Financial Time Series Forecasting with Deep Generative Models

Literature Survey

Piotr Borowiecki
Durham University
piotr.borowiecki@durham.ac.uk

I. Related Work

A. Traditional Approaches to Time Series Forecasting

Early investigations of time series, especially in the nineteenth century, were commonly influenced by the concept of a deterministic universe, until a significant contribution from Yule [1], who suggested that any time series might be viewed as the actualization of a stochastic process. The notion of autoregressive (AR) and moving average (MA) models was conceived by Slutsky, Walker, Yaglom, and Yule. In 1940, Kolmogorov formulated and solved the linear forecasting problem, by utilizing Wold's decomposition theorem [2]. Throughout the 1960s, the selection of a model depended heavily on the discretion of the researcher, since no algorithm existed to aid the process. Consequently, several techniques, such as Akaike's information criterion (AIC) [3], or Bayes information criterion (BIC) [4] , have been proposed to increase mathematical precision in the formulation of an Autoregressive Moving Average (ARMA) model [5]. The 1970 publication, ''Time Series Analysis: Forecasting and Control'' [6], compiled time series knowledge available at the time and introduced the Box-Jenkins method - uniform, adaptable, regressive and iterative approach, based on the assumption that past occurrences influence future events. A substantial amount of academic research on time series forecasting has been published since then. De Gooijer and Hyndman [7] provide a thorough overview of years 1982 - 2005, for example.

In the 1980s, exponential smoothing was frequently perceived as a minor, auxiliary set of method for univariate time series analysis and thus, received relatively little attention. These methods were initially developed by Brown [8, 9], Holt [10], and Winters [11]. Statistical underpinnings for Simple Exponential Smoothing (SES) was first provided in 1960 by Muth, who demonstrated that it delivers the most accurate predictions for a random walk with noise [12]. Further work towards incorporating exponential smoothing into a statistical context was delivered by Box and Jenkins [6] Roberts [13], or Abraham and Ledolter [14], who demonstrated that some linear exponential smoothing predictions emerge as special cases of the Autoregressive Integrated Moving Average (ARIMA) model. Two papers published in 1985 [15, 16], set the foundation for most of the subsequent research in that area. SES, Holt's linear trend method [17] and Holt-Winter's method [18] are among the most popular and commonly utilized exponential smoothing techniques. SES with a drift is a useful variation of the original and corresponds to Holt's method, when the trend parameter is set to zero. It was noted in 2003 that this approach is also comparable to Assimakopoulos and Nikolopoulos's Theta method [19], if the drift value is set to half the slope of a linear trend fitted to the data. The Theta technique scored exceptionally high in the M3-competition, however it remains unknown why this specific model and parameter selection is effective [20]. Other noteworthy autoregressive models include Vector Autoregression (VAR) [21], Vector Exponential Smoothing (VES) [22], Autoregressive Conditional Heteroskedasticity model (ARCH) [23], or Generalized Autoregressive Conditional Heteroskedasticity model (GARCH) [24]. VARs are vector auto-regressive models with lag sequences derived from time series data periods. In general, VAR models are susceptible to the overfitting problem, due to an excess of irrelevant free parameters. As a consequence, these models are unable to give accurate out-of-sample projections, despite strong within-sample fitting [25, 26]. All ARCH models characterize the dynamic changes in conditional variance as a deterministic (usually quadratic) function of historical returns. Due to the fact that the variance is known at time $t-1$, one-step-ahead forecasts are readily available, and multi-step-ahead forecasts can be computed recursively. Since extra dependencies on delays of the conditional variance are allowed, the GARCH model is simpler and more efficient. It has an ARMA-like representation, therefore many of the characteristics of both models are comparable [7].

Although the concepts have been available since the invention of Kalam filters [27], state space models were rarely used for time series forecasting until the 1980s. In 1965, Schwepper [28] demonstrated that Kalman filters are suitable for calculating one-step-ahead prediction errors and variances, required to generate the likelihood function [7]. Shumway and Stoffer [29] eventually merged Expectation-Maximization (EM) algorithm [30] and Kalman filters, to provide a universal method for predicting time series using state space models.

