*Proceedings of the 2025 Winter Simulation Conference*
*E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.*

# A HYBRID APPROACH FOR SHORT-TERM DEMAND FORECASTING: A COMPUTATIONAL STUDY

Raphael Herding[1,2], and Lars Mönch[1,3]

[1]Forschungsinstitut für Telekommunikation and Kooperation (FTK), Dortmund, GERMANY
[2]Westfälische Hochschule, Bocholt, GERMANY
[3]Dept. of Mathematics and Computer Science, University of Hagen, Hagen, GERMANY

## ABSTRACT

We consider a short-term demand forecasting problem for semiconductor supply chains. In addition to observed demand quantities, order entry information is available. We compute a combinational forecast based on an exponential smoothing technique, a long short-term memory network, and the order entry information. The weights for the different forecast sources and parameters for exponential smoothing are computed using a genetic algorithm. Computational experiments based on a rich data set from a semiconductor manufacturer are conducted. The results demonstrate that the best forecast performance is obtained if all the different forecasts are combined.

## 1 INTRODUCTION

Demand planning is an important function in semiconductor supply chains (Mönch et al. 2018a, 2018b). It is complicated since short product life cycles limit the amount of data that can be collected and to which statistical forecast approaches can be applied. Moreover, certain products in a company's portfolio may be substitutes, while other products may be complementary, for instance, a particular CPU and its associated chipset. The long fabrication cycle times observed in semiconductor supply chains resulting in long lead times, often require demand forecasts to be made well in advance of demand realization. This leads to increasing forecast uncertainty. The same chip may be sold in markets with different characteristics resulting in a different set of demand drivers for each market. The prices of most semiconductor devices decrease significantly over their life cycle (Uzsoy et al. 2018).

The results of demand planning are a crucial input for strategic network planning, capacity planning, master planning, and demand fulfillment models in semiconductor supply chains (Mönch et al. 2013). Hence, it is desirable to design, implement, and test advanced forecasting models for semiconductor supply chains.

It seems possible to apply methods based on statistical learning and data analytics for specific demand forecasting problems that are more short-term by nature taking advantage of specific leading indicators (Uzsoy et al. 2018). Some initial steps for short-term demand forecasting in the semiconductor industry are reported by Habla et al. (2007) where order entry information is used as leading indicator. There is some evidence from the literature that using advance order information in forecasting algorithms is beneficial for many forecasting problems (see, for instance, Kekre et al. 1990; Haberleitner et al. 2010). In the present paper, we are interested in generalizing the forecasting method of Habla et al. (2007). We propose an optimization-based forecasting technique relying on forecast and leading indicator combinations. Motivated by the survey paper of Wang et al. (2023), we are especially interested in combining statistical forecasts, forecasts from machine learning (ML) techniques, and leading indicators. We will apply the proposed framework to a dataset from a large semiconductor manufacturer.

The paper is organized as follows. In the next section, we describe the demand forecasting problem at hand, discuss related work, and state research questions. The different forecast approaches including a

hybrid of statistical forecasting and ML are discussed in Section 3. Results of computational experiments are reported in Section 4. Finally, conclusions and future research directions are provided in Section 5.

## 2 PROBLEM SETTING AND ANALYSIS

### 2.1 Forecasting Problem

The forecasting problem to be solved, a generalization of the problem considered by Habla et al. (2007), is based on the following data:

1. We assume that observed demand $DQ_{t-j}, j = 0, \dots, D$ is available where $D > 0$, i.e., we have demand information for $D + 1$ previous periods.
2. Moreover, we have $K \geq 1$ forecast sources labeled by $k = 1, \dots, K$. We assume that we have or alternatively can compute the historical forecast quantities $FC_{k,t-j,}\, j = 0, \dots, D, k = 1, \dots, K$ of the different sources for $D + 1$ previous periods.
3. In addition, we assume that we have $L$ leading indicators labeled by $l = 1, \dots, L$. The quantities $I_{l,t-j}, j = 0, \dots, D, l = 1, \dots, L$ form the historical leading indicator values.
4. Finally, we have the $K$ values for the different forecast sources and the $L$ values for the leading indicator values for the current period $t + 1$, i.e., we know $FC_{k,t+1,}\, k = 1, \dots, K$ and $I_{l,t+1}. l = 1, \dots, L$.

We are interested in computing a 1-step ahead forecast:

$$\widehat{FC}_{t+1} := \sum_{k=1}^{K} \alpha_k FC_{k,t+1} + \sum_{l=1}^{L} \beta_l I_{l,t+1}, \tag{1}$$

for appropriately selected weighting parameters $\alpha_k, k = 1, \dots, K$ and $\beta_l, l = 1, \dots, L$. Note that we do not know $DQ_{t+1}$. An example for the available data is shown in Table 1.

