Improving Sales Forecasting by Combining Key Account Managers' Inputs and Models Such as SARIMA, LSTM, and Facebook Prophet

Agneta Ramosaj
University of Fribourg

Nicolas Ramosaj
University of Applied Sciences and Arts, Western Switzerland

Marino Widmer
University of Fribourg

Sales forecasting is important for a company to plan its production. The quality of its forecasts influences finances and the product availability. The impact of sales forecasts on a company may result on an immobilization of cash flow by causing a high stock level, which is the opposite of out-of-stock impact. The purpose of this study was to find a suitable model for predicting the best company sales forecasts that has a better accuracy or production plan. The proposed method includes an adjustment of the prediction model by including the key account managers' expertise as qualitative forecasting method. This adjustment was analyzed using different time series forecasting techniques such as exponential smoothing, seasonal autoregressive integrated moving average and Facebook Prophet. These techniques were compared in parallel with neural network approaches such as long-short term memory. Comparisons were made using root mean square error and residual stock to determine whether the forecasts were too optimistic or pessimistic. The proposed model is dynamic. Adjustments of the qualitative inputs could directly influence the proposed values obtained using different quantitative methods.

Keywords: demand forecast, exponential smoothing, SARIMA (seasonal autoregressive integrated moving average), Facebook Prophet, LSTM (long-short term memory), KAM (key account manager)

INTRODUCTION

Sales forecasting is becoming an important subject even in small and medium enterprises (SMEs). Poor predictions have several negative impacts for companies, such as overstock (immobilization of cash flow) or stockout and lack of components. Improving sales accuracy means also improving the company's future business projections (Haberleitner et al., 2010).

For decades, it has been proved many times that quantitative forecasting models provide better results than qualitative forecasting models (Ramosaj & Widmer, 2020). The most well-known quantitative model is time series forecasting, in which historical observations are collected and analysed to develop an applicable model. The goal of time series forecasting has often been to improve forecast accuracy (Siami-

Namini et al., 2018). Time series methods have been applied to improve different areas such as forecasting the power load for the electricity market (Bozkurt et al., 2017), wind energy production (Hui et al., 2012), food retail demand (Pereira Da Veiga et al., 2014), road and traffic optimisation (Zhao et al., 2017), cryptocurrency exchange rates (Chen et al., 2021), COVID-19 infection rates (Wang et al., 2020), air pollution levels (Rani Samal et al., 2019), and warranty demand (Xie et al., 2017).

Lately, hybrid methods have been used to improve forecasting accuracy, such as auto-regressive integrated moving average (ARIMA) and artificial neural network (ANN) combined with empirical mode decomposition (Büyüksahin & Ertekina, 2019). Hybrid models such as long short-term memory (LSTM) are often compared with ARIMA (Chen et al., 2021). However, comparisons have also been made between ARIMA and Facebook Prophet (Taylor & Letham, 2017) or even between ARIMA, LSTM, and Facebook Prophet (Chikkakrishna et al., 2019). Many articles (e.g. Weytjens et al., 2019) have compared ARIMA and Facebook Prophet with LSTM multi-layers for cash flow prediction or demand forecasting based on multi-layer LSTM networks (Abbasimehr et al., 2020).

The aim of this study was to improve the sales forecasting of Swiss SMEs by using different time series forecasting models. Sales forecasts and real values were considered for 44 months, from January 2018 to August 2021. These data contain the sales in approximately 50 markets and 150 stock-keeping units.

The first model used was the well-known ARIMA, which is composed of auto-regressive models (AR), and moving average (MA). In the company used in the case study, a seasonal effect had been occurring every November, and seasonal auto-regressive integrated moving average (SARIMA) had been providing better results than ARIMA in this case. SARIMA predictions were compared with those of exponential smoothing and Facebook Prophet. In parallel, a neural network was developed using long-short term memory (LSTM).

After the four models were compared, a new factor was introduced through the key account managers (KAMs). Several KAMs were responsible for different sales markets. Each KAM is responsible for at least one market. KAMs are responsible for the sales and work in the front line to have a better overview of the sales by receiving all purchase orders from all markets. Owing to the information provided by the KAMs or 'KAM forecasts', the models could be consolidated and improved by adding weight to the KAM forecasts and the different predictions by the tested models.

This article is organized as follows: Next section refers to the state of the art of some existing forecasting methods. The following section outlines the proposed methodology used to solve the present research problem. Then, a section discussing the experimental set up and results and comparisons is presented, and the last section concludes the paper.

