International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-12, Special Issue-1, March-2024
https://doi.org/10.55524/CSISTW.2024.12.1.72
Computing for Sustainable Innovation: Shaping Tomorrow's World (CSISTW-2024) on 16th-17th February, 2024
Organized by B.N. College of Engineering and Technology, Lucknow, India

A Survey of Deep Learning Techniques for Time-Series Forecasting

Vinay Kumar[1], and Atebar Haider[2]

[1,2]Assistant Professor, Department of Computer Science & Engineering, B.N. College of Engineering & Technology, Lucknow, Uttar Pradesh, India

Correspondence should be addressed to Vinay Kumar;Vinay.bncet@gmail.com

ABSTRACT- Various deep learning architectures have been created to cater to the diverse nature of time-series datasets across various domains. This article examines prevalent designs for both encoder and decoder components used in time-series forecasting, covering both one-step-ahead and multi-horizon approaches. It delves into the incorporation of temporal information into predictions by each model. Additionally, recent advancements in hybrid deep learning models are highlighted, showcasing the integration of well-established statistical models with neural network elements to enhance methods in both categories. Lastly, the article outlines how deep learning can contribute to decision support in the context of time-series data. This piece is part of the theme issue on 'Machine learning for weather and climate modeling.

KEYWORDS- Time-Series Modeling, Deep Learning, Neural Networks, Forecasting Models, Machine Learning.

I. INTRODUCTION

Time-series modeling has been a prominent subject of academic exploration, playing a crucial role in diverse fields like climate modeling [1], biological sciences [2], medicine [3], and practical domains like retail [4] and finance [5]. Traditional approaches have predominantly relied on parametric models guided by domain expertise, such as autoregressive (AR) [6], exponential smoothing [7,8], or structural time-series models [9]. However, contemporary machine learning techniques offer a data-centric approach to capturing temporal dynamics [10]. With the surge in data accessibility and computing capabilities, machine learning has emerged as an indispensable component of the next wave of time-series forecasting models.

In recent times, deep learning has become widely recognized, driven by impressive accomplishments in tasks such as image classification [11], natural language processing [12], and reinforcement learning [13]. Deep neural networks leverage specialized architectural assumptions, known as inductive biases [14], that capture the intricacies of the underlying datasets. This enables these networks to acquire intricate data representations [15], reducing the reliance on manual feature engineering and model design. The accessibility of open-source Backpropagation frameworks [16,17] has further streamlined network training, facilitating customization of network components and loss functions.

Given the variety of time-series challenges found in different fields, numerous design options for neural networks have surfaced. This article provides an overview of prevalent strategies for time-series prediction using deep neural networks. Initially, we present cutting-edge techniques applicable to common forecasting issues, such as multi-horizon forecasting and uncertainty estimation. Subsequently, we examine the rise of a novel trend involving hybrid models, which integrate domain-specific quantitative models with deep learning elements to enhance forecasting accuracy. Following that, we delineate two fundamental approaches in which neural networks can contribute to decision support, focusing on interpretability and counterfactual prediction methods. Lastly, we summarize with insights into promising future research avenues in deep learning for time series prediction, particularly in the realms of continuous-time and hierarchical models.

II. LITERATURE REVIEW

A. *Deep Learning Architectures For Time-Series Forecasting*

Time-series forecasting models anticipate future values of a target variable, denoted as $\mathbf{y}_{i,t}$ for a specified entity i at time t. Each entity signifies a coherent collection of temporal data, such as readings from various weather stations in climatology or vital signs from diverse patients in medicine. These entities can be observed simultaneously. In its most straightforward manifestation, one-step-ahead forecasting models adopt the following structure:

$$\hat{y}_{i,t+1} = f(y_{i,t-k:t}, x_{i,t-k:t}, s_i),$$

In this context, $\hat{y}_{i,t+1}$ represents the model's forecast, and $y_{i,t-k:t} = \{y_{i,t-k}, \dots, y_{i,t}\}$, $\mathbf{x}_{i,t-k:t} = \{\mathbf{x}_{i,t-k}, \dots, \mathbf{x}_{i,t}\}$ denote the observations of the target and exogenous inputs, respectively, within a look-back window of size k. The variable $\mathbf{s}_i$ represents static metadata associated with the entity, such as sensor location, and f(.) represents the prediction function learned by the model. Although our primary focus in this survey is on univariate forecasting (i.e., 1-D targets), it's important to note that the same components can be extended to multivariate models without loss of generality [26–30]. For the sake of simplicity in notation, we will omit the entity index i in subsequent sections unless explicitly necessary.