Table 1: Exampled data for the forecasting problem at hand.

| **Period** | $FC_1$ | …. | $FC_K$ | $I_1$ | … | $I_L$ | **Forecast** $\widehat{FC}$ | $DQ$ |
|---|---|---|---|---|---|---|---|---|
| t-D | 100 | 200 | 150 | 20 | 90 | 60 | 120 | 130 |
| … | 120 | 210 | 130 | 20 | 100 | 70 | 100 | 90 |
| t-1 | 55 | 180 | 100 | 40 | 80 | 70 | 90 | 120 |
| t | 90 | 90 | 80 | 20 | 70 | 90 | 105 | 100 |
| **t+1** | **25** | **100** | **90** | **10** | **90** | **60** | **to be calculated** | **-** |

### 2.2 Discussion of Related Work and Research Questions

Methods for short-term demand forecasting are reviewed by Elias et al. (2006). However, methods based on a combination of leading indicator values such as advance order information with more conventional forecasting methods are not discussed in this paper. Habla et al (2007) propose such a method for a short-term demand forecasting problem arising in semiconductor manufacturing. Exponential smoothing techniques are combined with order entry information which can be considered as advance demand information. The two forecast sources are combined by appropriate weight values which are chosen together with parameters for the exponential smoothing scheme by nonlinear optimization. Only a 1-step-ahead forecast is computed. Habla et al. (2008) extends this approach towards a h-step-ahead forecasting method. In both papers, a commercial nonlinear solver is used.

A forecasting algorithm that takes into account advance order information is proposed by Haberleitner et al. (2010) based on earlier work by Kekre et al. (1990). The approach is imbedded into a demand planning system for a make-to-order manufacturer. The system is also able to select the optimal forecasting model type and the level of integration of advance demand information based on the patterns of a particular time series. The approaches by Habla et al. (2007) and Habla et al. (2008) can be considered as special forecast combination approaches. Based on the survey paper Wang et al. (2023) it seems beneficial to incorporate ML methods in forecast combination approaches.

Therefore, the following two research questions (RQs) are addressed in the present paper:

**RQ1:** Can we improve the results obtained by Habla et al (2007) by including state-of-the-art ML approaches as additional forecast sources in forecast combination approaches?
**RQ2:** Is it possible to replace the commercial nonlinear solver by a genetic algorithm?

## 3 SOLUTION APPROACHES

### 3.1 Forecasting Framework

The forecast quantities $FC_{k,t-j,}\, j = 0, \dots, D, k = 1, \dots, K$ are either available or can be computed based on the observed demand quantities $DQ_{t-j}, j = 0, \dots, D$. In the latter case, we may have to choose appropriate parameters for a given forecast source. Let us assume that the $n_k \geq 0$ parameters $(\gamma_{k1}, \dots, \gamma_{kn_k})$ must be selected for forecast source $k$. Therefore, we can see the forecast quantities as a function of the parameters $\gamma_{ki}$, i.e., we have $FC_{k,t-D+i}(\gamma_{k1}, \dots, \gamma_{kn_k})$. Of course, the value of $FC_{k,t-D+i}$ typically depends on the observed demand quantities. We proposed the following nonlinear optimization formulation for computing appropriate parameter values for the forecast value (1) for period $t + 1$. It is based on the following sets and indices, parameters, and decision variables.

Sets and indices:

| | |
|---|---|
| $k = 1, \dots, K$: | index of the $k$th forecast source |
| $l = 1, \dots, L$: | index of the $l$th leading indicator |
| $s_k = 1, \dots, S_k$: | index of the $s_k$th constraint for the parameters of the forecast approach for $k$th forecast source |
| $t - \mathrm{D}, \dots, t$: | periods in the past |
| $t + 1$: | current period |

Parameters:

| | |
|---|---|
| $DQ_{t-j}$: | observed demand for $j = 0, \dots, D$ |
| $FC_{k,t-j}$ : | historical forecast for period $t - j, j = 0, \dots, D$, and forecast source $k = 1, \dots, K$ |
| $I_{l,t-j}$: | historical leading indicator value for period $t - j, j = 0, \dots, D$, and leading indicator $l = 1, \dots, L$ |

Decision variables:

| | |
|---|---|
| $\alpha_k$: | weight for forecast value of the $k$th forecast source |
| $\beta_l$: | weight for the value of the $l$th leading indicator |
| $\gamma_{kj}$: | $j$th parameter to be chosen for the forecast approach of the $k$th forecast source. |

