International Journal of Information Management xxx (xxxx) xxx

Contents lists available at ScienceDirect

International Journal of Information Management

journal homepage: www.elsevier.com/locate/ijinfomgt

Opinion Paper

# Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management

H.D. Nguyen [a,b], K.P. Tran [b,*], S. Thomassey [b], M. Hamad [c]

[a] *Institute of Artificial Intelligence and Data Science, Dong A University, Da Nang, Viet Nam*
[b] *ENSAIT, GEMTEX - Laboratoire de Génie et Matériaux Textiles, Lille, F-59000, France*
[c] *CEO Driven, 54 Rue Norbert Segard, Hem, 59510, France*

ARTICLE INFO

*Keywords:*
Autoencoder
Long short term memory networks
Anomaly detection
One-class SVM
Forecasting

ABSTRACT

Making appropriate decisions is indeed a key factor to help companies facing challenges from supply chains nowadays. In this paper, we propose two data-driven approaches that allow making better decisions in supply chain management. In particular, we suggest a Long Short Term Memory (LSTM) network-based method for forecasting multivariate time series data and an LSTM Autoencoder network-based method combined with a one-class support vector machine algorithm for detecting anomalies in sales. Unlike other approaches, we recommend combining external and internal company data sources for the purpose of enhancing the performance of forecasting algorithms using multivariate LSTM with the optimal hyperparameters. In addition, we also propose a method to optimize hyperparameters for hybrid algorithms for detecting anomalies in time series data. The proposed approaches will be applied to both benchmarking datasets and real data in fashion retail. The obtained results show that the LSTM Autoencoder based method leads to better performance for anomaly detection compared to the LSTM based method suggested in a previous study. The proposed forecasting method for multivariate time series data also performs better than some other methods based on a dataset provided by NASA.

## 1. Introduction

In today's globally competitive economy, making good decisions is a key factor in the success of any business; a good decision is likely to generate value for the business (Acharya, Singh, Pereira, & Singh, 2018). As a result, the problem of decision-making support in supply chain management (SCM) is a major concern in a large number of studies (Chen, Das, & Ivanov, 2019; Dolgui et al., 2020; Hosseini, Ivanov, & Dolgui, 2019; Ivanov & Dolgui, 2020; Ivanov, Dolgui, & Sokolov, 2019). Among factors that lead to proper decision-making approaches, forecasting and anomaly detection in SCM are two very important tasks. A good forecasting method helps to balance supply and demand, and then avoid understocking or overstocking in retail inventory planning. As a result, other operations of the whole supply chain such as due date management, production planning, pricing, and achieving high customer service levels can be performed better. Meanwhile, a huge amount of data generated at every stage of SCM leads to an overload of data and the difficulty of discerning useful signals, which enable meaningful decisions from meaningless ones. The approaches of anomaly detection allow determining quickly anomalies or unexpected patterns for making more effective decisions. In the literature, many studies have been carried out to provide efficient solutions dealing with these two important tasks.

Related to the use of machine learning algorithms in anomaly detection, most of the current studies do not consider the previous or recent events in detecting the new incoming outlier, i.e., they are based purely on the learning of normally and anomaly behaviors (Bontemps, McDermott, & Le-Khac, 2016). Recently, LSTM emerges as a powerful technique to learn the long-term dependencies and represent the relationship between current events and previous events effectively. Malhotra, Vig, Shroff, and Agarwal (2015) suggested using stacked LSTM networks for anomaly detection in time series. Then, a multi-sensor anomaly detection method based on an LSTM encoder-decoder scheme is extended (Malhotra et al., 2016). A drawback in these two studies is that the authors used an assumption of multivariate Gaussian distribution for error vectors, which may not true in practice. To avoid this assumption, Tran, Du Nguyen, and Thomassey (2019) applied a control chart based method using a kernel quantile estimator. The

* Corresponding author.
*E-mail address:* kim-phuc.tran@ensait.fr (K.P. Tran).

https://doi.org/10.1016/j.ijinfomgt.2020.102282
Received 31 December 2019; Received in revised form 9 October 2020; Accepted 10 November 2020

authors have also pointed out that their LSTM based method outperforms the machine learning-based method in Schölkopf, Platt, Shawe-Taylor, Smola, and Williamson (2001). However, this method may not always effective for the multivariate time series data as only a single value of the characteristic of interest is outputted from the network. From these points of view, the goal of this paper is (1) to provide an LSTM based method for forecasting multivariate time series data and (2) to present an effective method for detecting anomaliesfrom multivariate time series data without using any assumptions for the distribution of prediction errors. In particular, we suggest using a one-class support vector machine (OCSVM) algorithm to separate anomalies from the data outputted based on the LSTM Autoencoder network. In order to assess the suitability of our proposed method, a real case study based on the fashion retailing supply chain is considered. Fashion retailing, and more especially the downstream supply chain, is a very challenging domain that requires advanced intelligent techniques. The considered scenario is described more specifically in the next section.

The rest of the paper is organized as follows. In section 3, we describe the scenarios that motivate the proposed approaches for forecasting and anomaly detection. Section 4 briefly presents the necessary concepts for the proposed method, including the LSTM network, the LSTM Autoencoder network, and the OCSVM algorithm. The approach for forecasting multivariate time series data and for detecting an anomaly in multivariate time series based on the LSTM Autoencoder network and the OCSVM algorithm is presented in Section 5. Section 6 shows the experiment and the obtained results from applying our method for benchmarking and real datasets. In section 7, we discuss the contributions, practical applicability, limitations, and future research direction of this research. Some concluding remarks are given in Section 8.

## 2. Related Works

As mentioned above, forecasting and detecting anomalies from multivariate time series data are critical tasks in SCM. A good performance of these problems enables managers to make better decisions in their work. However, the applications of forecasting and detecting anomalies from multivariate time series data are not limited to SCM. One might see the applications of these two important problems in many domains such as finance, banking, insurance, industrial manufacturing, etc. As a result, references devoted to them are abundant in the literature.

For the anomaly detection problem, Zhao, Dai, and Zhou (2013) improved the quick outlier detection (QOD) algorithm by clustering based on data streams applied to cold chain logistics. Roesch and Van Deusen (2010) suggested a quality control approach for detecting anomalies in the analysis of annual inventory data. Two anomaly detection techniques, including a statistical-based approach and clustering-based approach, were used to detect outliers in sensor data for real-time monitoring systems of the perishable supply chain (Alfian, Syafrudin, & Rhee, 2017). A number of studies focus on abnormal event detection in the supply chain based on radio frequency identification (RFID) technology can be seen in Huang and Wang (2014), Sharma and Singh (2013). Habeeb et al. (2019) provided a comprehensive survey on real-time big data processing for anomaly detection. The authors also proposed a taxonomy to classify existing literature into a set of categories involved in anomaly detection techniques and then analyzed existing solutions based on the proposed taxonomy. A comprehensive survey on deep learning approaches for anomaly detection is conducted in Chalapathy and Chawla (2019). A large number of references have been studied to provide an expansive overview of the problem. The deep learning-based anomaly detection models are divided into types, involving unsupervised, semi-supervised, hybrid, and one-class neural networks. The idea of deep hybrid models is to use deep neural networks mainly autoencoders as feature extractors. After learning within the hidden representations of autoencoders, these features are fed to traditional anomaly detection algorithms such as OCSVM and SVDD (support vector data description) to detect anomalies. This type of deep learning model has been applied in several situations with great success. However, the structure of these deep hybrid models for anomaly detection is just a combination of some separated deep networks like CNN (convolution neural network) and LSTM, with OCSVM or SVDD. Also, this type of model has not yet been applied to multivariate time series.

For the forecasting problem, the auto-regressive integrated moving average (ARIMA) model is commonly used as a methodology for linear time series data, however, it is not suitable for analyzing non-linear data (Zhang, 2003). The machine learning models such as support vector regression and random forest regressor are then developed to deal with non-linear data (Carbonneau, Laframboise, & Vahidov, 2008; Maqsood et al., 2020; Yang et al., 2020). By using nonlinear activation functions, recurrent neural networks (RNNs) are essentially a nonlinear time series model, where the non-linearity is learned from the data. A comparison of ARIMA and long short term memory (LSTM) networks in forecasting time series conducted in Siami-Namini, Tavakoli, and Namin (2018) showed that the LSTM model outperforms the ARIMA model as the average reduction in error rates obtained by LSTM was about 80% when compared to ARIMA. The time series forecasting methods with deep learning are reviewed broadly in Lim and Zohren (2020). The complex structures forming from combinations of deep learning networks like CNN-FNN, LSTM-FNN, CNN-BLSTM, RBM-LSTM-FNN are also introduced to deal with multivariate time series for forecasting (Deng et al., 2020; Ellefsen, Bjørlykhaug, Æsøy, Ushakov, & Zhang, 2019; Xia, Song, Zheng, Pan, & Xi, 2020), where FNN stands for feed-forward neural network, BLSTM stands for bi-directional long short-term memory, and RBM stands for restricted Boltzmann machines. It seems that one has to use more complex structures for deep learning models to get higher performance, and the use of simper deep learning networks for solving the forecasting problem is no longer paying much attention The objective of this study is then to consider the shortcomings in the literature discussed above.

## 3. Scenarios

In retailing, and more especially in fashion retailing, supply chain optimization is crucial to control costs, increase customer satisfaction, manage inventory, and finally improve the profit. The three main factors which make fashion retailing very specific are (Thomassey, 2014):

- the product variety is very high,
- the consumer demand is very fluctuating and sensitive to fashion trends, weather, and price,
- the supply chain of fashion products is very complex and particularly long compared to the short lifespan of products.

