Attention based Multi-Modal New Product Sales Time-series Forecasting

Vijay Ekambaram
IBM Research
vijaye12@in.ibm.com

Kushagra Manglik
IBM Research
kmangli1@in.ibm.com

Sumanta Mukherjee
IBM Research
sumanm03@in.ibm.com

Surya Shravan Kumar Sajja
IBM Research
suryasku@in.ibm.com

Satyam Dwivedi
IBM Research
satydw10@in.ibm.com

Vikas Raykar
IBM Research
viraykar@in.ibm.com

ABSTRACT

Trend driven retail industries such as fashion, launch substantial new products every season. In such a scenario, an accurate demand forecast for these newly launched products is vital for efficient downstream supply chain planning like assortment planning and stock allocation. While classical time-series forecasting algorithms can be used for existing products to forecast the sales, new products do not have any historical time-series data to base the forecast on. In this paper, we propose and empirically evaluate several novel attention-based multi-modal encoder-decoder models to forecast the sales for a new product purely based on product images, any available product attributes and also external factors like holidays, events, weather, and discount. We experimentally validate our approaches on a large fashion dataset and report the improvements in achieved accuracy and enhanced model interpretability as compared to existing k-nearest neighbor based baseline approaches.

KEYWORDS

New product sales forecast; Image based forecasting; Multi-modal embeddings; RNNs; Encoder-Decoder; Attention

ACM Reference Format:
Vijay Ekambaram, Kushagra Manglik, Sumanta Mukherjee, Surya Shravan Kumar Sajja, Satyam Dwivedi, and Vikas Raykar. 2020. Attention based Multi-Modal New Product Sales Time-series Forecasting. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394486.3403362

1 INTRODUCTION

Dead unsold inventory. One of the biggest problems confounding the fashion retail industry is the problem of dead unsold inventory. Despite having a lot of historical sales/inventory data for most fashion houses roughly 50-60% of products/stock keeping units (SKUs) sell well and rest go through severe markdowns [10]. Since fashion is heavily trend-driven and most retailers operate by season (for example, spring/summer, autumn/winter etc.), at the end of each season any unsold inventory is generally liquidated. While smaller retailers move the merchandise to second-hand shops large brands resort to recycling or destroying the merchandise so that the brand value is not diluted. For example, in 2018 H&M told it's shareholders [10] that it was sitting on $4.3 billion in unsold merchandise and was in news for burning defective products the retailer cannot sell to create energy [12]. The British fashion label Burberry recently destroyed more than £28m worth of its fashion and cosmetic products over the past year to guard against counterfeiting [8]. The environmental impact of the industry is even worse with the US EPA estimating 16 million tons of textile waste generated in 2017 in just the US, the majority of it going to landfill [1].

Mis-match between supply and demand. Behind many fashion brands is a highly complex supply chain. At a very high level unsold merchandise/inventory is mainly due to mis-match between supply and demand. It could be that the inventory has been over-produced or the inventory has not been distributed properly at the right location and at the right time, *mainly due to inaccurate demand/sales forecasts*[1]. The starting point for any supply chain planning is to have an accurate demand forecast for a product the retailer is planning to introduce this season. Once we have a good demand forecast rest of the supply chain planning (including assortment planning and stock allocation) fall into place.

New products without explicit historical sales data. At the heart of the problem we need to be able to accurately forecast the demand for a new product the retailer is planning to introduce for this season. Unlike other retail industries fashion is heavily trend driven, every season a substantial amount of new products are introduced and there is no reference historical sales data for this particular product to forecast the demand. For example, consider a retailer who wants to forecast the demand for say Pepsi for the the next month. Typically the retailer has access to the historical sales of Pepsi for the past few years. Standard time-series forecasting methods can be applied to forecast the sales (demand) based on the historical time-series data.

However, consider a fashion designer or merchandiser who is introducing a new *botanical print sleeveless top* and wants to forecast the demand for this product for the season to better manage the supply chain planning. Since this is a completely new product there

[1]Sales is the actual units of a product that was bought while demand is the latent demand (in terms of number of units) for a particular product. In general sales is not equal to demand due to demand transference (cannibalization,substitution etc). In this paper we use the sales as a surrogate for the true demand and use the terms interchangeably.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403362

is no explicit historical sales time-series data to base the forecast on. However, note that there is historical sales data but for a completely different set of products. Merchandisers currently resort to a slightly gut based approach to forecast the demand for new products. Based on the historical sales data the retailer could forecast the demand for *sleeveless tops*. The design team thinks that botanical prints are trending right now. Based on this he/she would manually adjust (increase) the demand forecast for this particular botanical print sleeveless top. Also fashion is highly visual and very often it may not be possible to describe all the attributes of a product. In such cases the merchandiser would identify visually similar tops in the last season and base the forecast on a particular reference product. The sales of a product not only depends on the product attributes but also on several merchandising related attribute such as price, discount, promotions as well as external factors such as weather, weekends, holidays and events.

In this paper, we propose and investigate the performance of several neural architectures that aim to forecast the sales time-series for new products based on product images, any available product and merchandising attributes and external factors. This enables the designer or merchandiser to upload a product image and forecast the sales over the span of several weeks. For a merchandiser this helps in better supply chain planning and for a designer this would help in making design choices that can potentially sell well. We focus mainly on fashion retail, since fashion is heavily trend driven and churns out new products at a rapid pace. However, the same tools could be used in any retail industry where new products are introduced and there is a need to forecast the demand based on historical sales data of other products.

