RNN-Based Counterfactual Time-Series Prediction

Jason Poulos[†]

University of California, Berkeley

Abstract

This paper proposes an alternative to the synthetic control method (SCM) for estimating the effect of a policy intervention on an outcome over time. Recurrent neural networks (RNNs) are used to predict counterfactual time-series of treated units using only the outcomes of control units as model inputs. The proposed method does not rely on pre-period covariates to construct the synthetic control and is consequently less susceptible to p-hacking. RNNs are also capable of handling multiple treated units and can learn nonconvex combinations of control units. In placebo tests, RNNs outperform SCM in predicting the post-intervention time-series of control units, while yielding a comparable proportion of false positives. The RNN-based approach contributes to a new generation of data-driven machine learning techniques such as matrix completion and the Lasso for generating counterfactual predictions.

Keywords: Causal inference; Recurrent neural networks; Randomization inference; Synthetic control

[†]PhD Candidate, Department of Political Science, 210 Barrows Hall #1950, Berkeley, CA 94720-1950. *Email:* poulos@berkeley.edu. *Telephone:* +1−510−642−6323. I acknowledge support of the National Science Foundation Graduate Research Fellowship (DGE 1106400) and the NVIDIA Corporation for the donation of the Titan Xp GPU used for this research.

1 INTRODUCTION

An important problem in the social sciences is estimating the effect of a policy intervention on an outcome over time. When interventions take place at an aggregate level (e.g., city or state), researchers make causal inferences by comparing the post-intervention outcomes of affected ("treated") units against the outcomes of unaffected units ("controls").

The synthetic control method (SCM) (Abadie, Diamond, and Hainmueller 2010) is a popular method for making causal inferences on observational time-series. The method compares a single treated unit with a synthetic control that combines the outcomes of multiple control units on the basis of their pre-intervention similarity with the treated unit. Specifically, the synthetic control is constructed by choosing a convex combination of weights $\mathbf{w} = (w_1, \ldots, w_j), w_j \geq 0, \sum w_j = 1$, of control time-series that minimizes $\|\mathbf{X_1} - \mathbf{X_0}\mathbf{w}\|_\mathbf{v}$, where $\mathbf{X_1}$ and $\mathbf{X_0}$ are the pre-intervention covariates of the treated and control units, respectively, and $\mathbf{v} = (v_1, \ldots, v_J)$ are importance weights chosen to minimize the prediction error produced by $\mathbf{w}^*\mathbf{v}$ during cross-validation.

The SCM has several limitations. First, the convexity restriction of the synthetic control estimator precludes dynamic, nonlinear interactions between multiple control units. Ferman and Pinto (2016) point out that the convexity restriction implies that the SCM estimator may be biased even if selection into treatment is only correlated with time-invariant unobserved covariates. Second, the specification of the estimator can produce very different results. Ferman, Pinto, and Possebom (2018) show how cherry-picking between common SCM specifications can facilitate p-hacking. Third, while SCM can be generalized to handle multiple treated units (e.g., Dube and Zipperer 2015; Xu 2017), the generalized SCM is not capable of sharing model weights when predicting the outcomes of multiple treated units. Fourth, pre-intervention covariates are not available in all empirical applications.

This paper proposes an alternative to SCM that is capable of automatically selecting appropriate control units at each time-step, allows for nonconvex combinations of control units, and does not rely on pre-intervention covariates. The method uses recurrent neural networks (RNNs) to predict a counterfactual time-series of treated units using only control unit outcomes as model inputs. RNNs

are a class of neural networks that take advantage of the sequential nature of time-series data by sharing model parameters across multiple time-steps (El Hihi and Bengio 1995). Non-parametric models such as RNNs are useful for prediction problems because we do not have to assume a functional form on the data. In addition, RNNs can learn the most useful nonconvex combination of control unit outcomes at each time-step for generating counterfactual predictions. Relaxing the convexity restriction is useful when the data-generating process underlying the outcome of interest depends nonlinearly on the history of its inputs. RNNs are also capable of sharing learned parameters across time-steps and multiple treated units.

