	0.138	**0.109**
Malignant neoplasm of skin	0.253	0.243	0.227	0.264	0.256	**0.221**
Malignant neoplasm of bones	0.103	0.099	0.097	0.204	0.173	**0.08**
Malignant neoplasm of all other and unspecified sites	0.145	0.157	**0.147**	0.164	0.169	0.156
Lymphosarcoma	0.15	0.132	0.13	0.231	0.135	**0.122**
Benign neoplasms	0.366	0.362	0.332	0.398	0.331	**0.331**
Avitaminonsis	0.492	0.474	0.449	0.571	0.58	**0.414**
Allergic disorders	0.208	0.217	0.221	0.342	0.24	**0.202**
Multiple sclerosis	0.061	0.057	0.061	0.242	0.152	**0.056**
Rheumatic fever	0.325	0.31	0.287	0.345	0.313	**0.256**
Diseases of arteries	0.302	0.301	0.269	0.345	0.328	**0.238**
Influenza	0.141	0.141	0.155	0.23	0.217	**0.125**
Pneumonia	0.119	0.128	**0.1**	0.187	0.187	**0.1**
Pleurisy	0.246	0.246	0.247	0.29	0.272	**0.245**
Gastro-enteritis	0.386	0.369	**0.291**	0.394	0.398	0.295
Disease of teeth	0.344	0.312	0.305	0.413	0.361	**0.302**

TABLE III: RMSE of STNN-R over the 25 datasets in GHO for T+1, T+2,...,T+5

will be made over $T+1$ to $T+5$ time steps. The different model hyper-parameters are selected using a time-series cross-validation procedure called rolling origin as in [30], [31]. This protocol makes use of a sliding window of size T': on a series of length T, a window of size T' shifted several times in order to create a set of train/test folds. The beginning of the T' window is used for training and the remaining for test. The value of T' is fixed so that it is large enough to capture the main dynamics of the different series. Each series was re-scaled between 0 and 1.

We performed experiments with the following models:
(i) **Mean**: a simple heuristic which predicts future values of a series with the mean of its observed past values computed on the T' training steps of each training fold.
(ii) **AR**: a classical univariate Auto-Regressive model. For each series and each variable of the series, the prediction is a linear function of R past lags of the variable, R being a hyper-parameter tuned on a validation set.
(iii) **VAR-MLP**: a vectorial auto-regressive model where the predicted values of the series at time $t+1$ depend on the past values of all the series for a lag of size R. The predictive model is a multi-layer perceptron with one hidden layer. Its performance were uniformly better than a linear VAR. Here again the hidden layer size and the lag R were set by validation
(iv) **RNN-tanh**: a vanilla recurrent neural network with one hidden layer of recurrent units and tanh non-linearities. As for the **VAR-MLP**, one considers all the series simultaneously, i.e. at time t the RNN receives as input X_{t-1} the values of all the series at $t-1$ and predicts X_t. A RNN is a dynamical state-space model but its latent state Z_t explicitly depends through a functional dependency both on the preceding values of the series X_{t-1} and on the preceding state Z_{t-1}. Note that this model has the potential to capture the spatial dependencies since all the series are considered simultaneously, but does not

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 13:58:07 UTC from IEEE Xplore. Restrictions apply.

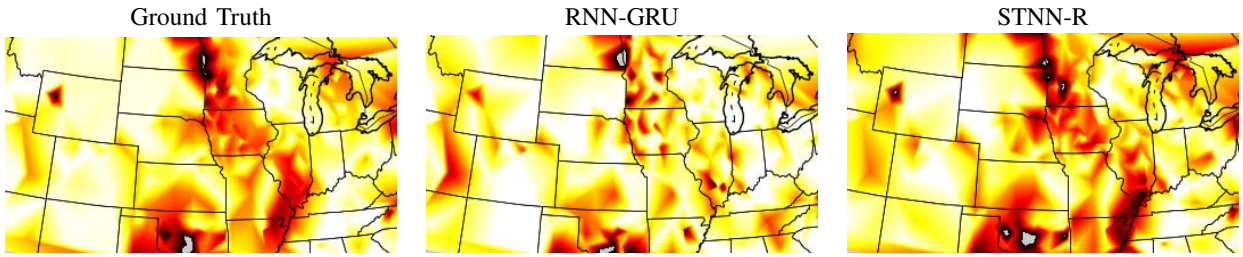

Fig. 2: Prediction of wind speed over around 500 stations on the US territory. prediction is shown at time-step $T+1$ for RNN-GRU (centre) and STNN-R (right).