B. Basic Building Blocks

Predictive relationships are acquired by deep neural networks through the utilization of nonlinear layers to create intermediate feature representations [15]. In the context of time-series scenarios, this process can be interpreted as encapsulating pertinent historical information within a latent variable, denoted as ***zt***. The ultimate forecast is then generated solely based on the value of ***zt***.

$$f(y_{t-k:t}, x_{t-k:t}, s) = g_{dec}(zt),$$

And

$$z_t = g_{enc}(y_{t-k:t}, x_{t-k:t}, s)$$

The encoder and decoder functions, denoted as $g_{enc}(.)$ and $g_{dec}(.)$, respectively, serve as fundamental components in deep learning architectures. It's important to note that in equation (2.1), the subscript i has been omitted for simplicity in notation, such as replacing yi,t with yt. These encoders and decoders are essential building blocks, and the selection of the network determines the types of relationships that the model can learn. This section delves into contemporary design choices for encoders, as depicted in Figure 1, and their connections to traditional temporal models. Additionally, we investigate common network outputs and loss functions employed in time-series forecasting applications.

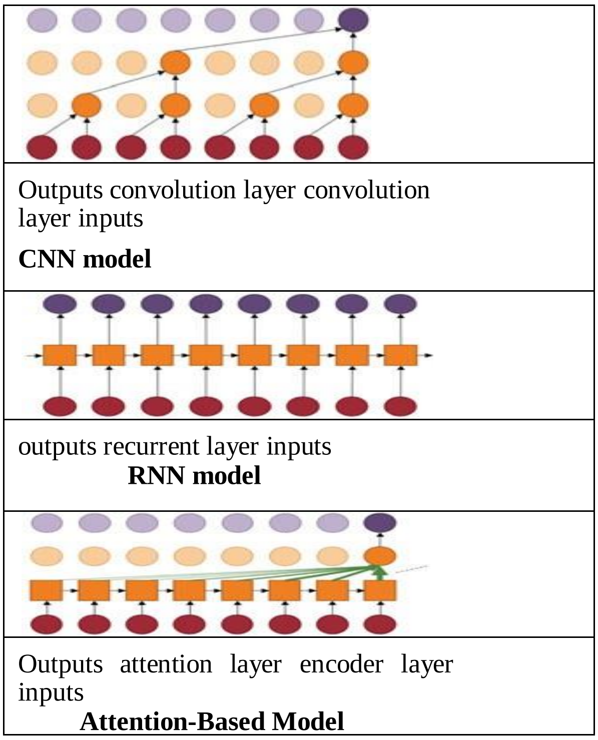

Figure 1: Incorporating temporal information using different encoder architectures. (a) CNN model, (b) RNN model and (c) attention-based model. (Online version in colour.)

C. Convolution Neural Networks

Originally designed for image datasets, convolution neural networks (CNNs) have been conventionally employed to capture local relationships that remain consistent across spatial dimensions [11,31]. To make CNNs suitable for time-series datasets, researchers incorporate multiple layers of causal convolutions [32–34]. These causal convolutions involve filters specifically designed to utilize only past information for forecasting. In the case of an intermediate feature at hidden layer l, each causal convolution filter follows the outlined form.

$$h_t^{l+1} = A((W * h)(l * t))$$

And,

$$(W * h)(l * t) = \sum_{\tau=0}^{k} W(l, \tau) h_{t-\tau}^{l},$$

where $\boldsymbol{h}^l \in \mathbb{R}^{Hin}$ is an intermediate state at layer l at time t, $*$ is the convolution operator, $W(l, \tau) \in \mathbb{R}^{Hout \times Hin}$ is a fixed filter weight at layer l, and $A(.)$ is an activation function, such as a sigmoid function, representing any architecture-specific nonlinear processing. For CNNs that use a total of L convolution layers, we note that the encoder output is then $\boldsymbol{z}_t = \boldsymbol{h}^L$.