B. Time Series Forecasting with Deep Learning

It is commonly argued that conventional approaches to time series forecasting not always adequately capture non-linear relationships in data, and therefore, other techniques should be utilized to uncover such patterns. In recent years, deep learning has been successfully applied in the domain of time series forecasting [31, 32], as well as for pre-training, clustering, or distance calculation [33, 34, 35]. A rather common theme across literature is the use of stacked autoencoders [36], transformers [37, 38], or Recurrent Neural Networks (RNNs) [39] with Long Short-term Memory (LSTMs) [40] in particular, on numeric time series data. Models such as Deep Global Local Forecaster (DeepGlo) [41], Long- and Short-term Time-series Network (LSTNet) [42], DeepAR [43], Multi-horizon Quantile Recurrent Neural Network (MQRN) [44], DeepState [45], or Temoral Fusion Transformer (TFT) [46] have been proposed, producing outstanding results. DeepGlo is built on a structure of global matrix factorization, regularized by a temporal convolutional network. LSTNet emphasizes both long-term relationships, which are recorded by a recurrent network structure and local multivariate patterns, which are modelled by a convolutional layer. Through the use of an autoencoder to simulate a probabilistic distribution, DeepAR blends RNNs and conventional autoregressive models. It uses extra time- and categorical variables to estimate parametric distributions from time series. A probabilistic generative model called DeepState uses RNNs to train it to parametrize a linear state space. The TFT architecture is effective in capturing long-term dependencies in the data by fusing self-attention layers typical of transformers with recurrent layers for local processing.

C. State-of-the-art in Time Series Forecasting

Despite the plethora of algorithms available, no single approach consistently outperforms others all the others in all situations [47]. Fusing models generally results in superior prediction accuracy, which has been repeatedly shown in numerous studies [48, 49, 50]. Raftery et al. [51] noted that single model selection is susceptible to problems with model uncertainty and suggested Bayesian model averaging for stacking as a substitute. Cawood and van Zyl [48] provided a thorough overview of ensembling strategies and suggested that model averaging is more reliable than model selection or stacking. Since time series are rarely purely linear or nonlinear in reality, hybrid models — first suggested by Zhang in 2003 [52] — fuse both perspectives, providing more accurate results. Due to their predictive accuracy, hybrid machine learning algorithms are now the favored category in time series prediction research [53].

Both of the highest rated entries in the M4 forecasting competitions [50] - Feature-based FORecast Model Averaging (FFORMA) [54] and xponential Smoothing - Recurrent Neural Network (ES-RNN) [55] - utilize ensemble learning techniques to achieve state of the art results. To learn the weightings for base learners, depending on multiple meta-features derived from the input data, FFORMA exploits stacking, by employing extreme gradient boosting. Each model's predictions are then combined to compute a feature-weighted average. The winning entry of the M4 Competition, ES-RNN is a hybrid forecasting model that combines a modified Holt-Winters and dilated LSTM stacks. Another cutting-edge model, Neural Basis Expansion Analysis (N-BEATS) [56], has obtained a 3 % accuracy improvement over ES-RNN. Another state-of-the-art model - Neural Basis Expansion Analysis (N-BEATS) [56] - has recently achieved a 3% accuracy increase over ES-RNN. N-BEATS proposes a unique doubly residual topology that stacks blocks of individual architectures to model various time-series components, while integrating a fully connected neural network with conventional time-series decomposition. Although still unique, it resembles the DenseNet architecture introduced by Huang et al. [57]. More recently, the M5 Competition led research efforts to validate the superiority of meta-model fusion in predicting homogenous data, and was the first in the series of M-competitions using hierarchical time-series data [49].

D. Time Series Forecasting as a Computer Vision Problem

1) Introduction

A relatively new approach to time series forecasting is the use of deep generative models, already yielding promising results. The primary objective of generative modeling is to create a model capable of representing a data distribution, given training samples from that distribution [36]. This is accomplished mostly via Density Estimation and Sample Generation. The goal in both situations is to record a probability distribution and match it to the true distribution of the input data. Visual time series forecasting is consequently a data-driven, non-parametric technique, unrestricted by a predefined set of parameters. In contrast to traditional time series forecasting methods, which are frequently adapted to the particulars of the data in issue, this method is flexible and adaptable to several data types.