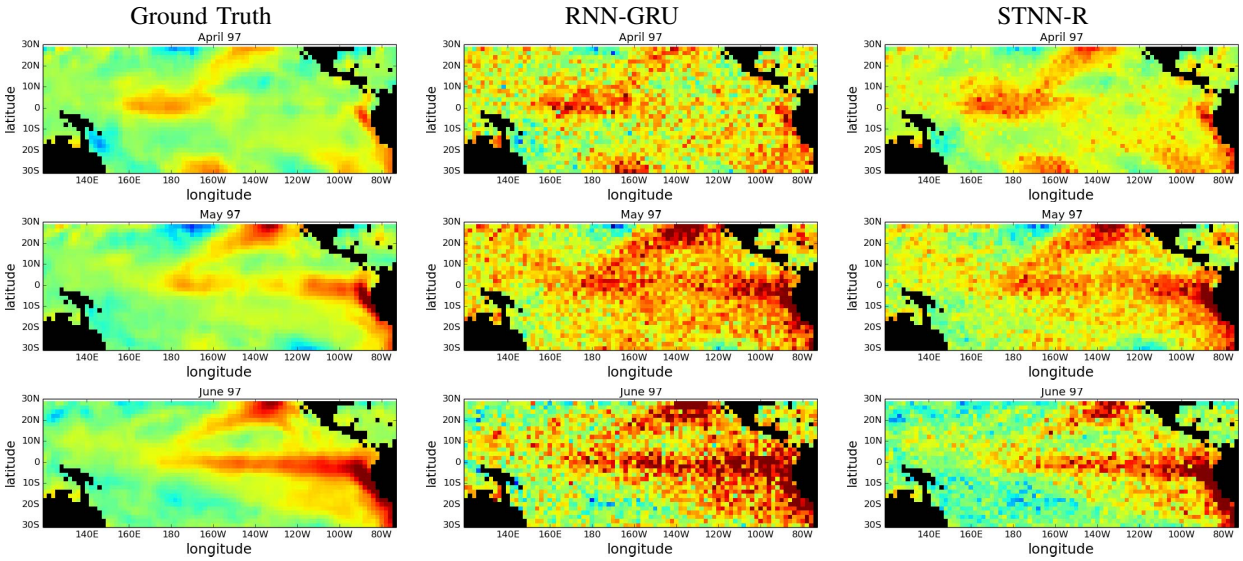

Fig. 3: Example of a 3 months prediction of Pacific temperature. Left column is the ground truth, central and right columns correspond respectively to RNN-GRU and STNN-R predictions at horizon $T+1$, $T+2$ and $T+3$ (top to bottom).

model them explicitly.

(v) **RNN-GRU**: same as the **RNN-tanh**, but recurrent units is replaced with gated recurrent units (GRU) units, which are considered state of the art for many sequence prediction problems today [4]. We have experimented with several architectures, but using more than one layer of GRU units did not improve the performance, so we used 1 layer in all the experiments.

(vi) **Dynamic Factor Graph (DFG)**: the model proposed in [17] is the closest to ours but uses a joint vectorial latent representation for all the series as in the RNNs, and does not explicitly model the spatial relations between series.

(vii) **STNN**: our model where g is the function described in equation (7), h is the $tanh$ function, and d is a linear function. Note that other architectures for d and g have been tested (e.g. multi-layer perceptrons) without improving the quality of the prediction. The λ value has been set by cross validation.

(viii and ix) **STNN-R** and **STNN-D**: For the forecasting experiments the γ value of the L_1 penalty (see equation (9)) wer set to 0 since higher value decreased the performance, a phenomena often observed in other models such as L_1-regularized SVMs. The influence of γ on the discovered spatial structure is further discussed and illustrated in figure 6.

The complete set of hyper-parameters values for the different models is given in appendix.

[4]We also performed tests with LSTM and obtained similar results as GRU.

A. Datasets

The different forecasting problems and the corresponding datasets are described below. The dataset characteristics are provided in table V-A.

- **Disease spread forecasting:** The **Google Flu** dataset contains for 29 countries, about ten years of weekly estimates of influenza activity computed by aggregating Google search queries (see http://www.google.org/flutrends). We extract binary relations between the countries, depending on whether or not they share a border, as a prior W.
- **Global Health Observatory (GHO):** This dataset made available by the Global Health Observatory, (http://www.who.int/en/) provides the number of deaths for several

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 13:58:07 UTC from IEEE Xplore. Restrictions apply.

