Annals of Operations Research (2024) 339:1051–1072
https://doi.org/10.1007/s10479-023-05447-7

ORIGINAL RESEARCH

Intermittent demand forecasting with transformer neural networks

G. Peter Zhang[1] · Yusen Xia[2] · Maohua Xie[2]

Accepted: 18 May 2023 / Published online: 23 June 2023

Abstract

Intermittent demand forecasting is an important yet challenging task in many organizations. While prior research has been focused on traditional methods such as Croston's method and its variants, limited research has been conducted using advanced machine learning or deep learning methods. In this study, we introduce Transformer, a recently developed deep learning approach, to forecast intermittent demand. Its effectiveness is empirically tested with a dataset of 925 intermittent demand items from an airline spare parts provider and compared with that of two traditional methods such as Croston's and the Syntetos–Boylan approximation as well as several popular neural network architectures including feedforward neural networks, recurrent neural networks, and long short-term memory. Our results based on six different forecasting performance measures show that Transformer performs very well against other methods in a variety of settings. We also examine how data sparsity impacts model performance and find that different models perform similarly when sparsity is low. Although the performance of all models generally gets worse as the sparsity level increases, the advantage of Transformer over other models increases with sparsity.

Keywords Intermittent demand · Forecasting · Neural networks · Transformer · Deep learning methods

1 Introduction

Intermittent demand forecasting remains an important yet challenging task in many organizations. Intermittent demand items are common for firms with slow moving items such as after-sales service parts and high-valued capital goods, as well as those with high number of stock keeping units (SKUs). The number of these infrequent demand items as well as the associated inventory cost could be significant, accounting for up to 60% or 70% of a firm's total inventory investment in some industries (Boylan & Syntetos, 2021; Hu et al., 2018;

✉ Yusen Xia
ysxia@gsu.edu

[1] Department of Management, Robinson College of Business, Georgia State University, Atlanta, GA 30303, USA

[2] Institute for Insight, Robinson College of Business, Georgia State University, Atlanta, GA 30303, USA

Johnston et al., 2003). Therefore, accurate forecasting of these items is essential to a firm's successful inventory planning and management. Boylan and Syntetos (2021) provide a nice discussion of the economic and environmental benefits of accurate intermittent demand forecasting, including lower inventory holding costs, higher revenues through increased service levels, and less environmental harm due to lower risk of obsolescence. The role of intermittent demand in supply chain forecasting is reviewed by Syntetos et al. (2016).

Forecasting intermittent demand is challenging because intermittent demand occurs sporadically and at random. The intermittent demand time series is characterized by zero demands in many periods interspersed with non-zero demands that are often highly variable. Unlike non-intermittent demand forecasting where only demand size is uncertain, both the timing of a positive demand occurrence and the size of the demand are uncertain in intermittent demand forecasting. Therefore, for intermittent demand forecasting, one is concerned with not only the demand size, but also when a positive demand will occur. This may be the reason that limited research attention has been given to intermittent demand forecasting (Nikolopoulos, 2021), although there has been an increasing interest in the topic in recent years.

Research on intermittent demand has largely followed Croston (1972) in which Croston developed the first systematic approach to intermittent demand forecasting. To deal with the aforementioned uncertainties inherent in intermittent demand time series, Croston estimates the demand size and the inter-demand interval separately using exponential smoothing and then the ratio of the two estimates is the demand forecast. Although Croston's method outperforms traditional exponential smoothing method for intermittent demand forecasting, Syntetos and Boylan (2001) show that Croston's method is positively biased. One of the most well-known modifications to Croston's method is proposed by Syntetos and Boylan (2005) who correct the bias with an approximation method (known as the Syntetos–Boylan Approximation or SBA in the literature), which has been shown to be generally more effective than the Croston method (Babai et al., 2014; Boylan & Syntetos, 2007; Eaves & Kingsman, 2004; Teunter & Sani, 2009). Other modifications include Babai et al. (2019), Doszyn (2019), Leven and Segerstedt (2004), Prestwich et al. (2014), Shale et al. (2006), Syntetos and Boylan (2001), Teunter et al. (2011), and Yang et al. (2021) among others.

Alternative methods have also been developed for intermittent demand forecasting. These include the bootstrapping method (Hasni et al., 2019; Syntetos et al., 2015; Willemain et al., 2004; Zhou & Viswanathan, 2011), the aggregate-disaggregate approach (Boylan and Babai, 2016; Nikolopoulos et al., 2011; Petropoulos et al., 2016), hybrid models (Hua & Zhang, 2006; Hua et al., 2007; Nasiri Pour et al., 2008; Zhuang et al., 2022), and others (Nikolopoulos et al., 2016; Zhu et al., 2017). A good overview of intermittent demand forecasting methods is provided by Boylan and Syntetos (2021). An extensive review of various methods for spare parts demand forecasting is given in Pince et al. (2021).

Neural networks represent another category of promising methods due to their attractive features for time series forecasting in general (Zhang et al., 1998) and intermittent demand forecasting in particular (Kourentzes, 2013). The main advantage of neural networks is their flexible, nonlinear modeling capability, allowing them to deal with complex structures in the intermittent demand data as Pince et al. (2021) state that neural networks, as a versatile tool, "can capture nonlinear patterns in the data, such as intermittence and lumpiness, better than most time-series methods." Although neural networks have received limited attention in the intermittent demand forecasting literature, several different models have been used for intermittent demand forecasting with varying levels of success (Babai, et al., 2020; Gutierrez et al., 2008; Kourentzes, 2013; Lolli et al., 2017; Mukhopadhyay et al., 2012). The majority of these models are feedforward and recurrent types of neural network.

In this study, we propose the use of transformer neural networks (or transformers) for intermittent demand forecasting. Transformers are newly developed neural network architectures based on the attention mechanism (Vaswani et al., 2017). The introduction of the attention mechanism has improved the success of various deep learning models (Bahdanau et al., 2015). Attention is a component of deep learning to capture important and relevant sequential information in inputs to improve output prediction accuracy. The use of the attention mechanism in neural network architectures has been increasingly common (Niu et al., 2021). Originally developed for natural language processing, transformers with the attention mechanism have gained significant interest and achieved great success in many fields including natural language processing, computer vision, and image analysis. Due to their capability of modeling long-term dependencies and interactions in sequential data, transformers hold great promise in time series modeling and forecasting. Although researchers have begun to explore the use of transformers for time series forecasting (Wen et al., 2022), to the best of our knowledge, no research has examined how effective transformers are in predicting intermittent demand.

The effectiveness of transformer models (Transformer hereafter) for intermittent demand is tested with a data set of weekly demand series for 925 SKUs from an airline spare parts provider. We compare the performance of Transformer with that of two traditional methods of Croston's and SBA as well as several neural network architectures including feedforward neural networks, recurrent neural networks, and long short-term memory. Our results based on six different forecasting performance measures show that Transformer is the overall best method under a variety of settings.

The rest of the paper is organized as follows. In the next section, we provide a focused review of the relevant studies in intermittent demand forecasting with neural networks. Section 3 describes data and methodology. Section 4 reports the results. Finally, Sect. 5 concludes the paper.

2 Intermittent demand forecasting with neural networks

Although neural networks have been widely used for time series forecasting (Zhang et al., 1998), only a few studies have used neural networks for intermittent demand forecasting (Lolli et al., 2017). One of the first studies is conducted by Guitierrez et al. (2008) who compare the performance of neural networks with that of several traditional methods such as the Croston's, single exponential smoothing, and SBA. Using 24 intermittent demand time series each with 967 daily observations, they find that the three-layer feedforward neural network architecture they proposed with two input nodes (representing the previous period demand and the number of periods separating the last two nonzero demands), three hidden nodes, and one output node (for the predicted demand) generally outperforms the traditional methods.