EXISTING FORECASTING MODELS

In this section, we describe and compare the four models (exponential smoothing, SARIMA, Facebook Prophet, and LSTM) to approximate the forecasts of the dataset. By knowing the influence of seasonality on the dataset due to the evolution of the market (higher sales in November), the models could be directly applied with the seasonal factor. The models were tested over the last 12 months before t (today), and the rest of the dataset was used for the training.

Exponential Smoothing

This is the most frequently used forecasting method (Stadtler & Kilger, 2002). In this method, only three types of data are needed to forecast the value for period t (F_t): the exponentially smoothed forecast made for the previous period (F_{t-1}), the actual demand in the previous period (A_{t-1}), and a smoothing constant alpha (α), which provides the weight of the committed error.

These data were used to calculate Equation 1:

$$F_t = F_{t-1} + \alpha\,(A_{t-1} - F_{t-1}) \tag{1}$$

SARIMA

ARIMA forecasting has been frequently used for reliable forecasts (Ho & Xie, 1998). The principal limitation of ARIMA is that it assumes time series as linear (Adhikari & Agrawal, 2013). ARIMA is one of the most widely used approach for time series forecasting by describing the auto-correlation in the data (Hyndman & Athanasopoulos, 2018). The initial parameters used are the order ARIMA(p,d,q), as they capture the key elements of the model (auto-regression [AR], integrated [I], and MA) by, for example, associating the key to the parameters by AR(p), I(d), and MA(q). The value of each parameter can be explained using tools such as the auto-correlation function (ACF), partial ACF, and the stationarity of the dataset measured by the p-value (the time series should be differentiated if the result is >5%). The tools are explained in the section dedicated to the proposed methodology.

ARIMA could be written as in Equation 2:

ARIMA (p,d,q):

$$\underbrace{(1 - \text{Ø}_1 B)}_{\text{Non-seasonal AR(1)}}\ \underbrace{(1 - B)}_{\text{Non-seasonal difference}}\ y_t = \underbrace{(1 + \theta_1 B)}_{\text{Non-seasonal MA (1)}}\ \varepsilon_t, \tag{2}$$

where p is the order of the auto-regressive part, d is the degree of the first differencing involved (I), and q is the order of the MA part.

To include the seasonality of the dataset, a seasonal ARIMA model was developed by including additional seasonal terms in the ARIMA, such as the seasonal_order (P, D, Q, Seasonality), as mentioned in SARIMA (Hyndman & Athanasopoulos, 2018).

SARIMA could be expressed as in Equation 3:

$$\text{SARIMA}\quad \underbrace{(p,d,q)}_{\text{Non-seasonal part of the model}}\quad \underbrace{(P,D,Q)s}_{\text{Seasonal part of the model}}, \tag{3}$$

where s is the number of observations in which uppercase notation could be observed and defined as the seasonal parts (s) of the model, as depicted in Equation 4.

$$\underbrace{(1- \text{Ø}_1 B)}_{\text{Non seasonal AR (1)}}\ \overbrace{(1-\phi_1 B^s)}^{\text{Seasonal AR(1)}}\underbrace{(1-B)}_{\text{Non-Seasonal difference}}\overbrace{(1-B^s)}^{\text{Seasonal difference}}y_t=\underbrace{(1+\theta_1 B)}_{\text{Non-seasonal MA (1)}}\ \overbrace{(1+\Theta_1 B^s)}^{\text{Seasonal MA (1)}}\varepsilon_t. \tag{4}$$

By training SARIMA on the different product categories, the values of parameters d and D remain 0 owing to the stationarity of the dataset. During the training, parameters p, q, P, and Q were applied and fine-tuned, and Seasonality was given by the ACF. The ACF and partial ACF (PACF) helped identify parameters AR(p) and MA(q) for the model. The auto-correlation (or correlation of the time series with itself) is the key statistics in time series analyses. The ACF measures the extent of the linear relationship between two variables. It is used as tool to explore the time series before forecasting, which helps check for seasonality, cycles, and other time series patterns. The PACF measures the degree of association between two variables that have no direct correlation.

