Product Backorder Prediction with Deep Neural Network on Imbalance Data

Author1[a] and Author2[b]

[a]Name of the University, Address, Country; [b]Name of the University, Address, Country;

ABSTRACT

The refusal of the ordered product is a common scenario in inventory and supply chain management systems. The prediction of the chance about product backorder surely minimizes the company's loss. As the number of backorders is extremely lower than the successful ones, applying a predictive model is a challenging task in this domain. This paper proposes a deep neural network-based backorder prediction model by handling the data imbalance introducing some efficient techniques. To make the dataset balanced, we employ different techniques that include minority class weight boosting, randomized oversampling, SMOTE oversampling and the combination of oversampling and under-sampling. Then the balanced training data is exploited by our proposed fully connected deep neural network model to train the predictive model. The predictive model learns the chance of product backorder using the training samples. We conducted experiments on the standard dataset to test the performance of our proposed DNN-based method. The experimental results achieved a new state-of-the-art performance and outperformed some prominent classification models in terms of standard evaluation metrics.

KEYWORDS

Product Backorder; Deep Neural Network; Synthetic Oversampling; Fully Connected Network;

1. Introduction

In inventory management systems, the products that are temporarily out of stock, but the customer still can place the order by considering the future inventory. In other words, the backorder usually indicates that the customer needs a product that is not in stock of inventory to supply. Nevertheless, the customer can order the product. For example, a giant renowned phone making company has announced that they are going to release a new smart-phone. But the starting production will not be sufficient as their phones have huge popularity in the market. The customer also has a burning demand for the new release, even they are willing to wait a bit for the production. Therefore they place the order for future release. In this situation, two different problems can occur. The customer may not wait after a certain period of time and they eventually cancel the order after the waiting time frame. On the other hand, if the company produces so many items in managing backorder, that increases the inventory cost. Moreover, the task of predicting future product backorder is not trivial. Principally, the predictive model needs a sufficient amount of training samples to learn the patterns by which the model decides whether a product will be backorder or not. But in reality, the number of samples where the product is being backordered is extremely lower than the non-backorder products. That leads the dataset to be an imbalanced one.

However, only one/two studies available that focus on product backorder with class imbalance problems (Hajek and Abedin 2020; de Santis, de Aguiar, and Goliatt 2017). In contrast, extensive work has been incorporated in the past to optimizing inventory

CONTACT Author1 Author. Email: author1 email

management. Inventory manager faces various challenges due to material shortages that may completely be backlogged or completely lost. Moreover, earlier literature classifies material backordering phenomenon as a fixed backorder, partial backorder and time weighted-backorder (Srivastav and Agrawal 2016). In fact, on account of good manners along with the reputation of the supplier, some customers are ready to wait until replenishment, especially if the backorder placement and waiting duration are short, while others are more impatient and go elsewhere. This situation points out the supplier has to forgo sales order and miss the chance to earn more revenue. It creates customers' dissatisfaction and may generate doubts in customers' minds about the nature of the storage capability of the supplier.

Based on the above significance, this study offers a comprehensive and novel methodological framework to predict inventory backorder that mitigates lost of sales order, establish the supplier-customer relationship, minimize product backordering costs, and eventually optimize the enterprises' revenue. However, predictive models based on deep neural networks currently have a significant impact on business intelligence (Heaton, Polson, and Witte 2017; Kraus, Feuerriegel, and Oztekin 2020; Evermann, Rehse, and Fettke 2016). An efficient predictive model can serve both the customer and company by predicting the future backorder of any specific product. Therefore, in this paper, we propose a new deep neural network-based predictive model by handling data imbalance to predict the product backorder.

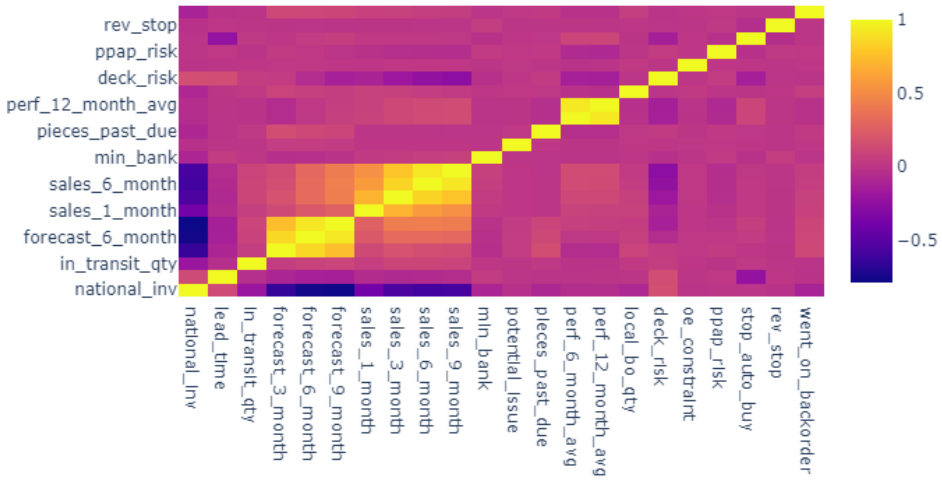


Figure 1. Heatmap

In this research, we used the Kaggle dataset which contains 8 weeks of data about the inventory management system. The dataset is highly imbalanced. The ratio of non-backordered and backordered products is 137:1. The heatmap of the dataset is illustrated in Fig. 1. The figure indicates that there is no correlation between the features. This makes the dataset more challenging. Handling the imbalanced dataset for any predictive model is a challenging task (Li 2017). Hence, these challenges make the task more formidable. However, there are some classical machine learning techniques have already been proposed including gradient tree boosting, random forest and logis-

tic regression (de Santis, de Aguiar, and Goliatt 2017). They applied undersampling and oversampling techniques to balance the dataset (Kang and Ramamohanarao 2014; García et al. 2018).

