Evaluation procedures for forecasting with spatio-temporal data

Mariana Oliveira[13], Luís Torgo[1,2,3], and Vítor Santos Costa[1,3]

[1] Porto University, Porto, Portugal,
mariana.r.oliveira@inesctec.pt
[2] Dalhousie University, Canada
[3] INESC TEC, Portugal

Abstract. The amount of available spatio-temporal data has been increasing as large-scale data collection (e.g., from geosensor networks) becomes more prevalent. This has led to an increase in spatio-temporal forecasting applications using geo-referenced time series data motivated by important domains such as environmental monitoring (e.g., air pollution index, forest fire risk prediction). Being able to properly assess the performance of new forecasting approaches is fundamental to achieve progress. However, the dependence between observations that the spatio-temporal context implies, besides being challenging in the modelling step, also raises issues for performance estimation as indicated by previous work. In this paper, we empirically compare several variants of cross-validation (CV) and out-of-sample (OOS) performance estimation procedures that respect data ordering, using both artificially generated and real-world spatio-temporal data sets. Our results show both CV and OOS reporting useful estimates. Further, they suggest that blocking may be useful in addressing CV's bias to underestimate error. OOS can be very sensitive to test size, as expected, but estimates can be improved by careful management of the temporal dimension in training.

1 Introduction

The problem of identifying whether a machine learning solution will perform well on unseen data is at the core of predictive analytics. Two questions must be addressed: **i)** are the evaluation metrics an appropriate fit to the application domain; and **ii)** does the evaluation procedure make the best use of valuable data to obtain accurate estimates of these metrics. This paper focuses on the second question, in the context of forecasting with geo-referenced time series data. The answer is not always obvious as spatio-temporal dependencies are present in the data.

Performance estimation procedures can be classified into two classes of methods, both widely used: *out-of-sample (OOS)* estimation and *cross-validation (CV)* strategies.

Hold-out validation is the simplest of OOS estimators. It operates by splitting the data into a training set – used to learn a model–, and a test set – used to

estimate the loss of the learned model in "unseen" data [12]. In the context of this study, only OOS procedures that respect an underlying order of the data are considered (e.g., in time series, the test set is always comprised of the more recent observations). These can also be called "last-block" procedures [3].

In CV, the total data is split several times into different training sets and test sets. Estimates of performance are obtained by averaging the losses over the several splits [30]. The use of different splits allows the whole data set to be used in the test set at least once. The data may be split in an exhaustive or partial manner, with partial splitting often being more computationally viable. The classical example of exhaustive splitting is leave-one-out cross-validation (LOOCV) where each observation plays the role of test set once. A common way to partially split the data is to divide it into K subsets of approximately the same size, and then having each subset successively used as test set – this strategy is referred to as K-fold CV [15]. However, standard CV procedures such as this assume that each test set is independent from the training set, which does not hold for many types of data sets, such as time series [2]. Several variations of CV procedures that do not require independence between sets have been proposed, with most of them being geared toward a time series setting [11,5,27]. Some of these methods have been proposed for spatio-temporal settings [7,18].

Our study aims at: **i)** providing a review of validation strategies in the presence of spatio-temporal dependencies; and **ii)** investigating the predictive ability of different cross-validation and out-of-sample strategies in a geo-referenced time series forecasting setting. To accomplish this goal we compare the loss estimated by different procedures against the loss incurred in previously withheld data. We consider artificial as well as real-world experimental settings.

2 Performance estimation with spatio-temporal dependence structures

Observations that have been made at different times and/or at neighbouring locations may be related through internal dependence structures within data sets, as there is a tendency for values of close observations to be more similar (or otherwise related) than distant ones.

Dependence between training and test sets may lead to overly optimistic estimates of the loss a model will incur when presented with previously unseen, independent data, and may also lead to structural overfitting and poor generalization ability [28]. In fact, more than one study has proven that CV overfits for choosing the bandwidth of a kernel estimator in regression [13,22].

2.1 Temporal dependence

Several performance estimation methods specifically designed to deal with temporal dependency have been proposed in the past.

In terms of OOS procedures in time series settings, decisions must be made regarding the split point between training/test sets, and how long a time-interval

to include in the training set, that is, the window settings (sliding/growing). Two approaches are worth mentioning: **a)** For *repeated time-wise holdout*, it is advised in [31] that holdout procedures should be repeated over different periods of time so that loss estimates are more robust. The selection of split points for each repetition of holdout may be randomized, with a window of preceding observations used for training and a fraction of the following instances used for testing. Training and test sets may potentially overlap across repetitions, similarly to random sub-sampling. These are also referred to as Monte Carlo experiments [32]; **b)** *Prequential evaluation* or interleaved-test-then-train evaluation is often used in data stream mining. Each observation (or block of non-overlapping observations) is first used to test and then to train the model [19] in a sequential manner. The term prequential usually refers to the case where the training window is growing, i.e., a block of observations that is used for testing in one iteration will be merged with all previous blocks and used for training in the next iteration.

