



A multi-source multi-layer-based transfer learning approach for forecasting customer demands of newly launched products

Supriyo Ahmed ^{a,b}, Ripon K. Chakraborty ^a,*, Daryl L. Essam ^a

^a School of Systems & Computing, UNSW Canberra at ADFA, Australia

^b Department of Electrical and Electronic Engineering, BRAC University, Dhaka, Bangladesh

ARTICLE INFO

Keywords:

Transfer learning
Supply chain
Sales forecasting of newly launched product
Deep learning
Multi-source transfer learning

ABSTRACT

Forecasting the future demand for newly launched products has been challenging for supply chain practitioners, often due to the lack of data. However, market surveys and extracting knowledge by examining similar market products to find the behaviour of a new product can be inaccurate and lead to erroneous results, which ultimately lead to a misestimation of the overall cost of a business. Meanwhile, with the advancement of artificial intelligence (AI) approaches, such as Transfer Learning (TL), this misestimation of cost can be reduced by more accurately forecasting the demand for newly launched products by seeking knowledge from the historical data of other similar products. Consequently, this paper investigates several classical AI-based TL approaches to predict customer demand for new products and stores. Thereafter, a novel Multi-Source Multi-Layer Transfer Learning approach with a Recursive Feature Elimination (MSML-TL-RFE) strategy is proposed to exploit the knowledge extraction power of the model from multiple sources for different days-ahead-prediction, distinguishing itself from the other investigated approaches. In this paper, an abstract concept of a supply chain, with information sharing among retailers, is investigated to show that such concepts can escalate the knowledge transfer ability of a system. A hierarchical two-echelon supply chain model with different attributes is developed to validate the proposed MSML-TL-RFE approach against a few other TL-based forecasting approaches. The feature-rich datasets are then transformed in such a way that they depict a hierarchical supply chain structure, allowing for the effective application of TL for forecasting consumer demand for recently introduced products. Continuing with that idea of information sharing, finding comparable sources for a quick and effective knowledge transfer procedure is investigated, considering all the peculiarities of a certain data set. MSML-TL-REF predictions and other TL-based approaches are analysed by calculating overall supply chain costs. Based on overall supply chain costs under static and dynamic lead time settings, the effectiveness and applicability of the proposed MSML-TL-RFE against traditional forecasting approaches are demonstrated. Incorporating MSML-TL-RFE with three sources improves accuracy, defined as the reciprocal of Root Mean Square Error (RMSE), from 4.83 (no TL) to 5.67 and further increases to 5.76 with additional sources, enabling more accurate predictions and reduced supply chain costs for businesses.

1. Introduction

Recent developments in AI, especially with deep learning models and their applications, have been substantial. In applications, such as fully autonomous cars (Faisal et al., 2019) and pre-trained chatbots (Floridi & Chiriatti, 2020) where human-like movements, reactions and generations of texts can be generated with the advent of AI-based approaches. These advancements in AI applications were made partly possible due to the application of TL (Niu et al., 2020). In simple terms, TL is a system where a deep learning architecture is pre-trained on a dataset D_S and then fine-tuned on a target domain dataset D_T based on the knowledge obtained from the training (Wang & Chen, 2023).

Current deep learning networks are successfully deployed in many areas, for example, medical care (Kaul et al., 2022), renewable energy (Abualigah et al., 2022), stock markets (Ahmed et al., 2022), production (Wang et al., 2018) and supply chain stock optimisation (Kilimci et al., 2019). Humans make decisions based on the predictions derived from these deep models. For example, in the field of demand prediction of spare parts (Ma et al., 2021), the purchasing team of a company can make informed decisions based on deep learning predictions (Amin-Naseri & Tabar, 2008).

In supply chain management, AI is being implemented gradually over time (Garinian et al., 2023). Through more accurate demand

* Corresponding author.

E-mail addresses: orphy123@gmail.com (S. Ahmed), r.chakraborty@unsw.edu.au (R.K. Chakraborty), d.essam@unsw.edu.au (D.L. Essam).

forecasting, businesses can gain significant competitive advantages in production, procurement and logistics (Ivanov et al., 2019; Liu et al., 2023). But the requirement of using neural networks is to have good quality data on the one hand while an appropriate quantity of data on the other (Oliveira et al., 2005). Compared to large companies, it is often difficult for small and medium-sized enterprises to have ample data to train such a network (Tang et al., 2023). This is relevant in the case of newly launched products, as there is often an insufficient amount of historical data for neural networks to make predictions (Li et al., 2022).

Time series data refers to sequential data points that are collected over time. Such data can be quite complex, displaying a range of diverse behaviours. As Syntetos et al. (2005) have pointed out, some time series data can be erratic and unpredictable, while others are smoother and more predictable. In some cases, the data can be lumpy and intermittent, with unpredictable spikes and dips. These various types of behaviours make it difficult to use common measurement metrics, like RMSE, to evaluate the data and also limit the use of a single machine learning or deep learning approach. Essentially, the nature of time series data requires a more nuanced and adaptive approach in order to analyse and extract insights from it effectively. Despite these challenges, researchers are keen to use machine learning (ML) and deep learning (DL) approaches for predicting such complex time series data. In doing so, scholars have found that it is not evident whether AI methods consistently produce better accuracy than statistical systems. For example, Kourentzes (2013) and Kiefer et al. (2021) showed that AI systems do not always provide users with the best results, indicating that there is a need for alternative methods. Additionally, research gaps have been identified in the field of lumpy and intermittent demand forecasting by Nikolopoulos (2021).

Over the years, along with the advancement of mathematical and machine learning theories, many algorithms with good performance for time series forecasting have been proposed by many prominent scholars. These include: Exponential Smoothing (ES) (Corberán-Vallet et al., 2011), Auto-Regressive Integrated Moving Average (ARIMA) (Lee & Tong, 2011) model, Gradient Boosting Machines (GBM) (Taieb & Hyndman, 2014), Neural Networks (NNs) (Ranganayaki et al., 2020), Long Short Term Memory (LSTM) (Liu, Long et al., 2022), Gated Recurrent Unit (GRU) (Veeramsetty et al., 2022) and numerous others (Gu & Dai, 2021). However, only linear and stationary time series data, such as ARIMA, can be used with linear models. In order to get reasonable prediction results, such as with the ARIMA model, practitioners try to transform nonlinear and nonstationary data into smooth time series data. Traditional linear prediction approaches struggle to adapt to real-world time series data because of their nonlinear and non-stationary nature. However, Difference processing can partially produce stationary time series data from non-stationary data by removing underlying trends and seasonality. According to Gu and Dai (2021), this is usually insufficient to explain how time series data adequately varies over time. In addition, if inaccurate or incomplete data have been added to a time series, difference processing cannot be properly carried out.

According to empirical investigations, nonlinear models typically function more effectively and reliably than linear algorithms (Gu et al., 2021). Since artificial neural networks can recognise non-linear cause-and-effect relationships, this paper intentionally concentrates on deep learning techniques. Time series forecasting of demand using deep learning approaches has already shown reliable results according to Criado-Ramón et al. (2022), Huang et al. (2022) and Ashtari et al. (2022). Two crucial assumptions are used in conventional demand forecasting with time series data (Gu & Dai, 2021): (i) the feature space from which the training and test sets of data are drawn should be the same, and those should follow the same probability distribution. (ii) enough training samples must be accessible in order to develop a strong prediction model. In many real-world application scenarios, the evolution of time series data over time results in a sizable disparity between recent and historical data. The amount of data that is readily

accessible in real-world applications, however, is quite minimal, which leaves inadequate training data. The two key presumptions cannot be satisfied in these circumstances.

To address all these challenges, TL has been suggested as a promising solution for time series prediction. By leveraging pre-trained models, TL can reduce the need for extensive training data and computational resources, which means it can produce good improvements, especially when the target data set is sparse (Pan & Yang, 2009). Recent studies, such as Gautam (2022) and Weber et al. (2021), have shown that TL can significantly improve the accuracy of time series prediction in comparison to traditional methods. By transferring knowledge from a similar domain, TL can be utilised to enhance a learner's knowledge of one particular domain. Suppose there is a learning task T_S in source domain D_S and a learning task T_T in target domain D_T . TL seeks to enhance the learning of the target predictive function $f(t)$ in D_T by acquiring knowledge from D_S and T_S . In this way, TL can be effectively applied in predicting time series data.

o date, TL methods have not been broadly investigated by researchers, especially in the field of supply chain domain Chen et al. (2023), Maschler (2023) and Li and Zheng (2020). However, there are some pieces of evidence of applications of TL in time series anomaly detection (Xiong et al., 2018) and in stock market predictions (He et al., 2019). This shows that the use of TL has gained momentum in other fields, while it needs further investigation in the supply chain domain to be able to predict outcomes for newly launched products with limited data. The accurate forecasting of demand for such products is crucial for effective inventory management and meeting customer expectations. However, insufficient data on a new product can significantly challenge the training of accurate predictive models (Liu, Lu et al., 2022; Zhang, Zhou et al., 2022).

1.1. Research gaps and contributions of this work

As discussed, although much effort has been dedicated to the transfer learning approach for forecasting customer demands, multiple essential research gaps in the body of knowledge should be addressed. They are as follows:

1. More research needs to be conducted to identify potential sources of information to enhance the TL processes.
2. As suggested by many researchers (Afrin et al., 2018; Chen et al., 2024; Ehrig et al., 2024; Kiefer et al., 2022; Li et al., 2024; Mahmood et al., 2022; Sagheer et al., 2021; Yang et al., 2019), more investigations need to be done to observe the effect of harvesting knowledge from single to multiple sources using one or more TL approaches.
3. The existing body of literature lacks explaining how the performance of different TL approaches varies over the selection of the different number of sources.
4. Although it has been mentioned as an abstract concept in past papers that information sharing leads to a more resilient SC network, almost no one has simulated this concept to make an empirical comparison.
5. Moreover, researchers mentioned cost minimisation as their future work, but a more thorough investigation needs to be done to check whether and how much TL really helps the Supply Chain (SC) to minimise cost, especially when it is related to a newly introduced market product.

To address these challenges/gaps, this paper's research methodology involves implementing a base CNN network for predictions, followed by establishing different aspects of the TL methodology.

Using CNN as the backbone of the TL network is motivated by CNNs' proven effectiveness in automatically extracting hierarchical features from data. CNNs have shown exceptional performance in various domains, including time series forecasting and sales prediction, due

to their ability to capture both local and global patterns within the data (Alsirhani et al., 2023; Ye & Dai, 2021). In this approach, CNN serves as the backbone to efficiently transfer knowledge from multiple sources while preserving the essential features necessary for accurate predictions. This is consistent with approaches like the one presented by Karb et al. (2020) in their work on network-based transfer learning for sales forecasting of new products, which also utilises CNNs for their feature extraction capabilities.

While implementing the TL methodology, a single item is identified from D_S that is close to D_T , and it is then used to train the CNN network. Pre-trained models are then used to make predictions in D_T . Next, the study identifies multiple sources of information for effective knowledge transfer and develops three different TL approaches for multiple sources, namely the Multi-Source Weighted Average TL Approach, Switching-based TL Approach, and Multi-source Multi-layer TL Approach. The performances of each of these approaches are compared to establish the superiority of one over the other in terms of accuracy. Additionally, two supply chain scenarios are simulated to showcase the benefits of information sharing among retailers, ultimately leading to better accuracy in the target domain and reducing supply chain costs along different channels. Cost calculations are also performed to show the improvement in supply chain performance along different channels of the network due to information sharing. Finally, the paper evaluates the cost minimisation potential of transfer learning in a supply chain scenario for predicting demand for a newly launched product. Hence, in short, the contributions stemming from this work are multi-fold and include:

- Utilisation of a robust framework for identifying optimal sources for efficient knowledge transfer in transfer learning applications. By utilising a k-nearest neighbours (KNN) model trained on a comprehensive set of sales features extracted from restructured datasets, the approach enables accurate identification of comparable items that exhibit sales patterns analogous to a newly introduced market product. This holistic consideration of all available data features ensures that the identified sources are not only relevant but also possess high potential for effective knowledge transfer, thereby improving the predictive accuracy of models applied to the target domain.
- The study extensively evaluates the impact of multi-source knowledge transfer on the effectiveness of various TL approaches. The research assesses their predictive performance in estimating future customer demand by pre-training up to five different TL models on identified similar sales datasets and subsequently fine-tuning them on data for a new product. This comparative analysis provides valuable insights into the relative strengths and weaknesses of different TL techniques when applied to real-world sales forecasting, demonstrating that multi-source learning can significantly enhance model accuracy and robustness compared to single-source approaches.
- A novel Multi-Source Multi-Layer Transfer Learning with Recursive Feature Elimination (MSML-TL-RFE) technique that advances the field of transfer learning by enabling the efficient transfer of knowledge across domains with differing data characteristics is proposed. This method integrates a multi-source and multi-layer learning architecture with an RFE strategy, optimising the feature set for transfer learning and improving model generalisation in the target domain. The MSML-TL-RFE technique demonstrates improved predictive performance by effectively distilling relevant knowledge from multiple sources and applying it to the target domain, making it a powerful tool for addressing complex prediction tasks in dynamic environments.

The remainder of this paper is organised as follows: Section 2 presents a brief overview of existing literature, mostly related to the application of TL. Based on certain assumptions regarding limited data,

the problem formulation is discussed in Section 3. Then, from the existing literature and problem definition, Section 4 provides a brief summary of the different TL approaches for making sales/demand forecasts. The implementation of the model, along with data collected from different public platforms and their feature extraction techniques, is discussed in Section 5. The findings from the experimentation are analysed and discussed in Section 6. The limitations and potential future research avenues are outlined in Section 7. Finally, the conclusions are presented in Section 8.

2. Literature review

The literature review section of this paper aims to provide an in-depth understanding of transfer learning and its current methods, as well as their application fields. This section explores the existing research on transfer learning and its different approaches and also discusses their advantages and limitations. Moreover, the different areas of application where transfer learning has been used with promising results are examined. Towards the latter part of this section, some research gaps in the existing literature are highlighted, and potential avenues to address these gaps are discussed. This comprehensive review aims to provide a solid foundation for the subsequent sections of the paper that present the specific contributions of this research work.

Businesses often face a challenge when launching new products in the market, as the demand for a new product is often unknown. With very limited historical data, it becomes difficult for a business to predict the future demand of their product using regular machine learning or deep learning tools. Of late, a lot of effort has been undertaken by researchers regarding image-based transfer learning (Alfano et al., 2024; Jangam et al., 2022; Janković Babić, 2024; Ju et al., 2022; Muhammad et al., 2020; Saleem et al., 2022; Wang, Lin et al., 2021). However, very little is being explored by the researchers in the area of time series analysis, especially in the context of supply chain sales prediction of newly introduced products.

2.1. TL in supply chains

In the field of supply chains, researchers are trying to use TL to reduce machine downtime through prompt and accurate detection of faults so that manufacturing organisations can stay competitive throughout the year (Rai et al., 2021; Wang et al., 2023; Zhang, Song et al., 2022). As more and more clients demand manufacturers to expedite the delivery of high-quality products cheaply, machine learning algorithms in the manufacturing industry will see growing utilisation for allowing better fault diagnosis of manufacturing systems. Using deformable convolutional neural network (CNN), deep long short-term memory (DLSTM) and transfer learning strategies, Wang, Liu et al. (2021) presented a CNN-DLSTM-based TL approach to finding the faults in the roller bearings of machines. By detecting potential faults in advance, manufacturers and suppliers can plan for maintenance and repair, reducing the risk of unplanned downtime and disruption to the supply chain and improving the overall efficiency of a network. Their system was trained using sample data (since real data is difficult to find) during a pre-training phase and then later tested on their system's experimental data to assess their approach.