The methodological flowchart follows the following steps. First, we preprocess the features of each product. The tasks of preprocessing include converting the qualitative data into quantitative, handling missing values and normalization to scale each feature value in a certain range. We noted earlier that the number of positive samples in this particular classification task is extremely smaller than the negative samples. That is why, we need to handle the class imbalance problem. To make the dataset balanced, we make use of an oversampling technique that randomly duplicates the minority examples. We also applied under-sampling for the same purpose. Since randomized oversampling duplicates the minority examples, that might affect the predictive models leading toward under-fitting. On the other hand, under-sampling may remove some important samples which are key for training the models. Therefore, we propose the combination of oversampling and under-sampling technique to equalize the majority and minority class examples. The SMOTE oversampling approach is introduced to artificially create minority class examples rather than random duplication. We applied SMOTE oversampling for balancing the data. Hence, this technique balances the dataset, as well as the synthetic minority class examples contribute the predictive model as like as real examples. Finally, we proposed a densely connected deep neural network (DNN) model that learns from the training examples and eventually can predict the product backorder. Accordingly, four different approaches have been compiled with data sampling approaches such as "*Weighted_DNN*," for which the weight for each class is determined; "*Ran_Over_DNN*," that is, the randomized oversampling technique is applied to generate minority class instances; "*SMOTE_Over_DNN*," where the synthetic minority oversampling technique is applied to generate artificial class example; "*Com_SMOTE_Under_DNN*," for which the combination of oversampling and undersampling is applied.

We carried out experiments applying our methods on a standard product backorder prediction dataset named "*Can you predict product backorder*". The experimental results in terms of area under ROC curve (AUC) demonstrated that our method achieved a new state-of-the-art performance in backorder prediction. We also compared the performance of our method with some prominent classification models. The comparison concludes that our method significantly outperformed the other known related models. The contributions of this paper are:

(1) This study proposed a theoretical framework to predict inventory backorder occurrences.
(2) We offered an inclusive data sampling technique, specifically; this study employed the combination of SMOTE oversampling and under-sampling to make the dataset balanced over a rare event domain of supply chain management.
(3) We proposed a DNN-based method to predict the product backorder that enhances the overall efficiency of the suppliers.
(4) The novel methodology offers a significant enhancement in terms of prediction accuracy that minimizes misclassification errors and eventually boosts up enterprises profitability.
(5) The proposed methodology is also homogeneously applicable in other domain such as credit card default prediction.

The structure of this paper is organized as the following: Section 2 presents the discussion about some prominent related works in the business domain. Our proposed

deep neural network-based backorder prediction model handling class imbalance is presented in section 3. To validate the performance of our method, we present the experimental results and performance comparison in section 4. Finally section 5 concludes the paper with some future directions.

2. Related Work

Predicting the material backorder might be beneficial for both the company and the consumers (de Santis, de Aguiar, and Goliatt 2017). Mart et al. proposed a Markov decision process-based system to decide the tactical inventory and backorder (Mart, Duran, and Bakal 2013). They consider the manufacturing systems as a stochastic process and predict the yield. There is another study that considers the periodic-review based stock inventory system with a partial backorder prediction (Xu, Bisi, and Dada 2017). To estimate the uncertainty in the inventory system, a framework has been proposed considering the demand distribution and parameter. They modeled the estimation errors and obtained a predictive lead time demand distribution that substituted into the inventory model (Prak and Teunter 2019). A reinforcement model is studied to decide the ordering policies in the inventory systems (Chaharsooghi, Heydari, and Zegordi 2008). Firstly they consider supply chain management as a multi-agent system and introduce the reinforcement model. Then they employed the Q-learning algorithm to solve the model. An objective function comprising of the sum of ordering, holding and product backordering cost in one-warehouse and N-retailer problem (Abdul-Jalbar, Gutiérrez, and Sicilia 2009).

Björk construct the economic order quantity problem as an optimization problem and introduced different models capturing the quantity with backorders. To handle the uncertainties of demand and lead time, their fuzzy number based optimization model has performed well (Björk 2009). Another study proposed a fuzzy model to incorporate human learning with backorders (Kazemi et al. 2015; Srivastav and Agrawal 2016). Researchers also developed an economic order quantity model with a special sale price, imperfect quality, quantity discounts and partial backordering (Taleizadeh et al. 2012; Lin 2010). An improved way to estimate imperfect items and managing backorder is studied introducing an integrated inventory model with a distribution-free approach for lead time demand to simultaneously optimize lot size, safety factor, the number of shipments, and lead time (Kim et al. 2018).

Kazemi and Jabel considered an inventory model with backorders in a fuzzy situation by exploiting two types of fuzzy numbers, which are trapezoidal and triangular (Kazemi, Ehsani, and Jaber 2010). They fuzzified the input parameters and the decision variables in their full-fuzzy model. Kuhn-Tucker conditions were applied to determine the optimal policy for the developed model with a graded mean integration model. The study is also proposed on decisions on spare parts allocation for the repairable isolated systems with dependent backorders (Guo et al. 2019). Feng et al. proposed a method to predict the demand for LRU parts with the backorder (Feng et al. 2012).

To develop the alternative backorder replenishment plan, a framework has been proposed to minimize the total cost and expected risk cost (Shin et al. 2012). They introduced a Bayesian Belief Network that modeled the relationship between risk and risk propagation. Wang and Tang studied a dynamic rationing policy that considered the dynamic properties of demand (Wang and Tang 2014). They developed a Markov decision model that obtained the optimal dynamic rationing levels for the multiple

demand class (Bao et al. 2018). Different classical machine learning-based techniques including Gradient tree boosting, logistic regression and the random forest has been also proposed for predicting the backorder of materials (de Santis, de Aguiar, and Goliatt 2017). However, in this research, we propose a DNN-based classification model that can learn from the existing training samples and can eventually predict the backorder.

3. DNN Based Backorder Prediction

Deep Neural Network is being used in many different areas as a classifier. In this research, we propose a deep neural network-based classification technique that predicts whether a particular product order would be canceled or not. We have noted earlier that product backorder is rare as compared to some other cases in inventory and supply chain management. Therefore we need to handle the class imbalance problems here in this research. Fig. 2 illustrates the major steps involved in training and testing of our proposed model to predict product backorder. First, we apply different preprocessing techniques to filter out noisy data, handle missing values, scale and normalize the data values in a certain range. To get ride of class imbalance problem, we then employ different prominent approaches that include class weight calculation, randomized oversampling, SMOTE oversampling, etc. Finally, we leverage our deep neural networks model to train the predictive model.