1.1 Problem formulation

We assume that each product is implicitly represented by its product image $\mathbf{I_p}$, which would essentially be the image embedding from a suitable convolutional neural network [13]. Along with the image we may also have structured attributes, represented as a vector of d attributes ($\mathbf{x_p} \in \mathbb{R}^d$). For example these could be design attributes such as color, pattern, sleeve style etc. or merchandising attributes such as list price, promotion etc. In case such structured attributes are not available it could as well be neural network based embedding for the textual descriptions. Associated with each product we have a historical time-series sales/demand data $\mathbf{y}(t)$ for $t = 1, ..., T$ at an appropriate time scale (days,weeks etc). Given a set of n products (images and attributes) and their historical sales time-series,

$$\gamma = [(\mathbf{I}_1, \mathbf{x}_1, \mathbf{y}_1(t)), (\mathbf{I}_2, \mathbf{x}_2, \mathbf{y}_2(t)), ..., (\mathbf{I}_n, \mathbf{x}_n, \mathbf{y}_n(t))], \quad (1)$$

the task is to learn a time-series forecasting model f to predict the sales $\mathbf{y}(t)$ for a new product based on the image $\mathbf{I}$ and attributes $\mathbf{x}$,

$$\mathbf{y}(t) = f(\mathbf{I}, \mathbf{x}, t|\gamma). \quad (2)$$

The sales is also influenced by exogenous regressors such as weekends, holidays, events, markdown, promotions etc. We would like our model to account for these too. Note that unlike the product attributes the exogenous regressors are another time-series. Let $\mathbf{z}(t) = [z_1(t), ..., z_k(t)]$ be k exogenous regressors. The task is now to forecast the demand based on the intrinsic product image ($\mathbf{I}$) and attributes ($\mathbf{x}$) and k exogenous regressors $\mathbf{z}(t)$.

$$\mathbf{y}(t) = f(\mathbf{I}, \mathbf{x}, \mathbf{z}(t), t|\gamma) \quad (3)$$

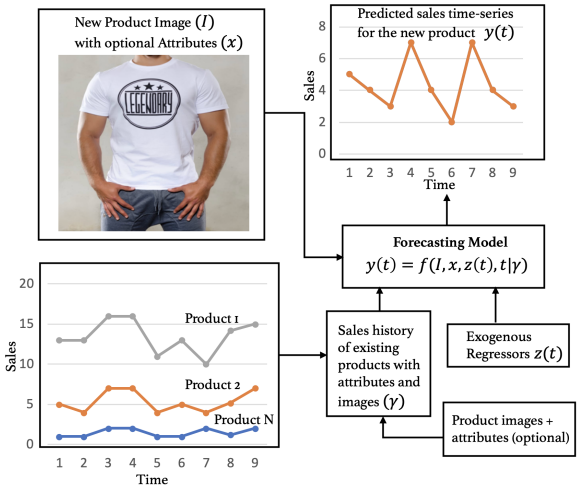

Figure 1: Image based New Product forecasting

Classical time-series forecasting methods (such as ARIMA, Holt-Winter, Theta method, fbprohet, Gaussain Process Regression etc.) are univariate methods and can only forecast based on the historical values for one time-series. In contrast we are interested in *analyzing multiple time-series together to learn a model to forecast for a new product*. In this paper, we propose and investigate the performance of several neural architectures to learn the forecasting model f. we depart from the traditional image similarity or attribute based k-nearest neighbor (KNN) approaches for new product time-series sales forecasting and propose several neural architectures.

1.2 Novel contributions

The following are the novel contributions of this paper.

- Classical time-series forecasting methods are uni-variate methods and can only forecast based on the historical values for one time-series. In contrast, we analyze multiple time-series together to learn a model to forecast the sales for a new product. All the time-series are modelled together based on the product images and optional attributes, while explicitly incorporating exogenous regressors like discount, holidays, events etc.
- We empirically study the effect of various network choices in modelling *image-based* new product time-series forecasting as an **Encoder-Decoder** sequence problem and also compare and contrast these approaches with several baselines. While sequence models have been used in the past in the context of image captioning [19], to the best of our knowledge this is the first attempt to use these models in the context of *image-based* new product time-series forecasting.
- One of the important requirements in new product forecasting is to opportunistically improve the forecasting accuracy by adding new data-sources, while guaranteeing the current performance of the model. To address this, we empirically study and evaluate the impact of various neural-network based multi-modal fusion techniques in new product time-series forecasting, which enables effective gradient flow towards data-sources leading to more information gain.
- To overcome the black box nature of predictions in deep learning models and enable improved explainability and accuracy in

new product time-series forecasting, we discuss the impact of various self-attention and cross-attention mechanisms in image and multi-modal scenarios.
- We conduct experiments on a large-scale fashion data set (comprising of 10,290 products distributed across 45 categories) and report results and interesting findings to illustrate the benefits of modelling image based new product time-series forecast as an encoder-decoder sequence problem as opposed to the conventional k-nearest neighbor approaches.

The rest of the paper is organized as follows. In the next section, we briefly discuss research work related to the current problem. Section 3 explains the encoder and recurrent neural network architectures applied to the problem. Section 4 describes the data used for experiments, and a comparative study of the results obtained by applying different models.

2 RELATED WORK

Fast fashion retail has lead to an increased demand for the new product forecasting [3]. Traditional forecasting methods [9] can not be applied directly for new product forecasting problems, primarily due to lack of data [7]. Though historical sales data for new product does not exist, if similar items have been sold in past season, their sales can be used as a proxy for the forecast. Most research in the area of new product forecasting is based on clustering old products based on their categorical product attributes and sales performance and then classifying the new product to forecast its sales [14]. Hence, the forecasting systems used for new products, primarily differ in how the similar products (or proxies) are selected. Selection of appropriate clustering and classification methods is usually based on the nature of data and its size. For example, authors in [15] use k-means to cluster existing similar products based on their historical sales curves and all the products in a cluster share a forecasting model. They use a decision tree to classify their new products into these clusters based on the new product features. In [16], the authors use a Self-Organizing Map (SOM) to cluster past products based on their sales time-series and use a probabilistic neural network to link the clusters with product features. Authors in [2] propose a cluster-while-regress approach where clusters are formed based on a similarity in terms of both product features as well as sales behavior of past products and forecasting models for these clusters are built simultaneously. However, these clustering approaches do not utilize product images hence, ignore all the unattributed visual aspects of a new product.

Contrary to the approach of finding similar products to do new product forecasting, authors in [11] claim lack of similarity in sales curves for the products which have similar attributes. Hence, they propose models which captures the correlation between temporal features and product attributes together. However, the authors in [11] do not address the effect of product images in new product forecasting.

Image based new product forecasting has been addressed by authors in [5] to propose an image-embedding based KNN approach to forecast the sales trend of new products. In addition, they also fine-tune the image embeddings to capture the sales trend by training a Siamese Network across product pairs which reduces the embedding distance between 2 products based on their sales similarity. However, these KNN approaches suffer from scalability issues, as it demands the storage of all product embeddings and their corresponding sales trends, which are queried at the time of estimation. In addition, this approach is not suitable for large fashion datasets (say 10K products) because training a Siamese network with all product pairs explodes the training set size. Apart from the scalability issue, KNN approaches lack the ability to model complex non-linear relations across image, sales and exogenous features. They also lack the capability to merge distinct auto-regressive signals from time-series sequences. In this paper, we address these challenges by modelling the image based new product forecasting as an encoder-decoder sequence problem and also empirically analyze the impact of various architectural choices. While Encoder-decoder sequence models have been used in the past in the context of image captioning [19], to the best of our knowledge this is the first attempt to use these models in the context of image-based sales time-series forecasting of new products.

