electronics

MDPI

Article

Fog Computing Enabled Locality Based Product Demand Prediction and Decision Making Using Reinforcement Learning

Gone Neelakantam [1], Djeane Debora Onthoni [1] and Prasan Kumar Sahoo [1,2,*]

[1] Department of Computer Science and Information Engineering, Chang Gung University, Guishan 33302, Taiwan; D0421004@cgu.edu.tw (G.N.); D0421008@cgu.edu.tw (D.D.O.)
[2] Division of Colon and Rectal Surgery, Chang Gung Memorial Hospital, Linkou 33305, Taiwan
* Correspondence: pksahoo@mail.cgu.edu.tw; Tel.: +886-3-211-8800 (ext. 3804)

Abstract: Wastage of perishable and non-perishable products due to manual monitoring in shopping malls creates huge revenue loss in supermarket industry. Besides, internal and external factors such as calendar events and weather condition contribute to excess wastage of products in different regions of supermarket. It is a challenging job to know about the wastage of the products manually in different supermarkets region-wise. Therefore, the supermarket management needs to take appropriate decision and action to prevent the wastage of products. The fog computing data centers located in each region can collect, process and analyze data for demand prediction and decision making. In this paper, a product-demand prediction model is designed using integrated Principal Component Analysis (PCA) and K-means Unsupervised Learning (UL) algorithms and a decision making model is developed using State-Action-Reward-State-Action (SARSA) Reinforcement Learning (RL) algorithm. Our proposed method can cluster the products into low, medium, and high-demand product by learning from the designed features. Taking the derived cluster model, decision making for distributing low-demand to high-demand product can be made using SARSA. Experimental results show that our proposed method can cluster the datasets well with a Silhouette score of $\geq$ 60%. Besides, our adopted SARSA-based decision making model outperforms over Q-Learning, Monte-Carlo, Deep Q-Network (DQN), and Actor-Critic algorithms in terms of maximum cumulative reward, average cumulative reward and execution time.

Keywords: fog computing; product demand prediction; artificial intelligence; reinforcement learning; SARSA; PCA; ERP; K-Means

check for updates

Citation: Neelakantam, G.; Onthoni, D.D.; Sahoo, P.K. Fog Computing Enabled Locality Based Product Demand Prediction and Decision Making Using Reinforcement Learning. *Electronics* **2021**, *10*, 227. https://doi.org/electronics10030227

Academic Editor: Nancy Alonistioti
Received: 22 November 2020
Accepted: 17 January 2021
Published: 20 January 2021

1. Introduction

With the technological advancements for the convenient lifestyle, it is observed a massive growth in human migration towards the urban smart cities [1]. Hence, high density vertical infrastructures [2] in smart communities accommodate the massive population in limited smart city areas [3]. Furthermore, excessive demand and consumption of food in and around the localities of smart city area increases rapidly. Food supply chain industry distributes daily consumable products through hypermarkets, supermarkets, convenience stores and so on. According to United States Department of Agriculture [4], Taiwan hosts 2299 supermarkets that are deeply indulged into daily shopping routines of people living in the localities. Each supermarket sells and promotes a wide variety of products that are stacked on the supermarket shelves, which can be categorized as perishable and non-perishable products. In general, perishable products create huge wastage of food in the supermarket industry. For instance, according to Food and Agriculture Organization (FAO), 194∼389 kgs per person and 1.3 billion tons of food products are wasted annually [5]. The delay in monitoring and decision making leads [6] to vast wastage of low demand products in supermarkets. When a customer reaches at the supermarket, the most concerning factors include the date of expiry and sale of the products. Due to huge density of population in the smart city, number of supermarkets in a locality could be more than one, holding a wide variety of products in different regions that are monitored by the Internet of Things (IoT) enabled devices.