In another work, researchers like Wu et al. (2022) used a knowledge transfer technique from an industrial chain to predict stock market movement. The authors presented a TL idea of fusing information related to the industry chains with deep learning models to forecast stock prices. The authors' research work was based on the TL of industrial chain data of upstream businesses. After being first trained using the stock data of upstream businesses, the DL models (i.e., MLP, RNN, LSTM and GRU) were then retrained using the stock data of the appropriate downstream businesses to forecast the trend of the target stock. The authors found that MLP's performance was better in transferring knowledge, having an accuracy of 68.4%, as compared to

other approaches such as RNN, LSTM and GRU, whose accuracy was in the range of 57%–58% while predicting the target's stock market index.

Continuing with TL systems, [Ramezankhani et al. \(2021\)](#) proposed a TL architecture for the autoclave composite manufacturing process. The authors used historical data to train a fully connected neural network (FCNN) and transferred the knowledge to a new process. They avoided over-fitting with limited data by using a sequential freezing method. The model's generalisation loss was reduced by 88%, but the authors used only one type of DL technique, while other ML/DL models remain unexplored. Moreover, they suggested changing the number of frozen layers to study the model's performance and conducting a similarity analysis between the source task and target task to maximise knowledge transfer and minimise negative transfer. This could help identify suitable knowledge sources for future TL researchers.

Considering the source of knowledge could also contribute to selecting an appropriate technology as a source. According to [Da Silva et al. \(2019\)](#), industries do not commence their operation with all the characteristics of Industry 4.0 from the very beginning. They evolve over time and adapt new technologies from other stakeholders. The study conducted by [Da Silva et al. \(2019\)](#) encouraged the continuous development of technologies that need to be transferred to meet market demand [De Carvalho and Viana Cunha \(2013\)](#). TL could be an effective way to deal with this technological knowledge as a transfer system can aid in the performance management of industries through managing control systems and evaluating the performance indicators of a business.

2.2. TL in predicting future data related to supply chains

In the case of future data predictions in supply chains, researchers proposed that TL can be an efficient tool. A study by [Karb et al. \(2020\)](#), investigated a TL approach for DNN to predict the demand of newly launched food sales for an Austrian food retail industry. The authors transferred knowledge from preexisting stock products to predict the demand for newly introduced products and showed that DNN with TL works better than DNN without TL. However, the researchers trained only one single model for their entire application, and using a parameter-based TL approach with multiple neural networks and hyper-parameter optimisation remains unexplored.

In order for TL approaches to work effectively, it is necessary to identify the proper sources of information. Over the years, researchers have tried to harvest information from a single source and also from multiple sources ([Huang et al., 2023](#); [Nicholson et al., 2022](#); [Zhao et al., 2023](#)). In an effort to deal with gathering knowledge from a single source and also trying to simulate real-life scenarios where there are limitations of labelled data, [Ye and Dai \(2021\)](#) proposed a DTr-CNN approach, that is a deep transfer learning method using the architecture of CNN to cope with insufficient data to make time series forecasting. Their transfer learning framework uses dynamic time wrapping (DTW) and Jensen–Shannon divergence to identify similar source tasks to the target task and then uses DTr-CNN to transfer knowledge from the source to the target. The authors conducted experiments on three types of datasets and found that DTr-CNN outperformed other approaches in terms of RMSE and sMAPE. However, this approach can only select one source dataset from multiple datasets and can only work with time series data with the same feature size. The authors suggested adapting multi-source transfer-learning techniques for better accuracy and using an appropriate selection approach in the future for varying feature sizes.

Research related to time series forecasting using TL is rare, especially in the context of supply chains. An article by [Kiefer et al. \(2022\)](#) analysed the use of TL in combination with DL for time series forecasting in the context of supply chains. The authors used M5 competition data to conduct their experiments and found that the TL approach can reduce prediction error for all time series data, particularly for short and medium time series. However, for time series with a long history,

benchmark models work better than TL. The paper suggests future research directions, such as conducting a similarity analysis of data in the source and target domains, using more advanced neural networks, and training a model on multiple source domains.

In the same time series forecasting domain, [Ehrig et al. \(2024\)](#) investigated the role of dataset characteristics, specifically similarity and diversity, in the success of transfer learning applied to time series forecasting. The authors pre-train a probabilistic recurrent neural network, DeepAR, on multiple public and non-public datasets and assess its performance in both zero-shot and fine-tuned applications to various target datasets. By analysing and quantifying the similarity and diversity between the source and target datasets through feature extraction and principal component analysis (PCA), the study finds that greater diversity in source data improves point forecast accuracy and uncertainty estimates, while source–target similarity reduces bias. These effects are particularly pronounced in zero-shot learning scenarios. However, the study's conclusions are limited by the random selection of source datasets, which may not be optimally similar or diverse relative to the target datasets, leading to results that are only valid in a relative context without establishing absolute thresholds for transfer learning success.

Another study conducted by [Wang et al. \(2024\)](#) proposed a deep transfer learning method with unsupervised dynamic domain adaptation (DDA) to improve quality prediction in industrial production across products with varying specifications, where traditional models struggle due to differing data distributions. The method leveraged a Wasserstein distance ([Nguyen & Ho, 2024](#)) adapter to effectively match source and target domains and combines dynamic distribution and adversarial adaptations to enhance prediction accuracy. Experimental results showed significant improvements in prediction metrics compared to traditional and other transfer methods. The study also introduced a multi-source DDA transfer framework, though it highlights the need for integrating physical mechanisms with data-driven approaches in future work.

A TL framework was presented by [Sagheer et al. \(2021\)](#) to address the hierarchical time series prediction problem, which was not directly related to the supply chain problem. In their research work, the authors used the DLSTM autoencoder (DLSTM-AE) system in a TL framework. They generated a base forecast using DLSTM-AE from the bottom of a hierarchy, froze the weights of the base models, and transferred the learned features to attain synchronous training to the time series for the upper levels in the hierarchy. The authors showed that their proposed DLSTM-AE produced a more accurate and coherent forecast than ARIMA and ES over different levels of the hierarchy. They plan to adapt their technique to other domains in the future.

One of the most challenging tasks for any business is to identify how a newly launched product would behave in the market. Although a lot of market surveys can be done before a launch, its behaviour could still be better identified by TL approaches ([Wu et al., 2022](#)). During the launching of a new product in the market, it is important to determine its demand as early as possible. Research has found that a lack of consumer-related information makes this demand prediction unreliable and difficult. That is why scholars have been trying to harvest information from other market products to predict the demand for recently introduced products.

Newly introduced products have some commonality and/or differentiation with their predecessors, and such data is widely available and studied when required ([Afrin et al., 2018](#)). [Yang et al. \(2019\)](#) proposed a mathematical architecture to forecast demand for a newly launched product by using a weighted product differentiation index from the market demand of its predecessor. The authors predicted the demand for a new product by using an exponentially weighted moving average machine learning forecasting approach. The method was tested on automobile industry data, and the authors demonstrated its efficacy. However, the method needs validation on other datasets, and the weights for different components and options of products

were manually assigned, which is cumbersome for large datasets. The relationship between different features of products that influence their corresponding demand is yet to be established, which can be done using a deep neural network model.

As mentioned previously, companies often face challenges predicting the demand for newly launched products due to the lack of historical data. Kharfan et al. (2021) proposed a three-step model to predict the demand for newly launched products in fashion retail using clustering, classification, and prediction. They identified similar products from a training set using clustering, established a relationship between the styles of pre-assigned clusters from the train set and new styles from the validation and test sets through classification, and predicted sales of the new product using the identified clusters. The authors suggested that regression trees can be used for complex clustering problems with multiple life-cycle lengths, while k-NN and linear regression approaches can be used for identifying high sales volume and average unit retail products. The paper had some limitations as well, such as not considering lost sales and the potential use of deep TL instead of averaging historical sales data.

2.3. Summary of the literature review and the significance of this study

The existing research on TL for forecasting customer demands highlights several key gaps that need to be addressed. Firstly, there is a lack of studies focused on identifying potential sources of information that could enhance TL processes. Additionally, while researchers have suggested the importance of harvesting knowledge from multiple sources, there is limited investigation into the effects of using single versus multiple sources with various TL approaches. Furthermore, the current literature does not adequately explain how the performance of different TL methods varies depending on the number and selection of sources. Another critical gap is the absence of empirical simulations to validate the concept that information sharing contributes to more resilient supply chain networks despite its mention as an abstract idea in past studies. Lastly, although cost minimisation is frequently cited as a potential benefit of TL, there is insufficient research to empirically evaluate its effectiveness in reducing supply chain costs, particularly for newly introduced market products. These gaps underscore the need for deeper exploration to advance the practical application of TL in supply chain management, which also motivates this study to consider open research questions to address.

Addressing the above-mentioned gaps will certainly benefit practitioners in making informed decisions and choices regarding the use of TL approaches to predict the demand for a newly launched product. It will also help them to run their business at a lower cost by reducing unnecessary holding and lost sales costs. Consequently, this paper has identified multiple sources of information for effective knowledge transfer in predicting demand for a newly launched product. This research addresses critical gaps in TL for forecasting customer demands by employing a robust methodology centred on the implementation of a base CNN for predictions. The CNN serves as the backbone for efficiently transferring knowledge from multiple sources while retaining essential predictive features. The methodology involves identifying a single item from the source domain (D_S) that closely matches the target domain (D_T) using the KNN model trained on comprehensive sales features. Pre-trained models are fine-tuned on D_T for accurate predictions, followed by the exploration of three innovative multi-source TL approaches: the Multi-Source Weighted Average TL Approach, the Switching-Based TL Approach, and the Multi-Source Multi-Layer TL Approach. These approaches are compared for their accuracy in knowledge transfer and predictive performance. Additionally, supply chain scenarios are simulated to demonstrate the benefits of information sharing among retailers, which enhances accuracy, reduces costs, and improves supply chain efficiency, particularly for newly introduced market products.

This study contributes significantly to the field by developing a novel MSML-TL-RFE technique. This method optimises feature selection and enhances model generalisation in the target domain by integrating multi-source, multi-layer architectures with recursive feature elimination. The research provides valuable insights into the effectiveness of multi-source TL for real-world sales forecasting, demonstrating its superiority in improving predictive accuracy, reducing costs, and addressing complex dynamic challenges in supply chain networks.

3. Problem definition

In order to provide a practical idea of how much TL approaches actually benefit an SC network, all the TL approaches explored in this paper are applied to supply chain data to evaluate their impact. It is worth mentioning that all the experiments are conducted based on the assumption that a product has been recently launched for around one month in the market. Based on the sales pattern of that product during that time, similar items are identified from other groups of products to extract knowledge. For such identifications, a well-known KNN network is used, and then a base CNN model is trained with data obtained from similar products, and then the weights and bias of the pre-trained network are transferred to form a new NN which is then retrained using one month of data from the recently launched product to predict 6 months of data.

To further extrapolate the assumptions, the collected data is restructured in such a way that it represents a hierarchical supply chain structure where there is one supplier and three retailers selling different products. Single and multi-source TL approaches are implemented on three datasets under two scenarios (1) no information sharing and (2) with information sharing. A detailed representation of horizontal information sharing among the stores is shown in Fig. 1.

From Fig. 1, it can be seen that each store sells numerous items for some previous time, which act as D_S data, while there is also one product (Item X) which is supposedly the new launch product (according to the assumptions) in the market to act as the D_T data in this experimental analysis. With no information sharing, only the locally long-term sold items per store are considered, among which the most suitable product/s are chosen as per the KNN algorithm, while with information sharing, items sold at other stores are also taken into consideration. The two concepts are discussed in detail in the latter part of this subsection. Once the predictions and accuracy of the different TL approaches are established, those predictions are then passed through a supply chain model for further cost calculations. The working principle of the supply chain model is also discussed later in this paper.

For properly establishing the TL approaches on the no information sharing concept, with distance matrix being the determinant, similar items are ranked and identified using KNN from a fixed set of products that are sold by a single retailer, e.g. Store 1, thus keeping the knowledge sharing concept contained within the specified Store 1. Once the items are ranked, the first three products that are the shortest distance from the target product are chosen. The Euclidean distances from KNN are used to calculate the weights of each similar item through a method called inverse weighted distance (Pereira et al., 2022). These weights are later assigned to find the impact of the weighted average of a multi-source TL system where multiple layers get information from multiple pre-trained models in the entire TL system.

On the other hand, if the retailers agree to share their product information with each other, then similar items can be identified from a larger pool of items using the same KNN system. According to Fig. 1, with the horizontal information sharing concept, each store now has information on the items and sales patterns of other stores. This means that information can now be extracted from finding out similar items from other retail outlets, as well as its own set of products. It has been mentioned by several researchers in the past (Dolgui et al., 2020; Ivanov et al., 2021, 2018), that sharing information among competitors adds value to the entire supply chain network. It enhances flexibility

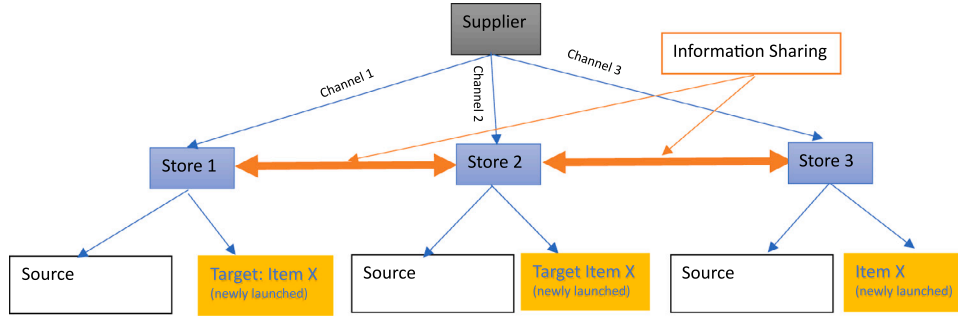


Fig. 1. Supply chain information sharing concept.

and makes the network more resilient. This paper focuses on the cost-minimisation aspect of information exchange through TL in supply chains.

Next, a hierarchical supply chain model is considered in this paper, which consists of at least 3 retailers and 1 distributor (supplier). This means that there are three channels through which products can be delivered to customers. Each of the channels is basically a two-stage supply chain architecture that consists of a retailer and a supplier. The supplier–retailer simple model is developed based on the following assumptions: the retailer sells the newly launched product alongside other commodities, there is no backorder cost, but rather a cost for lost sales is incurred whenever the retailer fails to satisfy customers' needs, the supplier has storage facility to store and deliver the finished product to the retailer after an order is placed by the retailer.

Meanwhile, motivated by the work of Gonçalves et al. (2020), the safety stock (ss) is determined by using Eq. (1), as given below. ss has two associated dimensions: demand variability and lead time variability of suppliers. Variations in demand can be found from the ML/DL predictions, but with lead time, there are two different possibilities: fixed lead time and dynamic lead time. When the lead time is fixed, $\sigma_L = 0$, and hence Eq. (1) is reduced to Eq. (2). The symbols used in Eq. (2) have their usual meanings.

$$ss = Z \sqrt{(\sigma_d^2 \bar{L} + (\sigma_L^2) \bar{d}}; \quad (1)$$

where σ_d is the standard deviation of predicted demand, σ_L is the standard deviation of Lead time, $\bar{L}$ is the average lead time, $\bar{d}$ is average predicted demand, and Z represents the number of standard deviations corresponding to service level probability.

$$ss = Z \sigma_d \sqrt{\bar{L}}; \quad (2)$$

The developed supply chain model also calculates the predicted demand during the lead time, named lead time predicted demand ($LTPD$), which is assumed to be provided by the TL forecasting algorithms. Using the variation of previous lead time and the predicted demand, the model can be operated with a certain amount of ss . ss is an extra quantity of goods stored in the inventory which prevents a business from a stock-out situation. The summation of $LTPD$ and ss together creates a reorder amount (RA), as shown in Eq. (3). The decision of whether to place an order or not by the retailers is determined by this RA . The model also keeps track of the total number of orders in delivery. So, for each of the channels, whenever RA becomes greater than the summation of the current inventory level and orders in delivery, an order placement is issued to the supplier for replenishment. RA can also be varied by changing the level of safety stock. This is an added feature to make the model more realistic.

$$RA = LTPD + ss \quad (3)$$

The model has four costs associated with it: purchasing costs, ordering costs, holding costs and lost sales. The summation of these costs is

Table 1

Input data for the supply chain model.