3 METHODOLOGY

The main goal is to forecast the sales time-series of a new product given its image using the sales history of other products. To address this problem, Section 3.1 discusses various baseline K-Nearest Neighbour (KNN) approaches which are commonly adopted in industry and Section 3.2 discusses different Encoder-Decoder Sequence models which could improve the forecasting accuracy and explainability as compared to the baseline approaches.

3.1 Baseline KNN approaches

Traditional techniques leverage product similarity methods to fetch similar products from the history and aggregate its sales (considering the temporal alignment) to predict the sales time-series of the new product. In this context, the following 2 approaches are described to set the baseline for later proposed approaches.

3.1.1 ***Attribute KNN.*** This approach assumes products with similar attributes exhibit similar sales trends. Using product attribute similarity metric it fetches k-nearest neighbors from the historical data and the central tendency of k-nearest neighbors is used to report the new product forecast. This simple heuristic often gives decent ball-park estimates, if the attributes are selected judiciously.

Let $X_p, \mathcal{Y}_p$ represent the attribute set and sales time-series associated with the product p. $\theta(X_{p_i}, X_{p_j})$ is the distance metric between two given products p_i and p_j. Let $\mathbf{P}$ represent set of all products. $\mathcal{N}_k(X_p|\mathbf{P}, \theta)$ reports the k nearest neighbors for the product p from the set $\mathbf{P}$, and distance metric θ. The estimator $\mathcal{E}_a$ produces a sales time-series estimate by performing weighted aggregation on time-series of k nearest neighbors.

$$\mathcal{E}_a(p|\mathbf{P}, \theta) = \sum_{n \in \mathcal{N}_k(X_p|\mathbf{P},\theta)} \frac{\theta(X_n, X_p)}{\sum_{i \in \{n\}} \theta(X_i, X_p)} \mathcal{Y}_n$$

3.1.2 ***Embedding KNN.*** Detailed product information such as product description, image, cannot be consumed in the attribute-based KNN framework directly. The popular approach to address this situation is to represent the product features as a vector embedding. The vector embedding defines a proper distance metric. It is expected that similar products are close to each other compared to dissimilar products in this embedding space.

Let, $\mathcal{X}_p$, and $\mathcal{Y}_p$ represents the image/unstructured data and sales time-series attributed to product p. Φ is the differrentiable map for vector embedding, $\Phi(\mathcal{X}_p)$ represents the vector embedding of the product p. $d(\Phi(\mathcal{X}_{p_i}), \Phi(\mathcal{X}_{p_j}))$ is the distance metric between product p_i, and p_j in the embedding space. $\mathcal{N}_k(\mathcal{X}_p|\mathbf{P}, \Phi, d)$ reports the nearest k neighbors of $\mathcal{X}_p$ chosen from the set $\mathbf{P}$ having embedding map Φ, and distance metric d. Now similar to attribute-based KNN, a KNN estimator $\mathcal{E}_I$ is defined as

$$\mathcal{E}_I(p|\mathbf{P}, \Phi, d) = \sum_{n \in \mathcal{N}_k(\mathcal{X}_p|\mathbf{P}, \Phi, d)} \frac{d(\Phi(\mathcal{X}_n), \Phi(\mathcal{X}_p))}{\sum_{i \in \{n\}} d(\Phi(\mathcal{X}_i), \Phi(\mathcal{X}_p))} \mathcal{Y}_n$$

3.2 Encoder-Decoder based Sequence models

Issues with KNN approaches are threefold: (i) scalability as it demands the storage of all product embeddings and their corresponding sales time-series, which are queried at the time of estimation, (ii) inability to model complex non-linear relations across image, sales and exogenous features, (iii) inability to merge distinct autoregressive signals from time-series sequences.

Encoder-Decoder Sequence Modelling can be used to address the above challenges. However, various deep-net architectures are possible in Encoder-Decoder Sequence Models and we observe that in practice, these architectures play a crucial role in impacting the forecasting accuracy. Thus we explore different encoder-decoder sequence modeling architectures for image-based new product forecasting in this section. We start with standard models, then transition to attention-based approaches and finally the explainable multi-modal approach. These approaches can be grouped into 3 main classes in the order of their increasing structural complexity and interpretability.

- Sequence learning with encoded image input **(Image RNN)**
- Sequence learning with encoded multi-modal inputs. **(Multi-modal RNN)**
- Explainable sequence learning with attended multi-modal inputs. **(Cross-Attention RNN)**

3.2.1 ***Image RNN.*** As depicted in Figure 2, Image RNN framework comprises of an Encoder and a Decoder module. Encoder module captures a compact embedding for the given input image and merges it with the temporal features (i.e. past sales and exogenous features associated at time-step t_i) to feed as an input embedding to the RNN Decoder.

On the other hand, RNN Decoder receives the input embedding from the encoder module and predicts the sale value at time-step t_i by capturing the non-linear temporal relations across image embedding and temporal features. Besides predicting the sales value at time-step t_i, RNN Decoder also outputs a hidden state which is used by subsequent RNNs to capture the past sequence context. Thus, the RNN Decoder could recursively predict the product's sales value at every time-step t_i based on the input product image, temporal features associated at t_i and past hidden RNN state. In specific, the following temporal features are considered for the i^{th} sales prediction, viz. Sales at t_{i-1}, t_{i-2} and Exogenous features at t_i, t_{i-1}. The RNN model thus trained on all historical product images and associated sales & exogenous time-series can be used to predict the sales time-series forecast of a new product, given an image and exogenous time-series associated with the prediction window. Since we are focusing on new product forecast with no past data, we prepend 2 start delimiters to sales and exogenous time-series of all products to bootstrap the model training for the first two time-steps (where there is no past data). For later time-steps, Teacher Forcing technique [18] has been applied to the training process. Thus, Image RNN acts as a unified single model built to forecast for all products, making it easy to manage and deploy. In contrast, classical time-series methods has to build and maintain models specific to each product which leads to scalability & deployment issues.