Examining the 1-D scenario, it becomes evident that equation (2.5) shares a striking resemblance with finite impulse response (FIR) filters in digital signal processing [35]. This observation gives rise to two significant implications concerning the temporal relationships learned by CNNs. Firstly, akin to the spatial invariance assumptions inherent in standard CNNs, temporal CNNs operate under the assumption that relationships are time-invariant. This entails utilizing the same set of filter weights across all time steps. Moreover, CNNs are constrained to utilize inputs within their designated look back window or receptive field for making predictions. Consequently, careful tuning of the receptive field size (k) is essential to ensure that the model effectively incorporates all pertinent historical information. It is noteworthy that a single causal CNN layer is essentially equivalent to an autoregressive (AR)

D. Recurrent Neural Networks

Historically, recurrent neural networks (RNNs) have found application in sequence modeling [31], demonstrating notable performance across various natural language processing tasks [36]. Due to the inherent representation of time-series data as sequences of inputs and targets, numerous RNN-based architectures have been created for temporal forecasting purposes [37–40]. Fundamentally, RNN cells incorporate an internal memory state that serves as a concise recapitulation of preceding information. This memory state undergoes recursive updates with new observations at each time step, as depicted in figure ***1b***.

$$z_t = v(z_{t-1}, y_t, x_t, s)$$

where zt ∈ RH here is the hidden internal state of the RNN, and ν(.) is the learnt memory update function. For instance, the Elman RNN [41], one of the simplest RNN variants, would take the form below

$$t_{t+1} = \gamma_y \left(W_y z_t + b_y \right)$$

and

$$z_t = \gamma_z \left(W_{z1} z_{t-1} + W_{z2} y_t + W_{z3} x_t + W_{z4} s + b_z \right)$$

where W., b. are the linear weights and biases of the network respectively, and γy(.), γz(.) are network activation functions. Note that RNNs do not require the explicit specification of a lookback window as per the CNN case. From a signal processing perspective, the main recurrent layer—i.e. equation (2.9)—thus resembles a nonlinear version of infinite impulse response (IIR) filters.

The challenge of learning long-range dependencies in data arises in older versions of Recurrent Neural Networks (RNNs) due to limitations in their ability to handle infinite lookback windows. This issue is attributed to problems with exploding and vanishing gradients, as discussed in previous studies [42,43] and related to resonance in the memory state [31]. To overcome these limitations, Long Short-Term Memory networks (LSTMs) [44] were developed. LSTMs enhance gradient flow within the network by introducing a cell state (ct) that stores long-term information. This information is modulated through a series of gates, addressing the challenges posed by older RNN variants.

input gate: $i_t = \sigma(W_{i1} z_{t-1} + W_{i2} y_t + W_{i3} x_t + W_{i4}$

output gate $o_t = \sigma(W_{O1} z_{t-1} + W_{O2} y_t + W_{O3} x_t + W_{O4}$

forget gate $f_t = \sigma(W_{f1} z_{t-1} + W_{f2} y_t + W_{f3} x_t + W_{f4} s$

where zt−1 is the hidden state of the LSTM, and σ (.) is the sigmoid activation function. The gates modify the hidden and cell states of the LSTM as below

hidden state $z_t = O_t \odot tanh(c_t)$

cell state $c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{c1} z_{t-1} + W_{c2} y_t + W_{c3} x_t + W_{c4} s + b_c)$

where ⨀ is the element-wise (Hadamard) product, and tanh(.) is the tanh activation function.

In [39], the connection between Bayesian filters [45] and Recurrent Neural Networks (RNNs) is explored. Both Bayesian filters and RNNs share a common characteristic in maintaining a hidden state that undergoes recursive updates over time. In the case of Bayesian filters, exemplified by the Kalman filter [46], the inference process involves updating the sufficient statistics of the latent state through a sequence of state transition and error correction steps. The deterministic equations employed in Bayesian filtering steps to modify sufficient statistics can be likened to the RNN's simultaneous approximation of both steps. In this analogy, the memory vector in the RNN encompasses all pertinent information necessary for making predictions.

E. Attention Mechanisms

The advancement of attention mechanisms [47,48] has resulted in enhanced capabilities for learning long-term dependencies. Transformer architectures, in particular, have demonstrated state-of-the-art performance across various natural language processing applications [12,49,50]. Temporal features are aggregated by attention layers using dynamically generated weights (illustrated in figure 1c). This enables the network to concentrate on crucial time steps from the past, even when they are significantly distant within the lookback window. In essence, attention serves as a mechanism for key-value lookup based on a provided query [51], as outlined below.

$$h_t = \sum_{\tau=0}^{k} \alpha(k_t * q_\tau) v_{\tau-r},$$

where the key κ t, query qτ and value vt−τ are intermediate features produced at different time steps by lower levels of the network. Furthermore, α(κt, qτ) ∈ [0, 1] is the attention weight for t − τ generated at time t, and ht is the context vector output of the attention layer.