SC parameters	Dataset 1
Ordering cost (per order)	\$20
Purchasing cost (per unit)	\$2
Holding cost (per unit per year)	\$7
Lost sales cost (per unit)	\$5
Retailer's beginning inventory (units)	250
Retailer lot size Q (units)	200

considered as the total cost (TC) of the supply chain according to Eq. (4).

$$TC = \sum(PC, OC, HC, LS); \quad (4)$$

The following are the assumptions regarding the associated costs:

- Purchasing cost (PC): The purchasing cost per unit product is known and fixed.
- Ordering Cost (OC): It is assumed to be known, fixed and independent of order quantity.
- Holding Cost (HC): It increases or decreases linearly with the amount of inventory in stock.
- Lost Sales (LS): It is assumed to be fixed per unit, and the overall lost sales increase linearly as more goods are out of stock.

Out of three datasets, only one dataset (Dataset 1) is considered for evaluation of the proposed TL approach in terms of supply chain cost in this paper, for which the supply chain model's parameters are set as follows. But it could be slightly changed to ensure proper triggering of order placement for different datasets, because of the different sales levels in different datasets. The parameters used for the experimentation are presented in Table 1.

From Table 1, it can be seen that ordering cost is assumed to be \$20 per order while purchasing cost is \$2. The annual holding cost is \$7 per unit, while the cost of a lost sales opportunity is assumed to be \$5 for each product unit. For each fresh set of experiments, the retailer's beginning inventory is initialised to 250 units, and once the remaining inventory orders in place drop below RA , 200 units are ordered to replenish the stock, which is the lot size Q in this case.

4. Proposed methodology

In this paper, a simple base CNN network is first implemented for predictions, and then it is extended to establish different aspects of our TL methodology. From D_S , a single item is identified which is close to D_T , and it is used to train the CNN network. The pre-trained models are later used for making predictions in D_T . Next, instead of limiting the knowledge pool to a single source, multiple sources are identified to enhance the performance of the TL system. For multiple sources, three different TL approaches are developed: (i) Multi-Source Weighted Average TL Approach, (ii) A switching-based TL approach and (iii)

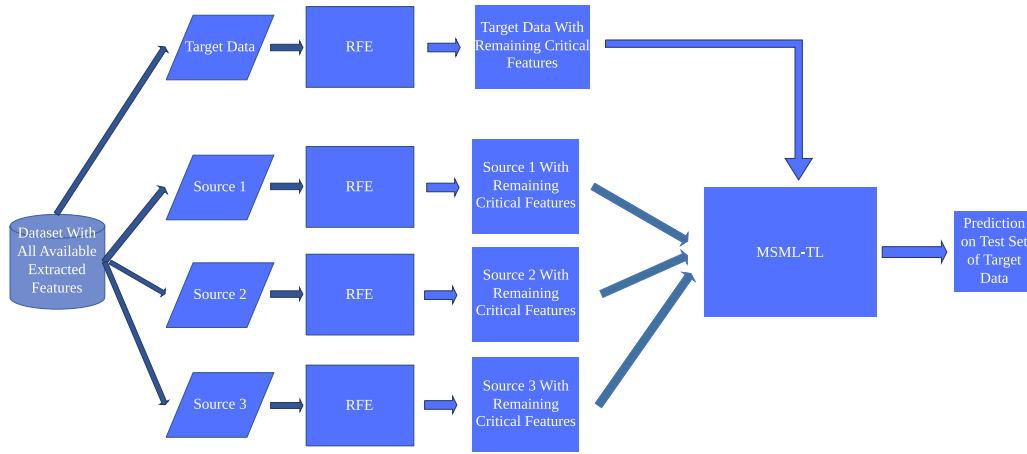


Fig. 2. Overall flowchart of the MSML-TL-RFE.

Multi-source Multi-layer based TL approach. This Multi-source Multi-layer based TL (MSML-TL) approach is further extended by the addition of RFE for enhanced performance and it is demonstrated by an overall flowchart in Fig. 2. The performances of each of these approaches are compared to establish the superiority of one over the other in terms of accuracy.

4.1. Simple CNN network: No-TL approach

Initially, a simple CNN, adapted from the literature (Mehtab & Sen, 2022), is deployed in the target domain data for predictions. CNN is a kind of deep NN that is commonly used in experimentation with image data. With sequential data, Convolutional layers have the capability to withdraw critical information and learn hidden patterns of data (Livieris et al., 2020). The Convolution function defines the inner product (product and sum) of window data with a filter matrix which is a fixed set of weights. The objective of the MaxPooling layer is to simultaneously minimise the eigenvectors' size, which is needed to simplify the network's computational complexity and compression and extraction of critical features. Fig. 3 shows a one-dimensional CNN model.

In this paper, another simple CNN-based neural network from the literature (Mehtab & Sen, 2022) is deployed with slight modification by some additional layers. It is trained and validated with one month of data which is related to the recently launched product (target product). The CNN network contains three 1D convolutional layers with MaxPooling layers in between each. The last two layers are flattened and dense layers. The output of the final convolution layer is passed through the flattened layer, whose purpose is to convert the output into a one-dimensional array so as to create a single long feature vector. Finally, the dense layer is used to combine that single vector to produce the predicted result. Since this network is only deployed in the target domain, hence this system is making predictions without any previous knowledge from any other source, and thus, this system is termed a No-TL system.

4.2. Single source TL: SS-TL

The same CNN-based network, mentioned in Section 4.1, was used to obtain a pre-trained model using sales data of a similar item, and then the weights and biases from the pre-trained network of the first 4 layers are frozen, and the remaining lower level layers, along with the frozen layers, were combined together to create a new NN which was trained and validated with the data from the newly launched item. A layout diagram of the Single source TL (SS-TL) approach is illustrated in Fig. 4.

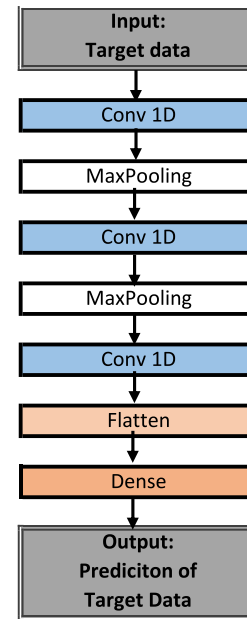


Fig. 3. One dimensional CNN (No-TL approach) (Jin et al., 2020).

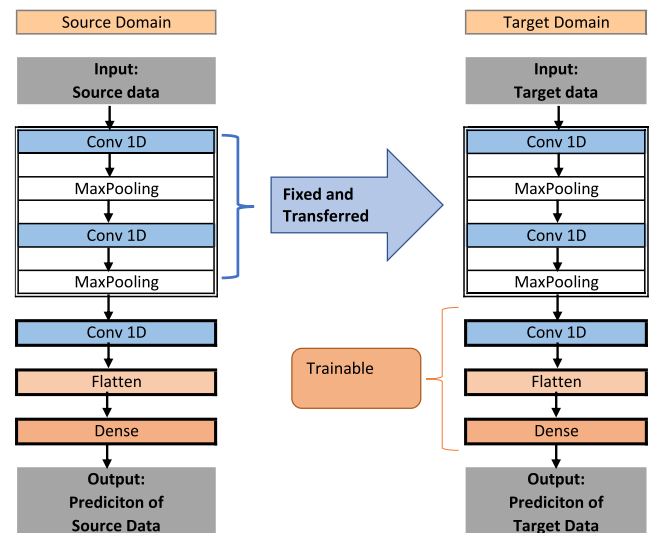


Fig. 4. Single Source TL network (SS-TL).

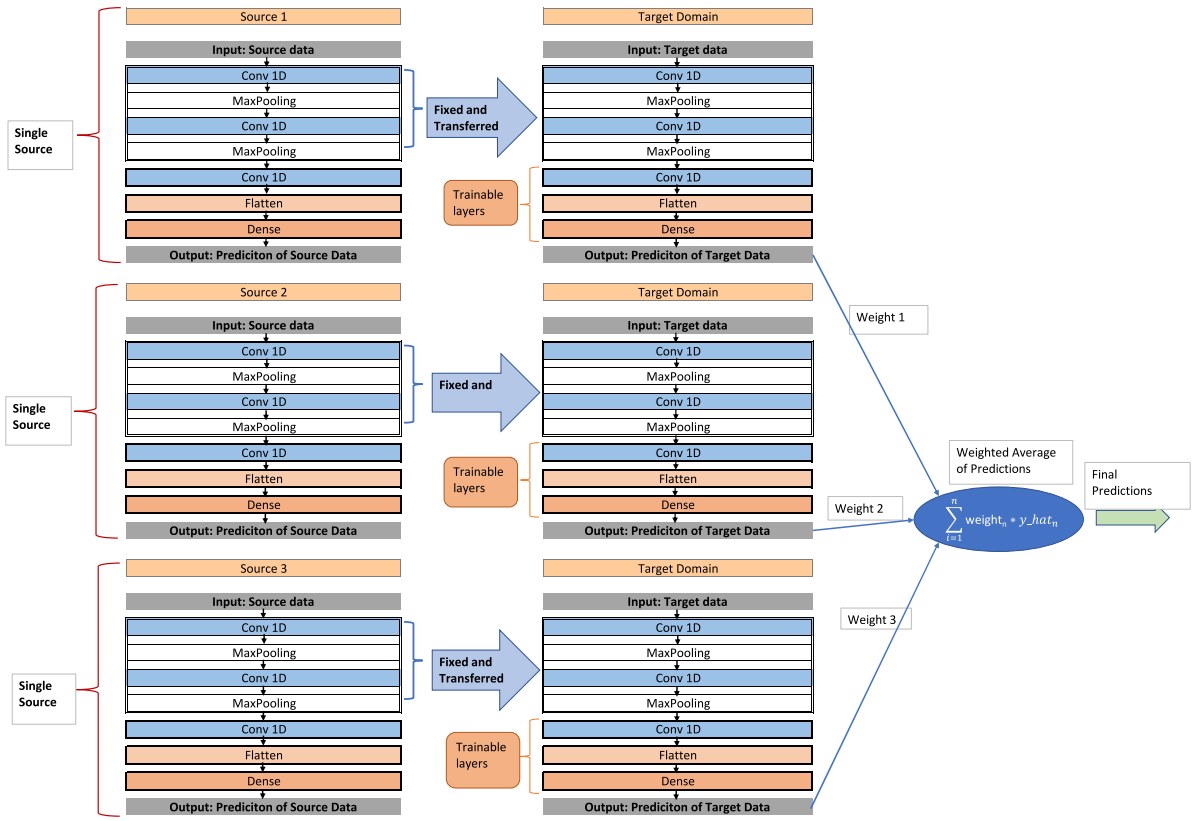


Fig. 5. Multi-Source Weighted Average TL network (MSWA-TL).

In this approach, a similar source product is first identified using KNN, which is a very well-known classification machine learning tool. This KNN algorithm works by comparing the features of the target product to those of the sources and finding the k -nearest sources based on a Euclidean distance. Once the source item is identified using KNN, it is used to train the CNN network belonging to the source domain according to the diagram 4. Then, some layers are frozen and transferred to form a new CNN network, which is then re-trained and validated using target domain data. It is worth mentioning that apart from the frozen layers, the other lower-level layers are kept trainable so that the system adapts itself to the new target domain data to make predictions. For each day ahead of prediction, the experiment is repeated, and the mean predictions are calculated.

4.3. Multi source TL approaches

This paper explores three multi-source TL approaches. A multi-source weighted average is considered where different single-source models are trained using different source data and ultimately tested using a single target set. The total prediction is a weighted sum of the individual predictions obtained from different models pre-trained using different sources. Another way to transfer knowledge from multiple sources is a switching-based TL approach, which is also presented in this paper. Here the selection of a particular TL approach is switched between different approaches based on validation data in the target domain. The last TL approach investigated in this paper is a Multi-Source, Multi-Layer TL approach enhanced with recursive feature elimination (RFE). In this system, important features are extracted using RFE from all source and target data before training the pre-trained networks. After training, the weights and biases from multiple pre-trained networks are combined together using the weighted average method to form a new network in the target domain. All these approaches showed promising results, and a detailed explanation of each one is given in the following sections.

4.3.1. Multi-Source Weighted Average TL approach: MSWA-TL

To explore another dimension of the TL framework, the single source TL approach from Section 4.2 is used three times with three different sources with the same target domain data for a particular dataset. The source domain items/stores are selected based on the KNN algorithm. The structure of the Multi-source Weighted Average TL Approach used in this paper is presented in Fig. 5.

It can be seen from Fig. 5 that there are basically three single-source TL approaches, where each of those is fed with similar source data for training. After that, the obtained pre-trained models are retrained using the target domain data. The same process is carried out for other similar products acquired from the distances of each product or store produced by the KNN model. The distances are then used to assign the corresponding inverse distance weights of each item/store. The output predictions of each of the systems acquiring knowledge from a particular source are then multiplied with the corresponding weights (which will be discussed later in this paper) and added together to find the final predictions of the testing set of the target dataset. To remove randomness, multiple experiments are carried out, and the average of all the weighted average predictions is calculated to find the final predictions. The finally predicted results are later compared against the testing set of the target dataset for further evaluation.

4.3.2. Multi-source switching based TL approach: MSSB-TL

The same single source network mentioned in Section 4.2 was used multiple times to make a multi-source TL network. Multiple similar sources of information are identified using KNN, and then each of the sources is used to train the network to obtain a pre-trained network for each source. The layers from each one are then transferred to create a new individual CNN network for each source, which is then re-trained and validated using the target dataset. Fig. 6 represents the architecture of the multi-source switching-based TL (MSSB-TL) approach, which has been explored to establish that knowledge transfer from multiple courses often leads to better performance in a target domain.

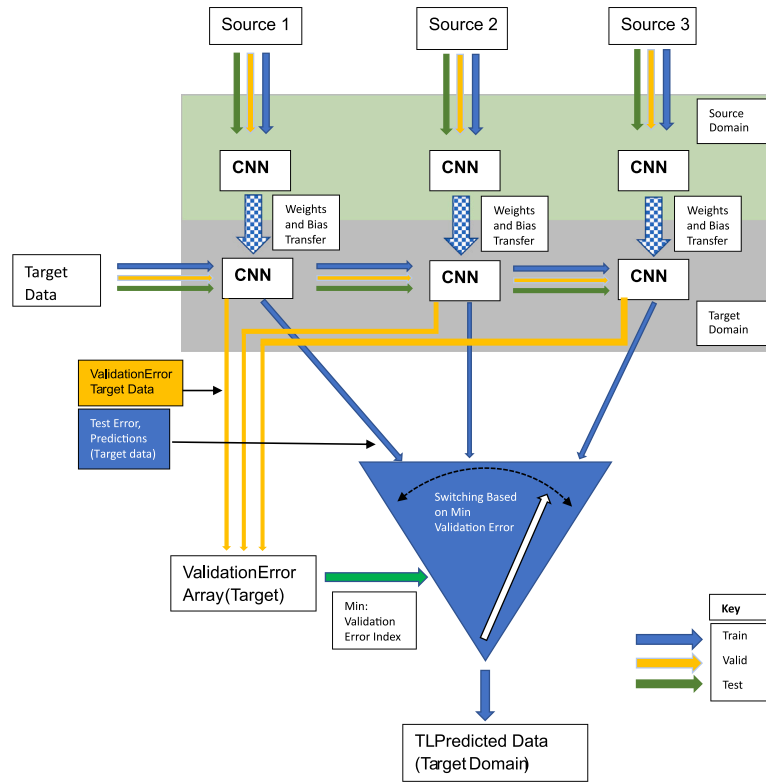


Fig. 6. Multi-source switching based TL network (MSSB-TL).