Nasiri Pour et al. (2008) is another early study that adopts neural networks for intermittent demand forecasting. They propose a hybrid approach with neural networks used to forecast the occurrence of non-zero demand, and traditional methods used to predict the demand size. The results using 30 spare parts demand series with 69 monthly observations show that the hybrid model performs better than SBA and several standalone neural network models including feedforward, recurrent, and generalized regression networks.

Using the same data from Guitierrez et al. (2008), Mukhopadhyay et al. (2012) conduct another comparative study involving the feedforward neural network. They find that the neural network model and the 5-day weighted moving average model perform well relative to several traditional methods including simple exponential smoothing (SES), Croston's, and SBA. The neural network used in the study has the same basic structure as Guitierrez et al. (2008) except that one of the input nodes represents the cumulative number of successive periods with zero demand instead of the number of periods separating the last two non-zero demands. Mukhopadhyay et al. (2012) also examine the effect of different data splitting strategies, forecast lead time, and performance measures on the forecasting performance of each method.

Kourentzes (2013) proposes two feedforward neural network models inspired by Croston's method. Both models use non-zero demands and inter-demand intervals as inputs, while the output layer differs in that one employs two nodes to represent predicted demand and inter-demand interval separately (NN-Dual), while the other employs just one node to represent predicted demand rate (NN-Rate). Based on 1000 simulated intermittent time series, the results show that these models perform worse than Croston's method and several modified Croston's methods in terms of forecast accuracy. However, the neural networks, especially NN-Rate, perform much better than Croston's method and its variants with regard to inventory metrics (i.e. service levels).

All of the above-mentioned studies adopt feedforward type of neural networks. In addition to the feedforward network, Lolli et al. (2017) also consider two other types of neural networks: recurrent and time-delay, for intermittent demand forecasting. They use three input nodes in their networks: last demand size, the number of periods separating the last two nonzero demands, and the cumulative number of successive periods with zero demand, which is the combined set of inputs used by Guitierrez et al. (2008) and Mukhopadhyay et al. (2012). With an experiment study using two different learning approaches (back-propagation and extreme learning machine), two learning mode (batch and online), and three neural network architectures (feedforward, recurrent, and time-delay) as well as 24 weekly auto spare parts demand intermittent time series with varying length from 61 to 414 observations, they find that neural network models generally perform well relative to Croston's method and SBA in terms of forecast accuracy.

A more recent study by Babai et al. (2020) makes a further modification of the neural network structure proposed by Guitierrez et al. (2008). In addition to the same two inputs (i.e., the last period demand and the number of periods separating the last two nonzero demands) used by Guitierrez et al. (2008), Babai et al.'s (2020) proposed model also adds multiple lagged demands prior to the last period and the number of periods between the forecast period and the last zero demand period. In addition, they consider varying numbers of hidden nodes such as three, five and nine, instead of three used in Guitierrez et al. (2008). Using a monthly demand dataset of 5,135 SKUs from an airline each with 123 observations and considering both forecasting and inventory performance metrics, their results are somewhat mixed depending on the performance metrics used. Generally speaking, the proposed neural network models outperform those of Guitierrez et al. (2008) and are better than or comparable to the parametric methods. Surprisingly, the results also show that SES performs very well overall in terms of both forecast accuracy and inventory efficiency, while the two bootstrapping methods examined do not perform well.

Several observations can be made following the above discussion. First, neural networks are promising for intermittent demand forecasting. Although there are mixed findings, neural networks generally outperform traditional parametric or nonparametric methods in terms of forecasting and/or inventory performances. Second, the number of intermittent time series and the number of observations in each series vary quite significantly in published studies. For example, both Guitierrez et al. (2008) and Lolli et al. (2017) use 24 time series in their studies, but Lolli et al.'s (2017) time series length varies from 61 to 414 observations while Guitierrez's et al. (2008) time series all have 967 observations. On the other hand, Babai et al. (2020) employ over 5,000 time series, but with a relatively small number of observations (123) in each series. Third, although the focus has been on forecasting accuracy, there is no consensus on best forecast accuracy or error measures. Thus, different error measures are used by different researchers. Finally, the neural network architectures considered in the literature are similar with the feedforward multiple layer networks the most popular. Guitierrez et al. (2008) neural network model appears influential because many subsequent studies either adapt their model with different inputs or use the model as a benchmark for comparison purposes.

3 Data and methods

3.1 Data

The data used for this study is from an airline service parts provider/distributor that is located in the Southeast region of U.S. The company provides us with weekly demand data for a total of 5,800 SKUs from January 2018 to December 2021. Following the literature (e.g., Willemain et al. 1994*),* we use the squared coefficient of variation of demand (CV^2) and the average demand interval (ADI) to identify noise and intermittence levels. Then using the classification scheme devised by Syntetos et al. (2005) with their cutoff points of 0.49 for CV^2 and 1.32 for ADI, we divide our data into four categories: intermittent, lumpy, smooth, and erratic. Because each category has its unique characteristics, we elect to focus in this study on SKUs in the intermittent category (CV$^2 < 0.49$ and ADI > 1.32). There are 925 items in this category. Because the starting and ending dates may differ for different SKUs, the length of time series varies. The minimum number of observations is 106 and the maximum is 207.

Table 1 provides a summary of demand characteristics for our sample. It shows that there is a high degree of variation in both demand size and demand intermittence. For example, the mean demand is 2.55 with the minimum of 1.07 and the maximum of 404.75. While the mean demand interval is 9.27 weeks, the minimum mean interval is 1.26 weeks and the maximum mean interval is 184 weeks. The demand per period also shows significant lumpiness with the lowest equal to 0.07 and the largest equal to 10.87.

Table 1 Demand characteristics

	Demand size (units)		Demand interval (weeks)		Demand per period	
	Mean	Std dev	Mean	Std dev	Mean	Std dev
Min	1.07	1.66	1.26	0.00	0.07	0.28
25th	2.25	0.89	5.09	4.68	0.16	0.65
Median	2.55	1.29	9.27	8.32	0.30	0.94
75th	2.94	1.70	16.18	15.25	0.57	1.33
Max	404.75	203.36	184.00	91.92	10.87	71.74

Table 2 Sparsity frequency distribution

Sparsity level	Zero percentage	Number of SKUs
1	[0.2, 0.3)	3
2	[0.3, 0.4)	4
3	[0.4, 0.5)	14
4	[0.5, 0.6)	32
5	[0.6, 0.7)	78
6	[0.7, 0.8)	126
7	[0.8, 0.9)	280
8	[0.9, 1.0)	388

Because different methods may vary in performance due to different levels of demand intermittence, we group SKUs in our sample based on sparsity defined as the percentage of zero demand periods in each time series. Table 2 shows the frequency distribution of the SKUs with regards to different levels of sparsity. Since no SKUs have zero demand percentage below 0.2, we have a total of eight different levels of sparsity starting from level 1 which is for SKUs with zero demand percentages between 0.2 and 0.3, level 2 with zero percentages between 0.3 and 0.4, and so on. Interestingly, the number of SKUs increases with the sparsity level with only 3 at level 1 and 388 at level 8. Most SKUs (794 or 85.8%) are at levels 6–8, indicating an overall high degree of demand intermittence in our data.

All data are normalized using the Min–Max transformation so that all values are between 0 and 1. Normalizing numeric inputs generally avoids the problem that when some values or features dominate others in magnitude because the model performance is likely impacted by very high input values and thus under-weigh low valued inputs regardless of their potential contributions. In addition, we partition the data into three parts: 60% in the training set, 20% in the validating set, and 20% in the testing set.