Four alternatives to standard CV proposed for time series should be highlighted: **a)** *Modified CV* is Similar to K-fold CV, except that l observations preceding and following the observation(s) in the test set are discarded from the training set after shuffling and fold assignment [11]. Also referred to as non-dependent cross-validation in [3]; **b)** *Block CV* is a procedure similar to K-fold CV where, instead of the observations being randomly assigned to folds, each fold is a sequential, non-interrupted time series [29]; **c)** *h-block CV* is based on LOOCV, except h observations preceding and following the observation in the test set are removed from the training set [5], and **d)** *hv-block CV* is a modification of h-block CV where, instead of having single observations as test sets, a block of v observations preceding and following each observation is used for testing (causing test sets to overlap), with h observations before and after each block being removed from the training set [27].

Note that while in all types of block-CV, each test set is composed of a sequential non-interrupted time series (or a single observation), a fold in modified CV will almost certainly have non-sequential observations. If K is set to the number of observations in modified CV, then it works the same as h-block CV. Moreover, note that only hv-block CV allows test sets to overlap.

A number of empirical studies compare estimation methods for time series. Bergmeir *et al.* [3,4] suggest that cross-validation (in particular, hv-block CV) may have advantage over OOS approaches, especially when samples are small and the series stationary. Cerqueira *et al.* [9] indicate that, although this might be valid for synthetic time series, the same might not apply in real-world scenarios where methods preserving the order of the series (such as OOS Monte Carlo) seem to better estimate loss in withheld data. Mozetic *et al.* [20] reinforce the notion that blocking is important for time-ordered data.

2.2 Spatial dependence

A major change when switching from temporal dependence to spatial dependence is that there is not a clear unidirectional ordering of data in 2D- or 3D- space

as there is in time. This precludes using prequential evaluation strategies in the spatial domain. However, other strategies can be adapted quite straightforwardly to deal with spatial dependence.

Cross-validation approaches seem to be most commonly used in spatial settings. To avoid the problems arising from spatial dependence, block CV is often adopted. As in the temporal case, blocks can be designed to include neighbouring geographic points, forcing testing on more spatially distant records, and thus decreasing spatial dependence and reducing optimism in error estimates [33]. Methods that would correspond to h-block or hv-block CV are usually referred to as "buffered" CV in the spatial domain as a geographic vicinity of the testing block is removed from the training set.

The validity of these procedures was empirically tested by Roberts *et al.* in [28]. The authors find that block CV (with a block size substantially larger than residual autocorrelation) and "buffered" LOOCV (a spatial version of h-block CV, with h equivalent to the distance at which residual autocorrelation is zero) better approximate the error obtained when predicting onto independent simulations of species abundances data depending on spatially autocorrelated "environmental" variables.

2.3 Spatio-temporal dependence

When both spatial and temporal structures are present in the data, authors often resort to one of the procedures described in previous sections, effectively treating the data as if it was spatial-only (e.g., [16]) or temporal-only (e.g., [1,8]) for evaluation purposes. Others, while treating the problem mostly from a temporal perspective, then make an effort towards breaking down the results across space (e.g., [21]), or vice-versa (e.g., [7]), without the evaluation procedure itself being specifically designed to accommodate this.

In [28], no experimental results are presented specifically for spatio-temporal data, but there is a mention of data often being structured in both space and time in the context of avoiding extrapolation in cross-validation. When models are only meant to interpolate, the provided intuitions are that blocks should be no larger than necessary, models should be trained with as much data as possible, and predictors should be equally represented across blocks or folds. While conservatively large blocks can help avoid overly optimistic error estimates, the potential for introducing extrapolation is also increased. It is suggested that this effect may be mitigated by using "optimised random" or systematic (patterned) assignment of blocks to folds. Roberts *et al.* [28] also provide a general guide on blocking for CV, proposing the following five steps: assess dependence structures in the data, determine prediction objectives, block according to objectives and structure, perform cross-validation, and make "final" predictions.

Recent work by Meyer *et al.* [18] highlights how, for spatio-temporal interpolation problems, the results of conventional CV differ from the results of what they call "target-oriented" CV (versions of CV that address each and/or both dimensions, namely, "leave-location-out", "leave-time-out" and "leave-location-and-time-out"). The authors attribute the lower error estimated by conventional

CV to spatio-temporal over-fitting of the models and propose a forward feature selection procedure to improve interpolation results.

The applicability of solutions that consider the temporal and/or spatial autocorrelation is worth exploring, but the optimal strategy will depend on the modeling goal. It is important to make the distinction, as previous works have, between interpolation and forecasting problems. Unlike previous work on spatio-temporal data, the focus of this study is on forecasting, meaning that the aim is to make predictions about the future/new locations. Even after that is established, it may still be the case that the best evaluation procedure when the goal is to make predictions about unseen locations might differ from the best strategy when the aim is to make predictions in known sites.

3 Experiments

The different estimation procedures being compared are presented in Sec. 3.1. We first investigate their performance on datasets of randomly-generated artificial spatio-temporal data, as they provide a foundation for understanding the real-world case studies presented in Sec. 3.2. Section 3.3 describes the experimental design. Code for replication of these experiments is freely available[1].

3.1 Estimation procedures

The estimators tested here included time-wise holdout methods (one-time, H, Monte Carlo, MC), cross-validation (CV), and prequential evaluation (P).

Train/test allocation strategies Table 1 summarises the different train/test assignment procedures used for CV and prequential evaluation methods.

