Multidisciplinary Journal of Engineering, Technology and Sciences, Volume 1 Number 1, 2024
Online publication with Google Scholar indexing, Email: mjets85@gmail.com
Title: Design and Implementation of an Intelligent Business to Business Stock Control System Using Machine Learning Technique
Authors: [1]Ifeyinwa Chidum Dimson [2]Onyia Tochukwu Cyprian, [3]Ezekwem Chigozie ThankGod

Design and Implementation of an Intelligent Business to Business Stock Control System Using Machine Learning Technique

[1]Ifeyinwa Chidum Dimson [2]Onyia Tochukwu Cyprian, [3]Ezekwem Chigozie ThankGod
[1,2,3]Electronic and Computer Engineering Department, Nnamdi Azikiwe University Awka, Anambra State, Nigeria
Email: [1]ic.obiora-dimson@unizik.edu.ng [2]onyiacypraintochi@gmail.com; [3]ezekwemchigoziethankgod@gmail.com
[2*]Corresponding author: [1]onyiacypraintochi@gmail.com

Abstract
Stock control is a requirement for automatic monitoring of merchandise within a stock database and predicts when it is due for restocking in real-time. To achieve this the methodology used are data collection, data analysis, feature selection, feature transformation, machine learning, and the Internet of Things (IoT) to develop a stock control system. The data used is a 3-year Out of Stock(OOS) records in Shoprite Enugu State shopping mall, was collected and subjected to a series of processing steps. The processed data features were selected with Chi-square and then transformed into a compact feature vector using Principal Component Analysis (PCA) to train a Feed-Forward Neural Network (FFNN) algorithm and generate a model for OOS prediction. Upon achieving this, an IoT algorithm that utilized Simple Mail Transfer Protocol (SMTP) was used to notify the stock admin of the need for restocking of the identified OOS product. The system was implemented using MATLAB and JavaScript programming language. The results of the evaluation process showed that the proposed model recorded tolerable error during the training process with a Mean Square Error (MSE) of 0.17168 and a Regression (R) of 0.96907, which suggested a very good prediction model. To validate the model, a k-fold cross-validation approach was applied, and the results recorded an MSE average of 0.09686, while the R reported 0.97168. Comparative analysis with other state-of-the art algorithms was performed, considering the MSE results of the new and existing OOS prediction models, and the results showed that the new model was among the best three performing models compared. However, the new model, due to its IoT features, was the most reliable as it was capable of notifying the stock admin in real time of the stock status and the need for restocking of products. Experimental validation of the model as a software considering several products which are running out of stock showed the ability of the system to monitor in real time and notify the admin through email on the need to restock the products
Keywords: Stock, shopping mall, FFNN, OOS, SMTP, IoT, Chi-square, PCA

1. Introduction

The advancements in technology and globalization are resulting in new challenges and avenues for business administration, specifically commercial marketing. One of the most discussed challenges in this field is the management of Out of Stock (OOS) products, which Giacona and Chamas (2023) defined as the unavailability of a product upon customer request. Practically, ideal suggestions to address this challenge include buying more products; however, the dynamics and complexities characterized by customer product preference at every given time, the perishable nature of certain products, and the cost implications are some of the constraints that may hinder this suggested opinion. Over the years, many innovative ideas to improve the tracking of OOS products have been presented by researchers (Praveen et al., 2020). Notably among the ideas is the use of Data Mining (DM) techniques. DM is the process of pattern discovery from relationships between features within a large dataset. It involves the analysis of vast amounts of data to discover hidden patterns and trends that can be applied for various purposes (Aljohani, 2023). In the context of this research, DM involves the analysis of stocks within an inventory management system (all the stock items, merchandise, and goods) to extract information that might help with OOS tracking, control, and management (Nobil et al., 2023; Piao et al., 2023; Gonzalez et al., 2023). DM utilizes machine learning (ML) algorithms such as artificial neural network (ANN), support vector machines (SVM), Decision trees (DT), Random Forest (RF), etc. to learn patterns from huge volume of stock data (Chaudhary et al., 2023). For instance, (Hajeb and Banafi, 2022; Kousar et al., 2023; Arindam and Praveen, 2023; Osman et al., 2022) all applied ML algorithms for OOS management and control. While these studies have made significant contributions to optimizing OOS management, there is a need to consider the uncertainties and dynamics of customer purchasing behaviour. The dynamics stem from the product diversity of customer preference, while the uncertainties align with the fact that demand can be influenced by many factors such as government policies, diseases, weather, cash flow, etc. In addition, while these studies have recorded good success with their proposed model for OOS prediction, most of the outcomes were tested in ideal conditions without actual validation in a real-world marketing scenario. Finally, the integration of the Internet of Things (IoT) is also necessary to facilitate real-time notification during stock control. Therefore, this study proposes the design and implementation of an intelligent business-to-business stock control system using data mining and IoT technique. The data mining will employ ML algorithm, while the IoT will be applied for real-time notification of admins about stock levels.

2. Literature Review

Praveen et al., (2020) researched on the use of machine learning technology for inventory management. XGBoost regression model was trained with data of company product stocks to generate a model to forecast product demand. The result of the system implementation presents that the machine learning model had a Root Mean Squared Error (RMSE) score of 0.6778 with week 3 data and RMSE of 0.7015.

Preil and Krapp (2021) presents a Monte Carlo Tree Search (MCTS) based approach for the development of an artificial intelligence-based inventory management system. The research creates both an online and offline model that uses real-time data to inform choices. The study explores, for demonstration purposes, a four-actor supply chain structure with stochastic lead times and demand, akin to the traditional beer game. In order to estimate the values of actions, the MTCS technique used in the study is based on four steps: simulation, expansion, back propagation, and selection. On the other hand, the inventory supply chain is managed using artificial intelligence. It then shows that compared to other previously used AI-based techniques, both the offline and online MCTS models outperform them. The outcome of the study reports that AI-based dynamic order policies provide interesting insights into complex environments and reveal new advantageous patterns of behaviour.

Olomi and Ikegwuru (2024) presented a study on the transformation of warehouse inventory management using artificial intelligence. By using documentary analysis to determine the influence of artificial intelligence on the transformation of warehouse inventory management, this study critically analysed the relationship between artificial intelligence and warehouse inventory management. The study's findings show that AI techniques may be applied to identify an organization's best inventory-related decisions, and that a coordinated AI system can greatly lower the system's overall inventory. The study draws the conclusion that AI theoretically affects warehouse inventory management on the basis of this, and it should be empirically investigated in organisations that employ AI technology in their warehouse operations, particularly in inventory management.

There is need to improve this work integrating Internet of things features, to facilitate remote monitoring of OOS products and ensure that products can be properly managed irrespective of the location o the stock admin.