The proposed method builds on a new literature that uses machine learning techniques such as matrix completion (Athey et al. 2017) and the Lasso (Doudchenko and Imbens 2016) for data-driven alternatives to SCM. RNNs are a natural choice for generating counterfactual predictions because they are specifically structured for sequential data and have been shown to outperform various linear models on time-series prediction tasks (Cinar et al. 2017). In a series of placebo tests using data common to the SCM literature, I present evidence that the RNN-based method outperforms SCM in terms of minimizing prediction error while maintaining a comparable false positive rate.

In the section immediately below, I describe the approach of using RNNs for counterfactual time-series prediction; Section 3 details the procedure for evaluating the models in terms of predictive accuracy and statistical significance; Section 4 presents the results of the placebo tests and discusses when the proposed method is expected to outperform SCM; Section 5 concludes by discussing the contributions of the paper and offering potential avenues for future research.

2 RNNs FOR COUNTERFACTUAL PREDICTION

The proposed method estimates the causal effect of a discrete intervention in observational time-series data; i.e., settings in which treatment is not randomly assigned and there exists both pre- and post-intervention period observations of the outcome of interest. Brodersen et al. (2015) originally

propose an alternative to SCM that uses the pre-period time-series of control units to train a model to predict the counterfactual time-series of the treated unit. The key assumption of this approach is that the relationship between predictors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\tau)}$ and the treated time-series $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(\tau)}$ modeled prior to the intervention persists after the intervention. The approach also assumes that the control units are not themselves affected by the intervention; i.e., there is no spillover effect that would indeterminately bias the estimate and possibly lead to a false positive.

In its most basic form, the pre-period relationship can be modeled by linear regression,

$$\mathbf{y}^{(t)} = \alpha_0 + \beta \mathbf{x}^{(t)} + \epsilon^{(t)} \qquad \forall\, t = 1, \ldots, n, \tag{1}$$

where time-step $t = 1, \ldots, n, \ldots, \tau$ is the temporal ordering of the time-series and n denotes the end of the pre-period. As long as $\mathbf{x}^{(t)}$ was not impacted by the intervention, it is plausible that the modeled relationship persists after the intervention. The fitted model is then used to predict the counterfactual time-series of the treated group in the post-period:

$$\hat{\mathbf{y}}^{(t)} = \alpha_1 + \beta \mathbf{x}^{(t)} + \zeta^{(t)} \qquad \forall\, t = n+1 \ldots., \tau, \tag{2}$$

The key assumption of this model is that the relationship between controls $\mathbf{x}^{(t)}$ and the treated time-series $\mathbf{y}^{(t)}$ modeled prior to the intervention persists after the intervention. Under this assumption, the inferred causal effect of the intervention on the treated group is the difference between the observed time-series of the treated units and the counterfactual time-series that would have been observed in the absence of the intervention:

$$\hat{\phi}^{(t)} = \mathbf{y}^{(t)} - \hat{\mathbf{y}}^{(t)} \qquad \forall\, t = n+1, \ldots, \tau. \tag{3}$$

This treatment effect is calculated at every post-period time-step and is thus useful for understanding the temporal evolution of the causal effect.

Unlike linear models, RNNs are non-parametric and are structured for time-series data. RNNs consist of a hidden state $\mathbf{h}^{(t)}$ and an output $\mathbf{y}^{(t)}$ which operate on a sequence $\mathbf{x}^{(t)}$. At each time-step

t, RNNs input $\mathbf{x}^{(t)}$ and pass it to the hidden state, which is updated with a function $g^{(t)}$ using the entire history of the sequence (pp. 337 Goodfellow, Bengio, and Courville 2016):

$$\begin{aligned}\mathbf{h}^{(t)} &= g^{(t)}\left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)}\right) \\ &= f\left(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}\right), \end{aligned} \tag{4}$$

where $f(\cdot)$ is a nonlinear function that operates on all time-steps and input lengths. The updated hidden layer is used to generate a sequence of output values $\mathbf{o}^{(t)}$ in the form of log probabilities that correspond to $\mathbf{x}^{(t)}$. The loss function computes $\hat{\mathbf{y}}^{(t)} = \mathrm{linear}(\mathbf{o}^{(t)})$ and compares this value to $\mathbf{y}^{(t)}$.

