Artificial Intelligence in Supply Chain Management: Investigation of Transfer Learning to Improve Demand Forecasting of Intermittent Time Series with Deep Learning

Daniel Kiefer
ESB Business School,
Reutlingen University
Daniel.Kiefer@
Reutlingen-University.de

Florian Grimm
ESB Business School,
Reutlingen University
Florian.Grimm@
Reutlingen-University.de

Clemens van Dinther
ESB Business School,
Reutlingen University
Clemens.Van_Dinther@
Reutlingen-University.de

Abstract

Demand forecasting intermittent time series is a challenging business problem. Companies have difficulties in forecasting this particular form of demand pattern. On the one hand, it is characterized by many non-demand periods and therefore classical statistical forecasting algorithms, such as ARIMA, only work to a limited extent. On the other hand, companies often cannot meet the requirements for good forecasting models, such as providing sufficient training data. The recent major advances of artificial intelligence in applications are largely based on transfer learning. In this paper, we investigate whether this method, originating from computer vision, can improve the forecasting quality of intermittent demand time series using deep learning models. Our empirical results show that, in total, transfer learning can reduce the mean square error by 65 percent. We also show that especially short (65 percent reduction) and medium long (91 percent reduction) time series benefit from this approach.

1. Introduction

The latest developments in the field of artificial intelligence, specifically deep learning models and their subsequent applications, are highly impressive. For example, in certain areas cars drive fully autonomously [1] and artificial intelligence can generate human-like text [2]. Advances in autonomous driving and natural human-like text generators, such as the generative pre-trained transformer 3 (GPT-3), are based on deep learning architectures [2, 3].

These latest improvements and applications in artificial intelligence were made possible, in part, by applying transfer learning [4]. In its simplest form, a deep learning model is pre-trained on a data set D_S and then fine-tuned on the target data set D_T.

Modern deep learning architectures are now being successfully applied in areas such as healthcare [5], energy [6], financial markets [7], production [8] and logistics [9]. The predictions derived from these models often serve as a basis for human decision-making, for example in the area of demand forecasting for spare parts to support a company's purchasing team [10].

Artificial intelligence is also gradually being implemented in supply chain management [9]. Companies can gain significant competitive advantages in production, procurement, and logistics through more accurate demand forecasts [11]. However, the use of neural networks, for example, requires good data quality on the one hand, and a sufficient quantity of data on the other [12]. Especially the latter requirement is difficult for small and medium-sized companies to fulfill, because they often start systematically collecting data later than large companies do [13]. For newly introduced products, where there is not yet a sufficiently long product history, the amount of available data is also often not sufficient.

According to Syntetos et al. [14], demand time series, i.e., the temporal sequence of demand for a product, can be divided into the categories erratic, smooth, lumpy and intermittent, as shown in Figure 1. Lumpy and intermittent demand time series are particularly difficult to predict because they are characterized by many zero periods. This not only limits the use of measurement metrics, such as mean absolute percentage error (MAPE), but also of algorithms, such as some variants of holt-winters and auto-regressive integrated moving average (ARIMA) models.

Machine learning and deep learning methods are therefore of particular interest for the prediction of these complex time series. For example, a hybrid method of exponential smoothing (ES) and a recurrent neural

URI: https://hdl.handle.net/10125/79537
978-0-9981331-5-7
(CC BY-NC-ND 4.0)

network (RNN) won the M4-Competition[1]. In the M5-Competition which included intermittent time series, a light gradient boosting machine (L-GBM) achieved the best overall result. Nevertheless, it is not clear whether artificial intelligence methods always produce better results in practice compared to statistical methods, for example, Kourentzes [15] and Kiefer et al. [16] showed that artificial intelligence methods did not necessarily always produce the best results. Nikolopoulos [17] also highlights the existing research gap in the field of intermittent and lumpy demand forecasting. At the same time, methods in information systems are rapidly advancing. Hence, new developments in the field of deep learning, such as transfer learning, should also be considered.