The optimization formulation is stated as follows:

$$\min \sum_{i=0}^{D} \left\{ \sum_{k=1}^{K} \alpha_k FC_{k,t-D+i}(\gamma_{k1}, \dots, \gamma_{kn_k}) + \sum_{l=1}^{L} \beta_l I_{l,t-D+i} - DQ_{t-D+i} \right\}^2 \quad (2)$$

subject to

$$\sum_{k=1}^{K} \alpha_k + \sum_{l=1}^{L} \beta_l \leq 1 \tag{3}$$

$$f_{ks_k}(\gamma_{k1}, \dots, \gamma_{kn_k}) = 0, k = 1, \dots, K, s_k = 1, \dots, S_k \tag{4}$$

$$\alpha_k \geq 0\ , \beta_l \geq 0, \gamma_{ki_k} \in D_{i_k}, k = 1, \dots, K, l = 1, \dots, L, i_k = 1, \dots, n_{k.} \tag{5}$$

The nonlinear objective function to be minimized, i.e. $F(\alpha_1, \dots, \alpha_K, \beta_1, \dots, \beta_L, \gamma_{11}, \dots, \gamma_{Kn_K}) := \sum_{i=0}^{D}\left(\sum_{k=1}^{K} \alpha_k FC_{k,t-D+i}(\gamma_{k1}, \dots, \gamma_{kn_k}) + \sum_{l=1}^{L} \beta_l I_{l,t-D+i} - DQ_{t-D+i}\right)^2$, can be found in (2). It is the sum of the difference of the weighted combined forecast and leading indicator value for period $t - D + i$ and the corresponding observed demand quantity. (3) is a constraint for the weighting parameters for different forecast sources and leading indicators. Constraints for possible parameters of the different forecast approaches are modeled by constraint set (4). Here, $f_{ks_k}: \mathbb{R}^{n_k} \to \mathbb{R}$ are smooth functions. The non-negativity constraints for the decision variables $\alpha_k$ and $\beta_l$ and the domains $D_{i_k}$ for the decision variables $\gamma_{ki_k}$ are expressed by (5).

### 3.2 Exponential Smoothing

Exponential smoothing is a widely used statistical forecast approach (Elias et al. 2006; Bisgaard and Kulahci 2011) to compute forecasts based on observed demand. For the sake of simplicity, we suppress the index of the forecast source in the following description. A possible forecast source is given by an exponential smoothing approach with two parameters to be selected. We start from the expression

$$(1 - \gamma_2) \cdot \gamma_1 + \gamma_2 \cdot DQ_{t-D}, \tag{6}$$

where $\gamma_1$ is the initial value for the first period in the past and $\gamma_2 \in [0,1]$ is the smoothing parameter of the scheme. After repeating this recursive relationship $i$ times, we obtain the following explicit expression:

$$FK_{t-D+i}(\gamma_1, \gamma_2) := \gamma_1 \cdot (1 - \gamma_2)^{i+1} + \gamma_2 \cdot \sum_{j=0}^{i} (1 - \gamma_2)^{i-j} \cdot DQ_{t-D+j},\ i = 0, \dots, D. \tag{7}$$

### 3.3 Long Short-term Memory-based Approach

A long short-term memory (LSTM) network is a specialized type of recurrent neural network (RNN) trained by using backpropagation through time (BPTT). It is specifically designed to overcome the vanishing gradient problem which often hinders the training of traditional RNNs (Bengio et al. 1994). For a more detailed explanation of how an LSTM model is used for time series analysis, we refer to Song et al. (2020).

LSTM networks are well-suited for regression tasks, particularly when the input data has a sequential structure because they are designed to capture dependencies and patterns over time. Their internal memory cells and gating mechanisms allow them to retain relevant information from previous time steps, which is crucial for predicting continuous values that depend on a historical context. This makes LSTMs highly effective for tasks such as time series forecasting, where the current prediction is influenced by past trends/patterns and values. The architecture of the used LSTM model is shown in Figure 1.

The number of influencing factors of the demand forecasting prediction determines the dimension of the input layer. In the present situation, the product and the observed, i.e. historical, demand are used as input data which results in a two-dimensional input layer. The LSTM layer is embedded to memorize and extract containing information from input data, where the Adam optimizer (Kingma and Ba 2015) is used to update the weights and bias. The mean squared error is used as fitness function. A fully connected layer

provides the final output. Since we are only interested in predicting a single forecast value, we only require one dimension in the output layer.