A special variant of RNNs that are suitable for handling variable-length sequential data are encoder-decoder networks (Cho et al. 2014). Encoder-decoder networks are the standard for neural machine translation (Bahdanau, Cho, and Bengio 2014; Vinyals et al. 2014) and are also widely used for predictive tasks, including speech recognition (Chorowski et al. 2015) and time-series forecasting (Zhu and Laptev 2017).

Encoder-decoder networks are trained to estimate the conditional distribution of the output sequence given the past input sequence, e.g., $p(\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(t)} | \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)})$, where the input and output sequence lengths can differ. The encoder RNN reads in $\mathbf{x}^{(t)}$ sequentially and the hidden state of the network updates according to Eq. 4. The hidden state of the encoder is a context vector $\mathbf{c}$ that summarizes the input sequence, which is copied over to the decoder RNN. The decoder generates a variable-length output sequence by predicting $\mathbf{y}^{(t)}$ given the encoder hidden state and the previous element of the output sequence. Thus, the hidden state of the decoder is updated recursively by

$$\mathbf{h}^{(t)} = f\left(\mathbf{h}^{(t-1)}, \mathbf{y}^{(t-1)}, \mathbf{c}\right), \tag{5}$$

and the conditional probability of the next element of the sequence is

$$P(\mathbf{y}^{(t)}|\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(t-1)},\mathbf{c}) = f\left(\mathbf{h}^{(t)},\mathbf{y}^{(t-1)},\mathbf{c}\right). \tag{6}$$

Effectively, the decoder learns to generate outputs $\mathbf{y}^{(t)}$ given the previous outputs, conditioned on the input sequence.

3 PREDICTIVE ACCURACY AND STATISTICAL SIGNIFICANCE

In applications in which the counterfactual time-series is known (i.e., placebo tests) models can be evaluated in terms of the MSPE between the predicted and actual post-intervention time-series among control units. Specifically, I calculate:

$$\text{MSPE} = \frac{1}{\tau - n}\sum_{n+1:\tau}^{\tau}\left(\hat{\phi}^{(t)}\right)^2, \tag{7}$$

where $\hat{\phi}^{(t)}$ is defined in Eq. 3. Eq. 7 measures the accuracy of the counterfactual predictions, and consequently the accuracy of the estimated treatment effect. However, this metric does not tell us anything about the statistical significance of estimated treatment effects.

Abadie, Diamond, and Hainmueller (2010) propose a randomization inference approach for calculating the exact distribution of placebo effects under the null hypothesis. Following Cavallo et al. (2013), the procedure described below extends this method to the case of multiple (placebo) treated units by constructing a distribution of *average* placebo effects under the null hypothesis:

1. Estimate the observed test static μ^* by estimating Eq. 7 for all J, which results in a matrix of dimension $(\tau - n) \times J$. Taking the row-wise mean results in a $\tau - n$-length array of observed average placebo treated effects.

2. Calculate every possible average placebo effect μ by randomly sampling without replacement which $J-1$ control units are assumed to be treated. There are $\mathcal{Q} = \sum_{g=1}^{J-1}\binom{J}{g}$ possible average placebo effects. The result is a matrix of dimension $(\tau - n) \times \mathcal{Q}$. Note that $\mathcal{Q}$

can be computationally burdensome when there are many control units. In the applications described below, I set $\mathcal{Q} = 10,000$ in which $J > 16$.

3. Take a column-wise sum of the number of μ that are greater than or equal to μ^*.

Each element of the $(\tau - n) \times J$ matrix of counts obtained from the last step is divided by $\mathcal{Q}$ to estimate an array of exact two-sided p values, $\hat{p}$. I then calculate a single false positive rate by FPR $= \frac{\text{FP}}{(\tau-n)\times J}$, where FP is the number of false positives defined as the number of p-values less than or equal to $\alpha = 0.5$ and J is the number of placebo treated units.

Assuming that treatment has a constant additive effect Δ, I construct an interval estimate for Δ by inverting the randomization test. Let δ_Δ be the test statistic calculated by subtracting all possible μ by Δ. I derive a two-sided randomization confidence interval by collecting all values of δ_Δ that yield $\hat{p}$ values greater than or equal to a significance level α. I find the endpoints of the confidence interval by randomly sampling 1,000 values of Δ.