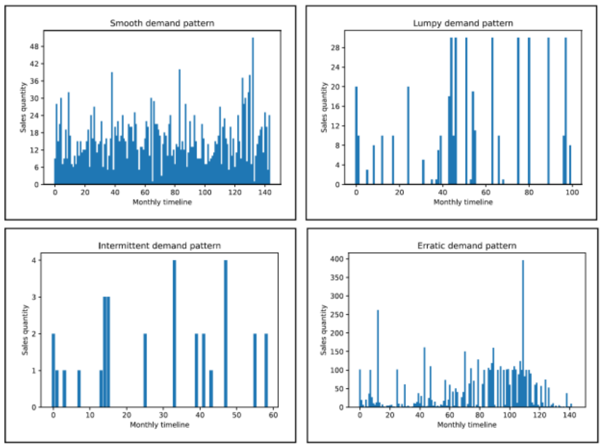

Figure 1. Demand patterns according to Syntetos et al. [14]

So far, the transfer learning method has not been extensively investigated or applied in the field of intermittent time series demand forecasting. There are, however, some first experiences in time series anomaly detection [15] and in forecasting of financial markets [16]. Therefore, this paper investigates whether the technique transfer learning, originating from computer vision, can be successfully applied to the specific problem of intermittent time series demand forecasting using deep learning methods, in accordance with the design science research (DSR) [17]. Based on the identified research gap in the following Chapter 2, the following research questions were derived:

RQ 1: Can the method of transfer learning, from the computer vision domain, improve the forecasting quality in the field of intermittent time series demand forecasting?

RQ 2: Can especially short time series with few data points benefit from transfer learning?

RQ 3: Do time series with a long history benefit from transfer learning?

The remainder of this paper is structured along seven Chapters. In Chapter 2, we explore existing literature about transfer learning as well as first attempts to use it in domains other than computer vision and natural language processing. Furthermore, the existing research gap regarding transfer learning for demand forecasting intermittent time series is highlighted. Chapter 3 addresses the experimental design of this article to deliver answers to the identified research gap and the formulated research questions. In Chapter 4, the results of the benchmark and the transfer learning experiments are analyzed using a real-world data set. Special attention is devoted to the results of both short and long time series to gain further insights about when transfer learning is possibly useful. Finally, we provide a conclusion in Chapter 5. The references cited are shown in Chapter 6, while in Chapter 7, the appendix, the variables used in the deep learning architectures are provided in tabular form.

2. Transfer Learning and Related Work

The presented research work is related to areas of deep learning, transfer learning, and demand forecasting of intermittent time series. The following chapter explains the subject matter in more depth where necessary and refers to the relevant literature where a detailed explanation is not required.

Over the past decades until now, along with the development of mathematics and machine learning theories, many algorithms with good performance for time series forecasting have been proposed, including ES [18], ARIMA [19] model, gradient boosting machines (GBM) [20], neural networks (NNs) [21], long short term memory (LSTM) [22], gated recurrent unit (GRU) [23], and several others [24].

However, linear models such as ARIMA, for example, can only handle linear and stationary time series data. For nonlinear and nonstationary data, practitioners attempt to convert them into smooth time series data to obtain relatively useful prediction results, e.g., in the ARIMA model. As time series data in real-world applications are often nonlinear and nonstationary, traditional linear prediction techniques have difficulty adapting to these situations. Difference processing can partially achieve stationary time series

[1] The makridakis competitions are a series of open competitions organized by Spyros Makridakis to evaluate and compare the accuracy of different forecasting methods.

data, which is usually insufficient to fully express the changes in time series data over time. At the same time, difference processing cannot be performed if the time series has missing or incorrect data have been introduced [24].

Empirical studies show that nonlinear models usually have better, and more reliable, performance compared to linear algorithms [24]. Therefore, in the subsequent work, we deliberately focus on deep learning methods in the form of artificial neural networks since these can recognize non-linear cause-effect relationships. In the context of demand forecasting intermittent time series with deep learning methods, [25] and [26] have already achieved reliable results.

Two important assumptions for traditional demand forecasting time series exist, namely [24]:

i. The training data and testing data should come from the same feature space and follow the same probability distribution.
ii. Sufficient training samples must be available to learn a good prediction model.

In many practical application scenarios, the time series often change over time leading to a significant discrepancy between new and old data. At the same time, in real-world applications, the amount of available data is rather small, resulting in an insufficient amount of training data. In these cases, the two important assumptions cannot be met [24].

In transfer learning, on the other hand, the domains, tasks, and distributions used in training and testing can be different. In particular, when there is limited data in the target data set, transfer learning can usually achieve good improvements and it is therefore a common method in deep learning applications [27].

In the real world, we can observe examples of transfer learning. For example, learning to play an electric guitar can facilitate learning to play drums. Research on transfer learning is motivated by the idea that people can intelligently apply previously learned knowledge to solve new challenges faster or with better solutions. The fundamental for transfer learning in the field of machine learning was set in a NIPS-95 workshop on “Learning to Learn” [27].

Transfer learning is used to improve a learner from one domain by transferring information from a related domain. Figure 2 visualizes the difference between standard domain learning tasks and transfer learning.

