A Product Backorder Predictive Model Using Recurrent Neural Network

LAWAL, S.O[1], AKINTOLA, K.G.[2]

[1] *Centre for Research and Development (CERAD), Federal University of Technology, Akure, Nigeria.*

[2] *Department of Software Engineering, Federal University of Technology, Akure, Nigeria*

Abstract- Increasing demand of products is a common cause of out of product inventory, and the adoption of backordering to satisfy outstanding customer orders after its occurrence cannot be undermined. However, wrong management of backorders incurs several issues such as delay in product delivery, low customer satisfaction, and many more. Therefore, it is necessary to ascertain products with high tendencies of shortage beforehand in order to undertake proactive measures and potentially mitigate both tangible and intangible costs. Hence, this paper proposes a backorder predictive model using recurrent neural network (RNN) on large and imbalanced inventory dataset. The data was pre-processed using Min-Max Scaler, while three data balancing methods (ADASYN, SMOTE, and Random Under Sampling) were applied on the imbalanced data simultaneously and their output were fed into RNNto predict which item goes on backorder. The evaluation of the result obtained showed ADASYN+ RNN had performed better with 0.901 precision, 0.879 recall, and 0.889 F1-Score. The proposed model when compared with other machine learning algorithms shows significant impact on prediction of product backorder.

I. INTRODUCTION

Inventory management is one of the important business processes which ensure that the supply of raw materials and finished goods remain continuous throughout the business operations. It could be during manufacturing or production to ensure smooth operations and organization as it relates to purchases, sales and logistic activities [1]. Inventory management systems has the objective of ensuring smooth running of the production process, reduce the ordering cost of inventory, take advantage of quantity discount, and avoid opportunity loss on sales [2].

Over the years, inventory models have been used in determining the optimum level of inventories that should be maintained in a production process, managing frequency of ordering, deciding on quantity of goods or raw materials to be stored, tracking flow of supply of raw materials and goods to provide uninterrupted service to customers without any delay in delivery.

However, an important component of inventory models is the inventory backorder prediction which identifies products that is about or completely out of stock and prompts organization to make swift request to suppliers to restock [3]. Backordering could be beneficial to business organizations as it protects customers base, ensure responsive supply chain, and robust risk management. Other the other hand, if handled badly creates additional tangible cost such as monetary and effort of procuring, manufacturing, and delivery of products at stipulated time as well as intangible cost such as low customer satisfaction, and shift of customer loyalty to competitors [4]. Hence, solving backorder problems becomes critical in the business processes and the solution to the problem could be more precise backorder prediction.

In recent times, supervised machine learning (ML) techniques are utilized by some companies to predict the out-of-stock products to overcome the associated tangible and intangible costs of backorders [5]. These models encounter imbalanced problem as the number of items which goes on backorder is extremely negligible to the amount of active items. In this study, we proposes the application of a deep learning that combines some sampling methods to help identify patterns and behaviors and predict unusual inventory backorder.

The remaining of the paper is organized as follows: Section II provides review of related works on inventory and backorder predictive models. Section III provides the methodology applied in developing the proposed predictive system, with background of the dataset and the algorithms used in this study. Section IV shows the results and discussions gotten from the methods. Lastly, Section V concludes and includes some future works recommendations.

II. RELATED WORKS

There are extensive number of literatures on inventory management system which has attempted to proffer solutions to different inventory problems including backorder predictions. A few of this literature is discussed as follows:

Anigbogu*et al.* [6] proposed an intelligent model for sales and inventory management using fuzzy logic technique. The method used involved three learning techniques namely: Reordering Learning technique to determine when to order more product to prevent shortages while avoiding overstock, Lead learning technique to determine the time between; when an order is placed and when the physical good is actually delivered, and Quantity Order learning technique to determine the number of products ordered for at a particular point in time. Fuzzy logic was employed in each learning technique to ensure flexibility in the system's decision making. However, system was neither evaluated nor compared with existing systems to ascertain prediction accuracy and efficiency.