4 APPLICATION: SCM PLACEBO TESTS

I evaluate the proposed RNN-based approach on three datasets common to the SCM literature. In each dataset, I remove the actual treated unit and evaluate the models on their ability to produce low error rates on control units; i.e., estimating treatment effects of zero. A secondary evaluation criteria is the probability of falsely rejecting the null hypothesis of the randomization test as measured the FPR. The synthetic control estimator is implemented using the publicly available R code associated with each of the three referenced studies, and importance weights are chosen by cross-validation.

4.1 RNNs implementation details

In the following application, I train a baseline RNN in the form of a single unidirectional Long Short-Term Memory (LSTM) network (Schmidhuber and Hochreiter 1997) with output space dimensionality equivalent to the number of treated units. The encoder takes the form of a two-layer

bidirectional LSTMs, each with 128 hidden units, and the decoder is a single-layer Gated Recurrent Unit (GRU) (Chung et al. 2014) also with 128 hidden units. RNN weights are learned with stochastic gradient descent on mean squared prediction error (MSPE), $L_{\text{MSPE}}^{(t)} = \mathrm{E}\left[\left(\mathbf{y}^{(t)} - \hat{\mathbf{y}}^{(t)}\right)^2\right]$ using `Adam` stochastic optimization (Kingma and Ba 2014). As a regularization strategy, I apply dropout to the inputs and L2 regularization losses to the network weights.

RNNs are implemented with the `Keras` neural network library (Chollet et al. 2015) in Python on top of a TensorFlow backend. RNNs are trained in batches of size four or eight for 5,000 to 10,000 epochs, which takes about 20 minutes to run on a 12GB NVIDIA Titan Xp GPU.

4.2 Basque Country

Abadie and Gardeazabal (2003) estimate the economic impact of terrorism in the Basque Country during the period of 1968 to 1997 by comparing per-capita gross domestic product (GDP) in the Basque Country against a synthetic control region without terrorism. The synthetic control is constructed using pre-period measures of illiteracy, educational attainment, investment, and means of GDP. The time series begins in 1955, which leaves only $n = 14$ pre-period time-steps.

Figs. 1 and 2 plots estimated treatment effects on control units for each model. We observe considerable variability in the SCM estimates and not as much variability in the RNNs. This is explained by the fact that SCM cannot handle multiple (placebo) treated units and thus a separate model has to be run for each (placebo) treated unit. RNNs can handle multiple treated units and thus benefit from parameter sharing across treated units. The standard deviation from the mean MPSE, which is reported in the first column of Panel A of Table 1, indicates that SCM is comparatively more variable in terms of its predictive accuracy. The baseline LSTM yields the lowest mean MSPE, 0.007 ± 0.002.

Fig. SM-4 plots the per-period randomization p-values corresponding to treatment effects on treated and control units. p-values corresponding to treatment effects on the actual treated unit (i.e., Basque Country) are made by comparing the per-period treatment effects on Basque Country against the null distribution of average placebo effects. SCM has the comparatively lowest FPR

