PROPOSED MAS-SCM-SOA ARCHITECTURE TO FORECAST CUSTOMER'S DEMANDS IN AN ONLINE PRODUCT PURCHASING SCENARIO (OPPS) USING SUPPORT VECTOR METHOD

[1]Sujit Kr Singh, [2]Prof. (Dr.) Brij Bhushan, [3]Prof. (Dr.) S. Srinivasan

[1]Ph.D. Research Scholar, [2]Professor, Mewar University, [3]Professor, P.D.M. University
[1]Department of Computer Science & System Studies,
[1]Mewar University, Chittorgarh, Rajasthan, India

Abstract: Nowadays, SCM industries are facing so much issues with rapid change in demand of customers, and it is very difficult to forecast these demands with the traditional methods. These rapid changes cause so many problems like order fulfillment, be competitive with other SCM partners, notice customers experiences, irregular inventory, making/increasing sales with an increment of holding cost. Existing nonparametric trends of data are not sufficient for handling the irregularities present in these demands. A proposed MASSCM architecture with machine learning approach of artificial intelligence and integration of Time Series and Support Vector Machine (SVM), shows significant success in forecasting demand in a rapid change in customer's choice and taste. This work with SVMs reduces mean absolute deviation, mean absolute percent error and mean square error, provides a fast and simple methodology to get the forecasting data, and indirectly reduces the inventory cost of a system. The experimental data was taken from a well-known Mobile manufacturing company where professional's advice has been considered accordingly.

***Index Terms* - Multi-Agent System (MAS), Service Oriented Architecture (SOA), Online Product Purchasing Scenario (OPPS), Support Vector Machine (SVM), Mean Absolute Deviation (MAE), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE).**

I. INTRODUCTION

Traditional forecasting methods, such as Time series methods and smoothing averages, which are designed for regular, high-volume demand, not work well for a rapid change in demands.

These methods, commonly used by most of the industries and other forecasting software systems, fail because they try to identify familiar patterns in the demand data, such as trend cyclic and seasonality.

Support Vector Machine is a supervised machine learning algorithm that has popularity in constructing an optimal hyperplane that categorizes new examples because of its effectiveness in high dimensional spaces and memory efficiency.

In this paper First, I describe the working of Online product Purchasing Scenario (OPPS) with their issues faced by most of SCM Industries like optimization of their inventories, to sustain in the highly competitive market and provide quality products to their customer in low cost.

Therefore, a proper prediction of customer demands almost need by most of SCM industries. Among various types of demand, some occur at infrequent, irregular, and often unpredictable intervals known as intermittent demand.

Secondly, I proposed an agent-based architecture for incorporating support vector to fulfil the needs of SCM Industries to solve all the above issues by providing accurate forecasting of customer demands in advance and also helps customers in their buying based on their search and need in an Online Product Purchasing Scenario (OPPS).

In my proposed architecture I integrated different agents such as: User Driven Site Analysis Agents, Rating Agents, Aggregator Agents, Model trainer Agents, Predictor Agents, Decision Maker Agents, Data Persistence Agents, Demand Predictions Agents and WEB API agents.

Every individual agent will perform their role and as well as social interaction to achieve common goals. Then I explained the data acquisition process and types of data collected once a product search initiated by the customers, also I explained sequence flow diagram to show the processing of data among various agents.

Lastly, I introduced support vector regression model to process the historical data till SVM generated final forecasting value and then calculate the MAD, MAPE and MSE to see the performance of model's forecasting accuracy.

II. LITERATURE REVIEW

Support Vector Machines (SVM) is learning machines implementing the structural risk minimization inductive principle to obtain good generalization on a limited number of learning patterns. The theory has originally been developed by Vapnik [1].

A version of a SVM for regression has been proposed in 1997 by Vapnik, Steven Golowich, and Alex Smola [2] SVM was initially developed by Vapnik as a new generation of the learning algorithm, an optimum margin classifier, which is a Forecasting Supply Chain Sporadic Demand linear classifier that constructs a separating hyperplane such that the distance between the positive and negative samples is maximized [2].