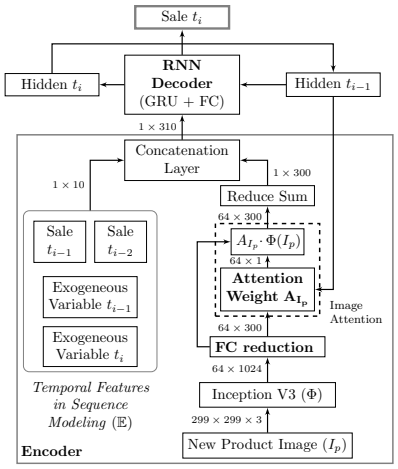

Figure 2: Image RNN encodes product images and temporal features as input to forecast sales time-series using RNN architecture. The study compares two models with and without Image attention layer (dotted box). Trainable modules (in bold) are jointly learnt in the training process.

In context of the above described RNN decoder architecture, the following two Encoder variants are studied.

(1) **Standard-Image RNN**: In this architecture, the flattened image embedding derived using InceptionNet-V3 architecture [13] is directly concatenated with the temporal features to produce a joint embedding for the decoder framework.
(2) **Attended-Image RNN**: It is important to note that, not all features in the image embedding will be relevant or useful for the forecasting task. In practice, we observe that these irrelevant image features significantly affect the forecasting accuracy by adding noise. To address this, the reshaped output of the last convolution layer from Inception-V3 (64×1024) is passed through an intermediate attention layer (Bahdanau attention [4]) to learn the relative importance of various features in the image embedding based on the past RNN hidden state. This attended image embedding is further concatenated with the temporal features to produce a joint embedding for the decoder framework. From results depicted in Section.4, we observe that jointly learning the attention layer while training the RNN Decoder has a significant impact in improving the forecasting accuracy as compared to the Standard Image RNN approach.

The only difference between the above two encoder variants is the presence/absence of the image attention layer, represented as dotted box in Figure 2.

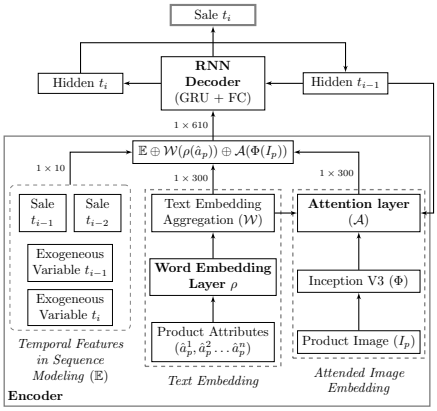

Figure 3: Concat Multi-modal *RNN* which learns joint embedding derived by concatenating embeddings of individual input modality. Trainable modules (in bold) are jointly learnt in the training process.

3.2.2 ***Multi-modal RNN.*** In this section, we describe Multi-modal RNN (Figures 3, 4), which can be employed for improved forecasting, when other modalities describing the product (Ex. text description, attributes) are available in addition to the product image. Encoder-decoder sequence models allow seamless integration of these multi-modal inputs into a unified framework. Decoder component of the Multi-modal RNN model is same as the decoder of the Image-RNN model. However the encoder component is modified to enable multi-modal fusion of various data sources. Let's assume $\{X_p^k\}_{k=1\ldots M}$, represents various input features (M distinct input modes) associated with the product p and Φ_k be the differentiable map that represents k^{th} feature as an embedding. In multi-modal scenarios ($M > 1$), let Θ be the operation that combines these distinct maps, $\{\Phi^1, \Phi^2, \ldots \Phi^M\}$, into a joint feature embedding $\Theta(\Phi^1(X_p^1), \Phi^2(X_p^2), \ldots \Phi^M(X_p^M)) \in \mathbb{R}^D$.

In the multi-modal context, we have considered product image ($\mathcal{I}$), and product description (X) as different input sources. The product description includes five attributes (color, pattern, fit, fabric, product category). Each attribute ($\hat{a}_p$) is first converted into its corresponding word embedding using a trainable Embedding layer(ρ). These attribute embeddings($\rho(\hat{a}_p)$) are further passed to an aggregation function($\mathcal{W}$) which outputs a textual embedding for the product. Various aggregation techniques such as average, weighted average or TF-IDF could be considered for this purpose. For representing the image, we follow the same attention based image embedding technique used in Section. 3.2.1 with the following modification. Image Attention layer in Multi-modal RNN leverage both text features and past RNN hidden states to determine the attention weights. This is an important design choice which enables improved fine-tuning of the image features based on the textual data eventually resulting in more coherent multi-modal embedding across image and text.

Let Γ represent the differentiable map approximated by the RNN sequence learner. Learning Γ separately from Φ does not guarantee best model performance. Joint learning of Γ and Φ in DNN context, leads to various architectural choices. Different architecture allows different gradient definition, thereby affecting model convergence. We have extensively studied the following two architectures in this work:

(1) **Concat Multi-modal RNN**: Each embedding vector is concatenated to derive the joint embedding.

$$\Theta(\{\Phi^i(X_p^i)\}_{i=1\ldots M}) = \Phi^1(X_p^1) \oplus \Phi^2(X_p^2) \oplus \cdots \oplus \Phi^M(X_p^M)$$

In multi-modal scenarios, embeddings associated with each modality are of varying sizes, hence concatenation is often a popular choice of deriving joint embeddings (Figure 3). But in the scenarios, where the dimension of the embeddings so obtained, is of significantly higher dimension compared to the rest of the network, the model suffers from convergence issues.

(2) **Residual Multi-modal RNN**: Leverages Residual block to combine the embedding space in an additive fashion[6].

$$\Theta(\{\Phi^i(X_p^i)\}_{i=1\ldots M}) = \Phi^1(X_p^1) + \Phi^2(X_p^2) + \cdots + \Phi^M(X_p^M)$$

Residual blocks are specific neural network architecture, that provide an alternate option for gradient to flow through the dense deeper block or skip it, depending on information gain during the training phase (Figure 4). Such a network guarantees that the performance of the combined block is better than either of the blocks operating in silos. The two multi-modal inputs (product image, product description), can be used in three different ways i.e. only image embedding, only text embedding, or joint image and text embedding. In the proposed construction, residual block combines these three embedding constructions for improved gradient flow.