3. Methodology

The methodology adopted for the development of this study began with data collection of historical OOS report from a large-scale business enterprise. Then, the collected data was analysed through data mining techniques which follows theprocessing, feature selection, feature transformation, before training a neural network algorithm strategy to generate a model for the prediction of OOS. In order to notify the stock manager, an IoT algorithm was developed which utilized email notification approach to inform the stock manager of products which need restocking. The next step involves,the OOS prediction model and the IoT algorithm to be integrated for OOS monitoring, detection and management in MATLAB and JavaScript programming language.

3.1 Data collection

The data used for this work was collected from Shoprite Shopping mall, Enugu State, Nigeria. The data was collected considering OOS report of the enterprise for 3 years, April 2020 till May, 2023. The sample size of data collected is 57 different products across 8 classes which are cosmetics, furniture, baby food, clothes, electronics, snacks, soft drinks, and beverages. The attributes considered for data collection are sale channels, priority order, quantity in stock, unit price, total revenue, unit cost, total cost, total profit, sale value, available stock and stock status. Table 1 presents the data description.

Table 1: Data description of the out-of-stock dataset

Variables	Data type	Data description
Sale channel	Categorical	The channel through which the product was sold
Priority order	Integer	The priority level of the order
Quantity in stock	Integer	The quantity of the product available in stock
Unit price	Integer	The price of one unit of the product
Total price	Integer	The total price of the product(s) sold
Total cost	Integer	The total cost incurred for the product(s)
Total profit	Integer	The total profit generated from the product(s) sold
Sale value	Integer	The value of the sales transaction

Available stock	Integer	The quantity of the product currently available for sale
Stock status	Categorical	The current status of the stock (e.g., In Stock, Out of Stock)

3.2 Data pre-processing

The data processing was performed on the collected OOS records using mean imputation method. The mean imputation computes the mean point of data within a row and then uses the average to replace the missing values (Alotaibi et al. 2023). The aim was to ensure data completeness before processing to other feature engineering processes which holistically improves the data quality.

3.3 Feature selection

Feature selection was applied on the data collected using the Chi-square technique. The aim was to identify the most significant features within the dataset and eliminate any irrelevant ones. The numbers of overall features are 10 as depicted in Table 1, but the selected key features are 8 in number. According to Rufai and Longo (2024), this process is crucial for creating precise and efficient models, especially in datasets with numerous dimensions. The stepwise of the Chi-square is presented as;

Chi-square algorithm (Rufai and Longo, 2024)

1. *Start*
2. *Measure the data dependence between observed and frequency variables*
3. *Identify the categorical values of the data points*
4. *Target variables: Test the target variables*
5. *Compute null hypothesis as P value*
6. *Set threshold to compare P*
7. *Select features which satisfies new P*
8. *Return as selected features*
9. *End*

3.4 Data transformation

Data transformation was applied in this work to convert the selected feature into a compact feature vector. To achieve this, Principal Component Analysis (PCA) was applied as the technique for the process. The PCA converts the data into a format the machine learning algorithm will understand easily, reduces the size of the data while maintaining quality. The algorithm of the PCA was presented as;

PCA algorithm (Sirakov et al., 2024)

1. Start
2. Load data features selected from Chi-square as X
3. For each feature (X), compute mean (X_m) and standard deviation (X_{sd})
4. Compute covariance matrix of data as (C)
5. Perform standardization of the points
6. Compute eigen values (v)and vectors (y) for the covariance
7. Determine components to keep (k) and corresponding vectors
8. Compute new data points
9. End

3.5 Machine learning algorithm

The machine learning algorithm used for the work is the Feed Forward Neural Network (FFNN). FFNN is a multi layered feed forward neural network which is made of interconnected neurons. The neurons have weights and bias, which are optimized to learn feature points and generate model for problem solving.

3.5.1 Mathematical modelling of the FFNN

The neuron is a unit of computation that reads the inputs given, processes the input and gives the output in processed form. To get the output of the Artificial Neuron from the activation function, it computes the weighted sum of the inputs as (Onuoha et al., 2022);

$$v_k = \sum_{i=1}^{N} w_{ki} x_i \tag{1}$$

Where

x_i is the neuron's input from the training dataset.

w_{ki} is the corresponding weight to the input x_i

The neuron's output is obtained by sending the weighted sum v_k as the input of the activation function φwhich resolves the output of the specific neuron. $y_k = \varphi(v_k)$. A step function with threshold t can be used to express a simple activation as;

$$\varphi(x) = \{1\ if\ x\ \geq t;\ 0\ if\ x\ < t \quad (2)$$

However, bias is mostly used instead of a threshold in the network to learn optimal threshold by itself by adding $x_o = 1$ to every neuron in the network. The step activation function for the bias becomes;

$$\varphi(x) = \{1\ if\ x\ \ 0;\ 0\ if\ x\ < 0 \quad (3)$$

For the learning process to speed up and also adaptive learning capacity, multiple neurons are were used as a multi layered network of neurons formed by feeding the output of one neuron to the input of another neuron as shown in Figure 1, with the four outputs modelling the four-difference stock status at various times. The neurons are connected by a link that has a weight which represents the connection strength between each interconnected neurons. w_{ij}^lrepresents the weight for a link between unit j in layer l and unit i in layer $l + 1$. Also b_i^lrepresents the bias of the unit i in layer $l + 1$.