In general, there are four types of deep generative modeling methods: autoregressive, flow-based, energy-based, and latent variable models [58]. Variational Auto-encoders (VAEs) [59] and Generative Adversarial Networks (GANs) [60] are two examples of latent variable models that are most frequently found in the literature. Some of the most notable positions of this kind were identified as [61, 62].

2) Autoregressive and Flow-based Generative Models

Generative methods frequently rely on tractable distribution classes or low-rank approximations to model the true distribution [63]. Several deep learning approaches, such as autoregressive [64] and generative, based on normalizing flows [65], have been suggested for this purpose.

Autoregressive Models (ARMs) are regarded as robust density estimators and one of the finest likelihood-based generative models, since they are capable of learning long-range statistics. Due to the autoregressive approach to creating new information, however, their sampling procedure is incredibly sluggish. Nonetheless, predictive sampling [66, 67] addresses this issue, providing considerable improvements. The primary disadvantage is that these models are less flexible thab others and demand a particular structure in the functional approximators, on the determinant of the Jacobian matrix, for example [68]. Additionally, they lack a latent representation and it is unclear how to change their internal data, which makes them less suitable for problems, such as compression or metric learning [69].

Flow-based models (Flows) are generative models that take advantage of the change of variables formula. The change of variables formula offers a logical way to represent the density of a random variable by modifying it through an invertible transformation f, such that $p(\mathbf{x}) = p(\mathbf{z} = f(\mathbf{x})) \left|\mathbf{J}_{f(\mathbf{x})}\right|$, where $\mathbf{J}_{f(x)}$ denotes the Jacobian matrix. As long as the Jacobian matrix can be computed, the invertible transformation may be parametrized by any neural network. Initial attempts centered on linear, volume-preserving transformations, such that $\left|\mathbf{J}_{f(\mathbf{x})}\right| = 1$ [70, 71], followed by Planar Flows [72] and Sylvester flows [73], utilizing matrix determinants properties.

Another strategy relies on designing invertible transformations for which the Jacobian-determinant can be easily determined, similar to the coupling layers in RealNVP [74]. Recently, arbitrary neural networks have been bound to be invertible, when the Jacobian-determinant is approximated, instead of being calculated directly [75, 76, 77].

Since ARMs and Flows are density estimators and provide exact likelihood values [58], they can both be used for lossless compression. In ARMs, conditional distributions can be factored and parameterized, while in Flows, the change of variables formula can be implemented using neural networks.

3) Latent Variable Generative Models

The key assumption underpinning latent variable models is that there exists a lower-dimensional hidden space, also known as representation of compressed data. Given a high-dimensional object, $\mathbf{x} \in X^D$ (such as an image, where $X \in \{0, 1, \ldots, 255\}$), and low-dimensional latent variables (low-dimensional manifold), $\mathbf{z} \in \mathcal{Z}^M$ (e.g., $\mathcal{Z} = \mathbb{R}$), the sampling process is such that $\mathbf{z} \sim p(\mathbf{z})$ and $\mathbf{x} \sim p(\mathbf{x} \mid \mathbf{z})$. We first sample z (which could be understand as marking the basic structure, such as size, shape or color), and then add further details, by sampling from the conditional distribution $p(x|z)$. The joint distribution is factorized as follows: $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})$, and the marginal likelihood function is $p(\mathbf{x}) = \int p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z}$.

GANs are generative models typically composed of two neural networks, a generator and a discriminator, trained against each other. In recent year,s they have been successfully used for image and audio generation, sequence forecasting, or imputation. Since its inception in 2014, many improvements over the original GAN or similar models were proposed, such as Deep Convolutional Generative Adversarial Networks (DCGANs) [78], Wasserstein Generative Adversarial Network [79], Progressive Growing of GANs [80], StyleGAN [81] or BigGAN [82], just to name a few. Brophy, Wang, She and Ward [83] provide an overview of GANs applied to time series. Despite their ability to produce convincing, realistic images, these architectures are still difficult to train, unstable, and often suffer from the mode collapse problem. For those reasons, other latent variable- and score-based models will receive the most coverage throughout the project.