To deal with these specificities, fashion retailers have developed a two-part supply chain management (Thomassey, 2010) as illustrated in Fig. 1, including (1) upstream from suppliers to warehouse, a cost-oriented supply chain with bulk procurement based on long-term forecasts, and (2) downstream from the warehouse to local stores, a responsive supply chain with frequent replenishment of stores mainly based on short-term Point Of Sales (POS) data.

In this study, we focus on the downstream supply chain of fashion retailers. As mentioned earlier, consumer demand very fluctuates. When the product variety is high, inventory allocations become very challenging for an extensive store network. Thus, companies rely on efficient and reactive information system to monitor POS data and compute replenishment of each store for the next day or next two days. Combined with efficient transportation and distribution logistics, this process enables companies to drive their local inventories in most situations. However, the high sensitivity of the demand to pricing effect and weather conditions frequently involves sharp and immediate

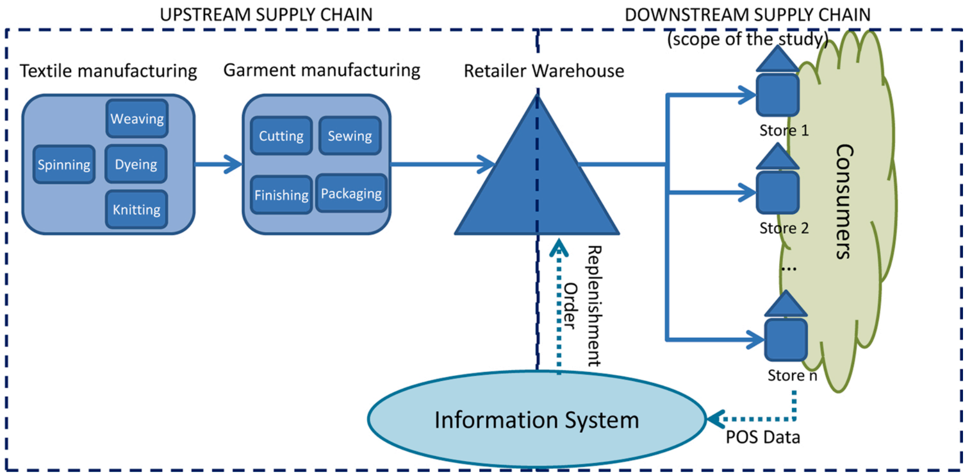


**Fig. 1.** An illustration of a two-part supply chain management.

fluctuations that can not be predicted by the POS data-based replenishment system. Taking into account the different constraints such as small store surfaces, limited staff numbers to manage product reception, shelving, and sales force, these high fluctuations generate significant profit loss. Therefore, a short-term sales forecasting system should be developed to cope with this problem. Different models have been proposed in the literature for this task ((Sirovich, Craparotta, & Marocco, 2018). However, the product variety and extensive store network generate a huge number of situations which are as many sources of forecast errors. To deal with these issues, the proposed approach which combines new advances in forecasting with the LSTM network, the LSTM Autoencoder network, and the OCSVM algorithm. In this context, the aim of our method is not only to predict the exact sales by stock-keeping unit (SKU) and store but also to detect and anticipate exceptional sales in order to enable practitioners to make a suitable decision and adjust their replenishment for highlighted SKU/stores accordingly.

## 4. The needed concepts

In this section, we briefly review some artificial intelligence algorithms that are necessary to build the proposed algorithm for forecasting and anomaly detection, including the LSTM network, the autoencoder network, and the one-class support vector machine algorithm.

### 4.1. Long short term memory networks

LSTM is a type of Recurrent Neural Network (RNN) that allows the network to retain long-term dependencies between data at a given time from many timesteps before. It has a form of a chain of repeated modules of neural networks, where each module includes three control gates, i.e. the forget gate, the input gate, and the output gate. Each gate is composed out of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layers output numbers in the interval [0, 1], representing a portion of input information that should be let through. As the use of a RNN for time series data, the LSTM reads a sequence of input vectors $\mathrm{x} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_t, \ldots\}$, where $\mathrm{x}_t \in \mathbb{R}^m$ represents an m-dimensional vector of readings for $m$ variables at time-instance $t$. We consider the scenario where multiple such time-series can be obtained by taking a window over a larger time-series. Even LSTM can work with any time-series data, one should consider that its performance is not always the same as it could vary depending on the input.

Given the new information $\mathrm{x}_t$ in state $t$, the LSTM module works as follows. Firstly, it decides what old information should be forgotten by outputing a number within $[0, 1]$, say $f_t$ with

$$f_t = \sigma_1(\mathrm{W}_f.[h_{t-1}, \mathrm{x}_t] + \mathrm{b}_f), \tag{1}$$

where $h_{t-1}$ is the output in state $t-1$, $\mathrm{W}_f$ and $\mathrm{b}_f$ is the weight matrices and the bias of the forget gate. Then, $\mathrm{x}_t$ is processed before storing in cell state. The value $i_t$ is determined in the input gate along with a vector of candidate values $\widetilde{C}_t$ generated by a tanh layer at the same time to updated in the new cell state $C_t$, in which

$$i_t = \sigma_2(\mathrm{W}_i.[h_{t-1}, \mathrm{x}_t] + \mathrm{b}_i), \tag{2}$$

$$\widetilde{C}_t = \tanh(\mathrm{W}_c[h_{t-1}, \mathrm{x}_t] + \mathrm{b}_c), \tag{3}$$

and

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t, \tag{4}$$

where $(\mathrm{W}_i, \mathrm{b}_i)$ and $(\mathrm{W}_c, \mathrm{b}_c)$ are the weight matrices and the biases of input gate and memory cell state, respectively. Finally, the output gate, which is defined by

$$o_t = \sigma_3(\mathrm{W}_o.[h_{t-1}, \mathrm{x}_t] + \mathrm{b}_o), \tag{5}$$

$$h_t = o_t * \tanh(C_t). \tag{6}$$

where $\mathrm{W}_o$ and $\mathrm{b}_o$ are the weight matrix and the bias of output gate, determines a part of the cell state being outputed. Fig. 2, which has been reproduced from Fig. 1 in (Tran et al., 2019) with the modifications, presents an illustration of the structure and the operational principle of a typical LTSM module. In this figure, the cell state runs straight down the entire chain, maintaining the sequential information in an inner state and allowing the LSTM to persist the knowledge accrued from subsequent time steps.

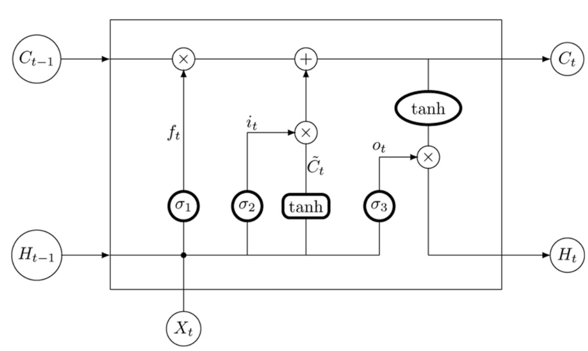


**Fig. 2.** A module of LSTM network.

There are also various variants of LSTM suggested by different authors. A direct comparison of popular variants of LSTM made by Greff, Srivastava, Koutník, Steunebrink, and Schmidhuber (2016) showed that these variations are almost the same; a few among them are more efficient than others but only in some specific problems.

### 4.2. LSTM Autoencoder

Autoencoder is an unsupervised neural network that aims to learn the best encoding-decoding scheme from data. In general, it consists of an input layer, an output layer, an encoder neural network, a decoder neural network, and a latent space. When the data is fed to the network, the encoder compresses them into the latent space, whereas the decoder decompresses the encoded representation into the output layer. The encoded-decoded output is then compared with the initial data and the error is backpropagated through the architecture to update the weights of the network. In particular, given the input $\mathrm{x} \in \mathbb{R}^m$, the encoder compress x to obtain an encoded representation $\mathrm{z} = e(\mathrm{x}) \in \mathbb{R}^n$. The decoder reconstruct this representation to give the output $\hat{\mathrm{x}} = d(\mathrm{z}) \in \mathbb{R}^m$. The autoencoder is trained by minimizing the reconstruction error

$$L = \frac{1}{2}\sum_{\mathrm{x}} \| \mathrm{x} - \hat{\mathrm{x}}\|^2. \tag{7}$$

The main purpose of the autoencoder is not simply to copy the input to the output. By constraining the latent space to have a smaller dimension than the input, i.e. $n < m$, the autoencoder is forced to learn the most salient features of the training data. In other words, an important feature in the design of autoencoder is that it reduces data dimensions while keeping the major information of data structure.

Several types of autoencoders have been proposed in the literature, such as vanilla autoencoder, convolutional autoencoder, regularized autoencoder, and LSTM autoencoder. Among these types, the LSTM autoencoder refers to the autoencoder that both the encoder and the decoder are the LSTM network. The ability of LSTM to learn patterns in data over long sequences makes them suitable for time series forecasting or anomaly detection. That is, the use of the LSTM cell is to capture temporal dependencies in multivariate data. It is shown in (Malhotra et al., 2016) that an encoder-decoder model learned using only the normal sequences can be used for detecting anomalies in multivariate time-series. The encoder-decoder has only seen normal instances during training and learned to reconstruct them. When it is fed with an anomalous sequence, it may not be reconstructed well, leading to higher errors. This has a practical meaning since anomalous data are not always available or it is impossible to cover all the types of these data. Many advantages of using the autoencoder approach have been discussed in (Provotar, Linder, & Veres, 2019). The use of LSTM autoencoder for anomaly detection on multivariate time series data can be seen in several studies, for example, Pereira and Silveira (2018) and Principi, Rossetti, Squartini, and Piazza (2019).