Artificial Intelligence based approaches have also been proposed, most often, Artificial Neural Networks but these do need extensive training datasets [3].

A comparison between a Neural Network approach and an approach using the regression feature of SVM was made in financial market prediction [4].

Support vector regression is the natural extension of large margin kernel methods [5] used for classification to regression analysis.

III. ONLINE PRODUCT PURCHASING SYSTEM (OPPS)

3.1 Introduction

All customers buying product on the basis of a common philosophy, their principle is simple: individuals buy any item on an "as-needed" basis without the costs and who is selling.

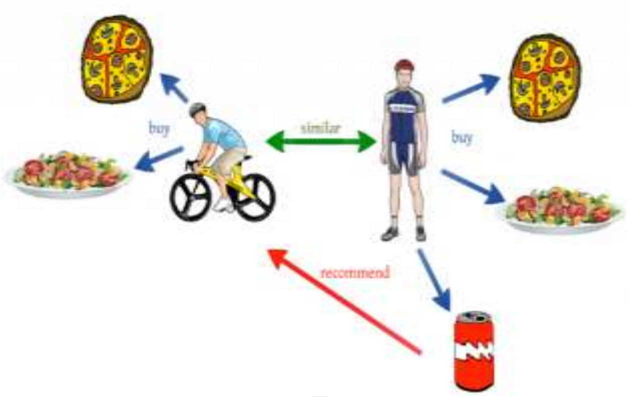

Figure 1: Classical Online Purchasing Scenario

There are available items displaying as per as customer search criteria in classical online purchasing scenario. As shown on the figure, online vendors offer a wide list of products of all brands located across the country.

In an Online Product Purchasing Scenario, it can be categorized into different prospects depending on their features:

1) It displays the item search by the customers.
2) This tries to display the same products of different brands.
3) This displays similar items products of all other brands.
4) It added the item into customer's cart.
5) In these customers select or type the address or location where he/she received or delivered the products.
6) In this customer opted the payment option to confirm their booking.
7) Online vendors delivered the ordered item to the given locations.

3.2 Online Product Purchasing: Main Issues

Current literature lists factors that now most of SCM industries they are facing some of the issues while selling their product through e-commerce, which are as follows:

1) Competition: Be competitive in the market is always a biggest challenge for almost all the industries, especially in the e-commerce space. They all have to keep up with competitive pricing, products and service.
2) Order Fulfilment: Being as a SCM industries they have to overwhelm with more orders than they produced. Order fulfilment should be outsourced to a third-party fulfilment company whenever possible for increased efficiency.
3) Customer Experience: As a primarily e-commerce business, difficult to figure out how to offer your customers the same experience they would get in a brick-and-mortar store can be challenging. One of the most overlooked areas of the customer experience in moving to e-commerce is pricing and customer segmentation.
4) Return/refund policy: Most SCM Industries' success and failure depends on their return/refund policy to provide the customer satisfaction as their first priority, and whatever they are selling should be the same as what's advertised.
5) Finding the right market: For any business, finding a suitable market for their product is tedious task. Product-market fit is the degree to which a product satisfies market demand. Finding the right market for your product isn't the easiest task, though.
6) Making/increasing sales: Once SCM industries figured out website set up and their marketing set, always they have to focus on increase of sales and this only possible when they have the right product at the right price.

3.3 Proposed Architecture to solve OPP Issues

3.3.1 Proposed Strategy using MAS SCM

Here we propose an agent-based architecture for incorporating support vector to help SCM Industries to solve all the above issues by providing accurate forecasting of customer demands in advance and also helps customers in their buying based on their search and need in an Online Product Purchasing Scenario (OPPS). In our proposed architecture, we list six agents with distinct roles: Sensor Agent, Aggregator Agent, Predictor Agent, Model Trainer Agent, Decision Maker Agent and User Interface Agent. All the agents offer an ideal solution to abstract away issues of the different existing platforms and communication protocols. The system is divided into the following groups of agents: User Driven Site Analysis Agents, Rating Agents, Aggregator Agents, Model trainer Agents, Predictor Agents, Decision Maker Agents, Data Persistence Agents, Demand Predictions Agents and WEB API agents.