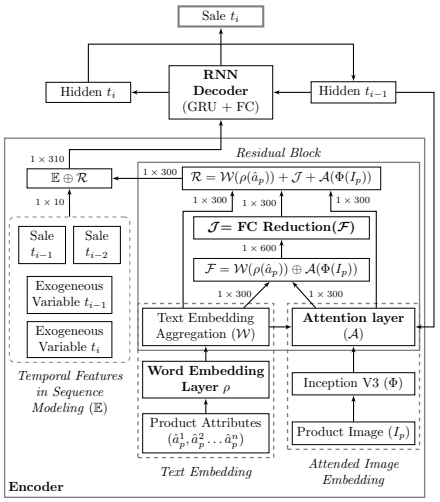

Figure 4: Residual Multi-modal *RNN* - leverages residual connections to guarantee improved model performance over any single input modality. Trainable modules (in bold) are jointly learnt in the training process.

3.2.3 ***Explainable Cross Attention RNN.*** Explainable machine learning has recently emerged as a new focus in deep-learning research to overcome the black-box nature of deep learning predictions. With this motivation, in Cross Attention RNN - we have introduced two distinct attention mechanisms in the encoder component viz. (i) Cross Attention across multi-modal inputs ($\mathbb{A}_m$), and (ii) Self Attention across temporal features ($\mathbb{A}_t$). Figure 5 explains the encoder-decoder architecture of this model.

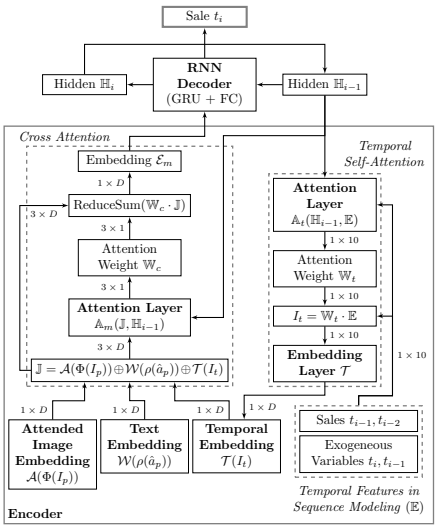

Figure 5: Cross-Attention *RNN* learning multi-modal attention weights and temporal attention weights for explainable multi-modal time-series forecasting for new products. Trainable modules (in bold) are jointly learnt in the training process.

Different temporal factors (Ex. holidays, discount, past sales, sale price, etc.) could dominate the sales decision at different time points. To capture this, Self-attention technique [17] is applied on the temporal features to determine its relative importance at every time-step prediction. Self-attention mechanism enables the temporal features to interact with each other to determine who they should pay more attention to. In specific, the temporal self-attention layer proposed in our architecture uses i^{th} temporal features and $(i-1)^{th}$ hidden state of RNN ($\mathbb{H}_{i-1}$) to derive the temporal attention weights ($\mathbb{W}_t$) for the i^{th} prediction. Attention layer then applies a dot product between temporal weights and features ($\mathbb{W}_t \cdot \mathbb{E}$) which is further passed through a trainable embedding layer ($\mathcal{T}$) to output a D-dimension temporal embedding. Leveraging this attended temporal embedding as compared to the original temporal features improves the forecasting accuracy, as attention overcomes the noise in the training introduced by irrelevant features.

In order to capture both product and temporal context, the temporal embedding obtained above is further fused with the text and attended image embedding to derive a joint multi-modal embedding. We apply the same process as followed in Multi-modal RNN Model (Section. 3.2.2) to construct the text ($\mathcal{W}(\rho(\hat{a}_p))$) and attended image embedding ($\mathcal{A}(\Phi(I_p))$). However, to enable the multi-modal fusion in a more explainable way, we propose the following cross-attention technique.

Cross-Attention technique derives relative importance across the Multi-modal inputs ($\mathcal{A}(\Phi(I_p))$, $\mathcal{W}(\rho(\hat{a}_p))$, $\mathcal{T}(I_t)$) based on the past hidden RNN state and current multi-modal inputs. Multi-modal inputs are fed as inputs to Cross-Attention Layer ($\mathbb{A}_m$) to derive a cross-attention weight vector ($\mathbb{W}_c$) of size $M \times 1$, where M is number of modalities considered. It must be noted that, $\mathbb{W}_c$ is temporal context dependent and changes at every time-step prediction. These Attention weights partition the gradient flow from each mode to the sequence model, thereby representing their relative contribution to the final prediction. The attention layer then performs a context dependent weighted sum. To perform the same, all input embeddings are brought to same dimension (D) and stacked together to perform a dot product with $\mathbb{W}_c$ (i.e, ${\mathbb{W}_c}^T \cdot [\mathcal{A}(\Phi(I_p)), \mathcal{W}(\rho(\hat{a}_p)), \mathcal{T}(I_t)]$). This dot product output(of shape 3xD) is then passed to a Reduce Sum block to derive a 1xD multi-modal embedding which is fed as input to the RNN Decoder. Thus, cross-attention technique enables an improved multi-modal fusion and forecasting by relatively focusing only on the important modalities.

As supplementary benefits, temporal attention and cross attention layers also output the attention weights ($\mathbb{W}_t, \mathbb{W}_c$) that explain the dominance of each temporal feature and multi-modal input at every time-step prediction respectively. This information highly improves the local explainability of the model predictions.

4 EXPERIMENTS AND RESULTS

In this section, we empirically validate various new product time-series forecasting approaches that have been described in Section 3. For this purpose, we have considered 2 years of historical time-series dataset of a reputed fashion house (source anonymized) comprising of 10,290 products distributed across 45 categories. Each product has following metadata associated with it: (i) product attributes (color, design, fit, fabric, category), (ii) product image and (iii) product name. The dataset also comprises of product time-series data aggregated at week level, where each time-point represents a week number with the following metadata: (i) total sales of a product in the week, (ii) discount offered in the week, (iii) sale price of the product, (iv) number of holidays in the week. This dataset has further been split into training and test set following 80-20 rule. Discount, sale price and holiday acts as exogenous features that would capture various patterns related to consumer behavior and market sentiment.