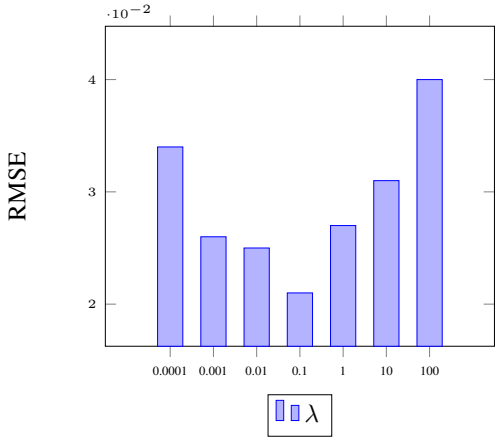

Fig. 4: RMSE on Google Flu w.r.t λ

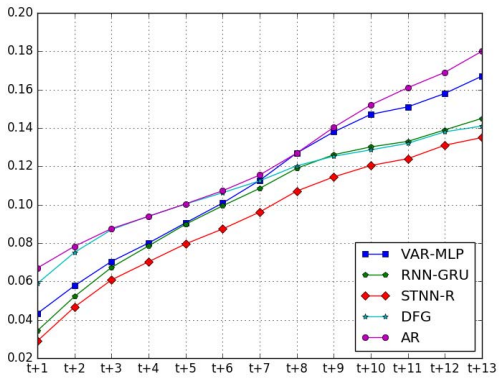

Fig. 5: RMSE on the Google Flu dataset at horizon $T+1$ to $T+13$

diseases. We picked 25 diseases corresponding to **25 different datasets**, each one composed of 91 time series corresponding to 91 countries (see table V-A). Results are averages over all the datasets. As for Google Flu, we extract binary relations W based on borders between the countries.

- **Geo-Spatial datasets:** The goal is to predict the evolution of geophysical phenomena measured on the surface of the Earth.
  The **Wind** dataset (www.ncdc.noaa.gov/) consists of hourly summaries of meteorological data. We predict wind speed and orientation for approximately 500 land stations on U.S. locations. In this dataset, the relations correspond to a thresholded spatial proximity between the series. Given a selected threshold value d, two sources are connected ($w_{i,j}=1$) if their distance is below d and not connected ($w_{i,j}=0$) otherwise.
  The **Pacific Sea Temperature (PST)** dataset represents gridded (at a 2 by 2 degrees resolution, corresponding to 2520 spatial locations) monthly Sea Surface Temperature (SST) on the Pacific for 399 consecutive months from January 1970 through March 2003. The goal is to predict future temperatures at the different spatial locations. Data were obtained from the Climate Data Library at Columbia University (http://iridl.ldeo.columbia.edu/). Since the series are organized on a 2D grid, we extract 8 different relations : one for each cardinal direction (north, north-west, west, etc...). For instance, the relation $north$, is associated to a binary adjacency matrix $W^{(north)}$ such that $W^{(north)}_{i,j}$ is set to 1 if and only if source j is located 2 degree at the north of source i (the pixel just above on the satellite image).
- **Car Traffic Forecasting:** The goal is to predict car traffic on a network of streets or roads. We use the **Beijing dataset** provided in [32], [33] which consists of GPS trajectories for ~ 10500 taxis during a week, for a total of 17 millions of points corresponding to road segments in Beijing. From this dataset, we extracted the traffic-volume aggregated on a 15 min window for 5,000 road segments. The objective is to predict the traffic at each segment. We connect two sources if they correspond to road segments with a shared crossroad.

For all the datasets but PST (i.e. Google Flu, GHO, Wind and Bejing), we defined the relational structure using a simple adjacency matrix W. Based on this matrix, we defined K different relations by introducing the powers of this matrix: $W^{(1)}=W$, $W^{(2)}=W\times W$, etc. In our setting K took values from 1 to 3 and the optimal value for each dataset has been selected during the validation process.

B. Forecasting Results

A quantitative evaluation of the different models and the baselines, on the different datasets is provided in table II. All the results are average prediction error for $T+1$ to $T+5$ predictions. The score function used is the Root Mean Squared Error (RMSE). A first observation is that STNN and STNN-R models which make use of prior spatial information significantly outperform all the other models on all the datasets. For example, on the challenging PST dataset, our models increase by 23% the performance of the GRU-RNN baseline. The increase is more important when the number of series is high (geo-spatial and traffic datasets) than when it is small (disease datasets). In these experiments, STNN-D is on par with RNN-GRU. The two models do not use prior information on spatial proximity. STNN makes use of a more compact formulation than RNN-GRU for expressing the series mutual dependency but the results are comparable. Vectorial AR logically improves on mono-variable AR (not shown here) and non linear MLP-VAR improves on linear VAR.

