applied sciences

MDPI

Article

Framework of 2D KDE and LSTM-Based Forecasting for Cost-Effective Inventory Management in Smart Manufacturing

Myungsoo Kim, Jaehyeong Lee, Chaegyu Lee and Jongpil Jeong *

Department of Smart Factory Convergence, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon 16419, Korea; sioals@skku.edu (M.K.); objective@skku.edu (J.L.); leechgyu@skku.edu (C.L.)
* Correspondence: jpjeong@skku.edu; Tel.: +82-031-299-4260

Abstract: Over the last decade, the development of machine-learning models has enabled the design of sophisticated regression models. For this reason, studies have been conducted to design predictive models using machine learning in various industries. In particular, in terms of inventory management, forecasting models predict historical market demand, predict future demand, and enable systematic inventory management. However, in most small and medium enterprise (SMEs), there is no systematic management of data, and because of the lack of data and the volatility of random data, it is difficult for prediction models to work well. Since the predictive model is a core function derived from the management of the enterprise's inventory data, the poor performance of the model causes the company's inventory data-management system to be degraded. Companies that have poor inventory data because of this vicious cycle will continue to have difficulty introducing data-management systems. In this paper, we propose a framework that can reliably predict the inventory data of a firm by modeling the volatility of a firm stochastically. The framework makes the prediction using the point prediction model by means of LSTM(Long Short Term Memory), the 2D kernel density function, and the prediction result reflecting inventory-management cost. Through various experiments, the necessity of interval prediction in demand prediction and the validity of the cost-effective prediction model through the readjustment function were shown.

Keywords: LSTM; 2D kernel density estimation; forecasting framework; smart manufacturing

check for updates

Citation: Kim , M.; Lee, J.; Lee, C.; Jeong, J. Framework of 2D KDE and LSTM-Based Forecasting for Cost-Effective Inventory Management in Smart Manufacturing. *Appl. Sci.* **2022**, *12*, 2380. https://doi.org/10.3390/app12052380

Academic Editors: Ahmad Barari, Alexandre Carvalho and Marcos de Sales Guerra Tsuzuki

Received: 19 December 2021
Accepted: 17 February 2022
Published: 24 February 2022

1. Introduction

Since the global financial crisis, the global manufacturing industry has faced growth limitations due to the long-term economic recession and rising labor and raw material costs. In developed countries, the manufacturing labor population decreases due to falling fertility rates and avoidance of manufacturing, and the industrial structure changes to a service-oriented economic structure due to low-wage manufacturing avoidance and service preference. In this situation, smart manufacturing technology is emerging as a new way to overcome the growth limitations of the manufacturing industry and innovate. Recently, as product functions, quality, performance, and environmental demands required by major industries increase, the technical and economic burden for the production of competitive products is increasing. In addition, manufacturing powerhouses are promoting manufacturing revival policies by establishing plans for the 4th Industrial Revolution through the convergence of manufacturing ICT (Information and Communications Technology). The production method is changing from partial automation to automatic production using communication and simulation between machines, agile production, customized production, and mass production of multiple varieties based on the Internet of Things, big data, cloud computing, and robots. Not only is the process optimization achieved by connecting the inside of the factory and the outside of the factory through a network, but it is also promoting productivity improvement and cost reduction by connecting manufacturing and real-time.

Smart Manufacturing is a concept and technology for achieving enterprise-wide optimization by connecting both industrial devices and production processes to a network through the convergence of ICT and manufacturing industries. This provides process optimization, efficiency, and flexibility in production facilities for problems such as production, process control and repair of existing factories, and workplace safety. With these advances in technology, mass customization to satisfy the detailed requirements of various customers basically requires on-demand factory technology. Mass customization refers to new production and marketing methods that lower costs through mass production of customized products and services, and requires multiple capabilities, but the most important factor is managing product processes and supply chains to respond to changes in customer needs.

The development of machine learning has had a major effect across many industries. Among them, Smart Manufacturing has an environment where various machine-learning solutions such as a massive amount of data are gathered by means of various sensors and investment is concentrated. In fact, various solutions have already been developed and applied to energy efficiency [1–5], equipment-condition monitoring [6–11], and product-defect detection [12–14] in Smart Manufacturing construction. In particular, the application of machine-learning algorithms to inventory management can help to effectively reduce costs for many companies. Until now, we have been predicting market demand based on human intuition or on poorly performing data-prediction techniques. On the other hand, by means of machine learning, it is possible to predict future outcomes effectively by using accumulated data. In fact, research is currently under way to predict market demand. In the area where data collection such as the electric-power market is organized, various demand-forecasting systems based on machine learning have already been introduced to the business solution system. For SMEs that small produce a large variety of products, a systematic inventory-management system is needed to minimize losses caused by inventories and provide a smooth supply of products. However, most SMEs do not have systematic inventory management, and many companies manage inventories based on handwritten inventories created by judging orders for products by intuition. However, relying on intuition is a big risk for the enterprise and can lead to huge inventory-management costs. Machine-learning-based demand-forecasting models, which have already proven effective in many cases, are expected to be introduced in SMEs to help reduce their risk significantly. Several studies have already been done on the design of predictive models. However, there are few studies that can be applied to actual SMEs. One of the most important points in the design of the predictive model is that each company must have enough historical data. However, in most SMEs, systematic data management is not under way, and the needed data is either missing or has a large error in the recorded value. Poor data collected without a systematic data-management system can adversely affect the performance of the predictive model and render the predicted results meaningless. Therefore, a prediction system that can indicate the reliability of the data together with the output of the predicted results by means of machine learning is needed for the SMEs. In this paper, we first design a demand-prediction model based on LSTM (Long-Short-Term Memory). Next, we quantify the variability of the data as a function of the 2D kernel mill, and the error range is visualized by calculating the maximum and minimum errors of the prediction result according to the reliability. Finally, we propose a demand forecasting model that minimizes the cost by adjusting the predicted amount to reflect the cost incurred if there is insufficient prediction compared to the actual value and the excess cost. Briefly, we examine the usefulness of 2D KDE-based interval prediction techniques with various advantages over cost-effective and general point prediction-based demand forecasting using inventory data from volatile and difficult to handle SMEs. The main contributions of our paper are as follows.

- Mix of point and interval presentations based on LSTM and 2D kernel density function's algorithmic utility test and application through cost-effective functions;
- The composition of a demand forecasting framework and utility test that can cope with various rapidly changing situations using actual SMEs in Korea, forwarding data for five years.

This paper is organized as follows. In Section 2, we introduce a previous study on demand forecasting by means of machine learning. Then, we introduce the LSTM technique and the kernel-density estimation technique, which are mainly used in the proposed algorithm. Section 3 presents the overall framework of the demand forecasting proposed in this paper and the algorithm for each part. In Section 4, we examine the effect of the proposed algorithm on randomly generated data and actual demand data of SMEs. Finally, Section 5 summarizes the results of the study and concludes by introducing some future research directions.