Guanghui[7]worked on Demand Forecasting on Supply Chain based on Support Vector Regression (SVR) Method. Weekly sales from a paper company were collected, preprocessed, and fed to SVR model adapted with genetic algorithm as optimization componentto forecast the demand on supply chain. The developed model result was compared with the RBF neural network result as SVR shows smaller results of the relative mean square error and higher forecast accuracy. However, SVR performs poorly on complex and imprecise data.

Boniface*et al.*[8] proposed an Automated Inventory Control System for Nigeria Power Holding Company was proposed to accurately forecast spare parts requirements using significant decision support.The authors employed simulation project life cycle as it incorporates phases such as intelligent phase, managerial phase, Quality Assurance phase, implementation, and operation and management phase. These phases are all integrated to build the proposed inventory control model. However, the proposed system has limited capability of intelligent analysis.

Šustrová [9]presentedan Artificial Neural Network a wholesale Company's Order-Cycle Management to enhance company's ordering system as data were obtained from company sales history. The collected data (29 instances) were preprocessed and split into training and test set before been fed into ANN. Attributes such as current demand, demand in the next 3 months, purchase price, and transport cost were input to ANN. The model was evaluated and the MSE obtained shows that ANN as a suitable predictive method for analyzing sales inventories. However, the developed model recorded a low prediction accuracy which could have been attributed to the limited amount of dataset used.

Madamidola *et al.*[10]presented a Web - Based Intelligent Inventory Management System that provide coordination and monitoring of stores at different locations in an intelligent manner in order to increase productivity. The developed system employed fuzzy logic to provide intelligent reporting of information needed to make decisions for inventory managers. The system was implemented in a distributed manner utilizing client – server model of architecture in a web – based environment.

Santis et al., [2]presented a paper titled "Predicting Material Backorders in Inventory Management using Machine Learning." The motivation of the study was the need to develop an effective predictive model for imbalance material backorder data. The methodology involved using standard but imbalanced backorder dataset obtained from kaggle data repository to build backorder predictive models. The obtained dataset was preprocessed, and sampling techniques such as random under-sampling and Smote methods were applied respectively to create a balanced dataset. Thereafter, the balanced data were split into two sets (training and test set). The training set was used to train three (3) different ensemble algorithms (random forest, gradient tree boosting, and blagging) respectively. Validation on the models were carried out on the test set and evaluated using ROC and Precision-Recall curves. However, gboost ensemble is highly sensitive outliers which could be impractical in real world scenario.

Inprasit and Tanachutiwat[11]presented aReordering Point Determination using Machine Learning for Inventory Management to optimize ordering point for inventory management. The methodology involved the use of a public dataset for the simulation of the model. The data is split into training and test set. Each of the set was preprocessed and significant features were extracted. Five extracted features were fed as input variables into ANN with reordering point as output.

III. METHODOLOGY

The conceptual diagram in Figure 1 shows the workflow of the proposed backorder predictive system. Input data was preprocessed by way of missing values imputation, non-numeric to numeric feature conversion and normalization. Thereafter, the normalized data is split into training and test set. The training set is passed into a data balancing module to ensure equal class distribution and avoid biasness in learning model decisions. The imbalanced training data were subjected sampling as we concurrently fed the data into three sampling techniques namely Synthetic Minority Over-sampling Technique (SMOTE), Random Under Sampling (RUS), and Adaptive Synthetic (ADASYN). We employed RNN as our learning modelto learn from input data and predict product backorders. This choice was due to its automatic and effective processing of features without a need for manual feature engineering. The predictive models were validated on test data and their performances were evaluated.