3.2 Methods

3.2.1 Croton's method

While simple exponential smoothing (SES) has been widely used for forecasting, it has significant limitation in intermittent demand forecasting because it weights recent data more heavily than older observations. As a result, the SES would produce a forecast biasedly high after a demand occurs but low before a demand realizes, creating unnecessarily high stock levels. Croston (1972) tries to address this bias by estimating the demand size and demand interval separately. To make the method easier to implement, he uses SES to estimate these quantities. Specifically, let Z_t (D_t) be the estimated (actual) non-zero demand for time t, P_t (Q_t) be the estimated (realized) inter-demand interval since the last nonzero demand. The Croston's updating formulae based on SES are given below:

$$Z_t = \alpha D_t + (1 - \alpha) Z_{t-1} \tag{1}$$

$$P_t = \alpha Q_t + (1 - \alpha) P_{t-1} \tag{2}$$

where α is a smoothing constant between 0 and 1. Note that Croston's original method uses the same smoothing constant for updating both smoothed estimates, although different smoothing constants can be used in the above equations. Croston (1972) suggests a range between 0.1 and 0.3 for α. We use the same smoothing coefficient as indicated in the above updating equations in this study.

The forecast demand for the next time period, Z_{t+1}, is then estimated as a ratio of the above quantities:

$$Z_{t+1} = Z_t / P_t \tag{3}$$

Note that $\mathrm{P_t}$ is strictly positive. If the actual demand (D_t) is zero, then Z_t, P_t, and Z_{t+1} will remain the same, i.e., no updating occurs. On the other hand, if demand occurs in every period ($P_t = 1$), then $Z_{t+1} = Z_t$, or Croston's method would produce the same forecasts as SES.

In order to find the best smoothing coefficient, we conduct an experiment with α varying from 0.1 to 0.5 with a step size of 0.01. The best coefficient for each time series is chosen by minimizing the mean squared error (MSE) as Lolli et al. (2017) suggested that using MSE as an accuracy measure is analogous to the neural network training method. However, unlike Lolli et al. (2017) who use the training set to determine the optimal parameters, we use the validation set as this is more in line with how the neural network model building and parameter estimation are performed.

3.2.2 Syntetos–Boylan approximation

Because of the positive bias in Croston's method, several adjustments or modifications to Croston's method have been proposed (Syntetos & Boylan, 2001, 2005; Teunter & Sani, 2009). Among them, the Syntetos–Boylan approximation (SBA) developed by Syntetos and Boylan (2005) has received the most attention from the literature and most studies use it as a benchmark. The SBA method adjusts the Croston forecast downward by multiplying a factor of (1- α/2) as follows:

$$Z_{t+1} = \left(1 - \frac{\alpha}{2}\right) Z_t / P_t \tag{4}$$

The optimal smoothing coefficient for the SBA method is obtained using the same approach as with Croston's method. That is, we select the smoothing constant over the range of 0.1 to 0.5 in steps of 0.01 that minimizes MSE of the validation sample.

3.2.3 Feedforward and recurrent neural networks

We use the same feedforward neural network (FNN) and recurrent neural network (RNN) architectures as in Lolli et al. (2017). Both FNN and RNN models are fully connected with a single hidden layer. In addition, we use the following three inputs in these networks: the demand at time period t, the number of periods separating the last two non-zero demand transactions at the end of the immediately preceding period, and the cumulative number of successive periods with zero demand. See Lolli et al. (2017) for details on the structures of these models as well as how the models are trained.

3.2.4 Long short-term memory model

The long short-term memory (LSTM) model is proposed to improve the performance of RNN due to RNN's shortcomings of potential information loss or vanishing/exploding gradient

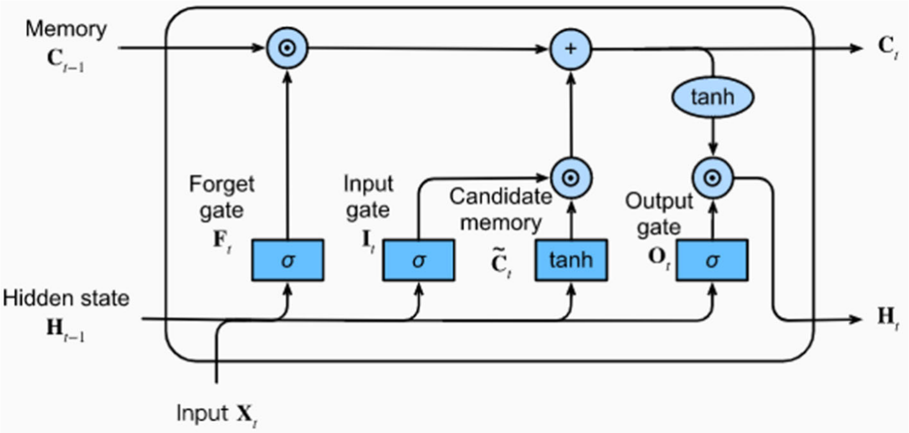
Memory $\mathbf{C}_{t-1}$
Forget gate $\mathbf{F}_t$
Input gate $\mathbf{I}_t$
Candidate memory $\tilde{\mathbf{C}}_t$
Output gate $\mathbf{O}_t$
σ
σ
tanh
σ
tanh
Hidden state $\mathbf{H}_{t-1}$
Input $\mathbf{X}_t$
$\mathbf{C}_t$
$\mathbf{H}_t$

Fig. 1 Illustration of the LSTM architecture

problems, especially over long time periods (Hochreitier and Schmidhuber, 1997; Zhang et al., 2021). LSTM is a special type of RNN that is capable of modeling long-term dependencies. Specifically, LSTM adds four gates/cells to an RNN model as shown in Fig. 1:

- Forget gate: this gate helps decide what type of information to keep and what information to forget. This is useful for long time series because it helps keep the most useful or relevant information. At any time t, an LSTM receives input vector $\mathrm{X}_t \in R^{n\times d}$, the previous time period hidden state $\mathrm{H}_{t-1} \in R^{n\times h}$, the previous time period cell state $C_{t-1} \in R^{n\times h}$ where n represents the number of samples in a batch, h is the number of cells in the hidden layer, and d is the number of inputs. The gate with the sigmoid activation function ($\sigma(x) = \frac{1}{1+e^{-x}}$) determines what information to be removed from X_t and H_{t-1} (Zhang et al., 2021). At the Forget gate, the model takes the input of X_t with the hidden state information H_{t-1}, and outputs the following:

$$F_t = \sigma\left(\mathrm{X}_t W_{xf} + \mathrm{H}_{t-1} W_{hf} + \mathrm{b}_f\right)$$

  where $W_{xf} and\ W_{hf}$ are weight parameters, and b_f is a bias parameter. Note that the range of sigmoid function is between 0 and 1, the values of F_t will be no more than 1 as well, essentially keeping part of the information for further processing.

- Input gate: this gate is used to quantify the importance of the information from the input. With the input vectors of X_t and H_{t-1} through the sigmoid activation function, the output concatenates the two vectors, which further becomes part of the input to the memory state: where $\odot$ refers to the elementwise Hadamard product operator; $\widetilde{C}_t$ is a term that represents what current information is kept for future memory updating purpose. Its values are in the range of (−1, 1) by utilizing the tanh function as shown below,

$$I_t = \sigma\left(\mathrm{X}_t W_{xf} + \mathrm{H}_{t-1} W_{hf} + \mathrm{b}_f\right)$$

- Memory gate: the primary purpose of this gate is to determine what information to carry over to the next stage/period. The next period memory information considers information from the previous period, the current period output from the forget gate, the hidden state, and the current period input:

$$C_t = F_t \odot C_{t-1} + I_t \odot \widetilde{C}_t$$

where $\odot$ refers to the elementwise Hadamard product operator; $\widetilde{C}_t$ is a term that represents what current information is kept for future memory updating purpose. Its values are in the range of $(-1, 1)$ by utilizing the tanh function as shown below,

$$\widetilde{C}_t = tanh(\mathrm{X}_t W_{xc} + \mathrm{H}_{t-1} W_{hc} + \mathrm{b}_c)$$

- Output gate: the output gate serves two purposes. The first is to generate the output values/vectors for the current period if such outputs are needed. Specifically, the output combines the input and hidden state with the sigmoid function:

$$O_t = \sigma(\mathrm{X}_t W_{xo} + \mathrm{H}_{t-1} W_{ho} + \mathrm{b}_o)$$

The second purpose of this gate is to derive the hidden state information for the next period from the current output and the long-term memory (C_t):

$$H_t = O_t \odot \tanh(C_t)$$

Overall, the next period hidden state depends on the current period output and concatenates with the current period memory state after the tanh activation function.

