Sample Path Generation for Probabilistic Demand Forecasting

Dhruv Madeka*
Forecasting Data Science, Amazon
New York, New York
maded@amazon.com

Lucas Swiniarski**
New York University, Center for Data Science
Forecasting Data Science, Amazon
New York, New York
swilucas@amazon.com

Dean Foster
SCOT, Amazon
New York, New York
foster@amazon.com

Leo Razoumov
Forecasting Data Science, Amazon
New York, New York
razoumov@amazon.com

Kari Torkkola
Forecasting Data Science, Amazon
Seattle, Washington
karito@amazon.com

Ruofeng Wen
Forecasting Data Science, Amazon
Seattle, Washington
ruofeng@amazon.com

ABSTRACT

The state of the art in probabilistic demand forecasting [40] minimizes Quantile Loss to predict the future demand quantiles for different horizons. However, since quantiles aren't additive, in order to predict the total demand for any wider future interval all required intervals are usually appended to the target vector during model training. The separate optimization of these overlapping intervals can lead to inconsistent forecasts, i.e. forecasts which imply an invalid joint distribution between different horizons. As a result, inter-temporal decision making algorithms that depend on the joint or step-wise conditional distribution of future demand cannot utilize these forecasts. In this work, we address the problem by using sample paths to predict future demand quantiles in a consistent manner and propose several novel methodologies to solve this problem. Our work covers the use of covariance shrinkage methods, autoregressive models, generative adversarial networks and also touches on the use of variational autoencoders and Bayesian Dropout.

CCS CONCEPTS

• **Applied Computing** → **Forecasting**;

KEYWORDS

time series, demand forecasting, supply chain management, generative adversarial networks

1 INTRODUCTION

Demand Forecasting plays a central role in any inventory management system. Given past demand series, effective forecasts seek to characterize the probability distribution of future demand in order to improve decision making across the entirety of a supply chain, from buying decisions [23], to inventory management [28] and aggregate financial planning. The problem lies in the domain of probabilistic time series forecasting. Traditionally, time series models such as ARIMA models [5] are used to predict the next value of a demand series as a function of the previous values. Recent methods, such as [4], [2], [14] utilize artificial neural networks such as Feed-Forward Networks or Recurrent Neural Networks (RNN) [11], along with likelihood based loss functions to produce time series forecasts. Recently, [40] showed that an RNN trained using Quantile Loss (QL) [25] can produce state of the art results, by directly optimizing for the relevant quantiles of all future time steps. Specifically, the RNN outputs the distribution quantiles of the time series D_t at any future horizon t with $t < T$.

The horizon-specific quantile forecast is sufficient for most time series applications (such as risk management, energy capacity, weather forecasting etc) but Demand Forecasting requires that distributional properties for the sum of future demand be characterized as well. For a given future planning period $[l, l+s]$, where l (lead-time) is the distance to when the buying period begins and s (span) is the duration of the buying period, quantiles for the total demand $D_{[l,l+s]} = \sum_{t\in[l,l+s]} D_t$ are needed, because they provide the optimal inventory level for a single period newsvendor problem applied on that interval[23]. Horizon-specific quantile forecasts at any step $t \in [l, l+s]$ only characterize the marginal distribution at t. Since quantiles are non-additive, generating quantiles of $D_{[l,l+s]}$ usually requires the joint distribution of $(D_l, \cdots, D_{l+s})$. To bridge this gap, a simple solution is to directly forecast $D_{[l,l+s]}$ itself, in addition to $D_t, t \in [l, l+s]$, i.e. to train a model that simultaneously predicts quantiles of all future intervals of interests (and interpolate if necessary). We call this the multi-span solution. The forecasted quantiles for each lead time and span can then be used to fit to a parametric form such as a Shifted Gamma or a Lognormal Distribution which in turn imply a full distribution for the future demand.

While this approach does provide accurate quantile forecasts, since there is no restriction on the overlapping output quantities, the quantile forecasts might not imply a statistically valid joint distribution. A simple example is that $E(D_{[1,2]}) \neq E(D_1) + E(D_2)$ is possible if the expectations are inferred by fitting a parametric distribution on each quantile. One might also find that $\rho_{(D_1,D_2)} > 1$ if the correlation is inferred from the marginal variance by the identity $\mathrm{Var}(D_{[1,2]}) = \mathrm{Var}(D_1) + \mathrm{Var}(D_2) + 2\rho\mathrm{Var}(D_1)\mathrm{Var}(D_2)$. As a result, any dynamic model for inventory management or planning such as [28] that relies on the joint or conditional distribution of future demand will fail. The simplest method to overcome these inconsistent forecasts is to ignore the information provided by the multi-span forecasts (i.e. the forecast for any interval wider than a single period) and assume independence between the future demands of each buying period. [10] studied the effects of the

**Both authors contributed equally

independence assumption and show that system cost and target inventory levels increase as demand autocorrelation increases - implying that an independence assumption drastically underestimates both the cost and the target inventory level.

In this work, we seek to generate consistent probabilistic forecasts with valid joint distributions of future demand, by adopting the structure of sample paths. Sample paths are simply multivariate samples of future demand curves drawn from either a parametric distribution or a nonparametric generative model. They provide a full joint empirical distribution, through which any statistics of interest including total demand quantiles can be computed directly. We propose a novel shrinkage estimator to repair the inconsistent forecasts generated by the multi-span solution of the quantile RNNs of [40] by approximating the nearest admissible (positive semi-definite) covariance matrix and then use a parametric distribution to generate sample paths from these covariance matrices. We also propose novel modifications on state of the art generative models such Generative Adversarial Networks (GAN) [18], Variational Autoencoders [24] and Bayesian Dropout [15], which provide non-parametric ways of generating sample paths. In this paper, we provide a full analysis of the production-ready shrinkage method and analyze the obstacles to providing a fully consistent forecast from the state of the art neural generative methods.

1.1 Related Literature

While there is a plethora of literature related to shrinkage estimators [37] [26] [27] - none have been applied to the problem of producing consistent demand forecasts. A related approach is the nearest correlation matrix method advocated by [21], which apart from being more computationally intensive suffers from the flaw that the projection of the correlation matrix doesn't impact the variances for each horizon. In other words, the projection does not modify the diagonal of the covariance matrix - an approach which reduces the shrinkage and improves the quantiles generated by the projection. This is remedied by shrinking in covariance rather than in correlation space. To our best knowledge, there is no literature on approximating a consistent covariance matrix in the manner proposed by us in Equation 2.