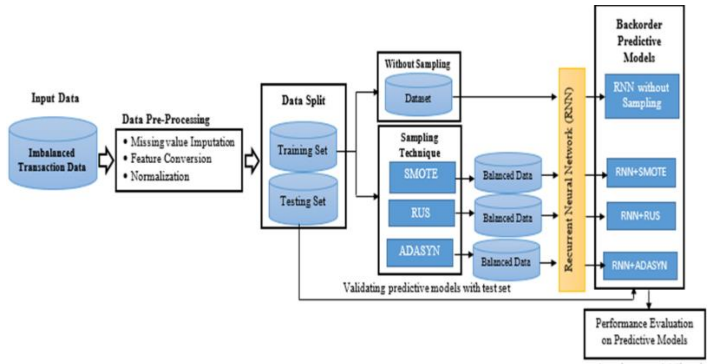

Figure 1 Architecture of Proposed Model.

A. Data Source and Description

This dataset used in this study was obtained from kaggle, an online repository from website "https://www.kaggle.com/mushfique917/transaction-data-with-backorder". It is an historical backorder data used in a competition "Can You Predict Product Backorder?'. It is a collection of 8 weeks inventory information of several products prior to the week to be predicted [2]. The dataset contains highly imbalanced 1,929,935 sample data with 13,981 samples going on backorder and 1,915,954 samples not in the backorder position. Each sample in the dataset is described by 22 features with a label to indicate if a product went on backorder or not. Six (6) of the features are categorical or non- numeric, while the remaining Sixteen (16) features are numeric. The features and their description are depicted in Table 1

Table 1: Attribute Description

S/N	Attributes	Attribute Description
x_1	Sku	Random ID for the product
x_2	national_inv	Current inventory level for the part
x_3	lead_time	Transit time for product
x_4	in_transit_qty	Amount of product in transit from source

x_5	forecast_3_month	Forecast sales for the next 3 month
x_6	forecast_6_month	Forecast sales for the next 6 month
x_7	forecast_9_month	Forecast sales for the next 9 month
x_8	sales_1_month	Sale quantity for prior 1 month
x_9	sales_3_month	Sale quantity for prior 3 month
x_{10}	sales_6_month	Sale quantity for prior 6 month
x_{11}	sales_9_month	Sale quantity for prior 9 month
x_{12}	min_bank	Minimum Recommend amount to stock
x_{13}	potential_issue	Source issue for part identified
x_{14}	pieces_past_due	Part overdue from source
x_{15}	perf_6_month_avg	Source performance for prior 6 month
x_{16}	perf_12_month_avg	Source performance for prior 12 month
x_{17}	local_bo_qty	Part risk flag
x_{18}	deck_risk	Part risk flag
x_{19}	oe_constraint	Part risk flag
x_{20}	ppap_risk	Part risk flag
x_{21}	stop_auto_buy	Part risk flag
x_{22}	rev_stop	Part risk flag
y	went_on_backorder	Product went on back order

B. Pre-processing

Data pre-processing is introduced to transform raw inventory data into an understandable format for prediction. The pre-processing step employed include Handling of missing values, feature conversion, and normalization.

i. Missing value imputation

Handling missing values in backorder inventory data became necessary as the column-wise distribution of missing values conducted on our dataset revealed the occurrence of missing values in ‘Lead Time’ attribute. The missing values were handled using Simple Imputer with median strategy from sklearn. The imputer computes the median of all attribute values under Lead Time column and filled the missing values with the median value.

ii. Feature value conversion

The backorder data contain both categorical and numeric features. However, the estimator used in this study could only work with numbers. Hence, to make the data suitable for our estimator, there is need to transform the categorical feature to numeric features. In view of this, the features were assigned integer values in sequential order. For instance, categorical features with “YES” and “NO” values were assigned “0” and ‘1’ respectively. Table 3 shows the six non-numeric features and conversion process.