We also provide in table III the score for each of the 25 diseases of the GHO dataset. STNN-R obtained the best performance compared to STNN and STNN-D. It outperforms state-of-the-art methods in 20 over 25 datasets, and is very close to the RNN-GRU model on the 5 remaining diseases where RNN-GRU performs best. It thus shows that our model is

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 13:58:07 UTC from IEEE Xplore. Restrictions apply.

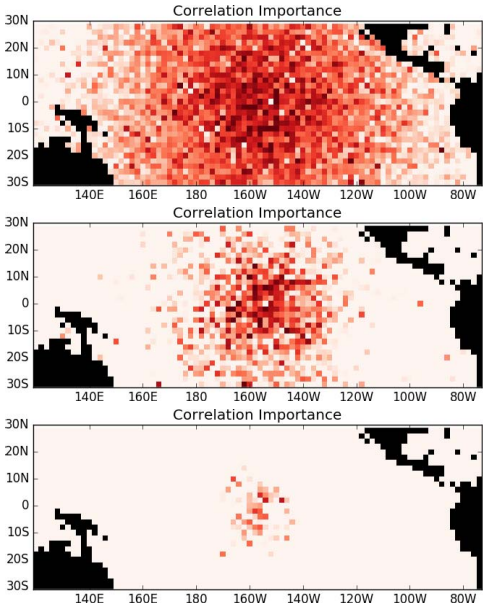

Fig. 6: Illustrations of correlations Γ discovered by the STNN-D model, with γ in $\{0.01, 0.1, 1\}$ (from top to bottom).

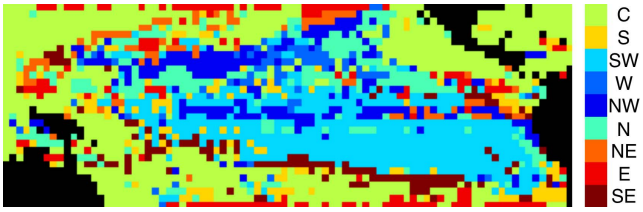

Fig. 7: Spatial correlations extracted by the STNN-R model on the PST dataset. The color of each pixel correspond to the principal relation extracted by the model.

able to benefit from the neighbour information in the proximity graph.

Figures 2 and 3 illustrate the prediction of *STNN-R* and *RNN-GRU* on the meteorology and on the oceanography datasets along with the ground truth. Clearly on these datasets, STNN qualitatively performs much better than RNNs by using explicit spatial information. STNN is able to predict fine details corresponding to local interactions when RNNs produce a much more noisy prediction. These illustrations are representative of the general behavior of the two models.

We also provide the performance of the models at different prediction horizons $T+1, T+2, ...T+13$ on figure 5 for the Google Flu dataset. Results show that STNN performs better than the other approaches for all prediction horizons and is thus able to better capture longer-term dependencies.

Figure 4 illustrates the RMSE of the STNN-R model when predicting at $T+1$ on the Google Flu dataset for different values of λ. One can see that the best performance is obtained for an average value of λ: low values corresponding to weak temporal