According to Fig. 6, the switching between different models trained from different sources is based on the minimum validation of RMSE. In the MSSB-TL system, predictions from a single set of TL networks are selected when there is minimum validation RMSE in the target domain. The same experiment is repeated, and the average predictions are taken into consideration. The same is repeated for different forecasting horizons.

4.3.3. Multi-source multi-layer TL approach: MSML-TL

Adapting and extending the idea further from SS-TL, MSWA-TL, and MSSB-TL networks, another multi-source TL framework is presented in this paper, where the same network is trained using multiple sources at different times. Then, the weights and biases of each of the networks from the pre-trained models are combined together using the weighted average method to form the weights and biases of some new frozen layers of another CNN model. The newly formed model with its new frozen layers with other trainable lower-level layers is re-trained and validated using data from the target domain. A layout of the proposed Multi-Source Multi-layer based TL (MSML-TL) approach that harnesses knowledge from multiple sources is illustrated in Fig. 7.

From Fig. 7 it can be seen that there are three pre-trained models from which knowledge is harnessed. As mentioned before, this paper initially focused on three sources when multi-source TL is mentioned. The three sources are used to pre-train three CNN models, and then a weighted average is calculated for weights and biases of each layer of the three CNN models. The weighted average of each of the corresponding layers is then passed to new CNN models where those particular layers are frozen, and the entire network is re-trained with target domain data. As usual, for each day ahead of predictions, multiple experiments are carried out, and the average value is calculated to overcome randomness in the system.

4.3.4. Multi-source multi-layer TL approach with RFE: MSML-TL-RFE

Before feeding data into the previously mentioned MSML-TL approach, critical features were extracted using a popular technique

called Recursive Feature Elimination (RFE) (Lee et al., 2022). The process is presented in Fig. 8, where the important features are mainly selected from similar source data.

Using RFE, around 40%–60% of features are extracted from each data set. This was done to train the networks using the most important features and thus enhance the knowledge transfer ability of the system. In addition, RFE helps reduce the number of features of the datasets and thus aids in minimising the complexity of the entire model. As usual, the weights from each of the layers of the trained models from different sources are added using the weighted average method to create the initial frozen layers of the new model. The frozen layers with other trainable layers are retrained together using target data and this approach enhances the knowledge transfer ability of the system with increased prediction accuracy in the target domain. The combined method of extracting important features from multiple similar sources using RFE and feeding the information into a Multi-source, Multi-Layer TL framework is presented as MSML-TL-RFE in this paper.

4.3.5. Pseudo-code of the proposed MSML-TL-RFE

The proposed MSML-TL-RFE technique is further explained by a pseudo-code in Algorithm 1.

The pseudo-code in Algorithm 1 represents a TL workflow for predicting values of a target domain test set and evaluating with RMSE. The workflow includes the following steps: first, the KNN algorithm is applied to identify three similar items to the target domain item based on Euclidean distances. Next, RFE is used to reduce the number of features for the target item and similar products. The data is then split into training, validation, and test sets. Independent CNN models are trained for each similar item, and their weights and biases are combined to form a new CNN network using weighted averaging based on the Euclidean distances. Finally, the new CNN network is retrained using the target domain test data, and the RMSE is calculated using the predicted values and the true values of the target test set.

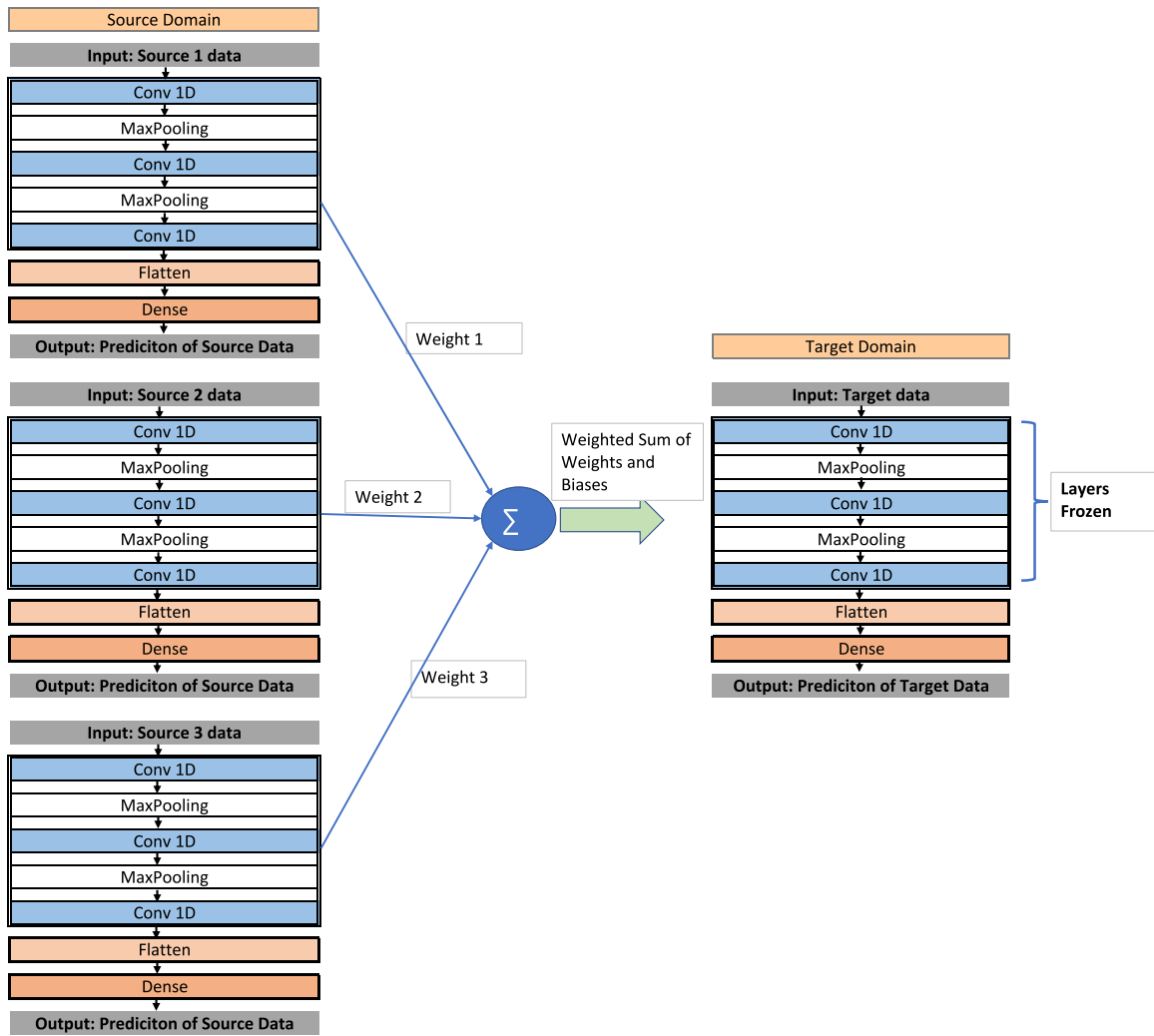


Fig. 7. Multi-source Multi-Layer TL network (MSML-TL).

5. Model implementation

This section presents the implementation of the proposed MSML-TL-RFE approach for the analysis of supply chain data. The implementation is divided into several stages, starting with data collection, followed by feature extraction, data structuring, experimentation design, and finally, the evaluation of the hierarchical supply chain model. Each stage is described in detail in its respective subsection, providing a step-by-step guide for the implementation of the MSML-TL-RFE approach. Overall, this section provides a complete guide to the implementation of the MSML-TL-RFE approach, from data collection to final evaluation.

5.1. Data collection

Before starting to apply input data to the various TL frameworks, it was essential to acquire data from acceptable sources to ensure the credibility of the proposed frameworks. Three very common and publicly available datasets were used for the experimentation purposes of this paper. These datasets are also used by many prominent researchers for their studies. The first dataset (Dataset 1) (Joseph et al., 2022; Kharfan et al., 2021) was taken from Kaggle (link: <https://www.kaggle.com/c/demand-forecasting-kernels-only/data?select=train.csv>) and it is known as the Store Item Demand Challenge. It contains five years (2013–2017) of sales data on 50 items sold over 10 different stores. The aim of the challenge was to make 3 months of predictions and it has the following attributes: date of sales, Store ID, Item ID and the number

of items sold at a given date. In total, the dataset contains 9 130 000 features.

This paper uses only the information of the first 3 stores and their corresponding 10 sold items in each of those three stores. For the initial part of the experiment, the first 9 items from each store were considered source products, while the 10th item (i.e., Item X) is the target domain data.

The second dataset (Dataset 2) was collected from a paper (Mancuso et al., 2021) where the authors used an ML approach, termed as Neural Network Disaggregation (NND), which was developed by combining MLP with CNN for making hierarchical time series forecasting. This particular dataset contains 118 daily time series, which represents the demand for pasta from 01/01/2014 to 31/12/2018 (link: <https://data.mendeley.com/datasets/njdkntcpc9/1>). Apart from being univariate time series data, promotional activities are also linked with the sales of pasta in the dataset.

This paper also uses a third dataset (Dataset 3), commonly known as Rossmann Store Sales data (Pavlyshenko, 2019, 2022), which is also available in the Kaggle competition platform (link: <https://www.kaggle.com/code/mithileysh/time-series-analysis-and-forecasts-with-prophet/data>). The daily sales of different Rossmann stores are described in this dataset. The data also has some main features, such as store type, holiday, and Store Type.

Thus, in this paper, as many as three datasets were used to carry out the experimental analysis. All the downloaded datasets were separated into two parts: source data and target data, according to the previously



Fig. 8. Multi-source Multi-Layer TL network with RFE (MSML-TL-RFE).

mentioned assumptions in Section 3. One particular product from each channel is assumed to be the newly launched product, which makes it the target domain data with limited data (1 month of data). On the other hand, the remaining other products or stores are all considered potential sources of information. The source data was then split into three parts, mainly for training, validation and testing the network. The target dataset was also split in the same way as the source data, but with less training and validation data, while the testing data had 6 months of sales information, which was later used to verify the accuracy of the networks.

Table 2 describes the duration of data from Dataset 1, that was used to obtain the pre-trained network in the source domain and also the duration of data used in the target domain to train, validate and test the TL approaches. This was explicitly done for Dataset 1 for explanation purposes, and the other two datasets were also split in a similar way.

Instead of mentioning the duration, all three datasets are represented in a number of time-steps and their proportions in Table 3. In the source domain, a majority portion of data (around 80%) is used to train the network, while 10% is kept for validation and the remaining 10% is used for testing. Whereas in the target domain, a very small portion of the dataset is used to re-train and validate (around 6%–7% in each task) a pre-trained network, while a significant part (around 85% data) is used for testing the performance of the TL approach.

5.2. Feature extraction

It is essential to extract features from data so that ML/DL has enough information for training. In order to train and re-train the ML/DL models, important features, such as year, month, week and day were extracted. Using python's built-in function to_datetime, year, month, week, and day were acquired, while if other information was

Algorithm 1 Proposed MSML-TL-RFE approach.**Input:** Dataset with available and extracted features, Target Domain Item as per assumption**Output:** Predicted Values of the target domain test set and the test RMSE

```

// KNN to identify three similar items with corresponding
Euclidean Distances
1: S1, S2, S3 = KNN(Dataset - Target Domain Item, Target Domain
   Item)
2: distance1 = euclidean_distance(Target Domain Item, S1)
3: distance2 = euclidean_distance(Target Domain Item, S2)
4: distance3 = euclidean_distance(Target Domain Item, S3)
// RFE to reduce the number of features for target item and similar
products
5: Target_RFE, S1_RFE, S2_RFE, S3_RFE = RFE(Target Domain Item,
   S1, S2, S3)
// Split Data
6: train_data, val_data, test_data = split_data(Target_RFE, S1_RFE,
   S2_RFE, S3_RFE)
// Train independent CNN networks for each similar item
7: CNN1 = train_CNN(S1_RFE_train, S1_RFE_val)
8: CNN2 = train_CNN(S2_RFE_train, S2_RFE_val)
9: CNN3 = train_CNN(S3_RFE_train, S3_RFE_val)
// Combine weights and biases from trained networks to form a
new CNN network
10: weights1, biases1 = CNN1.get_weights()
11: weights2, biases2 = CNN2.get_weights()
12: weights3, biases3 = CNN3.get_weights()
13: sum_distance = distance1 + distance2 + distance3
14: weights_avg = [distance1/sum_distance, distance2/sum_distance,
   distance3/sum_distance]
15: new_weights = [weights1*weights_avg[0] + weights2*weights_avg[1] +
   weights3*weights_avg[2]]
16: new_biases = [biases1*weights_avg[0] + biases2*weights_avg[1] +
   biases3*weights_avg[2]]
17: new_CNN = CNN_model()
18: new_CNN.set_weights(new_weights, new_biases)
// Train the new CNN network using the target domain test data
19: train(new_CNN, Target_train_data, Target_val_data)
20: test_predictions = predict(new_CNN, Target_test_data)
21: test_RMSE = RMSE(test_predictions, Target_true_values)

```

Table 2

Dataset 1: Duration of Source/Target domain data for Training/Validation and testing the TL networks.

Source domain	Duration	Target domain	Duration
TRAIN 1/1/2013– 31/12/2016	3 Years		
Validation 1/1/2017– 30/6/2017	6 Months	TRAIN 1/6/2017– 15/6/2017	15 Days
		Validation 16/6/2017– 30/6/2017	15 Days
Test 1/7/2017– 31/12/2017	6 Months	Test 1/7/2017– 31/12/2017	6 Months

available from the data, it was appended. Among the three datasets, not all of them have the same set of additional information, the first dataset has no information except the sales of stores, while the 2nd set has promotional information. The third dataset has other store-related information, such as promotional activities, holidays, whether a store is open or closed, and customer-related information. All this additional

Table 3

Time-Steps of Source/Target domain data for Training/Validation and testing the TL networks.

Dataset 1	Source domain		Target domain	
	No. of time steps	Percentage	No. of time Steps	Percentage
TRAIN	1461	80.01%	15	6.98%
Valid	181	9.91%	15	6.98%
Test	184	10.08%	185	86.05%
Dataset 2				
Train	1443	80.26%	14	6.73%
Valid	176	9.79%	15	7.21%
Test	179	9.96%	179	86.06%
Dataset 3				
Train	577	61.25%	16	7.55%
Valid	184	19.53%	15	7.08%
Test	181	19.21%	181	85.38%

Table 4

Data structuring.

Independent variables		Target variable
Variable X1	Variable X2	y
$X1_{t-n-days_ahead_prediction}$	$X2_{t-n-days_ahead_prediction}$	$y_{t-n+days_ahead_prediction}$
...	...	...
$X1_{t-2-days_ahead_prediction}$	$X2_{t-2-days_ahead_prediction}$	$y_{t-2+days_ahead_prediction}$
$X1_{t-1-days_ahead_prediction}$	$X2_{t-1-days_ahead_prediction}$	$y_{t-1+days_ahead_prediction}$
$X1_{t-days_ahead_prediction}$	$X2_{t-days_ahead_prediction}$	$y_{t+days_ahead_prediction}$

Where t : current time; n : number of past time steps; $days_ahead_prediction$: forecasting horizon

information is appended as features alongside the previously mentioned extracted `_datetime` features. Thus, the data frame contains at least 4 features for the first dataset, 5 features for the 2nd dataset and as many as 8 features for the 3rd dataset, as shown in Fig. 9, with sales of a particular store item being the target variable.