Table 2: Feature Conversion

S/N	Categorical features	Feature values and Binary Equivalent
1	deck_risk	NO = 0, YES =1
2	oe_constraint	NO = 0, YES =1
3	ppap_risk	NO = 0, YES =1
4	stop_auto_buy	NO = 0, YES =1
5	rev_stop	NO = 0, YES =1
6	potential_issue	NO = 0, YES =1

iii. Data Normalization

Data were normalized using min-max normalization in order to make training less sensitive to the scale of features. Min-max normalization converts the data into [0,1] distribution using equation (1)

$$v' = \frac{v - min_f}{max_f - min_f} \quad (1)$$

where v' represents the new value, v denotes the observed value (that is, the value to be normalized),

max_f and min_f are maximum and minimum values of feature f respectively. However, not all the attributes were normalized as some attributes contain value within close range of 0 and 1. Hence attribute that needed to be normalized were ['national_inv', 'in_transit_qty', 'forecast_3_month', 'forecast_6_month', 'forecast_9_month', 'min_bank', 'local_bo_qty', 'pieces_past_due', 'sales_1_month', 'sales_3_month', 'sales_6_month', and , 'sales_9_month']

C. Data Split

Data split in this context implies partitioning of the dataset into training and test set. This is an extremely important step in machine learning. The training set is used during model learning process by the classifier, and once training is complete, testing is to be performed to ascertained predictive model performance [12]. However, performing testing will involve splitting the entire dataset into two distinct portions. In this study, our split ratio is set as 70:30 as training set is apportioned 70% of the dataset, while the test takes the remaining 30%. That is, given1,929,935 samples, training set is apportioned 1,350, 954 samples and test set with 578,981 samples.

D. Data imbalance

One of the fundamentalproblems in machine learning is when learning models are trained with imbalanced dataset as it causes impairment in model's decision making by favouring class labels with more instances. The backorder dataset employed in this study is observed to be highly imbalanced (Table 3 shows the imbalance in the dataset.). However, balancing the data will provide the predictive model a fair sense of judgement as we considered three sampling techniques (SMOTE, ADASYN, and RUS).

Table 3: Imbalance Class Distribution

Class Label (Went on Backorder)	Training Set	Training Set Class Distribution (%)	Test Set	Test Set Class Distribution (%)
No	1341250	99.2	574705	99.2
Yes	9704	0.8	4276	0.8
Total	1350954	100	578981	100

i. SMOTE: This is an over-sampling approach in which minority class is over-sampled by creating "synthetic" samples based on the feature space similarities between exiting minority samples, by considering the k-nearest neighborin Euclidean space [2]. Asynthetic sample$s$ as captured in equation (2) is generated by addingminor sample $m$and d, then multiplied by a random number i between [0,1].

$$s = m + d * i \qquad (2)$$

where$d = m - r$ that is, the positive difference between feature vectors of a randomly selected minority neighbor r and minority sample m.

ii. RUS: This is an under-sampling technique which involves randomly selecting a small portion of the majority class without altering the number of the minority class. This implies that in our study, the majority class label "No" was downsized to the amount or number of the minority class 'Yes'. However, in RUS sampling, significant information about the majority class might be lost, which will in turn, affect the performance of the predictive model.

iii. ADASYN: the adaptive synthetic (adasyn) sampling approach is a generalized form of the SMOTE algorithm with the objective to oversample the minority class by generating synthetic instances. Unlike SMOTE which generate arbitrary number of synthetic minority examples to rectify the imbalance in dataset, adasyn algorithm uses weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn [13]. This approach is essential as the data generated by the algorithm will not only ensure a balanced representation of class distribution, but it will also force the learning algorithm to focus on those difficult to learn examples. A look at Table 4 give us an insight of the number the data generate by the three sampling approaches applied in the study.