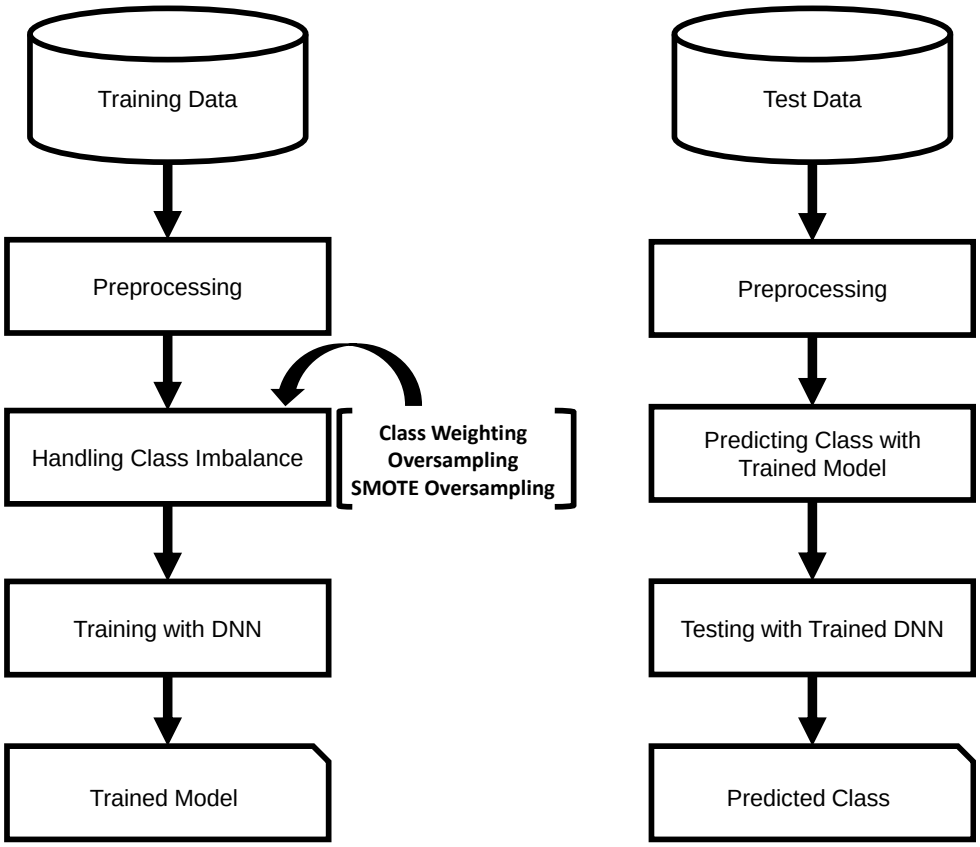


Figure 2. Overview diagram of backorder prediction applying DNN handling data imbalance

3.1. *Preprocessing*

For a given product, there are numerous types of attributes including transit time, transit quantity, forecast sales, the minimum amount in stock, risk factors, etc. The values widely vary across different attributes. Moreover, some attributes are quantitative and the rest of them are categorical. Therefore we need to convert categorical attributes into quantitative ones. For example, some attributes contain the categorical values either YES or NO. We convert them into binary values correspondingly. The missing values are also a common problem in the dataset. For handling missing values, we exploit the median computed using other samples' attributes to select as attributes value for missing one. The values are also being normalized using the MinMax normalization to scale the value on a particular range as the following:

$$\bar{x} = \frac{x' - min(x)}{max(x) - min(x)}$$

where $\bar{x}$ is the normalized value, and x is an original attribute's value.

3.2. *Handling Class Imbalance*

Maximum standard classification algorithms are designed to train using class balanced dataset. But the imbalance dataset contains very few example objects of minority class which make the classification task difficult. Therefore, learning from imbalanced data makes a common problem in supervised classification. Different existing techniques might be applied to balance the dataset. The techniques may include class weight boosting, oversampling, undersampling, etc. Oversampling techniques are utilized to generate some artificial data to achieve balance in class distribution. This is one of the key strategies that balance the class distribution by assigning artificial values for each attribute of an object. To achieve balanced class distribution, we exploit some prominent oversampling techniques including randomized oversampling, class weight boosting and SMOTE oversampling (Chawla et al. 2002; Douzas and Bacao 2017; Last, Douzas, and Bacao 2017). SMOTE considers the oversampling as well as undersampling at the same time to balance the dataset.

3.2.1. Class Weight Boosting

Generally, the classification techniques applied a similar weight to each training sample. But in some cases, we may need to employ different class weight considering the class importance. In our case, product backorder is a rare example in reality. Therefore, identifying the product backorder utilizing the general classifier is not enough. We need to emphasize one class, i.e need greater weight values. The dataset naturally has some training examples that are easier to classify than others (Sun, Kamel, and Wang 2006). Those examples may contribute to the classifier to achieve 99% accuracy. On the contrary, some other challenging examples might still exhibit poor performance. Therefore, those easily classified examples are continuously contributing to increase the classifier loss. To mitigate this issue, we introduced weight boosting to minority class examples for balancing the dataset (Sun, Kamel, and Wang 2006) and applied it to train our model for identifying product backorder.

3.2.2. Minority Class Oversampling

Oversampling is a way to increase the examples of minority class by artificially generating some attributes values (Cao et al. 2013; Nguyen, Cooper, and Kamei 2009; Tao et al. 2019). To balance the dataset, we randomly copy the minority class examples to achieve class balance. We continue the same process to create artificial minority class examples until the size of the minority class will be equal to the majority class.