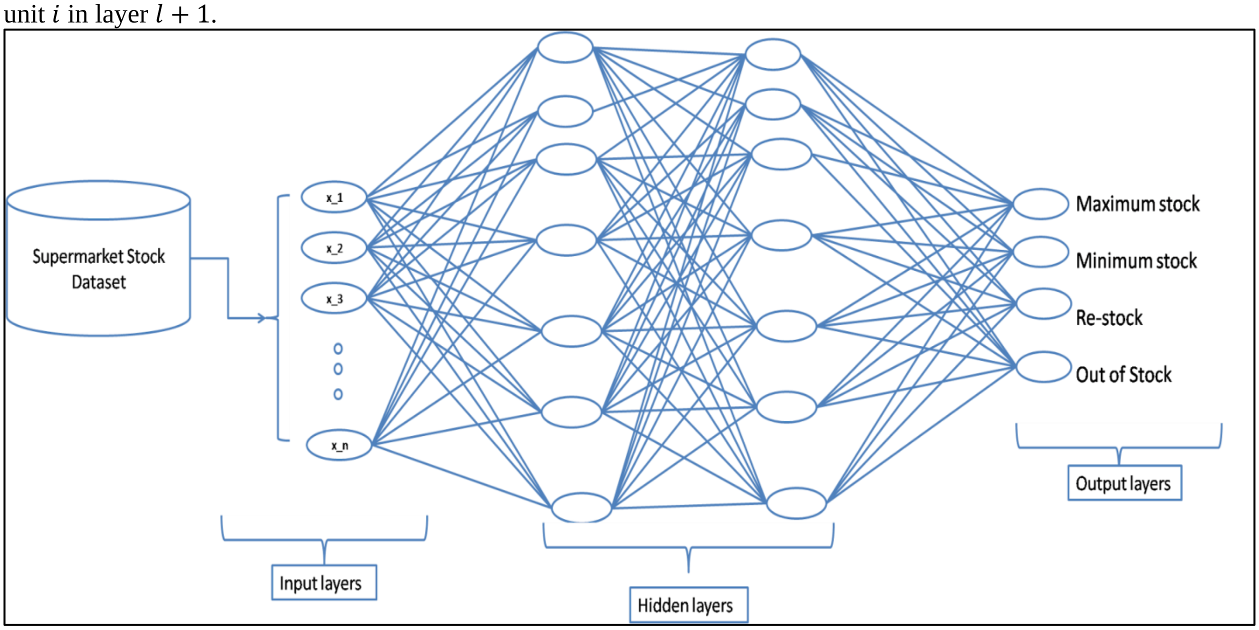


Figure 1: Architecture of the FFNN

For the FFNN modelling as a multi layered neural network, Let the activation of unit i in layer l be represented by a_i^l, then the input for the layer labelled as $L_1$the systemhas$a_i^1 = x_i$ for the ith input of the whole network. Other layers are given by$a_i^l = f(z_i^l)$, where z_i^l is the total weighted sum of the inputs to unit i in layer l in addition to the bias term. The activation function of the ANN with bias is presented as;

$$a_n^2 = f(w_{n1}^1 x_1 + w_{n2}^1 x_2 + w_{n3}^1 x_3 \dots \dots \dots \dots \dots \dots \dots . + b_n^1) \quad (4)$$

Where n is the number of input classes from the dataset of stocks feed to the neural network for training and generation of the four classes as output.

3.5.2 Training of the neural network

To train the FFNN algorithm, the sequence diagram for the process is presented in the Figure 2 using components such as the data, neural network and training algorithm;

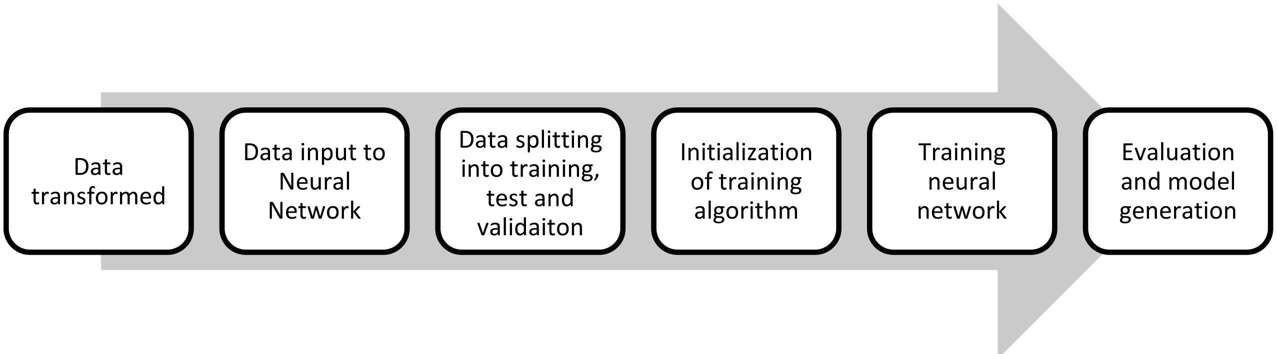


Figure 2: Sequence diagram of the neural network training process

The figure 2 presents the sequence for training of the neural network. First the data transformed with the PCA was imported to the ANN algorithm, which automatically splits the features into training, test and validation sets. The training sets were used to train the network at the initial phase, then the test was applied to evaluate the model and finally the validations set used to validate the results. Overall, the training process utilized the levembarque back-propagation algorithm to optimize the neurons, through the adjustment of the hyper-parameters values, while evaluating the performance until the best version of the model is generated.

3.6 Performance evaluation metrics

The metrics for the evaluation of the model for OOS prediction are regression, mean absolute error, R-square error and accuracy.

1. Regression: It measures how well the model's predictions match the true out-of-stock periods, providing insights into whether the model accurately anticipates when items will be out of stock.
2. R-squared Error (R2): R-squared is a statistical measure that represents the proportion of the variance in the actual out-of-stock periods that is explained by the predicted out-of-stock periods. In OOS prediction, R-squared helps assess how well the model captures the variation in the actual out-of-stock periods. A higher R-squared value suggests that the model's predictions explain a larger portion of the variability in the actual out-of-stock periods.

3.7 The out of stock prediction model

The OOS prediction model is the output of the trained neural network algorithm to learn the features and patterns during OOS and then used to predict future OOS scenarios. The components of the OOS prediction model were presented in the figure 3.

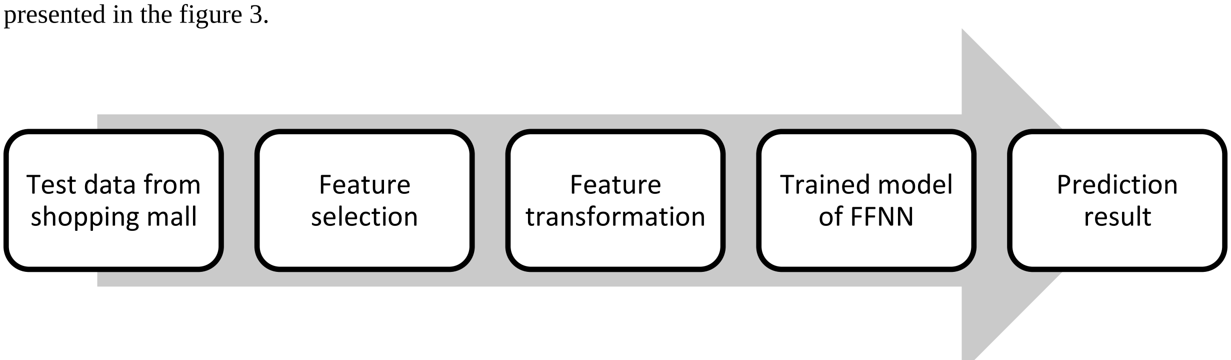


Figure 3: Sequence diagram of the OOS prediction model

The prediction model operates using the input test data from the shopping mall. This data model the daily availability of stocks in the store as the test input. The features of the stock data were selected and then transformed into a compact feature vector using PCA and then feed to the trained FFNN to match the features and predict the OOS stock level.