2. Related Work

2.1. Machine-Learning Based Demand Forecasting

As companies continue to develop, the level of demand for accuracy in predicting product demand only increases. There are factors that further complicate the relationship between them in an environment where the size of the company expands and consumer demand changes due to changes in personal preferences. Therefore, we need to choose a more suitable prediction methodology to solve these problems. The inventory demand prediction problem is the time series prediction problem. Several attempts have been made to predict market demand based on machine learning. In general, research on market-demand forecasts is actively under way in the data-rich power market. Data-based market-demand forecasting requires enormous amounts of data. The power market, which automatically collects data based on sensors, can make various attempts for prediction. In [15,16], the authors applied the deep-learning model to the prediction of the power load and showed that it works quite well. J. Toubeau et al. [17] predicted multivariate power-market scheduling based on deep running. Predictions of not only demand but also solar-power generation were done in various studies, and it was possible to find significant prediction results for complex natural phenomena [18–20]. There have been many attempts to predict demand by means of machine learning for other items besides the electric-power market. Ref. [21] proposed a technique for accurately predicting rice demand using RNN and LSTM techniques. In addition, because big data is collected in various areas, research on prediction models using machine-learning techniques is being done for predicting demand for water resources in urban areas [22], electric-charge demand for electric vehicles [23], and crude-oil demand [24]. In order to predict the demand of fashion companies, the development of algorithms from a data-driven perspective was studied by using machine learning techniques and identifying important predictors. Many of these studies show that demand forecasting is very difficult.

2.2. LSTM

In the last decade, because of the development of computing capabilities and the increasing availability of analytical data, various data-analysis techniques have developed rapidly. In particular, models that predict future outcomes based on historical data have been studied and applied in various industries. Predictive models by means of neural networks have shown excellent performance in predicting complex nonlinear time-series data [25], especially when compared to sophisticated physical models [26]. The Recurrent Neural Network (RNN) model has a recursive structure that can transfer the information of the last state to the current state. It works well in time-series data analysis. As shown in the Figure 1, the weights of each node are updated in the course of learning for the hidden layer existing between the input layer and the output layer. The traditional RNN algorithm is effective, but has some limitations. A typical problem is gradient loss, in which a part of the information disappears in each feedback process [27]. As the distance between certain information and the location of the information becomes longer, the gradient gradually decreases in the back-propagation process, and thus the time dependence cannot be stably captured when the model moves out of a certain step. In addition, it has the disadvantage of analysis based on past data while disregarding information contained in a future situation.

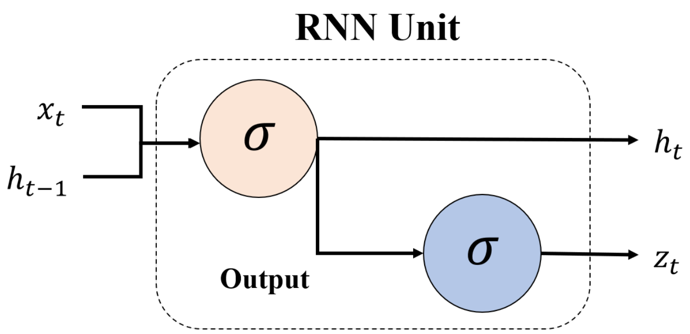


Figure 1. RNN unit.

In order to overcome the gradient loss problem of the RNN structure, the proposed network structure is LSTM (Long-Short Term Memory). The LSTM adds a cell state to the hidden layer of the RNN. The structure is shown in the Figure 2. The learning is performed by discarding or updating a value according to the weight by means of the gates included in the cell state. Each cell guarantees the reliability of information transmission and avoids the problem of loss of gradient [28]. The formula of the LSTM structure is as follows.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (3)$$

$$g_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (4)$$

$$c_T = f_T \circledcirc c_t + i_t \circledcirc g_t \quad (5)$$

$$h_t = o_t \circledcirc c_t tanh(c_t) \quad (6)$$

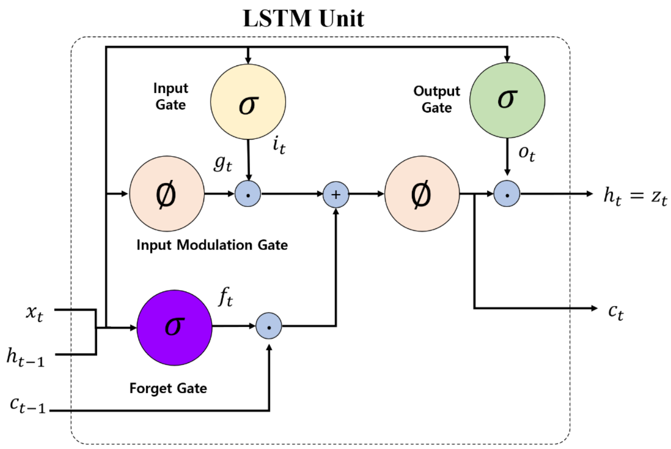


Figure 2. LSTM unit.

The most influential factors in predicting inventory demand are sudden changes in situations and events that require real-time response. These factors can be said to have nonlinear properties. Various data analysis techniques have been rapidly developed over the past decade with the development of computing capabilities and the increase of analytical data. In particular, the development of a prediction model that predicts future results based on past data is being studied and applied in various industries. More and more predictive models through neural networks have shown good performance in predicting complex nonlinear time series data, and case studies using linear and nonlinear models using ensemble techniques are also being actively used. Refs. [29,30] used LSTM to predict the life of the mechanical device, and [31] used an LSTM model to predict power demand for a short time. Ref. [32] conducted a study on smart metering customers in general homes, and deep-RNN surpassed ARIMA 19.5%, SVR 13.1%, and general RNN 6.5%, in terms of RMSE, compared to the latest techniques of predicting loads in the home.

2.3. Kernel Density Estimation

Kernel Density Estimation (KDE) is a non-parametric technique for estimating the statistical function of a random variable. The method for estimating the random variable can be classified into parametric and nonparametric methods. The parametric density estimation begins by mathematically modeling the Probability Density Function (PDF) prior to the estimation. Typically, it is used to follow the modeling by means of the normal distribution. Although this parametric estimation technique has the merit of being simple, it is difficult to apply it in reality, because it must assume the distribution of data in advance. The nonparametric parameter estimation technique estimates the probability density function based on the collected data. For the observed random variable x, the probability density function can be estimated as follows.

$$f(I_n) = \frac{1}{n \cdot h} \sum_{t=1}^{n} K(\frac{I_n - I_n(t)}{h}) \tag{7}$$

In order to estimate kernel density, it is important to decide which function is used as a kernel function. In this paper, we use the most frequently used Gaussian kernel functions, which we can be modeled as follows

$$K(\frac{I_n - I_n(t)}{h}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{I_n - I_n(t)}{h})^2} \tag{8}$$

$$h = 1.06 \cdot \sigma n^{-\frac{1}{5}} \tag{9}$$

In this case, h is the bandwidth, and the optimal parameter values in [33] are given as follows. In this paper, the reliability of the predicted data is expressed by measuring the variability of the data and modeling the probability distribution of the variability by means of the kernel-density estimation technique. So far, we have organized the composition and utilization of the LSTM model and 2D KDE model, which were mainly used in this study, as well as previous studies on demand forecasting using machine learning. In the next section, we will examine the demand forecast framework configured using these models.

3. 2D KDE and LSTM-Based Forecasting Framework for Cost-Effective Inventory Forecasting

3.1. 2D KDE and LSTM-Based Prediction Algorithms for Inventory Forecasting

In this paper, we design a forecasting model that predicts future results based on past data for demand data of a one-dimensional time-series type. The prediction model is largely composed of the following three steps. (1) Point-prediction through LSTM; (2) Interval prediction through 2D kernel density estimation; (3) Perform cost-effective rebalancing of forecast results through interval prediction. In the first step, point prediction is performed by means of LSTM. Point prediction refers to a prediction technique that outputs the result of a prediction as a single value. One performs point prediction based on past data and derives the result as a single predicted value. Second, interval prediction is done by means of a 2D kernel-density estimation. In interval prediction, unlike the point prediction, a range in which a predicted value exists is derived as a section. Finally, in the third step, we use the result of the two-step interval prediction to readjust the point estimate of the first step. At this time, the cost function of the inventory management is reflected in the rebalancing process, so that the cost can be minimized. The overall structure of the framework is shown in Figure 3. In the case of the data used here, since it is SME's forwarding data for five years and it is a SME, there is a large change in demand, and there are many outliers too.