Fig. 3 provides an illustration of a LSTM autoencoder network.

### 4.3. One-class support vector machine

One-class support vector machines (OCSVM) is a machine learning algorithm that aims to estimate the support of distribution. Given a data set $\{\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_i, \ldots, \mathrm{y}_N\}$, $\mathrm{y}_i \in \mathbb{R}^d$, the basic idea behind the OCSVM is to find a hyperplane defined in a high-dimensional Hilbert feature space $\mathcal{F}$ with maximum margin separation from the origin. The data are mapped to space $\mathcal{F}$ through a nonlinear transformation $\Phi(.)$. Then, the problem of separating the data set from the origin is equivalent to solving the following quadratic program (Schölkopf et al., 2001):

$$\underset{\mathrm{w},\mathrm{a},\xi,\rho}{\text{Minimize}} \frac{1}{2}||\mathrm{w}||^2 + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i - \rho \tag{8}$$

$$\text{subjectto}\,(\mathrm{w}.\Phi(\mathrm{y}_i)) \geq \rho - \xi_i, \xi_i \geq 0 \quad \forall i = 1\ldots N, \tag{9}$$

where w is a vector perpendicular to the hyperplane in $\mathcal{F}$, $\rho$ is the distance to the origin, $\xi_i \geq 0$ are slack variables to deal with outliers that may include in the training data distribution, and $\nu \in (0, 1]$ is the parameter to control the tradeoff between the number of examples of the training set mapped as positive by the decision function

$$f(\mathrm{y}) = \mathrm{sgn}((\mathrm{w}.\Phi(\mathrm{y})) - \rho). \tag{10}$$

It should be considered that in this algorithm, it is not necessary to work directly on the scalar product $(\boldsymbol{\Phi}(\mathrm{y}_i).\boldsymbol{\Phi}(\mathrm{y}_j))$. Instead, one can use a kernel function $k(\mathrm{y}_i, \mathrm{y}_j)$ as an efficient alternative. The most commonly used kernel is the radial basis functions (RBF, or Gaussian) kernel:

$$k(\mathrm{y}_i, \mathrm{y}_j) = exp\left(-\frac{||\mathrm{y}_i - \mathrm{y}_j||^2}{2\sigma^2}\right) \tag{11}$$

where $\sigma > 0$ stands for the kernel width parameter. In the feature space, the distance between two mapped samples $\mathrm{y}_i$ and $\mathrm{y}_j$ is:

$$\begin{aligned} ||\phi(\mathrm{y}_i) - \phi(\mathrm{y}_j)||^2 &= k(\mathrm{y}_i, \mathrm{y}_i) + k(\mathrm{y}_j, \mathrm{y}_j) - 2k(\mathrm{y}_i, \mathrm{y}_j) \\ &= 2\left[1 - exp\left(-\frac{||\mathrm{y}_i - \mathrm{y}_j||^2}{2\sigma^2}\right)\right] \end{aligned} \tag{12}$$

Eq. (12) shows a positively proportional relation between $||\phi(\mathrm{y}_i) -$

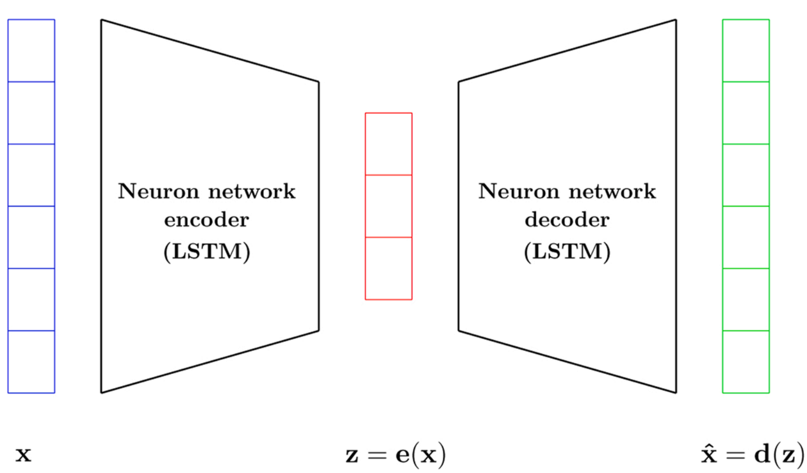


**Fig. 3.** An illustration of a LSTM Autoencoder network.

$\phi\left(\mathrm{y}_j\right)\Big|\Big|$ and $\Big|\Big|\mathrm{y}_i-\mathrm{y}_j\Big|\Big|$. That is to say, the ranking order of the distances between samples in the input and feature spaces is preserved by using the Gaussian kernel.

By using the Lagrangian method and the kernel function, Schölkopf et al. (2001) showed that the problem of solving the quadratic program (8) can be transferred to the following dual optimization:

$$\alpha_i^\star = \underset{\boldsymbol{\alpha}}{\mathrm{Argmin}} \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j k(\mathrm{y}_i,\mathrm{y}_j) \tag{13}$$

$$\mathrm{subjectto}\sum_{i=1}^{N}\alpha_i = 1, 0\le\alpha_i\le\frac{1}{\nu N}, \quad \forall i=1\dots N \tag{14}$$

Samples $\mathrm{y}_i$ that correspond to $0<\alpha_i^\star<\frac{1}{\nu N}$ are called *support vectors*. Let $N_{\mathrm{SV}}$ stands for the number of support vectors, then the discriminant function is reduced to:

$$f(\mathrm{y}) = \mathrm{sgn}\left(\sum_{i=1}^{N_{\mathrm{SV}}}\alpha_i^\star k(\mathrm{y},\mathrm{y}_i)-\rho\right). \tag{15}$$

## 5. Proposed approaches

### 5.1. Multivariate time series forecasting using LSTM

Multivariate time series refers to a time series that has more than one time-dependent variable. That means each variable depends not only on its past values but also has some dependency on other variables. This dependency of multivariate time series is convenient in modeling interesting interdependencies and forecasting future values. However, because of its nature, it can be difficult to build accurate models for multivariate time series forecasting, an important task in many practical applications. In the literature, several multivariate time series predictive models have been proposed such as the vector auto-regressive (VAR) model and the Bayesian VAR model. A summary of advanced multivariate time series forecasting approaches based on statistical models can be seen in (Wang, 2018). Recently, the rapid developments of artificial neuron networks provide a powerful tool to handle a wide variety of problems that were either out-of-scope or difficult to do with classical time series predictive approaches. For example, a multivariate time series forecasting method using LSTM has been suggested for forecasting air quality (Freeman, Taylor, Gharabaghi, & Thé, 2018). The method will be explained in detail below to apply in our situation.

Let $\mathrm{x}_t=\{x_t^{(1)},x_t^{(2)},...,x_t^{(k)}\}, t=1,b,\dots$ denote a multivariate time series at the time $t$ where $k$ is the number of variables. In a supply chain, $\mathrm{x}_t$ could be the value of some specific features such as sales, temperature, humidity and product price. The LSTM network is trained based on a sequence of observed data $\{\mathrm{x}_1,\ \mathrm{x}_2,\dots,\mathrm{x}_N\}$, where $N$ is the number of samples, as follows. Firstly, individual observations are scaled using the MinMaxScaler function by the formula

$$x_{\mathrm{scaled}}^{(i)} = \frac{x^{(i)}-x_{\mathrm{min}}^{(i)}}{x_{\mathrm{max}}^{(i)}-x_{\mathrm{min}}^{(i)}}, i=1,\dots,k, \tag{16}$$

where $x_{\mathrm{max}}^{(i)}$ and $x_{\mathrm{min}}^{(i)}$ are the maximum and minimum values of $x^{(i)}$ in the data set, respectively. To make the notations simple, we write $x^{(i)}$ for $x_{\mathrm{scaled}}^{(i)}$ and understand that this is scaled data. Then, in the training process, we set up a sliding window of size $m$, $m<N$. That is to say, $m$ consecutive multivariate variables are fed to the LSTM at the same time. We will use these $m*k$ inputs to predict the next value of the characteristic of interest, say $x_*^{(1)}$. For example, at the first window, the sequence $\{\mathrm{x}_1,\ \mathrm{x}_2,\dots,\mathrm{x}_m\}$ in the training data set is taken to feed the LSTM and the network can predict the value $\widehat{x}_{m+1}^{(1)}$. In the second one, based on the sequence $\{\mathrm{x}_2,\ \mathrm{x}_3,\dots,\mathrm{x}_{m+1}\}$, the LSTM can predict the value $\widehat{x}_{m+2}^{(1)}$. This process continues until the windows slide to the end of the training data set. The weights of the LSTM network is trained to minimize the loss function of error prediction:

$$L=\sum_{i=m+1}^{N}e_i, \tag{17}$$

where $e_i=\parallel\widehat{x}_i^{(1)}-x_i^{(1)}\parallel$. The performance of the LSTM network is evaluated using the loss metric root mean square error (RSME):

$$\mathrm{RMSE}=\sqrt{\frac{1}{N-m-1}\sum_{i=m+1}^{N}\left(\widehat{x}_i^{(1)}-x_i^{(1)}\right)^2}. \tag{18}$$

After training, the network is used for forecasting. In particular, the value $\widehat{x}_{N+1}^{(1)}$ can be predicted from the LSTM based on the input $\{\mathrm{x}_{N-m+1},\ \mathrm{x}_{N-m+2},\dots,\mathrm{x}_N\}$. In practice, some of the parameters of the model need to be optimized based on the input data to achieve the best performance. In our study, the learning rate, the number of cells, and the dropout will be optimized. The choice of the sliding window is also a question in some situations. However, one should consider the ability to learn the long temporal dependence of the LSTM. This ability makes LSTM not need to pre-determine a specified time window: it can find the optimal look-back number on its own. That is to say, we can try some specific values for the size of the sliding window and let LSTM learn from the data. If one wants to try another value for the sliding window size, other parameters need to be re-optimized and it can take more time. In this study, we will assign a particular value for the sliding window size based on our knowledge of the data. Appendix A provides a pseudocode for the proposed method.