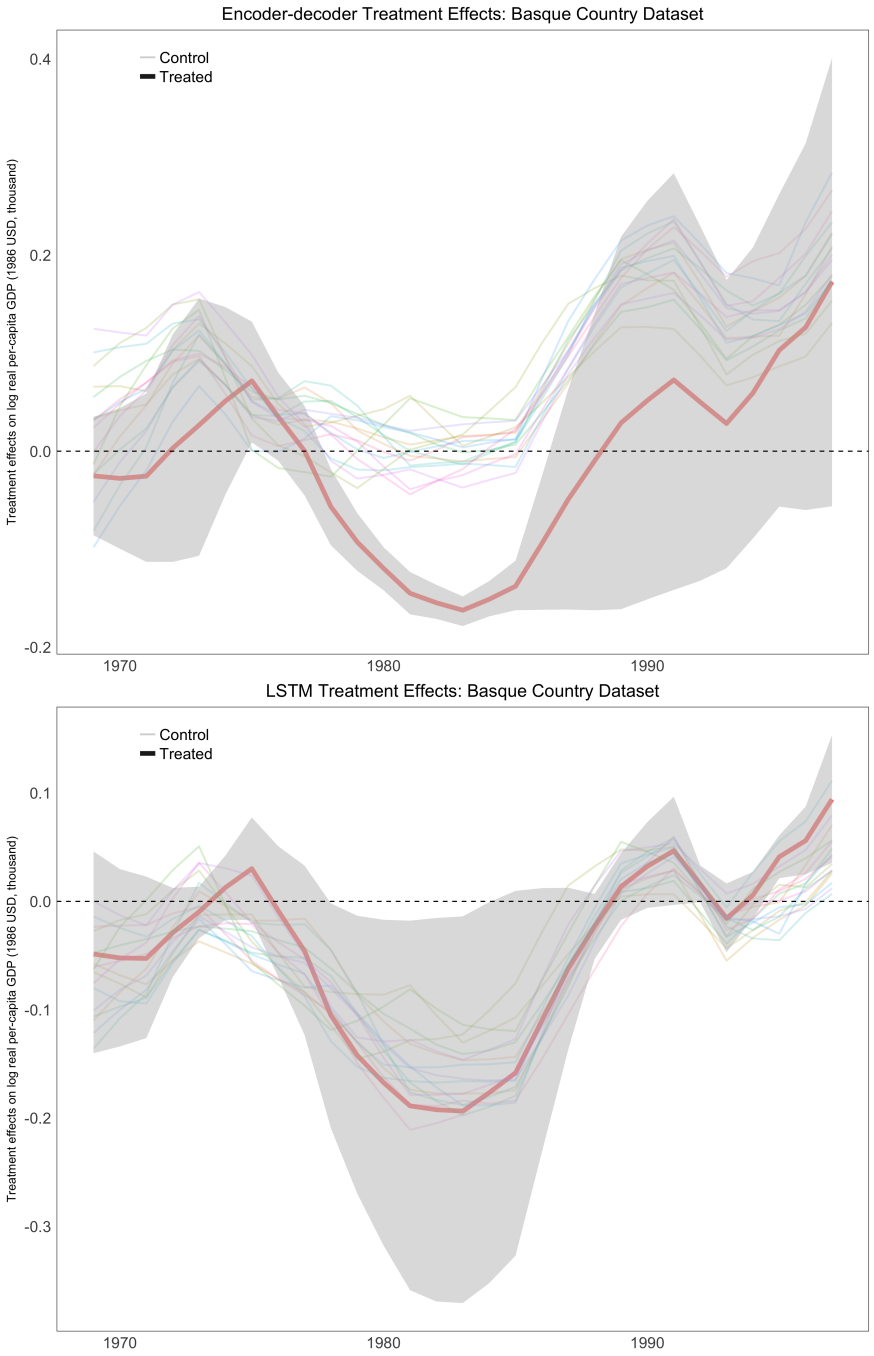

Figure 1: Encoder-decoder (top) and baseline LSTM (bottom) estimates of post-period treatment effects in Basque Country dataset. Darker line represents the effect on the actual treated unit and each lighter line represents the effects on control units. Shaded regions represent 95% randomization confidence intervals.