Another example of models based on latent variables are Variational Autoencoders. In contrast to flow-based models, they are adaptable, which means that encoders and decoders can be built with any neural network. Due to their flexibility, it is feasible to construct extremely deep structures, capable of delivering astonishing results. The most important, recent examples of such architecture are BIVA [84], NVAE [85], and very deep VAEs [86]. Another noteworthy perspective on deep hierarchical VAEs was proposed by Gatopoulos, Ioannis and Tomczak [87], who additionally employed a series of deterministic functions.

In contrast to autoregressive models, VAEs they are able to learn a low-dimensional data representation (latent space), dimensionality of which can be controlled with architectural choices. Nevertheless, they also are not free of potential issues, such as posterior collapse [88], the Hole Problem [89], or the necessity for an efficient integral estimation method. When compared with each other, GANs are capable of producing high-quality, realistic results and are difficult to train, whereas VAEs are easy to train, but the results are often blurry due to minimizing the reconstruction error.

4) Energy-based Generative Models

In energy-based models (EBMs), $p_\theta(x)$ may be expressed as $p_\theta(x) = \frac{e^{-f_\theta(x)}}{Z_\theta}$, where $f_\theta(x)$ is an energy function parameterized by θ , and Z_θ is the normalization constant. Song and Kingma [90] describe the process of training energy-based models and show that EBMs perform well in learning high dimensional data distributions, at the cost of being difficult to train. Brakel, Stroobandt and Schrauwen [91] describe the training process, when related to time-series imputation.

TimeGrad [63] is an autoregressive model that samples the data distribution at each time step by estimating its gradient. At inference time, the model transforms white noise into a sample of the distribution of interest, using a Markov chain and Langevin sampling. The model learns gradients by optimizing a variational constraint on the data likelihood and has a close relationship with EBMs and Score Matching. The suggested autoregressive denoising diffusion model is a cutting-edge multivariate probabilistic forecasting approach on real-world datasets. It combines advantages of autoregressive models, such as strong performance in extrapolating into the future, with the adaptability of EBMs as general-purpose high-dimensional distribution models, while staying computationally tractable.

As highlighted by Du and Mordatch [92], EBMs exhibit better out-of-distribution (OOD) detection than other likelihood models. Such task demands high likelihood on the data manifold and a low likelihood everywhere else. Despite advantages, such as flexibility, stability of training (no mode collapse issue), and a relatively high sample quality - energy-based models have shortcomings. To obtain a sample, the Monte Carlo procedure must be executed, which is often computationally expensive. Since humans frequently operate on high-dimensional data, this is still a significant barrier to the widespread application of EBMs in practice.

5) Diffusion-based and Score-based Generative Models

Diffusion-based Deep Generative Models (DDGMs) [93] can be perceived as infinitely deep hierarchical VAEs with a particular type of variational posteriors [94, 95, 96], namely, Gaussian diffusion processes [93]. As noted by several authors, the Evidence Lower Bound (ELBO) may be used as the objective function in both DDGMS and VAEs, tightening the relationship between the two [97]. In [94], the authors emphasized continuous diffusion models and provided a bridge to infinitely deep hierarchical VAEs. This connection has recently been investigated further by Kingma, Salimans, Poole, and Ho [98], who defined a Variational Lower Bound (VLB) learning objective in terms of the signal-to-noise ratio, bringing the forward diffusion process even closer to VAEs. The recently suggested Latent Score-based Generative Model (LSGM) [99] can be treated as a VAE with a score-based prior. Since variational posteriors are fixed, we may see them as Gaussian noise added to each layer [95]. Such approach solves the potential Posterior Collapse issue in VAEs, because the standard Gaussian distribution is arrived at in the final layer by design [95]. Since these models are strongly linked to stochastic differential equations, their properties have been extensively researched in recent years [100, 101, 102], which could also be attributed to the unprecedented quality of generated samples [103, 104, 105].

The general idea behind this set of methods is to generate samples by utilizing diffusion processes [106, 94, 107, 100]. In the forward diffusion process, an image is passed through a number of steps that consecutively add a small portion of noise to it, whereas the backward diffusion is the exact opposite, where a generative model is taught to progressively denoise the input. Given sufficient number of forward diffusion steps, images become indistinguishable from the isotropic Gaussian noise. New images are then generated by applying the backward diffusion to the noise sampled from the standard Gaussian distribution.