### 5.2. Anomaly detection in using Autoencoder LSTM and OCSVM

The LSTM based method presented in the previous section is for forecasting a specific variable in a multivariate time series. This value can be used to detect anomalies as proposed in (Tran et al., 2019). However, using only this value for anomaly detection can be ineffective in several situations as the dependence of these predicted values on the predicted values of other variables is ignored. In this section, we propose an alternative for anomaly detection using the autoencoder LSTM and OCSVM. The proposed method is as follows.

Suppose that the autoencoder LSTM has been trained from a normal sequence $\{\mathrm{x}_1,\ \mathrm{x}_2,\dots,\mathrm{x}_N\}$, where $N$ is the number of samples and $\mathrm{x}_t=\{x_t^{(1)},x_t^{(2)},\dots,x_t^{(k)}\}, t=1,2,\dots$ is the value of the multivariate time series at the time $t$ with $k$ number of variables (these notations are from previous section). Using a sliding window of size $m$, the trained autoencoder LSTM can read the input sequence $\mathrm{X}_i=\mathrm{x}_t,\dots,\mathrm{x}_{t-m+1}$, encode it and recreate it in the output $\widehat{\mathrm{X}}_i=(\widehat{\mathrm{x}}_t,\dots,\widehat{\mathrm{x}}_{t-m+1})$, with $i=m+1,\dots,N$.. Fig. 4 presents an illustration of the operation of the autoencoder LSTM network for the sliding window of size 2. Since these values has been observed from the data, one can calculate the prediction error vector $\mathrm{e}_i=\widehat{\mathrm{X}}_i-\mathrm{X}_i,\ \ i=m+1,\dots,N$. The anomaly detection is then based on these prediction error vectors. In (Malhotra et al., 2016), the authors supposed that these error vectors follow a Gaussian distribution and then used the maximum likelihood estimation method to estimate the parameters of this distribution. This method is similar to the one suggested in (Malhotra et al., 2015). However, one can argue that the assumption of Gaussian distribution for error vectors may not be true in practice. We overcome the disadvantage of this method by using machine learning algorithms that do not require any specific assumption of data. Among the machine learning algorithms, OCSVM is a very effective algorithm that can be used to detect the anomaly. Since the dependency in the multivariate time series is eliminated by using the autoencoder LSTM, the error vectors $\mathrm{e}_i, i=m+1,\dots,N$ can be considered as independent. From these vectors, the OCSVM can define a

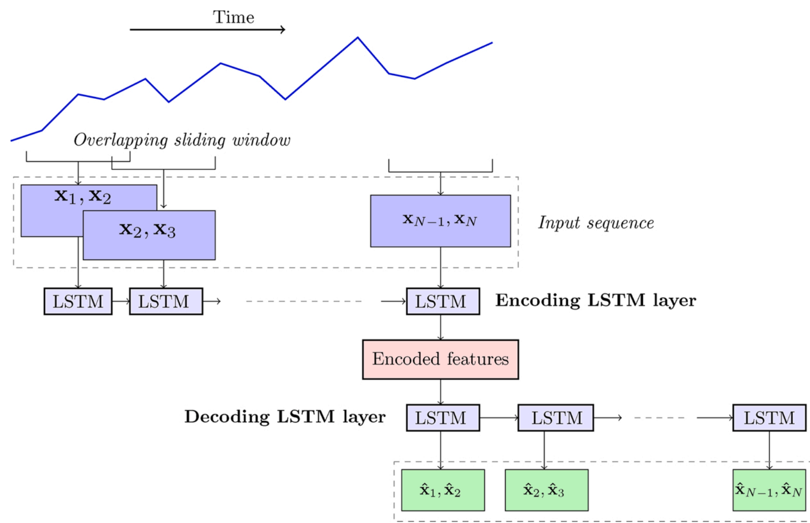


**Fig. 4.** An illustration of the operation of the autoencoder LSTM network for the sliding window of size 2.

hyperplane to separate the abnormal observations from normal samples. Another possible method to avoid the Gaussian distribution assumption is to use the kernel quantile estimation (KQE) method as applied in (Tran et al., 2019). Compared to the anomaly detection method suggested in (Tran et al., 2019), the proposed method in this study has more advantages. The autoencoder LSTM using in this study allows extracting important features from the multivariate time series more efficiently. Moreover, by outputting a vector rather than a component of the vector, the dependence between the components of the predicted vector is held. As a result, it makes the machine learning algorithms for classification or anomaly detection more efficient. Similar to the previous section, the learning rate and the number of cells will be optimized based on the input data rather than being pre-determined for achieving better performance of the model. Pseudocode for the proposed method can be seen in Appendix B.

## 6. Experiment and results

### 6.1. Benchmarking datasets

In this section, we verify the performance of our proposed methods based on two simulated datasets: the C-MAPSS datasets for forecasting and the generated datasets for detecting an anomaly. The code used in this section is available in: https://github.com/huudunguyen/Forecasting_Anomaly_Detection_Auto_LSTM.

#### 6.1.1. C-MAPSS datasets used for forecasting

C-MAPSS (Commercial Modular AeroPropulsion System Simulation) is a simulated turbofan engine degradation datasets produced and provided by NASA and it is widely used in the study of remaining useful life prediction (Saxena & Goebel, 2008). In order to assess the performance of the LSTM based method for forecasting multivariate time series data, similar to Xia et al. (2020) we evaluate the method based on the first dataset of C-MAPSS, i.e. the FD001 dataset. The C-MAPSS FD001 is split into the training set and the test set of multiple multivariate time series. The training set contains the run-to-failure condition monitoring data stream for 100 engines of the same type, while the testing set contains the same type ò·data of engines that ends sometime before failure occurs. The length of condition monitoring data is inconsistent from one engine to another, and it is contaminated with sensor noise, making it a challenging task to predict the remaining useful lifetime (RUL). (Xia et al., 2020). Table 1 presents more details of this dataset.

The objective is to predict the true RUL of each engine in the testing set by using the data from the training set. That is, the data from the

**Table 1**
The C-MAPSS FD001 dataset.

| FD001 | Training set | Test set |
|---|---|---|
| Number of engines | 100 | 100 |
| Number of data | 20631 | 13096 |
| Minimum running cycle | 128 | 31 |
| Maximum running cycle | 362 | 303 |
| Mean running cycle | 206.31 | 130.96 |

training set are fed to train the model and the trained model is used to predict the RUL of testing engines. In the training process, the number of cells, dropouts, and the learning rate of the model are optimized. The optimized model for this dataset is presented in Appendix D. Our computation is performed on a platform with 2.6 GHz Intel(R) Core(TM) i7 and 32GB of RAM. It took about 5 h for the training parameters of the model during the training process. After being optimized, these parameters have been used to re-train the model and to predict RUL on the testing set, this stage took only a few minutes. It should be considered that one can obtain a higher performance of the model by finding optimized parameters with different structures of LSTM. However, it might take more time for training. Figs. 5 and 6 sketch the difference between the predicted RUL and the true RUL from the testing set and the corresponding line plots of train loss and validation loss using our proposed method. The obtained result shows that the predicted values are very close to the true ones. Also, after a few epochs, errors on the training sets and the validation sets decrease remarkably. That is to say,

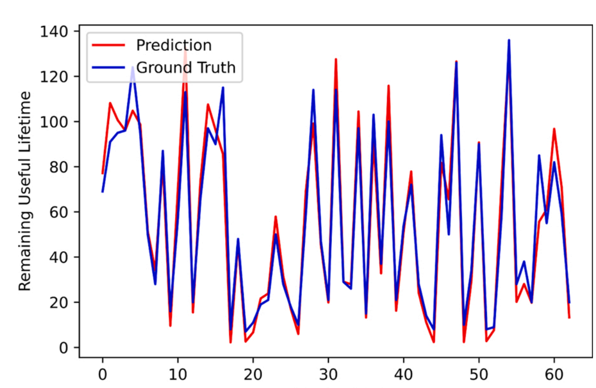


**Fig. 5.** The true RUL and the predicted RUL using LSTM autoencoder for the FD001 dataset.

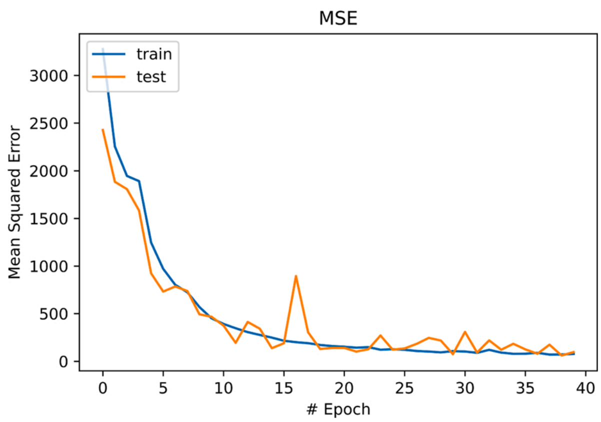


**Fig. 6.** Line plot of train and validation loss from the proposed model during training on the FD001 dataset.

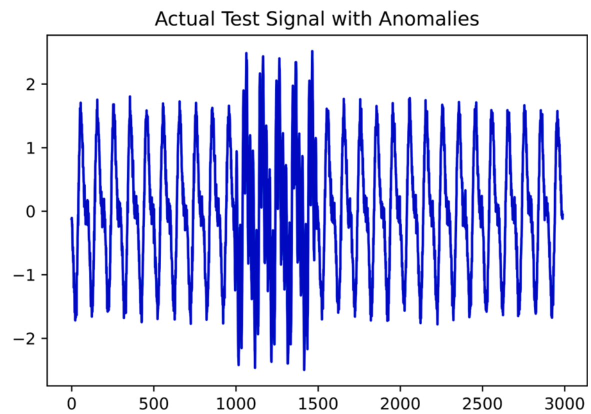


**Fig. 7.** An illustration of the generated data (testing phase).

the proposed LSTM based method for forecasting multivariate time series from the C-MAPSS FD001 dataset is effective.