Note that multiple attention layers can also be used together as per the CNN case, with the output from the final layer forming the encoded latent variable zt.

Recent research has highlighted the advantages of incorporating attention mechanisms into time-series forecasting applications, showcasing enhanced performance compared to similar recurrent networks [52–54]. In one example, attention is employed to consolidate features extracted by RNN encoders, generating attention weights as described below in [52].

$$\alpha(t) = softmax(\eta_t)$$

And

$$\eta_t = W_{\eta_1} \, tanh(W_{\eta_2} \, k_{t-1} + W_{\eta_3} \, q_\tau + b_\eta)$$

where α(t) = [α(t, 0), ... α(t, k)] is a vector of attention weights, κ t−1, qt are outputs from LSTM encoders used for feature extraction, and softmax(.) is the softmax activation function.

In recent times, transformer architectures have been explored in [53,54], utilizing scalar-dot product self-attention [49] on features extracted within a specified look back window. From the viewpoint of time-series modeling, attention offers two crucial advantages. First and foremost, networks with attention can focus directly on noteworthy events. In applications such as retail forecasting, this capability allows the model to consider events like holidays or promotions, which may impact sales positively. Secondly, as demonstrated in [54], attention-based networks can acquire the ability to learn temporal dynamics specific to different regimes, achieving this by employing distinct attention weight patterns for each regime.

F. Multi-Horizon Forecasting Models

In numerous scenarios, having the ability to obtain predictive estimates at various future time points proves advantageous. This enables decision-makers to observe trends over an extended period, allowing for the optimization of actions across the entire forecasted trajectory. From a statistical standpoint, the concept of multi-horizon forecasting can be perceived as a slight adjustment to the conventional one-step-ahead prediction problem, as illustrated in equation (2.1) below.

$$\hat{y}_{t+\tau} = f(y_{t-k:t}, x_{tik:t}, u_{t-k:t+\tau}, s, \tau)$$

where $\tau \in \{1, \ldots, \tau max\}$ is a discrete forecast horizon, ut are known future inputs (e.g. date information, such as the day-of-week or month) across the entire horizon, and xt are inputs that can only be observed historically.
In line with traditional econometric approaches [57,58], deep learning architectures for multi-horizon forecasting can be divided into iterative and direct methods—as shown in figure 2 and described in detail below.

G. Iterative Methods

Multi-horizon forecasting often involves iterative methods that leverage AR deep learning architectures [37,39,40,53]. These approaches generate forecasts for multiple time horizons by iteratively inputting target samples into future time steps, as illustrated in figure 2a. The process is repeated to create various trajectories, and forecasts are derived by utilizing sampling distributions for target values at each step. Predictive means, such as the Monte Carlo estimate, can be extracted through this methodology.

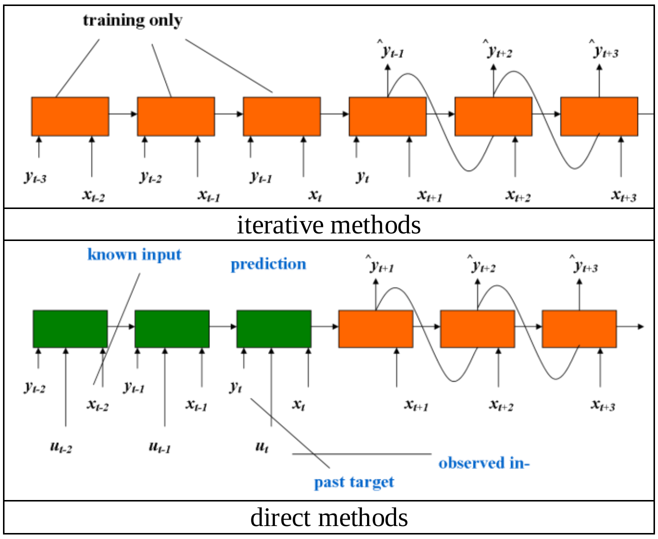

Figure 2: Main types of multi-horizon forecasting models. (a) Iterative methods (b) direct methods. (Online version in color.)

AR models are trained using the same method as one-step-ahead prediction models, employing backpropagation through time. The iterative approach facilitates the seamless extension of standard models to multi-step forecasting. However, since a minor amount of error is generated at each time step, the recursive nature of iterative methods may result in significant error accumulation over extended forecasting periods. Furthermore, these methods operate under the assumption that all inputs except the target are known at runtime, necessitating only target samples to be inputted into future time steps. This assumption can pose limitations in practical scenarios where observed inputs are present, highlighting the necessity for more adaptable methods.