5.3. Data structuring

A min–max scalar normalisation function (Géron, 2019) was applied to scale the features of the data between 0 and 1. This was done so that it can easily be handled if there are any sigmoid activation functions in the output layer of any neural network. Therefore, the normalisation function was calculated based on the following formula in Eq. (5):

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}; \quad (5)$$

In the next stage, each scaled data frame was structured in a sequential pattern. For multivariate predictions, sales (dependent variable) are based on other independent variables (year, month, week, day, trend and seasonality). Independent variables are termed X , and the dependent variables are y . For each data frame, the data is structured in a 2D array consisting of Independent and dependent variables, according to Table 4.

In this way, three different segments were selected from the data frame for training, validation and testing, using datasets from both the source and target domains. CNN, being a part of neural networks and also being the baseline algorithm used in this paper, X_{train} , X_{val} and X_{test} are converted to suitable 3D arrays using a sliding window approach (Selvin et al., 2017). For each of the subsets of the data frame, the time series analysis of each subset is structured in a sequential pattern using a sliding window technique before feeding it into the network. The parameter `days_ahead_prediction` is adjusted to define the forecasting horizon for training, validation, and testing. For instance, it is set to 1 for short-term prediction, 15 for mid-term forecasting, and 30 for long-term forecasting across all phases. In the experiments, a window of size 10 (working days of two consecutive weeks) is used to predict one-day ahead (short term) and 5-day ahead (mid-term) sales information.

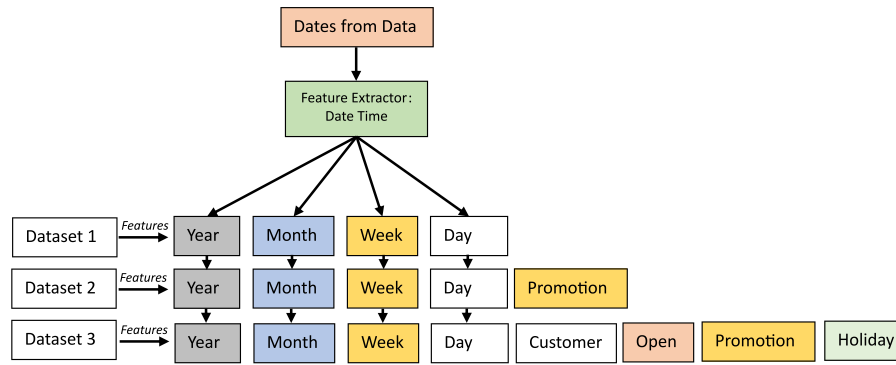


Fig. 9. Extraction of features from data using date-time feature extraction method.

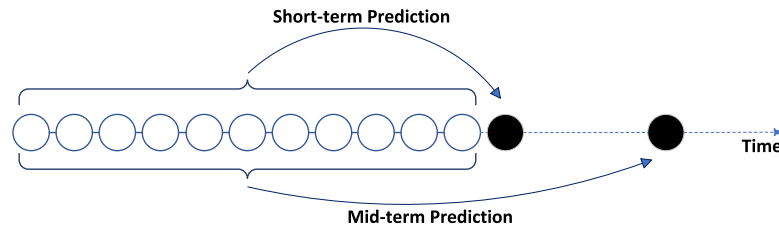


Fig. 10. Illustration of short-term and mid-term prediction.

Fig. 10 illustrates the predictors (input values) and the target (output) while making short-term and mid-term predictions. An input vector of 10-time steps and numerous columns of data features is formed. The columns consist of the date time features, as well as other available features of the data, such as holiday or promotional activities. Using these columns of information, data is structured as the independent variables (X train) (10 days of features and past sales information) and the dependent variables (Y train), which is either the 11th day's (short term) true sales or 15th day's (mid-term) sales, depending on the type of predictions the model is making. The window is then shifted one step along, and this time, it contains information from the 2nd time step until the 11th time step of the training data and the 12th day's data as the output of that particular segment while making short term predictions. The model can also be used to predict any time between 1 and 5 days ahead into the future. Since it is assumed that there is limited data available in the target domain, hence, as a limitation, in this paper only 5 days ahead of prediction is made using this system.

5.4. Experimental design for TL approaches

Sales of the target items were initially predicted using the simple base CNN model mentioned in 4.1. In this approach, No-TL was used for predicting the sales pattern of the assumed target item. From there, the system was customised to the SS-TL method. A single similar item was identified based on one month of sales pattern of the target data and then a matching item was found during that duration from other products for the same retailer. For more accurate prediction, the system was further upgraded to be a multi-source TL network where multiple items (at least 3 or more) were identified from the same retailer in order to obtain the pre-trained models and then was re-trained and the model was tested with target data. It was later extended to simulate the benefits of information sharing in a SC network. The items from other retailers were also considered in the TL approaches.

5.4.1. Identification of similar products using KNN

KNN is a very popular approach often adopted by researchers to determine similar objects (Kuźelewska, 2022; Tao et al., 2022). To quantify the similarity between different items or stores, the distance

between them was calculated. In this paper, Euclidean distance was calculated for finding out the similarity between two items. Using this, items that have similar sales patterns are often closely grouped together at a smaller distance, whereas dissimilar items are situated further apart. All the available features of each dataset were used with the KNN algorithm for the determination of closely related items.

From each store, the first 10 items were chosen, while it was assumed that the 10th item was the targeted newly introduced market product. This made the other 9 items, with their extracted and available features, potential sources of information to compile a KNN network for finding the most appropriate information source. It is worth mentioning that, considering the practicality of the situation, it was assumed that the target product had been in the market for only 30 days, which means that there was limited sales data available for analysis. Using that 30 days' worth of sales data in the target domain, the corresponding sales data of other items were extracted during that period. Thus KNN calculates the distance between items having 30 days of data among which only the top 3 closest items are considered for further analysis using the TL approaches.

5.4.2. Prediction of newly launched item

For the first two datasets, it was assumed that the 10th item from each retailer/brand was the newly launched product, while for the 3rd dataset, it was presumed that in a certain region, the 10th store had recently started its operation with nine other stores in the same region. There were also other groups of stores carrying out operations in other regions of a particular country. To establish the effectiveness of the TL approaches, the first 10 items related to one particular store from the first two datasets were selected, and from the 3rd dataset, the sales patterns of the first 10 stores were chosen for analysis. As mentioned previously, the 10th item or the sales of Store 10 or Item 10 were chosen as the new market being introduced, while it was assumed that other items or stores were in operation for quite some previous time in the market, which means the entire duration of the datasets were considered as potential sources.

For Datasets 1 and 2, initially without any TL approach (No-TL), where a base CNN approach was used to predict six months of sales from the target Item 10 (refer to Fig. 1 Item X) of Store 1 or Brand 1 only. For Dataset 3, the sales of 6 months of sales of Store 10

Table 5

Euclidean distance and weights of closest three Items/Stores in relation to channel 1 target Item/Stores for 3 datasets.

	Source domain Items/Store		
	Source 1	Source 2	Source 3
Dataset 1	Store 1 Item 7	Store 1 Item 8	Store 1 Item 2
Distance	101.58	103.50	115.28
Inverse distance weights	0.3493	0.3428	0.3078
Dataset 2	Brand 1 Item 4	Brand 1 Item 6	Brand 1 Item 8
Distance	24.98	26.85	26.85
Inverse distance weights	0.3496	0.3252	0.3252
Dataset 3	Store 6 Sales	Store 2 Sales	Store 1 Sales
Distance	4901.97	5767.16	5895.77
Inverse distance weights	0.3729	0.3170	0.3101

were forecasted. Then, the TL approaches are introduced gradually. For the TL systems to come into effect, it was essential to identify similar source data from the remaining 9 items or store sales. For the identification of closely related items or stores, KNN was used with all the extracted available features of data mentioned in Fig. 9. Then, according to the distance, the first three items were selected from each dataset, especially for Channel 1. The distance and weights of each of the datasets in relation to the target item or brand or store are presented in Table 5.

From Table 5, it can be seen that for Dataset 1, Items 7, 8 and 2 are the items from Store 1 that are the three closest to Item 10 of the same store. The distance of Item 7 from Item 10 of store 1 is 101.58, while Item 8 is the nearest to Item 10 with a separation of 103.50 units. Item 2 is the next closest item, with a distance of 115.28 units. The inverse weights are as follows: Item 7 has a weight of 0.3793, Item 8 has 0.3428, and the last Item 2 has 0.3078. The sum of all the weights is equal to 1. With this Inverse Distance method, the closest item with minimum distance gets the highest weight and the items with higher distance gradually have lower weights assigned.

The 2nd dataset (Dataset 2) is also similar to Dataset 1 but contains a bit of extra information, such as promotional activities. Here, the KNN algorithm determines the three similar products belonging to Brand 1 that are closest to the target product (Item 10 of Brand 1). The closest Items are Items 4, 6 and 8, having distances of 24.98, 26.85 and 26.85, respectively, while the corresponding weights are 0.3495, 0.3252 and 0.3252. The 3rd dataset is a little bit different, with only the overall sales information of each store rather than the itemised sales pattern. Assuming the fact that the first 10 stores (Store 1 to Store 10) are geographically located in the same region (Region 1), while the next 10 stores (Store 11 to Store 20) are located in a different region (Region 2) and from Store 21 to Store 30 are situated in a 3rd region (Region 3). Considering the first 10 stores of Region 1, Store 10 was the assumed target store, with Stores 6, 2 and 1 being the closest source stores as per the KNN algorithm. The associates' distances and weights are also stated accordingly in Table 5. The identified sources were then used in the TL approaches for the prediction of Target domain data.

5.4.3. Simulated SC scenario 1: Without information sharing

In the context of supply chains, there is a usual assumption among retailers that they want to keep their business information secret. Keeping that in mind, a retailer can use its own products to find similar source products and then use the TL approach to make predictions for items that are newly introduced by that retailer. An outline of the

Table 6

Euclidean distance and weights of closest three Items/Stores in relation to target Item/Stores for 3 datasets with information sharing.

	Source domain Items/Store		
	Source 1	Source 2	Source 3
Dataset 1	Store 3 Item 6	Store 2 Item 9	Store 3 Item 7
Distance	63.98	68.16	70.94
Inverse distance weights	0.3520	0.3305	0.3175
Dataset 2	Brand 1 Item 4	Brand 2 Item 3	Brand 3 Item 2
Distance	24.98	25.98	26.00
Inverse distance weights	0.3422	0.3290	0.3288
Dataset 3	Store 23 Sales	Store 14 Sales	Store 6 Sales
Distance	3723.62	3978.98	4901.97
Inverse distance weights	0.3710	0.3472	0.2818

previously mentioned TL approaches explored in this paper is presented in Fig. 11.

According to Fig. 11, the source and target data all belong to Store 1 of Dataset 1 due to the consideration of the fact that Stores do not wish to publicise their sales information to other retail outlets. In this case, while training the single source approach, only the best or the closed source data is considered. For the remaining three multi-source TL approaches, all three are considered, and the ultimate effectiveness of each of these approaches is evaluated using the test set of the target domain data, which is 6 months of Sales data of Store 1 Item 10.

With the remaining two datasets (Datasets 2 and 3), the same approach was used but with different source data for training the different networks. The target data is also changed accordingly as for Dataset 2, the target data is Item 10 of Brand 1 and for Dataset 3, it is the Sales of Store 10, which is considered as the Target data. Likewise the source domain data is also changed while dealing with Dataset 3. Please refer to Table 5 for the source data of each of the datasets. All the experimental analyses were carried out to establish that obtaining knowledge from other sources enhances the TL capability over systems without the implementation of the information-sharing concept.

5.4.4. Simulated SC scenario 2: With information sharing

The same three datasets were used to simulate the information-sharing concept. For demonstration purposes, the target data of each of the datasets was the same, while the source data were now selected based on a larger pool of source data. This was intentionally done to establish whether choosing more similar sources leads to a more accurate TL prediction system and also enhances the overall effectiveness of TL approaches when information is shared among stakeholders. The source data with distance and weights of the different source data are presented in Table 6.

The concept of information sharing among stakeholders is simulated in this section. Assuming that competitors share data with each other creates a larger pool of data to identify closer sources. From Table 6, it can be seen that three closer sources are identified for the same target domain data for each of the datasets. For Dataset 1, the closest item to the target item (Store 1 Item 10) is Item 6 of Store 3. The next closest item is Item 9 from Store 2, and the third closest item is Item 7 from Store 3. The distance and weights are assigned according to the Euclidean Distance from KNN. For Dataset 2, the closest items are not confined to a particular brand but now similar items from other brands are also considered potential sources of information. In the case of Dataset 3, Stores from another region can also be used to extract important information to train models.

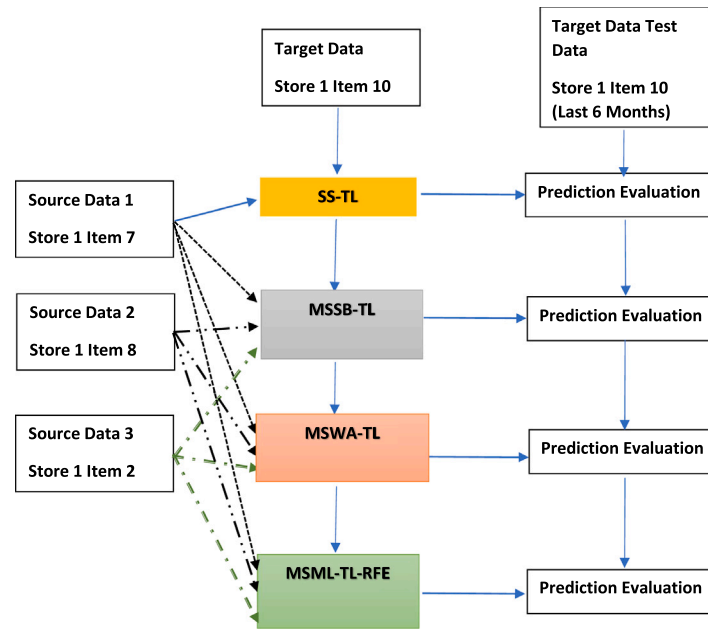


Fig. 11. Prediction of newly launched without info sharing using different TL approaches.

Table 7

Average RMSE for different approaches using 3 Datasets, with and without information sharing.

Approaches	No-TL	SS-TL		MSWA-TL		MSSB-TL		MSML-TL-RFE	
		Without info-sharing	With info-sharing	Without info-sharing	With info-sharing	Without info-sharing	With info-sharing	Without info-sharing	With info-sharing
Dataset 1	0.2067	0.1938	0.1842	0.2035	0.1932	0.1976	0.1901	0.1763	0.1736
Dataset 2	0.1049	0.1120	0.1120	0.1081	0.1104	0.1119	0.1120	0.1099	0.1075
Dataset 3	0.2833	0.3145	0.3070	0.3018	0.3027	0.2895	0.2891	0.2873	0.2856

5.5. Effectiveness of the TL predictions in the context of a Hierarchical SC Model

The predictions from different TL approaches were practically considered by developing a Hierarchical SC Model. The model calculates costs for a newly launched item along each Channel (Refer to Fig. 1 for more information). The presented architecture has 3 channels for each of the datasets. For the first dataset (Dataset 1) The target domain data for Channel 1 is Store 1 Item 10, for Channel 2 is Store 2 Item 10 and for the 3rd channel (Channel 3) it is Item 10 of Store 3. Similarly, it can be deployed over the other two datasets as well for cost calculation, which can be implemented in the future to observe the cost of new business using different channels. For Dataset 2, over three channels, the probable three Target domain data are as follows: Channel 1: Brand 1 Item 10, Channel 2: Brand 2 Item 10 and Channel 3: Brand 3 Item 10. In the last dataset (Dataset 3), the target domain probable stores for the future calculation can be Store 10, Store 20 and Store 30 as Channel 1, 2 and 3, respectively. Considering the target Items of each channel, the Source Items are identified accordingly using KNN. For the time being, only Dataset 1 is taken into consideration for such cost estimation over different channels.