In the literature, the C-MAPSS FD001 dataset has been extensively studied for verifying an RUL prognostic model and many related studies have been published. Table 2 compares the prognostic performance of our proposed model with some other recent models based on the metric RMSE. It can be seen from Table 2 that although the structure of our LSTM based model is simpler than other ensemble or hybrid models, it still leads to the smallest RMSE. That is, we can say that the proposed method has a superior performance in forecasting multivariate time series data.

### 6.1.2. *Generated data used for detecting anomaly*

For evaluating the performance of the LSTM autoencoder-OCSVM based method in detecting anomaly from multivariate time series, we simulate a training data set of 6988 normal samples and a validation data set of 1398 normal samples representing the normal sales. A function for generating data has been shown in Appendix C. The optimized LSTM autoencoder model from the training process based on this simulated data is displayed in Appendix E.

After training, we compare the performance of the LSTM-KQE based method applied in (Tran et al., 2019), the LSTM Autoencoder-KQE based method, and the LSTM Autoencoder-OCSVM based method proposed in this study through 2989 normal and abnormal samples of a simulated testing data set. In this testing data set, we simulate a small shift from 999th sample to 1499th sample. Fig. 7 displays a graph of generated data for the testing phase.

The comparison is made by using the following measures:

- Accuracy $= \frac{TP + TN}{TP + FP + TN + FN}$
- Recall $= \frac{TP}{TP + FN}$
- Precision $= \frac{TP}{TP + FP}$
- F-score $= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

where TP (True Positive) stands for the number of anomalies correctly diagnosed as anomalies, TN (True Negative) stands for the number of normal events correctly diagnosed as normal, FP (False Positive) stands for the number of normal events incorrectly diagnosed as anomalies, and FN (False Negative) stands for the number of anomalies incorrectly diagnosed as normal events. By their definition, Precision is used to evaluate how accurate the result is, and Recall is used to evaluate how complete the result is. Also, F-score is used to seek a balance between Precision and Recall.

The obtained results are given in Table 3. As can be seen from this Table, the LSTM Autoencoder based method leads to better performance compared to the LSTM based method in Tran et al. (2019). In particular, the Accuracy, the Precision, and the F-score corresponding to the LSTM Autoencoder (in the second row and the third row) are significantly larger than the ones corresponding to the LSTM (in the first row). In addition, the use of the OCSVM algorithm for classification brings the best results with an Accuracy of 98.36%, a Precision of 98.45%, and F-score of 96.98%, and a Recall of 99.59%. That is to say, our proposed method of using the LSTM Autoencoder combining with OCSVM outperforms other methods, ensuring more accurate detection of anomalies in sales. Therefore, this method will be applied in the next section for anomaly detection in a real fashion retail data set.

## 6.2. *Real fashion retail data*

The data are collected from a store in the center of a city in France from 01/01/2015 to 18/11/2019. They are considered as a multivariate time series with five variables, involving sales quantity (of the T-shirts), price discount, temperature, rain (precipitation in mm) and initial price (without discount). Fig. 8 presents the distribution of variables from the collected data.

Fig. 8 illustrates the historical data which are collected. The daily sales (Fig. 8(a)) demonstrate different seasonal effects:

- an annual seasonality related to the product typology (T-Shirt) with higher sales during the summer,
- a weekly seasonality related to consumer behavior, common in retailing activities, with higher sales on Saturdays.

An overall decreasing trend can also be detected since the among of sales seems to decline every year. These features are typically well dealt with time series models. However, some peaks and sharp surges often occur in sales. These variations are produced by different factors. Sales of fashion products are generally considered as very sensitive to price

**Table 2**
RMSE comparison with the literature on the C-MAPSS FD001 dataset.

| Method &Refs. | RMSE |
|---|---|
| MTW-BLSTM ensemble (Xia et al., 2020) | 12.61 |
| LSTM- FW-CatBoost (Deng et al., 2020) | 15.8 |
| RBM-LSTM-FNN (Ellefsen et al., 2019) | 12.56 |
| Proposed method | **9.71** |

**Table 3**
Compare the performance of our proposed method and the method suggested in Tran et al. (2019).

| Method | DR (Recall) | Precision | Accuracy | F-score |
|---|---|---|---|---|
| Method in Tran et al. (2019) | 0.9815 | 0.9465 | 0.9384 | 0.8805 |
| LSTM Autoencoder with KQE | 0.9807 | 0.9583 | 0.9484 | 0.9029 |
| LSTM Autoencoder with OCSVM | 0.9959 | 0.9845 | 0.9836 | 0.9698 |

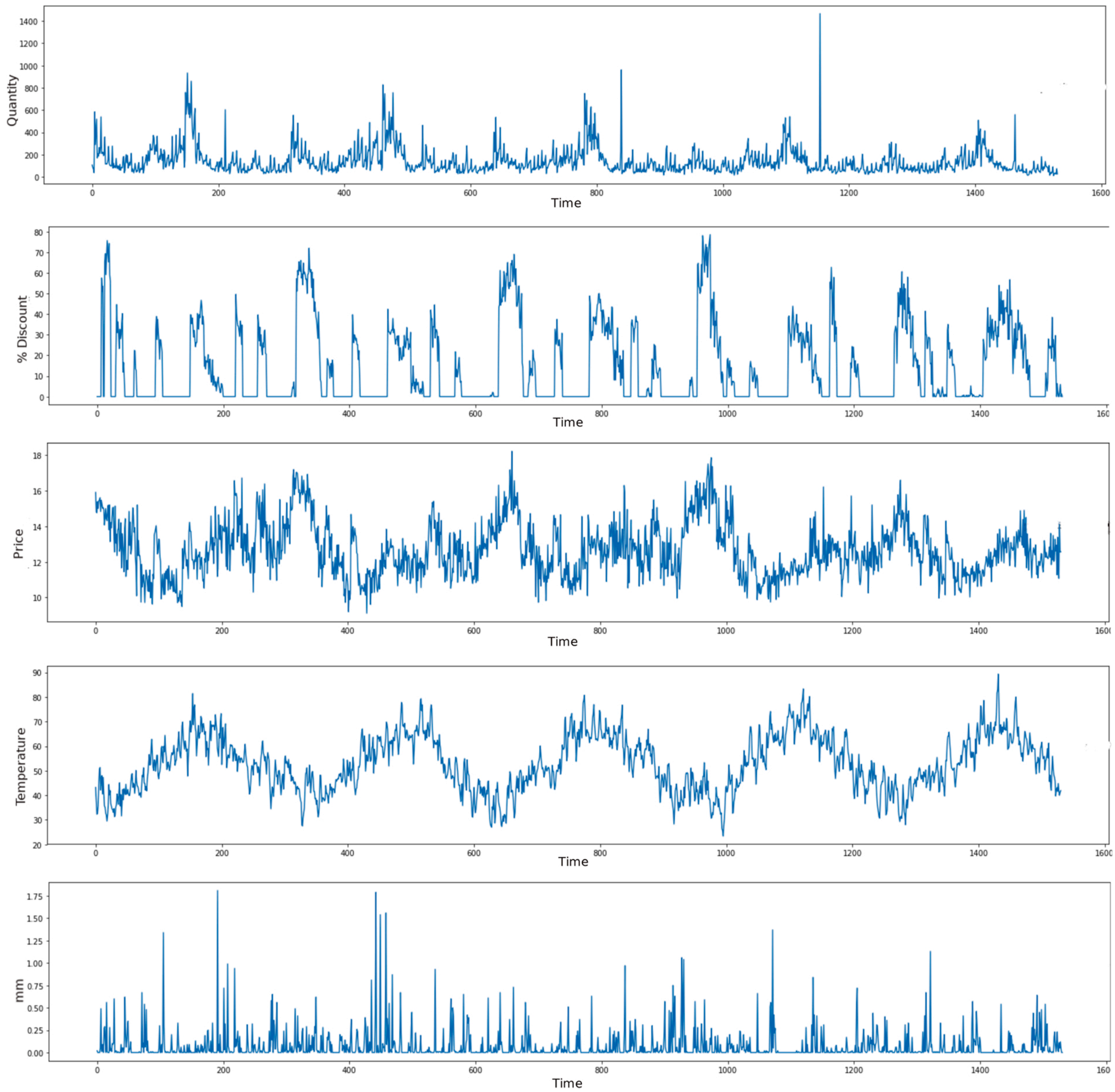


**Fig. 8.** The distribution of variables from multivariate time series data.

discounts and weather data (Thomassey, 2010, 2014). Impacts of these explanatory variables on sales are often complex, nonlinear, period-dependent, and inter-correlated. Consequently, the analysis of these impacts requires a multivariate time series model. Figs. 8b and c show the discount rates and the original price (average) of the T-Shirts. Sales increasing can be identified during discount periods. However, it appears that similar discount rates have very different impacts on sales. Thus, the original price is also considered to complete the information on the discount rate. The weather data, temperature (Fig. 8d), and rainfall (Fig. 8e) give further information to explain the peaks in the sales. It is difficult to measure visually the impacts of these variables since they are generally very brief. The purpose of the proposed forecasting model is to deal with the combinations of all these factors (sales features, discounts, weather data) to provide a forecast as accurate as possible. Nevertheless, unexpected variations, visible in Fig. 8a and identified more specifically in Fig. 12, can not be taken into account by the forecasting model. For this reason, the proposed anomaly detection model aims to detect these variations to enable decision-makers to modify and adapt the replenishment strategy accordingly.