Since we are dealing with time-series forecasting we have considered Mean Squared Error (MSE) as the primary loss function to train all our models. We also observed a varying scale across all the temporal features. Hence min-max normalization has been applied to every feature independently to bring them to a common 0-1 scale. In addition, for every studied model we also carried out hyper-parameter tuning wrt. batch size, epoch, learning rate and model size, in order to arrive at the right set of training parameters. Dropout layers have been used to ensure that there is no over-fitting, while at the same time correlation between training and validation loss has been carefully monitored at every epoch to determine the most optimal epoch at which the training needs to be terminated.

4.1 Evaluation Metric

We have preferred to use weighted mean absolute percentage error (wMAPE) as the primary evaluation metric as compared to MAPE, since MAPE inherently suffers from the zero denomination issue [9].

$$wMAPE^2 = \frac{\sum_{i=1}^{n} \sum_{t=1}^{t_i} |\hat{y}_{it} - y_{it}|}{\sum_{i=1}^{n} \sum_{t=1}^{t_i} y_{it}}$$

It is important to note that, wMAPE is not bounded by 100 and it highly penalizes the error when sales volume is low. By analyzing the sales distribution in the dataset, we observe that 80% of the

[2] y_{it} and $\hat{y}_{it}$ represents the actual and predicted sales of product i at time t.

Table 1: KNN Baselines Vs Image RNNs: Product Category View

Model	Jeans			Shirt			T-shirt			Top		
	MAE	wMAPE	PCC	MAE	wMAPE	PCC	MAE	wMAPE	PCC	MAE	wMAPE	PCC
Category KNN	3.91	99.38	0	6.09	79.28	0.18	5.39	90.18	0.06	6.82	88.51	0.118
Color+Category KNN	3.94	99.95	0.01	6.02	78.38	0.15	5.39	90.15	0.06	6.70	86.98	0.13
Embedding KNN	2.48	63.12	0.34	4.36	56.78	0.40	3.63	61.11	0.37	4.69	61.26	0.41
Standard-Image RNN	3.53	89.92	0.13	5.72	74.45	0.23	4.79	80.61	0.23	6.17	80.53	0.24
Attended-Image RNN	**1.75**	**43.91**	**0.76**	**2.92**	**37.99**	**0.68**	**2.79**	**46.83**	**0.69**	**3.25**	**42.31**	**0.72**

Table 2: KNN Baselines Vs Image RNNs: Overview

Model	MAE	wMAPE	PCC
Category KNN	5.27	88.72	0.09
Color+Category KNN	5.37	90.30	0.11
Embedding KNN	3.66	61.79	0.39
Standard-Image RNN	4.85	81.91	0.25
Attended-Image RNN	**2.68**	**44.92**	**0.75**

products have an average sales of 10 or less per week. Considering this lower volume of sales which could commonly lead to higher wMAPE [9], we also consider Mean Absolute Error (MAE) and Pearson Correlation Coefficient (PCC) as other evaluation metrics to report the results. While wMAPE and MAE estimate the error, PCC (which ranges from -1 to 1) estimates whether the actual and predicted time-series move in the same (+1) or opposite direction (-1).

4.2 Comparing Baseline KNNs with Image RNNs

In this section, we compare and contrast Attribute and Embedding KNN approaches with Image RNN models. In Attribute KNN, we consider two models namely Category KNN and Category+Color KNN. Category KNN defines similarity of products based on the category match and Category+Color KNN defines the similarity of products based on both color and category match. On the other hand, Embedding KNN defines the similarity of products based on the cosine distance between the product image embeddings (obtained from Inception-V3 [13]). Going forward, Category KNN, Category+Color KNN and Embedding KNN have been considered as the 3 baselines for our experiments. Table 2 compares the 4-week forecasting error between KNN baselines and Image RNN models and report the avg. wMAPE, MAE and PCC across all the products. Based on the insights from the Fashion retailers, we observed that the first 4 week sales of a new product determine the extent of future pullout/replenishment. Hence, we have formulated our models as a 4-week forecasting problem. However, in Section 4.5, we also show the effect of forecasting errors when forecast horizon changes to 8 or 12 weeks.

Standard-Image RNN model would be the first obvious design choice for any deep learning scientist if they were asked to model the new product forecast as a sequence learning problem leveraging the image data. However, as depicted in Table 2, we observe that Standard-Image RNN model underperforms as compared to the Baseline Embedding KNN model by a wMAPE difference of 20%, which is further highlighted by the fact that PCC gets reduced by 0.14. The reason being the noise (i.e. irrelevant/unrelated features) in the image embedding (obtained from Inception-V3 [13]) which is not properly filtered out by the RNN. The fact that inception embeddings are learnt from a different classification task and input images may not exactly focus on the product-of-interest, leads to the noise in the image embeddings. To tackle this problem, Attended-Image RNN has been proposed to learn the relative importance across features in the image embedding, which further leads to effective noise cancellation arising out of unrelated/irrelevant image features. This attention-based filtering enables effective sequence learning in RNN models which is evident from the results depicted in Table 2. As explained in Section 4.1, since we are dealing with very low sales volume, we are expected to have higher wMAPE in general. Even in this scenario, Attended-Image RNN is able to manage with a wMAPE of 44.92% and outperforms the baseline Embedding KNN by a wMAPE difference of 17%. We also observe a drastic improvement in PCC (0.75) for Attended-Image RNN indicating a strong positive correlation. As an example use-case to illustrate good vs bad forecast, Figure 6 compares the actual ground truth vs predicted 12-week sales across the studied models for a particular product, and we observe that correlation between ground-truth and prediction gradually improves as we move towards Attended-Image RNN. We also observe that, both Image RNN techniques (i.e. with and without attentions) perform significantly better than Baseline 1 and 2 (i.e. Category KNN and Color+Category KNN).

Likewise, Table 1 compares the baseline approaches with Standard-Image RNN and Attended-Image RNN for the top-4 product categories (based on sales). Even at the category level, we observed similar insights, where Embedding KNN outperforms Standard-Image RNN while Attended-Image RNN outperforms the Embedding KNN approach. Results wrt. MAE metric also reflect similar insights.