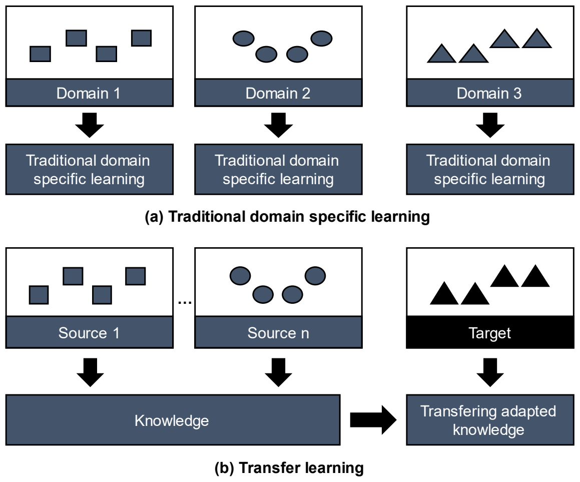
Domain 1
Domain 2
Domain 3
Traditional domain specific learning
Traditional domain specific learning
Traditional domain specific learning
(a) Traditional domain specific learning
Source 1
...
Source n
Target
Knowledge
Transfering adapted knowledge
(b) Transfer learning

Figure 2. Comparison of the traditional domain specific learning and transfer learning approach [28]

A simple explanation regarding the primary function of transfer learning and notation for this work follows, based on Pan [27].

Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T(\bullet)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$. Further and more detailed explanations are provided in the study of transfer learning by Pan [27].

The latest developments and applications – and their successful results in the field of artificial intelligence – are based on transfer learning [4]. Particularly noteworthy are recent developments in the areas of computer vision and natural language processing. Transfer learning has already been successfully applied in the context of time series analysis, including anomaly detection [15] and forecasting of financial markets [16], which correspond to smooth or erratic time series.

In the area of intermittent time series demand forecasting, we could not find any research results. Therefore, in this paper we focus on investigating whether and how the transfer learning method can improve intermittent time series demand forecasting. Of particular interest is the behavior with different lengths, respectively amounts, of data of the target time series to be forecasted.

3. Suggested Experimental Design

To answer the research questions and expand on existing investigations, we propose an experimental design (Figure 3) that is adapted to the shortcomings mentioned in the previous chapters.

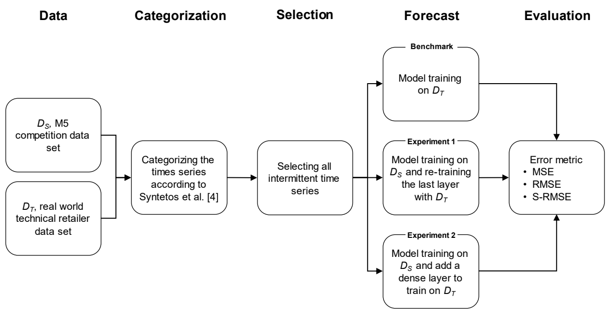
Data
Categorization
Selection
Forecast
Evaluation
D_S, M5 competition data set
D_T, real world technical retailer data set
Categorizing the times series according to Syntetos et al. [4]
Selecting all intermittent time series
Benchmark
Model training on D_T
Experiment 1
Model training on D_S and re-training the last layer with D_T
Experiment 2
Model training on D_S and add a dense layer to train on D_T
Error metric
• MSE
• RMSE
• S-RMSE

Figure 3. Suggested experimental design

We have two data sets available to review the research questions posed.

D_S corresponds to the public M5 data set, which contains mainly intermittent and lumpy time series. The data is provided by Walmart and includes 30,490 SKU-level hierarchical daily time series with a length of 1,941 time steps for each series [29].

D_T is provided by a technical business-to-business distributor and includes approximately 8,782 hierarchical daily time series at the product level with varying lengths from 1 to 3,960 time steps for each time series.

These time series were categorized using the approach described by [14]. Based on the calculated average demand interval (ADI) and the squared coefficient of variation (CV^2), the time series were categorized according to their demand behavior. The ADI represents the average demand interval between two successive demands. This metric is a measure of intermittency; the higher it is, the more intermittent the time series.

$$ADI = \frac{Quantity\ of\ time\ periods}{Quantity\ of\ non\ zero\ demand\ periods} \quad (1)$$

CV^2 describes the magnitude of demand variability in a time series [14]. If the value is high, this indicates that the demand variability in the series is also high.

$$CV^2 = \left(\frac{Demand\ Standard\ Deviation}{Demand\ Mean}\right)^2 \quad (2)$$

In Figure 4, it can be seen that the distribution of the coefficient of variation from the M5 data set is slightly different to the distribution of the real data set of the technical trader. The median coefficient of variation value for M5 is 0.32 and for the real-world data set it is 0.21, which means that the demand in the M5 data is more variating than the demand from the real-world data set.