6. Result analyses

This section provides a comprehensive analysis of the results obtained from applying the proposed MSML-TL-RFE approach to three datasets. The purpose of this analysis is to validate the suitability of the MSML-TL-RFE approach for the analysis of supply chain data. In the following subsections, the results obtained from the application of the MSML-TL-RFE approach are presented in detail. This is followed by a behavioural analysis and theoretical derivation of the findings,

which will shed light on the underlying mechanisms and relationships in the data. Finally, managerial implications are discussed based on the results, providing practical guidance for decision-makers in the supply chain domain. Overall, this section presents a thorough and comprehensive evaluation of the MSML-TL-RFE approach and its potential for use in the supply chain domain.

6.1. Result analyses using online datasets

As many as 4 different TL approaches are explored in this paper, over three datasets, demonstrating two SC scenarios. Limited target data was presumed for predictions, which led to adapting knowledge from other domains. For the simplicity of the analysis and to show the potency of the TL approaches, target domain data was initially limited to one particular channel for all three datasets. This was later extended to three channels for more analysis on using an SC model. The average RMSE from each of the approaches over individual datasets, demonstrating the value of information sharing, is presented in Table 7.

Based on the data in Table 7, a comparison between the different approaches used in the experiments for each of the three datasets is also shown in Fig. 12. It also highlights the impact of information sharing on the performance of the approaches.

To evaluate the performance of each of the models, 1 to 5 days ahead of predictions was obtained from each individual model. For each day ahead of prediction, multiple different experiments were performed, and the mean predicted value was calculated. The predicted mean value was then compared against the true target domain test data, and the corresponding RMSE was calculated for performance evaluation. From Table 7, it can be seen that in the majority of the datasets, information sharing improves the accuracy of the cases.

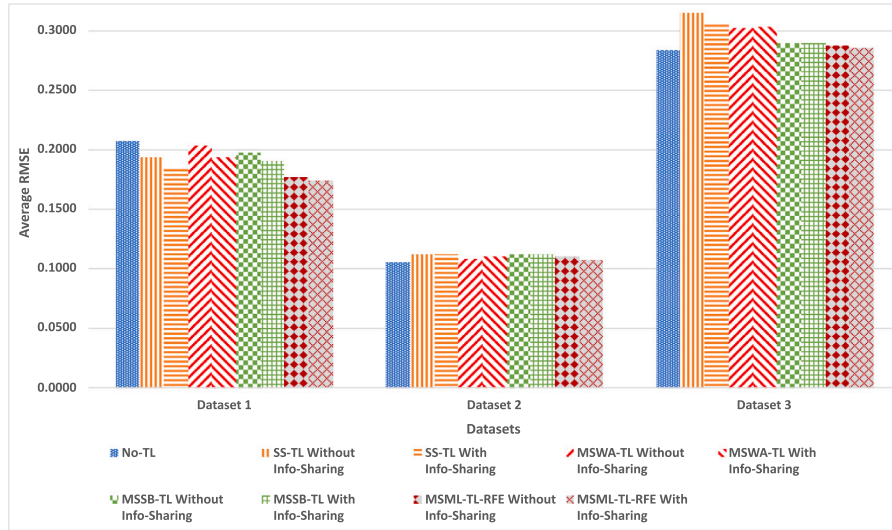


Fig. 12. Average RMSE for different approaches using 3 datasets, with and without information sharing.

Table 8

Average RMSE of different approaches, with and without information sharing.

Forecasting approaches with limited target data	No-TL	SS-TL	MSWA-TL	MSSB-TL	MSML-TL-RFE	Mean RMSE
Without Info	0.1983	0.2068	0.2044	0.1996	0.1911	0.2005
With info	0.1983	0.2011	0.2021	0.1971	0.1889	0.1937

Moreover, the average RMSE for SS-TL for both without and with information sharing (0.1938 and 0.1842, respectively) is lower than the No-TL approach, especially for Dataset 1, which is 0.2067. However, for Datasets 2 and 3, the RMSE values are slightly higher with information sharing. While information sharing enhances accuracy in some cases, its effectiveness slightly varies across different datasets. For the latter two datasets, SS-TL has comparable results with the No-TL system, even though the RMSE of both No-TL and SS-TL approaches are slightly on the higher side. This may be due to the fact that Datasets 2 and 3 have extreme changes in the data, which needs further investigation. With the addition of multiple sources in the TL architectures, the average RMSE for each dataset gradually decreases. The overall effect can be observed from the average RMSE of the three datasets for all approaches, as presented in Table 8. Statistical tests were also carried out using the average RMSEs to demonstrate the significance of the TL approaches. In the next phase of the experimentation, the performance of the different TL approaches was investigated over two abstract SC scenarios.

The information in Table 8 is used to visually represent the differences in average RMSE values for the different approaches, with and without information sharing, which is presented in Fig. 13. It enables comparison between the different datasets and helps identify trends or patterns in the accuracy of different approaches.

From Table 8 and Fig. 13, it can be deduced that an average RMSE of 0.2068 is produced with the SS-TL system, which is the highest when no information has been shared by the retailers. Then, with the addition of numerous sources, the RMSE gradually decreases. The MSWA-TL approach produces an RMSE of 0.244, while the MSSB-TL approach is the next best in terms of an RMSE of 0.1996. The MSML-TL approach with RFE (MSML-TL-RFE) and Frozen Layers produces the lowest RMSE of 0.1911, which is the best among all approaches. The TL approaches are also compared against the non-TL approach, which produced an RMSE of 0.1983.

When the stakeholders agree to share sales information with each other, closer sources can be identified for better knowledge harvesting. With more similar sources, all the previous 4 TL models are trained

as before. The RMSE produced by each of the models is relatively lower than the previous set of RMSEs, where no information was shared. The SS-TL approach has an RMSE of 0.211, while the MSWA-TL technique has an RMSE of 0.2021. Both these two RMSEs are lower than their predecessors. The MSSB-TL and MSML-TL-RFE approaches also produced comparatively lower RMSE of 0.1971 and 0.1889, respectively. The same RMSE was reported with the No-TL approach as it has no sharing of information. From observing the mean RMSE of the two SC scenarios, with and without information sharing, it is evident that sharing information among the retailers leads to more accurate predictions, regardless of the considered TL approach. The average RMSE without information sharing is 0.2005, while with the sharing of information, it drops to 0.1937. Hence, sharing of information adds value to the knowledge acquired from closer sources, as it gives more freedom to the TL approaches to gain knowledge from a closer source.

6.1.1. Statistical analysis

In this subsection, two statistical tests were conducted to establish the significance of the different TL approaches presented in this paper. Using the RMSE obtained from the approaches, a Friedman Test was first carried out to find the rank, and then, taking the best three approaches, a Wilcoxon Signed Rank Test was conducted to establish the significance of the TL frameworks. Both the two SC information-sharing concepts were taken into consideration, and the tests were conducted on the overall RMSEs of each of the approaches on all 3 datasets. Table 9 shows the results of the mean rank of each method obtained from the Friedman Test.

It is evident from Table 9 that the best-performing TL Approach is the MSML-TL-RFE architecture, which has the lowest mean rank of 2.60 (in bold). Apart from that, the next two best-performing approaches are MSSB-TL and No-TL, which have mean rank values of 2.88 and 2.93. Taking the best three approaches into consideration, Wilcoxon Signed Rank Tests were carried out to find the significance of the TL systems. The results of the Wilcoxon test are reported in Table 10.

The Wilcoxon Signed Rank Test was conducted on the RMSEs of the best three approaches. Using a 5% significance level, one of three

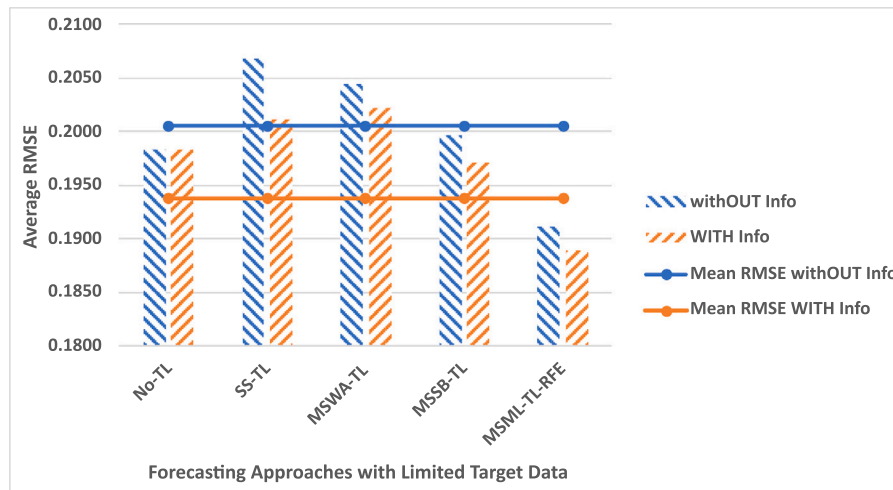


Fig. 13. Average RMSE of different approaches, with and without information sharing.

Table 9

Friedman Test: Mean rank.

TL Approaches	Mean rank: Combined dataset	Position (Combined dataset)
No-TL	2.93	3
SS-TL	3.18	4
MSWA-TL	3.41	5
MSSB-TL	2.88	2
MSML-TL-RFE	2.60	1

Table 10

Results of the Wilcoxon test using RMSE-testing.

TL Approaches	Better	Similar	Worst	p-value	Decision
MSSB-TL vs. No-TL	150	0	150	0.38	≈
MSML-TL-RFE vs. No-TL	177	0	123	0.00	+
MSML-TL-RFE vs. MSSB-TL	161	0	139	0.005	+

signs (+, – and ≈) was assigned, where the “+” sign means the first algorithm is significantly better than the second one, “–” means that it is significantly worse and “≈” represents that there is no significant difference between these approaches. According to Table 10, the test between MSSB-TL and No-TL method is not statistically significant as it produced a p -value of 0.38. The other two tests conducted between the MSML-TL-RFE and No-TL Approaches, and with MSML-TL-RFE and MSSB-TL, produced p -values of 0.00 and 0.005, respectively, which are much less than 0.05. Thus, the statistical test reveals that the proposed MSML-TL-RFE is statistically better against its two nearest competitive approaches when RMSE is the performance measure.

6.2. Application of TL on the designed Hierarchical SC Model

From Section 6.1, it is quite clear that the best-performing TL approach is the proposed MSML-TL-RFE system. The predictions from that best approach with predictions from two other approaches, namely the No-TL and SS-TL approaches, were considered for cost analysis while using the Hierarchical SC Model. Using these approaches, up to 5 days ahead of predictions were obtained for all three channels of one particular dataset (Dataset 1). It is worth noting that for each individual channel, the target domain item or store was assumed as the newly launched product with limited data and then keeping that assumption fixed, similar items are identified using KNN as before. The predictions against each target domain data for 3 different channels were then fed into the SC system for cost analysis.

Table 11

SC Cost (\$) for Fixed LT over 3 Channels using Dataset 1 with information sharing concept.

SC cost (\$)	Channel 1 fixed LT 5 days	Channel 2 fixed LT 5 days	Channel 3 fixed LT 5 days	Avg SC cost
No-TL	32 573.55	45 940.51	40 564.12	39 692.73
SS-TL	32 797.38	45 823.47	40 589.15	39 736.67
MSML-TL-RFE	32 298.28	45 180.57	39 959.41	39 146.09

6.2.1. Fixed lead time with variable safety stock

In the SC model, two different aspects are considered. One with fixed or dynamic lead time, while the other is with variable safety stock. The safety stock is considered to vary in 4 steps: No safety stock, 95% to 99% and 95.9%. The analysis of 4 different safety stocks and the average cost is reported in Table 11 for a fixed lead time of 5 days (see Fig. 14).

From Table 11, it can be seen that the more accurate approach creates a lower SC cost over all three channels. As the SS-TL system is less accurate than the No-TL approach, it produces comparatively higher costs of \$32 797.38 and \$40 589.15 for Channel 1 and Channel 3, respectively, while it is slightly lower (\$45 823.47) for Channel 2, i.e. Item 10 from Store 2 is considered as the target domain data. The MSML-TL-RFE method proposed in this paper is the most accurate one and produces the lowest overall SC cost (presented in bold) of all three channels. The overall average SC cost for the MSML-TL-RFE approach is also the lowest (\$39 146.09), if the average value is considered.

6.2.2. Dynamic lead time with variable safety stock

Changing the SC model to work in situations where the delivery lead time (LT) is dynamic (where it varied between 3 to 5 days), the same predictions from different models were fed into the SC models for cost calculations. This is often a more practical scenario because retailers and suppliers have some uncertainty in managing logistics. Thus, such a type of cost calculation is done in the case of newly launched products with variable safety stock to minimise stock out, and the result is presented in Table 12.

Unlike Fixed LT, where the SC model considered all 5 days ahead of predicted data, here the model did not consider all 5 days of prediction every time, rather it considered only 3 days of LT, and then there was a delay of an additional 1 to 2 days, mimicking the uncertainty in practical fields. It can be seen from the table that the SC model produces a lower cost for the SS-TL approach than the No-TL approach because the SC system, with safety stock in place, reduces the chance of stock out. Hence, the average cost produced by SS-TL over three

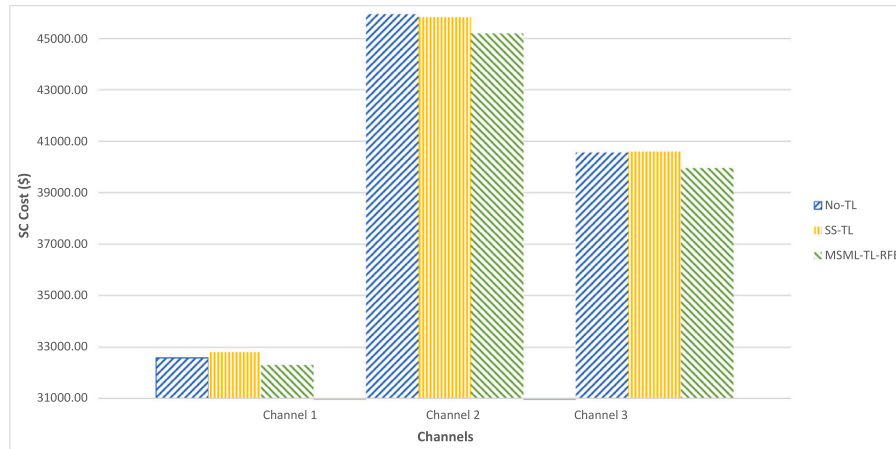


Fig. 14. Supply chain cost (\$) over 3 channels of SC having fixed LT.

Table 12

SC Cost (\$) for Dynamic LT over 3 Channels using Dataset 1 with information sharing Concept.

SC cost (\$)	Channel 1 Dynamic LT 3+2 days	Channel 2 Dynamic LT 3+2 days	Channel 3 Dynamic LT 3+2 days	Avg cost
No-TL	32 945.41	46 785.29	41 037.82	40256.17
SS-TL	32 738.96	46 658.70	40 844.37	40 080.68
MSML-TL-RFE	32 692.13	46 274.51	40 762.72	39 909.79

channels of the dynamic SC model is \$40 080.68, which is lower than the No-TL approach of \$40 256.17. As before, the most accurate approach is MSML-TL-RFE, and it consistently produces the lower SC cost (presented in bold), even in situations where the replenishment LT is dynamic. Its overall average SC cost of \$39 909.79 is the lowest among all of the approaches presented in Table 12. Comparing Tables 11 and 12, the average cost produced by MSML-TL-RFE is relatively higher for dynamic LT, as the system has to incur some additional cost when the uncertainty in delivery time is considered, which is a more practical representation of a real system.

6.3. Behavioural/sensitivity analysis

From previous experimental analysis and statistical tests, it can be concluded that the MSML-TL-RFE approach, having three sources with an information-sharing concept, provides the best result. It was further tested by adding a few more sources. Since the best-performing TL system has the capability of learning from three sources, its sensitivity was further evaluated by the addition of a few more sources. The system was tested with 6 sources and also with 9 sources with the information-sharing concept enabled. The time needed and the accuracy of each of the approaches is reported in Table 13.

