Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

FORECASTING WITH VISIBILITY USING PRIVACY PRESERVING FEDERATED LEARNING

Bo Zhang
Wen Jun Tan
Wentong Cai

School of Computer Science and Engineering
Nanyang Technological University
Singapore

Allan N. Zhang

Singapore Institute of Manufacturing Technology
Agency for Science, Technology and Research
Singapore

ABSTRACT

In the fluctuating and unstable supply chain environment, accurate demand forecasting is especially important. To improve the prediction accuracy, one possible way is to improve supply chain visibility by sharing information and knowledge among the supply chain entities. However, there is a potential risk that the raw data may be leaked to the competitors, affecting the business opportunities. To avoid information leakage, a secure demand forecasting with supply chain visibility is necessary. This paper proposes a Federated Learning based approach to predict demand for supplier with supply chain visibility while protecting the data privacy of other entities within the supply chain. To evaluate performance of forecasting accuracy, we designed a supply chain simulation model to generate data. From the experimental results, our proposed method outperforms the other demand forecasting methods without visibility and achieves a similar performance to the method with full visibility.

1 INTRODUCTION

Forecasting demand is essential in firms' operations to reduce management costs and provide considerate plans (Nenni et al. 2013). Accurate forecasting results can lead to low inventory cost, high service level stability, and reduction of bullwhip effects, which can help firms to increase profit and remain competitive in the current market (Barlas and Gunduz 2011; Wang and Disney 2016; Feizabadi 2022).

There are many existing works focusing on demand forecasting problems. For instance, some researchers use artificial neural network (ANN) to find patterns of future demand (Kochak and Sharma 2015; Feizabadi 2022). Other researchers used autoregressive integrated moving average (ARIMA) to study hidden patterns of demand trends (Babai et al. 2013; Feizabadi 2022). Even though Zhang and Zhang (2007) demonstrated that demand information sharing in supply chain helps to reduce total cost, the above mentioned approaches do not have full visibility in the whole supply chain. They are only able to perform demand forecasting using single firm data, e.g., forecasting only using the demand for supplier, and do not consider information from the downstream firms in the supply chain. Due to the variance of downstream firms' demand and customer order, demand forecasting using single firm data cannot efficiently predict future demand.

Nowadays, due to the recent COVID-19 pandemic, supply chain uncertainties are also increased and have a significant impact on firms (Nikolopoulos et al. 2021). The uncertainties increase the difficulty of demand forecasting and make supply chains frailer. To overcome the negative effects and improve forecasting performance, one possible solution is to improve supply chain visibility (Barlas and Gunduz 2011; Somapa et al. 2018; Hamadneh et al. 2021). Supply chain visibility means that firms share their inventory and demand information with other firms residing in the same supply chain in a timely and accurately way (Barratt and Oke 2007; Barratt and Barratt 2011).

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

Undoubtedly, sharing detailed information is essential in accurate demand forecasting. However, supply chain visibility brings a new problem that the shared information may be leaked (Chen and Özer 2019). Information leakage may put firms in a dangerous situation, especially some firms are highly dependent on private information to achieve their competitive edge (Tan et al. 2016). The privacy preserving concerns of firms become a major obstacle in gaining supply chain visibility.

The arising of Federated Learning (FL) provides a possible solution to improve supply chain visibility in privacy preserving manner. FL is a distributed machine learning approach that trains decentralized data across multiple servers without direct data exchange or leak (Yang et al. 2019). As such, FL can be applied to forecast demand – aggregate machine learning model across all firms can be trained without leaking sensitive data information (Vepakomma et al. 2018; Yang et al. 2019).

In this paper, we propose an FL model to forecast demand for supplier, combining the advantages of both increasing supply chain visibility and providing data privacy. The FL model can be considered as an aggregator model that contains partial models of all firms in the supply chain. During training, the partial model is updated by each firm separately. Then on a third party trust server, an aggregated model is used to combine output from all firms' partial models to forecast the demand for supplier. During this process, the only data needs to be exchanged are the intermediate model results. As a case study, we constructed a discrete event simulation model of a three-stage supply chain to train and evaluate our FL model. Compared to the existing forecasting methods, the proposed FL model can achieve better performance than the methods without supply chain visibility, and its performance is close to the method with full visibility.

The following are the contributions of our work:

- We propose a method to improve supply chain visibility for supplier forecasting while preserving data privacy.
- To the best of our knowledge, this is the first study using FL across multistage firms for supply chain demand forecasting.
- Using simulated data, the proposed method is shown to achieve better forecasting performance compared to those methods without supply chain visibility and comparable performance compared to the method with full visibility.

The remaining sections are organized as follows: Section 2 presents the background of supply chain visibility, demand forecasting approaches, and FL. Section 3 introduces a three-stage supply chain simulation model and the proposed FL model. Section 4 compares the performance of our method to the existing methods using the simulation results. Section 5 concludes our paper and discusses future work and limitations.