Agent-based systems comprise of multiple agents which are encapsulated software processes situated in an environment and exhibiting autonomy, social ability, responsiveness and proactiveness [6].

In our proposed architecture, we list six agents with distinct roles: E-commerce Data-tracker Agent, Aggregator Agent, Predictor Agent, Model Trainer Agent, Decision Maker Agent and User Interface Agent.

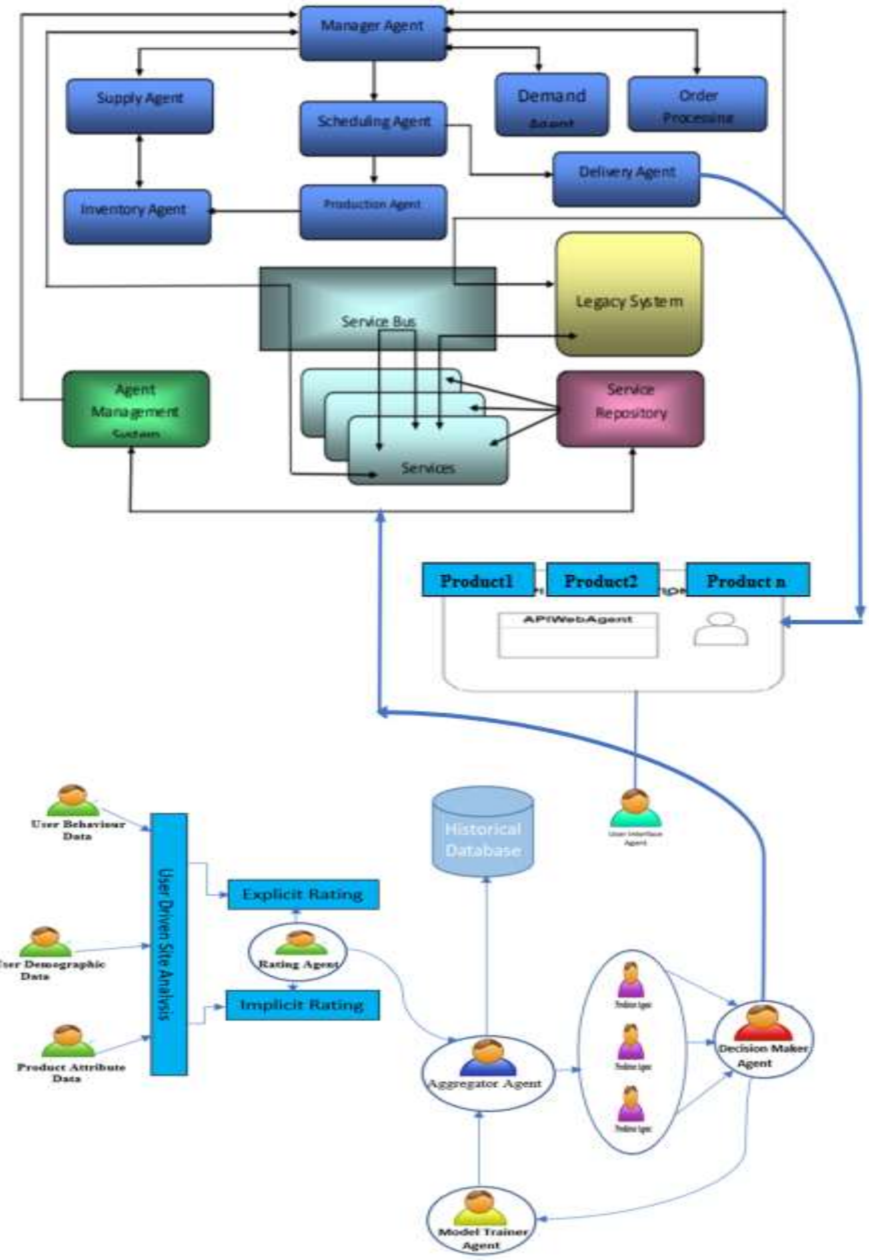

Figure 2: Proposed Strategy using MAS-SCM-SOA

3.3.2 Roles and Social Interactions of Several Agents in our Proposed Strategy using MAS-SCM-SOA

Table 1: Agent's system role while predicting customers' demands

Class	System Role
E-commerce Data-tracker Agent	Data source and interface to track or capture data from different platforms like mobile app, web site & stores and sends to the Aggregator Agent.
Aggregator Agent	Aggregate data from all different sources & sends aggregated data to the Predictor Agent and historical database.
Predictor Agent	Runs the equipped predictive model on every pre-determined batch of incoming data to compute prediction and sends computation result to Decision Maker Agent.
Model Trainer Agent	Manages, updates and deploys model and actively interacts with the Decision Maker Agent for deploying and updating model.
Decision Maker Agent	Obtains Predictions and makes final decision based on different approaches then communicates the User Interface Agent.
User Interface Agent	Reads and interprets results from the Decision Maker Agent.

3.4 Steps involved in proposed MASSCM using SOA to improvise the OPPS

3.4.1 Data Acquisition: Steps

Step 1: Start serving the website.

Step 2: User interface agent monitor all the users takes information about the user.

a) *User Product:* Check the preference towards specific products.
b) *Product:* Items with similar in nature, either by appearance or description.
c) *User:* Customers having similar choice or taste with respect to a particular product or service.

Step 3: Then User Driven Site Analysis collects all the information like:

a) *User Behavior Data:* Information about the engagement of the user on the product. It can be collected from ratings, clicks and purchase history.

b) *User Demographic Data:* User demographic information is related to the user's personal information such as age, education, income and location.