4. Internet of things for OOS notification

This section presented an IoT algorithm for the notification of stock admin on the stock level at the shopping mall. In realizing this, the use of email notification was applied using the email address of the stock manager to design a custom email which when the neural network trained model predict the OOS in the shopping mall, and then notify the user through mail of the stock level and status. The process used to achieve the IoT notification is presented in the block diagram of figure 4;

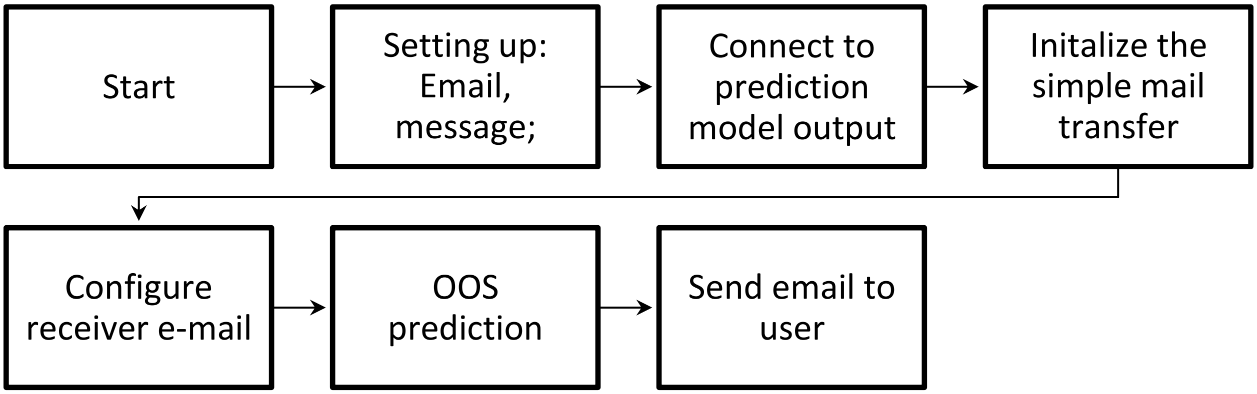


Figure 4: Block diagram of the Internet of things notification system

The figure 4 presented the block diagram of the Industrial Internet of Things (IIoT) notification system. First the email of the message is setup and the frequency at which the mail will be send is programmed. Secondly the trained OOS prediction model identified and then SMTP is initialized to facilitate the real time notification and stock control. Upon prediction of OO stock, the configured email is sent to the admin and notify of the need for restocking. The IIoT algorithm was presented as;

IoT notification algorithm

1. Start
2. Initialize OOS prediction model
3. Initialize email configuration (SMTP server, steverolans@gmail.com)
4. For OOS prediction = True
5. def send_email_notification (recipient, Out of Stock notification
6. Recipient_email = " Enter Recipient email address"
7. Email_subject = "out of stock notification"
8. Email_body = f" This email is a demonstration to prove that {percentage}
9. Send_email_notification(recipient_email, email_subject, email_body)
10. Elseif
11. Return to step (2)
12. End if;
13. End

4.1 Out of Stock Control Model

The OOS control model utilized the prediction model and the notification model to control the issues of OOS within the case study shopping mall. The block diagram of the integrated OOS control model is presented in the figure 5.

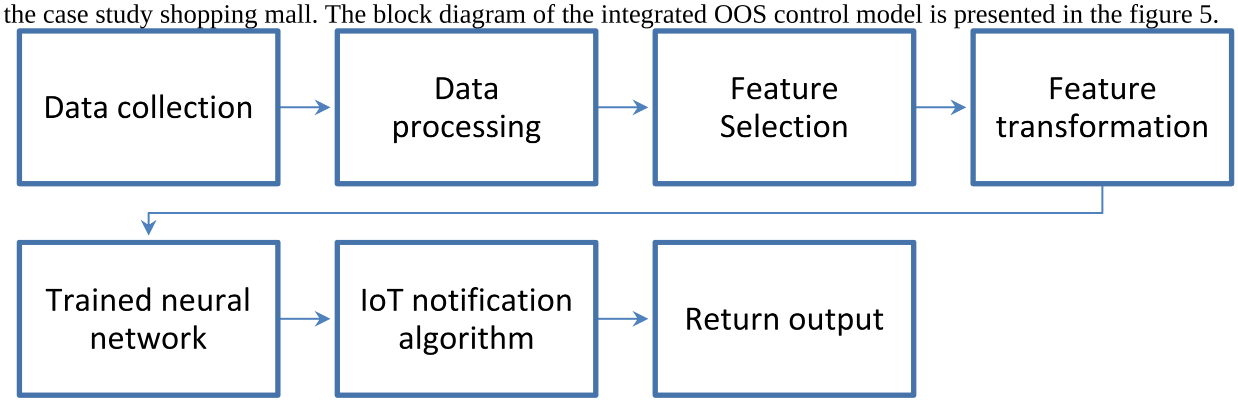


Figure 5: Block diagram of the stock control model

The figure 5 presents the block diagram of the stock control system. First the collected data of the stock is processed and the feature selection with Chi-square and then PCA is applied to transform the features to train neural network and generate a model for OOS prediction. To notify the stock admin of the predicted OOS level, the integrated IoT algorithm was applied to send email notification, reporting the need for restocking.

5. Implementation of the Stock control system

The stock control system implementation involves the utilization of two programming languages: MATLAB and JavaScript. MATLAB, renowned for its robust data processing and machine learning capabilities, serves as the primary tool for handling data and building predictive models. JavaScript, on the other hand, plays a crucial role in integrating the developed models with the IoT infrastructure for real-time stock control. In the initial phase of implementation, the system imports relevant data into MATLAB. This dataset typically comprises historical sales data, inventory levels, and other pertinent information essential for predicting stock shortages. Once the data is imported, pre-processing steps are undertaken to ensure its quality and suitability for analysis. This includes handling missing values through imputation techniques, crucial for maintaining the integrity of subsequent analyses. Following data pre-processing, feature selection techniques are applied to identify the most influential variables affecting stock levels and the occurrence of out-of-stock (OOS) instances. Principal Component Analysis (PCA) is employed to transform the selected features into a compact feature vector. This transformation not only reduces the dimensionality of the data but also retains its essential information, thus optimizing the subsequent modelling process. Subsequently, MLNN is trained using the transformed feature vector to generate a predictive model specifically tailored for forecasting OOS events in the shopping mall setting. The classification learner tool in Matlab facilitates this process, enabling efficient model training and evaluation. Once the predictive model is developed, the next step involves integrating it with the IoT algorithm using JavaScript. This integration enables seamless communication between the predictive model and the IoT infrastructure, allowing for real-time monitoring of stock levels and proactive management of inventory to mitigate OOS instances.