In summary, through these four different gates, LSTM is able to mitigate part of the gradient vanishing or exploding problem of RNN by transferring and keeping only some of the useful information from the past over time.[1]

We use the standard three-layer LSTM model in our study. For the LSTM model training, the hyperbolic tangent (tanh) is used as the activation function. We consider both the number of hidden neurons and the batch size as parameters and the optimal numbers are chosen from a fixed set of [5, 10, 20, 30, 40, 50, 60, 70] for the neurons and from the set of [10, 20, 30, 40, 50] for the batch size. The models are trained with the Adam optimization algorithm (Kingma & Ba, 2017), an extension to stochastic gradient descent that has been increasingly used in deep learning applications. The number of epochs is set at 100 with proper early stopping.

3.2.5 Transformer

Transformer represents a relatively new deep learning architecture proposed by Vaswani et al. (2017). It relies entirely on the self-attention mechanism to compute representations of input and output without resorting to recurrent or convolutional layers. At a high level, a Transformer model is composed of two components: position-wise feedforward network layer and multi-head attention layer. Position-wise network is a fully connected feedforward network, which is applied to each position separately and identically. This approach can ensure the position information of each symbol in the input sequence during the operation. Multi-head attention allows the model to focus on information from different representation subspaces from different positions by stacking multiple self-attention layers, just like multiple channels of a convolutional network.

When the sequence of inputs is long, LSTM may still have difficulty in interpreting the information. Part of the reason is that the input is sequential and entered one at a time. In contrast, transformers can take all the input at one time instead of sequentially (Vaswani et al., 2017). However, one challenge for simultaneous inputs is about how to treat various inputs differently by utilizing important information while ignoring noisy types of inputs.

[1] https://d2l.ai/chapter_recurrent-modern/lstm.html.

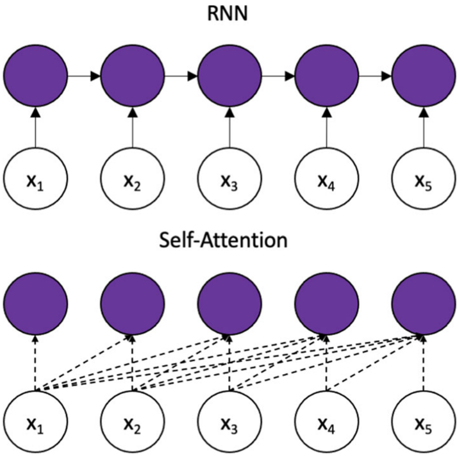

Fig. 2 Illustration of the self-attention

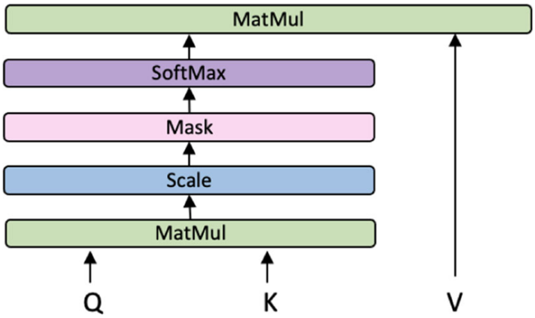

Fig. 3 Illustration of different components of the self-attention mechanism

Self-attention is an important mechanism that allows inputs to interact with each other and find out what information should be paid more attention to. It can extract importance of different period inputs, and assign weights to different inputs that will be aggregated into the final output. Figure 2 provides a high-level comparison between recurrent neural networks without and with self-attention. Note that when there is no self-attention mechanism in RNN (top part of the figure), inputs are taken into consideration sequentially, and they are treated equally. In comparison, when there is self-attention (lower part of the figure), each period takes inputs of all previous periods, and through training, weights can be assigned to each input, thus differentiating inputs by their importance.

Figure 3 shows a more detailed view of the self-attention mechanism regarding its major components. Each input is represented by three elements of query (Q), key (K), and value (V) vectors (Vaswani et al., 2017).[2] The core of the mechanism is to calculate the self-attention score, which is a measure of relevance between the current input and any other inputs in the previous sequence. In particular, the self-attention scores go through a softmax activation function defined as $softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}$, which scales to be a percentage of total focus that is given to an input in the sequence. Using the softmax function, the attention score by

[2] https://medium.com/mlearning-ai/transformer-implementation-for-time-series-forecasting-a9db2db5c820.

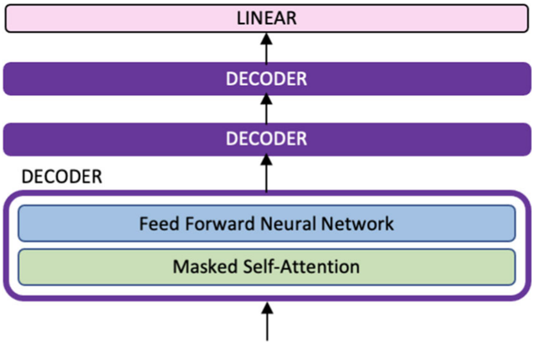

Fig. 4 Illustration of the transformer-decoder architecture

utilizing input vectors Q, K, and V can be calculated as:

$$\text{Attention}\,(Q,\ K,\ V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Note that self-attention scores are calculated by comparing query of the current input to the keys of all other inputs. In particular, it is scored as the dot product between the query vector of the current input and the key vector of the input being scored. Each input vector is multiplied by the corresponding softmax score. After which, the current input is encoded by summing all of the above scaled values.

The encoded input then goes to the Transformer-decoder as shown in Fig. 4. In each decoder component, first the self-attention scores are masked, meaning that certain values are hidden. The masks hide the ground truth information and help derive the right weights in the model during training. The question of how much information to hide/mask is treated as a hyperparameter, which can be derived by the prediction error in the training process. The masked self-attention scores pass through a fully connected feedforward neural network. And finally a linear model combines all relevant decoded values to generate an output as the predicted value. Note that the Transformer adopts an encoder and decoder scheme in which the encoder is used to derive the self-attention scores as illustrated in Fig. 3 while the decoder is used for prediction as shown in Fig. 4.

We use PyTorch, a machine learning framework based on the Python programming language to configure and train the transformer model. We make an effort to ensure that the encoder layer matches the input dimension size and the decoder layer fits the forecasting requirement for the dependent feature. The number of variables used as inputs to the model in encoder layer is three, which is the same as the input dimension size.[3] Since we are forecasting the intermittent demand in this study, the decoder layer is implemented as a linear layer. As with LSTM, transformer models are trained for 100 epochs with the Adam optimizer and MSE loss function.[4]

[3] https://www.medium.com/mlearning-ai/transformer-implementation-for-time-series-forecasting-a9db2db5c820

[4] https://www.medium.com/mlearning-ai/transformer-implementation-for-time-series-forecasting-a9db2db5c820.