The total data of 1441 days are divided into three parts: 56% (807 days of sales) of data is for the training, 14% of data (202 days of sales) is for validation, and the rest of 30% of the data (432 days of sales) is for testing. For the choice of the test base, we took the sales of a fiscal year as a test (01/04 2018–31/03/2019) and we took one week for each sales period (8 weeks). Moreover, we have tried different ratios which are quite popular in the literature and picked up this ratio since it gave the best performance. However, one should consider that the obtained result also depends on each dataset and the performance of the proposed method would vary based on the portions of the training.

#### 6.2.1. Sales forecasting

From the collected data set, we use a sliding window of the size 30, i. e., we take the data of 30 consecutive days to predict the sales of the next single day. Moreover, we apply the LSTM network with hyper-parameters of 50 LSTM cells, 10 epochs; the dropout is 0.1 and the learning rate is 0.001. The choice of these hyperparameters is based on the method of Grid search hyperparameters for LSTM models. A comparison of the loss function (mean square error) of the model on both training and validation data sets is given in Fig. 9. It shows the differences between the true value and the estimation from the model. Epochs

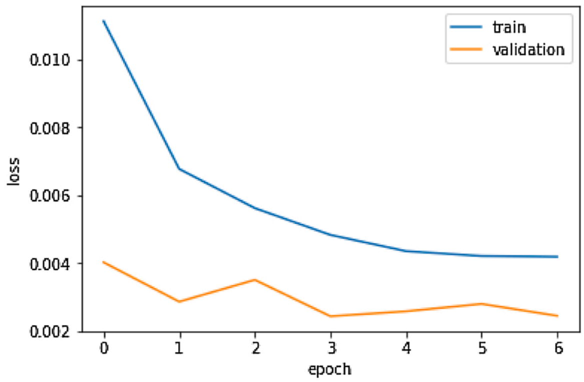


**Fig. 9.** Line plot of train and validation loss from the multivariate LSTM during training.

are defined as the number of complete training pass made by the model on the training dataset. The figure clearly shows a significant decrease in errors on both data sets d after some epochs.

After training the model, we use it to predict the sales on the test data set. Fig. 10 shows a comparison between the real sales and predicted sales using LSTM. From this figure, one can see that the predicted values catch the changing trend of real sales. Moreover, the mean square error corresponding to the scaled data is MSE = 0.05. This value is converted to a real MSE of 95.204. We also calculate the mean forecast error by the formula

$$\text{MFE} = \frac{1}{1441-30}\sum_i \frac{(x_i - \widehat{x}_i)}{x_i} \tag{19}$$

where $x_i$ and $\widehat{x}_i$ represent the real sales and predicted sales at the time $i$. The obtained result is 12.47%. These results show that the LSTM based forecasting model leads to a good prediction and it can be applied to predict the sales in practice. For further research, more features/variables that may affect the sales such as the color of products, the size, and the store could be considered in the multivariate time series to improve the performance of the model.

*6.2.2. Anomaly detection in fashion retail data*

The LSTM Autoencoder network and the OCSVM are trained based on the same training data set as for forecasting in the previous section. Then, anomaly detection is performed based on the testing set. As discussed above, the LSTM Autoencoder transforms input data in different ways using a set of mathematical operations until it learns the essential parameters and the format rules of the input data to reconstruct closed data. To illustrate the difference between the learned representation and the original time series, we consider the prediction error vectors e for the testing phase, we use the principal component analysis method. In Fig. 11, we display the output of LSTM Autoencoder reduced in a two-dimensional coordinate plane. The characteristics extracted from anomaly data tend to be split into a cluster and they are different from the ones extracted from normal data. That is to say, LSTM Autoencoder has done well in its mission to extract attributes from the input. Then, the OCSVM can classify accurately anomalies from those representations and color them.

Fig. 12 shows the anomaly points (red points) which are different from normal behavior. For example, we can see the unusually high values in September 2018 and there are times when the sales are abnormally low like in September 2019. The company should find out the factors that lead to these anomalies. It could be new sales policies, new sales staff, and new style products that lead to higher sales quality;

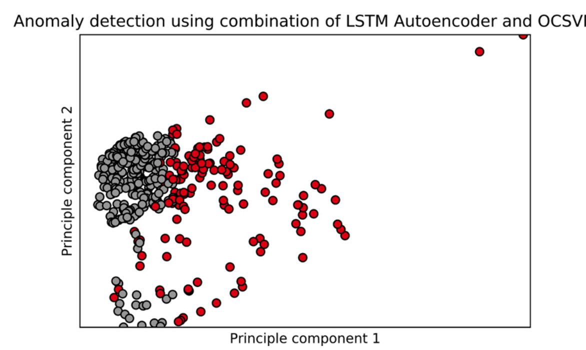


**Fig. 11.** Illustration of the learned representation of LSTM Autoencoder from the original multivariate time series using PCA method.