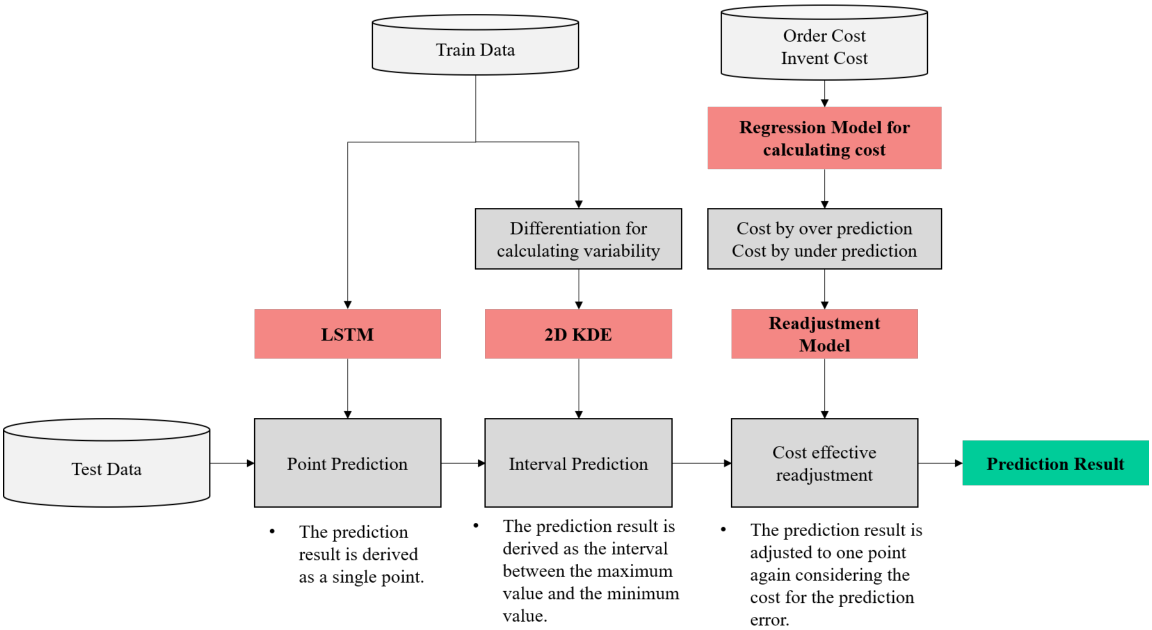


Figure 3. Overall architecture of the framework.

3.2. Interval Prediction by Means of 2D Kernel Density Estimation

The prediction model based on the LSTM described above is a point prediction method that outputs one value about what the future value will be based on the learned data. However, it is very difficult to find a specific pattern for randomly fluctuating data and to predict future outcomes. No matter how sophisticated the prediction model is, the predictive performance of such data drops. Therefore, the point prediction method has a disadvantage in that it is difficult to trust the predicted result according to the state of the data. In order to overcome these limitations, we generally use the interval prediction method. Interval prediction is a prediction method that outputs a section including a prediction result, rather than deriving a prediction result as a single value. It has a different range depending on the reliability: the higher the reliability, the larger the range, and the smaller the range, the more reliable the output. Generally, in the regression model, the prediction interval is calculated based on the actual value and the error between the models. This method is based not on the variability of the data but on how well the model is tailored to the learning data. However, for the learning model designed by means of LSTM, the training data is learned at a very sophisticated level and good predictions are the result. Therefore, since the error between the learning section and the model is very small, it is difficult to expect a great effect of the interval prediction based on the error. Therefore, in order to predict the interval of the LSTM model, the interval prediction method based on the variability of the data should be introduced rather than the learning rate of the model. In this paper, we propose a new interval-estimation method suitable for the LSTM model. The proposed interval-estimation method is based on the modeled probability distribution based on the variability of the data. In order to predict the interval, we first design a 2D KDE map based on past data. The 2D KDE map aims to model the probability distribution based on variability. Therefore, we derive the volatility by differentiating the original data. On the other hand, the differential results of the data are correlated with the original values. In the section where the value of the original data is large, the value is likely to drop, whereas in the section where the value of the original data is small, the value is likely to rise. In other words, since the probability-density function indicating the variability depends on the original data value, the original data is divided into several sections, and the probability-density function is modeled by differentiating the original data for each section. The algorithm is as follows (Algorithm 1) and the results are derived as shown in Figure 4.

Algorithm 1 Creating 2D KDE map

Input: rawData, smoothing rate, quanti
Output: KDEmap, rawScale, diffScale

procedure
  Creating 2D KDE map$(trainTimeSeries, trainLabel, testTimeseries, n, l)$
  $diffData < -calcDifferentiation(Dataset)$
  $diffData, diffScale < -Quantilization(diffData)$
  $diffData, rawScale < -Quantilization(rawData)$
  for I $< -len(diffData)$
  GroupBy$Location[rawData[i]].append(diffData[i])$
  $X = np.arrange(0, QD, 1/smoothingrate)$
  for I $< -len(QR)$
  $estimator = GaussianKDE(GroupByLocation[i])$
  $KDEmap[:, i] = estimator.evaluate(X)$
  return $KDEmap, rawScale, diffScale$
end procedure

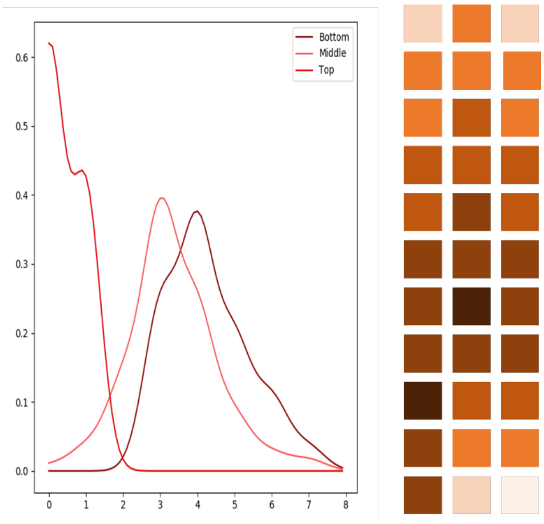


Figure 4. 2D KDEmap example. (**Left**) KDEmap represented by three consecutive probability density function. (**Right**) KDEmap where three discrete function are represented by a heatmap.

Algorithm 1 is used to derive a 2D KDE map modeled by the interval probability-density function. Next, we derive the interval prediction result according to the reliability by means of the derived 2D KDE map. For the interval prediction results, the point prediction results are first derived, and the result is combined with the 2D KDE map to derive the maximum and minimum values. One converts the point prediction result according to the scaling value derived from Algorithm 1 and selects a suitable probability-density function from among the various probability-density functions. Then, a suitable maximum value and minimum value are derived according to the input reliability. The algorithm is as follows (Algorithm 2).