Methods to assign observations into cross-validation folds that were tested include: standard CV, where instances are randomly assigned to folds, ($tRsR$), ignoring both dependency dimensions; time-sliced CV, where the spatial dimension is ignored and time-slices are assigned to folds randomly ($tRsA$); spatial block CV (also referred to as "leave-location-out" CV), where the temporal dimension is ignored and spatial blocks are assigned to folds either randomly ($tAsR$), in contiguous blocks ($tAsC$), or in a systematic, checkered pattern ($tAsS$).

When time is divided into blocks, prequential evaluation can also be applied. In this scenario, ($tBsA$), also referred to as "leave-time-out" CV, fold assignment ignores the spatial dimension. If space is also divided into blocks, then different types of spatio-temporal CV can be achieved by having the spatial assignment of folds be either random ($tBsR$), in contiguous blocks ($tBsC$), or in a (systematic) checkered pattern ($tBsS$).

Note that in what we call prequential evaluation, temporal order is always respected even when dividing data into spatio-temporal blocks, i.e., if a block in

[1] https://github.com/mrfoliveira/Evaluation-procedures-for-forecasting-with-spatio-temporal-data

space-time is used for testing, then only blocks with previous time-stamps are used for training. Whether the spatial region in the test set is included in the training set is optional (rmS indicates that spatio-temporal data from the past but in the spatial region of the test set are not used in training). Moreover, the number of previous blocks in time used for training can be either fixed – sliding window (slW), or increase at each blocked time step – growing window (grW).

Table 1. Cross-validation and prequential evaluation fold assignment procedures

		Time	Space		
Cross-validation	Standard	random	random	tRsR	• † ‡
	Time-sliced		all	tRsA	
	Spatial block	all	random block	tAsR	•
	Checkered spatial block		systematic	tAsS	
	Contiguous spatial block		contiguous	tAsC	•
Prequential evaluation	Time block	block	all	tBsA	†
	Spatio-temporal block		random block	tBsR	‡
	Spatio-temporal checkered block		systematic	tBsS	
	Spatio-temporal contiguous block		contiguous	tBsC	

† Time-buffered CV variation included

• Space-buffered CV variation included

‡ Space-time buffered CV variation included

Buffered CV Methods that remove a block of observations in the neighbourhood of the test set (in the temporal and/or spatial dimensions) from the training set have also been considered.

In the case of standard CV, for each instance in the test set, a number of past and future observations at that location are removed and/or past observations within a certain distance from the location are removed (CV-T, CV-S or CV-ST). This is akin to modified CV mentioned earlier in a time series context. The same process can be applied to spatio-temporal CV. In that scenario, if the buffer is set to the maximum distances between any two points in space/time (CV-STM), the result is what is called "leave-location-and-time-out" CV.

When time block CV is used, then a number of previous and future observations are removed around the test set (CV-T). This is similar to hv-block CV. However, while hv-block CV is repeated for each instance of the whole set (therefore including overlapping test sets), the procedure is only repeated here for each non-overlapping block of sequential time.

In spatial random or contiguous block CV, a spatial buffer can be applied, so that locations within a pre-defined spatial distance of the test set are removed from the training set (CV-S). This is, again, similar to hv-block CV in space.

3.2 Datasets

As previously mentioned, both artificially generated and real-world data sets were used for this study.

Artificial datasets Artificial data was generated by stationary spatio-temporal auto-regressive moving average (STARMA) models as proposed in [24] and implemented in *R* package *starma* [10].

The models are denoted by $STARMA(p_{\lambda_1\lambda_2\ldots\lambda_p}, q_{m_1m_2\cdots_p})$ where p is the autoregressive order, q is the moving average order, λ_l is the spatial order of the k^{th} autoregressive term, m_k is the spatial order of the k^{th} moving average term. If $q = 0$, then $STAR(p_{\lambda_1\ldots\lambda_p})$ will suffice; if $p = 0$, then it may be denoted by $STMA(q_{m_1\cdots_p})$. Non-linear versions of STAR models, $NLSTAR(p_{\lambda_1\ldots\lambda_p})$, are generated by applying a non-linear function at each autoregressive step (similar to what is done in [3] to obtain non-linear AR models).

In datasets generated by a $STAR(2_{10})$ model, a value measured at location i and time t will be directly influenced by the values of location i and of it's first-degree neighbours at time $t-1$, and by the values of location i at time $t-2$. Note that neighbours of lower order must be considered "closer" than neighbours of higher order (according to some metric of distance).

In this study, for each model of type STARMA (with $p = q$), STMA, STAR, and NLSTAR, two sets of coefficients of each order 2_{10}, 2_{01} and 2_{11} are generated randomly (within intervals likely to respect stationarity conditions) until the resulting STARMA models are stationary. In the case of NLSTAR, a non-linear function is also randomly selected from a pre-defined set. Then, using grids of 10×10 and 22×22 equally spaced locations, data is generated with time series lengths of 250 and 400. However, after this step, the first 100 observations at each location are discarded in an effort to avoid dependence on initial conditions; outer locations are ignored so each used location has information for its four first order neighbours – top, bottom, left and right. Thus, 150 and 300 observations on 8×8 and 20×20 grids are kept for forecasting performance analysis. For details on STARMA models and the data generation process, consult the Appendix.