Originating from non-equilibrium statistical physics, deep diffusion probabilistic models were first proposed in 2015 [93]. In a subsequent study [106], recent advances in deep learning were leveraged to train a robust and adaptable diffusion-based deep generative model, achieving state-of-the-art results in synthesizing new images. Nichol and Dhariwal [108] provided further improvements to the training stability and performance of DDGMs.

Potential drawbacks and limitations are such that these models are incapable of learning a latent feature space, since the input dimensions are maintained throughout the entire training process, similarly to flow-based models. VAEs, Flows, and DDGMs exhibit a multitude of commonalities when compared to one another. Hierarchical VAEs could be seen as a generalization of DDGMs. It remains an open debate, however, whether adopting fixed variational posteriors at the expense of having the latent representation is truly advantageous. Additionally, there is a relationship between flows and DDGMs. Both categories of models seek to go from data to noise. Flows do this by using invertible transformations, while DDGMs use a diffusion process. In flows, the inverse is known at the expense of computing the Jacobian-determinant, while DDGMs need flexible parameterization of the reverse diffusion, with no additional constraints.

According to Song and Kingma [90], Score-based Deep Generative Models (SDGMs) [107] are also strongly connected to DDGMs. They are based on the idea of Score Matching [109] and exhibit numerous similarities to latent variable models. Instead of the probability density function (pdf), however, gradients with respect to input dimensions are used. Addressing many of the challenges associated with traditional approaches to image synthesis, SDGMs appear to be gaining popularity lately, after demonstrating a potential of achieving similar results, or even outperforming current state-of-the-art solutions [107]. Given a probability density function $p(x)$, the score is defined as $\nabla_{\mathbf{x}} \log p(\mathbf{x})$, and is a vector field of the gradient at each location x, suggesting a direction in which to move, to maximize the likelihood. In contrast to likelihood-based models, such as flows or autoregressive models, SDGMs are easy to parametrize and do not require normalization. Unlike likelihood-based models, such as normalizing flows or autoregressive models, SGMs are easier to parameterize and do not need to be normalized. Score function is also unconstrained, making it simpler to model. In contrast to regular EBMs, score function is independent of the partition function, Z_θ, hence no intractability issues occur.

Existing score-based models, include Noise Conditional Score Networks (NCSN) [107], NCSN 2 [110], Denoising Diffusion Probabilistic Model (DDPM) [106], DDPM++ [100], Latent Space Generative Model (LSGM) [111], Critically Damped Langevin Diffusion Score-based Generative Model (CLD-SGM) [112], and Denoising Auto-Encoder and a Diffusion-based Generator (DAED) [97].

Recent applications of this model family also include time series forecasting. ScoreGrad [113] is a probabilistic multivariate time series forecasting system built on energy-based continuous generative models, achiving state-of-the-art results on real-world datasets. It consists of a time series feature extraction module and a score matching module based on conditional stochastic differential equations (SDEs). Predictions are arrived at by systematically solving reverse-time SDEs. It is perhaps the first every continuous energy-based generative model utilized for predicting time series.

In reality, score-based generative models are not easily since calculating scores quickly becomes computationally too expensive. Additionally, according to the Manifold Hypothesis, high dimensional data often resides in a low dimensional manifold, causing $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ to be undefined in some regions.

6) Limitations of Deep Generative Models

Deep Generative Models, such as VAEs, are incapable of detecting out-of-distribution instances [114]. Out-of-distribution data points are instances that adhere to a completely different distribution from the one used to train a model. A well trained deep generative model should assign a high probability to instances inside the distribution and a low probability to points outside the distribution. Sadly, this is not the case. The out-of-distribution problem remains one of the most significant unresolved issues in deep generative modeling [115].