3.2.3. SMOTE Oversampling

The cost of misclassifying an abnormal object as a normal object is generally higher than the cost of reverse error. Oversampling minority class randomly might increase the loss of classifier as they just copy some minority class examples to achieve the balance in class distribution. Product backorder is a highly rare example in supply chain management. Hence, we introduce SMOTE (**S**ynthetic **M**inority **O**ver-sampling **TE**chnique) to create minority class examples. SMOTE is an oversampling technique that creates synthetic minority class examples artificially rather than oversampling with replacement (Douzas and Bacao 2017; Last, Douzas, and Bacao 2017; Chawla et al. 2002). SMOTE chooses the minority class example for oversampling by taking the synthetic examples along with the line segments. First, it chooses k minority class nearest neighbors and then joins the line with all k minority class. Depending on the required number of over-sampled examples, k nearest minority class examples are selected. To generate and select synthetic minority class examples, we first take the difference between the feature vector of the sample under consideration and its nearest neighbor. Then the difference is multiplied by a randomly generated number in the range [0, 1]. Then it added that to the feature vector under consideration. Hence, this process causes the selection of a random point along the line segment between two specific features. In conclusion, this approach forces the decision region of the minority class to become more effective.

The number of synthetic examples is decided by the percentage of oversampling example. Let's consider the oversampling percentage is 100%, then every time a new synthetic example will be created for each instance. That means the size of the minority class will be double. For a given percentage of minority class oversampling n such that $n > 100$, the steps of SMOTE are given as followings:

(1) For each minority class example $\boldsymbol{X}$, k number of nearest neighbor examples are selected. The examples belong to the minority class and the value of k is selected such that $k = ceil\left[\frac{n}{100}\right]$. We applied Euclidean distance function to chose k nearest neighbors. Let $\boldsymbol{X} = [x_1, x_2, x_3, \cdots, x_n]^T$ and $\boldsymbol{Y} = [y_1, y_2, y_3, \cdots, y_n]^T$ be the feature vectors for minority class example and one of the k nearest neighbors, respectively, the Euclidean distance D is computed as follows:

$$D(\boldsymbol{X}, \boldsymbol{Y}) = \left[\sum_{i=1}^{n} (x_i - y_i)^2\right]^{\frac{1}{2}}$$

(2) The difference among the considered minority class example and the nearest k neighbor example is calculated using their corresponding feature vectors. Hence, we have k number of feature vectors containing the difference between the considered example and k nearest neighbors.

(3) Then, we multiply each element of k difference vectors by a random real number

ranges in [0,1].

(4) Finally, all k difference vectors after multiplication is added as the examples of the minority class.

3.2.4. Combination of Oversampling and Under-sampling

Oversampling only minority class examples might affect the predictive model leading towards negative direction since the size of the minority class examples is extremely lower. On the other hand, only the under-sampling may remove some important majority examples. Therefore, we introduced the combination of SMOTE oversampling and under-sampling to make the dataset balanced.

3.3. *Training with DNN*

Inspired by the success of deep learning in every computational area, we proposed a deep neural network (DNN) based classification based model for backorder prediction. The general architecture of our DNN model can be represented as follows:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}\left[\mathbf{a}^{(l+1)}\left(\mathbf{h}^{(l)}\left(\mathbf{a}^{(l)}\left(\ldots\left(\mathbf{h}^{(2)}\left(\mathbf{a}^{(2)}\left(\mathbf{h}^{(1)}\left(\mathbf{a}^{(1)}(\mathbf{x})\right)\right)\right)\right)\right)\right)\right)\right]$$

where $\mathbf{x}$ be the vector that represents product with the dimension n. There are n different attributes of a specific product. $\mathbf{a}^{(l)}$ denotes the activation function of l-th hidden layer $\mathbf{h}^{(l)}$.

Our proposed DNN model consists of four fully connected hidden layers with different numbers of neurons and one dropout layer. Since ReLu is successfully used the activation function for many deep learning applications, we applied it to the hidden layers. The definition of ReLu is given as:

$$f(x) = max(0, x) \rightarrow \frac{df}{dx} = 1 \; if \; x >= 0 \; else \; 0$$

One of the most important properties of ReLu is that this is non-saturated of its gradient that significantly accelerates the convergence of stochastic gradient descent compared to other activation functions (Krizhevsky, Sutskever, and Hinton 2012). In turn, the computational overhead will also be lower. Moreover, ReLu is sparsely activated because it will be zero for all negative input and it is likely for any given unit to not activate at all. Since our deep learning model designed for binary classification, we introduce *sigmoid* activation function in the output layer. The *sigmoid* activation function can be defined as followings:

$$f(x) = \frac{e^x}{e^x + 1}$$

The overall architecture of our proposed fully connected deep neural networks-based model is summarized in Table 1. We employ binary cross-entropy loss function as the model loss and applied different commonly used metrics including AUC.

The summary of our proposed method is presented in Algorithm 1. Com_SMOTE_Under_DNN($\mathbf{X}_{train}, \mathbf{Y}_{train}, \mathbf{X}_{test}$) algorithm exploits a fully connected deep neural network for training the model by handling the data imbalance problem

Table 1. The summary of proposed DNN model.

SL.	Layer	In/Out dim.	Activation Function
1	Dense Input	22, 22	Relu
2	Dense Hidden	22, 16	Relu
3	Dense Hidden	16, 32	Relu
4	Dense Hidden	32, 64	Relu
5	Dense Hidden	64, 128	Relu
6	Dense Dropout (0.5)	128, 128	—
7	Dense Output	128, 1	Sigmoid

Algorithm 1: Com_SMOTE_Under_DNN$(\mathbf{X}_{train}, \mathbf{Y}_{train}, \mathbf{X}_{test})$