Facebook Prophet

Prophet is an open-source framework used in Facebook for time series forecasting (Wang et al., 2020). It has three main components: trend, $g(t)$; seasonality, $s(t)$; and holidays, $h(t)$ (Taylor & Letham, 2017). They are combined together in Equation 5:

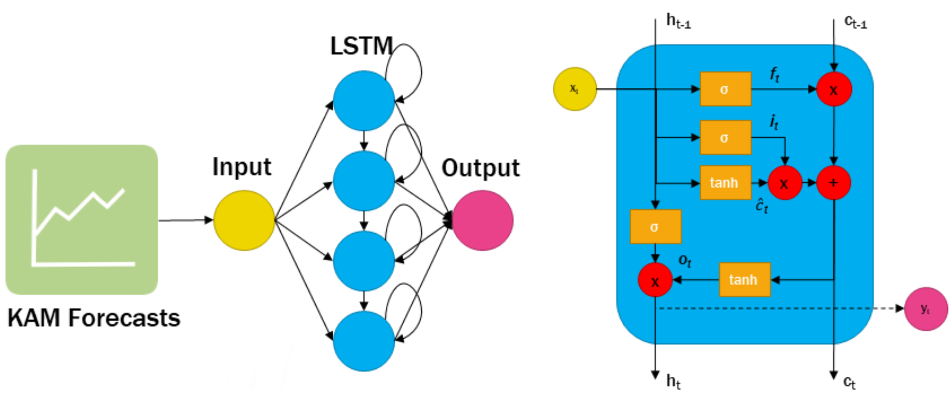
$$y(t) = g(t) + s(t) + h(t) + \epsilon(t), \text{ where } \epsilon(t) \text{ is the error term.} \quad (5)$$

This model has gained attention for its potential usefulness for seasonal datasets.

LSTM

LSTM is the most commonly used model in recurrent neural networks (RNNs). One of its most known advantages is the ability to memorise the data sequence (Siami-Namini et al., 2018). It was proposed by Hochreiter and Schmidhuber (1997). The initial parameters used in common are the hidden layer called LSTM containing 4 neurons, and a single output with 12 predicted values. The training is done on 250 epochs, and the train set should not be shuffled. Figure 1 presents an overview of the implementation of the LSTM predictor.

FIGURE 1
DESIGNED BY THE AUTHORS - LSTM WITH KAM



LSTM can be divided into three parts known as gates, which perform individual functions. The first part, the forget gate, checks whether the incoming information is relevant or irrelevant. If it is irrelevant, it can be forgotten; otherwise, the information must be remembered. The second part, the input gate, tries to learn new information from the input of the cell and decides which information from the current step could be added. In the third part, or the output gate, the current period provides the updated information to the next period. The output gate finalises the next hidden state.

The notations are as follows:

- x_t: the input value of the current period
- f_t: the forget gate of the current period
- h_t and h_{t-1}: the hidden states for the current and previous periods, representing the short-term memory.
- c_t and c_{t-1}: the cell states for current and previous periods, representing the long-term memory.
- σ: sigmoid activation function (a mathematical function with an 'S-shaped curve' characteristic)
- tanh: non-linear activation (allows multiple layers of neurons to learn the error)

- i_t: input gate for the current period
- $\hat{c}_t$: value generated by tanh
- o_t: output gate for the current period
- y_t : output value for the current period

Forget Gate

This gate helps decide which information in the sequence is useful. If the information is useful, it should be kept; otherwise, it should be thrown away. The equation of the forgot gate is presented in Equation 6:

$$f_t = \sigma(x_t \times W_f + h_{t-1} \times w_f) , \tag{6}$$

where W_f is the weight matrix between the forget and input gates and w_f is the weight associated with the input between the forget and input gates.

A sigmoid function is applied to the current input x_t and hidden state h_{t-1}. The hidden state represents the output of an LSTM cell. The sigmoid function generates values between 0 and 1. If the value is closer to 1, it means that the old output is necessary; otherwise, if it is close to 0, this output is irrelevant and can be ignored.

Input Gate

Through the input gate, the importance of the information can be quantified. The function of the input gate is presented in Equation 7:

$$i_t = \sigma(x_t \times W_i + h_{t-1} \times w_i) , \tag{7}$$

where W_i is the weight matrix between the input and output gates and w_i is the weight associated with the input between the input and output gates.

The new information passes through the cell state (representing the internal cell of the LSTM, which is not an output), with the function presented in Equation 8:

$$\hat{c}_t = \tanh (x_t \times W_c + h_{t-1} \times w_c) \tag{8}$$

where W_c is the weight matrix between the cell state information and the output gate and w_c is the weight associated with the input between the cell state information and output gate. With the tanh value, new information between −1 and 1 can be obtained. If the $\hat{c}_t$ value is positive, it is added to the cell state; otherwise, it is subtracted.