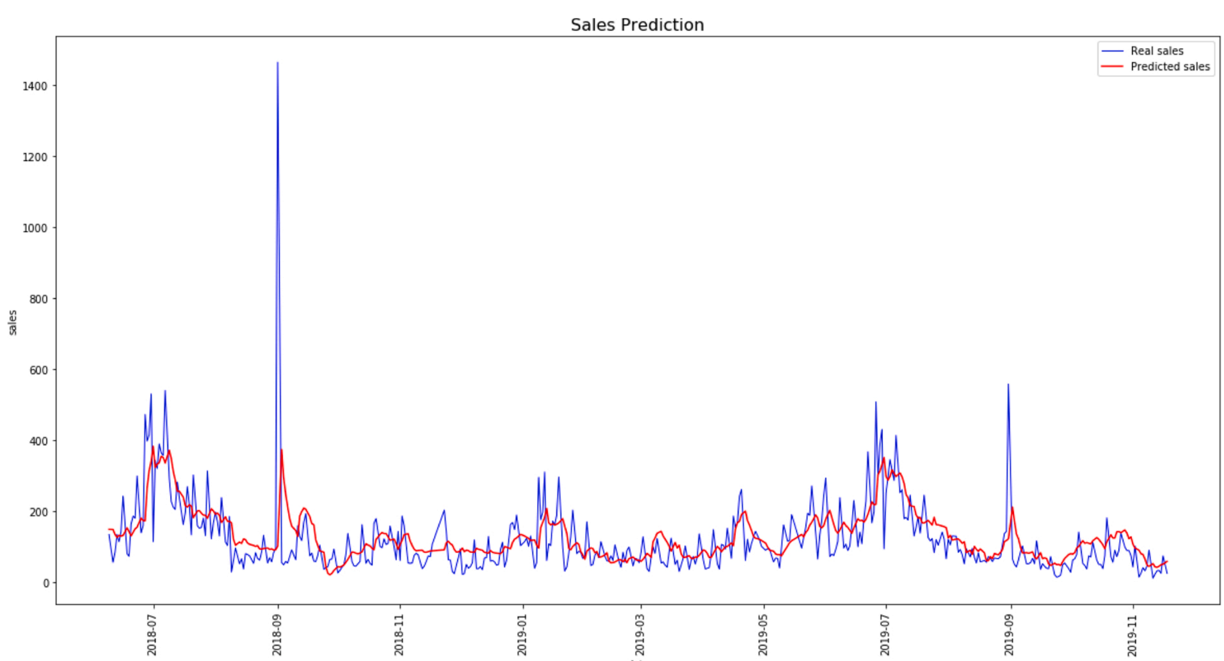


**Fig. 10.** A comparison between real sales and predicted sales using LSTM.

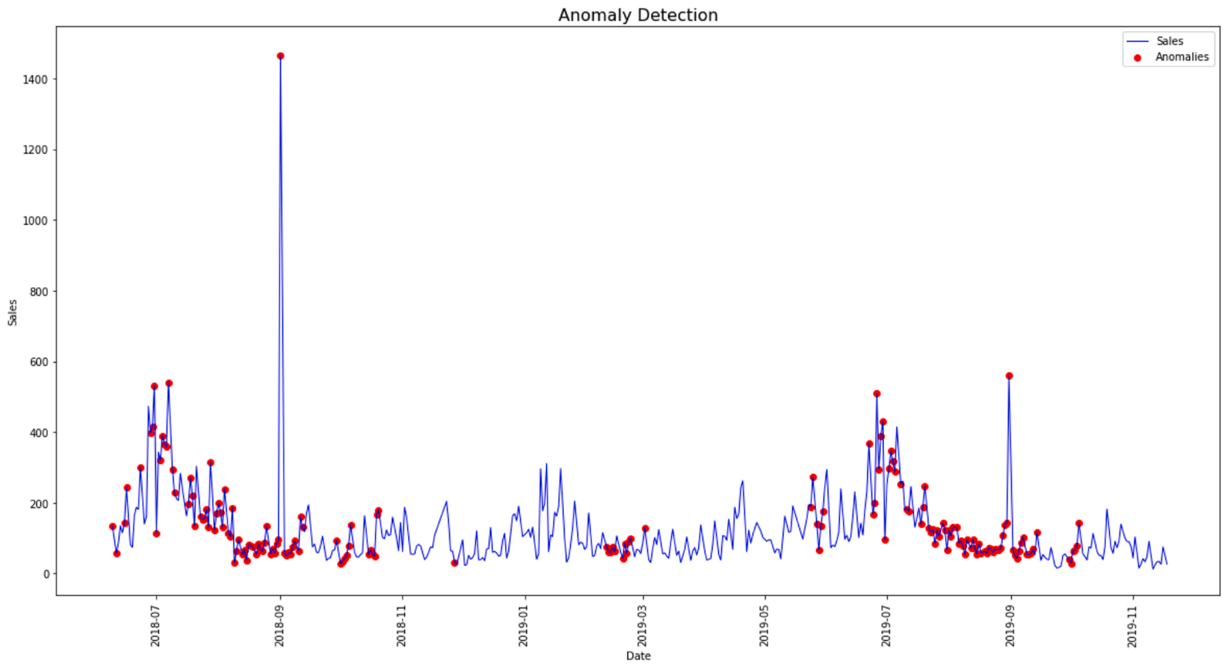


**Fig. 12.** The anomaly detection for real data based on the LSTM Autoencoder network and the OCSVM algorithm.

or they could also be the factors that lead to lower sales quality. Pointing out these anomaly sales may be very useful for companies to make better decisions for future management.

## 7. Discussion

The contributions and practical implications of the proposed methods in this study are discussed in this section.

### 7.1. Theoretical contribution

The theoretical contribution of this study includes two parts. Firstly, we develop a multivariate time series forecasting model based on LSTM with the application in sales forecasting. In order to verify the performance of the proposed forecasting model, we utilized a well-known dataset (i.e. C-MAPSS FD001 dataset provided by NASA) with a large number of samples, including 20631 samples for training and 13096 samples for testing. In the literature, this dataset has been widely used to evaluate the effectiveness of many complex deep learning models like CNN-FNN, LSTM-FNN, CNN-BLSTM, and RBM-LSTM-FNN. Our proposed model, which is simply an LSTM based model, is obviously simpler than these models. However, by considering optimizing the parameters (i.e. learning rate, number of cells, and dropout) rather than choosing a pre-determined value, it has brought a significantly higher performance compared to the performance of others. This finding could be very useful for other authors in designing their deep learning model for a specific purpose of forecasting dealing with not only multivariate time series but also other kinds of data. That is, they can consider a simpler structure and optimize its parameters instead of choosing more complex combinations. In applying our proposed model for a real situation in SCM, we have suggested using weather variables (i.e. temperature, rain - precipitation in mm) to integrate into the model along with traditional variables like initial price and price discount to predict the sales. This could be the first time a forecasting model of sales in SCM considering these weather attributes has been suggested. The use of these variables will help to improve the performance of the forecasting model. Secondly, we have developed a novel deep hybrid model for anomaly detection. The autoencoder LSTM is used as a feature extractor to extract important representations of the multivariate time series input and then these features are input to OCSVM for detecting anomalies. This model results in better performance compared to the performance from several previous studies. We also consider optimizing the hyperparameters of autoencoder LSTM. To the best of our knowledge, the idea of using autoencoder LSTM with optimized hyperparameters and OCSVM for anomaly detection has not been suggested in the literature. The proposed model has been applied to detect anomalies in sales from a real dataset of a fashion company in France.

### 7.2. Implications for practice

As shown in the experiment section, the proposed models can be applied for forecasting and detecting anomalies in sales. An accurate prediction for sales in the near future can help managers to have a good plan for stocking, enhancing economic efficiency, and optimizing the business of the company. In this study, only five variables are considered. However, more factors that may have a significant effect on the sales can be involved in the practice. The accuracy of the proposed method could be improved remarkably once these factors are included in the input. One should consult experts or experienced staff to find out them. Meanwhile, detecting accurately anomalies in sales enables the company to have an insight into its operating and marketing strategies. A negative anomaly in sales may correspond to not good strategies in marketing, leading to a decrease in sales. The strategies need to be reviewed and adjusted. By contrast, once a positive anomaly is detected from the model, it could be useful to investigate and explain the reason, thereby increasing sales and having appropriate strategies for the future. In addition, one should consider that the application of our proposed models is not limited to SCM. In fact, they can be applied to any scenarios related to multivariate time series. For example, the LSTM based forecasting model can be used for stock forecasting, power consumption forecasting, air pollution forecasting, RUL forecasting, etc. The anomaly detection model can be used for fraud detection, cyber-intrusion detection, medical anomaly detection, industrial damage detection, and so on.

### 7.3. Limitations and future research direction

One of the limitations of this study is the ability to access real data from the company due to business security issues. However, the obtained results on the real data are also impressive and they have been confirmed by the company. Another limitation is that the anomaly detection model is used to detect the anomalies that happened in past data. It could be more interesting if it can be used for predicting anomalies happening in the future. This could be considered for further research. For example, we are thinking about combining two models, including forecasting and detecting anomalies models for this task. We think of taking the past data until today to build the forecasting model, and then use this model to predict the sales for tomorrow. After that, we use the predicted value of the sales as the input of the anomaly detecting model, combining with the sales from previous *H*-1 days including the today sales (where *H* is the size of the sliding window), to find out if this value is an anomaly or not. This study could be very useful for companies to have an effective and early strategy. Finally, we are thinking of improving the performance of the proposed anomaly detection model by using another version of Autoencoder like the variational autoencoder (Kingma & Welling, 2019). An advantage of a variational autoencoder is that it can avoid overfitting and ensure that the latent space has good properties of enabling the generative process.

## 8. Conclusions

In this paper, we have focused on two important problems in supply chain management, involving forecasting sales and detecting an anomaly in sales. The LSTM based method for multivariate time series has been suggested for forecasting while the LSTM Autoencoder combining with the OCSVM has been used for anomaly detection. We have applied our proposed approaches to generated data and real data from fashion retail. The obtained results have shown that these methods worked well on both kinds of data. On generated data, the autoencoder LSTM OCSVM based method outperforms the LSTM based method suggested in (Tran et al., 2019) for anomaly detection. The LSTM based forecasting model also performs better some other complex models in RUL forecasting from the C-MAPSS FD001 dataset provided by NASA. On real data, the trend in real sales can be well predicted with a small MSE. The theoretical contributions include the proposed use of combined external and internal company data sources to enhance the predictability of the LSTM model while optimizing the hyperparameters of the LSTM model based on data to help us achieve higher performance than previously proposed methods on both benchmarking and real datasets. Similarly, the performance of the proposed new irregular algorithm is also improved by optimizing the hyperparameters of the model. In addition, the algorithms proposed in this study can be applied to any field with abnormal forecasting and detection needs. In future studies, we expect to be able to further improve the performance of this forecasting model by using additional external data sources such as GDP, unemployment rate, data from social networks, etc. In addition, combining the predictive model and the anomaly detection model to design an algorithm that can predict future anomalies in sales can be an interesting research direction.

## Authors' contribution

Kim Phuc Tran: Conceptualization, Methodology, Supervision, Reviewing and Editing, Huu Du Nguyen: programming, Writing – Original draft preparation. Sebastien Thomassey: Writing – Reviewing and Editing. Moez Hamad: programming.

## Acknowledgements

The authors would like to thank editors and the anonymous referees for their insightful and valuable suggestions which helped to improve the quality of the final manuscript.

## Appendix

### *A. Multivariate time series forecasting using LSTM*

**Algorithm 1**. Multivariate time series forecasting using LSTM

**Input**: $(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N)$ - a sequence of multivariate variables, number of epochs $B$, learning rate $\lambda$, sliding window size $m$, dropout rate, number of cell.
**Output**: The predited value $x^{(1)}_{N+1}$ which is the first component of $\mathrm{x}_{N+1}$.

Initialize the parameters of the LSTM
**for** $i \in \{1, \ldots, B\}$ **do**
  $\widehat{\mathrm{x}}^{(1)}_t \longleftarrow$ LSTM $(\mathrm{x}_t, \ldots, \mathrm{x}_{t-m+1})$
  Let the loss $L$ be defined as $\| \widehat{\mathrm{x}}^{(1)}_t - \mathrm{x}^{(1)}_t \|$
  Optimize the parameters LSTM based on the loss function $L$ and the back propagation method with learning rate $\lambda$.
**End for**
Input data in LSTM to generate the predicted value $\mathrm{x}^{(1)}_{N+1}$
**return** $\mathrm{x}^{(1)}_{N+1}$.

### *B. Multivariate time series anomaly detection using LSTM autoencoder and OCSVM*

**Algorithm 2**. Multivariate time series anomaly detection using LSTM autoencoder and OCSVM

**Input**: $(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N)$ – a sequence of multivariate variables, number of epochs $B$, learning rate $\lambda$, sliding window size $m$, dropout rate, number of cell
**Output**: The classified input as normality or anomaly.