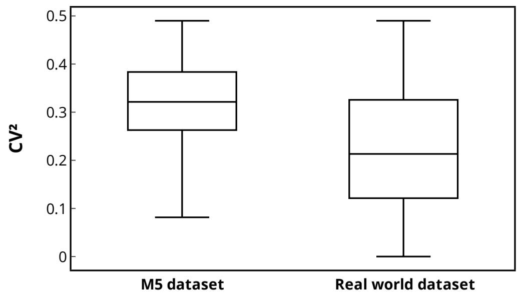

Figure 4. Distribution of the coefficient of variation

Figure 5 shows that the distribution of the average demand interval of the M5 data set is different to the distribution of the real data set of the technical trader. The median ADI value for M5 is 1.60 and for the real-world data set it is 9.30, which means that the data from M5 are less intermittent than those from the real-world data set.

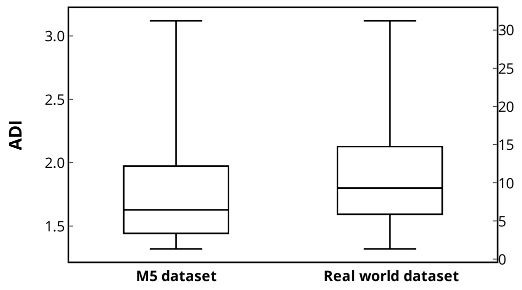

Figure 5. Distribution of the average demand interval

For the cut-off values, we adopted the values proposed by [14]. Figure 6 shows the cut-off values and the resulting classes. The corresponding demand categories are: erratic (but not very intermittent), lumpy, smooth and intermittent (but not very erratic) [14].

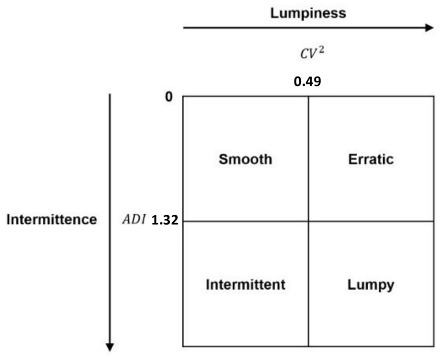

Figure 6. Demand pattern categorization scheme [14]

By categorizing the M5 time series according to [14], 10,095 intermittent time series could be identified, which were used as D_S for our transfer learning model.

Within the real data set of the technical trader 3,470 intermittent time series could be identified, which is the target domain D_T to predict.

The time series are present on a daily basis and are aggregated on a weekly basis. The aim of the forecast is to predict the next period, i.e., week. For this purpose, the last 52 weeks serve as input for the respective deep learning models. The input sequences are scaled with the MinMaxScaler (0-1) [30].

Of the target data set D_T, 80 percent of the data is used to train the models, and the remaining 20 percent is used to evaluate the forecasts.

The underlying benchmark deep learning model is a neural network based on 10 fully connected dense layers with a total of 953 neurons. More detailed specification of the architecture (Figure 7) and the hyperparameters (Table 5) can be found in Chapter 7. The model is trained univariately, i.e. on each time series individually with 80 percent of the data of this time series. The function $f_T(\bullet)$ resulting from the learning process is then used to predict the remaining 20 percent of the data on a rolling basis. For this, an input sequence of the last 52 weeks is used to predict the next week and so on. This is done for all time steps to be predicted in the time series.

The transfer learning models are almost identical to the benchmark model. There are in total two experiments.

In experiment 1, the benchmark model is trained on the source data set D_S before it is fine-tuned on the target data set D_T. The resulting model or learned content from D_S in the form of weights within layers and neurons is "frozen" to ensure this knowledge. The last layer of this architecture is removed and added again without the stored weights. This allows the model to be fine-tuned respectively trained on the target data set D_T with 80 percent of this data in the next step. The remaining 20 percent are predicted and evaluated as described above.

Experiment 2 basically works like experiment 1 and is also based on the architecture of the benchmark neural network. While in the previous experiment 1 the last layer is removed to allow fine tuning in this layer, in experiment 2 the complete model trained on D_S is used with the "frozen" weights of all layers. However, an additional dense layer, similar to the last corresponding layer of the architecture, is added to this model. In this specific layer the fine adjustment on the target data set D_T takes place. In simplified terms, this does not remove any previously learned knowledge from D_S.

In the following, we present the error metrics used to evaluate the predictions on the out-of-sample area of the data.

To assess the forecast quality, the mean squared error (MSE) is a commonly used metric in comparing time series models. Its nonnegativity as well as its symmetry are highly valuable features [31].