The next step is to decide if the information should be stored to the cell state.

$$c_t = f_t \times c_{t-1} + i_t \times \hat{c}_t. \tag{9}$$

The new cell state was determined by multiplying the cell state of the previous period to the forget gate of the current period, to which we added the value generated in the input gate multiplied to the tanh value generated. If the outcome was 0, then the values were dropped from the cell state.

Output Gate

The output gate provides information about the values of the next hidden state. The hidden state is used for predictions. The functions are presented in Equations 10 and 11:

$$o_t = \sigma(x_t \times W_o + h_{t-1} \times w_o) \tag{10}$$

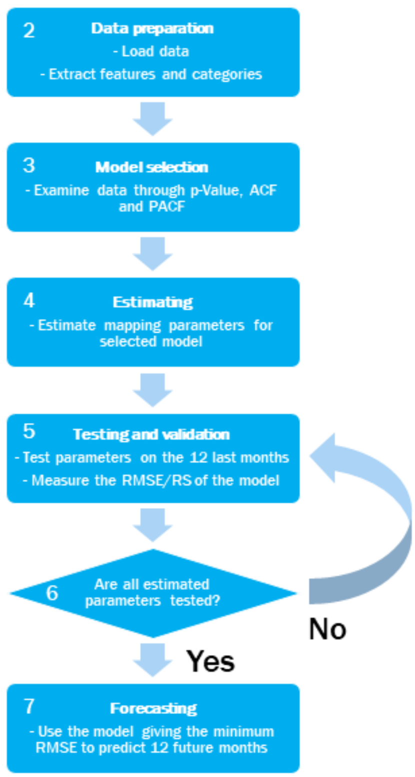
$$h_t = o_t \times \tanh(c_t), \quad (11)$$

where W_o is the weight matrix of the output gate and w_o is the weight associated with the input of the output gate. LSTM helps learn the forecast errors made by the KAMs.

THE PROPOSED METHODOLOGY

In order to improve the current forecasts, the following sequential methodology has been applied, as shown in Figure 2:

FIGURE 2
ANALYSIS PROCESS DESIGNED BY THE AUTHORS



(1) The data used for the evaluation of the different methods were provided by a Swiss SME. The sales history were available from January 2018 to August 2021.
(2) The sales history were filtered to obtain time series according to a global view and then deeper according to product category. The features such as the real sales and key account manager's forecasts are also extracted.
(3) The purposed time series were evaluated using the ACF, partial ACF (PACF), and p-value to identify the parameters to be assigned in the model. The ACF computes and presents how the auto-correlation evolves through the lags, and as a complement, the PACF allows for the suppression of the influence of the lag between lag0 and lagt. The stationarity of the dataset was evaluated with the p-value by a critical step of 5%; over this step, the time series should be differentiated.
(4) The next step involved the estimation of the mapping of the hyperparameters for each chosen model. The hyperparameters were configuration parameters that were external to the model,

whose values could not be estimated from the data. They were set before the training was started. They helped define the higher-level properties of the mapping function and learning process to inspect the performance on data.

(5) The dataset was split into a train set and a test set. The test set contained the most recent sales values (12 values represents 1 year of sales forecasting). The model trains the train set with the given hyperparameters and forecasts the sales.

(6) The operation was repeated until all combinations of the hyperparameters were visited. The metric root mean square error (RMSE), defined in Equation 12, was computed for each combination and helped define the optimal hyperparameters for the model by using the hyperparameters from the min (RMSE).

$$\mathrm{RMSE}= \sqrt{\frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\left(\mathrm{Y_i}-\widehat{\mathrm{Y}}_{\mathrm{i}}\right)^2} \quad where\ n \in [1,\ldots,n] \tag{12}$$

In the equation, n is the number of data points or observations, Y_i is the observed values (test set), and $\widehat{Y}_\mathrm{i}$ is the predicted values.

(7) The model with the best hyperparameters was used to make the forecasts, and the residual stock (RS), defined in Equation 13, was computed for each timestamp (every month), such as qualifying the model to be too optimistic or pessimistic. (A too optimistic model is of interest when the current stock is low and the company wants a margin of stock, whereas a too pessimistic model is of interest when the current stock is high and the company wants to reduce the stock of the product.)

$$RS = \sum_{i=1}^{n}\left(\widehat{\mathrm{Y}}_{\mathrm{i}}-\mathrm{Y_i}\right)\ where\ n \in [1,\ldots,n] \tag{13}$$

$\widehat{\mathrm{Y}}_\mathrm{i} > \mathrm{Y_i}$ yields a too optimistic forecast, whereas $\widehat{\mathrm{Y}}_\mathrm{i} < \mathrm{Y_i}$ yields a too pessimistic forecast.