H. Direct Methods

Direct methods overcome the drawbacks associated with iterative methods by generating forecasts directly through the utilization of all available inputs. Typically employing sequence-to-sequence architectures [52, 54, 56], these methods involve an encoder to succinctly summarize past information (including targets, observed inputs, and a priori known inputs) and a decoder to integrate this information with known future inputs, as illustrated in figure 2b. Another approach, detailed in [59], involves the use of simpler models to directly generate a fixed-length vector corresponding to the desired forecast horizon. However, this approach necessitates the specification of a maximum forecast horizon (τmax), with predictions only made at predefined discrete intervals.

III. CONCLUSION AND FUTURE DIRECTIONS

With the recent increase in data accessibility and computational capabilities, deep neural network architectures have garnered considerable success in addressing forecasting challenges across various domains. This article explores the primary architectures employed in time-series forecasting, shedding light on the essential components integral to neural network design. The examination delves into how these architectures incorporate temporal information to make one-step-ahead predictions and discusses their adaptability for multi-horizon forecasting. Additionally, we outline the emerging trend of hybrid deep learning models, which synergize statistical and deep learning elements to surpass the performance of purely specialized methods. Lastly, we provide an overview of two approaches to extending deep learning for enhanced decision support over time, with a focus on methods related to interpretability and counterfactual prediction.
Despite the numerous deep learning models designed for time-series forecasting, certain limitations persist. Firstly, deep neural networks typically necessitate time series to be discretized at regular intervals, posing challenges in forecasting datasets with missing observations or those arriving at random intervals. Although initial research on continuous-time models has been conducted using neural ordinary differential equations [82], further exploration is required to expand this approach for datasets featuring complex inputs (e.g., static variables) and to compare them against established models. Additionally, as highlighted in [83], time series often exhibit a hierarchical structure with logical groupings between trajectories, such as in retail forecasting where product sales in the same geography can be influenced by common trends. Consequently, there is potential for enhancing forecasting performance by developing architectures that explicitly consider such hierarchies, representing an intriguing research direction compared to existing univariate or multivariate models.

CONFLICT OF INTERREST

The authors declare that they have no conflict of interest.