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (X_i - Y_i)^2 \qquad (3)$$

with:

- X_i, predicted target value at time step i
- Y_i, true target value at time step i
- n, quantity of predicted time steps

While the MSE has a symmetry, the root mean squared error (RMSE) evaluates larger errors stronger than smaller ones making it more sensitive to outliers and penalizing them [32]. This property must be kept in mind and can be beneficial as well as negative. Therefore, this metric is used in a context-specific manner. In the area of demand forecasting, it can be useful to use this metric to evaluate the models as well, since large errors in the forecast lead to major economic losses, for example in form of excessive inventory.

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(X_i - Y_i)^2}{n}} \qquad (4)$$

with:

- X_i, predicted target value at time step i
- Y_i, true target value at time step i
- n, quantity of predicted time steps

Since both of the previously mentioned metrics are not scaled, it is difficult to compare the metrics from time series to time series and across multiple models. Kolossa and Siemsen [33] suggest for intermittent time series, to scale the RMSE by the series overall mean of the test set to obtain a scaled error measure that is comparable between time series. In the following we use the notation for this metric S-RMSE.

$$\mathrm{S-RMSE} = \frac{RMSE}{\frac{1}{n} \sum_{i=1}^{n} Y_i} \qquad (5)$$

with:

- Y_i, true target value at time step i
- n, quantity of predicted time steps

4. Results

In the following, the forecasting results of the benchmark model and the two transfer learning experiments are presented. We only report out of sample (i.e., test set) results, because superior results on the in-sample (i.e., training set) can be misleading regarding overfitting.

Overall: Table 1 shows the results of the benchmark deep learning model and the experiments with transfer learning. Considering the Ø MSE, both experiments outperformed the benchmark. The evaluation metric was reduced by 65 percent. Regarding the Ø RMSE, which is more sensitive to outliers, experiment 2 could improve the forecasting result by 23 percent. The proposed metric for the evaluation of intermittent demand forecasting by [33], the Ø S-RMSE, again shows that experiment 2 achieved the best result. The benchmark model without transfer learning has a 10 percent higher value. Looking at experiment 1, and considering the Ø S-RMSE, it achieved a slightly worse result than the benchmark model. In the other metrics, the experiment 1 is clearly better than the benchmark.

Table 1. Results of all time series of D_T

	Ø MSE		**Ø RMSE**		**Ø S-RMSE**	
Benchmark	*5.89*	-	*0.31*	-	3.77	-
Exp. 1	2.08	(-65%)	0.25	(-18%)	*3.85*	(+2%)
Exp. 2	**2.07**	**(-65%)**	**0.24**	**(-23%)**	**3.40**	**(-10%)**

In 2,708 time series, the experiment 2 model achieved a better prediction result across all metrics than the benchmark model did without transfer learning. This corresponds to about 78 percent. In 762 time series, the benchmark achieved a better result than the transfer learning experiment 2, which corresponds to approximately 22 percent.

Short time series: Table 2 shows the results of the benchmark deep learning model and the experiments with transfer learning on short time series. Short time series are defined in cases where the model can only learn from a history of less than 2 years of data. Overall, the results strongly support the findings from literature that transfer learning can achieve strong improvements when few data points are available. Regarding the Ø S-RMSE, both experiments could improve the forecast by 71 percent. Considering the Ø MSE, both experiments reduced this error metric by 65 percent, while on the Ø RMSE, the experiments reduced the error by roughly 55 percent.

Table 2. Results of short time[2] series of D_T

	Ø MSE		**Ø RMSE**		**Ø S-RMSE**	
Benchmark	*171.36*	-	*2.80*	-	*6.66*	-
Exp. 1	60.54	(-65%)	**1.24**	**(-56%)**	1.96	(-71%)
Exp. 2	**60.53**	**(-65%)**	1.25	(-55%)	**1.92**	**(-71%)**

[2] short time series <= 2 years of data

In 86 time series, the experiment 2 model achieved a better prediction result across all metrics than the benchmark model without transfer learning, which corresponds to about 75 percent. In 28 time series, the benchmark achieved a better result than the transfer learning experiment 2, which corresponds to about 25 percent.

Long time series: The results of the benchmark deep learning model and the experiments with transfer learning on long time series are shown in Table 3. Long time series are defined in cases where the model can learn from a history of more than 5 years of data. Experiment 2 again outperformed the benchmark on the Ø MSE, with an 11 percent better result. The gain of prediction accuracy from the experiment 2 on the long time series was considerably smaller than on the short time series.