2 RELATED WORK

2.1 Demand Forecasting

Demand forecasting in supply chain has been well studied since accurate prediction can help firms remain competitive by reducing operation cost and bullwhip effects, and growing profit (Seyedan and Mafakheri 2020). A variety of mathematics analysis approaches have been used for demand forecasting, including regression analysis, time series modeling, and neural networks (NN) (Wang et al. 2016). Regression analysis is widely used to describe the relationships between input features and target prediction value by given continuous functions. Compared to deep learning approaches, regression models are easy to train and have a better explanatory power. The weights of each input feature in the regression model demonstrate the influence of prediction variables. Merkuryeva et al. (2019) applied multiple linear regression and symbolic regression in pharmaceutical supply chain to predict demand. Based on the historical weekly sales data, symbolic regression achieves the lowest accuracy error among existing works (Merkuryeva et al. 2019). In addition, to improve prediction accuracy, Wang (2012) used genetic algorithm to optimize support vector regression in demand forecasting.

Time series analysis is used to discover unknown patterns over time sequence data (Ma et al. 2014). For demand forecasting, the demand and other features are converted in equal size intervals, referred to as time sequence data. The time sequence data are the input of the time series model to forecast future demand. Babai et al. (2013) proposed an ARIMA model for demand forecasting of a huge superstore. Feizabadi (2022) combines NN and ARIMA model in demand forecasting of a steel manufacturer problem.

Using NN models in demand forecasting is a way to find hidden patterns of input data. The NN is a set of neurons organized by certain architecture. It maps the input features and target outputs by finding the inherent correlations. Machine Learning (ML) can discover non-linear relations and fit a model with historical data (Chase Jr et al. 2016). With the advancement of ML techniques and availability of big data, NN approaches has became the most popular research methods in demand forecasting (Merkuryeva et al. 2019). Abbasimehr et al. (2020) used a long short term memory (LSTM) neural model in demand forecasting and demonstrated that the LSTM model achieves the highest prediction accuracy among other approaches. Feizabadi (2022) implemented a two layers perceptron NN and achieved high performance in demand forecasting scenarios.

2.2 Supply Chain Visibility

Supply chain visibility is defined as the ability of the firms within the supply chain to access accurate and timely information which helps firms to make decisions (Barlas and Gunduz 2011). For example, firms shared the demand information across the supply chain to increase the demand forecasting accuracy (Somapa et al. 2018). With the supply chain visibility, all firms gain benefits, increasing revenue from lower inventory cost, lower supply chain risk, and rapidly adjusted demand plans (Somapa et al. 2018). Furthermore, the accurate demand forecasting helps to mitigate the bullwhip effects in supply chain (Barlas and Gunduz 2011; Feizabadi 2022).

The supply chain visibility requires different firms to share information among each other. Even though firms may have trusted relationship with each other, sensitive information can be leaked to other unauthorized parties (Tan et al. 2016). In recent years, information leakages are growing and become major concerns and challenges for supply chain partners (Ried et al. 2021). Information leakage may lead to less efficient and lower potential profits for all entities in the supply chain (Kong, Rajagopalan, and Zhang 2013). Ensuring data privacy and supply chain visibility at the same time becomes an important issue in demand forecasting.

In order to preserve privacy, there are mainly three approaches – trusted third party, secure transformation, and secure multiparty computation (SMC) (Hong et al. 2014). The trusted third party approach requires all firms upload their local private information to the third party (Özener and Ergun 2008). The secure transformation approach suggests to randomized data and share the processed data to other entities (Hong et al. 2018). The SMC approach is to securely compute optimization function in a distributed computation (Hong et al. 2014). SMC provides a secure joint computation function over multi-party to keep the inputs private (Hong et al. 2014). Similarly, FL can be used to provide an efficient secure computation in different distributed entities. Compared to the trusted third party approach, FL does not require to transmit any raw data to third party. Atallah et al. (2004) applied SMC in ARIMA and linear regression to forecast customer demand in a single supplier multiple retailer supply-chain model. They built an aggregate SMC model over all retailers to protect retailers privacy, while our approach constructs an aggregate model across multiple stages in the supply chain.

2.3 Federated Learning

Typical ML merges all local data sets as a whole and trains a centralized model. However, this becomes an issue when sensitive data need to be shared and may be leaked in the process. To resolve this issue, FL has been proposed to train NN models locally without sharing raw data (Konečnỳ et al. 2016). FL has been widely used in privacy preserving cases (Yang et al. 2019). For example, Google built a mobile keyboard suggestion FL model by using client text data (Hard et al.), and different medical organizations can collaborate together using the FL model without sharing confidential patient data (Sheller et al. 2020).

There are many ways to achieve data privacy in FL, such as using homomorphic encryption to encrypt gradient (Yang et al. 2019), exchanging intermediate model results (also called split neural network) (Vepakomma et al. 2018), and using differential privacy (Abadi et al. 2016). Homomorphic encryption enables the encrypted data to perform mathematical operations. Due to this property, FL models can share and compute the parameter gradients in the encrypted manner. This encryption approach ensures raw data will not be leaked but it is time-consuming (Yang et al. 2019). The split neural network (SNN) divides the FL model and computes the partial model locally in different entities and only the intermediate computation results will be shared (Vepakomma et al. 2018). Differential privacy is to add noises in the gradients and share the noised gradient with other entities (Abadi et al. 2016). In this paper, we proposed a FL approach to forecast demand for supplier using SNN.