In this way, the result of the interval prediction according to the variability of the data is derived. The result can represent reliable predictions to the user along with the point prediction result. In addition, the prediction result can be readjusted by combining the result of the interval prediction and the result of the point prediction. Such a readjusted prediction result can be more reliable than the previous point prediction result. The results of the interval prediction are expressed as follows.

$$P_{max} = P + KDE_{+}(\sigma) \tag{10}$$

$$P_{min} = P + KDE_{-}(\sigma) \tag{11}$$

In this case, P represents the predicted value by means of the LSTM model, and KDE_+ and KDE_- denote the maximum and minimum values of the prediction interval derived from the 2D kernel density function, respectively.

Algorithm 2 CalcConfidenceInterval.

Input : Predicted, KDEmap, confidencial rate, rawScale
Output : Distance data, trainDistance, testDistance

procedure
  calcConfidenceInterval$(Predicted, KDEmap, confidencialrate, rawScale)$
  $Rate < -(1 - confidentialrate)$
  $Scaled < -rawScale.transform(Predicted)$
  for I $< -len(Predicted)$
  for j $< -KDEmap.shape[0]$
  $Intervalmin + -KDEmap[j, Scaled[i][0]$
  if$(Intervalmin >= Rate)$:
  $confidenceInterval[0] = j$
  break
  for j $< -KDEmap.shape[0]$
  $Intervalmax + -KDEmap[KDEmap.shape[0] - j - 1, Scaled[i][0]$
  if $Intervalmax >= Rate$:
  $confidenceInterval[1] - KDEmap.shape[0] - j - 1$
  break
  return $confidenceInterval$
end procedure

3.3. Calculation of Adjusted Estimates Reflecting the Cost of Prediction Error

In general, if the forecast for a demand for a product is wrong, costs will be incurred. To reduce these costs, the forecasting model must derive accurate predictions. At this time, the costs that result from the errors are not the same as when the demand is overestimated or insufficiently predicted. Excessive demand forecasts lead to inventory and storage costs. On the other hand, if the demand is insufficient, costs will also be incurred. In general, in terms of enterprise inventory management, it is advantageous to order large quantities of products, so that a higher discount rate can be applied to reduce the ordering cost per unit. If the demand is insufficiently predicted, the cost will be incurred by applying this discount rate relatively less. In this way, the costs incurred in each case when the demand is overestimated or underestimated are not the same. In this paper, we propose a prediction interval based on the proposed 2D kernel density estimation and propose a cost-saving prediction model derived from it. The model is constructed as follows. First, the per-unit transient prediction cost and the per-unit underprediction cost are mathematically modeled. In general, storage costs per unit in inventory management take a linearly increasing model of the number of items that must be kept. On the other hand, for order cost per unit, we adopt a model that exponentially decreases as the number of items is increased by applying the discount rate. Then, the cost is applied to the point prediction result and the interval prediction result by means of the previously designed prediction model. Excessive forecasts lead to increased storage costs, and poor forecasts lead to higher order costs. The revised forecasts focused on cost savings can be expressed by the following formula.

$$P_{re} = P - (P \times overweight) + (P \times underweight) \rightarrow overweight > underweight \quad (12)$$

$$P_{re} = P + (P \times overweight) - (P \times underweight) \rightarrow overweight < underweight \quad (13)$$

$$overweight = \frac{C_{over}}{C_{over} + C_{under}} \quad (14)$$

$$overweight = \frac{C_{under}}{C_{over} + C_{under}} \quad (15)$$

4. Performance Analysis

4.1. Experiment Design

We first designed a demand-prediction model based on LSTM. Second, we quantify the variability of the data as a function of the 2D kernel mill, and the error range is visualized by calculating the maximum and minimum errors of the prediction result according to the reliability. Last, we propose a demand forecasting model that minimizes the cost by adjusting the predicted amount to reflect the cost incurred if there is insufficient prediction compared to the actual value and the excess cost. We designed an experiment to evaluate the performance of the proposed algorithm. First, the prediction performance of the LSTM is evaluated with respect to the randomly generated periodic-function data. Second, the performance is verified by doing predictions based on actual inventory data. Then, the confidence interval of the prediction result is set up by means of the 2D kernel density-estimation technique proposed in this paper, and the effect is confirmed. Finally, we confirm whether the revised forecast is effective in terms of cost reduction, reflecting the cost of forecast error. The data used in the experiments is divided into randomly generated periodic-function data (Sine Data) and actual enterprise inventory data. First, randomly generated data has 50 values per year, and it is generated as a total of 250 datasets for 5 years. The period of the sine function shape is circular, and the deviation and size increase linearly with the year. The random noise is added to complete the dataset. Four years' worth of data, 80% of all data, are used as learning data, and the remaining 20% of one-year data is used as test data. The data on the demand data are data on one item among the various products of the S company, which is an SME company(plastic injection molding) of the Republic of Korea. The volume of shipments (demand) is organized by week and consists of data from 2016 to September 2019 (3Q). Of the total data, data from 2016 to 2019 (about 83%) will be used for training, and data for the remaining 2020 (until September) will be used for the test. Each experiment is designed to test the validity of the model. For Sine data, 50 test sets are divided into 5 sets of 10 tests to obtain the predicted values for each set, and the error rate is calculated. On the other hand, the warehousing out data is divided into 4 sets of 10 pieces, which are the predicted values for each set, because data for 2020 to be used as a test are not available after September. By means of this validation, the reliability of the experimental results is secured.

The PC used in the experiment is equipped with Intel R CoreTM i7-8700K CPU @ 3.70 GHz, two-core four-logic processor, and supports 32 GB RAM. Table 1 shows the parameter settings for the LSTM model configuration, and for optimal configuration, loss function evaluated using rmsse and map. The optimizer optimized the model using the adam function, which is commonly used in time series data, and set epoch 1000 times, batch size 4 and verbose 2 to fit the model. Figure 5 shows the Year-wise Box plot and Month-wise Box plot, respectively, to identify the characteristics of the data before time series analysis. In general, trends are identified through the annual Box plot, and Seasonality is identified through the monthly Box plot. In addition, in the case of circularity generally viewed, it reflects repeated but non-periodic fluctuations and usually requires a two-year period, but in this case, it meets the conditions because it is five-year data. When the properties of the data were identified using the Augmented Deckey Fuller Test (ADF Test) for the normality test, 0.23 was derived for the p-value value, which was larger than 0.05, so the normality test was completed. This experiment, which applies various time series analysis models using inventory data, aims to determine whether the desired level of predictive performance can be recorded when learning is performed on actual data.

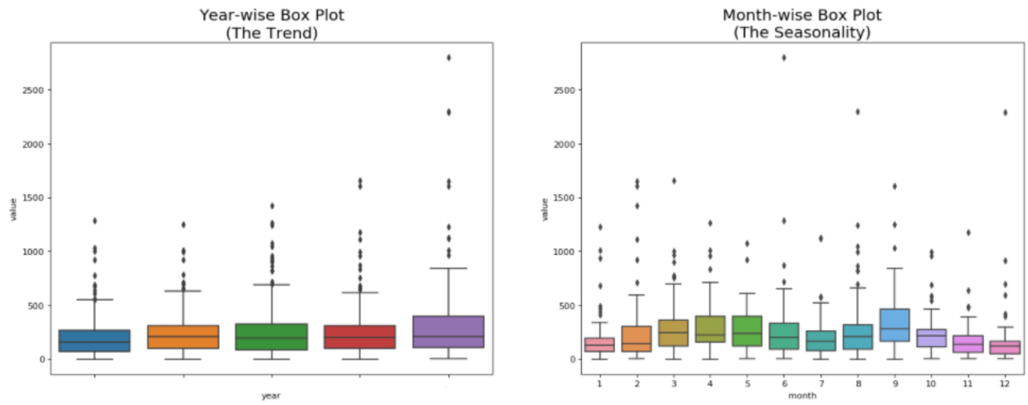


Figure 5. Box plot and seasonality of the real inventory data.