c) *Product Attribute Data:* Product attribute data is information related to the product itself such as genre in case of books, cast in case of movies, cuisine in case of food.

The reason for having separate agents for each task such as sensing, aggregation, prediction and decision making is to ensure the flexibility, hardware interoperability and heterogeneity of the system.

3.4.2 Type of Data Collected During the Process

The data collected regards all the products seen, booked and orders through its portal from signup/login to exit/logout from the portal. For my research I have collected the data, based on users visited the portal or placed order previously and it has the following information in the original dataset:

1) *Time Start:* timestamp of visiting, sign in to the portal,
2) *Time End:* timestamp of the exit or logout from the portal,
3) *Product ID:* unique product identifier,
4) *Vendor Name:* origin seller name,
5) *Delivery Location:* Delivery location of the user,
6) *Product Visited Time and Date:* customer track identifier,
7) *User's Rating:* Rated by the user,
8) *Customer ID:* Customer's unique identifiers.

I have downloaded the information in the form of csv files, one for each year. To process this information, the files were given to the multi-agent system. The WEBAPI agent loads all the data from the provided data source, the processor agent is notified by the persistence agent because new raw data is added to the system. The processor agent checks the data and request for rating and consumer's review based on the products search by and order by the customers. Before aggregator agents respond to the request, they check the availability of data with the persistence agent, if the data is available, it is sent immediately. On the other hand, if information is not available it is first collected, saved and then sent to the processor agent.

3.5 Sequence diagram to illustrate the processing of data loading by MAS-SCM-SOA

Once the request or search get started by the customer then the entire data processed through several agents to make final decision to help the users in buying their products with a reasonable cost as well as similar type of products if the same product is not available on their search.

Here the following figure shows a sequence diagram of the activities performed by each agent, the communication between the several agents for clarity and processing.