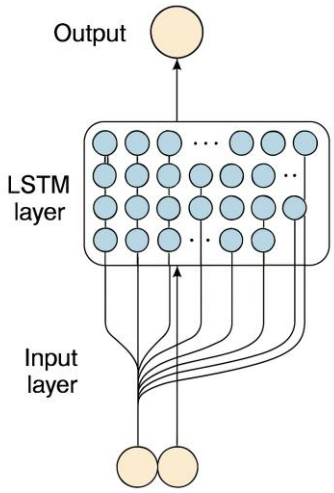


Figure 1: LSTM network architecture.

The number of neurons in the LSTM layer as well as the number of previously observed demand points that influence the prediction of the next demand forecast, have to be determined in the process of constructing the LSTM model. For determining suitable values for these two quantities, we use a simple grid search during the training of the LSTM model.

The proposed LSTM model is used to search for the forecast of a certain product where the observed demand as well as the product of the demand are used as input and the 1-step-ahead forecast of the next period as output. Therefore, if the product is given, the forecast can be predicted by the model. The following three steps must be performed to use the LSTM approach for the short-term demand forecasting problem at hand.

**Step 1: Data-Preprocessing**

Since we do not have any missing data in the data set, we can immediately normalize the data to the interval [0,1] to avoid model parameters being dominated by a large or small data range since the LSTM network is sensitive to the scale of the input data. The normalization function is given by

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}}, \tag{8}$$

where $x_{new}$ represents the normalized data points inserted in the training process, while $x_{old}$, $x_{min}$, and $x_{max}$ are the original value of the sample data and values for lower and upper bounds, respectively. Based on the characteristics of demand forecasting and the LSTM network, the model can be described as

$$FC(p, t+1) = FC(p, DQ_p(t), DQ_p(t-1), \dots, DQ_p(t-n)) , \tag{9}$$

where $FC(p, t+1)$ is the forecast value of product $p$ for period $t+1$. The quantities $DQ_p(t-k), k = 0, \dots, n$ describes the previously observed demand values belonging to the product $p$, and $n$ is the used time

window size (time lag). Typically, the more recent the observed demand is, the more it influences the forecast. To model this behavior the time lag is used to limit the number of past observed demand values.

Expression (10) shows an example input matrix where $n = 2$ is used for the sake of simplicity:

$$\overbrace{\begin{pmatrix} p & DQ_p(1) & DQ_p(2) & DQ_p(3) \\ p & DQ_p(2) & DQ_p(3) & DQ_p(4) \\ \dots & \dots & \dots & \dots \\ p & DQ_p(t-2) & DQ_p(t-1) & DQ_p(t) \end{pmatrix}}^{\text{Input}} \overbrace{\begin{pmatrix} FC(p,4) \\ FC(p,5) \\ \dots \\ FC(p,t+1) \end{pmatrix}}^{\text{Output}}. \quad (10)$$

Note that for each product $p \in P$ a separate matrix of the same structure exists.

**Step 2: Training of the LSTM model**

The reframed and normalized data is split into two parts. In typical regression tasks, this is commonly carried out using cross-validation. However, in time series analysis, preserving the order of the time series is crucial. Therefore, a more appropriate approach involves splitting the dataset chronologically into training and testing subsets is used.

A simple strategy is to allocate a fixed percentage of the earliest observations for training and reserve the remaining portion for testing purposes. In our experiments, the first 80% is taken as a training set and the last 20% as a testing set.

The LSTM model is fed by the training set during which appropriate values of the time window size and number of neurons of the LSTM layer are determined by a grid search. We use $\#neurons \in \{1, \dots, 50\}$, and for the time window size we use $n \in \{1, \dots, 28\}$ as a grid for the LSTM layer. It is useful to align the size of the time window with the number of data points of the time series.

**Step 3: Forecasting with the LSTM model**

To forecast demand during the inference phase, the testing set obtained in Step 2 contains the future forecast information. In the present paper, we are only interested in a 1-step-ahead forecast. An example input matrix for $p = p1$ and $n = 2$ is shown next:

$$\overbrace{(p1 \quad DQ(t-2) \quad DQ(t-1) \quad DQ(t))}^{\text{Input}} \overbrace{(FC(p1, t+1))}^{\text{Ouput}}. \quad (11)$$

Before we can obtain the final forecast, we have to denormalize the output data $FC(p1, t+1)$. For measuring the performance, the denormalized forecast can be compared with the observed demand $DQ(t+1)$ extracted from the testing set of Step 2.