Integration of Internet of Things (IoT) into supermarket system enhances the product monitoring and decision making to improve the customer quality of service. IoT enables smart supermarkets to monitor products in each supermarket with wide range of sensors such as shelf sensor, CCTV, motion sensor, Radio-frequency identification (RFID) and so on [7]. Moreover, IoT sensors embedded in smart supermarket assist in finding the product availability and date of expiry. Products monitoring and tracking the information of product status with embedded sensor can assist in retrieving information of a product at any stage of the food supply chain [8] and logistic [9]. The IoT sensor data generated by smart supermarket sensors are collected and analyzed in cloud computing platform. However, due to large distributed network of supermarkets and individual preference of locality-based customers, cloud platform lacks in predicting real-time demand of the products due to minimal context awareness [10]. Thus, we considered the fog based smart supermarket environment where the near edge deployment of fog computing nodes enhance the computational capabilities [11] and context awareness [12]. The data generated from the smart supermarkets and users in the smart city are processed and analyzed in respective localities through which a locality-based product sales data are collected from the local fog data centers. The fog data centers host intelligence algorithms-based computing platform to predict the locality-based supermarket products demand.

Artificial Intelligence (AI) is a trending technology, which retrieves meaningful information by processing data generated from widely deployed IoT sensors and devices in smart supermarkets. Wide range of Machine Learning (ML) methods such as Supervised Learning (SL), Unsupervised Learning (UL) and Reinforcement Learning (RL) algorithms assist in extracting various parameters to build the decision making and prediction model [13]. Moreover, various SL and UL algorithms were adopted such as Random Forest (RF) and Keras Neural Network [14] to predict demand of grocery items, Boosted Decision Tree Regression (B-DTR) and Deep Neural Network (DNN) to predict sale of popular beverage among competing brands [15]. Moreover, RL can be used for designing sequential decision making. RL based real-time decision making assists end-user in real-time smart city transportation application [16]. It enables an agent to learn directly from the environment by taking an action and getting a reward as feedback, which can be used for an agent to take the next action. RL does not depend on the historical data and it learns through trail-and-error basis. Therefore, we adopted a feature-based learning method using State-Action-Reward-State-Action (SARSA) to assist both supermarket system and customer in predicting the high demand products from all supermarkets in a region and assisting with real-time decision making.

1.1. Motivation and Goals

Significantly, limited technological intervention in the supermarket product monitoring systems stands as a prevailing drawback in this era of Information and Communication Technology (ICT). Supermarket system is engaged in selling multiple types of products that are mainly categorized into perishable and non-perishable products. Conventionally, these products are monitored using Enterprise Resource Planning (ERP) through Supply Chain Management (SCM) applications. ERP in SCM has some benefits such as supply and monitoring of products in the supermarkets. For instance, ERP enables the product demand forecasting, which can estimate the prediction and inform the supermarket supply chain. However, it has been reported that ERP has suffered from high failure rate of 60% up to 90%. It has also been found that ERP is a complex system that can lead to the failure in achieving the expected level of ERP investments [17]. Besides, ERP software is costly, which is more costlier than the implementation and maintenance of ERP where manpower is highly required. Therefore, it is important to get more profits for supermarkets with less failure rate by developing precise prediction model using ML. By doing so, we plan to adopt a sophisticated ML algorithm, which can handle the complex data and system, and provide an intelligent system with accurate prediction model. In addition to ERP cost implementation, Internet of Things (IoT) has become a key that can automatize massive data collection, storage, and processing. The integration of IoT and ERP can help in reducing the implementation cost of ERP and human intervention. The usage of IoT devices can create closer connection between the supermarket and customers on a real-time basis [18].