There has been a great deal of literature related to generating sample paths for time series. [14] propose the Seq2Seq model DeepAR, which directly outputs the parameters of a negative binomial, making it a parametric approach (a similar approach is also advocated by [31]). [38] also propose a likelihood based latent state model - while the generative adversarial network approach advocated by our work is does not rely on any likelihood assumptions. [41] develop a metric for learning GANs for point processes, while [12] propose a similar learning procedure for real valued times series. While both methods propose a learning procedure similar to ours, there are some crucial differences with our approach beyond our architectural choice of using a Deconvolutional Network, while they propose to use Recurrent Networks. Both methods seek to generate samples from the true distribution, while we seek to generate samples from the future distribution of demand. The same distinction also applies to the work done in generating sample paths from Variational Autoencoders such as [13].

Finally, there has been a lot of study into conditioning generative models in order to better control the data generating process. Structured output prediction for GANs was first addressed by [29], where the authors propose that the conditioning information is fed to the generator and discriminator to allow the generator to produce images conditional on the additional attributes. [7] put forward InfoGAN, an unsupervised conditioning of GANs. In this setting, the discriminator learns to predict a latent code fed to the generator, as well as to discriminate between true and generated samples. [32] advance a supervised conditioning technique for GANS: an auxiliary classifier GAN. The discriminator is used to classify real samples, and the generator is trained to belong to the correct class. However, the literature addresses the issue of structuring the output prediction by conditioning with respect to a discrete or low-dimensional continuous variable. Our prediction typically requires conditioning not only with respect to high dimensional discrete variables (such as product category) but also with respect to real values (such as past demand, distance to holidays etc.). Furthermore, the latest conditioning techniques require a discriminator capable of predicting parts of the conditioning variables, such methods have little hope to work with a high-dimensional conditioning manifold. This makes our problem unique, and our hope is that this paper not only proposes methodologies but also motivates further study into conditioning with respect to real values by convincing the reader that a conditional generative model offers a natural way to address the demand forecasting problem.

2 METHODOLOGIES

2.1 Shrinkage Methods

Covariance shrinkage methods aim to modify the invalid covariance matrix implied by the multi-span approach. We first formulate how an (inconsistent) covariance matrix can be computed based on distribution quantiles over a full set of lead-time/span intervals $[l, s]$, and then introduce our shrinkage solutions.

2.1.1 Implied Covariance. Consider a multi-horizon forecast bounded by $t < T$, then the multi-span approach will output marginal distribution quantiles of all possible $D_{[l,l+s]}$ where $[l, l+s] \subseteq [0, T]$ (some by interpolation, in practice). Note there are $T(T-1)/2 + T$ such intervals. Suppose a convenience parametric distribution is fitted to each of these marginal distributions, then all possible $\mathrm{Var}(D_{[l,l+s]})$ is available. Note there are $T(T-1)/2$ linear identities $\mathrm{Var}(D_{[l,l+s]}) = \sum_{t\in[l,l+s]} \mathrm{Var}(D_t) + 2\sum_{i<j} \mathrm{Cov}(D_i, D_j)$, while the number of unknown terms $\mathrm{Cov}(D_i, D_j)$ is also $T(T-1)/2$, which leads to a unique solution. As mentioned, such an inferred covariance matrix can be invalid: see Figure [1] for the spectrum of the covariance matrix for a single product's forecast, showing negative eigenvalues.

2.1.2 Covariance Shrinkage. For a single product, let $\hat{\Sigma}$ be the implied covariance matrix. Denote by C, the set of admissible covariance matrices for that product. Without loss of generality, we assume $\hat{\Sigma} \notin C$.

Using the fact that the set of all possible covariance matrices is a convex set (see [3]), we only need to consider a single point inside C - let us denote this point by Σ^* and define the shrinkage estimator Σ as:

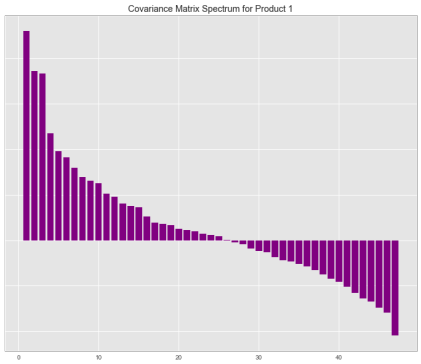

Figure 1: The spectrum for the implied covariance matrix of a single product shows negative eigenvalues, violating the positive semi-definiteness of true covariance matrices.

$$\Sigma = \lambda\Sigma^* + (1-\lambda)\hat{\Sigma} \tag{1}$$

where $\lambda \in [0, 1]$ can be interpreted as a degree of shrinkage or *shrinkage constant.*

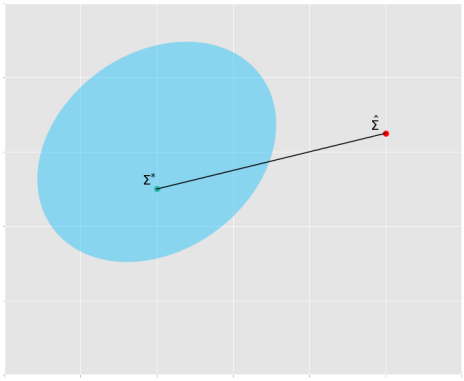

Figure 2: Since the set of covariance matrices is a convex set, we can take a linear combination of an admissible covariance matrix and our sample covariance matrix.