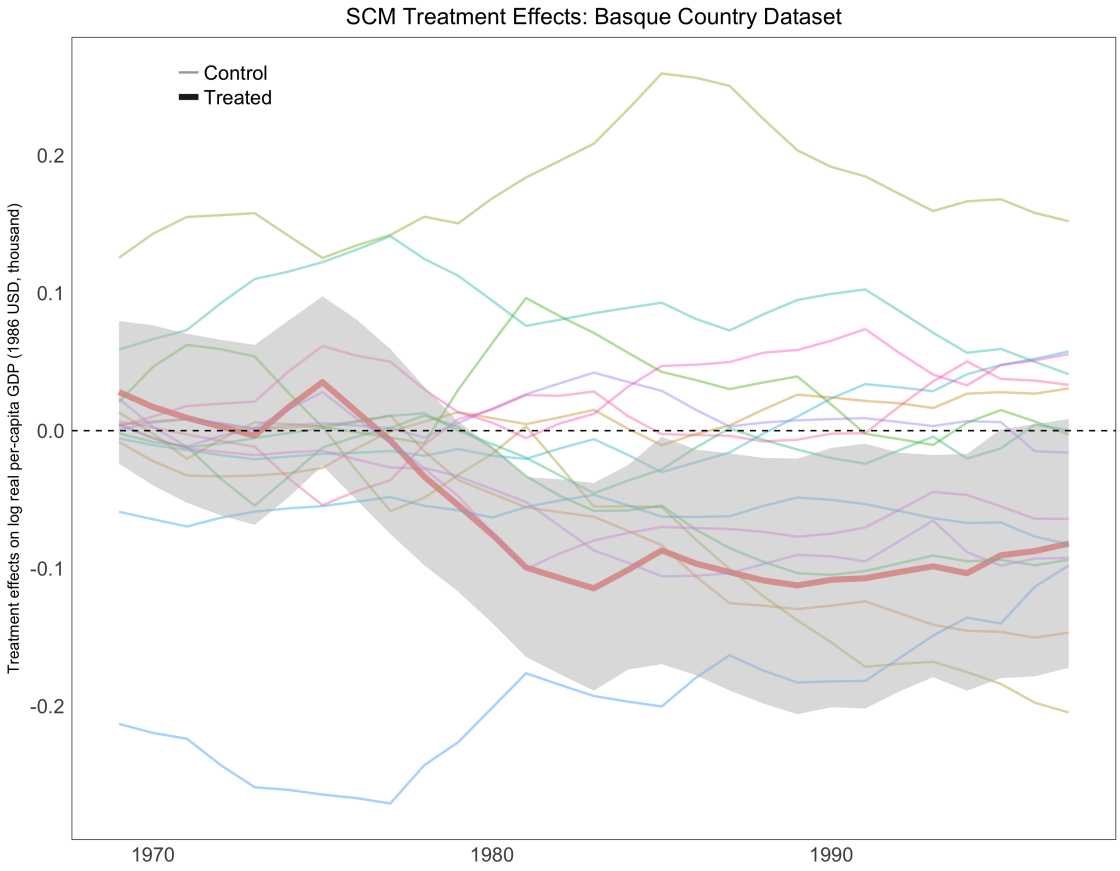

Figure 2: Synthetic control method estimates of post-period treatment effects in Basque Country dataset. See notes to Fig. 1.

Table 1: Evaluation metrics on SCM placebo tests.

Panel A: MSPE	Basque Country	California	West Germany
Encoder-decoder	0.01 ± 0.002	0.005 ± 0.004	0.03 ± 0.01
LSTM (baseline)	0.007 ± 0.002	0.008 ± 0.004	0.02 ± 0.01
SCM	0.007 ± 0.01	0.11 ± 0.13	0.06 ± 0.11
Panel B: FPR			
Encoder-decoder	0.28	0.38	0.25
LSTM (baseline)	0.32	0.33	0.25
SCM	0.27	0.33	0.25

NOTE: Error bars represent $\pm$ one standard deviation from the MSPE.

(Panel B of Table 1) among all models: the model falsely rejects the null hypothesis about a quarter of the time. Note that the reported p-values are not adjusted for multiple comparisons.

4.3 California

Abadie, Diamond, and Hainmueller (2010) applies SCM to estimate the effects of a large-scale tobacco control program implemented in California in 1988. The study spans the period of 1970 to 2000, providing $n = 19$ pre-period time-steps. The synthetic control is constructed using pre-period covariates including income, beer sales, demographics, and means of the dependent variable, which is log per-capita cigarette consumption.

Fig. SM-6 shows that for RNNs, control unit treatment effects are tightly centered around zero — as expected — whereas SCM control treatment effects (Fig. SM-7) are more dispersed. Encoder-decoder networks yield the lowest mean MSPE, 0.005 ± 0.004, while yielding comparatively higher FPR.

4.4 West Germany

Lastly, Abadie, Diamond, and Hainmueller (2015) constructs a synthetic West Germany in order to estimate the impact of the 1990 German reunification on log real per-capita GDP during a post-period that extends to 2003. The time-series begins in 1960, which leaves $n = 30$ pre-period time-steps. The synthetic control is constructed using pre-period means of GDP, trade, industry, schooling, and investment.

SCM and RNN-based approaches both assume the absence of spillover effects. Abadie, Diamond, and Hainmueller (2015) acknowledge that spillover effects is a valid concern in their study because German reunification likely have effects on GDP in the 16 OECD member countries that serve as controls. Indeed, Fig. SM-10 shows that RNNs estimate mostly positive (and increasing) treatment effects on control units, which suggests that reunification might have had a less negative impact on German GDP than the authors' estimates suggest.