3.3 Accuracy measures

Although there is no consensus on what forecast accuracy (or error) measures are most appropriate to evaluate intermittent demand forecasting methods, several measures are commonly used in the literature. While most studies adopt two or three such measures (Babai et al., 2014; Lolli et al., 2017), we elect to use five error measures, including the Mean Error (ME), the Mean Absolute Error (MAE), the Mean Absolute Percentage Error (MAPE), the Root Mean Squared Error (RMSE), and the Mean Absolute Scaled Error (MASE). We use the scaled measures as they are appropriate for intermittent time series forecasting. The formulas to calculate these measures are given below:

$$\mathrm{ME} = \frac{\sum_{t=1}^{n}(Z_t - D_t)}{\sum_{t=1}^{n} D_t}, \quad \mathrm{MAE} = \frac{\sum_{t=1}^{n}|Z_t - D_t|}{\sum_{t=1}^{n}|D_t - \frac{1}{n}\sum_{t=1}^{n} D_t|}, \quad \mathrm{RMS} = \frac{\sqrt{\frac{1}{n}\sum_{t=1}^{n}(Z_t - D_t)^2}}{\frac{1}{n}\sum_{t=1}^{n} D_t}$$

$$\mathrm{MAPE} = \frac{\sum_{t=1}^{n}|Z_t - D_t|}{\sum_{t=1}^{n} D_t}, \quad \mathrm{MASE} = \frac{\frac{1}{n}\sum_{t=1}^{n}|Z_t - D_t|}{\frac{1}{n_1-1}\sum_{t=2}^{n_1}|D_t - D_{t-1}|}$$

where D_t and Z_t are actual demand and the forecast, respectively; n is the number of periods in the test sample and n_1 is the number of periods in the training sample. In addition to these overall forecast error measures, we also use the percentage of best forecasts (PB) with a particular error measure to compare the performance of various methods. PB has been used by Gutierrez et al. (2008), Mukhopadhyay et al. (2012), and more recently Doszyn (2019).

4 Results

We largely follow the approach used by Doszyn (2019) to report results. We first report results for all data combined. Then we delve into how the performance of various methods varies with data sparsity. Because of potential outliers in results, we calculate both mean and median performance measures. While most prior studies on intermittent demand forecasting report only mean performance, Doszyn (2019) focuses exclusively on the median performance level.

Table 3 provides a summary of mean and median performance metrics (ME, MAPE, MAE, RMSE, and MASE) with different methods across all 925 SKUs. Several observations can be made based on the results in Table 3. First, SBA performs consistently better than Croston judged by both overall bias (ME) and forecast error measures with both means and medians. Second, Transformer is the best performer based on almost all error measures except for the median RMSE, in which case, its overall median RMSE is 3.100 which is lower than that for Croston and SBA, but higher than that with FNN, RNN, and LSTM. Although Transformer also has the lowest ME based on the mean, it is not the lowest with the median. The lowest median ME is achieved by LSTM although FNN, RNN, and SBA are not far behind. It is interesting to note that except for the median ME with the transformer, all methods yield negative MEs, suggesting that the methods generally overforecast demand. However, the positive bias is much higher with Croston than with all other methods based on both mean and median MEs. Because of the similarity in results using either the mean or the median, we will focus on the mean results in the following discussion.

To examine how data sparsity impacts forecasting performance, we divide our data into eight different buckets based on the percentage of zero demand in each time series as discussed in Sect. 3.1. Table 4 shows the results of mean ME with different levels of sparsity that are also

Table 3 Overall results

	Mean					Median				
	ME	MAPE	MAE	RMSE	MASE	ME	MAPE	MAE	RMSE	MASE
Croston	– 1.216	2.819	1.592	4.027	2.047	– 0.329	1.990	1.144	3.168	1.208
SBA	– 0.718	2.383	1.359	3.805	1.707	– 0.078	1.783	1.042	3.131	1.083
FNN	– 0.573	2.269	1.300	3.815	1.591	– 0.071	1.719	1.029	3.083	1.064
RNN	– 0.392	2.068	1.195	3.590	1.411	– 0.071	1.700	1.022	3.058	1.053
LSTM	– 0.505	2.172	1.248	3.630	1.515	– 0.062	1.715	1.014	3.071	1.059
Transformer	– 0.302	2.009	1.162	3.558	1.384	0.082	1.574	0.979	3.100	0.989

Table 4 Mean ME results with sparsity

Sparsity	Croston	SBA	FNN	RNN	LSTM	Transformer
1	– 0.0803	– 0.0475	– 0.0835	– 0.0665	– 0.0623	– 0.0227
2	– 0.0454	0.0089	– 0.1202	– 0.1249	– 0.0958	– 0.1324
3	– 0.0654	0.0199	– 0.0291	– 0.0389	– 0.0197	0.0442
4	– 0.1369	– 0.0394	– 0.0963	– 0.0948	– 0.0972	– 0.0743
5	– 0.1077	– 0.0144	– 0.0579	– 0.0640	– 0.0633	0.0142
6	– 1.3991	– 0.9140	– 0.4156	– 0.2146	– 0.4689	– 0.4353
7	– 0.6485	– 0.3498	– 0.2831	– 0.2133	– 0.2789	– 0.2923
8	– 1.9409	– 1.1576	– 1.0037	– 0.6874	– 0.8271	– 0.3638

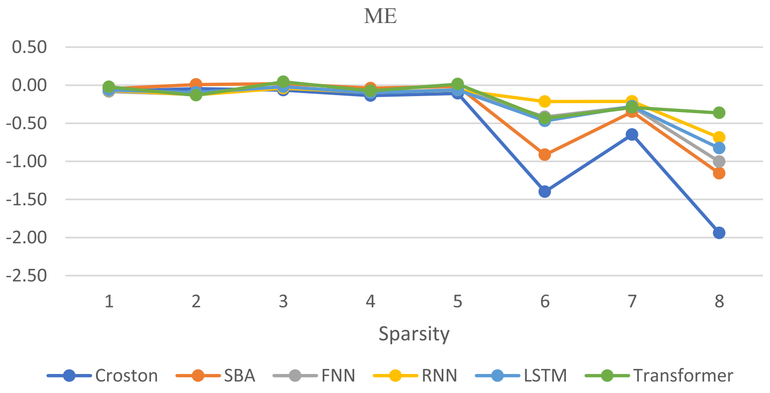

Fig. 5 Mean ME at different sparsity levels

reflected in Fig. 5. We find that mean MEs are almost all negative for all models at different sparsity levels except for SBA at the sparsity levels of 2 and 3, and Transformer at the sparsity levels of 3 and 5, again suggesting that overall, the models overestimate demand. However, at lower levels of sparsity (sparsity levels 1–5), MEs are close to zero for all models, indicating relatively low forecasting bias. As the sparsity level increases especially at high levels (from 6 to 8), the bias generally increases as well. We also find that while Transformer's overall performance in ME is among the best across all sparsity levels, Croston is clearly the worst especially at higher sparsity levels.

Tables 5, 6, 7 and 8 report results of mean MAPE, MAE, RMSE, and MASE, respectively, with different sparsity levels. The corresponding Figs. 6, 7, 8 and 9 show graphically how the performance of different models varies with sparsity. In general, we observe an upward trend across all models with each error measure as sparsity increases. This suggests that the higher the sparsity level is, the worse forecast accuracy is for every model. In other words, it becomes more difficult to forecast accurately as sparsity gets higher. We also observe that at low levels of sparsity (5 and lower), the overall performance of all models is similar while at high levels of sparsity (6 and higher), there are noticeable differences in performance among these methods with Croston being the worst and Transformer either the best or close to the best.