Input: A set of training and testing examples with attributes' value, $\mathbf{X}_{train}$, $\mathbf{X}_{test}$ respectively, and the training labels, $\mathbf{Y}_{train}$
Output: $\mathbf{Y}_{test}$, the backorder predictions for $\mathbf{X}_{test}$

for *each* $\mathbf{x}_i \in \mathbf{X}_{train}$ **do**
  // Preprocessing (Sec. 3.1)
  for *each* $fv \in \mathbf{x}_i$ **do**
    if $isCategorical(fv)$ **then**
      $fv \leftarrow toQuantitative(fv)$ // Converting categorical value to quantitative
    end
    if $isMissing(fv)$ **then**
      $fv \leftarrow$ applyMedian(fv) // Appling median for missing value
    end
    $fv \leftarrow doNormalization(fv)$ // MinMax normalization
  end
end
$\mathbf{X}_{train}, \mathbf{X}_{val}, \mathbf{Y}_{train}, \mathbf{Y}_{val} \leftarrow Split\_train\_val(\mathbf{X}_{train}, \mathbf{Y}_{train}, ratio)$
$\mathbf{X}_{train}, \mathbf{Y}_{train} \leftarrow SMOTE(\mathbf{X}_{train}, \mathbf{Y}_{train}, \delta)$ // Oversampling (Sec. 3.2.4)
$\mathbf{X}_{train}, \mathbf{Y}_{train} \leftarrow underSampling(\mathbf{X}_{train}, \mathbf{Y}_{train}, \gamma)$ // Oversampling (Sec. 3.2.4)
$trained\_model \leftarrow applyDNN(\mathbf{X}_{train}, \mathbf{X}_{val}, \mathbf{Y}_{train}, \mathbf{Y}_{val})$ // Oversampling (Sec. 3.3)
// Our proposed DNN model
$\mathbf{Y}_{test} \leftarrow trainned\_moodel(\mathbf{X}_{test})$

with the combination of SMOTE oversampling and under-sampling. Then the trained model is employed to predict the product backorder. Each statement represents the workflow of our proposed backorder prediction method.

4. Experimental Results and Evaluation

4.1. *Dataset Collection*

The effectiveness of our proposed DNN-based method has been tested conducting a wide range of experiments on a benchmark dataset named "*Can You Predict Product Backorder*[1]". We collect the dataset from Kaggle. The interested reader may find it here (https://github.com/rodrigosantis1/backorder_prediction). The dataset consists of a large number of information. Each order has a set of 22 attributes that describes

[1]https://github.com/rodrigosantis1/backorder_prediction

the history of the past eight weeks. The attributes are summarized in Table 2.

Table 2. The description of the attributes of an individual order

Attribute	Notation	Description
Inventory Level	x_1	Current inventory level of ordered component
Transit Time	x_2	Amount of registered time for product transit
Quantity in Transit	x_3	Amount of product in transit
Sales Forecast	$x_{4,5,6}$	Sales forecast in upcoming 3, 6 and 9 months
Future Sales	$x_{7,8,9,10}$	Amount of sales quantity before 1, 3, 6 and 9 months
Stock Amount	x_{11}	Minimum amount of product recommended in stock
Parts Overdue	x_{12}	Amount of parts overdue from sources
Source Performance	$x_{13,14}$	Performance of product sources
Stock Order Overdue	x_{15}	Amount of stock orders that overdue
General Risk	x_{16-21}	Some risk flags
Went on Backorder	y	Final decision whether the product went on backorder or not

As we noted earlier that product backorder is not a common scenario, the class distribution of this dataset is not also balanced. Out of 1,929,936 examples, only 13,981 product orders went backorder. On the contrary, the number of negative examples is 1,915,954 which far larger than the positive example. The Fig. 3 illustrates the class distribution of this dataset. The imbalance ratio for this particular dataset is 1:137 that indicates that this is a highly imbalanced dataset.

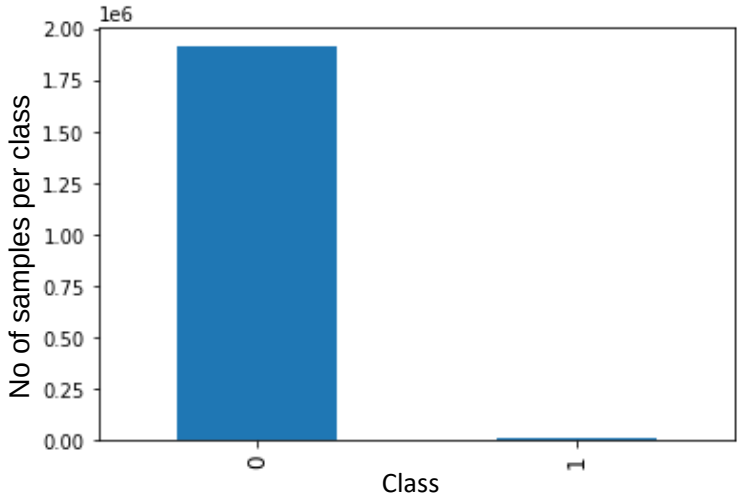


Figure 3. Class distribution

4.2. *Evaluation Metrics*

Generally, the performance of any classifier in binary classification task is estimated by *Accuracy*, *Precision* and *Recall*. These standard evaluation metrics are calculated

by utilizing the confusion matrix illustrated by the **Table** 3.

Table 3. Confusion Matrix

	Predicted Negative	**Predicted Positive**
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

In short the summary of the confusion matrix can be explained as the followings:

- **True Negative (TN)**: The number negative samples correctly classified as negative.
- **False Positive (FP)**: The number of negative samples incorrectly classified as positive.
- **False Negative (FN)**: The number of positive samples incorrectly classified as negative.
- **True Positive (TP)**: The number of positive samples correctly classified as positive.

The most generally used evaluation metrics to measure the performance of any classification algorithm is *accuracy*. The accuracy, *Acc* can be defined as follows:

$$Acc = \frac{TP + TN}{TP + FN + FP + FN}$$

In a balanced dataset, *Acc* can be used as a prominent metric and at the same time, the $loss = 1 - Acc$ can also be used as a standard metric. But in case of measuring the performance of any classifier on class imbalanced datasets with unequal loss, only the *accuracy* is not enough to indicate the efficiency (Chawla et al. 2002). Moreover, accuracy does not make any difference between the number of correctly classified examples of different classes. In the presence of class imbalance, Receiver Operating Characteristics (ROC) can be a better metric to estimate the efficiency of a classifier (Chawla et al. 2002).