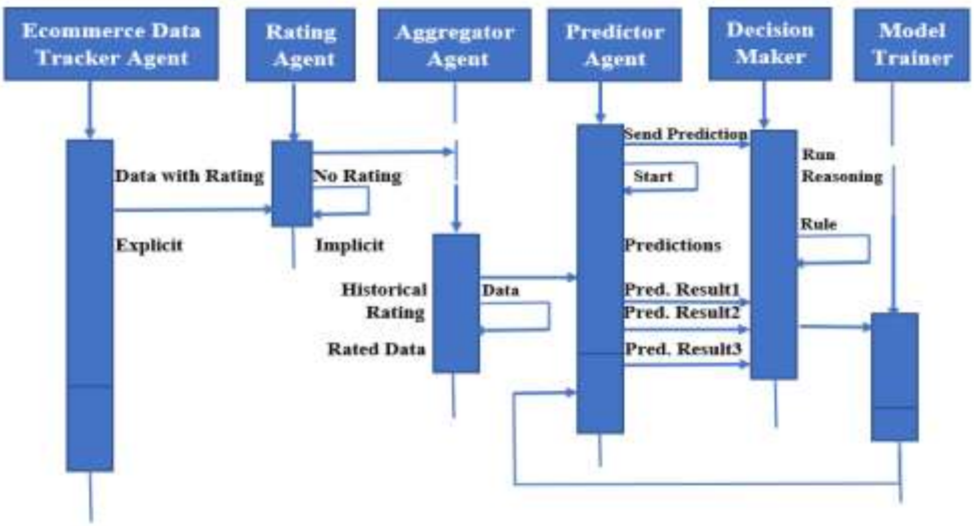

Figure 3: Sequence diagram of Processing of Customer's data loading by Multi-Agent System

After this process, the stored output data contains the following information on Products rating and Review: based on Year, Month, Day of the Week, Day of the month, week of the year, hour, season (winter, spring, summer or autumn), weekends, holidays and festival season related information. The information that is available in relation to the products on a particular day, is the following: the maximum order, minimum order and average order placed, the products review, checked by the customers.

Before procreating the predictor data agent in the multi agent system, an exploratory analysis of the processed data was performed. This analysis determined what model should be included in the predictor data agent.

IV. SUPPORT VECTOR MACHINE (SVM)

4.1 Introduction

The support vector machine (SVM) is a predictive analysis data-classification algorithm that assigns new data elements to one of labelled categories. SVM is, in most cases, a binary classifier; it assumes that the data in question contains two possible target values [7].

Support Vector Machine (SVM) is a popular machine learning method for classification, regression, and other learning tasks.

Support Vector Machines (SVM) is learning machines implementing the structural risk minimization inductive principle to obtain good generalization on a limited number of learning patterns [1].

SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes [8].

4.2 Flow Chart

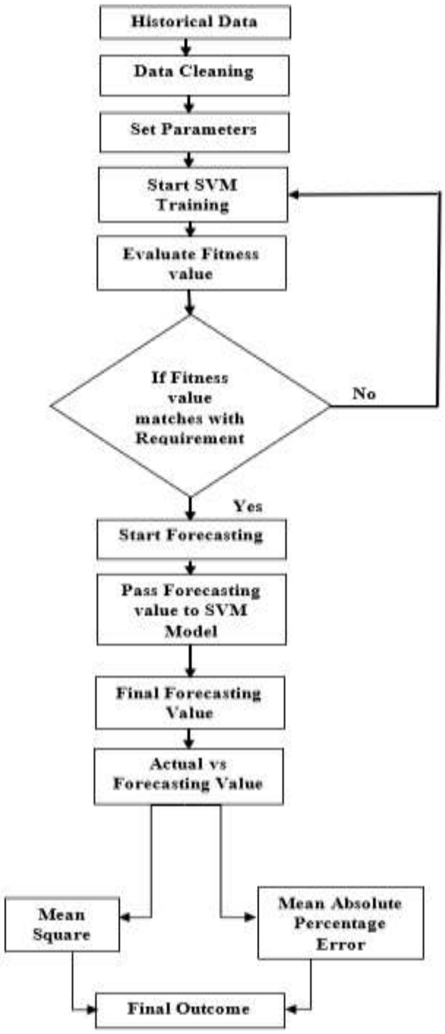

Figure 4: Model Development Flow Chart

At first, the data acquisition takes place based on the previous search, buying & review records of various products and then preparation of dataset shall be done.