### 3.4 Order Entry as Leading Indicator and Simple Reference Approaches

Firm orders $FO$ are given for the periods $t - D, \dots, t+1$. Firm orders for period $t+1$ are determined in period $t+1-k$ by sum up all the order entry quantities with a planned completion time in period $t+1$ that are arrived before or within period $t+1-k$. The quantity $k$ is given by lead time considerations. In our experiments we use $k = 1$ period.

The first reference approach is the firm order scheme. We simply consider the quantity of the firm orders for the current period as forecast. The procedure to compute a forecast for period $t$ can then be described as follows:

$$\widehat{FC}_{t+1} := FO_{t+1}, \tag{12}$$

where $FO_{t+1}$ denotes the order entry for period $t+1$. This procedure is clearly memoryless because it does not consider any observed demand information. We denote this approach by order entry (OE).

The second approach is based on the idea that the ratio of the demand and the firm order quantities of the current period is equal to the same ratio for the corresponding quantities of the previous period for a given product. We obtain:

$$\widehat{FC}_{t+1} := \begin{cases} \frac{FO_{t+1}}{FO_t} DQ_t, \text{ if } FO_t > 0 \\ FO_{t+1}, \text{otherwise} \end{cases}, \tag{13}$$

where $DQ_t$ denotes the observed demand value for a certain product in period $t$. We refer to this method as book-to-bill (BB) procedure (cf. Habla et al. 2007).

### 3.5 Solution Aspects

We use a genetic algorithm (GA) to implement the optimization-based forecasting framework described in Subsection 3.1. GAs have been used extensively for solving hard and large-scale optimization problems, especially when the objective function is nonlinear (Gallagher and Sambridge 1994).

Since the objective function (2) of the optimization formulation (2)-(5) belonging to the forecasting framework is nonlinear, a GA seems a promising solution. Such an approach avoids the usage of commercial nonlinear solvers based on methods such as, for instance, the generalized reduced gradient (GRG2) algorithm (Lasdon et al. 1974; Lasden and Waren 1978) or trust region methods (Conn et al. 2000).

A GA is a population-based approach (Michalewicz 1996). A single iteration is called a generation. The individuals of the new generation are obtained from the individuals of the previous one by applying genetic operators such as crossover and mutation. We use the one-point crossover for reproduction as well as the swap mutator as mutation procedure.

We are interested in optimizing the set of decision variables $\alpha_k, \beta_l, \gamma_{kj}$ from the forecasting framework. Therefore, an array with real numbers as entries is used as representation in the GA. The array is divided into three sections where each section represents one of the three sets of decision variables. Figure 2 shows the structure of the array for encoding the optimization problem (2)-(5). The two red lines are used as visual separators of the three sections. Clearly, we see from (5) that the length of the encoding array is $K + L + \sum_{k=1}^{K} n_k$.


Figure 2: Representation used in the GA.

We need to evaluate the fitness of each of individual, i.e. chromosome, of the current population where only the fittest individuals are selected for the next population. We use the objective function (2) of the forecasting framework as fitness function. Since the GA is not capable to ensure the feasibility of the constrains used in the optimization formulation (2)-(5), we have to reformulate the fitness function. We use

$$F(\alpha_1, \dots, \alpha_K, \beta_1, \dots, \beta_L, \gamma_{11}, \dots, \gamma_{Kn_K}) := \sum_{i=0}^{D} \Big( \sum_{k=1}^{K} \alpha_k FC_{k,t-D+i}(\gamma_{k1}, \dots, \gamma_{kn_k}) + \sum_{l=1}^{L} \beta_l I_{l,t-D+i} - DQ_{t-D+i} \Big)^2 + \xi \cdot \Big\{ \Big( \sum_{k=1}^{K} \alpha_k + \sum_{l=1}^{L} \beta_l - 1 \Big)^+ \Big\}^2, \tag{14}$$

where we introduce the penalty value $\xi \geq 0$ to enforce that the constraint $\sum_{k=1}^{K} \alpha_k + \sum_{l=1}^{L} \beta_l \leq 1$ is fulfilled. Here, we abbreviate $x^+ := \max(x, 0)$. Additional penalty values may be required to ensure that the constraint set (4) is fulfilled.

## 4 COMPUTATIONAL EXPERIMENTS

### 4.1 Design of Experiments

To evaluate the performance of the different forecasting schemes, real-world historical data of a large semiconductor company is applied. The data contains demand and firm order information for 1189 products from 28 months. Based on the different forecast approaches discussed in Section 3, we are interested in investigating the following five scenarios:

1. **LSTM:** In this scenario, we only use the LSTM approach as forecast source.
2. **ES-OE-GA:** This scenario consists of the exponential smoothing forecast source and the order entry as leading indicator value. We combine them using the GA as described in Subsection 3.5.
3. **OE-LSTM-GA:** The third scenario again uses the order entry as leading indicator value as well as the LSTM forecast source. We combine them by using the GA.
4. **LSTM-ES-GA:** In this scenario, we apply the exponential smoothing forecast source and the LSTM forecast source. We again combine them by using the GA.
5. **LSTM-ES-OE-GA:** In this last scenario, we use the two forecast sources and the leading indicator. We again combine them using the GA.