However, the other two metrics also utilized as good measures for performance estimation. The Precision, P considers the accuracy of the classifier when it predicts the positive samples. The Precision P can be defined as

$$P = \frac{TP}{TP + FP}$$

One the other hand, the recall is estimated considering the total number of positive samples. Recall R indicates the ability to find positive examples of a classifier. It can also be called as sensitivity or true positive rate. The definition is given as

$$R = \frac{TP}{TP + FN}$$

The false-positive rate is also a widely used evaluation metric for any classification technique. The false-positive rate, F can be defined as

$$F = \frac{FP}{FP + TN}$$

The curve employing precision and recall can also be used to check the efficiency of a classifier. It can also be used to visualize for selecting the decision function. However, the performance of any classification technique on the class imbalanced dataset is widely estimated by Area Under the ROC Curve (AUC) (Chawla et al. 2002). AUC is calculated based on the ROC curve. As we noted earlier, the ROC curve can be a better way to show the effectiveness of a classifier in predicting samples on class imbalance datasets. , the ROC can visualize the trade-off between the precision and false positive rate, similar to the precision-recall curve. ROC curve indicates that by increasing the false positive samples, any classification techniques can increase the true positive samples. AUC can be defined as follows:

$$AUC = \frac{1 + P - F}{2}$$

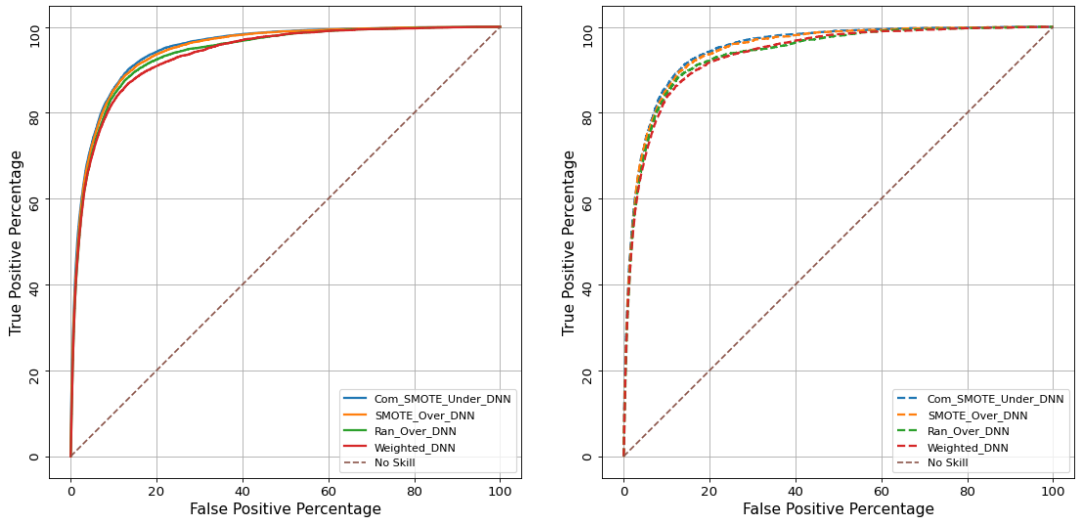
AUC is a standard measure that can singlehandedly indicate the classifiers' performance on the imbalanced dataset. In our experiments, we applied the AUC evaluation metric to validate the performance of our proposed product backorder prediction method.

4.3. *Experimental Setup*

We have conducted a wide range of experiments to test the performance of our proposed method for product backorder prediction considering the imbalance properties of the dataset. First, we carried out an experiment with our proposed DNN-based method applying class weight boosting technique as the dataset is highly imbalanced and this setting is denoted as *Weighted_DNN*. Then in the next experimental setting *Ran_Over_DNN*, the minority class examples were over-sampled by using the randomized oversampling technique. Since the number of minority class examples is very low, randomly duplicating the class examples is not always a better option. Hence, we artificially create synthetic minority class examples to handle class imbalance using the SMOTE technique. This setting is indicated as *SMOTE_Over_DNN*. Finally, the combination of SMOTE oversampling and undersampling is then employed to make the dataset balanced. The final experimental setting is denoted by *Com_SMOTE_Under_DNN*. Table 4 presents a summary of all experimental settings.

Table 4. Summary of experimental settings

Experimental Setting	Description
Weighted_DNN	The weight for each individual class is calculate for the data imbalance problem. Then DNN is applied to train and predict the backorder prediction.
Ran_Over_DNN	Randomized oversampling technique is applied to generate minority class example.
SMOTE_Over_DNN	Synthetic minority oversampling technique is applied to create artificial class example.
Com_SMOTE_Under_DNN	The combination of oversampling and undersampling is used. SMOTE and randomized undersampling techniques are employed for oversampling and undersampling, respectively.



(a) ROC curve (Training)

(b) ROC curve (Testing)

Figure 4. Receiver Operator Characteristic (ROC) curves of our proposed DNN-based model with different variations for training and testing. The X-axis represents the percentage of the false positive and the Y-axis represents the true positive percentage.

4.4. *Experimental Results*

The performance of our proposed deep neural network based model to predict product backorder handling imbalanced dataset is reported in Table 5. The table indicates the effectiveness of all experimental settings in predicting the backorder.

Table 5. The performance of our method in terms of AUC. The best results is in **bold**.

Method	AUC (train)	AUC (test)
Weighted_DNN	0.9402	0.9350
Ran_Over_DNN	0.9528	0.9475
SMOTE_Over_DNN	0.9624	0.9544
Com_SMOTE_Under_DNN	**0.9648**	**0.9586**

The experimental setting *Weighted_DNN* exploited weight boosting to estimate the class weight of each individual class. We can see that the performance in training and testing achieved reasonable considering the data imbalance. The randomized oversampling technique *Ran_Over_DNN* to make the dataset balanced got better AUC in training and testing as compared to the weight boosting techniques. The oversampling technique utilizing producing artificial synthetic examples of minority class also performed effectively in testing. However, our final experimental setting *Com_SMOTE_Under_DNN* that mitigated the imbalanced problem by combining undersampling majority class examples and oversampling with SMOTE has got a new state-of-the-art result in both training and testing phase.