4.3 Comparing Multi-modal approaches

In this section, we illustrate the impact of leveraging multi-modal data for new product time-series forecasting. Table 4 compares the 4-week forecasting error across the studied Multi-modal RNN models by reporting the achieved avg. wMAPE, MAE and PCC across all the products. As we observe in Table 4, Concat Multi-modal RNN outperforms the Attended-Image RNN by a wMAPE difference of 2%. Likewise, Residual Multi-modal RNN outperforms Concat Multi-modal RNN by a wMAPE difference of 1%, and Cross-Attention RNN slightly outperforms the Residual Multi-modal RNN by a marginal wMAPE difference of 0.5%. Thus the best performing Multi-modal RNN (i.e Residual or Cross-Attention RNN) outperforms the attended-Image RNN (which do not leverage multi-modal data) by a wMAPE difference of 3.5%. This iterates the importance

Table 3: Comparison of Multi-modal RNN models: Product Category View

Model	Jeans			Shirt			T-shirt			Top		
	MAE	wMAPE	PCC	MAE	wMAPE	PCC	MAE	wMAPE	PCC	MAE	wMAPE	PCC
Attended-Image RNN	1.75	43.91	0.76	2.92	37.99	0.68	2.79	46.83	0.69	3.25	42.31	0.72
Concat Multi-modal RNN	1.63	41.53	0.74	2.84	36.93	0.74	2.79	46.83	0.70	3.19	41.55	0.75
Residual Multi-modal RNN	**1.53**	**38.58**	**0.80**	**2.76**	**35.94**	**0.78**	**2.79**	**47.00**	**0.71**	**3.07**	**40.09**	**0.77**
Cross-Attention RNN	**1.54**	**39.29**	**0.78**	**2.62**	**34.05**	**0.79**	**2.63**	**44.21**	**0.72**	**2.98**	**38.89**	**0.79**

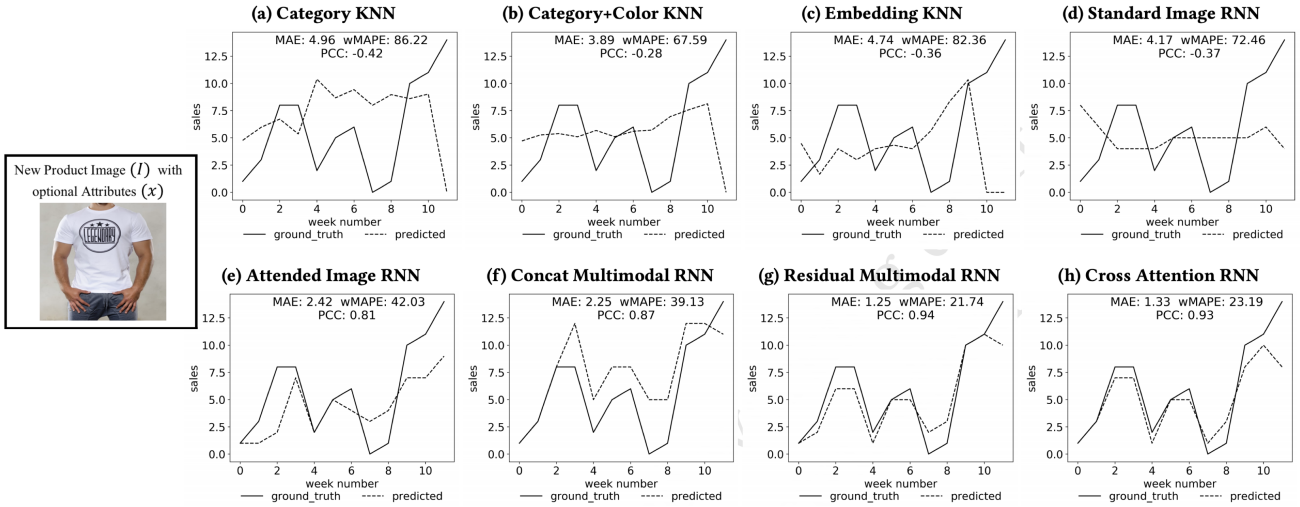

Figure 6: Example illustrating the variation in forecast across Models for a particular product

of augmenting textual data with image data for improving the forecasting accuracy of new products. Multi-modal fusion also leads to strong positive correlation between the actual and predicted sales which improves the PCC to peak at 0.8 (Table 4) for multi-modal RNN as compared to 0.75 achieved by the Attended-Image RNN. Figure 6 illustrates the improvement in the achieved correlation

Table 4: Comparison of Multi-modal RNN models: Overview

Model	MAE	wMAPE	PCC
Attended-Image RNN	2.68	44.92	0.75
Concat Multi-modal RNN	2.55	42.88	0.76
Residual Multi-modal RNN	**2.49**	**42.00**	**0.79**
Cross-Attention RNN	**2.46**	**41.44**	**0.80**

through an example product. We also repeated our experiments for Top-4 product categories which reflects similar insights as depicted in Table 3. It is important to note that, the proposed multi-modal fusion techniques not only aim to improve accuracy but also prevents reduction in the accuracy when noisy data-sources are added either by skipping connections (Residual RNN) or learning attention weights (Cross-Attention RNN). This aspect is very crucial in the industrial settings where new data sources have to be opportunistically added to improve the new product forecasting while guaranteeing the current performance of the model. Among the best performing Multi-modal RNN models (i.e Residual or Cross-Attention RNN), Cross-Attention RNN has an added advantage of providing explainability to its predictions, which is discussed in the next section.

4.4 Explainability from Cross-Attention RNN

Attention weights derived from Cross-Attention RNN provide better explainability to the model predictions. An example mentioned in Figure 7 illustrates various supplementary explanations which the Cross-Attention RNN would output in addition to the forecast. For every time-step prediction in Cross-Attention RNN, system outputs the following: (i) exogenous attention weights - which highlights the percentage contributed by each exogenous attributes towards the prediction, (ii) multi-modal attention weights - which highlights the percentage contributed by each modality towards the prediction. Forecasting graph in Figure 7(a) peaks at fourth week and if one is curious to know the reason behind it, explanations provided by Cross-Attention RNN could help. Explanations from Figure 7(b) highlight that holiday is the main reason for the sale peak, as the attention weights of holiday dimension is relatively higher than the other exogenous dimensions in that particular week. We also cross-validated it by fetching the number of holidays in that particular week from the actual dataset and we noticed 2 holidays in that particular week. Thus, cross-attention RNN could provide better explainability wrt. exogenous attributes. In addition, Figure 7(c) explains that temporal and text features are the most dominating factors for this specific instance of prediction as compared to the image features. These multi-modal attention weights not only enable explainability in models, but also work in an effective manner to filter out irrelevant or repetitive noisy features which are very common in multi-modal data.