Note that the **ES-OE-GA** is exactly the approach proposed by Habla et al. (2007), although a commercial nonlinear solver based on the GRG2 algorithm is applied there.

We use the symmetric mean absolute percentage error (SMAPE) as performance measure. The SMAPE value is defined as

$$SMAPE := 100\% \cdot \left(1 - \frac{\sum_{t=1}^{T} |\widehat{FC}_t - DQ_t|}{\sum_{t=1}^{T} (\widehat{FC}_t + DQ_t)}\right), \tag{15}$$

where $\widehat{FC}_t$ is the total forecast for period $t$ and $DQ_t$ defines the corresponding total observed demand for period $t$ of the planning horizon $T$. We use equation (1) for $D = 8$ periods to calculate $\widehat{FC}_t$. The SMAPE produces performance values between 0 (worst) and 100 (best). Periods with large demand have a larger influence on the accuracy than periods with small demand. The SMAPE is widely used to measure the performance in demand forecasting.

### 4.2 Implementation Issues and Parameter Setting

The LSTM network is implemented on the basis of Keras (cf. Keras 2025), a deep learning library using Tensorflow as backend. The entire workflow is coded using the Python 3.11 programming language and is executed on a PC with Intel ® Core™ i7-14700 3.60GHz CPU and 32 GB of RAM. The forecasting framework is coded in the Python programming language. The experiments are carried out on a computer with an Intel ® Core™ i7-14700 3.60GHz CPU and 32 GB of RAM. We use the GALib framework (Wall 1999) to implement the GA.

For the GA, we apply 250 generations, each consisting of 30 individuals. The crossover probability is set to 0.9 while the mutation probability is chosen as 0.1 A replacement rate of 0.55 is used in the GA with overlapping populations. These settings are determined based on preliminary experiments in combination with a trial and error strategy. We use a very large positive number for the penalty value $\xi$.

### 4.3 Results

We show the results of the performed computational experiments in Table 1. Note that the entries in this table are percentage values due to the definition of the SMAPE measure. The best performing scenario is marked bold.

Table 1: Computational results.

| Scenario | SMAPE |
|---|---|
| LSTM | 64.19% |
| ES-OE-GA | 56.37% |
| OE-LSTM-GA | 76.84% |
| BB | 52.79% |
| OE | 49.31% |
| LSTM-ES-GA | 73.60% |
| **LSTM-ES-OE-GA** | **86.28%** |

The results found in Table 1 are visualized in Figure 3.