The Receiver Operator Characteristic (ROC) curves of different variations of our proposed deep neural network-based prediction models for training and testing are depicted in Fig. 4. The curves of training and testing indicate that the combination of undersampling and SMOTE oversampling can deal with the imbalance problem in a better approach. The area under the curve is also reflecting the performance. However,

the other techniques to deal with the data imbalance also achieved effective results in predicting product backorder. In conclusion, we can say that our proposed techniques are efficient in handling imbalance problems in the product backorder dataset and our proposed DNN-based predictive model has been achieved state-of-the-art performance in this domain.

4.5. *Discussion*

4.5.1. Performance Comparison

The comparative performance of our proposed methods with known classical predictive models (de Santis, de Aguiar, and Goliatt 2017) in product backorder is presented in Table 6. The table indicates that our proposed deep learning-based predictive model with the combination of undersampling majority and oversampling minority class examples significantly outperformed the other classical machine learning-based classifiers. In turn, our proposed method has got better performance in training and testing that reflects the consistency in predicting product backorder. However, they (de Santis, de Aguiar, and Goliatt 2017) applied gradient tree boosting in *GBOOST* and blagging which achieved better results among their methods. But the performance of other techniques based on logistic regression (LOGIST), classification tree (CART), random undersampling (RUS), etc. is not good. Therefore, we can say that the experimental results on a benchmark dataset demonstrate that our proposed deep neural network-based models are efficient and can be used as a standard framework in predicting product backorder.

Table 6. The performance comparision among our method and some classical machine learning models (de Santis, de Aguiar, and Goliatt 2017). The best results is in **bold**.

Method	AUC (train)	AUC (test)
Com_SMOTE_Under_DNN	**0.9624**	**0.9586**
LOGIST	0.9196	0.9206
CART	0.9443	0.9381
RUS	0.9412	0.9317
GBOOST	0.9317	0.9482

4.5.2. Robustness Check

To illustrate the applicability of our method, we also applied our proposed DNN-based on *credit card fraud detection*[2] dataset. The data has been collected from Kaggle. This is also a highly imbalanced dataset. It consists of the transaction occurred in September 2019 by European cardholders. In total, there are 284,807 transactions and only 492 transactions were identified as fraud. That means, only 0.17% transactions were detected as fraud Hence, this is a highly imbalanced data. The dataset contains 30 different attributes per transaction. The values of the attributes were transformed by using principal component analysis. They did not provide the original feature due to confidentiality issues.

However, we applied our proposed DNN-based method including all experimental settings to this dataset. The performance is presented in Table 7. The table indicates that the performance of our method is also consistent as like as the backorder

[2]Credit Card Fraud Detection: https://www.kaggle.com/mlg-ulb/creditcardfraud

Table 7. The performance of our proposed method on creadit card fraud ditection.

Method	AUC (train)	AUC (test)
Weighted_DNN	0.9582	0.9252
Ran_Over_DNN	0.9427	0.9215
SMOTE_Over_DNN	0.9657	0.9211
Com_SMOTE_Under_DNN	0.9775	0.9427

prediction task. We can see that, our final experimental setting that combines both the SMOTE oversampling and under-sampling before applying DNN model. In both training and testing, the performance of credit card fraud detection in terms of AUC is satisfactory. Hence, our proposed DNN-based method is effective to learn by handling data imbalance.

5. Conclusion and Future Directions

The product backorder prediction is one of the complex issues of inventory management faced by many industries nowadays. In a supply chain, producer distributes materials and goods through retailers. The products usually produced before the placement of the demand, and then the inventory occurs. As a part of inventory management, the backorder happens when demand exceeds inventory supplies. The producer needs to manage inventory in a feasible way. However, one typical scenario is that the inventory manager usually faces skewed class problem in the product backorder prediction event. That is to say, the non-backorder items extensively outnumber backorder items that the current study fixes up.

Linking to the above backgrounds, this paper proposed a new classifier based on densely connected deep neural networks by handing the data imbalance problem to predict the product backorder. The class imbalanced problem was tackled by introducing some effective techniques including the combination of SMOTE oversampling and under-sampling. To train the model for predicting product backorder, we also proposed a new DNN-based model. However, the experimental results concluded the effectiveness of our proposed method in predicting product backorder. Our method also significantly outperformed some known classical machine learning techniques that presented the efficiency of our method.

The methodological settings and findings of this study will provide significant benefits to the enterprises, corporate stakeholders, and regulatory bodies. Under the conditions of non-stationary prices, constraint resources, and imperfect product processes; the automatic backorder predictions generate future revenue and minimize misclassification costs, which in turn, save a high amount of product losses. Accurate predictions also assist companies in distributing their limited manufacturing items to end-users considering the effects of alteration in the production costs and prices over time. We deem that such scenarios are frequently observed, particularly in new product launches. Our experimental analysis exposes that the forecasting power of deep learning-based inventory backorder systems exceeds when compared to the customary inventory management policies. Nevertheless, it should be noted that the achievement of automatic inventory management depends mostly on the enterprises' ability to predict the prices/costs and production capabilities for the planning horizon. As an extension of this study, in the future, we have a plan to apply the recurrent neural network-based approach for the same purpose. We also have a desire to apply our

proposed method in some other research domains in the supply chain management system.

References

Abdul-Jalbar, Beatriz, José M Gutiérrez, and Joaquín Sicilia. 2009. "A two-echelon inventory/distribution system with power demand pattern and backorders." *International Journal of Production Economics* 122 (2): 519–524.

Bao, Lina, Zhiying Liu, Yimin Yu, and Wei Zhang. 2018. "On the decomposition property for a dynamic inventory rationing problem with multiple demand classes and backorder." *European Journal of Operational Research* 265 (1): 99–106.

Björk, Kaj-Mikael. 2009. "An analytical solution to a fuzzy economic order quantity problem." *International journal of approximate reasoning* 50 (3): 485–493.

Cao, Hong, Xiao-Li Li, David Yew-Kwong Woon, and See-Kiong Ng. 2013. "Integrated oversampling for imbalanced time series classification." *IEEE Transactions on Knowledge and Data Engineering* 25 (12): 2809–2822.