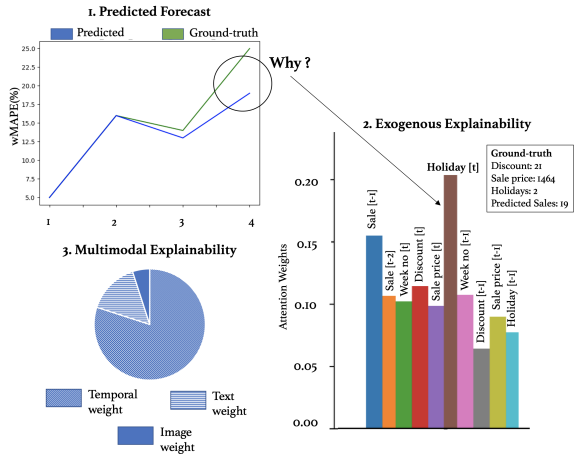

Figure 7: Local Explainability with Cross-Attention RNN

4.5 Effect of forecast horizon

In this section, we vary the forecast horizon across 4, 8 and 12 weeks and observe its impact wrt. forecasting errors. This experiment is conducted on Embedding KNN, Attended-Image RNN and Cross-Attended RNN which are the top performing models in baseline, image-only and multi-modal scenarios. These models are trained with varying time horizons and associated wMAPE and PCC is reported in Table 5. From results we observe that, Attended-Image RNN and Cross-Attention RNN fare better than Baseline Embedding KNN by a good margin in all varied forecast horizons. Besides this, Cross-Attention RNN also showed an improvement in accuracy and correlation as compared to Attended-Image RNN across all varied forecast horizons.

Table 5: Effect of Sequence Length

Time (weeks)	Embedding KNN		Attended-Image RNN		Cross-Attention RNN	
	PCC	wMAPE	PCC	wMAPE	PCC	wMAPE
4	0.39	61.79	0.75	44.92	0.80	41.44
8	0.37	64.3	0.73	49.56	0.78	49.27
12	0.42	67.58	0.74	51.75	0.78	50.89

However it is important to observe that, an increase in the forecast-horizon is accompanied by a corresponding increase in the error irrespective of the model under study. So, the preferred design choice is to use new product forecast techniques to predict initial shorter horizons when no historical data is available and later switch back to classical forecasting techniques when historical data is available. This hybrid approach could lead to an improved forecasting accuracy. Based on the insights from Fashion retailers, we obverse that the first 4 week sales of a product decides further continuation cycles. Hence forecast horizon of 4 weeks is of utmost importance and is a minimal requirement for new product forecasting techniques.

5 CONCLUSION AND FUTURE WORK

In this paper we proposed and empirically evaluated several variants of attention based multi-modal sales time-series forecasting for new products. This allows us to naturally incorporate product images, any available attributes and external factors into the forecasting model. Unlike classical uni-variate time-series forecasting which look at one time-series at a time, our model analyses all the available time-series together in one single forecast model. Currently we are working on extending this model to predict store-level sales forecasts, as a consequence of which we will a have an embedding for a store along with the product embedding.

The fashion industry is considered to be the world's second largest polluter, after oil and gas. Accurate new product sales forecast helps in better supply chain planning and reducing unsold inventory which is one step into this direction to make fashion industry more sustainable. While we focus mainly on fashion retail, the same models could be used in any retail industry where new products are introduced and there is a need to forecast the demand based on historical sales data of other products.

REFERENCES

[1] United States Environmental Protection Agency. 2017. *Textiles: Material-Specific Data.* https://www.epa.gov/facts-and-figures-about-materials-waste-and-recycling/textiles-material-specific-data

[2] Lennart Baardman, Igor Levin, Georgia Perakis, and Divya Singhvi. 2017. Leveraging comparables for new product sales forecasting. *SSRN 3086237* (2017).

[3] Tsan-Ming Choi, Chi-Leung Hui, and Yong Yu. 2013. *Intelligent fashion forecasting systems: models and applications.* Springer.

[4] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems.* 577–585.

[5] Giuseppe Craparotta, Sébastien Thomassey, and Amedeo Biolatti. 2019. A Siamese Neural Network Application for Sales Forecasting of New Fashion Products Using Heterogeneous Data. *International Journal of Computational Intelligence Systems* 12 (11 2019). https://doi.org/10.2991/ijcis.d.191122.002

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR.* 770–778.

[7] Kenneth B Kahn. 2014. Solving the problems of new product forecasting. *Business Horizons* 57, 5 (2014), 607–615.

[8] Nadia Khomami. 2018. *Burberry destroys £28m of stock to guard against counterfeits.* https://www.theguardian.com/fashion/2018/jul/19/burberry-destroys-28m-stock-guard-against-counterfeits

[9] Spyros Makridakis, Steven C Wheelwright, and Rob J Hyndman. 2008. *Forecasting methods and applications.* John wiley & sons.

[10] Elizabeth Paton. 2018. *H&M, a Fashion Giant, Has a Problem: $4.3 Billion in Unsold Clothes.* https://www.nytimes.com/2018/03/27/business/hm-clothes-stock-sales.html

[11] Pawan Kumar Singh, Yadunath Gupta, Nilpa Jha, and Aruna Rajan. 2019. Fashion Retail: Forecasting Demand for New Items. In *The fourth international workshop on fashion and KDD.* Anchorage, Alaska - USA.

[12] Jesper Starn. 2017. *A Power Plant Is Burning H&M Clothes Instead of Coal.* https://www.bloomberg.com/news/articles/2017-11-24/burning-h-m-rags-is-new-black-as-swedish-plant-ditches-coal

[13] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *CVPR.*

[14] Sébastien Thomassey. 2014. Sales forecasting in apparel and fashion industry: A review. In *Intelligent fashion forecasting systems: Models and applications.* Springer, 9–27.

[15] Sébastien Thomassey and Antonio Fiordaliso. 2006. A hybrid sales forecasting system based on clustering and decision trees. *Decision Support Systems* 42, 1 (2006), 408–421.

[16] Sébastien Thomassey and Michel Happiette. 2007. A neural clustering and classification system for sales forecasting of new apparel items. *Applied Soft Computing* 7, 4 (2007), 1177–1187.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. (2017).

[18] Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* 1, 2 (1989), 270–280.

[19] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. (2015).