Table 5 Mean MAPE results with sparsity

Sparsity	Croston	SBA	FNN	RNN	LSTM	Transformer
1	0.8525	0.8449	0.8405	0.8384	0.8349	0.8247
2	0.9253	0.9112	0.9241	0.9246	0.9132	0.9212
3	0.9974	0.9803	0.9935	0.9866	0.9837	0.9664
4	1.2026	1.1730	1.1821	1.1815	1.1772	1.1957
5	1.3884	1.3502	1.3602	1.3610	1.3625	1.3476
6	2.8516	2.4389	1.9600	1.7418	1.9836	1.9789
7	2.2863	2.0505	1.9987	1.9078	1.9820	2.0136
8	3.7146	2.9890	2.9077	2.5639	2.6811	2.2735

Table 6 Mean MAE results with sparsity

Sparsity	Croston	SBA	FNN	RNN	LSTM	Transformer
1	1.0387	1.0294	1.0254	1.0223	1.0183	1.0053
2	1.0526	1.0366	1.0546	1.0551	1.0423	1.0512
3	1.0540	1.0359	1.0489	1.0412	1.0378	1.0171
4	1.0449	1.0203	1.0248	1.0239	1.0198	1.0338
5	1.0473	1.0198	1.0245	1.0249	1.0255	1.0137
6	1.6785	1.4556	1.2178	1.1030	1.2263	1.2217
7	1.3086	1.1764	1.1464	1.0969	1.1360	1.1501
8	1.9516	1.5733	1.5284	1.3521	1.4112	1.1988

Table 7 Mean RMSE results with sparsity

Sparsity	Croston	SBA	FNN	RNN	LSTM	Transformer
1	1.0759	1.0727	1.0519	1.0571	1.0564	1.0482
2	1.1817	1.1732	1.1456	1.1533	1.1592	1.1456
3	1.2791	1.2734	1.2744	1.2647	1.2691	1.2684
4	1.4760	1.4684	1.4729	1.4705	1.4643	1.4817
5	1.7673	1.7590	1.7515	1.7472	1.7459	1.7534
6	3.4206	3.0882	2.5921	2.3910	2.5580	2.5536
7	3.2607	3.1808	3.1255	3.0386	3.0815	3.1077
8	5.5928	5.2315	5.4588	5.0514	5.0622	4.8711

The percentage best (PB) results are provided in Table 9 for all SKUs and in Table 10 with different sparsity levels. For each overall error measure (in the case of ME, we use the absolute value), we identify the best method as the one associated with the lowest value of the error measure, and then the percentage best for a method is calculated as the number of times the method is the best divided by the total number of SKUs in the whole sample (Table

Table 8 Mean MASE results with sparsity

Sparsity	Croston	SBA	FNN	RNN	LSTM	Transformer
1	0.8321	0.8245	0.8189	0.8173	0.8135	0.8005
2	0.8491	0.8358	0.8447	0.8456	0.8349	0.8423
3	0.8169	0.8031	0.8127	0.8071	0.8045	0.7897
4	0.9397	0.9173	0.9192	0.9186	0.9143	0.9302
5	0.9573	0.9319	0.9369	0.9384	0.9386	0.9286
6	2.1968	1.8377	1.3239	1.1410	1.3807	1.3648
7	1.4845	1.3235	1.2600	1.1973	1.2529	1.2777
8	2.7811	2.2116	2.1464	1.8201	1.9511	1.6274

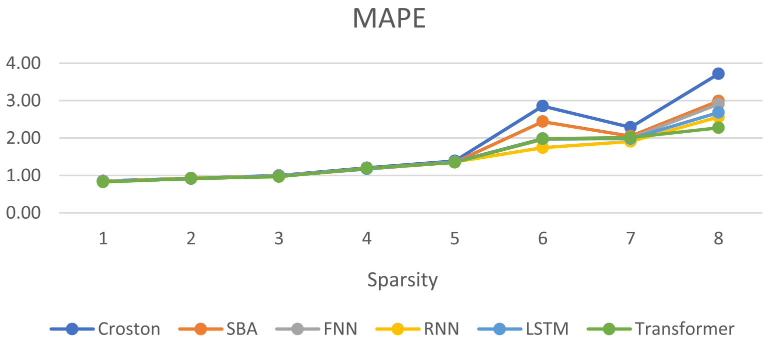

Fig. 6 Mean MAPE at different sparsity levels

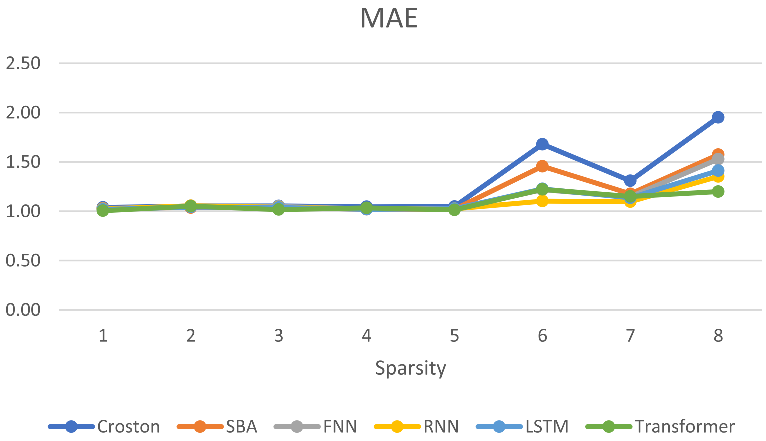

Fig. 7 Mean MAE at different sparsity levels

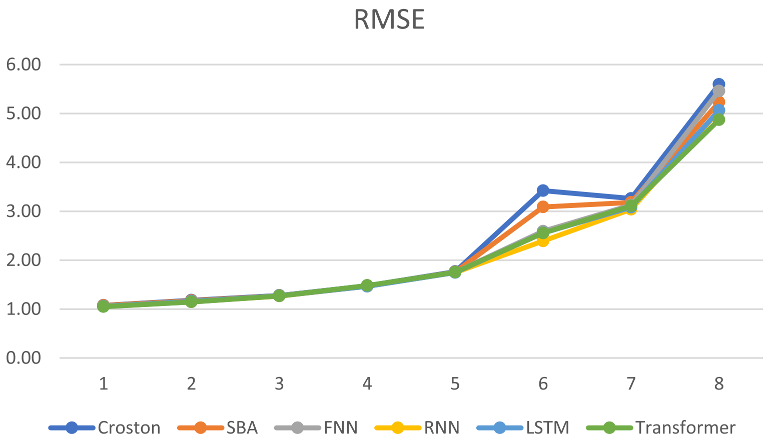

Fig. 8 Mean RMSE at different sparsity levels

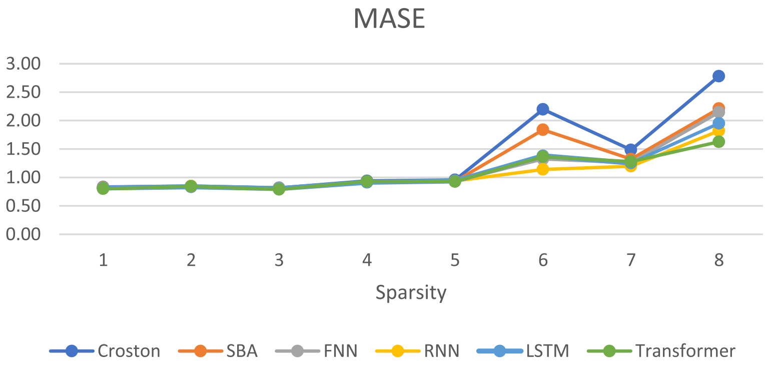

Fig. 9 Mean MASE at different sparsity levels

Table 9 Percentage best results for all SKUs

Error	Croston	SBA	FNN	RNN	LSTM	Transformer
|ME|	0.160	**0.32**	0.15	0.12	0.111	0.154
MAPE	0	0.22	0.08	0.13	0.070	**0.497**
MAE	0	0.22	0.08	0.13	0.069	**0.499**
RMSE	0.04	0.1	0.200	**0.25**	0.177	0.231
MASE	0	0.22	0.09	0.13	0.070	**0.496**

9) or at a specific sparsity level (Table 10). The best PB is in bold for each error measure. As Doszyn (2019) discussed, the same lowest error measure could be associated with multiple methods, causing the sum of percentage best across all methods not equal to one. But this does not happen often with our results.