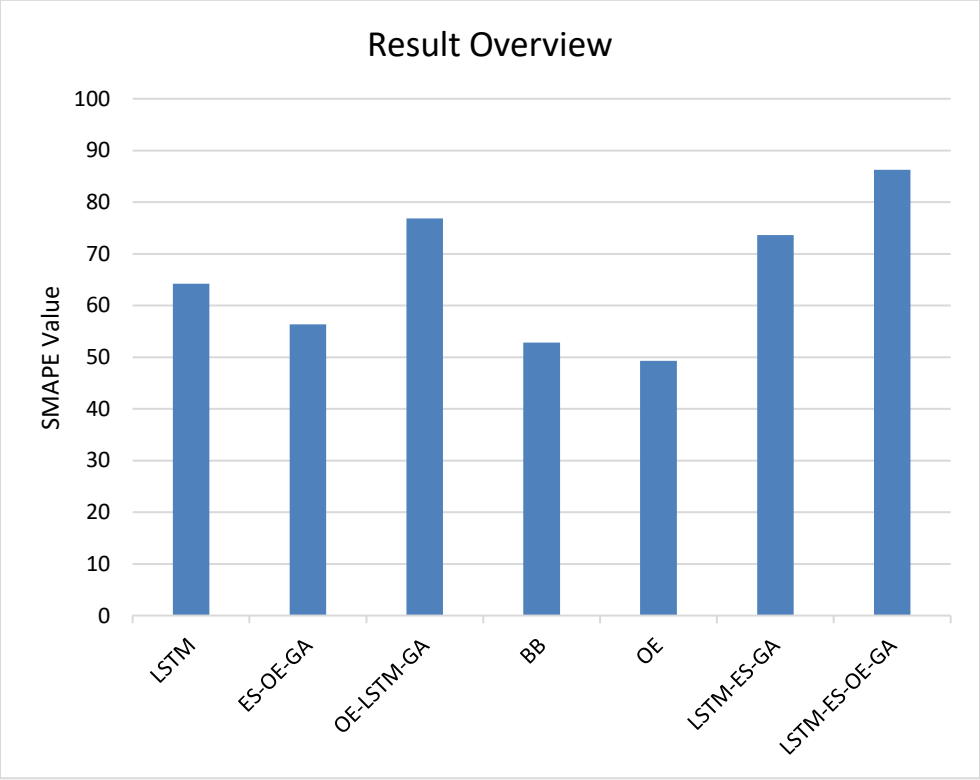


Figure 3: SMAPE values (in %) depending on the scenario.

We can see that the combination of the different forecast sources and the order entry, i.e. the scenario LSTM-ES-OE-GA, leads to the largest SMAPE value. Using only exponential smoothing and the order entry results in a fairly small SMAPE value. This approach only outperforms the two simple reference schemes. We also observe that the LSTM approach alone leads to a fairly large SMAPE value. This is

reasonable since the LSTM approach works well for recognizing patterns in time series. This assumption is also supported by the fact that all scenarios where the LSTM is involved lead to fairly large SMAPE values.

We also observe that the parameter optimization in general helps to increase the SMAPE values. Whenever the GA is applied, we find large SMAPE values. This is reasonable since the GA optimizes the algorithm parameters of the exponential smoothing as well as the weights of the different forecast sources and the involved indicator value.

On the one hand, including the hyperparameters of the LSTM approach into the GA-based optimization does not lead to any further SMAPE improvements. On the other hand, the overall computing time of the optimization increases dramatically since the GA has to evaluate thousands of potential solutions where for each solution a complete retaining of the LSTM has to be performed. It turns out that a proactive upfront training of the LSTM approach including a grid search for hyperparameter tuning as described in Subsection 3.3 leads to similar SMAPE values while the computing time is much smaller.

## 5 CONCLUSIONS AND FUTURE WORK

We presented a framework for a short-term forecasting problem in semiconductor manufacturing based on forecast combinations including statistical forecast methods and ML approaches. A GA was used to select parameters for the forecast sources and for weighting the different forecast sources and leading indicators. A hybrid approach including exponential smoothing, LSTM, and order entry information performed best among the different possible combinations of forecast sources and leading indicators.

There are several directions for future research. First of all, we believe that it is possible to extend the optimization model in such a way that h-step-ahead forecasting is possible. Time series with trend and seasonality should be considered too. This can be carried out, for instance, by replacing the exponential smoothing approach used in in the present paper by the Holt-Winters method. Moreover, it would be interesting to consider larger time series, in this paper, we use only 28 data points per time series. Another direction is considering more leading indicator values. Appropriate examples can be found, for instance, in Elias et al. (2008). Finally, we are interested in replacing the grid search approach to determine hyperparameters of the LSTM model by a metaheuristic approach (cf., for instance, Song et al. 2020).

## ACKNOWLEDGMENTS

The authors would like to thank Hans Ehm, Infineon Technologies, for fruitful discussions on demand forecasting in semiconductor manufacturing. They were supported by the SC4EU project which receives funding from the CHIPS JU under grant agreement No. 101139949.

## REFERENCES

Bengio, Y., P. Simard, and P. Frasconi. 1994. “Learning Long-term Dependencies with Gradient Descent is Difficult”. *IEEE Transactions on Neural Networks* 5(2):157–166.

Bisgaard, S., and M. Kulahci. 2011. *Time Series Analysis and Forecasting by Example*. Hoboken: Wiley.

Conn, A. R., N. I. M. Gould, and P. L. Toint. 2000. *Trust-region Methods*. Philadelphia: SIAM.

Elias, R., D. C. Montgomery, and M. Kulahci. 2006. “An Overview of Short-term Statistical Forecasting Methods”. *International Journal of Management Science and Engineering Management* 1(1):17-36.

Elias, R., D. C. Montgomery, S. A. Low, and M. Kulahci. 2008. “Demand Signal Modelling: A Short-range Panel Forecasting Algorithm for Semiconductor Firm Device-level Demand”. *European Journal of Industrial Engineering* 2(3):253-278.

Gallagher, K., and M. Sambridge. 1994. “Genetic Algorithms: A Powerful Tool for Large-scale Nonlinear Optimization Problems”. *Computers & Geosciences* 20(7/8): 1229-1236.