Secondly, data cleaning process should be performed to eliminate missing and need less data means collected data is formatted, and parameters are declared.

In the third phase, we shall prepare parameters list around which training and testing of data carried out with the help of SVM model.

At last, by giving input to the model, SVM training process is started. With the help of algorithm SVM model works itself until the fitness value obtained and then it validates or matches fitness value with our requirement if yes then based on the fitness value forecasting value get started and if it not matched then again, we go back to earlies phases to improvise the SVM training process until we get required forecasting value which then passes to SVM model to consider few of them to get Final forecast values are obtained. After getting the final value, a tabular dataset shall be prepared with actual and forecast values. Performance level is measured through calculating the error percentage by traditional approaches. Such as MAPE and MSE and also to see the efficacy of the forecasting values. Finally, the model concludes the process.

4.3 Support Vector Model: Working & Result

We have collected the data from an ABC Mobile Manufacturing company. The data contains sales value of the product form April 2018 to March 2019.

Data Collections:

The data chart of weekly sales value for the year April 2018 to March 2019 is given in Table 2.

Table 2: Weekly sales value of ABC Mobile Manufacturing Company

Week	Sales Data (Weekly)	Week	Sales Data (Weekly)	Week	Sales Data (Weekly)	Week	Sales Data (Weekly)
1	11	15	5	29	8	43	1
2	9	16	6	30	4	44	2
3	7	17	3	31	1	45	0
4	6	18	2	32	5	46	8

5	8	19	7	33	3	47	5
6	5	20	5	34	7	48	7
7	10	21	4	35	8	49	9
8	4	22	8	36	2	50	3
9	3	23	4	37	4	51	2
10	8	24	3	38	3	52	6
11	6	25	8	39	5	53	7
12	1	26	9	40	4	54	4
13	0	27	7	41	5	55	1
14	9	28	6	42	3	56	3

4.4 Algorithm

```
SVM regression for forecasting customer demand
Assign actual values to random variable a_1 to a_n
        for i=1 to m do
                Perform SVM to find a regression model
        end for
        for j=1 to n do
                Predict values with the regression model up to period n
        end for
print results
return c
```

Here, using the collected data parameters are declared. Then, input the data to the model, SVM training process is started. The algorithm works itself until it finds a proper regression model. With the help of this model, all the values that the algorithm finds are tested and then the results are declared.

4.5 Results

All the collected data are plotted in the algorithm and after training the algorithm, the results are found. The results are then compared with the actual value collected from the company to find how accurate the predictability of our algorithm is.
Predicted sales value of 6 weeks of November 2019 is given in Table 3.

Table 3: Forecasted Sales Value/ Week of 1 Month

Week	Forecasted Data
1	11.43
2	9.65
3	7.33
4	6.78
5	8.24
6	5.37

Table 4: MAPE Calculation to Evaluate Accuracy of Forecasting Result

Week	Actual Data	Forecasted Data	Error
1	11	13	(2)
2	9	11	(2)
3	7	5	2
4	6	9	(3)
5	8	11	(3)
6	5	4	1

Mean Absolute Deviation (MAD)