Overall, the baseline LSTM yields the lowest error in terms of mean MSPE, 0.02 ± 0.01, with a FRP comparable to SCM.

4.5 Discussion

I compare the predictive accuracy of the RNN-based approach against SCM by running a series of placebo tests using data from three datasets common to the SCM literature. The models are evaluated primarily on their ability to produce low error rates on control units (i.e., estimating treatment effects of zero); a secondary evaluation criteria is minimizing the probability of falsely rejecting the null hypothesis of the randomization test.

I find that either encoder-decoder networks or LSTM outperform SCM on each of the three datasets in terms of having the lowest MSPE, with FPRs comparable to SCM. The baseline LSTM outperforms encoder-decoder networks in two of the three datasets. When applied to datasets with low-dimensional predictor sets, the LSTM performs well but deeper networks such as encoder-decoder networks are susceptible to overfitting. Overfitting in this case means that the networks learn dependencies on a small subset of predictors and cannot generalize well to unseen data. Overfitting occurs when training encoder-decoder networks on the Basque Country dataset (Fig. SM-1a), which is has the lowest dimensions of the three SCM datasets. Even in this case of obvious overfitting, model check-pointing is employed so that the model with the lowest validation error is used to produce counterfactual time-series.

The results suggest that RNNs should outperform SCM in all cases as long as the complexity of the network architecture is proportional to the dimension of the predictor set. Encoder-decoder networks outperform the other models when the predictor set is comparatively large (i.e., $J = 38$ in the California dataset), while the baseline LSTM outperforms all other models on smaller predictor sets ($J = 16$ for Basque Country and West Germany datasets).

5 CONCLUSION

This paper proposes a novel alternative to SCM, which is growing in popularity in the social sciences despite its limitations; the most obvious being that the choice of specification can lead to different results, and thus facilitate p-hacking. By inputting only control unit outcomes and not relying on pre-period covariates, the proposed method offers a more principled approach than SCM. RNNs are also capable of handling multiple treated units and can learn nonconvex combinations of control units. The former attribute is useful because the model can share parameters across treated units, and thus generate more precise predictions in settings in which treated units share similar data-generating processes. The latter attribute is beneficial when the data-generating process underlying the outcome of interest depends nonlinearly on the history of its inputs.

The RNN-based approach joins a new generation of data-driven machine learning techniques such as matrix completion (Athey et al. 2017) and the Lasso (Doudchenko and Imbens 2016) for generating counterfactual predictions. While the strength of SCM lies in its simplicity in setup and implementation, several problems arise from the lack of guidance on how to specify the SCM estimator. Kaul et al. (2015) show, for instance, that the common practice of including lagged versions of the outcome variable as separate predictors can render all other covariates irrelevant.

Machine learning techniques in general have an advantage over SCM in that they automatically choose appropriate predictors without relying on pretreatment covariates; this capability limits "researcher degrees of freedom" that arises from choices on how to specify the model. RNNs have an advantage over alternative machine learning algorithms because they are specifically structured to exploit the sequential nature of time-series data by sharing model parameters across time-steps.

Future research might investigate through simulations how the interaction between RNN complexity (as determined by the number of hidden layers or nodes) and data dimensionality impacts predictive accuracy. Simulations will also allow us to assess the exact impact of data dimensionality, the proportion of treated units, convexity versus non-convexity in the modeled relationship, and the length of the pre-period on the choice between RNNs and SCM.

REFERENCES

Abadie, Alberto, Alexis Diamond, and Jens Hainmueller. (2010). "Synthetic Control Methods for Comparative Case Studies: Estimating the Effect of California's Tobacco Control Program." *Journal of the American Statistical Association* 105, no. 490 (2010): 493–505.

———. (2015). "Comparative Politics and the Synthetic Control Method." *American Journal of Political Science* 59, no. 2 (2015): 495–510.

Abadie, Alberto, and Javier Gardeazabal. (2003). "The Economic Costs of Conflict: A Case Study of the Basque Country." *The American Economic Review* 93, no. 1 (2003): 113–132.