Further Investigation

(8) Forecasts given by KAMs are more accurate than forecasts given by models over a short term (today + 3 months). The combination of the two forecasting methods is relevant when using a set of weights given an important ponderation of the KAMs' forecasts in the short term and an important ponderation of the model in the long term. The sum of both weights should be equal to 1.

(9) A deeper approach is to combine the forecasts given by the KAM and the so-called LSTM model to learn the error on the KAM's forecasts and predict an adjustment of the KAM's forecasting.

EXPERIMENTAL SETUP AND RESULTS

Three statistical time series forecasting methods (exponential smoothing, SARIMA, and Facebook Prophet) and one neural network (LSTM) were considered to evaluate the performance of the proposed approach. All the models were implemented in Python: Exponential Smoothing, SARIMA and FB Prophet were executed and optimized using the statsmodels package while the LSTM was performed using Keras. To compare all models, the RMSE and RS metrics were used.

Performance of the Different Models

The statistical time series forecasting methods were benchmarked through the process purpose in the previous section. The results are illustrated later. The main observation is that SARIMA remains the best choice for the purposed dataset by providing the best RMSE for all product categories. Exponential smoothing and Facebook Prophet did not satisfy the request of the dataset, as they provided too optimistic or pessimistic RSs.

Evaluation With the Residual Stock

Visualization by the RS is interesting to observe over the months when the selected model is too optimistic or pessimistic. The main challenge is to provide a good forecast for the month of November because of promotional days during this month, such as Black Friday, Cyber Monday, and Singles' Day. The selling is difficult to predict. The benefit of introducing KAM forecasts is a valuable input that is useful in identifying the more accurate model, such as correcting the gap between the real and model-predicted sales.

Introduction of the Key Account Manager Forecasts

KAMs are better in predicting the short-term sales. They keep contact with the retailer and know the needs of each market owing to the order they receive. However, KAMs make optimistic forecasts when examining the RS only on the basis of their forecasts; in which case, the RS is always observed to be high. Awareness of this information allows for adding the KAM forecasts as 'feedforward' for the prediction by a simple weighting or learning method.

KAM Forecasts and Statistical Time Series Forecasting Methods

The method consists of an integration of KAM forecasts and the time series forecasting method by giving weight to each forecast. The weight is given on both curves for each timestamp. A high weight is given to the KAM in the short term, and the rest of the weight is given to the statistical time series forecasting methods. For the long term, a high weight is given to the statistical time series forecasting methods, and the rest of the weight is given to the KAM forecasts. The results of the combination of the two curves were analyzed using the RMSE and RS.

In our case, consider the weights as the predictions from the different forecasting models and KAM forecasts. The forecasts were improved in terms of RMSE. The weight applied was 0.9*KAM + 0.1*prediction_model for the forecasts for the first three months, 0.3*KAM + 0.7*prediction_model for the forecasts for months 3 to 6, and 0.1*KAM + 0.9*prediction_model for the forecasts for months 6 to 12. The accuracy of the RS analysis was better by combining the prediction models and the KAM forecasts.

KAM Forecasts and Learning Method LSTM

The approach consists of learning the KAM curve through a RNN called LSTM. The neural network is trained by the history of KAM forecasts and labelled by the real sales. The timestamp values were predicted using LSTM.

KAM Forecasts, Statistical Time Series Forecasting Methods and Learning Methods LSTM

Both previous methods performed to the last method presented, that is, a combination of KAM forecasts, statistical time series forecasting methods, and learning methods. The principle is to train real sales history through a statistical time series method and predict the future based on the trained models. This will provide an additional 'feedforward' to the learning methods by creating for each timestamp a tuple of the KAM forecast value and value obtained using the statistical time series forecasting method. A tuple is an ordered and unchangeable collection in Python. Each list of tuples will be trained on LSTM, and the model will be used to predict future sales.

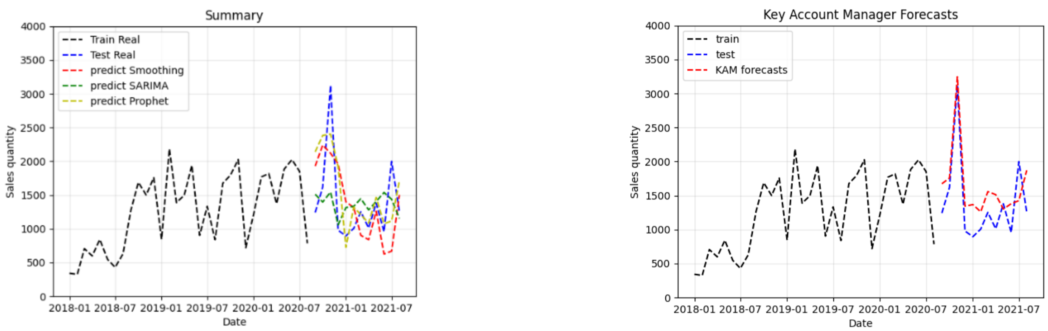
Results

The results are presented in sub-chapters in order to follow the steps of the development.