The integration of IoT devices is a smart approach to address the issue that assists the store management to monitor the products, analyze real-time demand of the product and predict the future demand estimation. In smart city supermarket system, the internal and external factors indulge in the products sale and wastage of the products. The external factors include the locality, lifestyle, weather, calendar, and peek or off-peak hours. For instance, when there is a festival as marked on a local calendar, the estimated number of product type in location A could not have that significant sale in location B. Internal factors information that include product sale, product type, product price and date of expiry can be monitored by the IoT sensors. A huge volume of IoT data is processed in cloud computing environment, which is stated as conventional approach; where computational latency is inevitable in cloud computing platform. The edge/fog computing enhances the IoT data processing near to edge of the network that constrains the delay in data analysis and real-time decision making. The real-time decision making in supermarket system enhances the customer and management, quality of experience by demand prediction and reduces the waste parallelly. Therefore, we adopted Principal Component Analysis (PCA), K-means, and SARSA to design locality-based product demand prediction model and product distribution decision making for smart supermarket system. Thus, our goals can be summarized as follows:

- Design locality-based product demand prediction model using UL.
- Design low-demand product distribution decision making using RL.

The paper is organized as follows. Related works are presented in Section 2, and the system model of the paper is given in Section 3. The proposed method is described in Section 4. Performance evaluation is made in Section 5 followed by the concluding remarks in Section 6.

2. Related Works

ML technique has been used for predicting the product demand where three types of learning were explored, such as SL, UL, and RL. Conventional methods in demand prediction followed by the ML-based approach are studied in different literature. In supermarket system, demand for perishable and non-perishable product varies day-to-day with various internal and external influencing parameters. The demand forecasting of products plays a vital role such that over underestimation leads to out-of-stock situation and overestimation leads to products wastage with approaching shelf-time of product. The authors in [6] proposed Demand Support System (DSS) by applying extension of auto regressive integrated moving average that incorporates the external factors such as weather forecasting and support daily operational decisions. Poor demand estimation leads to products wastage in case of products with minimum shelf life or nearby expiry date needs to be sold with pricing decision to minimize the wastage. Similarly, the authors in [19] proposed a DSS model considering shelf life of a product and 432 urban consumer preferences for perishable products with varying offers. In supermarket system, the internal and external factors are the key to assess the product price. The authors in [20] compared three types of product pricing strategies on performance and sustainability of supermarkets by adopting manual customer survey from 100 supermarkets. The products on discount price or on sale indirectly affect the products, which are sold in respective supermarkets. The authors in [21] stated this type of condition as cannibalization effect.

Based on the supermarket scanner historical data and sale promotion schedule for same time period, the authors analyzed that putting price on promotion reflects in sales cutoff non-promoted products. Moreover, the perishable and non-perishable products need to be put on sale to re-order in the next inventory that is calculated by estimating the order quantity model [22]. Furthermore, the integration of IoT enables the supermarkets in various locations in a smart city to monitor and make an intelligent decision of the product sales. The food supply chain assists the supermarkets by transporting perishable and non-perishable products. The perishable products reach from farm to supermarket with the connectivity of products logistics with cyber world/Internet that is termed as Internet of Perishable Logistics (IOPL) [23]. Alongside logistics IoT devices are embedded in the supermarket system with the vast integration of smart IoT sensors such as shelf sensor, motion sensor, RFID sensor, and so on. For instance, RFID tagged to a product provides a unique identity and generates sensor data, which helps in real-time tracking and monitoring the quality of product. The authors in [8] adopted block chain technology for real-time monitoring of products in food supply chain. Moreover, the data generated from the supermarkets need to be collected, processed and analyzed for the prediction and decision making of products pricing and inventory moving.