Table 1. LSTM model parameter.

LSTM	units	100
	return_sequence	True
compile	loss function	root_mean_squared_error
		Mean_absolute _percentage _error
	optimizer	adam
fit	nb_epoch	1000
	batch_size	4
	verbose	2

4.2. Performance Accuracy Analysis

Prior to confirming the effectiveness of the algorithm by means of actual inventory data, we verify the performance of the LSTM algorithm by means of our own test data. The accuracy of the prediction model according to the noise level of the test data is confirmed.

Table 2 shows the prediction performance of the LSTM prediction model for the random period signal. An experiment was conducted based on random signal data five times, and each RMSE and MAPE value was derived. A total of five experimental averages were obtained so that the more severe the noise, the more significant the difference was. As can be seen from the experiment, the greater the random variation because of the increase in noise, the less accurate the prediction. That is, the variability of the data greatly affects the performance of the prediction algorithm. For SMEs, inventory data is difficult to be regularized because of the nature of the enterprise; the system of inventory management is inadequate, and there are many factors that greatly affect the performance of the company. Therefore, inventory data of SMEs has a large random variation. This leads to a decline in performance in predicting inventory data by means of predictive models. The following is the result of forecasting performance from the D company's warehouse data. As explained in Experimental design, a total of four LSTM models were tested from 2016 to 2019 September data as training sets and from the remaining 2020 September data as test datasets. The purpose of this experiment is to find out how volatile the warehousing data of SMEs is and to find out the similarity with random signal data. Table 3 shows the predictive performance of the LSTM model for warehousing out data. The prediction error is higher than that for random data of the same condition. This implies that the warehousing out data is very fluctuative. The data used in the experiments are data from actual SMEs, which showed much more growth in 2019 than in the previous year. This irregularity lowers the reliability of the results of the prediction model and makes it difficult to apply it to the field. Therefore, for the inventory-forecasting system for SMEs, it is necessary to design an appropriate forecasting model that takes into consideration the cost of inventory management, while expressing the reliability of the forecasting results in view of such volatility.

Table 2. Performance of LSTM prediction for random signal.

Noise	**0.5**					
	#1	#2	#3	#4	#5	Average
RMSE	81.26246	117.6317	51.70315	82.43249	88.79546	84.36505
MAPE	3.498915	4.735661	2.440641	5.03232	5.401816	4.22187
Noise	**1.0**					
	#1	#2	#3	#4	#5	Average
RMSE	139.8895	199.44	231.1574	173.0788	173.7208	183.4573
MAPE	5.077092	7.807049	11.14622	9.198444	8.848312	8.415424
Noise	**1.5**					
	#1	#2	#3	#4	#5	Average
RMSE	320.0853	269.0784	262.721	269.6649	193.4566	263.0013
MAPE	11.6281	9.711254	11.57804	12.08087	10.46633	11.09292
Noise	**2.0**					
	#1	#2	#3	#4	#5	Average
RMSE	317.9036	300.7283	335.1237	272.3147	228.4517	290.9044
MAPE	12.67522	10.22078	13.586	14.58546	10.63073	12.33964

Table 3. Performance of LSTM prediction for warehousing out data.

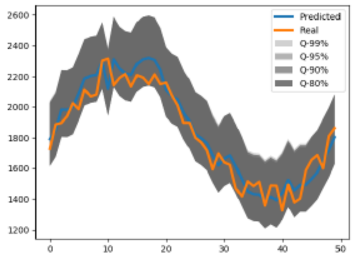
Warehousing Out					
	#1	**#2**	**#3**	**#4**	**Average**
RMSE	653.9884	727.7047	1051.383	1094.891	881.992
MAPE	90.57324	32.43212	37.1611	37.0847	49.285

4.3. The Effect of Interval Prediction by 2D Kernel Density Estimation

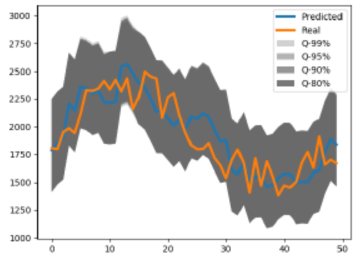
The 2D kernel density estimation proposed in this paper can be used to predict the location of the predicted results according to the confidence level. Interval prediction measures the variability of the accumulated data and models the probability distribution to see how much difference there can be from the predicted results. The following experiment confirms the effect of the interval prediction by estimating the 2D kernel density based on the variability. First, we check the estimated quantity of demand and the quantity of demand for randomly generated data.

The increase in noise means that the volatility increases accordingly. As can be seen in Figure 6, as the noise increases, the prediction interval becomes wider. That is, the increase in the variability because of noise adversely affects the prediction result by means of the LSTM, which makes it difficult to obtain accurate predicted values. Since the interval prediction has an advantage, in that it can secure the reliability of the prediction result, it can be suitably used to obtain the prediction result by means of the data of the SMEs with high volatility.

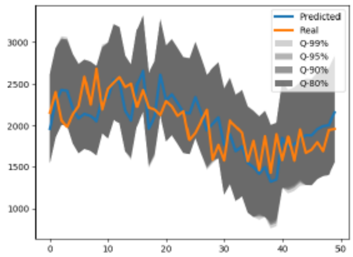
In Figure 7, it can be seen whether the actual value of is included in a prediction interval. Generally, the higher the reliability, the wider the range, and the lower the reliability, the narrower the range. Therefore, the fact that the actual value is located in the lower reliability means that the prediction result is similar to the actual value. By means of this experiment, we confirmed that only two values out of the 200 actual values (Dataset: 4; length of sequence: 10; number of sequence: 5) were out of the range.



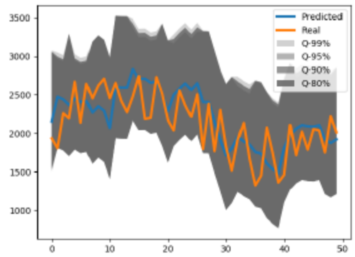
(a) Noise: 0.5



(b) Noise: 1



(c) Noise: 1.5



(d) Noise: 2

Figure 6. The result of interval prediction for random period data.