Table 10 Percentage best results with different sparsity levels

Sparsity	Error	Croston	SBA	FNN	RNN	LSTM	Transformer
1	|ME|	0.000	0.000	0.000	0.000	0.333	**0.667**
	MAPE	0.000	0.333	0.000	0.000	0.000	**0.667**
	MAE	0.000	0.333	0.000	0.000	0.000	**0.667**
	RMSE	0.000	0.000	**0.667**	0.000	0.000	0.333
	MASE	0.000	0.333	0.000	0.000	0.000	**0.667**
2	|ME|	0.000	**0.750**	0.250	0.000	0.000	0.000
	MAPE	0.000	**0.500**	0.000	0.000	0.250	0.250
	MAE	0.000	**0.500**	0.000	0.000	0.250	0.250
	RMSE	0.000	**0.500**	0.250	0.000	0.000	0.250
	MASE	0.000	**0.500**	0.000	0.000	0.250	0.250
3	|ME|	**0.429**	0.286	0.000	0.071	0.000	0.214
	MAPE	0.000	0.143	0.143	0.071	0.143	**0.500**
	MAE	0.000	0.143	0.143	0.071	0.143	**0.500**
	RMSE	0.071	0.071	0.071	**0.357**	0.143	0.286
	MASE	0.000	0.143	0.143	0.071	0.143	**0.500**
4	|ME|	0.250	**0.344**	0.094	0.125	0.094	0.094
	MAPE	0.000	**0.281**	0.125	0.156	0.219	0.219
	MAE	0.000	**0.281**	0.125	0.156	0.219	0.219
	RMSE	0.125	0.125	0.188	0.188	**0.219**	0.156
	MASE	0.000	**0.281**	0.125	0.156	0.219	0.219
5	|ME|	0.231	**0.423**	0.064	0.077	0.038	0.167
	MAPE	0.000	0.244	0.128	0.077	0.077	**0.474**
	MAE	0.000	0.244	0.128	0.077	0.077	**0.474**
	RMSE	0.077	0.128	0.179	0.167	0.192	**0.256**
	MASE	0.000	0.256	0.128	0.077	0.077	**0.462**
6	|ME|	0.238	**0.325**	0.095	0.119	0.087	0.135
	MAPE	0.000	0.294	0.119	0.143	0.040	**0.405**
	MAE	0.000	0.294	0.119	0.143	0.040	**0.405**
	RMSE	0.056	0.087	0.183	0.183	0.206	**0.286**
	MASE	0.000	0.286	0.119	0.151	0.040	**0.405**
7	|ME|	0.143	**0.368**	0.146	0.129	0.093	0.121
	MAPE	0.004	0.243	0.082	0.154	0.064	**0.454**
	MAE	0.004	0.239	0.086	0.154	0.064	**0.454**
	RMSE	0.029	0.089	0.207	**0.300**	0.182	0.193
	MASE	0.004	0.243	0.086	0.154	0.064	**0.450**
8	|ME|	0.119	**0.263**	0.162	0.124	0.152	0.180
	MAPE	0.000	0.168	0.062	0.116	0.067	**0.588**
	MAE	0.000	0.165	0.062	0.116	0.064	**0.593**
	RMSE	0.036	0.106	0.206	**0.250**	0.162	0.240
	MASE	0.000	0.165	0.062	0.116	0.067	**0.590**

The PB results in Table 9 show that Transformer is the best method in terms of MAPE, MAE, and MASE. Based on each of these measures, Transformer performs the best for nearly 50% SKUs, which is much higher than that for any other method considered in this study. It is also the 2nd best approach with regard to RMSE with PB = 23.1%. The overall best method based on RMSE is RNN with PB = 24.6%. The best method based on |ME| is SBA (PB = 32.1%) while Transformer is tied with the 3rd best (PB = 15.4%). Table 9 also shows that Croston, FNN, and LSTM are among the worst judging from PB across different error measures.

Considering different sparsity levels, Table 10 suggests that Transformer is the overall best method especially at higher sparsity levels. For example, at sparsity levels 5 and 6, PB for Transformer is the highest with respect to all error measures except for |ME|. At sparsity levels 7 and 8, Transformer significantly outperforms other methods on three error measures with dominant PB. Surprisingly, Transformer also performs very well at several lower levels of sparsity (levels 1 and 3). On the other hand, we find that SBA is the best performer at sparsity levels of 2 and 4 based on almost all error measures. In fact, SBA performs well at other sparsity levels as its PB is often the 2nd highest among all methods considered.

Overall, we find that Transformer performs consistently well judged by both the parametric error measures and the nonparametric percentage best measure. This finding is different from that obtained by Doszyn (2019) who finds that the best methods by means of parametric measures (i.e., error measures) are different than those using the nonparametric measure (i.e., PB).

5 Conclusions

Intermittent demand forecasting is an important and challenging problem facing many organizations. As a result, there is an increasing research effort in searching for better forecasting systems for intermittent demand items. While most existing research has focused on traditional methods such as Croston' method and its variants, machine learning based methods have been proposed for intermittent demand forecasting in recently years.

This study proposes the use of Transformer, a newly developed deep learning method, for intermittent demand forecasting. In addition to Transformer, we also consider five other methods (Croston, SBA, feedforward neural networks, recurrent neural networks, and long short-term memory networks). Our empirical setting is a dataset of 925 intermittent demand items from an airline service parts provider. Six forecasting performance measures are employed, including the mean error, the mean absolute percentage error, the mean absolute error, the root mean squared error, the mean absolute scaled error, and the percentage best. We also consider both mean and median results which are similar, suggesting that outliers are not a major concern in our results.

The results show that overall, Transformer performs very well relative to all other methods with different performance measures. We also find the at low sparsity levels, all methods perform similarly in terms of forecast error measures (ME, MAE, MAPE, RMSE, and MASE). As sparsity increases, it becomes more difficult to make accurate forecasts, resulting in higher error measures for all methods including Transformer. However, Transformer's performance advantages over the Croston method, SBA, and various conventional neural network models (FNN, RNN, and LSTM) increase with the sparsity level. Using the measure of percentage best, we find that Transformer once again the overall best performer, although SBA performs well too especially at low sparsity levels.

Our findings confirm prior research that no forecasting method is the universal best under all circumstances. However, one method could be an overall better one for some forecasting settings. This research shows the promise of Transformer in forecasting intermittent demand. One of the limitations of this research is that we have only analyzed 925 time series from the same industry. In addition, the data we used are classified as "intermittent" based on Syntetos et al. (2005) classification scheme. Future research may extend this study by applying Transformer and other deep learning methods to different intermittent demand items in different industries and with different characteristics. Finally, this study shows how sparsity impacts the model performance. While sparsity is an important characteristic of intermittent demand, capturing the average intermittence in a time series, another important dimension is the variance of intermittence, which is a measure of how zero demand periods are dispersed. Studying how the variance of intermittence impacts the model performance could be an interesting future research direction.

Funding The authors have no relevant financial or non-financial interests to disclose.

References

Babai, M., Dallery, Y., Boubaker, S., & Kalai, R. (2019). A new method to forecast intermittent demand in the presence of inventory obsolescence. *International Journal of Production Economics, 209*, 30–41.