Athey, S., M. Bayati, N. Doudchenko, G. Imbens, and K. Khosravi. (2017). "Matrix Completion Methods for Causal Panel Data Models." *ArXiv e-prints* (October 2017). arXiv: 1710.10251 [math.ST].

Bahdanau, D., K. Cho, and Y. Bengio. (2014). "Neural Machine Translation by Jointly Learning to Align and Translate." *ArXiv e-prints* (September 2014). arXiv: 1409.0473 [cs.CL].

Brodersen, Kay H, Fabian Gallusser, Jim Koehler, Nicolas Remy, and Steven L Scott. (2015). "Inferring Causal Impact Using Bayesian Structural Time-series Models." *The Annals of Applied Statistics* 9, no. 1 (2015): 247–274.

Cavallo, Eduardo, Sebastian Galiani, Ilan Noy, and Juan Pantano. (2013). "Catastrophic Natural Disasters and Economic Growth." *Review of Economics and Statistics* 95, no. 5 (2013): 1549–1561.

Cho, K., B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." *ArXiv e-prints* (June 2014). arXiv: 1406.1078 [cs.CL].

Chollet, François, et al. (2015). *Keras*. https://keras.io, 2015.

Chorowski, Jan K, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. (2015). "Attention-Based Models for Speech Recognition." In *Advances in Neural Information Processing Systems,* 577–585. 2015.

Chung, J., C. Gulcehre, K. Cho, and Y. Bengio. (2014). "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." *ArXiv e-prints* (December 2014). arXiv: 1412.3555.

Cinar, Yagmur Gizem, Hamid Mirisaee, Parantapa Goswami, Eric Gaussier, Ali Aït-Bachir, and Vadim Strijov. (2017). "Position-Based Content Attention for Time Series Forecasting with Sequence-to-Sequence RNNs." In *International Conference on Neural Information Processing,* 533–544. Springer, 2017.

Doudchenko, N., and G. W. Imbens. (2016). "Balancing, Regression, Difference-In-Differences and Synthetic Control Methods: A Synthesis." *ArXiv e-prints* (October 2016). arXiv: 1610.07748 [stat.AP].

Dube, Arindrajit, and Ben Zipperer. (2015). "Pooling Multiple Case Studies Using Synthetic Controls: An Application to Minimum Wage Policies." IZA Discussion Paper No. 8944, 2015.

El Hihi, Salah, and Yoshua Bengio. (1995). "Hierarchical Recurrent Neural Networks for Long-Term Dependencies." In *Neural Information Processing Systems,* 400:409. 1995.

Ferman, Bruno, and Cristine Pinto. (2016). *Revisiting the Synthetic Control Estimator.* Available at https://mpra.ub.uni-muenchen.de/81941/, 2016.

Ferman, Bruno, Cristine Pinto, and Vitor Possebom. (2018). *Cherry picking with synthetic controls.* Available at: https://mpra.ub.uni-muenchen.de/85138/1/MPRA_paper_85138.pdf, 2018.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. (2016). *Deep Learning.* Cambridge, MA: MIT press, 2016.

Kaul, Ashok, Stefan Klößner, Gregor Pfeifer, and Manuel Schieler. (2015). *Synthetic Control Methods: Never Use All Pre-Intervention Outcomes Together With Covariates.* Available at: `http://www.oekonometrie.uni-saarland.de/papers/SCM_Predictors.pdf`, 2015.

Kingma, D. P., and J. Ba. (2014). "Adam: A Method for Stochastic Optimization." *ArXiv e-prints* (December 2014). arXiv: `1412.6980`.

Schmidhuber, Jürgen, and Sepp Hochreiter. (1997). "Long Short-Term Memory." *Neural Computation* 9, no. 8 (1997): 1735–1780.

Vinyals, O., L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton. (2014). "Grammar as a Foreign Language." *ArXiv e-prints* (December 2014). arXiv: `1412.7449 [cs.CL]`.

Xu, Yiqing. (2017). "Generalized Synthetic Control Method: Causal Inference with Interactive Fixed Effects Models." *Political Analysis* 25, no. 1 (2017): 57–76.

Zhu, L., and N. Laptev. (2017). "Deep and Confident Prediction for Time Series at Uber." *ArXiv e-prints* (September 2017). arXiv: `1709.01907 [stat.ML]`.