3 METHOD

In this paper, we study a three-stage supply chain, in particular the demand forecasting for the supplier. To generate the required data, we first develop a discrete event simulation model of the three-stage supply chain. Second, we propose an FL model for demand forecasting at the supplier based on this three-stage supply chain. The FL model will be trained and evaluated using the simulation model.

3.1 Three-Stage Supply Chain Model

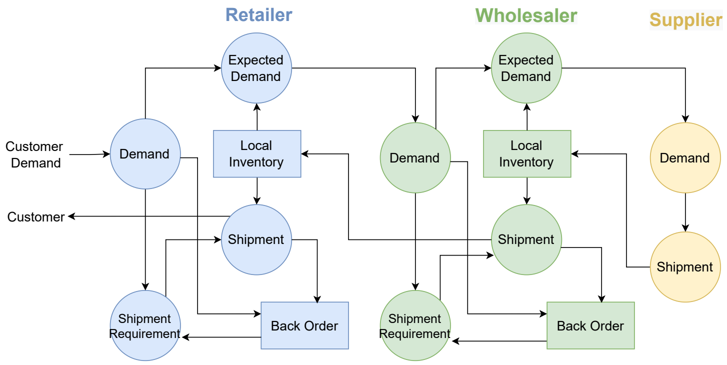


Figure 1: A three-stage supply chain model.

We design a three-stage supply chain model based on Barlas and Gunduz (2011). The main purpose of this simulation model is to generate downstream data to improve the demand forecasting at the supplier. The supply chain consists of a retailer, a wholesaler, and a supplier entity (see Figure 1). Each entity i in the supply chain only orders from its upper stream entity $i+1$ and fulfils the orders from its downstream entity $i-1$ immediately upon receiving the order from downstream if it has sufficient inventory. If there is insufficient inventory, it fulfils the orders partially and the back order will be satisfied after the inventory has been replenished. We assume that all goods require a constant shipment lead time (LT) to be received. We also assume that the supplier has an unlimited supply, hence there is no back order and inventory in supplier.

The customer demand, i.e., demand for retailer, (D_t^R) at time t is generated by random sample from a normal distribution (Kim et al. 2010), represented by:

$$D_t^R = \mathcal{N}\left(\frac{1}{t-1}\sum_{\tau=1}^{t-1} D_\tau^R + \mathcal{U}(-c,c),\ d\right) \quad [Products/Period]$$

where $\mathcal{N}$ is normal distribution with standard deviation d and $\mathcal{U}$ is uniform distribution with the range from $-c$ to c. The fluctuation in customer demand can be increased by increasing d and c.

The local inventory (LI) of entity i is increased by the products (S^{i+1}) shipped from the upstream entity $i+1$ and decreased by the products (S^{i-1}) shipped to the downstream entity $i-1$:

$$LI_t^i = LI_{t-1}^i + S_{t-LT^{i+1}}^{i+1} - S_t^{i-1} \quad [Products]$$

where LT^{i+1} is the shipment lead time from the upstream entity.

The shipment requirement (SR) for entity i at time t is the sum of the back order (BO) and demand received (D):

$$SR_t^i = BO_t^i + D_t^i \quad [Products/Period]$$

If there is sufficient inventory (LI), the required shipment (S) is delivered immediately in one period. If not, the unfulfilled part of the orders is added to the back order (BO).

$$S_t^i = min(SR_t^i, LI_t^i) \quad [Products/Period]$$

$$BO_{t+1}^i = BO_t^i + D_t^i - S_t^i \quad [Products]$$

Since the supplier (e.g., entity s) has unlimited supply, the products will be shipped according to the received demand, i.e., $S_t^s = D_t^s$.

Inventory is replenished based on forecast future demand. A simple time window averaging is used to forecast the demand. The expected demand ($\hat{D}_t^i$) at time t is the average of previous $\mathscr{T}$ time demand multiply the shipment lead time LT of entity i:

$$\hat{D}_t^i = LT^i * \frac{1}{\mathscr{T}} \sum_{\tau=1}^{\mathscr{T}} D_{t-\tau}^i \quad [Products/Period]$$

The orders, i.e., demand for the upstream entity $i+1$, is calculated based on the difference between the current inventory (LI) and expected demand ($\hat{D}$):

$$D_{t+1}^{i+1} = |LI_t^i - \hat{D}_t^i| \quad [Products/Period]$$

3.2 Split Neural Network Federated Learning Model

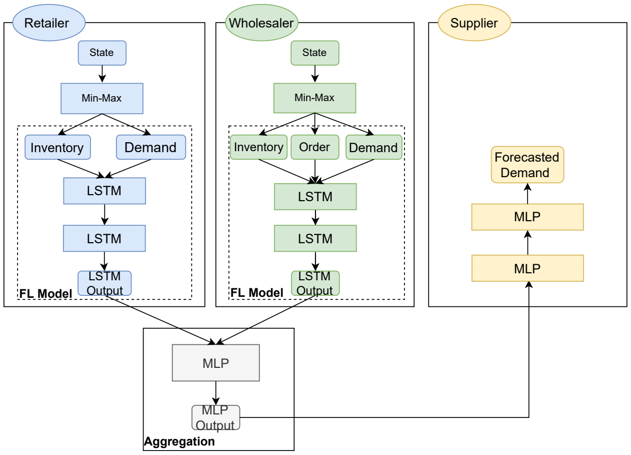


Figure 2: FL model based on the three-stage supply chain model.