2.1.3 Identifying an admissible matrix. If we assume that $\forall i \in \{1, \ldots, T\} : \text{Var}(D_i) > 0$, then we can consider the diagonalized matrix $\Sigma_1^* = \text{diag}(Var(D_1), Var(D_2), \cdots, Var(D_{52}))$. Alternately, the identity matrix Σ_I^* can be used. Beyond a diagonal matrix, we can assume any auto-covariance structure, e.g. those implied by a traditional time series model like an ARMA. Typically, this implies that for each product an ARMA model is specified, estimated and use to compute the auto-correlations - making the method considerably slower than simpler assumptions that capture most of the uncertainty in the future demand. Instead, we propose to use a different admissible matrix, one that arises from imposing a constant correlation ρ between the different time steps. This correlation can be simply estimated as the average of the model implied correlations[1]

Let $\hat{\rho} := \frac{1}{N}\sum_{i=1}^{N}\sum_{j>=i}^{N}\rho_{i,j}$ where $\rho_{i,j}$ denotes the implied correlation between the demand at time i and the demand at time j. Denote by $V = \text{diag}(\Sigma)$, we consider the admissible covariance matrix $\Sigma^*_{\text{const}} = V^{-1/2}RV^{-1/2}$, where R is a correlation matrix with all off-diagonal elements as $\hat{\rho}$.

Empirically, we see that the average (taken across different lead times) standard deviation (approximated by the safety stock[2]) implied by the quantile networks grows slightly faster than linearly across spans for evergreen products - making the constant correlation assumption produce safety stock values that are considerably lower than those implied by the model for longer durations. This implies that any buying decision made for a long period will drastically underestimate the amount of safety stock required to meet demand. As a result we propose a novel shrinkage extension which modifies not only the off-diagonal elements of the covariance matrix but also its diagonal elements i.e. it uses variances other than those implied by the forecasts for each individual time period.

A first model is one which grows linearly from the implied variance for the first time period[3] such that the maximum span variance implied by the matrix is equal to the maximum span variance implied by the model.

Denote by V_1 the variance implied by the mesh for the first time period and by $V_{\max}$ the variance for the demand of the maximum span (duration) implied by the model, our variance function is determined by the following minimization:

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & \left(V_{\max} - \Big(\sum_i V(\beta, t_i) + 2\sum_{i,j}\sqrt{V(\beta,t_i)}R_{i,j}\sqrt{V(\beta,t_j)}\Big)\right)^2 \\ \text{subject to} \quad & V(\beta, t_i) = V_1 + \beta(t_i - t_1),\ i = 1, \ldots, T. \end{aligned} \tag{2}$$

where T denotes the maximum span period and R denotes a correlation matrix with all off-diagonal elements equal to $\hat{\rho}$. We denote the shrinkage estimator for this admissible matrix $\Sigma^*_{\max}$

2.1.4 Sample Path Generation. Here we propose a simple methodology to generate sample paths from the corrected mean[4] and covariance matrix. The method takes the μ and Σ generated by the shrinkage and treats it as the mean and covariance matrix of a multivariate lognormal distribution. The parameters are then mapped to their mean and covariance of the associated multivariate normal[5]. The multivariate normal samples are then exponentiated to obtain 52 multivariate lognormal samples. Each realization is treated as a sample path, as seen in Figure 3.

Alternatively, any distribution characterized by its first two moments[6] can be used to sample the 52 dimensional vector of the next year's demand. In our work, we found that the estimation of the

[1] When the model implies a |correlation| > 1, we clip the value to +1 or -1 depending on the sign.
[2] defined as the 90th Quantile - 50th Quantile
[3] Usually this is the variance implied for the first week's demand
[4] The mean can be corrected by setting $\mathbb{E}[D_1 + D_2] = \mathbb{E}[D_1] + \mathbb{E}[D_2]$
[5] See [20] for how this is done
[6] One could use higher moments with certain assumptions

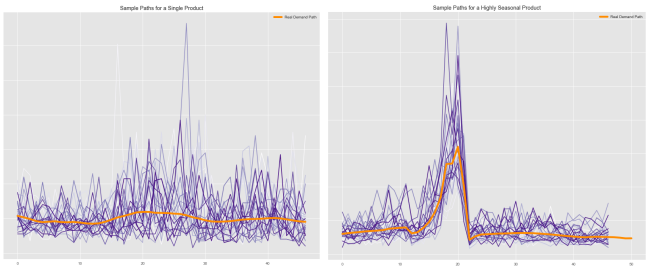

Figure 3: Lognormal paths generated for a single product (left) (whose spectrum is shown in Figure 1) and for a highly seasonal product (right) using the constant correlation shrinkage. The orange line shows the true demand.

quantiles is not very affected by the choice of the sampling distribution for evergreen products. For slow moving products however the distribution choice is much more significant.

2.2 Drawbacks of the Shrinkage Approach

The shrinkage approach, though effective, suffers from two major drawbacks. The first being that it seeks to approximate the nearest valid joint distribution to the mesh rather than directly forecasting a valid distribution. This means that more often than not, the quantile loss for its predictions are significantly worse than the direct forecasts from a quantile network such as MQRNN. In fact, the results in Section 8 indicate that while covariance shrinkage is a mature and consistent approach, it is only able to provide about $\sim$ 80% of the accuracy that a quantile network can provide over the independence assumption. As a result, we may consider that the quantiles forecasted by networks such as the MQRNN cannot be fully repaired, and improvements can be made to forecasts by generating predictive sample paths directly.

The second is that the generation of sample paths requires a parametric assumption. While the lognormal distribution is suitable for evergreen products with non-negligible demands, for slower products it becomes ineffective with most predictions requiring rounding and other ad-hoc methods for producing integer forecasts. As a result of these drawbacks, we investigate non-parametric methods to generate sample paths. Modern generative methods such as Bayesian Deep Learning [16], Generative Adversarial Networks [18], and Variational Autoencoders [24] provide promising approaches to solving this problem with few assumptions.

3 AUTOREGRESSIVE MODELS

Autoregressive models learn the joint probability of a distribution $d = \{d_1, \cdots, d_T\}$ by factorizing it as a product of conditionals : $p(d) = \prod_t p(d_t|d_1, \cdots, d_{t-1})$. In this setting, a deep model takes as input a fixed-size window of values in the past $d_{t-k}, \cdots, d_{t-1}$ and outputs the distribution's parameters for the next time point $p(d_t|d_{t-k}, \cdots, d_{t-1})$. This class of model has been shown to perform significantly better at generative tasks for complex distributions, such as the pixelRNN [34] for the distribution of natural images. It works remarkably well on raw audio files conditioned on latent information, as demonstrated by Wavenet [33]. Training

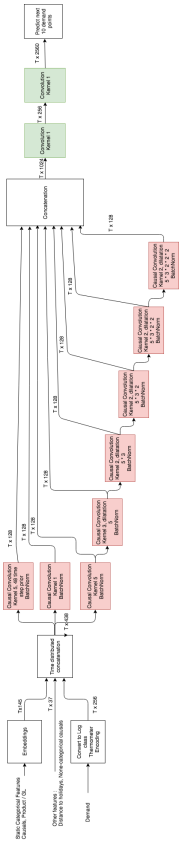

Figure 4: For our WaveNet architecture, we use 1D Convolutions on the past demand, with additional filter maps of the input containing the product features outlined in Section 7.

such a model is efficient, but generating samples is tedious as it is sequential by nature : feeding the past information the model output $p(d_1)$, we then sample $\tilde{d_1} \sim p(d_1)$ to feed it back to the model for generating the next time-steps. Recent work [35] has drastically improved the prediction time by generating all time-steps in parallel.