6. Results and Discussions

The figure 6 presents the results of the chi-square process, identifying the best attributes in the dataset. From the results, it was observed that out of the 9 attributes which model OOS behaviour of the OOS, key features such as quantity in stock, unit price, and available stock are key attributes which model the OOS problem. The figure 6 overall showed that only 9 features were selected correctly and then transformed to form the compact feature vector to train a neural network.

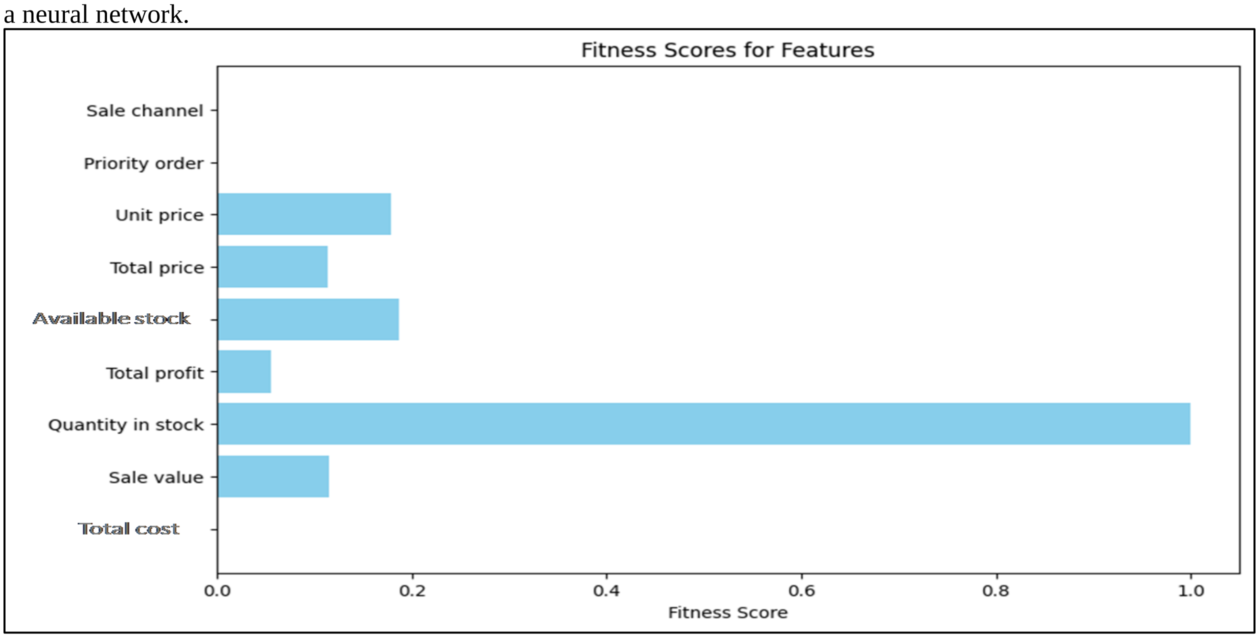


Figure 6: Feature selection process with Chi-square

The figure 6 demonstrated the result of the Chi-square algorithm with the feature with the most importance. PCA was then applied as in figure 7 to transform the features into a compact feature vector.

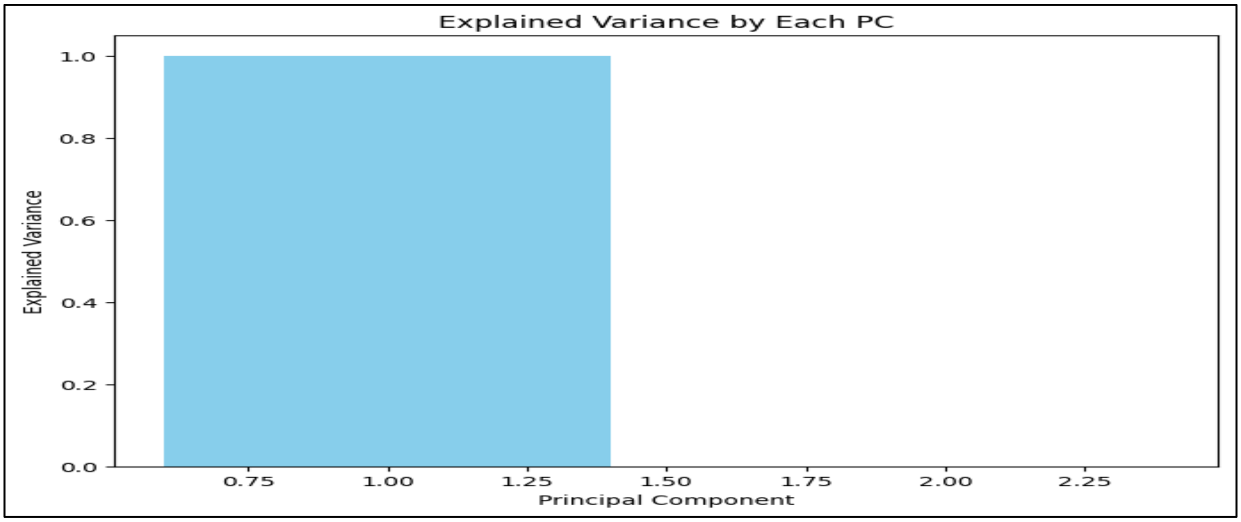


Figure 7: Result of the PCA algorithm for feature transformation

The feature transformation process with PCA in figure 7 was able to transform the values between 0.7 to 1.40 and at a variance level of 0.9897. This means that the PCA was able to transform 99% of the feature vector before training a neural network algorithm in the MATLAB environment. During the training process, back-propagation was used to optimize the neurons and then generate the model for the prediction of OOS in the shopping mall. The training process utilized several tool for evaluation of the model and among the notable tool are the Mean Square Error (MSE) which measured the deviation error between the actual values and predicted values of the stock. The MSE aim is to achieve values close to zero, which suggested tolerable error. The MSE of the neural network was presented in figure 8. In the next evaluation tool, regression which measures the relationships between the target and output values was also applied and reported in the figure 9.