Regarding the Ø RMSE, which is more sensitive to outliers, experiment 2 could improve the forecasting result by 5 percent. The Ø S-RMSE, shows that the experiment 2 again achieved the best result. The benchmark model without transfer learning has a 4 percent higher error metric value. Looking at experiment 1, it becomes evident that on long time series it cannot gain in prediction accuracy against the overall benchmark.

Table 3. Results of long[3] time series of D_T

	Ø MSE		**Ø RMSE**		**Ø S-RMSE**	
Benchmark	0.09	-	*0.20*	-	3.75	-
Exp. 1	*0.10*	*(+8%)*	*0.20*	*(+4%)*	*4.05*	*(+8%)*
Exp. 2	**0.08**	**(-11%)**	**0.19**	**(-5%)**	**3.61**	**(-4%)**

[3] long time series > 5 years of data

In 2,103 time series, the experiment 2 model achieved a better prediction result across all metrics than did the benchmark model without transfer learning. This corresponds to about 78 percent. In 598 time series, the benchmark achieved a better result than the transfer learning experiment 2, which corresponds to about 22 percent.

Medium time series: Table 4 shows the results of the benchmark deep learning model and the experiments with transfer learning on medium time series. Medium time series are characterized by a data history that is longer than 2 years but shorter than 5 years. Considering the Ø MSE, both experiments considerably outperformed the benchmark, with approximately 90 percent. Regarding the strong outlier sensitive metric Ø RMSE, transfer learning experiment 1 reduced the error by 17 percent and experiment 2 by 21 percent. On Ø S-RMSE, experiment 1 had a slightly higher error metric than the benchmark model. However, experiment 2 reduced this error by 15 percent.

Table 4. Results of medium time[4] series of D_T

	Ø MSE		Ø RMSE		Ø S-RMSE	
Benchmark	*1.04*	-	*0.34*	-	3.33	-
Exp. 1	0.11	(-90%)	0.28	(-17%)	*3.38*	(+2%)
Exp. 2	**0.09**	**(-91%)**	***0.26***	**(-21%)**	**2.82**	**(-15%)**

[4] medium time series 2 < x <= 5 years of data

In 519 time series, the experiment 2 model achieved a better prediction result across all metrics than the benchmark model without transfer learning. This corresponds to about 79 percent. In 136 time series, the benchmark achieved a better result than the transfer learning experiment 2. This corresponds to about 21 percent.

Based on the presented results, it is evident that deep learning methods benefit from the transfer learning approach for forecasting demand of intermittent time series. Especially on short and on medium time series, the transfer learning experiments, primarily experiment 2, considerably outperformed the benchmark deep learning architecture without transfer learning.

5. Conclusion

According to the current state of research it is unclear if transfer learning, which is a technique to improve deep learning models from the computer vision domain, also improves results on tabular data such as demand forecasting intermittent time series. Research on transfer learning is mostly conducted in the domains of computer vision and natural language processing. Time series analysis and especially forecasting demand transfer learning research is rare, even though almost every company works with forecasts and better forecasts are a competitive advantage.

One main contribution of this work is the analysis of the transfer learning approach in combination with deep learning methods to forecast demand of intermittent time series. To evaluate the performance, the MSE, the RMSE and the S-RMSE were used. The same deep learning architecture, but without the transfer learning methodology, was used as a benchmark. In total, 3,470 intermittent time series of a technical retailer *(D_T)* were forecast and evaluated on the test set. To deliver more insights about the behavior of the transfer learning experiments, the time series were divided into short, medium, and long time series.

Using this approach, it was possible to examine the results in more detail and to make statements about the gained improvement of the transfer learning methods depending on the data length. The M5 competition data set was used as domain source D_S to improve the target predictive function $f_T(\bullet)$ in D_T using the knowledge in D_S and T_S.

Referring to RQ 1 of Chapter 1, it could be seen that the transfer learning experiment 2 clearly outperformed the benchmark deep learning model. Depending on the error metric, the transfer learning approach could reduce the error of all time series of D_T by between 65 to 10 percent. Based on these results, it can be concluded that the transfer learning approach works on this specific problem and on the data.

For RQ 2 in Chapter 2, the time series of D_T were divided into short, medium, and long time series. In the analysis it becomes clear that short time series strongly benefit from the transfer learning approach. The improvement ranges from 71 to 55 percent depending on the error metric under consideration. In 75 percent of the short time series, it achieved a better forecast result. The medium time series could even be slightly better improved as the error metric could be reduced from 91 to 15 percent with transfer learning.

Regarding RQ 3 in Chapter 2, the transfer learning methods could not improve the long time series results in a meaningful way. Experiment 1 achieved worse results than the benchmark model in all error metrics. In contrast, experiment 2 could achieve slightly better results, but only in the range of 11 to 4 percent improvement.