Chaharsooghi, S Kamal, Jafar Heydari, and S Hessameddin Zegordi. 2008. "A reinforcement learning model for supply chain ordering management: An application to the beer game." *Decision Support Systems* 45 (4): 949–959.

Chawla, Nitesh V, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16: 321–357.

de Santis, Rodrigo Barbosa, Eduardo Pestana de Aguiar, and Leonardo Goliatt. 2017. "Predicting material backorders in inventory management using machine learning." In *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 1–6. IEEE.

Douzas, Georgios, and Fernando Bacao. 2017. "Geometric SMOTE: Effective oversampling for imbalanced learning through a geometric extension of SMOTE." *arXiv preprint arXiv:1709.07377* .

Evermann, Joerg, Jana-Rebecca Rehse, and Peter Fettke. 2016. "A deep learning approach for predicting process behaviour at runtime." In *International Conference on Business Process Management*, 327–338. Springer.

Feng, Guo, Liu Chen-Yu, Xu Feng-Lei, and Li Wei-Ling. 2012. "Demand Prediction of LRU Parts with Backorder for SRU." In *2012 Fifth International Symposium on Computational Intelligence and Design*, Vol. 2, 530–532. IEEE.

García, Salvador, Zhong-Liang Zhang, Abdulrahman Altalhi, Saleh Alshomrani, and Francisco Herrera. 2018. "Dynamic ensemble selection for multi-class imbalanced datasets." *Information Sciences* 445: 22–37.

Guo, Linhan, Yu Wang, Dandan Kong, Zhiyuan Zhang, and Yi Yang. 2019. "Decisions on spare parts allocation for repairable isolated system with dependent backorders." *Computers & Industrial Engineering* 127: 8–20.

Hajek, P., and M. Z. Abedin. 2020. "A Profit Function-Maximizing Inventory Backorder Prediction System using Big Data Analytics." *IEEE Access* .

Heaton, JB, NG Polson, and Jan Hendrik Witte. 2017. "Deep learning for finance: deep portfolios." *Applied Stochastic Models in Business and Industry* 33 (1): 3–12.

Kang, Sori, and Kotagiri Ramamohanarao. 2014. "A robust classifier for imbalanced datasets." In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 212–223. Springer.

Kazemi, Nima, Ehsan Ehsani, and Mohamad Y Jaber. 2010. "An inventory model with backorders with fuzzy parameters and decision variables." *International Journal of Approximate Reasoning* 51 (8): 964–972.

Kazemi, Nima, Ehsan Shekarian, Leopoldo Eduardo Cárdenas-Barrón, and Ezutah Udoncy Olugu. 2015. "Incorporating human learning into a fuzzy EOQ inventory model with backorders." *Computers & Industrial Engineering* 87: 540–542.

Kim, Min-Soo, Jong-Soo Kim, Biswajit Sarkar, Mitali Sarkar, and Muhammad Waqas Iqbal.

2018. “An improved way to calculate imperfect items during long-run production in an integrated inventory model with backorders.” *Journal of manufacturing systems* 47: 153–167.

Kraus, Mathias, Stefan Feuerriegel, and Asil Oztekin. 2020. “Deep learning in business analytics and operations research: Models, applications and managerial implications.” *European Journal of Operational Research* 281 (3): 628–641.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. “Imagenet classification with deep convolutional neural networks.” In *Advances in neural information processing systems*, 1097–1105.

Last, Felix, Georgios Douzas, and Fernando Bacao. 2017. “Oversampling for imbalanced learning based on k-means and smote.” *arXiv preprint arXiv:1711.00837* .

Li, Yuqi. 2017. “Backorder Prediction Using Machine Learning For Danish Craft Beer Breweries.” PhD diss., AALBORG UNIVERSITY.

Lin, Tien-Yu. 2010. “An economic order quantity with imperfect quality and quantity discounts.” *Applied Mathematical Modelling* 34 (10): 3158–3165.

Mart, Turgut, Serhan Duran, and İsmail Serdar Bakal. 2013. “Tactical inventory and backorder decisions for systems with predictable production yield.” *International Journal of Production Economics* 143 (2): 294–303.

Nguyen, Hien M, Eric W Cooper, and Katsuari Kamei. 2009. “Borderline over-sampling for imbalanced data classification.” In *Proceedings: Fifth International Workshop on Computational Intelligence & Applications*, Vol. 2009, 24–29. IEEE SMC Hiroshima Chapter.

Prak, Dennis, and Ruud Teunter. 2019. “A general method for addressing forecasting uncertainty in inventory models.” *International Journal of Forecasting* 35 (1): 224–238.

Shin, KwangSup, YongWoo Shin, Ji-Hye Kwon, and Suk-Ho Kang. 2012. “Development of risk based dynamic backorder replenishment planning framework using Bayesian Belief Network.” *Computers & Industrial Engineering* 62 (3): 716–725.

Srivastav, Achin, and Sunil Agrawal. 2016. “Multi-objective optimization of hybrid backorder inventory model.” *Expert systems with applications* 51: 76–84.

Sun, Yanmin, Mohamed S Kamel, and Yang Wang. 2006. “Boosting for learning multiple classes with imbalanced class distribution.” In *Sixth International Conference on Data Mining (ICDM’06)*, 592–602. IEEE.

Taleizadeh, Ata Allah, David W Pentico, Mirbahador Aryanezhad, and Seyed Mohammad Ghoreyshi. 2012. “An economic order quantity model with partial backordering and a special sale price.” *European Journal of Operational Research* 221 (3): 571–583.

Tao, Xinmin, Qing Li, Chao Ren, Wenjie Guo, Chenxi Li, Qing He, Rui Liu, and Junrong Zou. 2019. “Real-value negative selection over-sampling for imbalanced data set learning.” *Expert Systems with Applications* 129: 118–134.

Wang, Daqin, and Ou Tang. 2014. “Dynamic inventory rationing with mixed backorders and lost sales.” *International Journal of Production Economics* 149: 56–67.

Xu, Yanyi, Arnab Bisi, and Maqbool Dada. 2017. “A finite-horizon inventory system with partial backorders and inventory holdback.” *Operations Research Letters* 45 (4): 315–322.