We proposed a FL-based approach to forecast demand for the supplier using SNN. All entities in the supply chain need to cooperate together to build an aggregate FL model. The whole FL model composes of four partial models – retailer, wholesaler, supplier, and aggregation model (see Figure 2). The NN needs to be split into partial models as the retailer and wholesaler are not sharing their private data with the supplier. The output of *retailer* and *wholesaler* models represent the hidden states of the retailer and wholesaler.

The *aggregation* model conceals the information from the *retailer* and *wholesaler* models by not sharing the information directly with the *supplier*. Only the extracted information from the hidden states are shared with the *supplier* model. Finally, the *supplier* model is used to map the extracted information to forecast the demand.

As the demand for supplier is based on the order decision from the downstream firms, information from the retailer and wholesaler are sufficient for model construction. The *retailer* model uses the time sequence normalized state information of the demand for retailer and retailer inventory as the input features, extracts the hidden patterns from data, and generates the intermediate model output. Correspondingly, the *wholesaler* model uses the demand for wholesaler, wholesaler inventory, and wholesaler order as the inputs to generate intermediate output. The intermediate model outputs from the *retailer* and *wholesaler* models are combined at the *aggregation* model in a trusted third party server. The output from the trusted third party server is then used as the input to the supplier model to generated the predicted demand.

The exchange of the intermediate output from the *retailer* and *wholesaler* models to the *supplier* model can increase the supply chain visibility by enabling the *supplier* model to generate a more accurate demand prediction. The data privacy is preserved by using a third-party trust server that only exchanges the intermediate data instead of raw data between the entities (Vepakomma et al. 2018). All the partial models are required to execute together in both training and evaluation steps. Since only intermediate model outputs and gradients are involved in the exchange process, the original raw data can be well protected without leakage.

The LSTM models in *retailer* and *wholesaler* contain two LSTM layers. Abbasimehr et al. (2020) suggested that two LSTM layers structure achieves high demand forecasting accuracy comparing with other NN structures. The LSTM layers can be considered as the encoder that extracts and classifies the hidden information from input data and convert them to high dimensional data. The *aggregation* model and *supplier* model contain fully connected (FC) layers. The FC layers can be considered as decoder that converts the high dimensional data into demand prediction. To fit the structure mentioned in (Vepakomma et al. 2018) and protect data privacy, our FL model has three FC layers – the *aggregation* model has two FC layers and *supplier* contains one FC layer. Each partial model can be represented as a function – retailer (f_r), wholesaler (f_w), supplier (f_s), and aggregation (f_a). The FL models take the states of *retailer* (x_r) and *wholesaler* (x_w) as input, which can be expressed by:

$$y_r = f_r(x_r),\ y_w = f_w(x_w)$$

Then, the intermediate output y_r and y_w are concatenated and sent to the third party server and compute:

$$y_a = f_a(concat(y_r, y_w))$$

Lastly, the supplier takes aggregated information y_a as input to forecast the demand ($\hat{y}$):

$$\hat{y} = f_s(y_a)$$

Through simulation, detailed time series information regarding the states of the *retailer* and *wholesaler* can be generated. The FL models are trained with data generated from the simulation model. Then additional generated data is used to evaluate the FL model.

For the retailer and wholesaler models, the input features $x(t_i)$ are first normalized by min-max normalization:

$$\hat{x}(t_i) = \frac{x(t_i) - \min(x)}{\max(x) - \min(x)},$$

where $\hat{x}(t_i)$ is the normalized results of the t_i element, $\min(x)$ is the minimum value of x, and $\max(x)$ is the maximum value of x.

Secondly, they take the normalized states as input into two LSTM layers. For LSTM layers implementation, we apply an open source Python library Pytorch (Paszke et al. 2019). The LSTM layer in Pytorch exactly follows (Hochreiter and Schmidhuber 1997) and can be formulated as follows:

$$\begin{aligned}
o_t^l &= \sigma(W_o^l X_t^l + U_o^l h_{t-1}^l + b_o^l),\\
i_t^l &= \sigma(W_i^l X_t^l + U_i^l h_{t-1}^l + b_i^l),\\
f_t^l &= \sigma(W_f^l X_t^l + U_f^l h_{t-1}^l + b_f^l),\\
c_t^l &= (f_t^l \odot c_{t-1}^l) + (i_t^l \odot tanh(W_c^l X_t^l + U_c^l h_{t-1}^l + b_c^l)),\\
h_t^l &= o_t^l \odot tanh(c_t^l),
\end{aligned}$$

where $l \in 1,2$ represents the first or the second LSTM layer; X^l is the input of the LSTM layer l where the first layer takes $\hat{x}(t_i)$ as input and the second layers takes h_t^1 as input; h_t^l and c_t^l are the hidden state and cell state of layer l at time t; i_t, f_t, and o_t are the input gates, the forget gates, and the output gates respectively; W, U, and b are the weights and bias in above mentioned gates respectively; σ is the sigmoid function; and $\odot$ is the Hadamard product.