Statistical Time Series Forecasting Methods

By using only the canonical forecasting method, it was observed that the model made good forecasts when the variation was lower than the average but could not reveal high seasonality peaks. By contrast, KAMs know the market better in the short term but does not make good long-term forecasts.

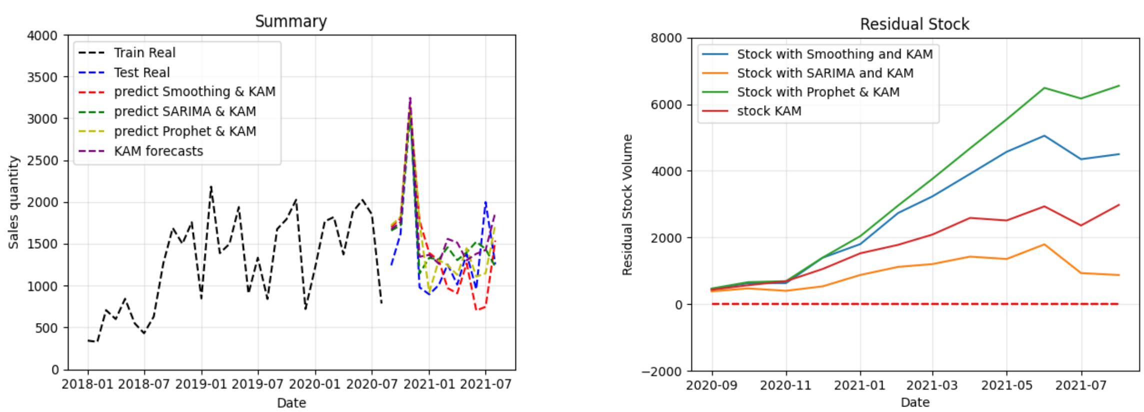
FIGURE 3
THE LEFT GRAPH PRESENTS THE CANONICAL MODEL OF FORECASTS, AND THE RIGHT GRAPH PRESENTS KAM FORECASTS



KAM Forecasts and Statistical Time Series Forecasting Methods

By combining KAM forecasts and models by assigning weight to both, a weak RS variation was predicted for the first months. The plus value was observed in the medium-long term, especially with SARIMA, which presented the best potential to forecast this type of dataset.

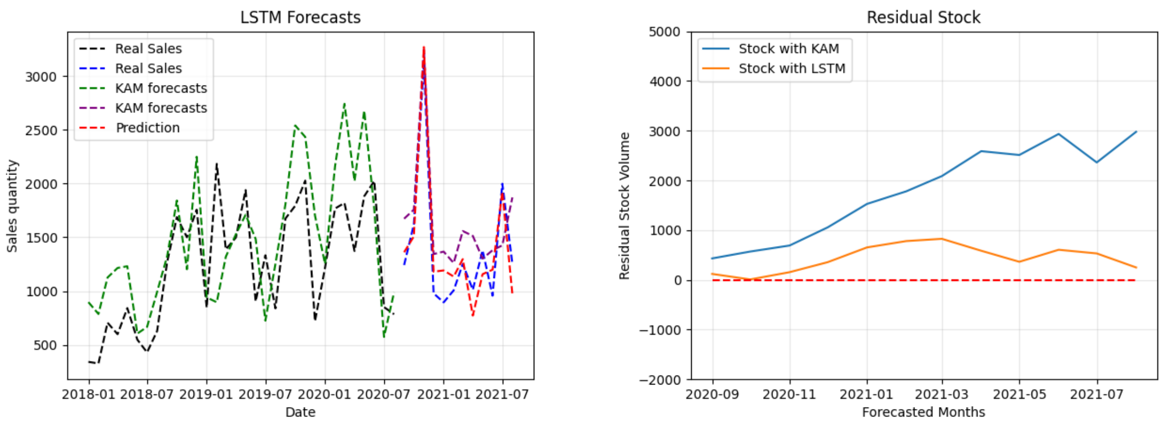
FIGURE 4
THE LEFT GRAPH PRESENTS THE PREDICTION MODELS COMBINED WITH KAM FORECASTS, AND THE RIGHT GRAPH PRESENTS THE RS OF THE PREDICTIONS



KAM Forecasts and Learning Method LSTM

Considering only KAM forecasting and KAM's behaviour too optimistic or too pessimistic, the strategy consists on an error learning of KAM forecasts due to adjust the prediction which helps to be more accurate.