The experiment was also conducted with the No-TL and SS-TL approaches without sharing information as benchmark approaches for comparison. It can be seen from Table 13 that predictably, learning from multiple sources takes a longer time as compared to learning from a single source. When it comes to accuracy, a larger number of sources slightly increases accuracy while taking a significant amount of extra time for the models to extract information from multiple sources. However, when the number of sources is increased from 3 to 6, accuracy actually decreases, while increasing it further slightly improves accuracy, showing some indications of negative knowledge transfer. However, it takes significantly longer (3519 s) to train, while with 3 sources, it takes only 1304 s with a more moderate accuracy of 5.668.

Another way of checking the sensitivity of the best TL approach (MSML-TL-RFE) is by changing the type of pool of knowledge sources

Table 13

Time and accuracy of different approaches for Dataset 1.

	Time(s)	Accuracy (Reciprocal RMSE)
Expt 1 No-TL	49	4.8340
Expt 2 SS-TL w/o info sharing	458	5.1749
Expt 3 MSML-TL-RFE With info sharing having 3 sources	1304	5.6668
Expt 4 MSML-TL-RFE with info sharing having 6 sources	2487	5.5102
Expt 5 MSML-TL-RFE with info sharing having 9 sources	3915	5.7538

Table 14

Performance of MSML-TL-RFE approach with different number of similar source items from same group.

	Without info sharing	With info sharing (3 Items/Stores from same group)	With info sharing (9 Items/Stores from same group)
Dataset 1	0.1763	0.1764	0.1722
Dataset 2	0.1099	0.1083	–
Dataset 3	0.2898	0.2814	0.2891
Mean RMSE	0.1920	0.1887	–

for transfer. In all of the previous experimentation, the 10th Item or the sales of the 10th store was assumed to be the target item for each channel, but in this section, the type of target item is treated as the determining factor in selecting potential sources from the pool of items or stores. Considering the same type of item or stores, information is now extracted from the same type or category of items sold or stores (including the 10th item of other channels), and the result of the analysis is presented in Table 14.

When no information is shared among the retailers, then at times, it is difficult to find items that are of a similar type, given the availability of limited information in Datasets 1 and 2. For Dataset 3, there are other criteria that were used to select the source items belonging to the type or category of different stores. In this section, for Dataset 3, source stores belonging to the same category as that of target Store 10 (which is Category 'a', as mentioned in Dataset 3) were identified to initiate the knowledge transfer process. When the information sharing concept was taken into consideration, the 10th item of each store was selected

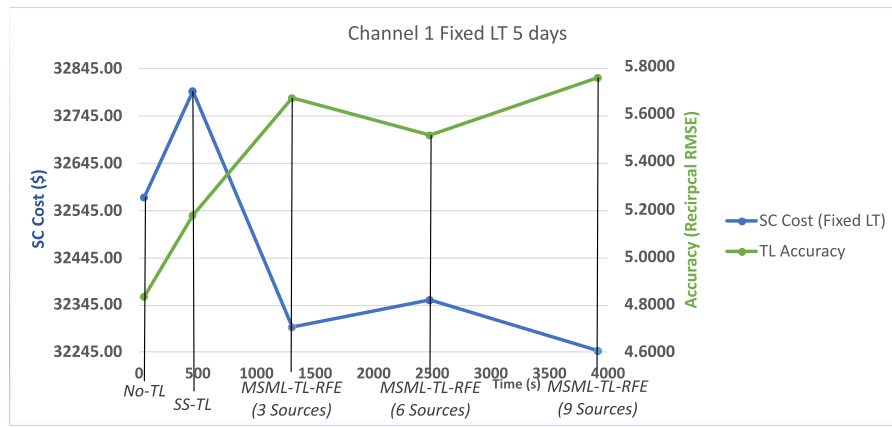


Fig. 15. Supply chain cost (\$) vs. Training Time vs. prediction accuracy of channel 1 of SC having fixed LT.

as the source items for Dataset 1, since, according to the dataset, they have the same corresponding characteristics as Target Item 10, which means for Target Store 1 Item 10, the corresponding available sources are Store 2 Item 10, Store 3 Item 10, Store 4 Item 10, etc. Dataset 2 is about pasta of different brands and types. Item 10 of one brand of pasta corresponds to Item 10 of other brands (e.g. Brand 1 Item 10 corresponds to Item 10 of Brand 2 and Item 10 of Brand 3) and were considered as the same category items as they are of the same type. This means for Dataset 1, the same item 10 is being sold at different stores while for Dataset 2, it is the same pasta having different brands. Considering a real-life scenario, for Dataset 1, Item 10 may refer to the same jacket sold across different stores having different locations whereas, for Dataset 2, it could be pasta of different shapes and flavours such as spaghetti, rigatoni, or lasagne that can be produced by different food manufacturing companies, like Barilla, DeCecco or Rana. So if spaghetti is the target item, then only sales of spaghetti from different brand companies are considered for information transfer. For the 3rd dataset (Dataset 3), stores belonging to Category 'a' (since it matches with the sales pattern of the target Store 10) were taken into consideration while similar stores were identified for harvesting knowledge.

From Table 14, it can be seen that when trying to deal with items belonging to the same group or category with the assumption of information sharing, the MSML-TL-RFE approach produces lower RMSE than without information sharing and with 3 sources. When 9 same category sources are considered, there is inconclusive evidence of improvement as there is missing information for Dataset 2. In Dataset 2, only 4 Brand level information is available, while it requires 9 Brand levels of information to carry out the experiment successfully, e.g. as discussed earlier, sales of spaghetti from four food manufacturing companies are available, but for successful experimentation, spaghetti sold by nine companies are required. That is why some spaces are left blank (-). Thus in terms of average RMSE, if there is available information regarding same-category items, then those could be considered as more accurate sources of information, which could lead to better prediction in the target domain.

6.4. Theoretical derivation

Considering the SC costs from Dataset 1, its corresponding accuracy and time for different approaches from Tables 11–13, two practical suggestions can be made. SC costs from Tables 11 and 12 show that decision makers can select the appropriate channel and continue selling the recently launched product based on expected cost. In this case, for the same SC parameters, Channel 1 produces the lowest cost, which means that it is safe to carry out selling the new product, while Channel 3 produces a comparatively higher cost for the same product. Channel 2 is the most expensive and risky channel for business practitioners to

carry out operations. Thus, this approach can help managers to make an informed decision regarding the continuation of selling their newly launched product through different channels in the market, which is basically the first of the two theoretical contributions.

Apart from the SC costs, if the Time and Accuracy of different TL approaches, having a different number of sources of information presented in Table 13, are taken into consideration, then by the combination of all three factors together, another practical suggestion can be made. Figs. 15 and 16 are plots of Cost vs. Accuracy vs. Time plotted using information from Tables 11, 12 and 13, respectively. Analysing Table 13, it can be seen that the multi-source approaches were not limited to 3 sources, but further experimental analyses were conducted with a great number of sources with source number increasing to as many as 6 and 9.

From Figs. 15 and 16, it is readily apparent that the No-TL approach produces higher cost with lower accuracy, but is fastest for both cases. As compared to the No-TL approach, SS-TL produces better accuracy and lower cost, while MSML-TL-RFE with 3 sources produces significantly better accuracy with even lower cost, although it takes a considerable amount of time to produce results when compared to the previous two approaches. Increasing the source size to 6 and 9 slightly improves accuracy with lower supply chain cost but also takes a significant amount of time to produce results, which may not be practical in some cases where it is difficult to collect that many sources for knowledge transfer. Based on these findings, managers and practitioners can choose between different operating models. They can select the No-TL approach for faster but less accurate predictions or SS-TL for better predictions at the cost of extra time. If they have more time to process data, MSML-TL-RFE with 3 sources can provide much more accurate predictions and lower supply chain costs. However, if they have a very large pool of data, they can identify more sources, which will take even more time to produce results with very little reduction in supply chain cost, as compared to selecting 3 sources for harvesting information.

6.5. Managerial implications

The study related to sales predictions using AI with TL is especially beneficial for practitioners who often deal with limited data. Since the restricted data makes DL approaches less accurate, the TL approach can solve this problem by predicting short-term to mid-term fluctuations with better accuracy. This enables decision makers to see a better picture of how a newly launched product will be treated by customers in the future. They will be able to analyse the viability of a new business model with such cases. With the advances in AI with TL, they will have the advantage of calculating future expenses and adjusting their pricing policies and inventory management. They could also think of future investments with smaller risks while carrying out operations

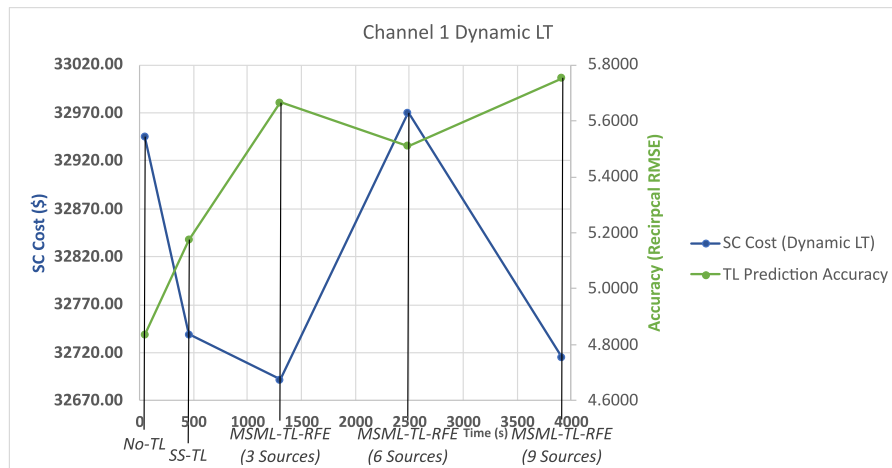


Fig. 16. Supply chain cost (\$) vs. Training time vs. Prediction accuracy of channel 1 of SC having dynamic LT.

in multiple channels of an SC. Supply chain forecasting enhances operations by planning for unforeseen events, delays, and changes in demand. One may reduce costs, satisfy customer demand, and maintain the efficiency of a supply chain by leveraging data and other insights to make smarter decisions, even with little data in the target domain to be used to predict items. Organisations can make more profit by earning consumers' confidence in their newly launched item, and the overall productivity increases will ultimately create a positive impact on Gross Domestic Product (GDP).

7. Limitations and possible future research areas

The research presented in this paper, while offering significant advancements in multi-source transfer learning, has certain limitations that could be addressed in future work. One limitation lies in the fixed parameter settings used throughout the experiments, even after extensive parameter tuning. While effective, the absence of a comprehensive grid search for optimal parameter settings may have constrained the predictive performance of the models. Incorporating RFE into other transfer learning approaches, such as LSTM-, GRU-, or RNN-based frameworks, also remains unexplored and represents a potential area for future investigation to benchmark and extend the versatility of transfer learning techniques. Likewise, in identifying similar products for knowledge acquisition, Euclidean distance was utilised. Future research could explore the effectiveness of knowledge transfer using alternative distance metrics, such as Manhattan distance, Hamming distance, or Cosine similarity.

Another limitation is the sequential nature of the model training process, where individual models were trained on each source domain one at a time, including in the multi-source approaches. This sequential training process, while effective, is computationally intensive and time-consuming. A parallel training strategy, where multiple models can be trained simultaneously, could significantly reduce training time without compromising accuracy. The results obtained from this research demonstrate the potential of transfer learning to improve predictions in scenarios with limited target data, highlighting the importance of source similarity and diversity. Future research should focus on optimising computational efficiency, extending the applicability of the proposed methods, and exploring additional architectures and training strategies to enhance the scalability and impact of multi-source transfer learning.

8. Conclusion

Limited data has always been a challenge for supply chain managers, especially when it comes to the decision-making process regarding whether to continue with a newly launched product or not. TL,

a part of ML, can be used in such situations to predict the demand for such a new product. This paper harvests knowledge from similar products that have similar sales patterns while considering the short duration of the product in the market. Then, using knowledge from similar sources, pre-trained models are re-trained in the target domain to make predictions about the newly launched product. Knowledge from a single source, as well as multiple sources have been investigated, and two different SC ideas were simulated, considering different information-sharing concepts. Using MSML-TL-RFE with three sources enhances accuracy from 4.834 (without TL) to 5.6668, and further rises to 5.7538 with additional sources, facilitating more precise predictions and lowering SC costs for businesses. After extensively comparing the RMSEs of multiple TL approaches over 3 different datasets and also through statistical tests, it has been shown that the MSML-TL-RFE approach was the best TL approach among five different approaches. Moreover, an SC model was used to demonstrate how the amount of cost for a business can change based on various approaches.

All approaches presented in this paper, including the proposed multi-source multi-layer TL approach, were carried out with a fixed set of parameters after extensive parameter tuning. However, for each set of experiments, a grid search to find the best parameter setting could provide a better result, which could be investigated in the future. The addition of RFE to other TL approaches, such as LSTM, GRU, or RNN-based TL approaches, along with the ones already mentioned in this paper, could also be done to examine the performance of the TL models. Regarding the training time of the TL systems, a single model was trained with a single source at a time. This was also true for the multi-source approaches, where each of the sources was selected sequentially to train a base model to obtain the pre-train models, which can be considered as a limitation of the proposed approach. In the future, if multiple models can be trained simultaneously, then the same accurate results could be obtained in a very short time.

CRedit authorship contribution statement

Supriyo Ahmed: Conceptualization, Methodology, Software, Data curation, Visualization, Investigation, Validation, Writing – original draft. **Ripon K. Chakraborty:** Conceptualization, Methodology, Resources, Visualization, Investigation, Validation, Supervision, Writing – review & editing. **Daryl L. Essam:** Conceptualization, Investigation, Validation, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