Thirdly, the outputs of *retailer* and *wholesaler* models are sent to the third party trust server as input. The trusted server concatenates the intermediate results and puts in FC layers:

$$\begin{aligned}
FC_t^1 &= W_{FC}^1 \cdot concat(y_r, y_w) + b_{FC}^1,\\
FC_t^2 &= W_{FC}^2 \cdot FC_t^1 + b_{FC}^2
\end{aligned}$$

where *concat* is the concatenation operation, W_{FC}^i and b_{FC}^i are the weights and bias in the i-th FC layers, FC_t^i are the results of the i-th FC layer outputs in the server at time t.

After the FC_t^2 output has been calculated, it will be sent to *supplier* model which computes the final prediction as follows:

$$\hat{y}_t = W \cdot FC_t^2 + b,$$

where $\hat{y}_t$ is the prediction of future demand at time t, W and b are the weights and bias in *supplier* model.

During the training, dropout layers have been implemented on LSTM layers and FC layers, which randomly zeros the input tensor elements. Dropout layers help to reduce the dependence over the neurons and prevent model overfitting. Moreover, by applying the mean squared error (MSE) between the prediction output $\hat{y}$ and actual value, the optimizer in *supplier* model computes the corresponding gradients and sends them back to the trusted server and retailer and wholesaler models, using back propagation in NN. All the models will update their own gradient via optimizers respectively.

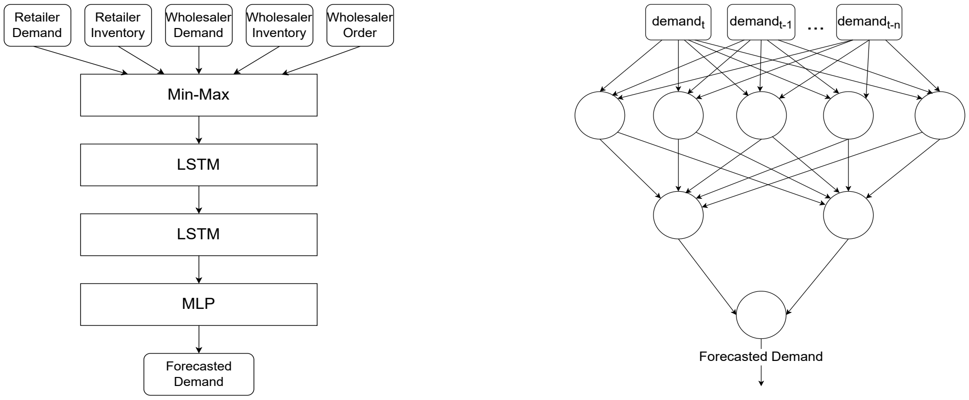
4 EXPERIMENT & EVALUATION

4.1 Simulation Setup

The experiment dataset is generated by the three-stage supply chain model described in Section 3. In our supply chain simulation, each time step represents a day, and we simulate for a total of 730 days. The warm-up period for the simulation is the first 30 days. The shipment lead time (LT) is assumed to be 2, based on the settings from Barlas and Gunduz (2011). The initial states of all entities, inventory level, raw material level and order quantity in entities, are randomly generated. The parameters used to generate customer demand, c and d, are different in each experiment. In our three experiments, we use three different parameters to generate data: $c = 2$ $d = 30$, $c = 5$ $d = 50$, and $c = 10$ $d = 100$ for three experiments respectively. For each experiment, it repeatedly runs 10 times using the same parameters with different initial states.

4.2 Demand Forecasting Method Settings

We compared our proposed FL model with other demand forecasting methods under different supply chain visibility to predict the future $\mathscr{T}$ days of demand for supplier. In the full visibility scenario, we assume all firms are willing to share all of their data and centralized into a dataset. We construct an LSTM model with full visibility (FV-LSTM) between all entities in the supply chain to forecast future demand for supplier (see Figure 3a). In the partial visibility scenario, we assume all firms are willing to cooperate with each other but not agree to share the raw data. Our proposed FL-model with partial visibility (PV-FL) is used to forecast demand for supplier, where firms can cooperate while maintaining privacy (see Figure 2). In no visibility scenario, only the supplier data is available. We apply ARIMA model (NV-ARIMA) (Babai et al. 2013) and NN model (NV-NN) (Feizabadi 2022) with no visibility to other firms to predict the demand for supplier.



(a) LSTM model with full visibility (FV-LSTM) – Retailer and wholesaler states as input.

(b) NN without visibility (NV-NN) – Only previous $\mathscr{T}$ time steps of demand for supplier as input.

Figure 3: Model structure for demand forecasting.

FV-LSTM model is a NN model composed of two LSTM layers and one FC layer (See Figure 3a). It takes the previous eight time step states of retailer and wholesaler as input, including inventory level and total order quantity. Compared to our proposed FL model, FV-LSTM model has a similar model structure with nearly the same total number of parameters inside the model and also uses $Min-Max$ to normalize the input. The hyper parameters of the ML layer are as follows: input layer dimension=5; number of hidden states in LSTM=64; and the FC neurons=20.