E. Time Series Forecasting with Simple Machine Learning

Despite their effectiveness in forecasting time series, a significant downside of deep learning approaches to the problem is that they can be overly complex and opaque. Often referred to as ”black boxes”, deep models identify underlying patterns in data and make judgments that cannot be fully explained or comprehended by humans. This is especially problematic in medical applications, where being able to fully understand how models arrive at certain decisions is of uttermost importance.g with a diagnosis label. Chatfield questioned whether artificial neural networks (ANNs) had not been overestimated as a miraculous approach to predicting [116, 117]. Several authors subsequently demonstrated that simpler models, such as the random walk, may outperform ANNs [118, 119].

Among the numerous methods aimed at achieving precision and minimizing errors and losses within time series forecasting, there are several classical and modern machine learning methods that proved their accuracy and computational relevance. Gradient Boosting Regression Tree (GBRT), for example, is a versatile, non-parametric statistical learning tool for both classification and regression. The idea was first introduced by Breiman [120], who observed that boosting may be utilized as an optimization algorithm.

F. Evaluation Metrics and Measures

Some of the most common evaluation metrics and measures encountered in the literature are Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Median Absolute Error (MdAE), Mean Absolute Percentage Error (MAPE), Mean Absolute Scaled Error (MASE) [121], Symmetric Mean Absolute Percentage Error (sMAPE) [122], and Percentage Better (PB). MSE, MAPE, MdAPE, and PB were among the measures employed in first M-competitions [123]. It was pointed out later, however, that MSE is not appropriate for comparison between series as it is scale dependent [124, 125].

Weighted Average (OWA) combines sMAPE and MASE, respectively calculated as:

$$sMAPE = \frac{1}{h}\sum_{t=1}^{h}\frac{2\left|Y_t - \hat{Y}_t\right|}{|Y_t| + \left|\hat{Y}_t\right|} \times 100\%,$$

$$MASE = \frac{1}{h}\frac{\sum_{t=n+1}^{n+h}\left|Y_t - \hat{Y}_t\right|}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|'},$$

where Y_t is the actual time-series value at time step t, $\hat{Y}_t$ is the predicted value of Y_t and h the length of the forecasting horizon. The denominator and scaling factor of the MASE formula are the in-sample mean absolute error from one-step-ahead predictions of the Naïve model 2, and n is the number of data points. The term m defines the time interval between each successive observation, i.e., 12 for time-series that have a monthly frequency. The OWA error is then computed as the average of the MASE and sMAPE errors relative to the Naïve model 2 predictions as follows:

$$OWA = \frac{1}{2}\left(\frac{MASE}{MASE_{\text{Naive 2}}} + \frac{sMAPE}{sMAPE_{\text{Naive 2}}}\right)$$

References

[1] George Udny Yule. On a method of investigating periodicities in disturbed series, with special reference to wolfer’s sunspot numbers. *Philosophical Transactions of the Royal Society of London Series A*, 226:267–298, 1927.

[2] Andreĭ Nikolaevich Kolmogorov. *Stationary sequences in Hilbert space*. 1940.

[3] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[4] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[5] Peter Whittle. *Hypothesis testing in time series analysis*, volume 4. Almqvist & Wiksells boktr., 1951.

[6] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[7] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473, 2006.

[8] Robert Goodell Brown. Statistical forecasting for inventory control. 1959.

[9] Robert Goodell Brown. *Smoothing, forecasting and prediction of discrete time series*. Courier Corporation, 2004.

[10] Charles C Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.

[11] Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

[12] John F Muth. Optimal properties of exponentially weighted forecasts. *Journal of the american statistical association*, 55(290):299–306, 1960.

[13] SA Roberts. A general class of holt-winters type forecasting models. *Management Science*, 28(7):808–820, 1982.

[14] Bovas Abraham and Johannes Ledolter. Forecast functions implied by autoregressive integrated moving av-

erage models and other related forecast procedures. *International Statistical Review/Revue Internationale de Statistique*, pages 51–66, 1986.

[15] Everette S Gardner Jr. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):1–28, 1985.

[16] RD Snyder. Recursive estimation of dynamic linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 272–276, 1985.

[17] C Holt. C.(1957). forecasting seasonals and trends by exponentially weighted averages. *Office of Naval Research*, 1957.

[18] Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

[19] Vassilis Assimakopoulos and Konstantinos Nikolopoulos. The theta model: a decomposition approach to forecasting. *International journal of forecasting*, 16(4):521–530, 2000.