REFERENCES

1. Mudelsee M. 2019 Trend analysis of climate time series: a review of methods. Earth Sci. Rev.190, 310–322. (doi:10.1016/j.earscirev.2018.12.005)
2. Stoffer DS, Ombao H. 2012 Editorial: special issue on time series analysis in the biological sciences. J. Time Ser. Anal. 33, 701–703. (doi:10.1111/j.1467-9892.2012.00805.x)
3. Topol EJ. 2019 High-performance medicine: the convergence of human and artificial intelligence. Nat. Med. 25, 44–56. (doi:10.1038/s41591-018-0300-7)
4. Böse JH, Flunkert V, Gasthaus J, Januschowski T, Lange D, Salinas D, Schelter S, Seeger M, Wang Y. 2017 Probabilistic demand forecasting at scale. Proc. VLDB Endow. 10, 1694–1705. (doi:10.14778/3137765.3137775)
5. Andersen TG, Bollerslev T, Christoffersen PF, Diebold FX. 2005 Volatility forecasting. Natl Bur. Econ. Res. 11188. (doi:10.3386/w11188)
6. Box GEP, Jenkins GM. 1976 Time series analysis: forecasting and control. San Francisco, CA: Holden-Day.
7. Gardner Jr ES. 1985 Exponential smoothing: the state of the art. J. Forecast. 4, 1–28. (doi:10.1002/for.3980040103)
8. Winters PR. 1960 Forecasting sales by exponentially weighted moving averages. Manage. Sci.6, 324–342. (doi:10.1287/mnsc.6.3.324)
9. Harvey AC. 1990 Forecasting, structural time series models and the kalman filter. Cambridge, UK: Cambridge University Press.
10. Ahmed NK, Atiya AF, Gayar NE, El-Shishiny H. 2010 An empirical comparison of machine learning models for time series forecasting. Econ. Rev. 29, 594–621. (doi:10.1080/07474938. 2010.481556)
11. Krizhevsky A, Sutskever I, Hinton GE. 2012 ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25 (NIPS) (eds F Pereira, CJC Burges, L Bottou, KQ Weinberger), pp. 1097–1105.
12. Devlin J, Chang MW, Lee K, Toutanova K. 2019 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), pp. 4171–4186.
13. Silver D et al. 2016 Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–503. (doi:10.1038/nature16961)
14. Baxter J. 2000 A model of inductive bias learning. J. Artif. Int. Res. 12, 149–198. (doi:10.1613/jair.731)
15. Bengio Y, Courville A, Vincent P. 2013 Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35, 1798–1828. (doi:10.1109/TPAMI. 2013.50)
16. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S. 2015 TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. Available from: http://tensorflow. org/.
17. Paszke A et al. 2019 PyTorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32, Vancouver, Canada, 8–14 December 2019, pp. 8024–8035.
18. Hyndman RJ, Khandakar Y. 2008 Automatic time series forecasting: the forecast package for R. J. Stat. Softw. 26, 1–22. (doi:10.18637/jss.v027.i03)
19. Nadaraya EA. 1964 On estimating regression. Theory Probab. Appl. 9, 141–142. (doi:10.1137/ 1109020)
20. Smola AJ, Schülkopf B. 2004 A tutorial on support vector regression. Stat. Comput. 14, 199–222. (doi:10.1023/B:STCO.0000035301.49549.88)
21. Williams CKI, Rasmussen CE. 1996 Gaussian processes for regression. In Advances in Neural Information Processing Systems (NIPS), Denver, CO, 27–30 November 1995.
22. Damianou A, Lawrence N. 2013 Deep Gaussian Processes. In Proc. of the Conf. on Artificial Intelligence and Statistics (AISTATS), Scottsdale, AZ, 29 April–1 May 2013.
23. Garnelo M, Rosenbaum D, Maddison CJ, Ramalho T, Saxton D, Shanahan M, Teh YW, Rezende DJ, Eslami SM. 2018 Conditional neural processes. In Proc. of the Int. Conf. on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018.
24. Waibel A. 1989 Modular construction of time-delay neural networks for speech recognition. Neural Comput. 1, 39–46. (doi:10.1162/neco.1989.1.1.39)
25. Wan E. 1994 Time series prediction by using a connectionist network with internal delay lines. In Time Series Prediction, pp. 195–217. Boston, MA: Addison-Wesley.
26. Sen R, Yu HF, Dhillon I. 2019 Think globally, act locally: a deep neural network approach to high-dimensional time series forecasting. In Advances in Neural Information Processing Systems (NeurIPS), Vancouver, Canada, 8–14 December 2019.
27. Wen R, Torkkola K. 2019 Deep generative quantile-copula models for probabilistic forecasting. In ICML Time Series Workshop, Long Beach, CA, 9–15 June 2019.
28. Li Y, Yu R, Shahabi C, Liu Y. 2018 Diffusion convolutional recurrent neural network: data- driven traffic forecasting. In Proc. of the Int. Conf. on Learning Representations ICLR, Vancouver, Canada, 30 April–3 May 2018.
29. Ghaderi A, Sanandaji BM, Ghaderi F. 2017 Deep forecast: deep learning-based spatio- temporal forecasting. In ICML Time Series Workshop, Sydney, Australia, 6–11 August 2017.
30. Salinas D, Bohlke-Schneider M, Callot L, Medico R, Gasthaus J. 2019 High-dimensional multivariate forecasting with low-rank gaussian copula processes. In Advances in Neural Information Processing Systems (NeurIPS), Vancouver, Canada, 8–14 December 2019.
31. Goodfellow I, Bengio Y, Courville A. 2016 Deep learning. Cambridge, MA: MIT Press.
32. van den Oord AV, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K. 2016 WaveNet: a generative model for raw audio. (http://arxiv.org/abs/1609.03499)
33. Bai S, Zico Kolter J, Koltun V. 2018 An Empirical evaluation of generic convolutional and recurrent networks for sequence modeling. (http://arxiv.org/abs/1803.01271)
34. Borovykh A, Bohte S, Oosterlee CW. 2017 Conditional time series forecasting with convolutional neural networks. (http://arxiv.org/abs/1703.04691)
35. Lyons RG. 2004 Understanding digital signal processing, 2nd edn. USA: Prentice Hall PTR.
36. Young T, Hazarika D, Poria S, Cambria E. 2018 Recent trends in deep learning based natural language processing [Review Article]. IEEE Comput. Intell. Mag. 13, 55–75. (doi:10.1109/MCI.2018.2840738)
37. Salinas D, Flunkert V, Gasthaus J. 2017 DeepAR: probabilistic forecasting with autoregressive recurrent networks. (http://arxiv.org/abs/1704.04110)
38. Rangapuram SS, Seeger MW, Gasthaus J, Stella L, Wang Y, Januschowski T. 2018 Deep state space models for time series forecasting. In Advances in Neural Information Processing Systems (NIPS), Montreal, Canada, 2–8 December 2018.
39. Lim B, Zohren S, Roberts S. 2020 Recurrent neural filters: learning independent Bayesian filtering steps for time series prediction. In Int. Joint Conf. on Neural Networks (IJCNN), Glasgow,UK, 19–24 July 2020.