Haberleitner, H., H. Meyr, and A. Taudes. 2010. “Implementation of a Demand Planning System Using Advance Order Information”. *International Journal of Production Economics* 128:518-526.

Habla. C., R. Drießel, L. Mönch, T. Ponsignon, and H. Ehm. 2007. “A Short-Term Forecast Method for Demand Quantities in Semiconductor Manufacturing”. In *Proceedings of the 2007 IEEE Conference on Automation Science and Engineering*, September 22nd-25th, Scottsdale, AZ, USA, 94-99.

Habla, C., L. Mönch, T. Ponsignon, and H. Ehm. 2008. "Optimierungsbasierte Verfahren für die kurzfristige Bedarfsvorhersage in der Hochtechnologiebranche". *Proceedings Teiltagung "Intelligente Systeme zur Entscheidungsunterstützung" der Multikonferenz Wirtschaftsinformatik*, February 26th-28th, München, Germany, 113-127.

Kekre, S., T. E. Morton, and T. L. Smunt. 1990. "Forecasting Using Partially Known Demand". *Journal of Forecasting* 6:115-125.

Keras. 2025. Keras Deep Learning API. https://keras.io/, accessed 12th May 2025.

Kingma, D. P., and J. L. Ba. 2015. "Adam: A Method for Stochastic Optimization". In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015),* May 5th-7th, San Diego, CA, USA, poster.

Lasdon, L. S., R. L. Fox, and M. W. Ratner. 1974. "Nonlinear Optimization Using the Generalized Reduced Gradient Method". *Revue Française d'Automatique, Informatique, Recherche Opérationnelle* 8(3):73-103.

Lasdon, L. S., and A. D. Waren, 1978. "Generalized Reduced Gradient Software for Linearly and Nonlinearly Constrained Problems". In *Design and Implementation of Optimization Software*, edited by H. D. Greenberg, 225-262. Alphen van de Ryn: Sijthooff & Noordhof.

Michalewicz, Z. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd ed., Berlin: Springer.

Mönch, L., J. W. Fowler, and S. J. Mason. 2013. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*. New York: Springer.

Mönch, L., R. Uzsoy, and J. W. Fowler. 2018a. "A Survey of Semiconductor Supply Chain Models Part I: Semiconductor Supply Chains, Strategic Network Design, and Supply Chain Simulation". *International Journal of Production Research* 56(13):4524-4545.

Mönch, L., R. Uzsoy, and J. W. Fowler. 2018b. "A Survey of Semiconductor Supply Chain Models Part III: Master Planning, Production Planning, and Demand Fulfillment". *International Journal of Production Research* 56(13):4565-4584.

Song, X., Y. Liu, L. Xue, J. Wang, J. Zhang, J. Wang, L. Jiang, and Z. Cheng. 2020. "Time-series Well Performance Prediction Based on Long Short-Term Memory (LSTM) Neural Network Model". *Journal of Petroleum Science and Engineering* 186:106682.

Uzsoy, R., J. W. Fowler, and L. Mönch. 2018. "A Survey of Semiconductor Supply Chain Models Part II: Demand Planning, Inventory Planning, and Capacity Planning". *International Journal of Production Research* 56(13):4546-4564.

Wang, X., R. J. Hindman, F. Li, and Y. Kang. 2023. "Forecast Combinations: An Over 50-year Review". *Journal of Forecasting* 39:1518-1547.

Wall, M. 1999. Galib: A C++ Library of Genetic Algorithms Components. http://lancet.mit.edu/ga/, accessed 12th May 2025.

## AUTHOR BIOGRAPHIES

**RAPHAEL HERDING** is a Professor for Software Engineering at the Westfälische Hochschule Bocholt. He received a master's degree in applied computer science and a Ph.D. in computer science from the University of Hagen, Germany. His current research interests are in multi-agent systems, cloud computing, and supply chain management, especially for the semiconductor industry. His email address is raphael.herding@w-hs.de.

**LARS MÖNCH** is full professor of Computer Science at the Department of Mathematics and Computer Science, University of Hagen where he heads the Chair of Enterprise-wide Software Systems. He holds M.S. and Ph.D. degrees in Mathematics from the University of Göttingen, Germany. After his Ph.D., he obtained a habilitation degree in Information Systems from Technical University of Ilmenau, Germany. His research and teaching interests are in information systems for production and logistics, simulation, scheduling, and production planning. His email address is Lars.Moench@fernuni-hagen.de. His website is https://www.fernuni-hagen.de/ess/team/lars.moench.shtml.