Conventional cloud computing provides a vast amount of computing resources. However, it has limits in addressing the streaming IoT data generated from the remote areas. There are largely deployed supermarkets in various localities of smart city embedded with IoT devices and distributed fog computing platform assists them in providing the computational platform. For instance, in [24], IoT data generated by industrial Internet of things are processed in fog computing platform by adopting service popularity-based resource partitioning method. Although the cloud computing platform comprises of vast amounts of computing resources, the computation latency is inevitable. Whereas, the fog computing nodes address the computation latency with local data processing and analysis. Moreover, the authors in [25] briefly illustrated a new paradigm of fog computing platform, where IoT data processing and analysis would be carried out on the near edge fog computing platform to provide reliability, availability, serviceability, hierarchy, programmability and agility. The fog computing platform has comparatively limited computing resources that need to be intelligently scheduled to minimize the latency, and computing resource consumption. Thus, the authors in [26] proposed game theory based intelligent multi-criteria IoT-Fog scheduling approach to assist the end user IoT devices. Comparison metrics such as CPU, RAM and bandwidth of fog computing nodes were used by comparing the centralized computing, Min-Min, and Max-Min approaches. They observed that the distributed fog computing provides low latency services with intelligent computing resource use. Thus, the proposed supermarket environment generates data in various locations of smart city environment, where these data could be processed using Artificial Intelligence enabled fog computing platform.

In particular store, the sale of a product is influenced by many internal factors that include the price among similar products and special promotions. Based on these factors, the authors analyzed and developed a prediction model of sugar-sweeten beverages for the public health guidance [15] by applying B-DTR and DNN. The beverage demand prediction was built by using transaction records data from 44 stores, which is composed of three large-scale groceries stores. Similarly, prediction of grocery items demand is developed in this work [14]. The prediction model was designed through data preprocessing, training set modification, and training process using the k-fold cross validation. By doing so, two SL algorithms were adopted, such as RF and Keras Neural Network on Kaggle dataset from South American groceries stores. However, internal factors are not enough to predict the product demands. Thus, daily demand prediction model was developed using Artificial Neural Network (ANN) and Gradient-Boosted Decision Tree by considering external factors include special calendar days [27]. Besides, the transaction data of 8 products from 141 stores were collected from 2014 until 2017 for training purpose. Moreover, in different point of view, the authors concerned about the overstock and stock outs of a product in store. These problems may cause a customer move or purchase same product from other competitor. Therefore, product demand prediction was developed by designing a few steps such as data collection, data cleaning, data enriching, data a priori, dimensionality reduction, clustering, prediction models, and final decision [28].

To train the model, different methods, including Deep Learning methodology, Support Vector Machine (SVM), and time series analysis model, were used on real sales data from

7888 products in SOKMarket. On the other hand, available data of new product is limited in a store. Due to this reason, the authors designed their product demand prediction model using transfer learning [29]. From Austrian food retailing data, 26 features, six different types including date, price/promotion, identity, lag, aggregate, and external features, were used along with the DNN algorithm to train the prediction model. Overstock or product waste may cause big problem in managing the retail inventory in a single warehouse of stores. Followed by unpredictable demand in different locations of the stores, important questions may arise such as when and where to supply the products. Due to these problems, the authors came up with Multi-product and Multi-node inventory management using Multi-agent Actor-Critic (A2C) RL algorithms. The model was built using simple environment, which is composed of a single warehouse, three stores, and brick and mortar stores open dataset from Kaggle [30]. Similar approach is applied in placing the products locally within a store using Deep Q-Learning Network (DQN) [31].