Table 4: Balanced Class

Class Label	Training Set	SMOTE	RUS	ADASYN
No	1341250	1341250	9704	1341250
Yes	9704	1341250	9704	1340991
Total	1350954	2682500	19408	2682241

E. Recurrent Neural Network (RNN)

In recent times, RNNs have become one of the machine learning models that have gained wide acceptance across different domains. They are special type neural network architecture with recurrentconnections to processsequence data. The recurrent connection performs a repetitive task for every sequence, with the output depending on information obtained from previous computations [14]. Memory of previous time steps is encoded into the RNN's hidden state. The hidden state allows the RNN to retain memory of past information and to learn temporal structure and long-range dependencies in data[15]. If given a backorder data $x_{i=1,2,3,\dots,n}$, RNN computes the hidden vector sequence $h_{i=1,2,3,\dots,n}$ to output prediction$\hat{y}$using equation (3) and (4):

$$h_i = f\,(w_h\, h_{i-1} + w_x x_t \;+\; b_h) \qquad (3)$$

$$\hat{y} = g(w_y h_i + b_y) \qquad (4)$$

Where w_h is the hidden weight matrix, w_x is the input weight matrix, w_y is the output weight matrix, and b_y and b_h represent the biases. The function f represents Rectified Linear Unit (ReLU) called activation function which accepts the previous state h_{i-1} and input x_i to output the current hidden state h_i. The current hidden state h_i will have to be passed to the same function f when reading a new input x_{i+1} to output the new state h_{i+1} and so on. The function g represents the classification function called the softmax function which is used to determine the target class for the given inputs.

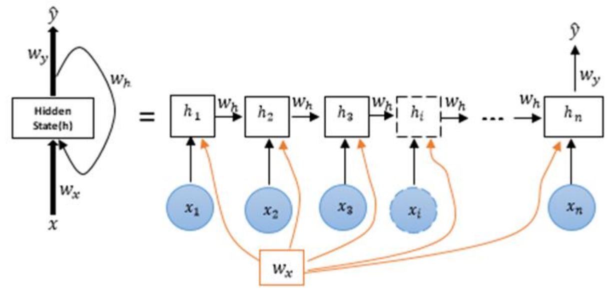

Figure 2: RNN architecture

Training the RNN network is done via Backward Propagation Through Time (BPTT) as it aims to minimize prediction errors. The loss function used in this study is the binary cross entropy function captured in equation (5) to compute the loss between the actual class and the predicted class.

$$\text{Loss} \;=\; -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_t) + (1 - y_i)\log(1 - \hat{y}_t) \qquad (5)$$

wherey_i represents the actual class, $\hat{y}_t$ is the predicted output, and N is the number of class label.

F. Performance Metrics

Evaluating learning models with imbalance problems as such in this study requires specific metrics other than the classification accuracy due to its inability to provide comprehensive assessment of the observed learning algorithm. Instead, we employed recall, precision, andArea Under Curve (AUC) performance metrics. These metrics were formulated from the counts of correctly classified and misclassified instances(TP, FP, FN, TN) in the confusion matrix (shown in Table 5).

Table 5: Binary Confusion Matrix Table

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

In this study,TP represents number of correctly predicted items that goes on backorder, FP represents the number of backordered items that was wrongly predicted as non-backorder products, TN represents number of correctly predicted items that are nonbackordered, FN represents the number of non-backorder items that was wrongly predicted as backorder.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (6)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (7)$$

$$\text{False Positive Rate} = \frac{FP}{TN+FP} \quad (8)$$

$$\text{AUC} = \frac{1+PR-FPR}{2} \quad (9)$$

G. Experimental Setup

All the program codes were implemented in Python programing language version 3.7, using libraries such as Scikit-learn 0.23, Keras 2.4.0, and Imbalanced-learn 0.7.0. The programs were implemented on HP EliteBook Folio 9470m with Intel(R) Core (TM) i5-3437U CPU @ 1.90GHz, 2401 Mhz, 2 Core(s), 4 Logical Processor(s), 8.00 GB RAM.In our experiment, three backorder RNN predictive models (SMOTE+RNN, RUS+RNN, and ADASYN + RNN) were developed from SMOTE, RUS, and ADASYN balanced training set respectively. Prior to the development of the models, we applied gridsearch algorithm in obtaining the best parameters (batch size: 1000, epoch: 30, no of layers: 20) in training RNN classifer. Table 6 shows the parameters and the ranges specified. Thereafter, the developed predictive models were validated using the test set and their results were documented.