Spatio-temporal embedding In order to apply standard regression techniques to the spatio-temporal forecasting problem, the generated data sets have to be transformed in some way so each instance has a set of predictors. A simple way to do this is by spatio-temporal embedding, i.e., by using previous values measured at the given location and it's neighbours as predictors. The order of spatio-temporal embedding can be denoted in the same way as the $STARMA$ order. All artificially generated data sets were embedded with order 3_{110}. In total, 96 artificial data sets were generated and embedded.

Real-world datasets Seventeen variables from seven different real-world data sources were used as independent univariate data sets for experimental validation of the performance evaluation procedures. The measured variables describe

environmental monitoring, from air pollution to climate and soil characteristics. A summary of the characteristics of each data set can be found in Tab. 2. The size of the data sets varies from small networks of 20 sensors to larger networks of 900 geolocations. Though most sensor networks are irregularly distributed in space, one of them forms a regular grid of 0.5×0.5 degrees of longitude/latitude. The data sets also vary in terms of time series size (from 280 time points to over 11k) and sampling frequency (from hourly to monthly). About half of the variables were measured at every point in time and space, with no missing values. However, for others, only a percentage of location and time-stamp pairs (from 39% to 74%) have available values, due to, for instance, some sensors only being installed later in the measurement period.

Table 2. Real-world data sets

			Time		Locations		Total		
Data set	#	Variables	#IDs	frequency	#IDs	distribution	#	% available	Source
MESA Air Pollution	1	NO_X concentration	280	bi-weekly	20	irregular	5.6k	100	[25] [2]
NCDC Air Climate	2	precipitation, solar energy	105	monthly	72	irregular	7.6k	100	[25][2]
TCE Air Climate	3	ozone concentration, air temperature, wind speed	330	hourly	26	irregular	8.6-9.4k	100	[25][2]
COOK Agronomy Farm	3	water content, temperature, conductivity	729	daily	42	irregular	22-23k	73-74	[17,14][3]
SAC Air Climate	1	air temperature	144	monthly	900	regular	130k	100	[25][2]
RURAL airBase	1	PM10 concentration	4382	daily	70	irregular	149k	49	[23][3]
BEIJ Beijing UrbanAir	6	PM25, PM10 & NO_X concentration, air temperature, humidity	11357	hourly	36	irregular	404-409k	39-41	[35][4]

Spatio-temporal indicators In order to compare performance, a learning approach had to be selected that would work with the different data set characteristics. Unlike the artificial data sets, most real-world sensor networks are not distributed in a regular grid, so the simple spatio-temporal embedding used for the artificial data sets seemed over-simplistic. The approach adopted instead was the one proposed in [21], using as predictors a temporal embed of values measured at the location; spatio-temporal indicators built by calculating summary statistics from the neighbouring observations within three dataset-specific boundaries of spatio-temporal distance, and ratios between the indicators of spatio-temporal

[2] Downloaded from: http://www.di.uniba.it/~appice/software/COSTK/index.htm
[3] Loaded from *R* packages *GSIF* (0.5-4) and *spacetime* (1.2-1).
[4] Downloaded from: https://www.microsoft.com/en-us/research/publication/u-air-when-urban-air-quality-inference-meets-big-data/

neighbourhoods of increasing radius. The temporal embed size was set to 7, resulting in a total of 20 predictors.

Missing data Some of the data sets have missing data, either due to failures in data acquisition or due to sensors being set up at later times. After calculating the predictors but before any experiments are carried out, all columns that have 20% or more of their data missing from the first 80% of time-points, are discarded as they should not be very useful predictors. The remaining missing data is dealt with as follows: first, any rows that have too many predictors missing (set at 20% of columns) are discarded from the training set; then, missing values for both the training and test sets are imputed as the median of that column in the set.

3.3 Experimental design

For each data set: **1)** The data is divided into an in-set and out-set. This is performed time-wise, so that the out-set consists of a percentage of the most recent observations; **2)** A regression model is trained in the in-set and tested on the out-set. The error on the out-set is considered to be the "gold standard" error that estimation methods should be able to estimate accurately; **3)** Several error estimation methods (cross-validation, prequential and out-of-sample methods), applied exclusively on data from the in-set, are used to approximate the "gold standard". The differences between the "gold standard" error and the error estimated by each estimation methodology can be compared over all data sets and learning model pairs.

Train/test sizing The in-set was set to be 80% of the time-points. When using cross-validation or prequential evaluation on the in-set, 16 folds were used for artificial data and 9 folds for real data. When using OOS procedures on the in-set, the splits are always made time-wise. For holdout, estimations were made with test sizes of 20% (same proportion as the out-set) and 6%/9% for artificial/real in-set data (the proportions used in the last block of time-block CV).