- Abualigah, L., Zitar, R. A., Almotairi, K. H., Hussein, A. M., Abd Elaziz, M., Nikoo, M. R., & Gandomi, A. H. (2022). Wind, solar, and photovoltaic renewable energy systems with and without energy storage optimization: a survey of advanced machine learning and deep learning techniques. *Energies*, 15(2), 578.
- Afrin, K., Nepal, B., & Monplaisir, L. (2018). A data-driven framework to new product demand prediction: Integrating product differentiation and transfer learning approach. *Expert Systems with Applications*, 108, 246–257.
- Ahmed, S., Chakraborty, R. K., Essam, D. L., & Ding, W. (2022). Poly-linear regression with augmented long short term memory neural network: Predicting time series data. *Information Sciences*.
- Alfano, P. D., Pastore, V. P., Rosasco, L., & Odone, F. (2024). Top-tuning: A study on transfer learning for an efficient alternative to fine tuning for image classification with fast kernel methods. *Image and Vision Computing*, 142, Article 104894.
- Alsirhani, A., Siddiqi, M. H., Mostafa, A. M., Ezz, M., & Mahmoud, A. A. (2023). A novel classification model of date fruit dataset using deep transfer learning. *Electronics*, 12(3), 665.
- Amin-Naseri, M. R., & Tabar, B. R. (2008). Neural network approach to lumpy demand forecasting for spare parts in process industries. In *2008 international conference on computer and communication engineering* (pp. 1378–1382). IEEE.
- Ashtari, B., Alizadeh Bidgoli, M., Babaei, M., & Ahmarinejad, A. (2022). A two-stage energy management framework for optimal scheduling of multi-microgrids with generation and demand forecasting. *Neural Computing and Applications*, 1–15.
- Chen, L., Jiang, M., Li, T., Jia, F., & Lim, M. K. (2023). Supply chain learning and performance: a meta-analysis. *International Journal of Operations & Production Management, ahead-of-print*(ahead-of-print).
- Chen, A., Wang, Z., Vidaurre, K. L. L., Han, Y., Ye, S., Tao, K., Wang, S., Gao, J., & Li, J. (2024). Knowledge-reused transfer learning for molecular and materials science. *Journal of Energy Chemistry*, 98, 149–168.
- Corberán-Vallet, A., Bermúdez, J. D., & Vercher, E. (2011). Forecasting correlated time series with exponential smoothing models. *International Journal of Forecasting*, 27(2), 252–265.
- Criado-Ramón, D., Ruiz, L., & Pegalajar, M. (2022). Electric demand forecasting with neural networks and symbolic time series representations. *Applied Soft Computing*, 122, Article 108871.
- Da Silva, V. L., Kovaleski, J. L., & Pagani, R. N. (2019). Technology transfer in the supply chain oriented to industry 4.0: a literature review. *Technology Analysis & Strategic Management*, 31(5), 546–562.
- De Carvalho, I. V., & Viana Cunha, N. C. (2013). Proposta de um modelo de transferência de tecnologia para as universidades públicas brasileiras. <https://hdl.handle.net/20.500.13048/731>. (Accessed 03 January 2025).
- Dolgui, A., Ivanov, D., & Rozhkov, M. (2020). Does the ripple effect influence the bullwhip effect? An integrated analysis of structural and operational dynamics in the supply chain. *International Journal of Production Research*, 58(5), 1285–1301.
- Ehrig, C., Cleophas, C., & Forestier, G. (2024). The impact of data set similarity and diversity on transfer learning success in time series forecasting. *arXiv preprint arXiv:2404.06198*.
- Faisal, A., Kamruzzaman, M., Yigitcanlar, T., & Currie, G. (2019). Understanding autonomous vehicles. *Journal of Transport and Land Use*, 12(1), 45–72.
- Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4), 681–694.
- Garinian, E. O. M., Fierro, T. E. S., Saucedo, J. A. M., & Aguilar, R. R. (2023). Machine learning applications for demand driven in supply chain: Literature review. In *Smart applications with advanced machine learning and human-centred problem design* (pp. 763–772). Springer.
- Gautam, Y. (2022). Transfer Learning for COVID-19 cases and deaths forecast using LSTM network. *ISA Transactions*, 124, 41–56.
- Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. "O'Reilly Media, Inc."
- Gonçalves, J. N., Carvalho, M. S., & Cortez, P. (2020). Operations research models and methods for safety stock determination: A review. *Operations Research Perspectives*, Article 100164.
- Gu, Q., & Dai, Q. (2021). A novel active multi-source transfer learning algorithm for time series forecasting. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 51(3), 1326–1350.
- Gu, Q., Dai, Q., Yu, H., & Ye, R. (2021). Integrating multi-source transfer learning, active learning and metric learning paradigms for time series prediction. *Applied Soft Computing*, 109, Article 107583.
- He, Q.-Q., Pang, P. C.-I., & Si, Y.-W. (2019). Transfer learning for financial time series forecasting. In *Pacific rim international conference on artificial intelligence* (pp. 24–36). Springer.
- Huang, L., Fan, J., Zhao, W., & You, Y. (2023). A new multi-source transfer learning method based on two-stage weighted fusion. *Knowledge-Based Systems*, 262, Article 110233.
- Huang, X., Li, Q., Tai, Y., Chen, Z., Liu, J., Shi, J., & Liu, W. (2022). Time series forecasting for hourly photovoltaic power using conditional generative adversarial network and Bi-LSTM. *Energy*, 246, Article 123403.
- Ivanov, D., Blackhurst, J., & Das, A. (2021). Supply chain resilience and its interplay with digital technologies: Making innovations work in emergency situations. *International Journal of Physical Distribution & Logistics Management*.
- Ivanov, D., Sethi, S., Dolgui, A., & Sokolov, B. (2018). A survey on control theory applications to operational systems, supply chain management, and Industry 4.0. *Annual Reviews in Control*, 46, 134–147.
- Ivanov, D., Tsipoulanis, A., & Schönberger, J. (2019). *Global supply chain and operations management: A decision-oriented introduction to the creation of value: vol. 2*, Springer.
- Jangam, E., Barreto, A. A. D., & Annavarapu, C. S. R. (2022). Automatic detection of COVID-19 from chest CT scan and chest X-Rays images using deep learning, transfer learning and stacking. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 52(2), 2243–2259.
- Janković Babić, R. (2024). A comparison of methods for image classification of cultural heritage using transfer learning for feature extraction. *Neural Computing and Applications*, 36(20), 11699–11709.
- Jin, X., Yu, X., Wang, X., Bai, Y., Su, T., & Kong, J. (2020). Prediction for time series with CNN and LSTM. In *Proceedings of the 11th international conference on modelling, identification and control* (pp. 631–641). Springer.
- Joseph, R. V., Mohanty, A., Tyagi, S., Mishra, S., Satapathy, S. K., & Mohanty, S. N. (2022). A hybrid deep learning framework with CNN and Bi-directional LSTM for store item demand forecasting. *Computers & Electrical Engineering*, 103, Article 108358.
- Ju, J., Zheng, H., Xu, X., Guo, Z., Zheng, Z., & Lin, M. (2022). Classification of jujube defects in small data sets based on transfer learning. *Neural Computing and Applications*, 34(5), 3385–3398.
- Karb, T., Kühl, N., Hirt, R., & Glivici-Cotruta, V. (2020). A network-based transfer learning approach to improve sales forecasting of new products. *arXiv preprint arXiv:2005.06978*.
- Kaul, D., Raju, H., & Tripathy, B. (2022). Deep learning in healthcare. In *Deep learning in data analytics* (pp. 97–115). Springer.
- Kharfan, M., Chan, V. W. K., & Firdolas Efendil, T. (2021). A data-driven forecasting approach for newly launched seasonal products by leveraging machine-learning approaches. *Annals of Operations Research*, 303(1), 159–174.
- Kiefer, D., Grimm, F., Bauer, M., & Van Dintner, C. (2021). Demand forecasting intermittent and lumpy time series: Comparing statistical, machine learning and deep learning methods.
- Kiefer, D., Grimm, F., Van, D., et al. (2022). Artificial intelligence in supply chain management: Investigation of transfer learning to improve demand forecasting of intermittent time series with deep learning. In *Proceedings of the 55th hawaii international conference on system sciences* (pp. 1656–1665).
- Kilimci, Z. H., Akyuz, A. O., Uysal, M., Akyokus, S., Uysal, M. O., Atak Bulbul, B., & Ekmis, M. A. (2019). An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain. *Complexity*, 2019.
- Kourentzes, N. (2013). Intermittent demand forecasts with neural networks. *International Journal of Production Economics*, 143(1), 198–206.
- Kuźlewska, U. (2022). Clustering algorithms for efficient neighbourhood identification in session-based recommender systems. In *International conference on dependability and complex systems* (pp. 143–152). Springer.
- Lee, M., Lee, J.-H., & Kim, D.-H. (2022). Gender recognition using optimal gait feature based on recursive feature elimination in normal walking. *Expert Systems with Applications*, 189, Article 116040.
- Lee, Y.-S., & Tong, L.-I. (2011). Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. *Knowledge-Based Systems*, 24(1), 66–72.
- Li, W., Huang, R., Li, J., Liao, Y., Chen, Z., He, G., Yan, R., & Gryllias, K. (2022). A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges. *Mechanical Systems and Signal Processing*, 167, Article 108487.
- Li, H., Li, X., Sun, Y., Dong, H., & Xu, G. (2024). First-arrival picking for out-of-distribution noisy data: A cost-effective transfer learning method with tens of samples. *IEEE Transactions on Geoscience and Remote Sensing*.
- Li, S., & Zheng, P. (2020). Transfer learning for smart manufacturing: A stepwise survey. *IFAC-PapersOnLine*, 53(5), 37–42.
- Liu, Q., Long, L., Yang, Q., Peng, H., Wang, J., & Luo, X. (2022). LSTM-SNP: A long short-term memory model inspired from spiking neural P systems. *Knowledge-Based Systems*, 235, Article 107656.
- Liu, S., Lu, Y., Zheng, P., Shen, H., & Bao, J. (2022). Adaptive reconstruction of digital twins for machining systems: A transfer learning approach. *Robotics and Computer-Integrated Manufacturing*, 78, Article 102390.
- Liu, C., Zhu, H., Tang, D., Nie, Q., Li, S., Zhang, Y., & Liu, X. (2023). A transfer learning CNN-LSTM network-based production progress prediction approach in IIoT-enabled manufacturing. *International Journal of Production Research*, 61(12), 4045–4068.

- Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN-LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 32(23), 17351–17360.
- Ma, Z., Wang, C., & Zhang, Z. (2021). Deep learning algorithms for automotive spare parts demand forecasting. In *2021 international conference on computer information science and artificial intelligence* (pp. 358–361). IEEE.
- Mahmood, T., Li, J., Pei, Y., Akhtar, F., Butt, S. A., Ditta, A., & Qureshi, S. (2022). An intelligent fault detection approach based on reinforcement learning system in wireless sensor network. *Journal of Supercomputing*, 78(3), 3646–3675.
- Mancuso, P., Piccialli, V., & Sudoso, A. M. (2021). A machine learning approach for forecasting hierarchical time series. *Expert Systems with Applications*, 182, Article 115102.
- Maschler, B. (2023). A survey on deep industrial transfer learning in fault prognostics. arXiv preprint arXiv:2301.01705.
- Mehtab, S., & Sen, J. (2022). Analysis and forecasting of financial time series using CNN and LSTM-based deep learning models. In *Advances in distributed computing and machine learning* (pp. 405–423). Springer.
- Muhammad, K., Khan, S., Del Ser, J., & De Albuquerque, V. H. C. (2020). Deep learning for multigrade brain tumor classification in smart healthcare systems: A prospective survey. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2), 507–522.
- Nguyen, K., & Ho, N. (2024). Energy-based sliced wasserstein distance. *Advances in Neural Information Processing Systems*, 36.
- Nicholson, M., Aghahari, R., Conran, C., Assem, H., & Kelleher, J. D. (2022). The interaction of normalisation and clustering in sub-domain definition for multi-source transfer learning based time series anomaly detection. *Knowledge-Based Systems*, 257, Article 109894.
- Nikolopoulos, K. (2021). We need to talk about intermittent demand forecasting. *European Journal of Operational Research*, 291(2), 549–559.
- Niu, S., Liu, Y., Wang, J., & Song, H. (2020). A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence*, 1(2), 151–166.
- Oliveira, P., Rodrigues, F., & Henriques, P. R. (2005). A formal definition of data quality problems. In *ICIQ*.
- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359.
- Pavlyshenko, B. M. (2019). Machine-learning models for sales time series forecasting. *Data*, 4(1), 15.
- Pavlyshenko, B. M. (2022). Analytics of business time series using machine learning and Bayesian inference. arXiv preprint arXiv:2205.12905.
- Pereira, G. W., Valente, D. S. M., de Queiroz, D. M., Santos, N. T., & Fernandes-Filho, E. I. (2022). Soil mapping for precision agriculture using support vector machines combined with inverse distance weighting. *Precision Agriculture*, 1–16.
- Rai, R., Tiwari, M. K., Ivanov, D., & Dolgui, A. (2021). Machine learning in manufacturing and industry 4.0 applications. *International Journal of Production Research*, 59(16), 4773–4778.
- Ramezankhani, M., Crawford, B., Narayan, A., Voggenreiter, H., Seethaler, R., & Milani, A. S. (2021). Making costly manufacturing smart with transfer learning under limited data: A case study on composites autoclave processing. *Journal of Manufacturing Systems*, 59, 345–354.
- Ranganayaki, V., Deepa, S., & Maheswari, C. (2020). Improved wind speed prediction using various neural network models. *International Journal of Electrical Engineering and Technology*, 11(2).
- Sagheer, A., Hamdoun, H., & Youness, H. (2021). Deep LSTM-based transfer learning approach for coherent forecasts in hierarchical time series. *Sensors*, 21(13), 4379.
- Saleem, S., Amin, J., Sharif, M., Anjum, M. A., Iqbal, M., & Wang, S.-H. (2022). A deep network designed for segmentation and classification of leukemia using fusion of the transfer learning models. *Complex & Intelligent Systems*, 8(4), 3105–3120.
- Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V. K., & Soman, K. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 international conference on advances in computing, communications and informatics* (pp. 1643–1647). IEEE.
- Syntetos, A. A., Boylan, J. E., & Croston, J. (2005). On the categorization of demand patterns. *Journal of the Operational Research Society*, 56(5), 495–503.
- Taieb, S. B., & Hyndman, R. J. (2014). A gradient boosting approach to the Kaggle load forecasting competition. *International Journal of Forecasting*, 30(2), 382–394.
- Tang, Y., Dehaghani, M. R., & Wang, G. G. (2023). Review of transfer learning in modeling additive manufacturing processes. *Additive Manufacturing*, 61, Article 103357.
- Tao, T., Sun, C., Wu, Z., Yang, J., & Wang, J. (2022). Deep neural network-based prediction and early warning of student grades and recommendations for similar learning approaches. *Applied Sciences*, 12(15), 7733.
- Veeramsetty, V., Reddy, K. R., Santhosh, M., Mohnot, A., & Singal, G. (2022). Short-term electric power load forecasting using random forest and gated recurrent unit. *Electrical Engineering*, 104(1), 307–329.
- Wang, J., & Chen, Y. (2023). *Introduction to transfer learning: algorithms and practice*. Springer Nature.
- Wang, G., Lin, L., Chen, R., Wang, G., & Zhang, J. (2021). Joint learning of neural transfer and architecture adaptation for image recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10), 5401–5415.
- Wang, Z., Liu, Q., Chen, H., & Chu, X. (2021). A deformable CNN-DLSTM based transfer learning method for fault diagnosis of rolling bearing under multiple working conditions. *International Journal of Production Research*, 59(16), 4811–4825.
- Wang, Y., Liu, R., Lin, D., Chen, D., Li, P., Hu, Q., & Chen, C. P. (2023). Coarse-to-fine: Progressive knowledge transfer-based multitask convolutional neural network for intelligent large-scale fault diagnosis. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2), 761–774. <http://dx.doi.org/10.1109/TNNLS.2021.3100928>.
- Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144–156.
- Wang, P., Wang, T., Yang, S., Cheng, H., Huang, P., & Zhang, Q. (2024). Production quality prediction of cross-specification products using dynamic deep transfer learning network. *Journal of Intelligent Manufacturing*, 35(6), 2567–2592.
- Weber, M., Auch, M., Doblander, C., Mandl, P., & Jacobsen, H.-A. (2021). Transfer learning with time series data: a systematic mapping study. *Ieee Access*, 9, 165409–165432.
- Wu, D., Wang, X., & Wu, S. (2022). Jointly modeling transfer learning of industrial chain information and deep learning for stock prediction. *Expert Systems with Applications*, 191, Article 116257.
- Xiong, P., Zhu, Y., Sun, Z., Cao, Z., Wang, M., Zheng, Y., Hou, J., Huang, T., & Que, Z. (2018). Application of transfer learning in continuous time series for anomaly detection in commercial aircraft flight data. In *2018 IEEE international conference on smart cloud* (pp. 13–18). IEEE.
- Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), 1–19.
- Ye, R., & Dai, Q. (2021). Implementing transfer learning across different datasets for time series forecasting. *Pattern Recognition*, 109, Article 107617.
- Zhang, X., Song, C., Zhao, J., Xu, Z., & Deng, X. (2022). Deep subdomain learning adaptation network: A sensor fault-tolerant soft sensor for industrial processes. *IEEE Transactions on Neural Networks and Learning Systems*.
- Zhang, X., Zhou, J., Sun, W., & Jha, S. K. (2022). A lightweight CNN based on transfer learning for COVID-19 diagnosis. *Computers, Materials and Continua*, 1123–1137.
- Zhao, K., Jia, F., & Shao, H. (2023). A novel conditional weighting transfer Wasserstein auto-encoder for rolling bearing fault diagnosis with multi-source domains. *Knowledge-Based Systems*, 262, Article 110203.