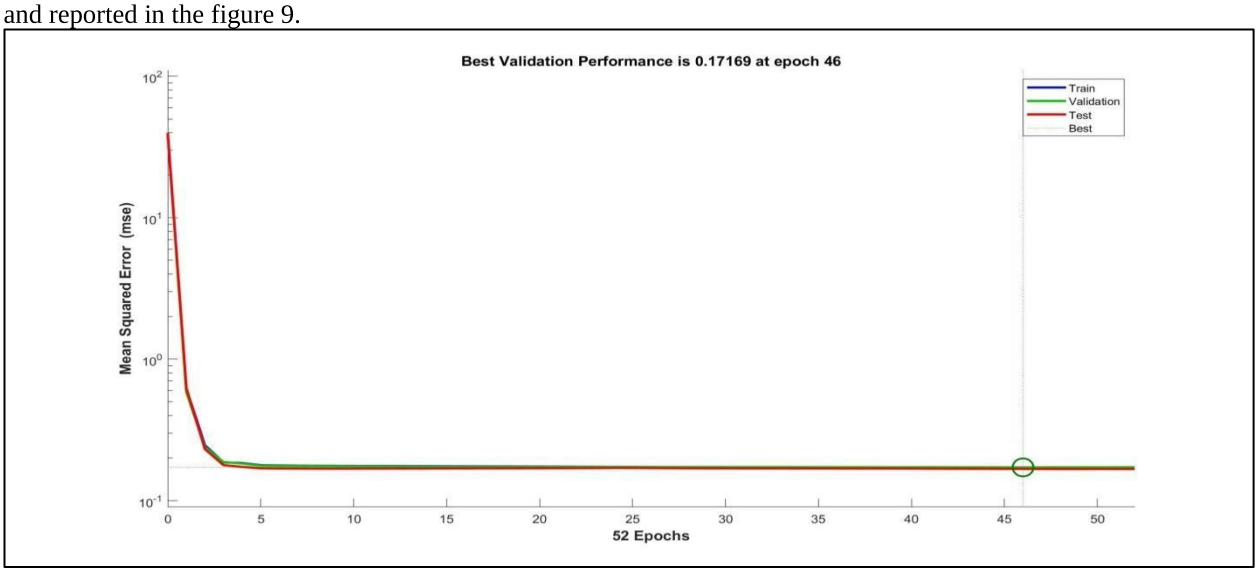


Figure 8: MSE result of the neural network training

The figure 8 presented the MSE of the neural network training, showcasing the relationship between the actual and predicted values and measuring the square of error differences. From the results, it was observed that the best error occurred at epoch 46 and values 0.17169. This result suggested that tolerable errors were achieved which implies that the training process was very good and hence a reliable model reliable. In addition, the results showed that the multi-sets trained which are the training, test and validation sets showed consistency in their performances and recorded considerable low error. The regression results in figure 9 also reported the average regression values for the training set =0.96927, test = 0.96885, validation reported 0.96843 while the overall results recorded R= 0.96907. The results suggested a good score between the target and output values. In addition, the results showed that there was a consistent

R score for the model performance, which is very good and showed the adaptive nature of the model in predicting out of stock situations and also when there is enough stock.

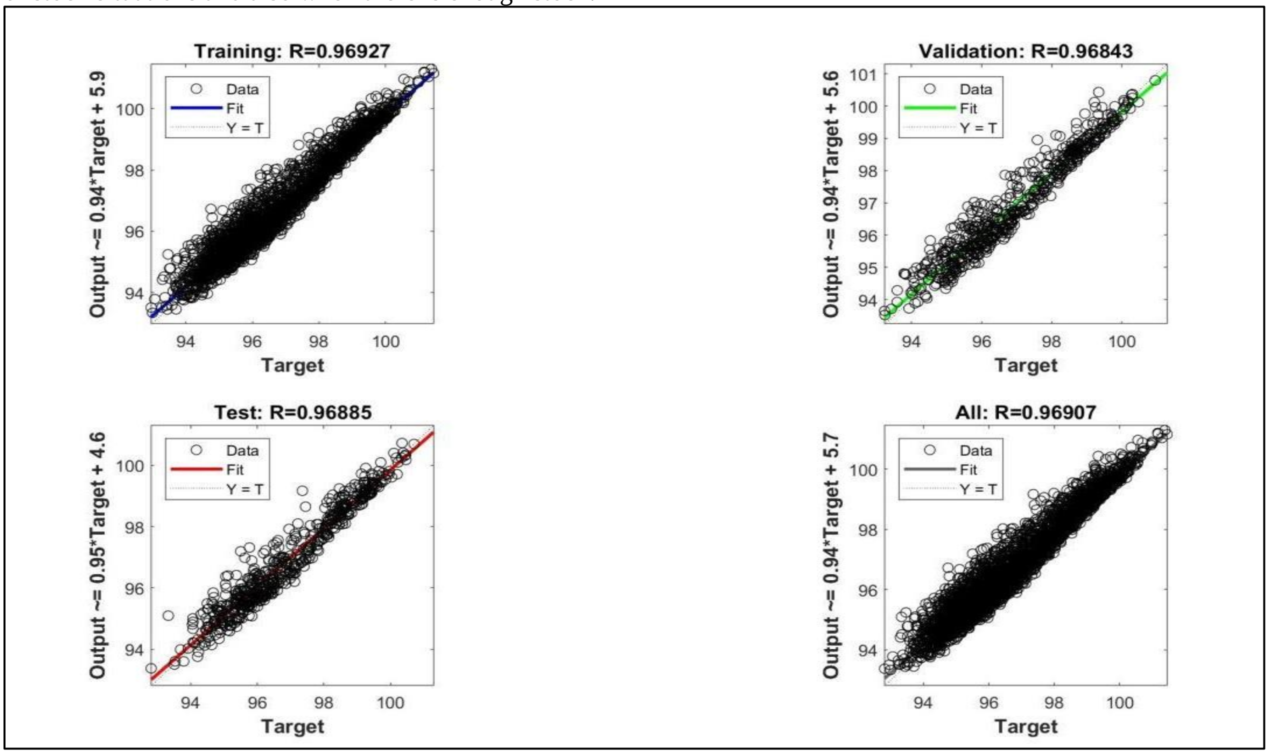


Figure 9: Regression result of the neural network training

The figure 9 presented the regression performance of the neural network training, and how it was able to correctly match the actual stock values with the predicted values.

6.1 Validation of the model using ten-fold approach

This process involved the repetition of the training process in ten folds and then computes the mean score of the results to determine the overall performance of the model generated for OOS prediction. The result of the process is presented in the Table 2.

Table 2: Results of k-fold Validation

Fold	MSE	Regression
1	0.17169	0.96907
2	0.14349	0.97355
3	0.10350	0.97656
4	0.01641	0.99675
5	0.05464	0.95675
6	0.10643	0.96467
7	0.10654	0.96465
8	0.05450	0.97567
9	0.10533	0.96466
10	0.10343	0.97453
Average	0.096596	0.971686

The Table 2 presents the reports of the k-fold validation performed on the prediction model and the outcome of the process. Overall, the average MSE is 0.096596 and R is 0.971686 respectively. These results implied that tolerable error was achieved, while high R values was also recorded which holistically suggest a very good prediction success rate. In addition, the model was also compared with other state of the art algorithms, considering MSE and the results presented in Table 3.