Note that the data set is divided into the same number (16 or 9) of equally-sized folds across all variations of CV. In the interest of fairness, the test size of time-wise holdout was defined to correspond to the size of one fold in CV. All of these methodologies use the whole given in-set to make estimates. However, time-wise Monte Carlo estimations, by definition, use only a fraction of the data set for each iteration – meaning the sizing of these competing procedures can never be made entirely "fair". The option taken was to keep the proportion between train and test sizes the same as that used in CV, i.e., the percentages used for training and testing in Monte Carlo correspond to the estimation on the last block of a 16-fold or 9-fold time block CV performed on 50% or 60% of the in-set. Thus, Monte Carlo estimations were averaged over 16 repetitions with training (testing) performed on 47% (3%) and 56% (4%) of the in-set for artificial data, and averaged over 9 repetitions of training (testing) on 44% (6%) and 53% (7%) of real data. Buffer sizes are set to the highest embed size or spatio-temporal neighbourhood radius.

Learning models The process is repeated over each data set using two different learning algorithms: a linear regression model, *LM* (*R* package *stats* [26]) and a random forest, *RF* (*R* package *ranger* [34]).

Error metrics The error of learning algorithms is measured by Normalized Mean Absolute Error (NMAE), defined by Eq. 1 where z is the observed value, $\hat{z}$ is the prediction, and $\bar{z}$ is the mean of Z. By opting for a normalized metric instead of the more widely used MAE, comparisons between error estimation methods across data sets can be made more easily.

$$NMAE = \frac{\sum_{i=0}^{n} |\hat{z} - z|}{\sum_{i=0}^{n} |z - \bar{z}|} \tag{1}$$

4 Results

The estimation error is defined as the difference between the error estimated by a procedure using the in-set, *Est*, and the "gold standard" error incurred on the out-set, *Gold*, $Err = Est - Gold$. Note that experiments with methods that rely on non-random spatial blocking were not carried out using real-world data sets due to issues arising from irregular spatial distributions. Time-buffering without time-blocking in real-world scenarios caused issues related with buffer size/neighbourhood radius. Results for variations of prequential evaluation using sliding window and/or removing locations in the test set from the training set are not reported as they were consistently out-performed by their growing window counterparts (though the difference was not statistically significant).

4.1 Median errors

Figures 1 and 2 show the distribution of estimation errors for artificial and real-world data sets. The sign of the median error indicates whether the procedure tends to underestimate the error meaning it is overly optimistic (negative median error), or overestimate it (positive median error).

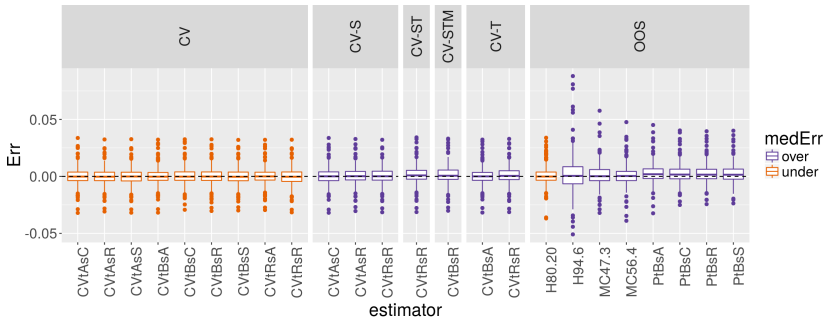


Fig. 1. Box plots of estimation errors incurred by cross-validation and out-of-sample procedures on 96 artificial data sets using 2 learning algorithms.

In Fig. 1, all procedures appear centered around zero. However, most cross-validation procedures under-estimate the error in more than half the cases, even when using some form of block CV. This effect is mitigated when a type of buffering is applied (either temporal, spatial or spatio-temporal). Most OOS procedures overestimate the error in more than half the cases, with the exception of holdout at 80%.

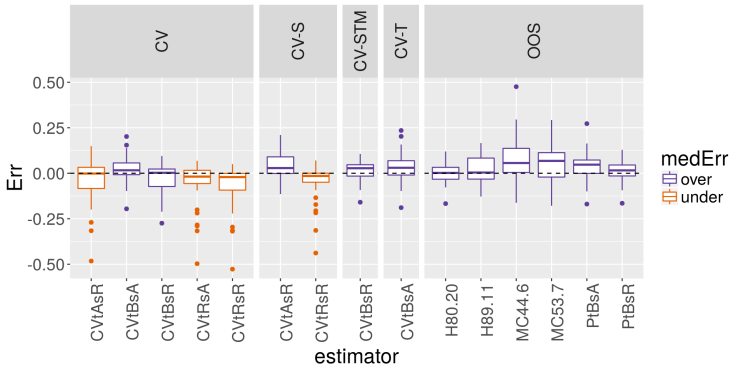


Fig. 2. Box plots of estimation errors incurred by cross-validation and out-of-sample procedures on 17 real world data sets using 2 learning algorithms.

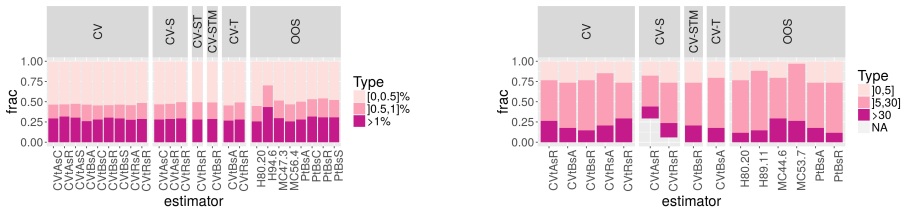
Figure 2 shows significant differences between procedures. Is is important to note that standard CV ($CVtRsR$) under-estimates the error in over 75% of cases. We observe this problem even after applying a spatial buffer. Note that spatial-buffered CV estimates were not obtained for a fraction of real data sets due to problems associated with the irregularity of sensor network locations.