[20] Spyros Makridakis and Michele Hibon. The m3-competition: results, conclusions and implications. *International journal of forecasting*, 16(4):451–476, 2000.

[21] Christopher A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, pages 1–48, 1980.

[22] Ashton De Silva, Rob J Hyndman, and Ralph Snyder. The vector innovations structural time series framework: a simple approach to multivariate forecasting. *Statistical Modelling*, 10(4):353–374, 2010.

[23] Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007, 1982.

[24] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

[25] Te-Ru Liu, Mary E Gerlow, and Scott H Irwin. The performance of alternative var models in forecasting exchange rates. *International Journal of Forecasting*, 10(3):419–433, 1994.

[26] Scott Simkins. Forecasting with vector autoregressive (var) models subject to business cycle restrictions. *International Journal of Forecasting*, 11(4):569–583, 1995.

[27] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[28] Fred Schweppe. Evaluation of likelihood functions for gaussian signals. *IEEE transactions on Information Theory*, 11(1):61–70, 1965.

[29] Robert H Shumway and David S Stoffer. An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis*, 3(4):253–264, 1982.

[30] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[31] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944, 2017.

[32] Alaa Sagheer and Mostafa Kotb. Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing*, 323:203–213, 2019.

[33] Abubakar Abid and James Y Zou. Learning a warping distance from unlabeled time series using sequence autoencoders. *Advances in neural information processing systems*, 31, 2018.

[34] S Mostafa Mousavi, Weiqiang Zhu, William Ellsworth, and Gregory Beroza. Unsupervised clustering of seismic signals using deep convolutional autoencoders. *IEEE Geoscience and Remote Sensing Letters*, 16(11):1693–1697, 2019.

[35] Alaa Sagheer and Mostafa Kotb. Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems. *Scientific reports*, 9(1):1–16, 2019.

[36] Naftali Cohen, Srijan Sood, Zhen Zeng, Tucker Balch, and Manuela Veloso. Visual time series forecasting: An image-driven approach. *arXiv preprint arXiv:2011.09052*, 2020.

[37] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *arXiv preprint arXiv:2205.13504*, 2022.

[38] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

[39] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[40] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[41] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32, 2019.

[42] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[43] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[44] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

[45] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim

Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018.

[46] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. *arXiv preprint arXiv:2002.06103*, 2020.

[47] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[48] Pieter Cawood and Terence van Zyl. Evaluating state of the art, forecasting ensembles-and meta-learning strategies for model fusion. *arXiv preprint arXiv:2203.03279*, 2022.

[49] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 2022.

[50] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020.

[51] Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial (with comments by m. clyde, david draper and ei george, and a rejoinder by the authors. *Statistical science*, 14(4):382–417, 1999.

[52] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[53] Gitanjali R Shinde, Asmita B Kalamkar, Parikshit N Mahalle, Nilanjan Dey, Jyotismita Chaki, and Aboul Ella Hassanien. Forecasting models for coronavirus disease (covid-19): a survey of the state-of-the-art. *SN Computer Science*, 1(4):1–15, 2020.

[54] Pablo Montero-Manso, George Athanasopoulos, Rob J Hyndman, and Thiyanga S Talagala. Fforma: Feature-based forecast model averaging. *International Journal of Forecasting*, 36(1):86–92, 2020.

[55] Slawek Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020.

[56] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

[57] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[58] Jakub M Tomczak. Why deep generative modeling? In *Deep Generative Modeling*, pages 1–12. Springer, 2022.

[59] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[60] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[61] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *arXiv preprint arXiv:2103.04922*, 2021.

[62] Lei Wang, Wei Chen, and Wenjia Yang. Fangming bi, and fei richard yu. a state-of-the-art review on image synthesis with generative adversarial networks. *IEEE Access*, 8(63514-63537):2, 2020.

[63] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868. PMLR, 2021.

[64] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[65] George Papamakarios, Eric T Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64, 2021.

[66] Auke Wiggers and Emiel Hoogeboom. Predictive sampling with forecasting autoregressive models. In *International Conference on Machine Learning*, pages 10260–10269. PMLR, 2020.