The hyper parameters of the FL model are as follows – for *supplier* and *wholesaler* models: input layer dimension=2, and number of hidden states in LSTM=64; for *aggregation* model: number of neurons=5; and for *supplier* model: first layer neurons=5; and second layer neurons=2.

For the no visibility scenario, the input data are the time-series sequence of supplier data. NV-ARIMA model takes previous eight time steps of demand for supplier as input and has three hyper parameters: p, d, and q, which are non-negative integers. In this case, we apply grid search technique to tune the hyper parameters where $p = [0,1,2,4,6,8,10]$, $d = [0,1,2]$, and $q = [0,1,2]$. The ARIMA model with the best performance in grid search will be used. NV-NN model has 2 FC layers, similar to (Feizabadi 2022). There are 5 neurons in the first FC layer and 2 neurons in the second FC layer (see Figure3b).

For all the above-mentioned NN and LSTM models, they are trained using the same parameters from Kingma and Ba (2014): epochs number=150, batch size=32, learning rate=0.001, and optimizer=Adam.

Table 1: Average MAE results.

Method	Visibility	c=2, d=30	c=5, d=50	c=10, d=100
NV-NN	No	38.7554	35.4623	61.2887
NV-ARIMA	No	31.0998	32.3612	46.8307
PV-FL	Partial	25.5622	29.8184	41.8509
FV-LSTM	Full	29.1657	23.7547	39.2612

4.3 Results

In our experiment, we evaluated different methods using the mean absolute error (MAE) between the prediction value and actual demand. We use 3 different sets of parameters in the simulation, as mentioned above, and each set of parameters repeatedly run 10 times with different initial simulation state. Then we compare the average MAE in these 10 runs among all methods.

The simulation data are divided into 60% training data, 20% evaluation data, and 20% testing data. All methods train on the same training dataset. For the ARIMA model, we first pick a set of hyper parameters with the minimum loss in the evaluation dataset and compute the MAE of this set of hyper parameters in test dataset.

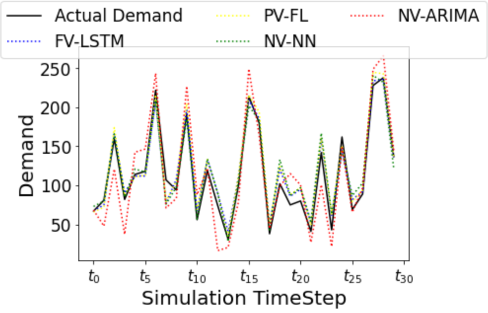
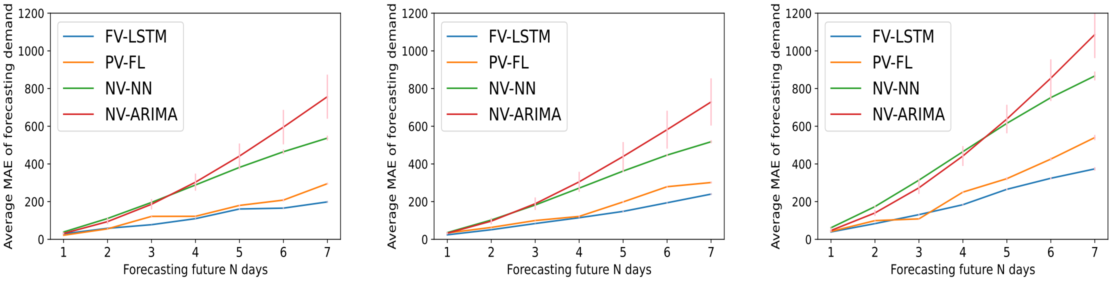


Figure 4: Actual data vs prediction in full, partial and no visibility methods in a month.

First, we compare the actual demand for supplier with forecasted next day demand. Figure 4 shows a month simulation time of the actual demand for supplier with the forecasted demand using different methods across the time for Exp. 3 ($c = 10$, and $d = 100$). In the figure, the black line is the real demand and the dashed lines represent forecast demand using various approaches. Table 1 shows the average MAE for different methods in the three experimental configurations. Overall, the MAE illustrates that our approach outperforms the methods with no visibility (i.e, NV-NN and NV-ARIMA), and achieves similar performance to the full visibility method (i.e, FV-LSTM). In the first experiment, $c = 2$, and $d = 30$, our model achieves even smaller MAE than the FV-LSTM model. A possible explanation for this could be that the structure of PV-FV fits well for forecasting the next day demand. Since the PV-FV model takes retailer and wholesaler inputs separately and the next day demand for supplier mainly depends on the inputs of wholesaler, the PV-FV model with a separate structure has the advantage to handle the situation.

Figure 5 demonstrates the average MAE results of future 1 to 7 days demand for supplier among three experiments. The blue line is the LSTM model with full visibility (FV-LSTM), the orange line is our approach (PV-FL), the green line is the neural network without visibility (NV-NN), and the red line is the ARIMA model without visibility (NV-ARIMA). The x axis represents forecasting of future $\mathscr{T}$ days and the y axis is the average MAE results. From Figure 5, the average MAE of the methods without visibility raise quickly due to lack of downstream information. The average MAE of our approach and FV-LSTM increases at a slower rate compared to methods without visibility. In addition, we can see from the error bar that our approach has small variances. This also indicates our stable performance under different scenarios.