The results of this study help to better understand forecast methods in the context of demand forecasting intermittent time series. Demand forecasting is highly relevant in the area of logistics and supply chain management. Through the analysis of three deep learning models, it could be shown that deep learning methods with transfer learning achieve better results than those without, especially on short and medium time series lengths.

Our work provides new and important insights, which are still partly limited and require further research. Additional investigations should be conducted to analyze the influence of similarity, considering distribution of D_S in terms of ADI and CV^2 in comparison to D_T to find further improvement possibilities by means of a potentially improved data selection.

Regarding the deep learning methods, we used a rather simple architecture, as shown in Chapter 7. Most recently developed architectures, such as transformer neural networks for example, could benefit even more from using a transfer learning approach. Furthermore, due to the limited data availability of intermittent time series data sets, only a rather simple transfer learning approach could be used. It is also possible to use multi source transfer learning with several $D_{S\,1}$, ..., $D_{S\,n}$.

This also leaves room to the idea of federated learning (also known as collaborative learning). By training a model on n $D_{S\,n}$, all participating collaborators benefit from a better target predictive function $f_T(\bullet)$

without sharing the data directly. Particularly for small and medium enterprises, this can provide a path to better deep learning models.

6. References

[1] A. Faisal, M. Kamruzzaman, T. Yigitcanlar, and G. Currie, "Understanding autonomous vehicles: A systematic literature review on capability, impact, planning and policy," *Journal of Transport and Land Use*, vol. 12, no. 1, pp. 45–72, 2019.

[2] L. Floridi and M. Chiriatti, "GPT-3: Its Nature, Scope, Limits, and Consequences," *Minds and Machines*, vol. 30, no. 4, pp. 681–694, 2020.

[3] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[4] K. Weiss, T.M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016.

[5] R. Miotto, F. Wang, S. Wang, X. Jiang, and J.T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.

[6] S. Shamshirband, T. Rabczuk, and K. Chau, "A Survey of Deep Learning Techniques: Application in Wind and Solar Energy Resources," *IEEE Access*, vol. 7, pp. 164650–164666, 2019.

[7] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[8] J. Wang, Y. Ma, L. Zhang, R.X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, pp. 144–156, 2018.

[9] Z.H. Kilimci *et al.*, "An Improved Demand Forecasting Model Using Deep Learning Approach and Proposed Decision Integration Strategy for Supply Chain," *Complexity*, vol. 2019, pp. 1–15, 2019.

[10] M. R. Amin-Naseri and B. R. Tabar, "Neural network approach to lumpy demand forecasting for spare parts in process industries," in *2008 International Conference on Computer and Communication Engineering*, 2008, pp. 1378–1382.

[11] D. Ivanov, A. Tsipoulanidis, and J. Schönberger, *Global Supply Chain and Operations Management*. Cham: Springer International Publishing, 2019.

[12] P. Oliveira, Fátima Rodrigues, and P. Henriques, "A Formal Definition of Data Quality Problems," in *ICIQ*, 2005.

[13] M. Bauer, C. Van Dinther, and D. Kiefer, "Machine learning in SME: An empirical study on enablers and success factors," in *AMCIS 2020 proceedings - Advances in information systems research: August 10-14, 2020, Online*, Atlanta, Georgia, USA: Americas conference on information systems: AMCIS/Association for Information Systems, 2020, pp. 1–10.

[14] A. A. Syntetos, J.E. Boylan, and J.D. Croston, "On the Categorization of Demand Patterns," *The Journal of the Operational Research Society*, vol. 56, no. 5, pp. 495–503, 2005.

[15] P. Xiong *et al.*, "Application of Transfer Learning in Continuous Time Series for Anomaly Detection in Commercial Aircraft Flight Data," in *2018 IEEE International Conference on Smart Cloud (SmartCloud)*, 2018, pp. 13–18.

[16] Q.-Q. He, P.C.-I. Pang, and Y.-W. Si, "Transfer Learning for Financial Time Series Forecasting," in *Lecture Notes in Computer Science, PRICAI 2019: Trends in Artificial Intelligence*, A. C. Nayak and A. Sharma, Eds., Cham: Springer International Publishing, 2019, pp. 24–36.

[17] V. Vaishnavi and W. Kuechler, *Design science research methods and patterns: Innovating information and communication technology*. Boca Raton: CRC Press, 2015.

[18] A. Corberán-Vallet, J.D. Bermúdez, and E. Vercher, "Forecasting correlated time series with exponential smoothing models," *International Journal of Forecasting*, vol. 27, no. 2, pp. 252–265, 2011.