40. Wang Y, Smola A, Maddix D, Gasthaus J, Foster D, Januschowski T. 2019 Deep factors for forecasting. In Proc. of the Int. Conf. on Machine Learning (ICML), Long Beach, CA, 9–15 June 2019.
41. Elman JL. 1990 Finding structure in time. Cogn. Sci. 14, 179–211. (doi:10.1207/ s15516709cog1402_1)
42. Bengio Y, Simard P, Frasconi P. 1994 Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. 5, 157–166. (doi:10.1109/72.279181)
43. Kolen JF, Kremer SC. 2001 In Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies, pp. 237–243.
44. Hochreiter S, Schmidhuber J. 1997 Long short-term memory. Neural Comput. 9, 1735–1780. (doi:10.1162/neco.1997.9.8.1735)
45. Srkk S. 2013 Bayesian filtering and smoothing. Cambridge, UK: Cambridge University Press.
46. Kalman RE. 1960 A new approach to linear filtering and prediction problems. Basic Eng. 82, 35. (doi:10.1115/1.3662552)
47. Bahdanau D, Cho K, Bengio Y. 2015 Neural machine translation by jointly learning to align and translate. In Proc. of the Int. Conf. on Learning Representations (ICLR), San Diego, CA, 7–9 May 2015.
48. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. 2014 Learning phrase representations using RNN Encoder–Decoder for statistical machine translation. In Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP), Dohar, Qatar, 25–29 October 2014.
49. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. 2017 Attention is all you need. In Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, 4–9 December 2017.
50. Dai Z, Yang Z, Yang Y, Carbonell J, Le Q, Salakhutdinov R. 2019 Transformer-XL: attentive language models beyond a fixed-length context. In Proc. of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy, 28 July–2 August 2019.
51. Graves A, Wayne G, Danihelka I. 2014 Neural turing machines. CoRR. abs/1410.5401.
52. Fan C, Zhang Y, Pan Y, Li X, Zhang C, Yuan R, Wu D, Wang W, Pei J, Huang H. 2019 Multi- horizon time series forecasting with temporal attention learning. In Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD).
53. Li S, Jin X, Xuan Y, Zhou X, Chen W, Wang YX, Yan X. 2019 Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In Advances in Neural Information Processing Systems (NeurIPS), Vancouver, Canada, 8–14 December2019.
54. Lim B, Arik SO, Loeff N, Pfister T. 2019 Temporal fusion transformers for interpretable multi- horizon time series forecasting. (http://arxiv.org/abs/1912.09363)
55. Choi E, Bahadori MT, Sun J, Kulas JA, Schuetz A, Stewart WF. 2016 RETAIN: an interpretable predictive model for healthcare using reverse time attention mechanism. In Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December2016.
56. Wen R, Torkkola K, Narayanaswamy B, Madeka D. 2017 A multi-horizon quantile recurrent forecaster. In NIPS 2017 Time Series Workshop.
57. Taieb SB, Sorjamaa A, Bontempi G. 2010 Multiple-output modeling for multi-step-ahead time series forecasting. Neurocomputing 73, 1950–1957. (doi:10.1016/j.neucom.2009.11.030)
58. Marcellino M, Stock J, Watson M. 2006 A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series. J. Econom. 135, 499–526. (doi:10.1016/j.jeconom.2005.07.020)
59. Makridakis S, Spiliotis E, Assimakopoulos V. 2018 Statistical and machine learning forecasting methods: concerns and ways forward. PLoS ONE 13, 1–26. (doi:10.1371/journal. pone.0194889)
60. Hyndman R. 2020 A brief history of forecasting competitions. Int. J. Forecast. 36, 7–14. (doi:10.1016/j.ijforecast.2019.03.015)
61. 2020 The M4 Competition: 100 000 time series and 61 forecasting methods. Int. J. Forecast. 36, 54–74. (doi:10.1016/j.ijforecast.2019.04.014)
62. Fildes R, Hibon M, Makridakis S, Meade N. 1998 Generalising about univariate forecasting methods: further empirical evidence. Int. J. Forecast. 14, 339–358. (doi:10.1016/S0169-2070 (98)00009-0)