(a) Exp. 1

(b) Exp. 2

(c) Exp. 3

Figure 5: MAE of different forecasting methods with increasing forecasting days.

The above experiment results demonstrate that our method can increase supply chain visibility and achieves similar performance to the method with full visibility. The reason is that our approach considers inputs from downstream firms and those inputs are highly correlated with demand for supplier. As such, our method is able to utilize the LSTM layers to extract the hidden state from the input features and uses the FC layers to analyze the hidden state.

5 CONCLUSION

In this paper, we demonstrated an approach to forecast demand for supplier by increasing supply chain visibility while preserving data privacy. We proposed a split-NN FL model using simulated data from a three-stage supply chain model. Intermediate data from the partial models are aggregated at a third party trust server to preserve privacy. The proposed FL model is evaluated with other demand forecasting methods with different supply chain visibility. The experimental results show that the proposed FL model has better performance than methods without visibility and manages to achieve close performance to full visibility method. Hence, the work presented in the paper demonstrates that supply chain visibility can be increased by exchanging intermediate data in the FL model.

In the future, we plan to improve the realism of our supply chain simulation with more transaction details and entities. Alternatively, real data can be also used in the FL model depending on the availability of the data. In addition, our FL model can be also extended to handle dynamic supply chains where different entities can join and leave the supply chain network.

ACKNOWLEDGMENTS

This work was supported by the A*STAR Cyber-Physical Production System (CPPS) – Towards Contextual and Intelligent Response Research Program, under the RIE2020 IAF-PP Grant A19C1a0018, and Model Factory@SIMTech

REFERENCES

Abadi, M., A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. 2016. “Deep Learning with Differential Privacy”. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, edited by E. Weippl and S. Katzenbeisser, 103–114. New York: Association for Computing Machinery.

Abbasimehr, H., M. Shabani, and M. Yousefi. 2020. “an Optimized Model Using LSTM Network for Demand Forecasting”. *Computers & Industrial Engineering* 143:106435.

Atallah, M., M. Bykova, J. Li, K. Frikken, and M. Topkara. 2004. "Private Collaborative Forecasting and Benchmarking". In *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*, edited by V. Atluri, 103–114. New York: Association for Computing Machinery.

Babai, M. Z., M. M. Ali, J. E. Boylan, and A. A. Syntetos. 2013. "Forecasting and Inventory Performance in a Two-stage Supply Chain with ARIMA (0, 1, 1) Demand: Theory and Empirical Analysis". *International Journal of Production Economics* 143(2):463–471.

Barlas, Y., and B. Gunduz. 2011. "Demand Forecasting and Sharing Strategies to Reduce Fluctuations and the Bullwhip Effect in Supply Chains". *Journal of the Operational Research Society* 62(3):458–473.

Barratt, M., and R. Barratt. 2011. "Exploring Internal and External Supply Chain Linkages: Evidence from the Field". *Journal of Operations Management* 29(5):514–528.

Barratt, M., and A. Oke. 2007. "Antecedents of Supply Chain Visibility in Retail Supply Chains: a Resource-based Theory Perspective". *Journal of Operations Management* 25(6):1217–1233.

Chase Jr, C. W. et al. 2016. "Machine Learning is Changing Demand Forecasting". *The Journal of Business Forecasting* 35(4):43.

Chen, Y., and Ö. Özer. 2019. "Supply Chain Contracts that Prevent Information Leakage". *Management Science* 65(12):5619–5650.

Feizabadi, J. 2022. "Machine Learning Demand Forecasting and Supply Chain Performance". *International Journal of Logistics Research and Applications* 25(2):119–142.

Hamadneh, S., O. Pedersen, and B. Al Kurdi. 2021. "an Investigation of The Role of Supply Chain Visibility into The Scottish Blood Supply Chain". *Journal of Legal, Ethical and Regulatory Issues* 24:1–13.

Hard, A., K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. "Federated Learning for Mobile Keyboard Prediction". https://arxiv.org/abs/1811.03604, accessed 8th November 2018.

Hochreiter, S., and J. Schmidhuber. 1997. "Long Short-term Memory". *Neural computation* 9(8):1735–1780.

Hong, Y., J. Vaidya, N. Rizzo, and Q. Liu. 2018. "Privacy-preserving Linear Programming". In *World Scientific Reference on Innovation: Volume 4: Innovation in Information Security*, 71–93.

Hong, Y., J. Vaidya, and S. Wang. 2014. "a Survey of Privacy-aware Supply Chain Collaboration: from Theory to Applications". *Journal of Information Systems* 28(1):243–268.

Kim, C. O., I.-H. Kwon, and C. Kwak. 2010. "Multi-agent Based Distributed Inventory Control Model". *Expert Systems with Applications* 37(7):5186–5191.

Kingma, D. P., and J. Ba. 2014. "Adam: A Method for Stochastic Optimization". https://arxiv.org/abs/1412.6980, accessed 22th December 2014.