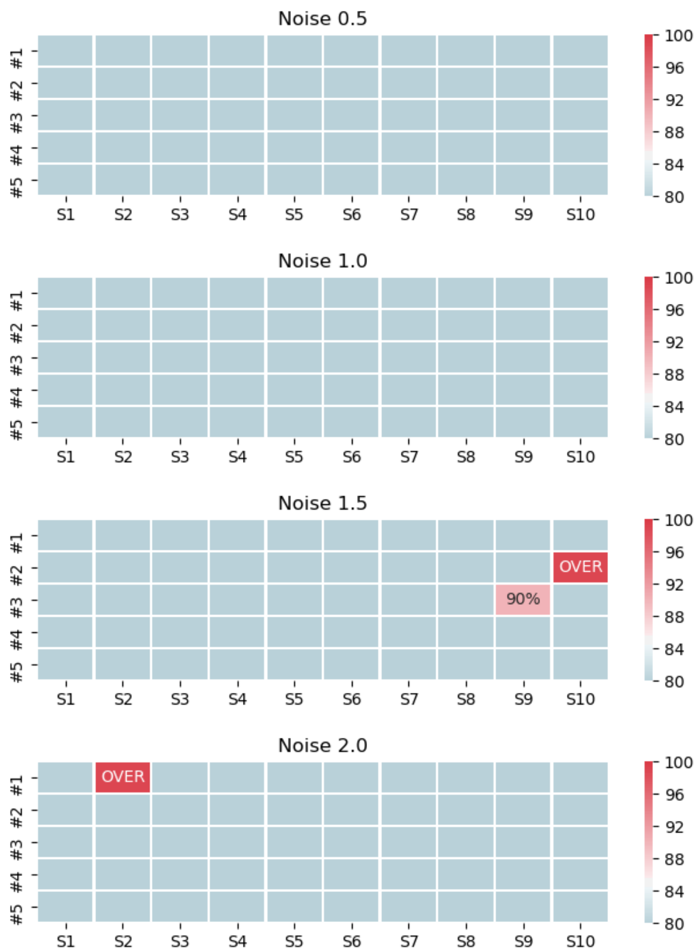


Figure 7. Random value: What prediction range the actual value belongs to.

As can be seen from the previous prediction performance experiments, the warehousing out data is more variable than the randomly generated periodic signals. Accordingly, the prediction interval by means of the interval prediction also has a very wide range. The experimental results show that all of the 40 real values exist in the predictor, the rest, except for 2, are located in the 80% confidence interval, and the result is very stable. As a result, the prediction model by means of the LSTM for the data of SMES with a large volatility is much less accurate, but the interval prediction by means of the 2D kernel density estimation technique helps to secure the reliability of the prediction result by modeling such variability (Figures 8 and 9).

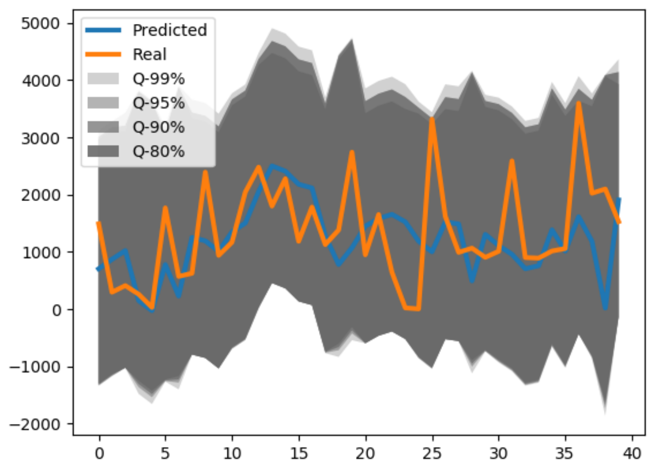


Figure 8. The result of interval prediction for warehousing out data.

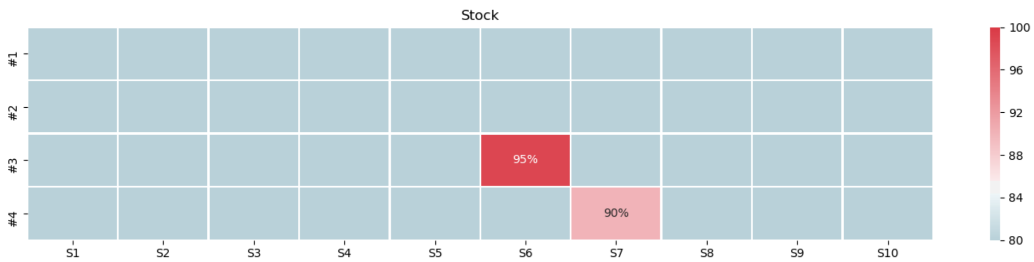


Figure 9. Warehousing out: What prediction range the actual value belongs to.

4.4. Calculation of Cost Savings by Incorporating Forecasting Error Cost

Finally, in this paper, we derive the adjusted prediction results, which are calculated by recalculating the optimal prediction results reflecting the cost of the prediction error, based on the results of the previously derived point prediction and the interval prediction. First, regression modeling is performed on the transient prediction cost and the underestimation cost. A regression model can be constructed from data from a small company, D, that provided the data used in the study. First, the cost of transient forecasting is directly related to the cost of storage per unit, since unnecessary inventories must be kept. For D firms, the storage cost per unit is fixed at KRW 687, which is linearly increased as the prediction error increases.

On the other hand, tribal forecasting costs are the opportunity costs that result from insufficient demand forecasts and high discount rates. Figure 10 illustrates how to derive the underestimation cost. The forecasting cost can be expressed by the following formula.

$$Price(V) = (-1331.6296) \times log(V + 13175.8341) + 37166.7634 \quad (16)$$

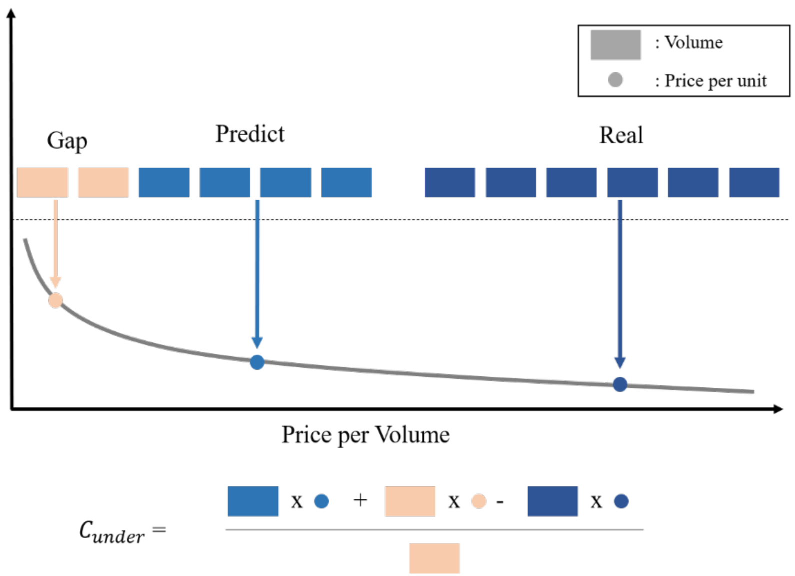


Figure 10. Calculating the underestimation cost.

In this case, V is the order quantity reflecting the prediction result, and *Price* is the product price per unit. In order to derive the underestimation cost, we first calculate the commodity price per unit according to the order quantity. The price per unit is not fixed, and the larger the order, the more the discount rate is applied. Therefore, the price per unit is regression modeled by means of curve fitting. The types of kernel functions to be used in the experiments are selected by using three types of functions: linear, log, and inverse. We used the data for 64 orders from D company and the order cost per unit. The experimental results are as follows.

The experimental results on Table 4 show that the logarithmic function model is suitable for modeling the order cost per unit. Using this, the price per unit can be modeled as follows.

Table 4. Comparison of the linear function model, log function model, and inverse function model.