Table 6: Parameters and Ranges

Hyper-parameter	Ranges
Batch Size	1000, 2000, 3000, 4000
Epoch	10, 20, 30, 40
Number of layers	20, 30, 40, 50

IV. RESULTS AND DISCUSSION

A study of the model's learnability (Fig. 3-5) was captured in terms optimization (training and validation loss). SMOTE+RNN and ADASYN + RNN show sign of overfitting as their plot of validation or test loss decreases to a point and increase again, while RUS+RNN shows to be well fit as both training and validation loss decreases to a point of stability.

Table 7shows the performances of each model in terms of precision, recall, and F1-score on the test set.

It is observed that ADASYN + RNN had the best performance of the three models with a precision of 0.901, recall of 0.879, and F1-score of 0.889 as compared with RUS+RNN's 0.841 (precision), 0.885 (recall), 0.862 (f1-score) and SMOTE+RNN's 0.894 (precision), 0.877 (recall), 0.886(f1-score).

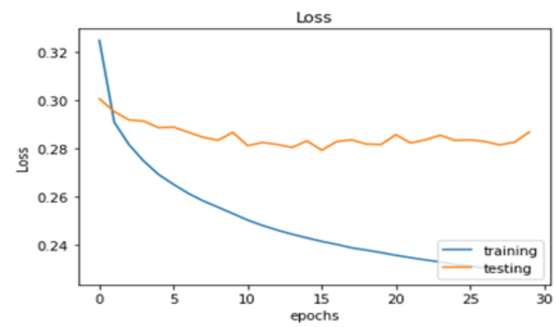

Figure 3: Train andTest loss on Smote + RNN

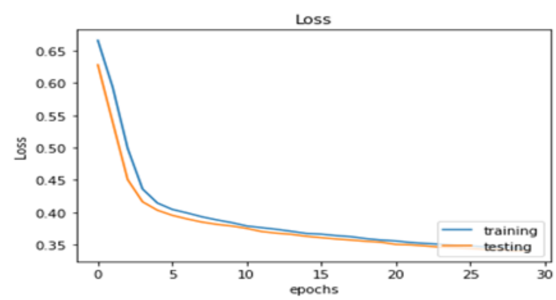

Figure 4: Train and Test loss on RUS + RNN

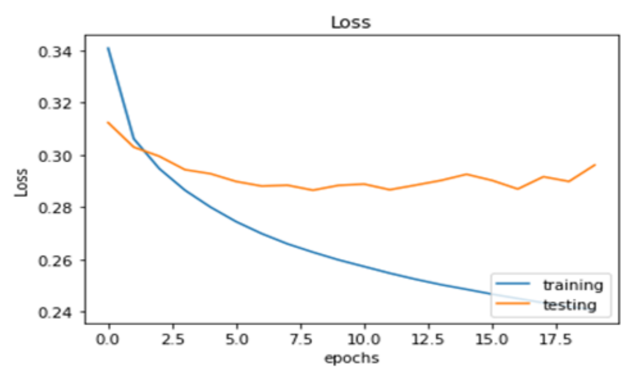

Figure 5: Train and Test loss on Adasyn + RNN

Table 7: Models performance

Model	Precision	Recall	F1-score
ADASYN + RNN	0.901	0.879	0.889
RUS + RNN	0.841	0.885	0.862
SMOTE + RNN	0.894	0.877	0.886