Our model (see Figure 4 for a visual depiction of the architecture) outputs a categorical distribution over the next demand value d_t, for each time point we have 256 classes each classes being an interval of demand between 0 and twice the maximum demand amount of the last year. This is achieved using a softmax and minimizing the negative log-likelihood loss. In order to leverage the ordering of the classes, when feeding classes to our network, we used thermometer encoding [6] instead of one-hot encoding. Thermometer encoding leverages class ordering by encoding the class k as a vector of 1s until index k, and 0 afterward. We did not find any improvement by leveraging this property with ordinal regression [8].

4 DROPOUT AS BAYESIAN APPROXIMATION

Since our problem deals with the generation of uncertainty in model outputs we attempt a principled approach for generating this uncertainty with neural networks through Bayesian inference with

Dropout [39] as a practical approximation technique. [15] propose to modify the traditional dropout procedure by applying the masks to each node not only during training but also during prediction, generating approximations of the model uncertainty. We modify this approach by using a neural network trained as a regression of the product's future median (though the mean could be used as well) for different lead times (with the duration being only a single time period or span = 1), and using dropout to inject uncertainty into the prediction - creating a sample path.

In order to understand the efficacy of the method, we trained the MQRNN proposed by [40] and added dropout to the LSTM layer as well as each of the decoder feedforward networks. For the LSTM Layer, we dropout linear connections with a probability of 0.4 and hidden connections with a probability of 0.6. We dropout nodes in the feedforward decoder layer with a probability of 0.4. These estimates were obtained through a grid-search.

5 VARIATIONAL AUTOENCODERS

Autoencoders [see [17] for a comprehensive overview] are trained to learn a lower dimensional representation of the data. In the case of the Variational Autoencoder (VAE) [24], the assumption is that this lower dimensional representation has a Gaussian prior. The loss used is a sum of the reconstruction error and the KL divergence of the reconstructed output from a Gaussian distribution. For our purposes, we use a traditional VAE, and find that a multivariate independent normal fails to capture enough uncertainty in the actual sample path. As a result, we propose an extension that incorporates correlations between the latent variables and extend the loss function to incorporate this. Denote by x the input of the demand time series of the previous year (possibly along with other features) and by $\tilde{x}$ the demand for each time period of the following year. Then, the classical VAE loss is as follows:

$$x \underset{\text{encoder}}{\rightarrow} (\mu, \log\sigma^2) \tag{3}$$

$$\text{Let } z = \mu + \sigma Z \underset{\text{decoder}}{\rightarrow} \tilde{x} \tag{4}$$

$$\mathcal{L}_{reconstruction} = \|x - \tilde{x}\|_{L1,L2,\dots} \tag{5}$$

$$\mathcal{L}_{KL} = \frac{1}{2}\sum(-1 + \mu^2 - \log\sigma^2 + \sigma^2) \tag{6}$$

$$\mathcal{L}_{VAE} = \mathcal{L}_{reconstruction} + \mathcal{L}_{KL} \tag{7}$$

where $Z \sim \mathcal{N}(0, 1)$. We extend this to incorporate correlations between the time points of our sample paths in the following way:

The first task is to modify the KL component of our loss function - the KL loss in the original VAE formulation is simplified due to the Gaussian posterior assumption. In the general case, the KL loss is the KL divergence between the posterior distribution given x and a Gaussian prior, namely:

$$q_\Phi(z|x) = \mathcal{N}(z; \mu(x), \sigma^2(x)) \tag{8}$$

$$p(z) = \mathcal{N}(0, 1) \tag{9}$$

$$\mathcal{L}_{KL} = KL(q_\Phi(z|x)||p(z)) \tag{10}$$

The first step towards having a VAE with correlated latent variables is to compute the KL divergence between two correlated multivariate Gaussians:

$$q(z) = \mathcal{N}(z; \mu_1, \Sigma_1) \tag{11}$$

$$p(z) = \mathcal{N}(\mu_2, \Sigma_2) \tag{12}$$

$$D_{KL}(q(z)||p(z)) = -\frac{1}{2}\Bigg[\log\frac{|\Sigma_2|}{|\Sigma_1|} - d + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)\Sigma_2^{-1}(\mu_2 - \mu_1)\Bigg] \tag{13}$$

Hence the KL loss becomes :

$$q_\Phi(z|x) = \mathcal{N}(z; \mu(x), \Sigma(x)) \tag{14}$$

$$\mathcal{L}_{KL} = \frac{1}{2}\Big[-\log|\Sigma(x)| - d + tr(\Sigma(x)) + \|\mu(x)\|_2^2\Big] \tag{15}$$

In order to learn a VAE with correlated latent variables, we need the determinant of the correlation matrix and require that the encoder output a symmetric positive definite matrix. We can have both by predicting a lower triangular matrix C with strictly positive diagonal terms, then model $S = CC^T$ (using the Cholesky decomposition) whose determinant is the product of diagonal terms of C squared. We use the L^2 loss on the difference of log-demand to generate our samples. Our Variational Autoencoder network architecture uses a feedforward network for both the encoder and decoder. Both hidden layer sizes were set to 512 and the latent size to 52, we use the Adam optimizer with a learning rate of 10^{-3}.