Spatial block CV ($CVtAsR$) and time-sliced CV ($CVtRsA$) are also overly optimistic in their error estimates. However, OOS procedures and variations of CV using time-blocks and/or time-buffers seem to be less prone to under-estimate the error.

4.2 Relative errors

Another useful metric to analyse is the relative error as defined by $RelErr = |Est - Gold|/Gold$. Figure 3 shows the distribution of low, moderate and high errors. The binning is somewhat arbitrary but chosen so that comparisons might be useful. In the real-world case, relative errors are generally higher so bins were chosen accordingly. Possible explanations for the lower relative errors found for artificial data sets when compared to the real-world case include the absence of missing data, the regularity of the grids and stationarity of the underlying data generation process.

In Fig. 3a, holdout ($H94.6$) stands out as the estimation method with the lowest percentage of low relative errors. In real-world scenarios (Fig. 3b), stan-



(a) Artificial (b) Real-world

Fig. 3. Bar plots of relative estimation errors incurred by cross-validation and out-of-sample procedures on 96 artificial and 17 real-world data sets using 2 learning algorithms. Note the different legends.

dard CV ($CVtRsR$) has one of the highest proportions of severe relative error, alongside spatial CV ($CVtAsR$) and MC procedures. MC procedures may be at a disadvantage due to using smaller fractions of the in-set for error estimation.

4.3 Absolute errors

Finally, we present results concerning the absolute errors incurred by estimation procedures, that is, $AbsErr = |estimated - real|$. The mean ranks for artificial data sets can be found in Tab. 3. Time-block CV either time-buffered ($CVtBsA\_T$) or plain ($CVtBsA$) are two procedures that can be found within the top 5 average ranks for both the linear and random forests learning models. Within OOS procedures, holdout ($H80.20$) and Monte Carlo ($MC56.4$) can be found in the top 3 average ranks for both learning models.

Table 4 shows average ranks for real-world data sets. Spatio-temporal block CV ($CVtBsR$) is within the top 5 average rank of both learning models, alongside space-buffered standard CV ($CVtRsR\_S$). The top 3 OOS procedures are consistently spatio-temporal block prequential evaluation ($PtBsR$), holdout ($H80.20$) and time-block prequential evaluation ($PtBsA$).

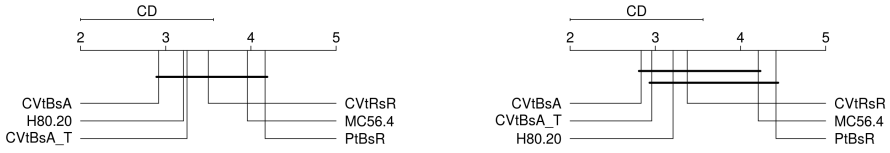
Only the aforementioned procedures, along with any other method that appears as the best for a certain learning model, and standard CV, are considered for statistical significance testing. The Friedman-Nemenyi test is applied, with estimation procedures used as the "classifiers" or "treatments" (using R package *scmamp* [6]). Since there is an assumption that the data sets should be independent, separate Friedman tests were carried out for the results obtained by linear and random forest learning models. Moreover, a test is performed for each subset of 24 artificial data sets with the same grid and time series size, since the same STARMA coefficients were re-used across different data sizes. Figures 4 and 5 show critical difference diagrams for a subset of artificial data sets and all the real-world data sets. In most cases, no significant difference between estimation procedures was found at a 5% confidence level. However, a significant difference was found for the smaller artificial data sets as seen in Fig. 4b.

Table 3. Average ranks of absolute errors incurred by cross-validation and out-of-sample procedures when estimating performance on 96 artificial data sets. Best results in bold.

	CVtAsC	CVtAsC_S	CVtAsR	CVtAsR_S	CVtAsS	CVtBsA	CVtBsA_T	CVtBsC
LM	8.35	8.56	8.07	**7.57**	8.64	8.16	8.02	8.41
RF	8.47	8.57	9.01	8.44	8.54	7.95	**7.55**	8.67
	CVtBsR	CVtBsR_STM	CVtBsS	CVtRsA	CVtRsR	CVtRsR_S	CVtRsR_ST	CVtRsR_T
LM	8.67	8.11	8.73	8.29	9.21	9.07	8.86	9.27
RF	8.50	8.96	8.30	7.93	9.07	8.75	8.88	8.42
	H80.20	H94.6	MC47.3	MC56.4	PtBsA	PtBsC	PtBsR	PtBsS
LM	4.23	6.21	4.24	**3.99**	4.09	4.33	4.54	4.36
RF	**4.02**	5.95	4.34	4.20	4.51	4.43	4.14	4.42

Table 4. Average ranks of absolute errors incurred by cross-validation and out-of-sample procedures when estimating performance on 17 real-world data sets. Best results in bold.