	RMSE	MAPE
Linear	1257.48	7.53
Log	1091.34	6.73
Inverse	1139.46	7.08

It is possible to calculate the forecasting error cost per unit, depending on the price of the product per calculated unit. We adjust the predicted value over the overestimation cost and underestimation cost per unit according to Equation (16). By means of the above equation, the prediction result is readjusted to reduce the cost according to the weight of each cost. Next, we test the effect of the cost reduction by comparing the prediction result with the existing point prediction and the re-adjusted prediction result according to the above formula. Based on the data for the years 2016–2019, we will output the forecast results for 10 weeks in 2020. As in the above experiments, four validations are done. We compare the predicted results by means of the general LSTM with the predicted results that are readjusted by the interval prediction.(99%, 95%, 90%, 80%, each confidence interval). At this time, the predicted result is not directly related to the order quantity. In order to derive the order quantity, it is necessary to comprehensively consider the quantity of stock currently possessed and the stock quantity to be secured for safety. Therefore, we add the constant value to the predicted result to formulate the order quantity. However, since there is no case where the order quantity becomes negative, the minimum constant value is set to 2000. In the experiment, the change of performance according to the change of the constant value is confirmed.

Experimental results show that cost increases are more effective as the constant value increases and the order quantity increases (Table 5). In addition, as the confidence interval of 2D KDE becomes larger, the cost-saving effect is reduced. As a whole, since the constant value has exceeded a certain level, the readjusted result is consistently meaningful, and the maximum savings of KRW 9,715,207 have occurred. On the other hand, according to the constant value, the cost-saving effect did not appear in less than a certain level. As a result, the smaller the order quantity, the larger the exponential increase in the predicted cost per unit. Finally, we confirm that the result of the rebalancing for cost reduction is realistic. According to the 64 order datasets used in the curve fitting step of the order cost per unit, the quantity ordered at one time ranges from a minimum of 111 to a maximum of 10,560. Using the Gaussian kernel estimation technique, we calculated the probability distribution of the order quantity. As a result, the quantity ordered at one time was more than 101 at 99% probability, more than 8453 at 95% probability, and more than 837 at 90% probability (Table 6).

According to the above experiment and Figure 11, the cost savings were continuously shown when the constants in # 2 and # 3 exceeded the order volume in 2010 and 2015, respectively. At this time, the minimum number of orders used in the formula is 70 and 113, respectively. In reality, the probability of such a small order is less than 2%. Therefore, it can be concluded that the proposed method can record meaningful results with a probability of 98.87% or more in reality.

Table 5. Cost savings based on constant values in each confidence interval (unit: KRW).

Constant	2000	3000	4000	6000	8000	Average
#1						
99%	1,859,188	943,453	2,107,159	3,682,191	4,787,882	2,675,975
95%	1,306,212	1,010,844	2,157,144	3,718,619	4,818,596	2,602,283
90%	1,088,777	1,046,362	2,183,678	3,738,059	4,835,016	2,578,378
80%	88,505	1,087,452	2,214,534	3,760,780	4,854,282	2,559,511
#2						
99%	−232,066	1,113,496	2,697,204	3,045,578	3,376,220	2,000,086
95%	167,343	1,238,132	2,726,510	3,032,720	3,369,273	2,106,796
90%	411,675	1,311,673	2,744,334	3,025,380	3,365,288	2,171,670
80%	704,250	1,404,160	2,767,349	3,016,418	3,360,421	2,250,520
#3						
99%	−2,171,166	708,077	1,074,873	1,816,772	2,248,345	735,380
95%	−992,408	695,522	1,096,472	1,825,633	2,223,315	969,713
90%	−389,000	689,064	1,110,058	1,831,240	2,208,289	1,089,930
80%	266,405	682,180	1,127,245	1,838,353	2,190,066	1,220,850
#4						
99%	4,812,424	878,052	3,558,210	7,319,740	9,582,977	5,230,281
95%	6,715,737	1,014,776	3,679,376	7,381,905	9,640,613	5,686,481
90%	7,156,724	1,093,754	3,749,402	7,417,947	9,674,145	5,818,394
80%	6,109,901	1,189,924	3,834,778	7,461,947	9,715,207	5,662,351
Average	1,730,906	1,006,684	2,426,770	3,994,580	5,015,621	

Table 6. The minimum value of the order quantity used in Equation (1) according to the constant value.

#2, const		2000	2010
99%		−232,066	34,466
95%		167,343	455,022
90%		411,675	691,177
80%		704,250	970,307
Minimum order quantity		70	
#3, const		2060	2065
99%		−690,314	44,527
95%		227,802	605,691
90%		685,110	512,652
80%		513,394	514,342
Minimum order quantity		113	
Confidence interval	99%	95%	90%
Order amount	101	453	837

As a result, it can be confirmed that the re-adjusted prediction result reflecting the prediction error cost is more effective in cost reduction than the general LSTM prediction model.

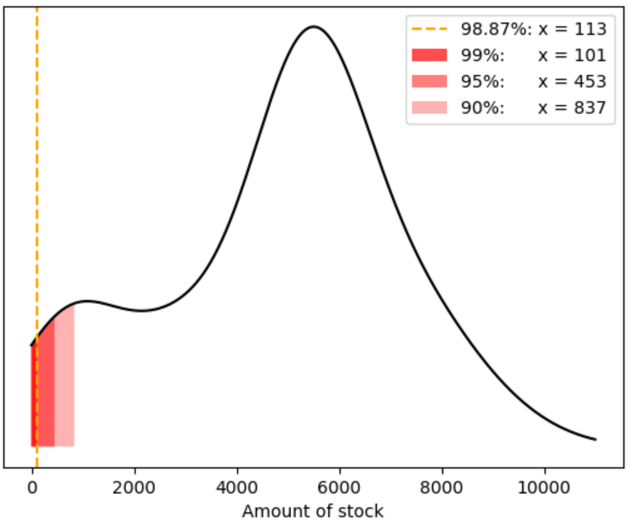


Figure 11. Probability distribution of order quantity (probability of more than 113 orders is more than 98.87% in reality).

5. Conclusions

In this paper, we propose an efficient demand-forecasting algorithm that reflects the characteristics of SMEs. In order to manage inventory, an order quantity or order point should be derived from the predicted results based on the output quantity data of the items held by the company. In terms of forecasting the volume of stock, the characteristics of SMEs are that they do not have a lot of data, because they have no systematic management, and the random variability is very large. This characteristic is a major factor that greatly degrades the accuracy of the prediction model. In this paper, first, we develop a demand-forecasting model by means of LSTM. In addition, we model the variability of historical data by means of the 2D KDE method, and derive the maximum and minimum values of the forecast results according to the reliability along with the forecasted output. Finally, cost-effective forecasting results are derived by reflecting inventory ordering costs and storage costs. Experimental results show that the proposed algorithm can perform reliable interval estimations and can derive prediction results that can reduce the inventory-management cost of the enterprise better than the single LSTM method can. In this paper, we propose an algorithm that can derive reliable prediction results for highly volatile data. These predicted results can be used as a core module in designing a simulator that calculates the timing and quantity of order items for a company by means of simulation. In the case of section prediction using 2D KDE, all 40 actual values were present within the prediction section, and the rest, except for 2, were located in the 80% confidence section. The results were found to be very stable, and the reliability of the prediction results was ensured. In addition, the cost-effective functional technique also showed a cost-saving effect with a 98.87% probability in the real data experiment. The algorithm that we actually developed was provided as a module for an inventory-management solution for the company that provided the data. Future studies will design simulators based on predictions to develop inventory-management solutions that can be used by small and medium-sized enterprises.