6 GENERATIVE ADVERSARIAL NETWORKS

All of our methods thus far have relied either on an explicit parametric assumption (such as the Shrinkage) or on an implicit one (such as the VAE/Dropout). However, we would like to model the dynamics of our multi-dimensional stochastic process without making any explicit assumptions about its form. Generative Adversarial Networks (See [18]) provide a powerful tool by formulating the learning problem as a mini-max game between two networks - a Generator, which produces synthetic samples and a Discriminator, which seeks to tell the synthetic samples apart from real ones. The optimal value for this loss function can provably be shown to be achieved when the generator samples come from the true data generating distribution.

We propose to use the Generative Adversarial Networks in the following way - the generator is trained to map gaussian noise to an N-dimensional vector of *future* demand, while the discriminator must classify between a generated path and a true future path during training. During prediction, the generator is used to produce predictive sample paths for each product.

While the original GAN proposes to use a Jensen-Shannon divergence, recently [1] proposed to use the Wasserstein distance as the primary objective for the minimax game. This creates many favorable behaviors including a more stabilized training procedure. We attempted both the classical GAN loss function, as well as the Wasserstein loss [see Figure 5 for a visual comparison of the JS loss training loss curve and the WGAN training loss]. We propose several different experiments to understand whether the problem of demand forecasting can be addressed by GANs and then propose architectures and methods for producing conditional forecasts for different products from a Conditional GAN architecture [29]. To

deal with the fact that our demand data has multiple scales we model $\log(1 + \text{demand}_t)$ rather than the demand directly.

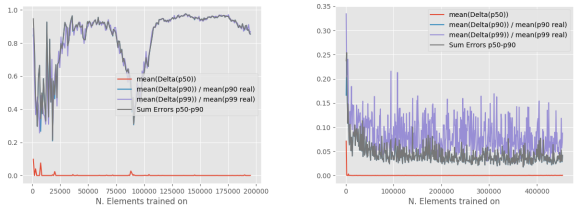

Figure 5: The training curve for the model using the WGAN-GP (right) shows relative stability when compared to the classical JS loss based GAN training (left)

6.1 Unconditional GANs

In order to understand the efficacy of a GAN in learning the quantiles of the future demand distribution, we begin by modifying the classical DCGAN architecture [36] to work for 1-dimensional inputs. Our inputs to the discriminator are the 52 week demands for a wide range of products, as well as the synthetic 52 dimensional samples produced by the generator.

6.1.1 Architectures. Our best performing architecture mimics the DCGAN architecture of [36] with 1 dimensional convolutional layers. Our generator is made of blocks of 1-dimensional transposed convolutions, 1-dimension batch normalizations [22] and Rectified Linear Unit (ReLU) activations [30]. The first block has a kernel of size 7, stride 1 and padding 0 and 256 output feature maps. It is followed by 3 blocks with kernels of size 4, stride 2 and padding 1, and respective feature maps 128, 64, and 1. The generator outputs a 56-dimensional vector, of which we keep the middle 52 values as our sample path. The Discriminator is symmetrical to our generator with convolutions and Leaky ReLU activations [42] of negative slope 0.2. The first block consists of a convolution of kernel size 4, stride 2 and padding 3, to match the 52-dimensional input. We use an Adam optimizer with learning rate $= 2 \times 10^{-4}$ and a exponential decay rate for the first moment estimates $(\beta_1) = 0.5$ and an exponential decay rate for the second-moment estimates $(\beta_2) = 0.999$.

6.2 Conditional GANs

In order to better control the data generating process, we propose to adapt the Conditional GAN variant developed by [29]. Since we would like to generate the demand of the future distribution, other conditional variants of GANs such as the Auxiliary GAN [32] which require the input to the generator to be both the features as well as the label are ruled out, as the label will not be available to us at prediction time. Our first architecture (which can be seen in Figure 6) extends the Deep Convolutionnal architecture [36] with one-dimensional convolutional layers, to address our conditional generation problem on times series. Our generator takes as input a noise vector of size 100 concatenated with the past demand and features of the time series. The discriminator has as input the generated future demand in one feature map, the past demand in another

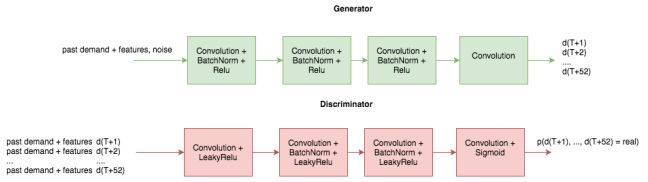

Figure 6: For our conditional DCGAN architecture, we use 1D Convolutions on the past demand, with additional filter maps of the input containing the product features outlined in Section 7.

and the time series features taking multiple feature maps, so that the sliding window of the convolution is applied to homogeneous input variables. We enforce the Lipschitz constraint on the discriminator by clamping the weights to 0.1, or using a gradient penalty as suggested in [19] with a gradient target norm of 0.1 and penalty term of 10.

7 DATA

For our analysis, we use the same dataset of 60,000 Amazon products as [40]. The data consists of weekly demand for around 60,000 sampled products from different categories within the US Market starting from the year 2012 to 2017. Demand data for 2016 is used to test the models by producing a single sample path for each of the 52 weeks in 2016 (for the unconditional networks) and for each of the weeks of 2016 (for the conditional networks). The covariates for our conditional models are the same used by [40] which are a range of suitably chosen and standard demand drivers in three categories: past demand, promotions and product catalog fields. We always forecast for a maximum duration of 52 weeks or 1 year from the week of forecast creation.

8 RESULTS

Since our covariance shrinkage approach can be used as a drop-in replacement for the distribution mesh of [40], we propose to compare the different shrinkage methods based entirely on their ability to approximate the mesh in terms of quantile loss. While this method of comparison is valid for approaches that seek to repair the mesh, they become less viable for generative models. This is because the efficacy of the sample paths produced by generative models, being non-parametric and directly forecasted, depend on the control algorithms utilizing them - specially the sensitivity of these algorithms to inter-temporal correlation and parametric assumptions. Furthermore, the difficulty in conditioning with respect to high dimensional discrete or continuous variables hampers the ability of generative models to produce well calibrated estimates of uncertainty. In this section, we analyze the results of each method both qualitatively and (where applicable) quantitatively - offering hypothesis for the model, its shortcomings and insights into its effectiveness.