	CVtAsR	CVtAsR_S	CVtBsA	CVtBsA_T	CVtBsR	CVtBsR_STM	CVtRsA	CVtRsR	CVtRsR_S
LM	4.53	7.06	5.29	5.35	**3.94**	5.12	4.82	4.35	4.53
RF	5.00	5.88	**4.00**	4.41	4.41	4.41	5.47	6.59	4.82
	H80.20	H89.11	MC44.6	MC53.7	PtBsA	PtBsR			
LM	**2.47**	3.71	4.47	4.35	3.47	2.53			
RF	3.12	3.35	4.12	4.41	3.18	**2.82**			

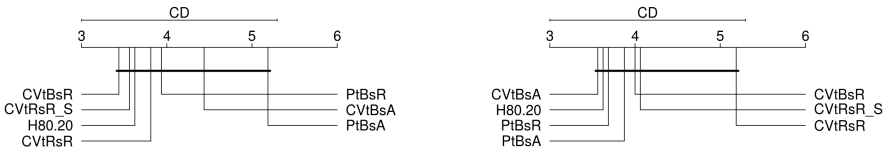


(a) LM (b) RF

Fig. 4. Critical difference diagram according to Friedman-Nemenyi test (at 5% confidence level) for a subset of estimation procedures using 24 artificial data sets (64×64 grid; 150 time-points each).

5 Conclusion

The problem of how to properly evaluate spatio-temporal forecasting methods is still an open one. Previous studies have empirically shown that dependence between observations negatively impacts performance estimation using standard error estimation methods like cross-validation for time series [3,4,9], time-ordered



(a) LM (b) RF

Fig. 5. Critical difference diagram according to Friedman-Nemenyi test (at 5% confidence level) for a subset of estimation procedures using real data sets.

Twitter data [20], spatial and phylogenetic data [28], and spatio-temporal interpolation [18].

In this paper, an extensive empirical study of performance estimation for forecasting problems using both artificially generated and real-world spatio-temporal data sets is provided. First, we observe that most often error estimates are reasonably accurate. Standard CV does have problems: it underfits and it exhibits a number of outliers of severe error underestimation. Moreover, though the best estimator in terms of absolute difference to the gold standard is not always the same, most top-performers block the data set in time. This is in line with previous research on time-ordered data [3,20]. Indeed, for artificial data sets, time-buffered time-block CV is one of the best in terms of approximating the gold standard error while also avoiding being overly optimistic in the estimates. For real-world datasets, spatio-temporal block CV and, when using random forests, time-block CV (this time without the buffer) not only approximate the error better than other methods, they also mostly avoid being overly optimistic about errors. Note that the fact that time-buffered time block CV did not perform as well in real-world data sets might have to do with buffer size parametrization. Out-of-sample procedures, in general, did not do as well in terms of absolute difference to the gold-standard, but they did tend to avoid underestimation of the error in almost all cases which might still be seen as an advantage over cross-validation. These results seem to point to the temporal dimension being more important to respect when evaluating spatio-temporal forecasting methods.

There is some bias in the experimental design, but results are still fairly consistent and some issues can be addressed in future work. Varying the in-set/out-set ratio, and setting the "gold standard" as forecasting future observations in new locations (instead of forecasting for known locations only) are two future settings of interest. Moreover, the effect of train/test and buffer sizes on the estimation methods should be analysed. It would also be interesting to control for the effect of including outer locations and/or introducing missing data in artificial data. Moreover, in the case of real-world (or artificial) data sets with

irregular grids, solutions to contiguous assignment of spatial blocks should be explored, possibly using quadtrees.

Acknowledgments This work is partially funded by the ERDF through the COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT as part of project UID/EEA/50014/2013. Mariana Oliveira is supported by a FCT/MAPi PhD research grant (PD/BD/128166/2016). Vítor Santos Costa is supported by the project POCI-01-0145-FEDER-016844.

References

1. Appice, A., Pravilovic, S., Malerba, D., Lanza, A.: Enhancing regression models with spatio-temporal indicator additions. In: Congr. Ital. Assoc. Artif. Intell. pp. 433–444. Springer (2013)
2. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. Stat. Surv. **4**, 40–79 (2010). https://doi.org/10.1214/09-SS054
3. Bergmeir, C., Benítez, J.M.: On the use of cross-validation for time series predictor evaluation. Inf. Sci. (Ny). **191**, 192–213 (2012). https://doi.org/10.1016/j.ins.2011.12.028
4. Bergmeir, C., Costantini, M., Benítez, J.M.: On the usefulness of cross-validation for directional forecast evaluation. Comput. Stat. Data Anal. **76**, 132–143 (2014). https://doi.org/10.1016/j.csda.2014.02.001
5. Burman, P., Chow, E., Nolan, D.: A cross-validatory method for dependent data. Biometrika **81**(2), 351–358 (1994). https://doi.org/10.1093/biomet/81.2.351
6. Calvo, B., Santafé Rodrigo, G.: scmamp: Statistical comparison of multiple algorithms in multiple problems. The R Journal, Vol. 8/1, Aug. 2016 (2016)
7. Carroll, S.S., Cressie, N.: Spatial modeling of snow water equivalent using covariances estimated from spatial and geomorphic attributes. J. Hydrol. **190**(1-2), 42–59 (1997). https://doi.org/10.1016/S0022-1694(96)03062-4
8. Ceci, M., Corizzo, R., Fumarola, F., Malerba, D., Rashkovska, A.: Predictive modeling of pv energy production: How to set up the learning task for a better prediction? IEEE T. Ind. Inform. **13**(3), 956–966 (2017)
9. Cerqueira, V., Torgo, L., Smailovi, J., Mozeti, I.: A comparative study of performance estimation methods for time series forecasting. In: Int. Conf. Data Sci. Adv. Anal. (DSAA). pp. 529–538 (2017). https://doi.org/10.1109/DSAA.2017.7
10. Cheysson, F.: starma: Modelling Space Time AutoRegressive Moving Average (STARMA) Processes (2016), `https://cran.r-project.org/package=starma`
11. Chu, C.K., Marron, J.S.: Comparison of two bandwidth selectors with dependent errors. Ann. Stat. **19**(4), 1906–1918 (1991)
12. Devroye, L., Wagner, T.: Distribution-free performance bounds for potential function rules. IEEE Trans. Inf. Theory **25**(5), 601–604 (1979)
13. Diggle, P.: Analysis of longitudinal data. Oxford University Press (2002)
14. Gasch, C.K., Hengl, T., Gräler, B., Meyer, H., Magney, T.S., Brown, D.J.: Spatio-temporal interpolation of soil water, temperature, and electrical conductivity in 3D+ T: The Cook Agronomy Farm data set. Spat. Stat. **14**, 70–90 (2015)
15. Geisser, S.: The predictive sample reuse method with applications. J. Am. Stat. Assoc. **70**(350), 320–328 (1975)