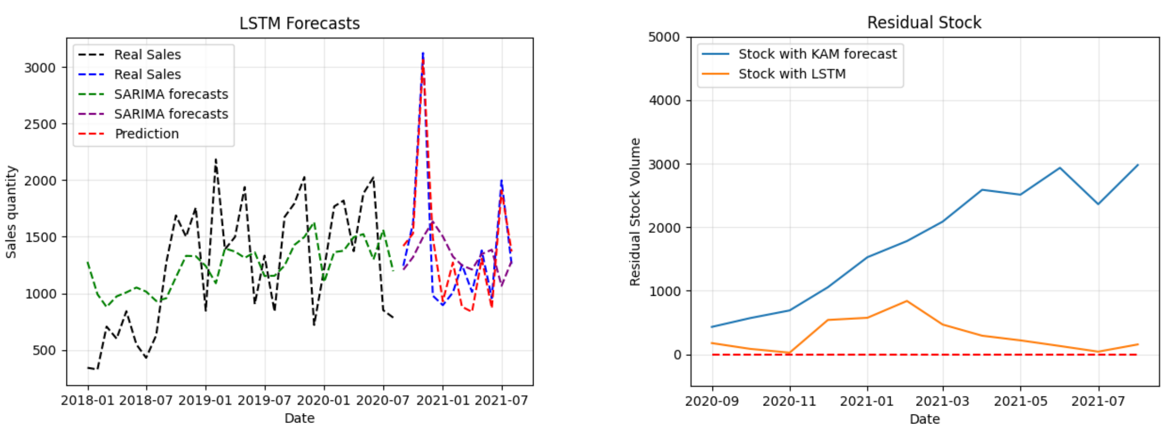
FIGURE 5
THE LEFT GRAPH PRESENTS THE PREDICTION BY LEARNING THE ERROR OF THE KAM FORECASTS, AND THE RIGHT GRAPH PRESENTS THE RS OF THE PREDICTIONS



KAM Forecasts, Statistical Time Series Forecasting Methods and LSTM Learning Method

The deeper approach consists of combining KAM forecasting and forecast models such as SARIMA through LSTM. It also consists of a prediction model that fine-tunes the previous method.

FIGURE 6
THE LEFT GRAPH PRESENTS THE PREDICTION BY LEARNING THE ERROR OF THE KAM FORECASTS AND SARIMA PREDICTION, AND THE RIGHT GRAPH PRESENTS THE RS OF THE PREDICTIONS



Results Summary

TABLE 1
SUMMARY OF RESULTS

Quantitative or qualitative method	Overall quantity (in units)	Delta with real	RMSE	Overall quantity (incl. KAM) (in units)	Delta with real (incl. KAM)	RMSE (incl. KAM)
Real sales	16‘718	0%	0			
KAM	19‘695	+17.8%	176.58			
Exponential Smoothing	21‘172	+26.6%	194.56	21‘217	+26.9%	204.09
SARIMA	16‘161	-3.3 %	109.71	17‘593	+5.2%	153.47
Facebook Prophet	24‘025	+43.7%	237.18	23‘268	+39.1%	231.59
LSTM & SARIMA	17‘236	+3.1%	108.18	16‘885	+1.0%	98.14

Scale: %; -5% to 0%; 0% to 10%; 10% to 20%; >20%

SARIMA remains to be the best starting point in predicting a sequence from the presented dataset. KAM forecasts help increase the prediction accuracy over a short term, whereas canonical forecasting methods are preferable for medium-long-term predictions. LSTM and error learning avoid the use of self-defined weights for KAM forecasts and forecasting methods. LSTM provides better accuracy for the presented dataset, as demonstrated in the table.

CONCLUSION

The main contributions of this paper are summarised as follows:

- Short-term forecasts (less than 3 months) are clearly improved with KAM inputs. Weighting short-term KAM forecasts and time series models for long-term forecasts provide better results than all the tested models. RS could be an added value metric to decide which time series model to apply for long-term forecasts.
- LSTM is a good choice for handling the dataset, but LSTM is also efficient as error learning on KAM forecasts.