Babai, M. Z., Syntetos, A. A., & Teunter, R. (2014). Intermittent demand forecasting: An empirical study on accuracy and the risk of obsolescence. *International Journal of Production Economics, 157*, 212–219.

Babai, M. Z., Tsadiras, A., & Papadopoulos, C. (2020). On the empirical performance of some new neural network methods for forecasting intermittent demand. *IMA Journal of Management Mathematics, 31*, 281–305.

Bahdanau, D., Cho, K., and Bengio, Y., (2015). Neural machine translation by jointly learning to align and translate. In: Proceedings of international conference on learning representations

Boylan, J. E., & Babai, M. Z. (2016). On the performance of overlapping and non-overlapping temporal demand aggregation approaches. *International Journal of Production Economics, 181*, 136–144.

Boylan, J. E., & Syntetos, A. A. (2007). The accuracy of a modified Croston procedure. *International Journal of Production Economics, 107*, 511–517.

Boylan, J. E., & Syntetos, A. A. (2021). *Intermittent demand forecasting: Context, methods, and applications*. Wiley.

Croston, J. D. (1972). Forecasting and stock control for intermittent demand. *Operational Research Quarterly, 23*, 289–303.

Doszyń, M. (2019). Intermittent demand forecasting in the enterprise: Empirical verification. *Journal of Forecasting, 38*, 459–469.

Eaves, A. H. C., & Kingsman, B. G. (2004). Forecasting for the ordering and stockholding of spare parts. *Journal of the Operational Research Society, 55*(4), 431–437.

Gutierrez, R. S., Solis, A. O., & Mukhopadhyay, S. (2008). Lumpy demand forecasting using neural networks. *International Journal of Production Economics, 111*, 409–420.

Hasni, M., Aguir, M., Babai, M., & Jemai, Z. (2019). On the performance of adjusted bootstrapping methods for intermittent demand forecasting. *International Journal of Production Economics, 216*, 145–153.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Hu, Q., Boylan, J. E., Chen, H., & Labib, A. (2018). OR in spare parts management: A review. *European Journal of Operational Research, 266*, 395–414.

Hua, Z. S., Zhang, B., Yang, J., & Tan, D. S. (2007). A new approach of forecasting intermittent demand for spare parts inventories in the process industries. *Journal of Operational Research Society, 58*, 52–61.

Hua, Z., & Zhang, B. (2006). A hybrid support vector machines and logistic regression approach for forecasting intermittent demand of spare parts. *Applied Mathematics and Computation, 181*(2), 1035–1048.

Johnston, F. R., Boylan, J. E., & Shale, E. (2003). An examination of the size of orders from customers, their characterisation and the implications for inventory control of slow moving items. *Journal of the Operational Research Society, 54*, 833–837.

Kingma, D. P. and Ba, J. (2017). Adam: A method of stochastic optimization. arXiv:1412.6980.

Kourentzes, N. (2013). Intermittent demand forecasts with neural networks. *International Journal of Production Economics, 143*(1), 198–206.

Levén, E., & Segerstedt, A. (2004). Inventory control with a modified Croston procedure and Erlang distribution. *International Journal of Production Economics, 90*, 361–367.

Lolli, F., Gamberini, R., Regattieri, A., Balugani, E., Gatos, T., & Gucci, S. (2017). Single-hidden layer neural networks for forecasting intermittent demand. *International Journal of Production Economics, 183*, 116–128.

Mukhopadhyay, S., Solis, A. O., & Gutierrez, R. S. (2012). The accuracy of non-traditional versus traditional methods of forecasting lumpy demand. *Journal of Forecasting, 31*, 721–735.

Nasiri Pour, A., Rostami Tabar, B., & Rahimzadeh, A. (2008). A hybrid neural network and traditional approach for forecasting lumpy demand. *Engineering and Technology, 40*, 384–390.

Nikolopoulos, K. (2021). We need to talk about intermittent demand forecasting. *European Journal of Operational Research, 291*, 549–559.

Nikolopoulos, K. I., Babai, M. Z., & Bozos, K. (2016). Forecasting supply chain sporadic demand with nearest neighbor approaches. *International Journal of Production Economics, 177*, 139–148.

Nikolopoulos, K., Syntetos, A. A., Boylan, J. E., Petropoulos, F., & Assimakopoulos, V. (2011). An aggregate–disaggregate intermittent demand approach (ADIDA) to forecasting: An empirical proposition and analysis. *Journal of the Operational Research Society, 62*(3), 544–554.

Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing, 452*, 48–62.

Petropoulos, F., Kourentzes, N., & Nikolopoulos, K. (2016). Another look at estimators for intermittent demand. *International Journal of Production Economics, 181*, 154–161.

Pince, C., Turrini, L., & Meissner, J. (2021). Intermittent demand forecasting for spare parts: A critical review. *Omega, 105*, 102513.

Prestwich, S. D., Tarim, S. A., Rossi, R., & Hnich, B. (2014). Forecasting intermittent demand by hyperbolic-exponential smoothing. *International Journal of Forecasting, 30*, 928–933.

Shale, E. A., Boylan, J. E., & Johnston, F. R. (2006). Forecasting for intermittent demand: The estimation of an unbiased average. *Journal of Operational Research Society, 57*, 588–592.

Syntetos, A. A., Babai, M. Z., & Gardner, E. S., Jr. (2015). Forecasting intermittent inventory demands: Simple parametric methods vs. bootstrapping. *Journal of Business Research, 68*, 1746–1752.

Syntetos, A. A., Babai, Z., Boylan, J. E., Kolassa, S., & Nikolopoulos, K. (2016). Supply chain forecasting: Theory, practice, their gap and the future. *European Journal of Operational Research, 252*, 1–26.

Syntetos, A. A., & Boylan, J. E. (2001). On the bias of intermittent demand estimates. *International Journal of Production Economics, 71*, 457–466.

Syntetos, A. A., & Boylan, J. E. (2005). The accuracy of intermittent demand estimates. *International Journal of Forecasting, 21*, 303–314.

Teunter, R., & Sani, B. (2009). On the bias of Croston's forecasting method. *European Journal of Operational Research, 194*(1), 177–183.

Teunter, R. H., Syntetos, A. A., & Babai, M. Z. (2011). Intermittent demand: Linking forecasting to inventory obsolescence. *European Journal of Operational Research, 214*(3), 606–615.

Vaswani, A. Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), Advances in neural information processing systems 30, pp. 5998–6008.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L., (2022). Transformers in time series: A survey. arXiv. arXiv:2202.07125.

Willemain, T. R., Smart, C. N., & Schwarz, H. F. (2004). A new approach to forecasting intermittent demand for service parts inventories. *International Journal of Forecasting, 20*, 375–387.

Yang, Y., Ding, C., Lee, S., Yu, L., & Ma, F. (2021). A modified Teunter-Syntetos-Babai method for intermittent demand forecasting. *Journal of Management Science and Engineering, 6*(1), 53–63.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting, 14*(1), 35–62.

Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2021). Dive into deep learning, arXiv:2106.11342.

Zhou, C. X., & Viswanathan, S. (2011). Comparison of a new bootstrapping method with parametric approaches for safety stock determination in service parts inventory systems. *International Journal of Production Economics, 133*(1), 481–485.

Zhu, S., Dekker, R., van Jaarsveld, W., Renjie, R. W., & Koning, A. J. (2017). An improved method for forecasting spare parts demand using extreme value theory. *European Journal of Operational Research, 261*(1), 169–181.

Zhuang, Z., Yu, Y., & Chen, A. (2022). A combined forecasting method for intermittent demand using the automotive aftermarket data. *Data Science and Management, 5*(2), 43–56.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.