Author Contributions: Conceptualization, M.K., J.L., C.L. and J.J.; methodology M.K., J.L. and J.J.; software, M.K. and J.L.; validation, M.K., J.L. and J.J.; formal analysis, M.K. and J.J.; investigation, J.L.; resources, J.L.; data curation, J.J.; writing—original draft preparation, M.K.; writing—review and editing, M.K. and J.L.; visualization, J.L.; supervision, J.J.; project administration, C.L. and J.J.; funding acquisition, J.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Smart Factory Technological R&D Program S2999744 funded by Ministry of SMEs and Startups (MSS, Korea).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the Smart Factory Technological R&D Program S2999744 funded by Ministry of SMEs and Startups (MSS, Korea).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fang, L.; Ma, K.; Li, R.; Wang, Z.; Shi, H. A statistical approach to estimate imbalance-induced energy losses for data-scarce low voltage networks. *IEEE Trans. Power Syst.* **2019**, *34*, 2825–2835. [CrossRef]
2. Zhu, L.; Lu, C.; Dong, Z.Y.; Hong, C. Imbalance learning machine-based power system short-term voltage stability assessment. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2533–2543. [CrossRef]
3. Tang, C.; Wang, S.; Zhou, C.; Zheng, X.; Li, H.; Shi, X. An energy-image based multi-unit power load forecasting system. In Proceedings of the 2018 IEEE International Conference on Industrial Internet (ICII), Seattle, WA, USA, 21–23 October 2018; pp. 69–78.
4. Bhattacharya, B.; Sinha, A. Intelligent fault analysis in electrical power grids. In Proceedings of the 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA, USA, 6–8 November 2017; pp. 985–990.
5. Wang, X.; Zhao, T.; Liu, H.; He, R. Power consumption predicting and anomaly detection based on long short-term memory neural network. In Proceedings of the 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 12–15 April 2019; pp. 487–491.
6. Helwig, N.; Pignanelli, E.; Schütze, A. Condition monitoring of a complex hydraulic system using multivariate statistics. In Proceedings of the 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Pisa, Italy, 11–14 May 2015; pp. 210–215.
7. Jiang, J.R.; Kuo, C.K. Enhancing Convolutional Neural Network Deep Learning for Remaining Useful Life Estimation in Smart Factory Applications. In Proceedings of the 2017 International Conference on Information, Communication and Engineering (ICICE), Xiamen, China, 17–20 November 2017; pp. 120–123.
8. Chen, Z.; Liu, Y.; Liu, S. Mechanical state prediction based on LSTM neural netwok. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 3876–3881.
9. Zhang, W.; Guo, W.; Liu, X.; Liu, Y.; Zhou, J.; Li, B.; Yang, S. LSTM-based analysis of industrial IoT equipment. *IEEE Access* **2018**, *6*, 23551–23560. [CrossRef]

10. Han, Q.; Liu, P.; Zhang, H.; Cai, Z. A wireless sensor network for monitoring environmental quality in the manufacturing industry. *IEEE Access* **2019**, *7*, 78108–78119. [CrossRef]
11. Han, J.H.; Chi, S.Y. Consideration of manufacturing data to apply machine learning methods for predictive manufacturing. In Proceedings of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 109–113.
12. Lee, T.; Lee, K.B.; Kim, C.O. Performance of machine learning algorithms for class-imbalanced process fault detection problems. *IEEE Trans. Semicond. Manuf.* **2016**, *29*, 436–445. [CrossRef]
13. Haddad, B.M.; Yang, S.; Karam, L.J.; Ye, J.; Patel, N.S.; Braun, M.W. Multifeature, sparse-based approach for defects detection and classification in semiconductor units. *IEEE Trans. Autom. Sci. Eng.* **2016**, *15*, 145–159. [CrossRef]
14. Luo, Y.; Qiu, J.; Shi, C. Fault detection of permanent magnet synchronous motor based on deep learning method. In Proceedings of the 2018 21st International Conference on Electrical Machines and Systems (ICEMS), Jeju, Korea, 7–10 October 2018; pp. 699–703.
15. Shi, H.; Bottieau, J.; Xu, M.; Li, R. Deep learning for household load forecasting-A novel pooling deep RNN. *IEEE Trans. Smart Grid* **2018**, *9*, 5271–5280. [CrossRef]
16. Gan, D.; Wang, Y.; Zhang, N.; Zhu, W. Enhancing short-term probabilistic residential load forecasting with quantile long–short-term memory. *J. Eng.* **2017**, *14*, 2622–2627. [CrossRef]
17. Toubeau, J.F.; Bottieau, J.; Vallée, F.; Grève, Z.D. Deep Learning-based Multivariate Probabilistic Forecasting for Short-Term Scheduling in Power Markets. *IEEE Trans. Power Syst.* **2018**, *34*, 1203–1215. [CrossRef]
18. Tang, N.; Mao, S.; Wang, Y.; Nelms, R.M. Solar power generation forecasting with a LASSO-based approach. *IEEE Internet Things J.* **2018**, *5*, 1090–1099. [CrossRef]
19. Wu, J.; Chan, C.K. The prediction of monthly average solar radiation with TDNN and ARIMA. In Proceedings of the 2012 11th International Conference on Machine Learning and Applications, Boca Raton, FL, USA, 12–15 December 2012; Volume 2, pp. 469–474.
20. Hejase, H.A.; Assi, A.H. Time-series regression model for prediction of mean daily global solar radiation in Al-Ain, UAE. *ISRN Renew. Energy* **2012**, 2012, 412471. [CrossRef]
21. Muthusinghe, M.R.S.; Palliyaguru, S.T.; Weerakkody, W.A.N.D.; Saranga, A.M.H. Towards Smart Farming: Accurate Prediction of Paddy Harvest and Rice Demand. In Proceedings of the 2018 IEEE 6th Region 10 Humanitarian Technology Conference, Colombo, Sri Lanka, 6–8 December 2018; pp. 1–16.
22. Chen, X. Prediction of urban water demand based on GA-SVM. In Proceedings of the 2009 ETP International Conference on Future Computer and Communication, Wuhan, China, 6–7 June 2009; Volume 14, pp. 285–288.
23. Ai, S.; Chakravorty, A.; Rong, C. Household EV Charging Demand Prediction Using Machine and Ensemble Learning. In Proceedings of the IEEE International Conference on Energy Internet (ICEI), Southampton, UK, 27–29 September 2021; pp. 163–168.
24. Zhu, M.; Zhang, J. Research of product oil demand forecast based on the combination forecast method. In Proceedings of the 2010 International Conference on E-Business and E-Government, Washington, DC, USA, 7–9 May 2010; pp. 796–798.
25. Wang, Y.; Shen, Y.; Mao, S.; Chen, X.; Zou, H. LASSO and LSTM integrated temporal model for short-term solar intensity forecasting. *IEEE Internet Things J.* **2018**, *6*, 2933–2944. [CrossRef]
26. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
27. Wen, R.; Torkkola, K.; Narayanaswamy, B.; Madeka, D. A multi-horizon quantile recurrent forecaster. *arXiv* **2017**, arXiv:1711.11053.
28. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
29. Kharfan, M.; Chan, V.W.K.; Firdolas Efendigil, T. A data-driven forecasting approach for newly launched seasonal products by leveraging machine-learning approaches. *Data Min. Decis. Anal.* **2021**, *303*, 159–174.
30. Gers, F. Learning to forget: Continual prediction with LSTM. In Proceedings of the 9th International Conference on Artificial Neural Networks: ICANN '99, Edinburgh, UK, 7–10 September 1999; Volume 1999, pp. 850–855.
31. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [CrossRef] [PubMed]
32. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; Routledge: London, UK, 2018.
33. Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. [CrossRef]