Table 3: Comparative MSE results

Author	Technique	MSE
Praveen et al. (2020)	XGBoost	0.6778
Hajeb and Banafi (2022)	ANN	0.353
Ahakonye et al., (2024)	MLP	0.3504
Guan et al. (2022)	BPNN	0.0163
	PSO based BPNN	0.0259
Vidal et al., (2022)	GA-ANN	0.1444
NEW SYSTEM	FFNN	0.096596

The Table 3 compared the various algorithms which have been successfully developed for the prediction of OOS in shopping mall. From the results, it was observed that the new model developed is among the best 3 of systems when compared with the existing systems, which is very good. In addition, while the existing model have recorded good prediction model, only the new model has internet of things integrated in it and this made it stand out holistically as the best for OOS control system.

6.2 Experimental Validation of the Stock Control

Here the OOS prediction model developed with IoT was integrated as a software for the tracking and stock control at the Shoprite shopping mall. This was achieved using Javascript progamming language. The programming process utilized MySQL to create the stock dataset which recorded the new test data as shown in the Figure 10, presenting the available number of stocks which are used as the test input to the OOS prediction model to determine the level of stock and predict when a particular product will be out and need restocking.

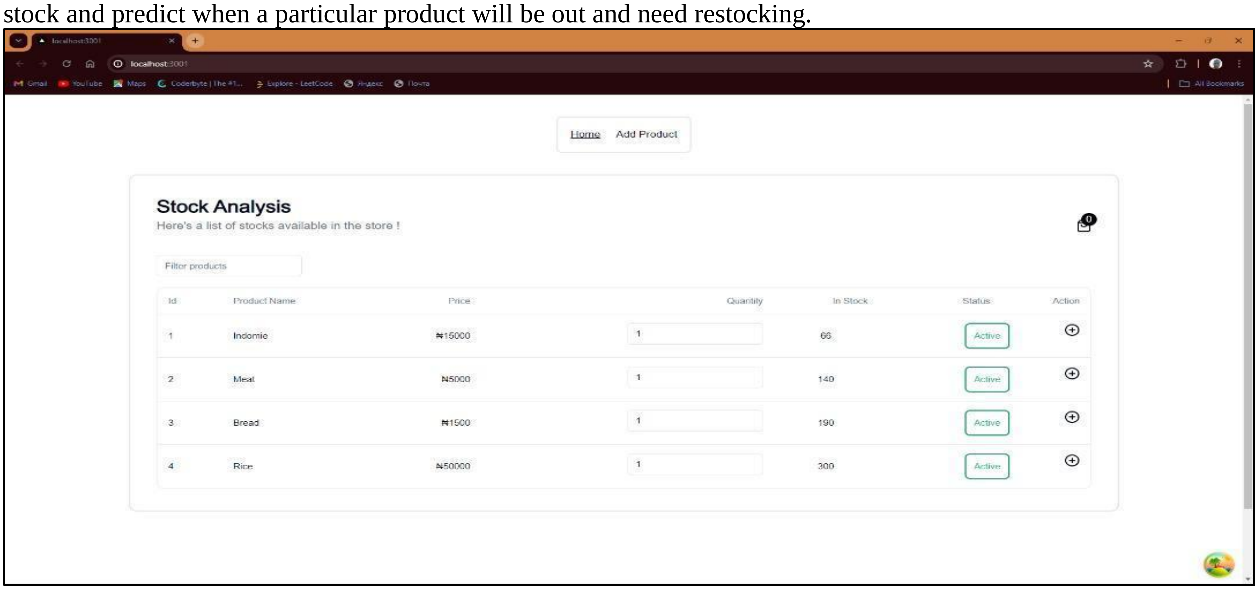
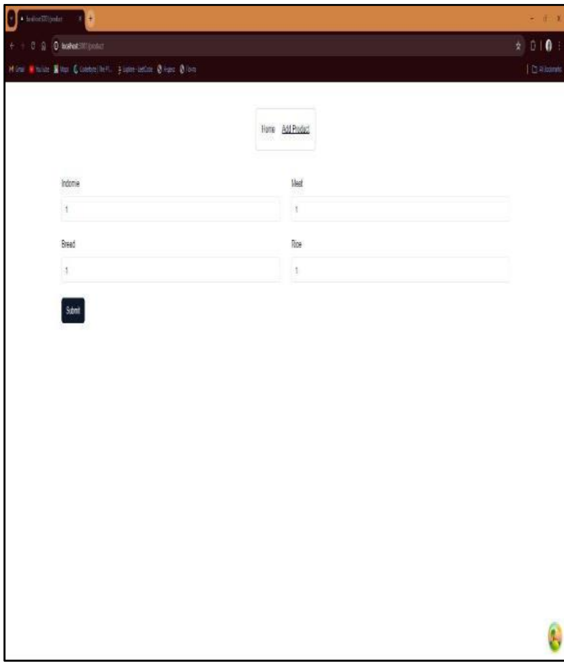


Figure 10: The current stock records

Figure 10 presents the stock information of the products records, which in this test case, four products classes were considered for simplicity. From the results, the number of indomie product is 66, meat is 140, bread is 190 and rice is 308. During the sales of each products, the number in the records keeps changing and the new data identified by the trained OOS prediction model and match to predict when a stock is needed. The Figure 10 showed the input to the data, which is the interface new products are recorded during the restocking process, while the IoT notification which through email alerts the stock manager of the need to restock a product was presented in the Figure 11.



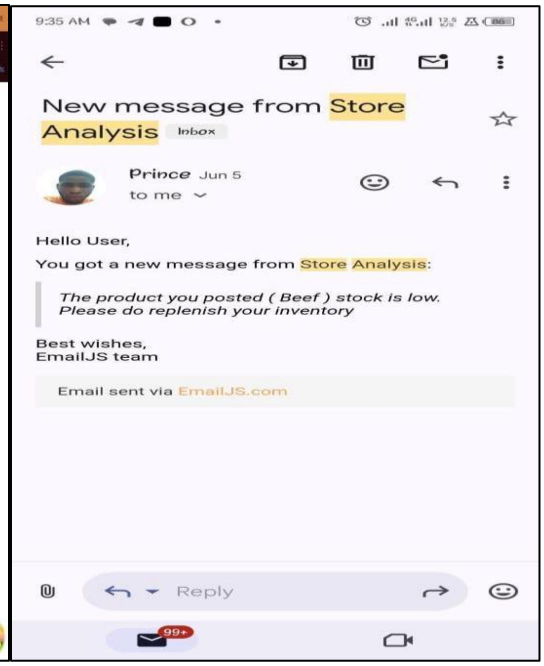


Figure 10: Product restocking interface Figure 11: IoT Email notification

The Figure 10 presented the software interface for product restocking for sell purposes. This section links each number of products bought as new stock to the data class of the stock, while during sales the attributes of the stocks such as level, duration before sale, etc keeps varying and generating new data which are identified by the trained neural network out of stock prediction model to process and determine when a particular stock is due for restocking. Upon this determination, the output is predicted, then the Simple Mail Transfer Protocol (SMTP) initializes and notifies the stock admin of the need for restocking as shown in the results of Figure 11. The Figure 11 presents the results of the IoT which used the input from the prediction model for OOS to initialize the mail transfer protocol, and then notify the user of the need for product restocking. This notification process facilitated immediate product restocking and ensuring that products needed by customers are always available.