Further investigation is needed to consider forecasts by KAMs, which are more accurate than forecasts by models for a short term (today + 3 months). The combination of the two forecasting methods is relevant when using a set of weights given an important ponderation to the KAMs' forecasts in the short term and an important ponderation to the model in the long term. The sum of the two weights is equal to 1. By changing the ratio of the KAMs' forecasts to the LSTM model, the RMSE could be reduced, thereby increasing the forecast accuracy.

REFERENCES

Abbasimehr, H., Shabani, M., & Yousefi, M. (2020). An optimized model using LSTM network for demand forecasting. *Computers & Industrial Engineering*, *143*, 106435. https://doi.org/10.1016/j.cie.2020.106435

Adhikari, R., & Agrawal, R.K. (2013). *An introductory study on time series modeling and forecasting*. arXiv. https://doi.org/10.48550/arXiv.1302.6613

Bozkurt, O.O., Biricik, G., & Taysi, Z.C. (2017). Artificial neural network and SARIMA based models for power load forecasting in Turkish electricity market. *PLoS ONE*, *12*(4), e0175915. https://doi.org/10.1371/journal.pone.0175915

Büyüksahin, U.C., & Ertekina, S. (2019). Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition. *Neurocomputing*, *361*, 151–163.

Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants. *International Journal of Forecasting*, *37*(2021), 28–43.

Chikkakrishna, N.K., Hardik, C., Deepika, K., & Sparsha, N. (2019). Short-term traffic prediction using SARIMA and Facebook Prophet. *IEEE 16th India Council International Conference (INDICON)*, pp. 1–4. IEEE. https://doi.org/10.1109/INDICON47234.2019.9028937

Haberleitner, H., Meyr, H., & Taudes, A. (2010). Implementation of a demand planning system using advance order information. *International Journal of Production Economics*, *128*(2), 518–526.

Ho, S.L, & Xie, M. (1998). The use of Arima models for reliability forecasting and analysis. *Computers & Industrial Engineering*, *35*(1–2), 213–216.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hui, L., Hong-qi, T., & Yan-fei, L. (2012). Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods. *Applied Energy*, *98*, 415–424.

Hyndman, R.J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice* (second edition). OTexts: Forecasting: Principles and Practice (2nd ed) (otexts.com).

Pereira Da Veiga, C., Pereira Da Veiga, C.R., Cataplan, A., Tortato, U., & Vieira Da Silva, W. (2014). Demand forecasting in food retail: A comparison between the Holt-Winters and ARIMA models. *WSEAS Transactions on Business and Economics*, *11*, 608–614.

Ramosaj, A., & Widmer, M. (2020). How stagger charts can improve forecast accuracy. Foresight: The International Journal of Applied Forecasting, International Institute of Forecasters, *Summer*(58), 34–41.

Rani Samal, K.K., Babu, K.S., Das, S.K, & Acharaya, A. (2019). Time series based air pollution forecasting using SARIMA and Prophet model. *ITCC Proceedings of the 2019 International Conference on Information Technology and Computer*, pp. 80–85. Association for Computing Machinery. https://doi.org/10.1145/3355402.3355417

Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A comparison of ARIMA and LSTM in forecasting time series. *17th IEEE International Conference on Machine Learning and Applications* (pp. 1394–1401). IEEE. https://doi.org/10.1109/ICMLA.2018.00227

Stadtler, H., & Kilger, C. (Eds.). (2002). *Supply chain management and advanced planning* (pp. 71–96). Springer.

Taylor, S.J., & Letham, B. (2017). Forecasting at scale. *The American Statistician*, *72*(1), 37–45. https://doi.org/10.1080/00031305.2017.1380080

Wang, P., Zheng, X., Li, J., & Zhu, B. (2020). Prediction of epidemic trends in COVID-19 with logistic model and machine learning technics. *Chaos, Solitons and Fractals*, *139*(110058), 1–7.

Weytjens, H., Lohmann, E., & Kleinsteuber, M. (2019). Cash flow prediction: MLP and LSTM compared to Arima and Prophet. *Electronic Commerce Research*, *21*, 371–391.

Xie, W., Shen, L., & Zhong, Y. (2017). Two-dimensional aggregate warranty demand forecasting under sales uncertainty. *IISE Transactions*, *49*(5), 553–565. https://doi.org/10.1080/24725854.2016.1263769

Zhao, Z., Chen, W., Wu, X., Chen, P., & Liu, J. (2017). LSTM network: A deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, *11*(2), 68–75.