[67] Yang Song, Chenlin Meng, Renjie Liao, and Stefano Ermon. Accelerating feedforward computation via parallel nonlinear equation solving. In *International Conference on Machine Learning*, pages 9791–9800. PMLR, 2021.

[68] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[69] Jakub M Tomczak. Flow-based models. In *Deep Generative Modeling*, pages 57–128. Springer, 2022.

[70] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[71] Jakub M Tomczak and Max Welling. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

[72] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[73] Rianne van den Berg, Leonard Hasenclever, Jakub M Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. *arXiv preprint*

arXiv:1803.05649, 2018.

[74] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[75] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019.

[76] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019.

[77] Yura Perugachi-Diaz, Jakub Tomczak, and Sandjai Bhulai. Invertible densenets with concatenated lipswish. *Advances in Neural Information Processing Systems*, 34:17246–17257, 2021.

[78] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[79] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[80] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[81] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[82] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[83] Eoin Brophy, Zhengwei Wang, Qi She, and Tomas Ward. Generative adversarial networks in time series: A survey and taxonomy. *arXiv preprint arXiv:2107.11098*, 2021.

[84] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. *Advances in neural information processing systems*, 32, 2019.

[85] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

[86] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

[87] Ioannis Gatopoulos and Jakub M Tomczak. Self-supervised variational auto-encoders. *Entropy*, 23(6):747, 2021.

[88] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[89] Danilo Jimenez Rezende and Fabio Viola. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

[90] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

[91] Philémon Brakel, Dirk Stroobandt, and Benjamin Schrauwen. Training energy-based models for time-series imputation. *The Journal of Machine Learning Research*, 14(1):2771–2797, 2013.

[92] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

[93] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[94] Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34:22863–22876, 2021.

[95] Jakub M Tomczak. Latent variable models. In *Deep Generative Modeling*, pages 57–128. Springer, 2022.

[96] Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

[97] Kamil Deja, Anna Kuzina, Tomasz Trzciński, and Jakub M Tomczak. On analyzing generative and denoising capabilities of diffusion-based deep generative models. *arXiv preprint arXiv:2206.00070*, 2022.

[98] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[99] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

[100] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[101] Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34:22863–22876, 2021.

[102] Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

[103] Prafulla Dhariwal and Alexander Nichol. Diffusion

models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[104] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47–1, 2022.

[105] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[106] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[107] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[108] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[109] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[110] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[111] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

[112] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. *arXiv preprint arXiv:2112.07068*, 2021.

[113] Tijin Yan, Hongwei Zhang, Tong Zhou, Yufeng Zhan, and Yuanqing Xia. Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. *arXiv preprint arXiv:2106.10121*, 2021.

[114] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

[115] Charline Le Lan and Laurent Dinh. Perfect density models cannot guarantee anomaly detection. *Entropy*, 23(12):1690, 2021.

[116] Chris Chatfield et al. Neural networks: Forecasting breakthrough or passing fad? *International Journal of Forecasting*, 9(1):1–3, 1993.

[117] Chris Chatfield. Positive and negative? *International Journal of Forecasting*, 11(4):501–502, 1995.

[118] Keith B Church and Stephen P Curram. Forecasting consumers' expenditure: A comparison between econometric and neural network models. *International journal of forecasting*, 12(2):255–267, 1996.

[119] Antonio J Conejo, Javier Contreras, Rosa Espínola, and Miguel A Plazas. Forecasting electricity prices for a day-ahead pool-based electric energy market. *International journal of forecasting*, 21(3):435–462, 2005.

[120] Leo Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at . . . , 1997.

[121] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[122] Spyros Makridakis. Accuracy measures: theoretical and practical concerns. *International journal of forecasting*, 9(4):527–529, 1993.

[123] Spyros Makridakis, Allan Andersen, Robert Carbone, Robert Fildes, Michele Hibon, Rudolf Lewandowski, Joseph Newton, Emanuel Parzen, and Robert Winkler. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of forecasting*, 1(2):111–153, 1982.

[124] Chris Chatfield. What is the 'best'method of forecasting? *Journal of Applied Statistics*, 15(1):19–38, 1988.

[125] J Scott Armstrong and Fred Collopy. Error measures for generalizing about forecasting methods: Empirical comparisons. *International journal of forecasting*, 8(1):69–80, 1992.