[19] Y.-S. Lee and L.-I. Tong, "Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 66–72, 2011.

[20] S. Ben Taieb and R.J. Hyndman, "A gradient boosting approach to the Kaggle load forecasting competition," *International Journal of Forecasting*, vol. 30, no. 2, pp. 382–394, 2014.

[21] L. Liang and F. Shao, "Notice of Retraction: The Study on Short-Time Wind Speed Prediction Based on Time-Series Neural Network Algorithm," in *2010 Asia-Pacific Power and Energy Engineering Conference*, Chengdu, 32010, pp. 1–5.

[22] H. Yang, Z. Pan, and Q. Tao, "Robust and Adaptive Online Time Series Prediction with Long Short-Term Memory," *Computational intelligence and neuroscience*, vol. 2017, p. 9478952, 2017.

[23] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang, "Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1539–1548, 2018.

[24] Q. Gu and Q. Dai, "A novel active multi-source transfer learning algorithm for time series forecasting," *Appl Intell*, vol. 51, no. 3, pp. 1326–1350, 2021.

[25] N. Kourentzes, "Intermittent demand forecasts with neural networks," *International Journal of Production Economics*, vol. 143, no. 1, pp. 198–206, 2013.

[26] D. Kiefer, F. Grimm, M. Bauer, and C. Van Dinther, "Demand Forecasting Intermittent and Lumpy Time Series: Comparing Statistical, Machine Learning and Deep Learning Methods," in *Proceedings of the 54th Hawaii International Conference on System Sciences*, 2021.

[27] S.J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[28] M. Ribeiro, K. Grolinger, H.F. ElYamany, W.A. Higashino, and M.A. Capretz, "Transfer learning with

seasonal and trend adjustment for cross-building energy forecasting," *Energy and Buildings*, vol. 165, pp. 352–363, 2018.

[29] S. Makridakis and E. Spiliotis, "The M5 Competition and the Future of Human Expertise in Forecasting," *Foresight: The International Journal of Applied Forecasting*, no. 60, pp. 33–37, 2021.

[30] Fabian Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[31] Z. Wang and A.C. Bovik, "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[32] J. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons," *International Journal of Forecasting*, vol. 8, no. 1, pp. 69–80, 1992.

[33] S. Kolassa and E. Siemsen, *Demand forecasting for managers*. New York: Business Expert Press, 2016.

[34] G. VanRossum and F.L. Drake, *The Python language reference,* 3rd ed. [Hampton, NH], [Redwood City, Calif.]: Python Software Foundation; SoHo Books, 2010.

[35] M. Abadi *et al.,* "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," Mar. 2016. [Online]. Available: http://arxiv.org/pdf/1603.04467v2

7. Appendix

Table 5 contains the selected parameters for the respective models to forecast the demand (for parameters that are not listed, the default value was used).

Table 5. Selected parameters of the model

Parameter	Value
Activation function	Scaled exp. linear unit (selu)
Batch size	32
Dropout	AlphaDropout (0.01)
Early stopping	False
Epochs	300
Input window width	52 time steps (weeks)
Kernel initializer	Lecun normal
Loss	Mean squared error
Optimizer	Adam (learning rate = 0.0001)
Shuffle	False
Validation split	0.2

The following figure visualizes the applied network architecture for the benchmark experiment, for the individual experiment model adaptions, please refer to Chapter 3.

The chosen model consists of 10 fully connected dense layers. Each layer uses a lecun normal kernel initializer and is followed by a scaled exponential linear unit (selu) as activation function. As dropout, an AlphaDropout layer is placed between each dense layer, having a dropout rate of 0.01. The number of units can be seen in the following Figure 7. The implementation is realized using python 3 [34] and the keras implementation in tensorflow 2 [35].

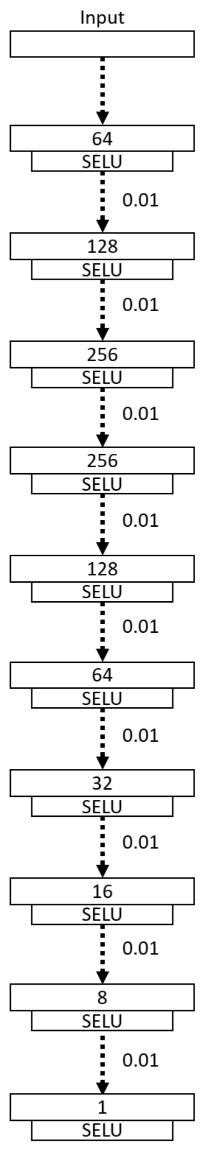

Figure 7. Deep learning architecture of the benchmark model