Kochak, A., and S. Sharma. 2015. "Demand Forecasting Using Neural Network for Supply Chain Management". *International Journal of Mechanical Engineering and Robotics Research* 4(1):96–104.

Konečnỳ, J., H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. 2016. "Federated Learning: Strategies for Improving Communication Efficiency". https://arxiv.org/abs/1610.05492, accessed 18th October 2016.

Kong, G., S. Rajagopalan, and H. Zhang. 2013. "Revenue Sharing and Information Leakage in a Supply Chain". *Management Science* 59(3):556–572.

Ma, J., M. Kwak, and H. M. Kim. 2014. "Demand Trend Mining for Predictive Life Cycle Design". *Journal of Cleaner Production* 68:189–199.

Merkuryeva, G., A. Valberga, and A. Smirnov. 2019. "Demand Forecasting in Pharmaceutical Supply Chains: a Case Study". *Procedia Computer Science* 149:3–10.

Nenni, M. E., L. Giustiniano, and L. Pirolo. 2013. "Demand Forecasting in the Fashion Industry: a Review". *International Journal of Engineering Business Management* 5:37.

Nikolopoulos, K., S. Punia, A. Schäfers, C. Tsinopoulos, and C. Vasilakis. 2021. "Forecasting and Planning During a Pandemic: COVID-19 Growth Rates, Supply Chain Disruptions, and Governmental Decisions". *European Journal of Operational Research* 290(1):99–115.

Özener, O. Ö., and Ö. Ergun. 2008. "Allocating Costs in a Collaborative Transportation Procurement Network". *Transportation Science* 42(2):146–165.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In *Advances in Neural Information Processing Systems 32*, 8024–8035.

Ried, L., S. Eckerd, L. Kaufmann, and C. Carter. 2021. "Spillover Effects of Information Leakages in Buyer–Supplier–Supplier Triads". *Journal of Operations Management* 67(3):280–306.

Seyedan, M., and F. Mafakheri. 2020. "Predictive Big Data Analytics for Supply Chain Demand Forecasting: Methods, Applications, and Research Opportunities". *Journal of Big Data* 7(1):1–22.

Sheller, M. J., B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen et al. 2020. "Federated Learning in Medicine: Facilitating Multi-institutional Collaborations Without Sharing Patient Data". *Scientific Reports* 10(1):1–12.

Somapa, S., M. Cools, and W. Dullaert. 2018. "Characterizing Supply Chain Visibility–a Literature Review". *The International Journal of Logistics Management*.

Tan, K. H., W. P. Wong, and L. Chung. 2016. "Information and Knowledge Leakage in Supply Chain". *Information Systems Frontiers* 18(3):621–638.

Vepakomma, P., O. Gupta, and R. Raskar. 2018. "Split Learning for Health: Distributed Deep Learning Without Sharing Raw Patient Data". https://arxiv.org/abs/1812.00564, accessed 3rd December 2018.

Wang, G. 2012. "Demand Forecasting of Supply Chain Based on Support Vector Regression Method". *Procedia Engineering* 29:280–284.

Wang, G., A. Gunasekaran, E. W. Ngai, and T. Papadopoulos. 2016. "Big Data Analytics in Logistics and Supply Chain Management: Certain Investigations for Research and Applications". *International Journal of Production Economics* 176:98–110.

Wang, X., and S. M. Disney. 2016. "The bullwhip Effect: Progress, Trends and Directions". *European Journal of Operational Research* 250(3):691–701.

Yang, Q., Y. Liu, T. Chen, and Y. Tong. 2019. "Federated Machine Learning: Concept and Applications". *ACM Transactions on Intelligent Systems and Technology (TIST)* 10(2):1–19.

Zhang, C., and C. Zhang. 2007. "Design and Simulation of Demand Information Sharing in a Supply Chain". *Simulation Modelling Practice and Theory* 15(1):32–46.

AUTHOR BIOGRAPHIES

BO ZHANG is a PhD student in the School of Computer Science and Engineering at NTU. His research interests include, simulation optimization, machine learning, data analytics. His e-mail address is bo003@e.ntu.edu.sg.

WEN JUN TAN is a postdoctoral research fellow at Nanyang Technological University (NTU), Singapore. He received his Ph.D. in Computer Science in 2020 from NTU. His research interests include cyber-physical systems, large-scale parallel and distributed simulations, cloud-based workflow management systems. His e-mail address is wjtan@ntu.edu.sg.

WENTONG CAI is a Professor in the School of Computer Science and Engineering at NTU. His expertise is mainly in the areas of Modeling and Simulation and Parallel and Distributed Computing. He is an associate editor of the ACM Transactions on Modeling and Computer Simulation (TOMACS) and an editor of the Future Generation Computer Systems (FGCS). His email address is aswtcai@ntu.edu.sg.

ALLAN NENGSHENG ZHANG is a deputy director at the Singapore Institute of Manufacturing Technology (SIMTech). His research interests include last mile logistics, supply chain visibility and risk, project planning and optimization, AI and data analytics. His email address is nzhang@simtech.a-star.edu.sg.