8.1 Shrinkage Methods

We begin by analyzing the quantile loss for the different shrinkage methods. For two different products, we see from Figure 7 that an

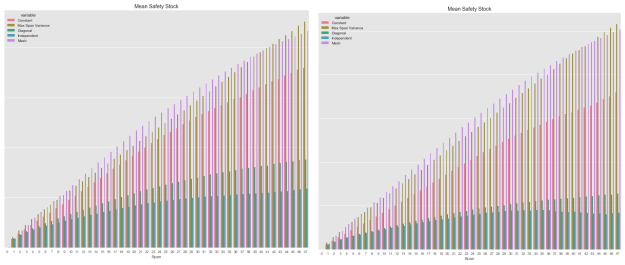

Figure 7: The safety stock estimates generated by assuming independence (green) grow as a square root, while the quantiles generated by MQRNN (purple) have most of their uncertainty captured by the constant correlation (red) and the max span (mustard) shrinkage methods.

Table 1: Quantile Loss Comparison for Different Shrinkage Methods relative % to MQRNN

Method	P90 Quantile Loss	P50 Quantile Loss	P10 Quantile Loss
MQRNN	100	100	100
Independence	122.25	101.25	176.63
Constant Correlation	104.09	100.04	119.86
Maximum Span Fit	108.48	102.84	111.80

independence assumption between the demand for different weeks produces safety stock estimates that grow like a square root as a function of the duration of the prediction (the span). Shrinking to the diagonal (Diagonal in the legend of the figure) provides some improvement while a simple constant correlation assumption captures a large part of the uncertainty that the independence assumption misses. The maximum span method proposed in Equation 2 however, is able to capture most of the uncertainty for the longer spans after shrinkage. This shows us that even a simplistic parametric method may be an effective fix for the problem.

In order to better understand these methods, a quantile loss comparison was done for each of the more effective shrinkage methods. For this analysis, we randomly sample 13,000 out of the 60,000 products in the dataset utilized by [40]. We compare the MQRNN forecast with the independence assumption as a baseline, the constant correlation shrinkage and the maximum span shrinkage. Table 1 shows that the constant correlation shrinkage is able to capture about $\sim 80\%$ of the accuracy provided by the MQRNN mesh over an independence assumption.

Figure 8 shows the P10, P50 and P90 Quantile Loss as a function of the duration (or span). As expected, the gap in QL between the mesh and the independent sample paths grows as a function of the span. The maximum span shrinkage is able to (by design) match the performance of the MQRNN forecast for longer spans, while the independent sample paths become progressively worse. The extremal quantiles (P10, P90) show a larger gap than the median since they tend to be more affected by estimates of the variance.

Our recommendation is to use the constant correlation shrinkage for most products, except those for which the target duration is large. In those cases, the maximum span shrinkage provides the best solution.

8.2 Dropout as Bayesian Approximation

Figure 9 shows the results of using dropout applied to MQRNN. In red is the mean demand, the MQRNN implied P90-P10 spread (the purple band) is clearly much larger than the P90-P10 spread achieved by dropout (the green band) - making this method ineffective. Our conjecture is that the method is primarily geared towards learning the endogenous uncertainty of a model, while our problem is concerned with the learning of exogenous uncertainty of demand. In fact, even for large dropout probabilities at each layer we were unable to produce well-calibrated estimates of uncertainty.

8.3 Variational Autoencoders

See Figures 10, 11 for a qualitative comparison of the paths generated by both the classical VAE and our correlated extension. We find that the VAE is able to learn the variance of the demand much more effectively than the classical VAE- however, neither seems to be able to capture the noise of the problem effectively. This might be due to the assumption that the latent variable follows a Gaussian distribution with only a single correlation parameter. A richer covariance structure or perhaps a distribution with fatter tails might be required to capture the distributional behavior of the future demand.

8.4 Conditional Generative Adversarial Networks

Since we found that the DCGAN with Wasserstein Loss and gradient clipping is able to learn the different quantiles of future demand extremely well, we proceed with our investigation of conditioning these networks to control the data generating process. Figure 12 show the paths generated by a DCGAN architecture with Wasserstein Loss and gradient penalty. The paths are able to produce enough noise (the blue band indicates the P10 and P90 of the paths for each time point) to be useful. However, the conditioning of the model remains a challenge. Periodically, the GAN falls into mode collapse and generates extremely similar paths for each product - as shown in Figure 13. We believe that while this method shows the most promise - the instability of training GAN's with conditioning variables that have high dimension or are continuous prevents it from being able to match the performance or quality of the sample paths produced by shrinkage.

8.5 WaveNet

We compare WaveNet to our other models and find that while it is able to capture rich temporal dynamics (see Figure [14]), it struggles during periods of high variance such as seasonal demand patterns. Furthermore, the autoregressive nature of the model means that errors tend to compound over longer periods leading to poorer long lead time forecasts.

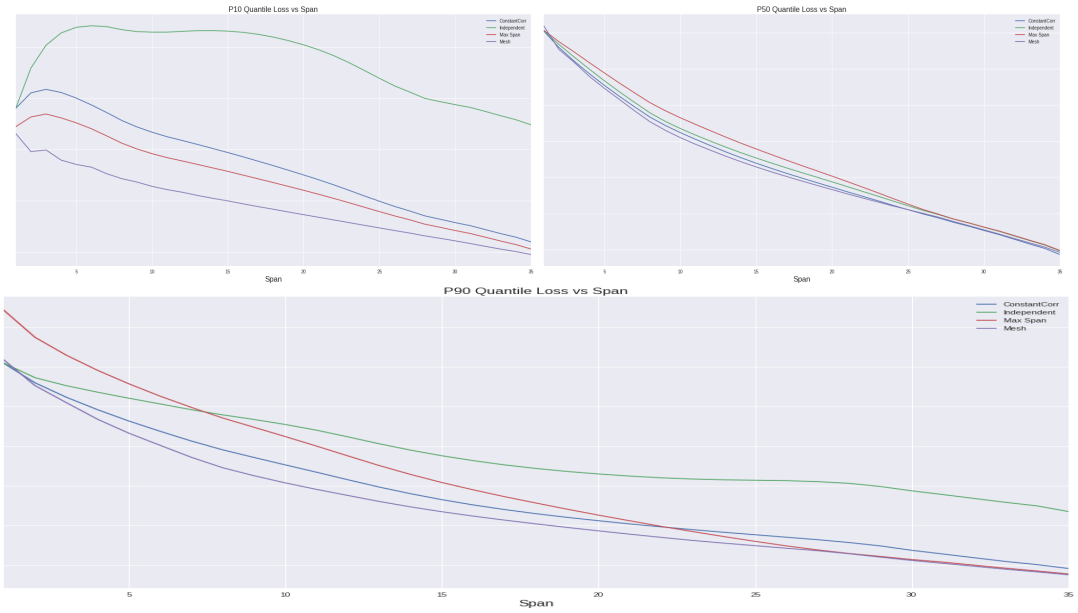

Figure 8: P10 (Top Left), P50 (Top Right), P90 (Bottom) Quantile Loss vs Span shows that the independence assumption (green) degrades as a function of Span while the constant correlation (blue) performs worse than the Maximum Span Variance Shrinkage (red) at longer spans. The QL of MQRNN as a function of span can be seen in Purple.