Initialize the parameters of the LSTM autoencoder
**for** $i \in \{1, \ldots, B\}$ **do**
  $\widehat{\mathrm{X}}_j = (\widehat{\mathrm{x}}_t, \ldots, \widehat{\mathrm{x}}_{t-m+1}) \longleftarrow$ LSTM autoencoder$(\mathrm{X}_j = \mathrm{x}_t, \ldots, \mathrm{x}_{t-m+1}), j = m+1, \ldots, N$

(*continued on next page*)

(*continued*)

Let the loss $L$ be defined as $\| \widehat{X}_j - \widehat{X}_j \|$
Let the predicted error vector $e_t$ be defined as $e_j = \widehat{X}_j - \widehat{X}_j, j = m+1, \ldots, N$
Optimize the parameters LSTM autoencorder based on the loss function $L$ and the back propagation method with learning rate $\lambda$.
Optimize the parameters OCSVM based on the predicted error vector $e_j$
**End for**
Input data in LSTM autoencoder - OCSVM for classification
**return** The classified input as normality or anomaly.

### C. Generating time series data

**Algorithm 3.** Generating time series data

Generate a synthetic wave by adding up a few sine waves and some noise
**Output**: the final wave
$t \longleftarrow$ an initial sequence of size $n$
wave1 $= \sin(2*2*\pi*t)$
noise $\longleftarrow$ a random normal sample of size $t$
wave1 $\longleftarrow$ wave1 + noise
wave2 $\longleftarrow \sin(2*\pi*t)$
t.rider $\longleftarrow$ an initial sequence of size $m, m \ll n$
wave3 $\longleftarrow -2*\sin(10*\pi*t.\text{rider})$
insert $\longleftarrow$ an interger value less than $n - m$
wave1[insert:insert + m] $\longleftarrow$ wave1[insert:insert + m] + wave3
**return**: wave1 - 2*wave2

### D. Optimized LSTM model for RUL based on FD001 dataset

#Best parameters:
num_cells=50
dropout_rate=0.1
lr=0.01

| Layer (type) | Output Shape | Param # |
|---|---|---|
| lstm_1 (LSTM) | (None, 30, 50) | 13600 |
| dropout_1 (Dropout) | (None, 30, 50) | 0 |
| lstm_2 (LSTM) | (None, 50) | 20200 |
| dropout_2 (Dropout) | (None, 50) | 0 |
| dense_1 (Dense) | (None, 1) | 51 |
| activation_1 (Activation) | (None, 1) | 0 |

Total params: 33851
Trainable params: 33851
Non-trainable params: 0

### E. Optimized LSTM autoencoder model for anomaly detection based on the generated dataset

#Best parameters:
num_cells=256/64/64/256
lr=0.01

| Layer (type) | Output Shape | Param # |
|---|---|---|
| lstm_1 (LSTM) | (None, 10, 256) | 264192 |
| lstm_2 (LSTM) | (None, 64) | 82176 |
| repeat_vector_1 | (None, 10, 64) | 0 |
| lstm_3 (LSTM) | (None, 10, 64) | 33024 |
| lstm_4 (LSTM) | (None, 10, 256) | 328704 |
| time_distributed_1 | (None, 10, 1) | 257 |

Total params: 708353
Trainable params: 708353
Non-trainable params: 0

## Appendix A. Supplementary Data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.ijinfomgt.2020.102282.

## References

Acharya, A., Singh, S. K., Pereira, V., & Singh, P. (2018). Big data, knowledge co-creation and decision making in fashion industry. *International Journal of Information Management, 42*, 90–101.

Alfian, G., Syafrudin, M., & Rhee, J. (2017). Real-time monitoring system using smartphone-based sensors and nosql database for perishable supply chain. *Sustainability, 9*(11), 2073.

Bontemps, L., McDermott, J., & Le-Khac, N.-A. (2016). Collective anomaly detection based on long short-term memory recurrent neural networks. *International conference on future data and security engineering, Springer*, 141–152.

Carbonneau, R., Laframboise, K., & Vahidov, R. (2008). Application of machine learning techniques for supply chain demand forecasting. *European Journal of Operational Research, 184*(3), 1140–1154.

Chalapathy, R., & Chawla, S. (2019). *Deep learning for anomaly detection: A survey*. arXiv preprint arXiv:1901.03407.

Chen, H. Y., Das, A., & Ivanov, D. (2019). Building resilience and managing post-disruption supply chain recovery: Lessons from the information and communication technology industry. *International Journal of Information Management, 49*, 330–342.

Deng, K., Zhang, X., Cheng, Y., Zheng, Z., Jiang, F., Liu, W., & Peng, J. (2020). A remaining useful life prediction method with long-short term feature processing for aircraft engines. *Applied Soft Computing*, 106344.

Dolgui, A., Ivanov, D., Potryasaev, S., Sokolov, B., Ivanova, M., & Werner, F. (2020). Blockchain-oriented dynamic modelling of smart contract design and execution in the supply chain. *International Journal of Production Research, 58*(7), 2184–2199.

Ellefsen, A. L., Bjørlykhaug, E., Æsøy, V., Ushakov, S., & Zhang, H. (2019). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety, 183*, 240–251.

Freeman, B. S., Taylor, G., Gharabaghi, B., & Thé, J. (2018). Forecasting air quality time series using deep learning. *Journal of the Air & Waste Management Association, 68*(8), 866–886.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems, 28*(10), 2222–2232.

Habeeb, R. A. A., Nasaruddin, F., Gani, A., Hashem, I. A. T., Ahmed, E., & Imran, M. (2019). Real-time big data processing for anomaly detection: A survey. *International Journal of Information Management, 45*, 289–307.

Hosseini, S., Ivanov, D., & Dolgui, A. (2019). Review of quantitative methods for supply chain resilience analysis. *Transportation Research Part E: Logistics and Transportation Review, 125*, 285–307.

Huang, S. P., & Wang, D. (2014). Research on supply chain abnormal event detection based on the RFID technology. *Applied Mechanics and Materials, Vol. 513* (pp. 3309–3312). Trans Tech Publ..

Ivanov, D., & Dolgui, A. (2020). A digital supply chain twin for managing the disruption risks and resilience in the era of industry 4.0. *Production Planning & Control*, 1–14.

Ivanov, D., Dolgui, A., & Sokolov, B. (2019). The impact of digital technology and industry 4.0 on the ripple effect and supply chain risk analytics. *International Journal of Production Research, 57*(3), 829–846.

Kingma, D. P., & Welling, M. (2019). *An introduction to variational autoencoders*. arXiv preprint arXiv:1906.02691.

Lim, B., & Zohren, S. (2020). *Time series forecasting with deep learning: A survey*. arXiv preprint arXiv:2004.13408.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). *LSTM-based encoder-decoder for multi-sensor anomaly detection*. arXiv preprint arXiv:1607.00148.

Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. *Proceedings, Vol. 89*, 89–94.

Maqsood, H., Mehmood, I., Maqsood, M., Yasir, M., Afzal, S., Aadil, F., Selim, M. M., & Muhammad, K. (2020). A local and global event sentiment based efficient stock exchange forecasting using deep learning. *International Journal of Information Management, 50*, 432–451.

Pereira, J., & Silveira, M. (2018). Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention. *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, 1275–1282.

Principi, E., Rossetti, D., Squartini, S., & Piazza, F. (2019). Unsupervised electric motor fault detection by using deep autoencoders. *IEEE/CAA Journal of Automatica Sinica, 6* (2), 441–451.

Provotar, O. I., Linder, Y. M., & Veres, M. M. (2019). Unsupervised anomaly detection in time series using LSTM-based autoencoders. *IEEE international conference on advanced trends in information theory (ATIT)*, 513–517.

Roesch, F. A., & Van Deusen, P. C. (2010). Anomaly detection for analysis of annual inventory data: A quality control approach. *Southern Journal of Applied Forestry, 34* (3), 131–137.

Saxena, A., & Goebel, K. (2008). C-MAPSS data set. *NASA Ames prognostics data repository*.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13* (7), 1443–1471.

Sharma, M., & Singh, M. (2013). Outlier detection in RFID datasets in supply chain process: A review. *International Journal of Computer Applications, 65*(25).

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of arima and LSTM in forecasting time series. *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, 1394–1401.

Sirovich, R., Craparotta, G., & Marocco, E. (2018). An intelligent fashion replenishment system based on data analytics and expert judgment. *In artificial intelligence for fashion industry in the big data era* (pp. 173–195). Springer.

Thomassey, S. (2010). Sales forecasts in clothing industry: The key success factor of the supply chain management. *International Journal of Production Economics, 128*(2), 470–483.

Thomassey, S. (2014). Sales forecasting in apparel and fashion industry: A review. *Intelligent fashion forecasting systems: Models and applications* (pp. 9–27). Springer.

Tran, K. P., Du Nguyen, H., & Thomassey, S. (2019). Anomaly detection using long short term memory networks and its applications in supply chain management. *IFAC-PapersOnLine, 52*(13), 2408–2412.

Wang, M. L. (2018). Advanced multivariate time series forecasting models. *Journal of Mathematics and Statistics*, (14), 253–260.

Xia, T., Song, Y., Zheng, Y., Pan, E., & Xi, L. (2020). An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation. *Computers in Industry, 115*, 103182.

Yang, R., Yu, L., Zhao, Y., Yu, H., Xu, G., Wu, Y., & Liu, Z. (2020). Big data analytics for financial market volatility forecast based on support vector machine. *International Journal of Information Management, 50*, 452–462.

Zhang, G. P. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing, 50*, 159–175.

Zhao, W., Dai, W., & Zhou, S. (2013). Outlier detection in cold-chain logistics temperature monitoring. *Elektronika ir Elektrotechnika, 19*(3), 65–68.