7. Conclusion

This paper on the design and implementation of an intelligent business to business stock control system using machine learning and data mining technique was focused on achieving automatic and smart OOS monitoring and notification system using machine learning and IoT. First data which model the OOS problem in a shopping mall was collected and subjected to series of processing steps. The processed data features were selected with Chi-square and then transformed into a compact feature vector using PCA, to train a feed forward neural network algorithm and generate a model for OOS prediction. Upon achieving this, an IoT algorithm which utilized SMTP was used to notify the stock admin of the need for restocking of the identified OOS product. The performances of the prediction algorithm were evaluated considering MSE and Regression. The results of the evaluation process showed that our model recorded tolerable error during the training process with MSE of 0.17168 and Regression of 0.96907, which suggested a very good prediction model. To validate the model, k-fold cross validation approach was applied and the results recorded an MSE average of 0.09686, while the R reported 0.97168. Overall these results showed that the OOS prediction model recorded very good results and fit for integration as software for OOS prediction. Comparative analysis with other state of the art algorithms was performed, considering the MSE results, of the new and existing OOS prediction model and the results showed that the new model was among the top three in terms of best model performance. However, the new model due to its IoT features was the most reliable as it is capable of notifying the stock admin in real time the stock status and need for restocking of products. In conclusion, an IOT algorithm for real time stock status notification was developed, a model for the prediction of OOS in a shopping mall was developed as well as software for the experimental validation of the system integration was developed.

References

Aljohani, A. Predictive Analytics and Machine Learning for Real-Time Supply Chain Risk Mitigation and Agility. *Sustainability* **2023**, *15*, 15088. https://doi.org/10.3390/su152015088

Alotaibi, O.; Pardede, E.; Tomy, S. Cleaning Big Data Streams: A Systematic Literature Review. *Technologies* **2023**, *11*, 101. https://doi.org/10.3390/technologies11040101

Andaur, J.M.R.; Ruz, G.A.; Goycoolea, M. Predicting Out-of-Stock Using Machine Learning: An Application in a Retail Packaged Foods Manufacturing Company. *Electronics* **2021**, *10*, 2787. https://doi.org/10.3390/electronics10222787

Chaudhary V., Bharadwaja K., Meena R., Acharjee P., Chaudhari N., & Gopinathan R., (2023) Exploring the Use of Machine Learning in Inventory Management for Increased Profitability. A Journal for New Zealand Herpetology. BioGecko Vol 12 Issue 01 2023 ISSN NO: 2230-5807

Giaconia, C.; Chamas, A. Innovative Out-of-Stock Prediction System Based on Data History Knowledge Deep Learning Processing. *Computation* **2023**, *11*, 62. https://doi.org/10.3390/computation11030062

Gonzalez-Ayala, P.; Alejo-Reyes, A.; Cuevas, E.; Mendoza, A. A modified simulated annealing (MSA) algorithm to solve the supplier selection and order quantity allocation problem with non-linear freight rates. *Axioms* **2023**, *12*, 459.

Hajeb H., &Banafi M., (2022) Predicting the Efficiency of Inventory Management Using Artificial Neural Networks. International Journal of Management, Accounting and Economics Volume 9, Issue 11, November, 2022 ISSN 2383-2126 (Online) DOI: 10.5281/zenodo.7421138

Kousar H., Gupta L., Mamoria P., Haldar B., & Naidu P., (2023)Artificial Intelligence and Machine Learning In Inventory Management: Optimizing Efficiency And Reducing Costs. Eur. Chem. Bull. 2023,12(Special Issue 12), 1095-1105

Nobil, A.H.; Nobil, E.; Cárdenas-Barrón, L.E.; Garza-Núñez, D.; Treviño-Garza, G.; Céspedes-Mota, A.; Loera-Hernández, I.d.J.; Smith, N.R. Discontinuous economic growing quantity inventory model. *Mathematics* **2023**, *11*, 3258

Obaji C. Okonkwo O., Ebere U.C. (2021)"Enhancing the performance of pipeline leakage detection system using artificial neural network "IJRIAS 6 (XII), Page No.: 39-44

Onuoha Solomon I., Alor Michael. Ebere Uzoka C. (2022)"Reducing instability in Power Distribution Network Using Intelligent Control of System Load frequency changes International Journal of Research and Innovation in Applied Science 7 (4), 34-39

Osman B., Alinkeel S., &Bhavshar D., (2022) A Study on Role of Artificial Intelligence to Improve Inventory Management System. International Research Journal of Modernization in Engineering Technology and Science Volume:04/Issue:03/March-2022

Piao, M.; Zhang, D.; Lu, H.; Li, R. A supply chain inventory management method for civil aircraft manufacturing based on multi-agent reinforcement learning. *Appl. Sci.* **2023**, *13*, 7510.

Praveen K., Prateek J., Kumar P., Pragathi G., & Madhuri J., (2020) Inventory Management using Machine Learning. International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERTV9IS060661

Praveen K., Prateek J., Kumar P., Pragathi G., & Madhuri J., (2020) Inventory Management using Machine Learning. International Journal of Engineering Research & Technology (IJERT) http://www.ijert.org ISSN: 2278-0181 IJERTV9IS060661

Raufi, B.; Longo, L. Comparing ANOVA and PowerShap Feature Selection Methods via Shapley Additive Explanations of Models of Mental Workload Built with the Theta and Alpha EEGBandRatios. *BioMedInformatics* **2024**, *4*,pp.853-876. https://doi.org/10.3390/biomedinformatics4010048

Sirakov, N.M.; Shahnewaz, T.; Nakhmani, A. Training Data Augmentation with Data Distilled by Principal Component Analysis. *Electronics* **2024**, *13*, 282. https://doi.org/10.3390/electronics13020282

Tao, S.; Liu, S.; Zhou, H.; Mao, X. Research on Inventory Sustainable Development Strategy for Maximizing Cost-Effectiveness in Supply Chain. *Sustainability* **2024**, *16*, 4442. https://doi.org/10.3390/su16114442