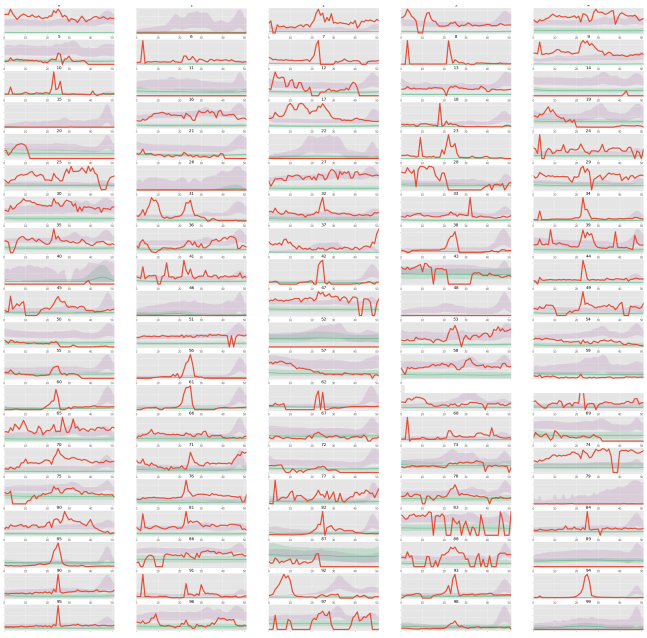

Figure 9: The quantiles implied by the sample paths generated from the dropout approach (in green) indicate that they fail to capture the uncertainty of demand when compared to forecasting the quantiles directly (in purple) by MQRNN.

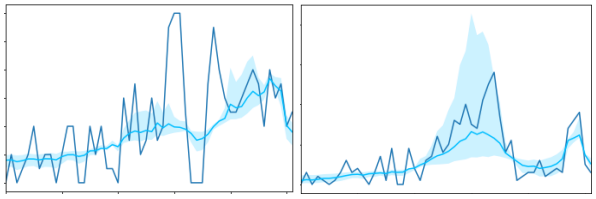

Figure 10: Demand for a single product (dark blue) along with the mean (light blue), 10th and 90th quantiles (blue band) of the generated sample paths from a classical VAE

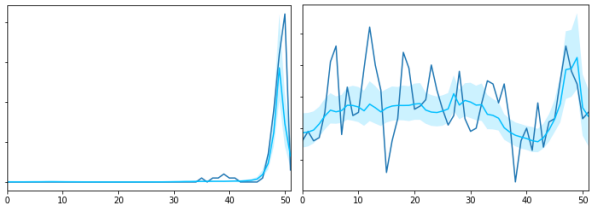

Figure 11: Demand for a single product (dark blue) along with the mean (light blue), 10th and 90th quantiles (blue band) of the generated sample paths from a correlated VAE

9 CONCLUSION AND FUTURE RESEARCH

We present a general framework for producing consistent forecasts from the accurate predictions of quantile networks such as MQRNN through sample paths and demonstrate a method in covariance shrinkage that can effectively produce these consistent

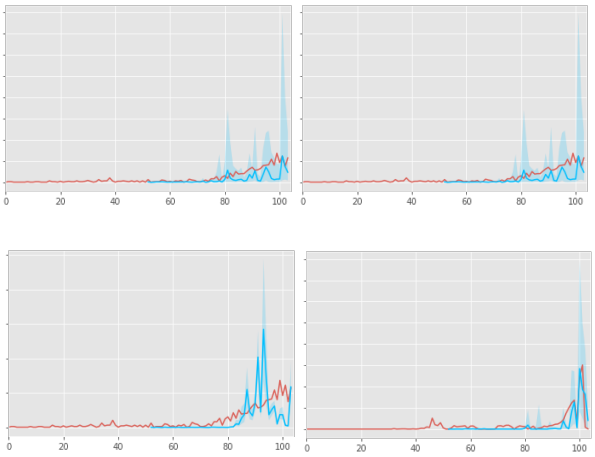

Figure 12: Demand for a single product (red) along with the mean (in blue), 10th and 90th quantiles (the blue band) of the generated sample paths from a Wasserstein DCGAN with Gradient Penalty

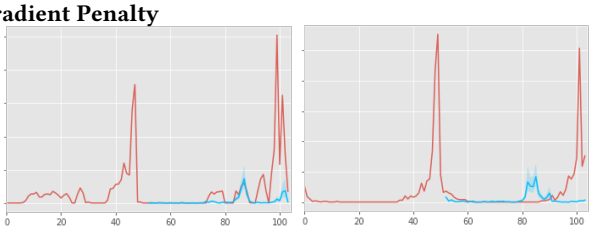

Figure 13: Demand for a single product (red) along with the mean (in blue), 10th and 90th quantiles (the blue band) of the generated sample paths from a Wasserstein DCGAN with Gradient Penalty shows that mode collapse causes the generator to produce similar paths for products with very different demand histories

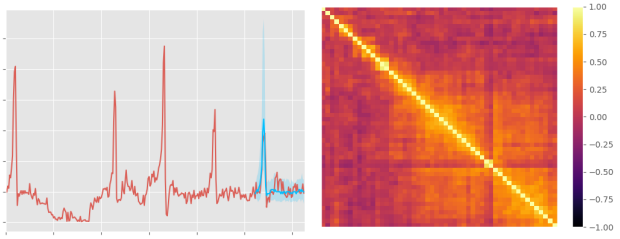

Figure 14: WaveNet forecasts and the associated autocorrelation matrix.

forecasts. However, we find that while shrinkage methods provide a consistent, scalable approach, they are only able to recover part of the accuracy that the quantile networks provide. This maybe due to the fact that the mesh forecasted by networks such as MQRNN cannot be repaired in a loss-less manner, indicating that improvements may be possible through direct prediction of the full joint distribution via sample paths.