16. Haberlandt, U.: Geostatistical interpolation of hourly precipitation from rain gauges and radar for a large-scale extreme rainfall event. J. Hydrol. **332**(1-2), 144–157 (2007). https://doi.org/10.1016/j.jhydrol.2006.06.028
17. Hengl, T.: GSIF: Global Soil Information Facilities (2017), `https://cran.r-project.org/package=GSIF`
18. Meyer, H., Reudenbach, C., Hengl, T., Katurji, M., Nauss, T.: Improving performance of spatio-temporal machine learning models using forward feature selection and target-oriented validation. Environ. Model. Softw. **101**, 1–9 (2018). https://doi.org/10.1016/j.envsoft.2017.12.001
19. Modha, D.S., Masry, E.: Prequential and cross-validated regression estimation. Mach. Learn. **33**(1), 5–39 (1998). https://doi.org/10.1109/ISIT.1998.708964
20. Mozetič, I., Torgo, L., Cerqueira, V., Smailović, J.: How to evaluate sentiment classifiers for Twitter time-ordered data? PLoS One **13**(3), 1–20 (2018). https://doi.org/10.1371/journal.pone.0194317
21. Ohashi, O., Torgo, L.: Wind speed forecasting using spatio-temporal indicators. In: Proc. 20th Eur. Conf. Artif. Intell. pp. 975–980. IOS Press (2012)
22. Opsomer, J., Wang, Y., Yang, Y.: Nonparametric regression with correlated errors. Stat. Sci. **16**(2), 134–153 (2001). https://doi.org/10.1214/ss/1009213287
23. Pebesma, E.: spacetime: Spatio-temporal data in R. J. Stat. Softw. **51**(7), 1–30 (2012), `http://www.jstatsoft.org/v51/i07/`
24. Pfeifer, P.E., Deutsch, S.J.: A three-stage iterative procedure for space-time modeling. Technometrics **22**(1), 35—-47 (1980)
25. Pravilovic, S., Appice, A., Malerba, D.: Leveraging correlation across space and time to interpolate geophysical data via CoKriging. Int. J. Geogr. Inf. Sci. **32**(1), 191–212 (2018). https://doi.org/10.1080/13658816.2017.1381338
26. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2017), `https://www.r-project.org/`
27. Racine, J.: Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. J. Econom. **99**(1), 39—-61 (2000)
28. Roberts, D.R., Bahn, V., Ciuti, S., Boyce, M.S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J.J., Schröder, B., Thuiller, W., et al.: Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography **40**(8), 913–929 (2017)
29. Snijders, T.A.B.: On cross-validation for predictor evaluation in time series. In: Proc. Work. Model Uncertain. its Stat. Implic. pp. 56–69 (1988). https://doi.org/10.1007/978-3-642-61564-1_4
30. Stone, M.: Cross-validatory choice and assessment of statistical predictions. J. R. Stat. Soc. B pp. 111–147 (1974)
31. Tashman, L.J.: Out-of-sample tests of forecasting accuracy : an analysis and review. Int. J. Forecast. **16**(4), 437—-450 (2000)
32. Torgo, L.: Data mining with R: learning with case studies. CRC press (2016)
33. Trachsel, M., Telford, R.J.: Estimating unbiased transfer-function performances in spatially structured environments. Clim. Past **12**(5), 1215–1223 (2016)
34. Wright, M.N., Ziegler, A.: ranger: A fast implementation of random forests for high dimensional data in C++ and R. J. Stat. Softw. **77**(1), 1–17 (2017). https://doi.org/10.18637/jss.v077.i01
35. Zheng, Y., Liu, F., Hsieh, H.P.: U-Air: when urban air quality inference meets big data. In: Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. pp. 1436–1444. KDD '13, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2487575.2488188