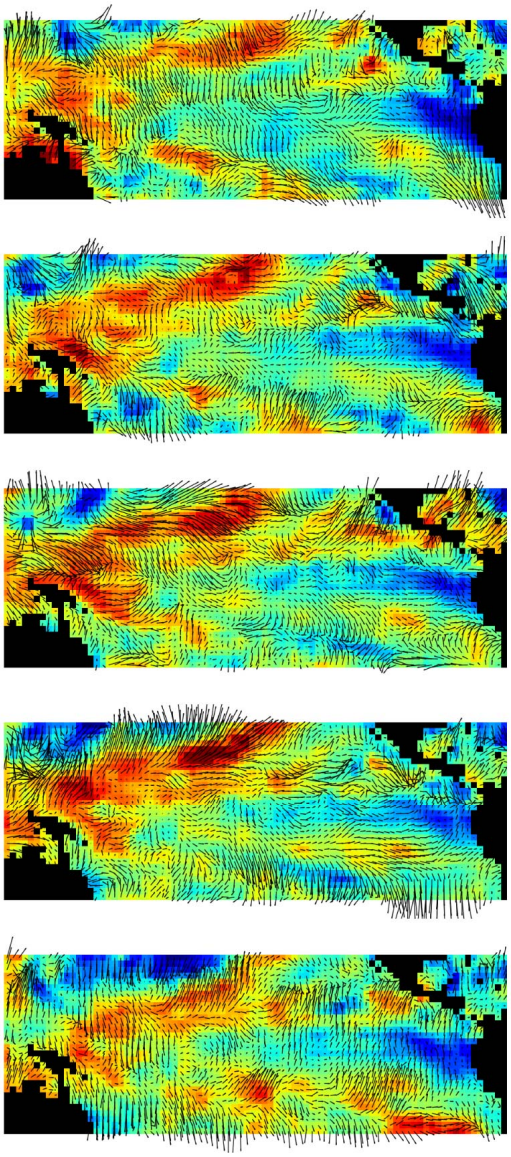

Fig. 8: Dynamic spatio-temporal relations extract from the PST dataset on the training set. The color represents the actual sea surface temperature. The arrows represent the extracted spatial relations that evolve through time.

constraints do not allow the model to learn the dynamicity of the series while high values degrade the performance of STNN.

C. Discovering the Spatial Correlations

In this subsection, we illustrate the ability of STNN to discover relevant spatial correlations on different datasets. Figure 6 illustrates the values of Γ obtained by STNN-D where no structure (e.g. adjacency matrix W) is provided to the model on the PST dataset. Each pixel corresponds to a particular time series and the figure shows the correlation $\Gamma_{i,j}$

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 13:58:07 UTC from IEEE Xplore. Restrictions apply.

discovered between each series j with a series i roughly located at the center of the picture. The darker a pixel is, the higher the absolute value of $\Gamma_{i,j}$ is (note that black pixels correspond to countries and not sea). Different levels of sparsity are illustrated from low (up) to high (down). Even if the model does not have any knowledge about the spatial organization of the series (no W matrix provided), it is able to re-discover this spatial organization by detecting strong correlations between close series, and low ones for distant series.

Figure 7 illustrates the correlations discovered on the PST dataset. We used as priors 8 types of relations corresponding to the 8 cardinal directions (South, South-West, etc...). In this case, STNN-R learns weights (i.e $\Gamma^{(r)}$) for each relation based on the prior structure. For each series, we plot the direction with the highest learned weight. The strongest direction for each series is illustrated by a specific color in the figure. For instance, a dark blue pixel indicates that the stronger spatial correlation learned for the corresponding series is the North-West direction. The model extracts automatically relations corresponding to temperature propagation directions in the pacific, providing relevant information about the spatio-temporal dynamics of the system.

The model can be adapted to different situations. Figure 8 represents the temporal evolution of the spatial relations on the PST dataset. For this experiment, we have slightly changed the STNN-R model by making the $\Gamma^{(r)}$ time dependent according to:

$$\Gamma^{(r)}_{t,j,i} = f_r(Z^i_t) \tag{11}$$

This means that with this modified model, the spatial relation weights depend on the current latent state of the corresponding series and may evolve with time. In the experiment, f_r is a logistic function. On figure 8, the different plots correspond to successive time steps. The color represent the actual sea surface temperatures, and the arrows represent the direction of the stronger relation weights $\Gamma^{(r)}_t$ among the eight possible directions (N, NE, etc). One can see that the model captures coherent dynamic spatial correlations such as global currents directions or rotating motions that gradually evolve with time.

VI. Conclusion

We proposed a new latent model for addressing multivariate spatio-temporal time series forecasting problems. For this model the dynamics are captured in a latent space and the prediction makes use of a decoder mechanism. Extensive experiments on datasets representative of different domains show that this model is able to capture spatial and temporal dynamics, and performs better than state of the art competing models. This model is amenable to different variants concerning the formulation of spatial and temporal dependencies between the sources.

For the applications, we have concentrated on forecasting (time based prediction). The same model could be used for interpolating (space based prediction or kriging) or for data imputation when dealing with time-series with missing values.

VII. Acknowledgments

Locust project ANR-15-CE23-0027-01, funded by Agence Nationale de la Recherche.

References

[1] N. A. C. Cressie and C. K. Wikle, *Statistics for spatio-temporal data*, ser. Wiley series in probability and statistics. Hoboken, N.J. Wiley, 2011.