63. Makridakis S, Hibon M. 2000 The M3-competition: results, conclusions and implications. Int. J. Forecast. 16, 451–476. The M3- Competition. (doi:10.1016/S0169-2070(00)00057-1)
64. Smyl S. 2020 A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. Int. J. Forecast. 36, 75–85. M4 Competition. (doi:10.1016/j. ijforecast.2019.03.017)
65. Lim B, Zohren S, Roberts S. 2019 Enhancing time-series momentum strategies using deep neural networks. J. Financ. Data Sci. 2, Fall 2020. (doi:10.2139/ssrn.3369195)
66. Grover A, Kapoor A, Horvitz E. 2015 A deep hybrid model for weather forecasting. In Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD), Sydney, Australia,10–13 August 2015.
67. Binkowski M, Marti G, Donnat P. 2018 Autoregressive convolutional neural networks for asynchronous time series. In Proc. of the Int. Conf. on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018.
68. Moraffah R, Karami M, Guo R, Raglin A, Liu H. 2020 Causal Interpretability for Machine Learning—Problems, Methods and Evaluation. (http://arxiv.org/abs/2003.03934)
69. Ribeio M, Singh S, Guestrin C. 2016 ‘Why Should I Trust You?’ Explaining the Predictions ofAny Classifier. In KDD.
70. Lundberg S, Lee SI. 2017 A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems (NIPS).
71. Simonyan K, Vedaldi A, Zisserman A. 2013 Deep inside convolutional networks: visualising image classification models and saliency maps. (http://arxiv.org/abs/1312.6034)
72. Siddiqui SA, Mercier D, Munir M, Dengel A, Ahmed S. 2019 TSViz: demystification of deep learning models for time-series analysis. IEEE Access 7, 67027–67040. (doi:10.1109/ACCESS. 2019.2912823)
73. Koh PW, Liang P. 2017 Understanding black-box predictions via influence functions. In Proc. of the Int. Conf. on Machine Learning (ICML).
74. Bai T, Zhang S, Egleston BL, Vucetic S. 2018 Interpretable representation learning for healthcare via capturing disease progression through time. In Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining (KDD), London, UK, 19–23 August, 2018.
75. Yoon J, Jordon J, van der Schaar M. 2018 GANITE: estimation of individualized treatment effects using generative adversarial nets. In Int. Conf. on Learning Representations (ICLR), Sydney, Australia, 6–11 August 2017.
76. Hartford J, Lewis G, Leyton-Brown K, Taddy M. 2017 Deep IV: a flexible approach for counterfactual prediction. In Proc. of the 34th Int. Conf. on Machine Learning (ICML), London, UK, 19–23 August 2018.
77. Alaa AM, Weisz M, van der Schaar M. 2017 Deep counterfactual networks with propensity dropout. In Proc. of the 34th Int. Conf. on Machine Learning (ICML), Vancouver, Canada, 30 April–3May 2018.

78. Mansournia MA, Etminan M, Danaei G, Kaufman JS, Collins G. 2017 Handling time varying confounding in observational research. BMJ 16, 359. (doi:10.1136/bmj.j4587)
79. Lim B, Alaa A, van der Schaar M. 2018 Forecasting treatment responses over time using recurrent marginal structural networks. In NeurIPS.
80. Li R, Shahn Z, Li J, Lu M, Chakraborty P, Sow D, Ghalwash M, Lehman LW. 2020 G-Net: a deep learning approach to G-computation for counterfactual outcome prediction under dynamic treatment regimes. (http://arxiv.org/abs/2003.10551)
81. Bica I, Alaa AM, Jordon J, van der Schaar M. 2020 Estimating counterfactual treatment outcomes over time through adversarially balanced representations. In Int. Conf. on LearningRepresentations(ICLR).
82. Chen RTQ, Rubanova Y, Bettencourt J, Duvenaud D. 2018 Neural ordinary differential equations. In Proc. of the Int. Conf. on Neural Information Processing Systems (NIPS).
83. Fry C, Brundage M. 2019 The M4 forecasting competition–a practitioner's view. Int. J. Forecast. 36, 156–160. (doi:10.1016/j.ijforecast.2019.02.013)