CONCLUSION

In this study, we developed a product backorder predictive model with the capability of identifying items to be backordered using Recurrent Neural Network (RNN). The proposed approach accept input data was pre-processed by way of missing values imputation, non-numeric to numeric feature conversion and normalization, and split into training and test set. The training set is passed into a data balancing module to ensure equal class distribution and avoid biasness in learning model decisions. The imbalanced training data were subjected sampling as we concurrently fed the data into three sampling techniques namely Synthetic Minority Over-sampling Technique (SMOTE), Random Under Sampling (RUS), and Adaptive Synthetic (ADASYN), fed into RNN to predict product backorders. The predictive models were validated on test data and their performances were evaluated. The evaluation of the result obtained showed ADASYN+ RNN had performed better with 0.901 precision, 0.879 recall, and 0.889 F1-Score.

REFERENCES

[1] Oluwaseyi, J. A., Onifade, M. K., and Odeyinka, O. F. (2017). Evaluation of the Role of Inventory Management in Logistics Chain of an Organisation. LOGI – Scientific Journal on Transport and Logistics, 8(2), 1–11

[2] De Santis, R. B., de Aguiar, E. P., and Goliatt, L. (2017). Predicting material backorders in inventory management using machine learning. IEEE Latin American Conference on Computational Intelligence (LA-CCI).1-6

[3] Zhang, R. Q., Wu, Y. L., Fang, W. G., and Zhou, W. H. (2016). Inventory Model with Partial Backordering When Backordered Customers Delay Purchase after Stockout-Restoration. Mathematical Problems in Engineering, pp. 1–16.

[4] Hajek, P., and Abedin, M. Z. (2020). A Profit Function-Maximizing Inventory Backorder Prediction System using Big Data Analytics. IEEE Access, 1–1.

[5] Islam, S., and Amin, S. H. (2020). Prediction of probable backorder scenarios in the supply chain using Distributed Random Forest and Gradient Boosting Machine learning techniques. Journal of Big Data, 7(1).

[6] Anigbogu S. O., Oladipo O.F and Karim U. (2011), An Intelligent Model for Sales and Inventory Management. Indian Journal of Computer Science and Engineering (IJCSE). 2(5):785-791

[7] Guanghui W., (2012). Demand Forecasting of Supply Chain Based on Support Vector Regression Method. International Workshop on Information and Electronics Engineering (IWIEE). Vol 29. Pp. 280-284

[8] Boniface, E., Nwokoye, C. H., Chukwuemeka, A. J. (2013). Automated Inventory Control System for Nigeria Power Holding Company. Department of Computer Science, Nnamdi Azikiwe University, Anambra State, Nigeria. 8(1): 50-60

[9] Tereza Šustrová (2016). A Suitable Artificial Intelligence Model for Inventory Level Optimization. Trendy Ekonomiky a Managementu Trends Economics and Management. 25(1):48-55

[10] Madamidola, O. A., Daramola, O.A., Akintola, K .G. (2017). Web – Based Intelligent Inventory Management System. International Journal of Trend in Scientific Research and Development, Volume 1(4), 2456-6470

[11] Inprasit, T., and Tanachutiwat, S. (2018). Reordering Point Determination Using Machine Learning Technique for Inventory Management. International Conference on Engineering, Applied Sciences, and Technology (ICEAST). Pp. 1- 4.

[12] Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qadir, J., and Hussain, A. (2016). Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study. IEEE Access, 4, 7940–7957.

[13] Haibo He, Yang Bai, Garcia, E. A., and Shutao Li. (2008). *ADASYN:* Adaptive synthetic sampling approach for imbalanced learning. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). Pp. 1322-1328

[14] Chopra, C., Sinha, S., Jaroli, S., Shukla, A., and Maheshwari, S. (2017). Recurrent Neural Networks with Non-Sequential Data to Predict Hospital Readmission of Diabetic Patients. Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics - ICCBB.

[15] Hughes, T. W., Williamson, I. A. D., Minkov, M., and Fan, S. (2019). Wave physics as an analog recurrent neural network. Science Advances, 5(12)