[2] M. T. Bahadori, Q. R. Yu, and Y. Liu, "Fast multivariate spatio-temporal analysis via low rank tensor learning," in *Advances in Neural Information Processing Systems*, 2014, pp. 3491–3499.

[3] H. Koppula and A. Saxena, "Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation," in *Proceedings of ICML*, 2013.

[4] Y. Bengio, "Neural net language models," *Scholarpedia*, vol. 3, no. 1, p. 3881, 2008.

[5] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in neural information processing systems*, 2015, pp. 2962–2970.

[6] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[7] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 802–810.

[8] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *Proceedings of the 32nd ICML-15*, D. Blei and F. Bach, Eds., 2015.

[9] N. Kalchbrenner, A. v. d. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Unsupervised learning of video representations using lstms," in *Proceedings of the 34nd ICML-17*, 2017.

[10] M. L. Stein, *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.

[11] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[12] K. R. Muller, A. J. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, and V. Vapnik, "Using support vector machines for time series prediction," *Advances in kernel methodssupport vector learning*, 1999.

[13] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 240–254, 1994.

[14] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *IIIE ICASSP*, 2013.

[15] I. Sutskever, J. Martens, and G. E. Hinton, "Generating text with recurrent neural networks," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 2011.

[16] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[17] P. Mirowski and Y. LeCun, "Dynamic factor graphs for time series modeling," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 128–143.

[18] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, no. 2014, 2013.

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

[20] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2980–2988.

[21] R. G. Krishnan, U. Shalit, and D. Sontag, "Deep kalman filters," *arXiv preprint arXiv:1511.05121*, 2015.

[22] C. K. Wikle and M. B. Hooten, "A general science-based framework for dynamical spatio-temporal models," *Test*, vol. 19, no. 3, pp. 417–451, 2010.

[23] C. K. Wikle, "Modern perspectives on statistics for spatio-temporal data," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 1, pp. 86–98, 2015.

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 13:58:07 UTC from IEEE Xplore. Restrictions apply.

[24] H. Rue and L. Held, *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.
[25] M. Ceci, R. Corizzo, F. Fumarola, D. Malerba, and A. Rashkovska, "Predictive modeling of pv energy production: How to set up the learning task for a better prediction?" *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 956–966, 2017.
[26] G. Dornhege, B. Blankertz, M. Krauledat, F. Losch, G. Curio, and K. robert Muller, "Optimizing spatio-temporal filters for improving brain-computer interfacing," in *in NIPS*, 2005.
[27] Y. Ren and Y. Wu, "Convolutional deep belief networks for feature extraction of eeg signal," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 2850–2853.
[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[29] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th ICML*, 2013.
[30] S. Ben Taieb and R. Hyndman, "Boosting multi-step autoregressive forecasts," in *Proceedings of The 31st International Conference on Machine Learning*, 2014, pp. 109–117.
[31] G. Ganeshapillai, J. Guttag, and A. Lo, "Learning connections in financial time series," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 109–117.
[32] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proceedings of the 17th ACM SIGKDD*. ACM, 2011, pp. 316–324.
[33] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: driving directions based on taxi trajectories," in *Proceedings of the 18th SIGSPATIAL*. ACM, 2010, pp. 99–108.

APPENDIX

Hyper-parameters selection

We list in this section the set of hyper-parameters that have been chosen by cross-validation.

- For **RNN-models:**
  - Size of the hidden state = $(20, 50, 80, 150, 300, 500)$
- For **AR:**
  - Number of lags $R \in \{1, 2, 5, 10, 15, 25\}$
- For **VAR-MLP:**
  - Number of lags $R \in \{1, 2, 5, 10, 15, 25\}$
  - Size of the hidden state $\in \{20, 50, 80, 150, 300, 500\}$
- For **STNN:**
  - Dimension of the latent space $N \in \{5, 10, 20, 50, 80, 10\}$
  - Soft-constraint parameter $\lambda \in \{0.001, 0.01, 0.1, 1, 10\}$
  - Sparsity regulation $\gamma \in \{0.001, 0.01, 0.1, 1\}$
  - K value $\in \{1, 2, 3\}$ (for datasets composed of 1 unique relation)
- For **DFG:**
  - Size of the hidden states $\in \{10, 20, 50, 100, 300, 500\}$

Authorized licensed use limited to: BEIHANG UNIVERSITY. Downloaded on January 21,2021 at 13:58:07 UTC from IEEE Xplore. Restrictions apply.