Mean absolute deviation (MAD) of a data set is the average distance between each data value and the mean. It is a way to describe variation in a data set. It helps us get a sense of how "spread out" the values in a data set are.

```python
import pandas as pd
dataSet = {"Actual Data":(11, 9, 7, 6, 8, 5),

           "Forecasted Data":(13, 11, 5, 9, 11, 4)

          }
dataFrame = pd.DataFrame(data=dataSet);
print("DataFrame:");
print(dataFrame)
# Calculate Mean Absolute Deviation of DataFrame columns
mad = dataFrame.mad();
print("Mean absolute deviation of columns:");
print(mad);
# Calculate Mean Absolute Deviation of DataFrame rows
mad = dataFrame.mad(axis=1);
print("Mean absolute deviation of rows:");
print(mad);
```

```
DataFrame:
   Actual Data  Forecasted Data
0           11               13
1            9               11
2            7                5
3            6                9
4            8               11
5            5                4
Mean absolute deviation of columns:
Actual Data        1.666667
Forecasted Data    2.888889
dtype: float64
Mean absolute deviation of rows:
0    1.0
1    1.0
2    1.0
3    1.5
4    1.5
5    0.5
dtype: float64
```

Mean Squared Error (MSE)

One of the most common metrics used to measure the forecast accuracy of a model is MSE, which stands for mean squared error. It is defined as mean or average of the square of the difference between actual and estimated values.
It is calculated as:

$$MSE = (1/n) * \Sigma \text{ (actual – forecast)}^2 \quad (1)$$

where:
Σ – a fancy symbol that means "sum"
n – sample size
actual – the actual data value
forecast – the forecasted data value
The lower the value for MSE, the better a model is able to forecast values accurately.

```
y = [11,9,7,6,8,5]
y_bar = [13,11,5,9,11,4]
summation = 0  #variable to store the summation of differences
n = len(y) #finding total number of items in list
for i in range (0,n):  #looping through each element of the list
  difference = y[i] - y_bar[i]  #finding the difference between observed and predicted value
  squared_difference = difference**2  #taking square of the differene
  summation = summation + squared_difference  #taking a sum of all the differences
MSE = summation/n  #dividing summation by total values to obtain average
print ("The Mean Square Error is: " , MSE)
```

```
The Mean Square Error is:  5.166666666666667
```

Mean Absolute Percentage Error (MAPE)

All the collected data are plotted in the algorithm and after training the algorithm, the results are found. The results are then compared with the actual value collected from the company to find how accurate the predictability of our algorithm is.

```
import numpy as np

def mape(actual,pred):
    return np.mean(np.abs((actual - pred) / actual)) * 100

actual = np.array([11,9,7,6,8,5])
pred = np.array([13,11,5,9,11,4])

#Calculate the MAPE
result = mape(actual,pred)

#print the result

print("The mean absolute percentage error: ",result)
```

```
The mean absolute percentage error:  29.412578162578164
```

V. CONCLUSION

As a result, I conclude the study by the fact that the success of Support vectors Model not only improvise the prediction but also it decreases percentage error during predicting customers' demands on a given data sets with around 4-5% enhancement at average MAPE results level.

The proposed implementation of MASSCM using SVM model presents final decision consider for correct prediction as well also improvise the model on every result set with the help of Model trained Agent.

REFERENCES

[1] Vapnik, V. 2000. The nature of statistical learning theory, Springer, NY.
[2] Vapnik, V. 1995. The Nature of Statistical Learning Theory. Springer, Berlin.
[3] Gutierrez, R.S., Solis, A.O. and Mukhopadhyay, S. 2008. Lumpy demand forecasting using neural networks, International Journal of Production Economics, 111 (2), 409–420.
[4] Lai, L. and Liu, J. 2010. Stock forecasting using support vector machine, Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, 1607-1614.
[5] Soman, K.P., Loganathan, R. and Ajay, V. Support vector machines and other kernel methods," Centre for Excellence in Computational Engineering and Networking Amrita Vishwa Vidyapeetham.
[6] Wooldridge, M. and Jennings, N.R. 1995. Intelligent agents: Theory and practice. The knowledge engineering review, 10(2):115–152.
[7] Bari, A., Chaouchi, M. and Jung, T. How Support Vector Machine Predictive Analysis Predicts the Future. [Online: https://www.dummies.com/programming/big-data/data-science/how-support-vector-machine-predictive-analysis-predicts-the-future/]
[8] Pupale, R. 2018. Support Vector Machines (SVM) - An Overview. [Online: https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989]