As a result, we also propose frameworks for using neural methods to directly generate these predictive sample paths. We propose several novel modifications of classical and modern generative approaches and find that the difficulty of conditioning generative models with respect to continuous or high dimensional discrete variables and the inability of autoregressive models to accurately capture the uncertainty of distant periods or periods of high variance (such as highly seasonal demand) prevents these methods from achieving state of the art performance. Despite this, our findings can help in the utilization of quantile forecasts by inter-temporal decision making algorithms (for e.g. the methods put forward by [28]) as well as advanced planning systems such as those proposed by [9]. We hope that this problem inspires further research into generative modelling for time series prediction, allowing fully non-parametric sample path generation.

Table 2: Quantile Loss Comparison for Lead Time 0/3 Span 1 relative % to MQRNN

	Lead Time 0		Lead Time 3	
Method	P90 QL	P50 QL	P90 QL	P50 QL
MQRNN	100	100	100	100
Ind.	114.15	106.02	102.31	101.74
Constant ρ	113.97	106.07	102.15	101.79
Max Span	114.76	150.96	102.86	144.88
WaveNet	102.60	204.91	153.83	181.83

REFERENCES

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).

[2] Filippo Maria Bianchi, Enrico Maiorino, Michael C Kampffmeyer, Antonello Rizzi, and Robert Jenssen. 2017. An overview and comparative analysis of recurrent neural networks for short term load forecasting. *arXiv preprint arXiv:1705.04378* (2017).

[3] Grigoriy Blekherman, Pablo A Parrilo, and Rekha R Thomas. 2012. *Semidefinite optimization and convex algebraic geometry.* SIAM.

[4] Joos-Hendrik Böse, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Dustin Lange, David Salinas, Sebastian Schelter, Matthias Seeger, and Yuyang Wang. 2017. Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment* 10, 12 (2017), 1694–1705.

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

[6] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. 2018. Thermometer Encoding: One Hot Way To Resist Adversarial Examples. In *International Conference on Learning Representations.* https://openreview.net/forum?id=S18Su--CW

[7] Xi Chen, Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 29.* 2172–2180.

[8] Jianlin Cheng, Zheng Wang, and Gianluca Pollastri. 2008. A neural network approach to ordinal regression. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on.* IEEE, 1279–1284.

[9] Andrew J Clark and Herbert Scarf. 1960. Optimal policies for a multi-echelon inventory problem. *Management science* 6, 4 (1960), 475–490.

[10] Lingxiu Dong and Hau L Lee. 2003. Optimal policies and approximations for a serial multiechelon inventory system with time-correlated demand. *Operations Research* 51, 6 (2003), 969–980.

[11] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[12] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. 2017. Real-valued (medical) time series generation with recurrent conditional GANs. *arXiv preprint arXiv:1706.02633* (2017).

[13] Otto Fabius and Joost R van Amersfoort. 2014. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581* (2014).

[14] Valentin Flunkert, David Salinas, and Jan Gasthaus. 2017. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *arXiv preprint arXiv:1704.04110* (2017).

[15] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. 1050–1059.

[16] Zoubin Ghahramani. 2015. Probabilistic machine learning and artificial intelligence. *Nature* 521, 7553 (2015), 452.

[17] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30*. 5769–5779.

[20] Leigh J Halliwell. [n. d.]. The lognormal random multivariate. In *Casualty Actuarial Society E-Forum; Spring*.

[21] Nicholas J Higham. 2002. Computing the nearest correlation matrixÃćÂĂÂŤa problem from finance. *IMA journal of Numerical Analysis* 22, 3 (2002), 329–343.

[22] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. 448–456.

[23] Moutaz Khouja. 1999. The single-period (news-vendor) problem: literature review and suggestions for future research. *Omega* 27, 5 (1999), 537–553.

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

[25] Roger Koenker. 2005. *Quantile regression*. Number 38. Cambridge university press.

[26] Olivier Ledoit and Michael Wolf. 2004. Honey, I shrunk the sample covariance matrix. *The Journal of Portfolio Management* 30, 4 (2004), 110–119.

[27] Olivier Ledoit, Michael Wolf, et al. 2012. Nonlinear shrinkage estimation of large-dimensional covariance matrices. *The Annals of Statistics* 40, 2 (2012), 1024–1060.

[28] Alvaro Maggiar and Ali Sadighian. 2017. Joint Inventory and Revenue Management with Removal Decisions. (2017).

[29] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[30] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

[31] Nathan Ng, Rodney A Gabriel, Julian McAuley, Charles Elkan, and Zachary C Lipton. 2017. Predicting Surgery Duration with Neural Heteroscedastic Regression. *arXiv preprint arXiv:1702.05386* (2017).

[32] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2016. Conditional image synthesis with auxiliary classifier gans. *arXiv preprint arXiv:1610.09585* (2016).

[33] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[34] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759* (2016).

[35] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al. 2017. Parallel WaveNet: Fast High-Fidelity Speech Synthesis. *arXiv preprint arXiv:1711.10433* (2017).

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

[37] Peter J Rousseeuw and Geert Molenberghs. 1993. Transformation of non positive semidefinite correlation matrices. *Communications in Statistics–Theory and Methods* 22, 4 (1993), 965–984.

[38] Matthias W Seeger, David Salinas, and Valentin Flunkert. 2016. Bayesian intermittent demand forecasting for large inventories. In *Advances in Neural Information Processing Systems*. 4646–4654.

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[40] Ruofeng Wen, Kari Torkkola, and Balakrishnan Narayanaswamy. 2017. A Multi-Horizon Quantile Recurrent Forecaster. *arXiv preprint arXiv:1711.11053* (2017).

[41] Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Xiaokang Yang, Le Song, and Hongyuan Zha. 2017. Wasserstein Learning of Deep Generative Point Process Models. In *Advances in Neural Information Processing Systems*. 3250–3259.

[42] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).