3. System Model

The smart city supermarket data monitoring system is built on a brisk computation environment trending as fog computing. A smart city is virtually segregated into a set of regions ID which is denoted as $R = \{r_1, r_2, \ldots, r_{|R|}\}$. Let $S = \{s_1, s_2, \ldots, s_{|S|}\}$ denote as a set of supermarkets in all region R. Accordingly, a supermarket s_i belongs to specific region r_j can be represented as s_i^j, where $i = \{1, 2, \ldots, |S|\}$ and $j = \{1, 2, \ldots |R|\}$. It is to be noted that all supermarkets in different regions such as Carrefour or Welcome are from the same chain. We assumed that each supermarket s_i^j is equipped with three types of IoT devices such as Temperature (Temp), RFID, and Barcode Reader (BR) sensors, which can be denoted by $\{Temp^{i,j}, RFID^{i,j}, BR^{i,j}\}$, respectively. Data is collected simultaneously through these IoT sensors in all supermarkets located at different regions of a city. From the collected data, we designed and extracted two types of features termed as internal and external factors. It is assumed that the internal features extracted from the BR sensors is composed of transaction information of the customers that contains date, time, supermarket ID, region ID, purchased product category, types, unit price and total amount of the transaction. Besides, the available quantity of products can be extracted from the RFID sensors. Information related to weather and calender events are considered to be external features. For example, information about rainy and sunny days are collected using Temp sensor and Christmas and national holidays are taken from the transaction dates. These data are collected at respective fog data centers f_k^j in particular region for processing and analysis purposes through Wi-Fi/4G/5G communications, where $k = \{1, 2, \ldots, |R|\}$. Data preprocessing, data analysis and decision making are performed in fog data centers, which act as the agents. Data analysis is carried out using K-means UL for clustering with three outcomes such as high, medium, and low-demand product clusters. The predicted outcomes include high and low-demand product which are used further for low-demand product distribution decision making using SARSA RL based. Furthermore, the prediction and decision making outcomes can be delivered to end-users such as customer, store manager, and stock supplier in form of applications. The overview of our system model is shown in Figure 1.

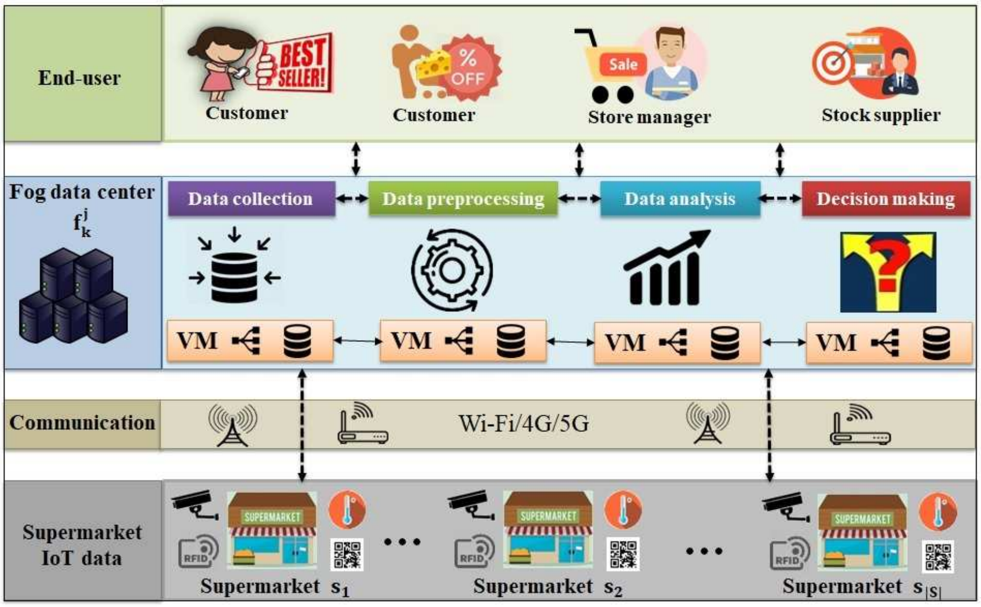


Figure 1. Proposed system model.

3.1. Fog Computing Environment

Fog computing enabled supermarket system assists to collect IoT data distributively from supermarkets in respective region. Fog computing enables low latency computing near to edge of the network in smart city supermarket system. Fog data center process the IoT data from each supermarket and shares the information in fog data center network. Our contribution is to reduce wastage of products by predicting high, medium, and low-demand product and then making a decision to distribute the set of low-demand products from source supermarket to high-demand product destination supermarkets in same or different regions. It is noted that low and high-demand product are the same. The fog data center assists in collecting, preprocessing, and analyzing real-time supermarket IoT data to predict the product demand based on internal and external features. The fog data center ML-based feature products demand clustering assists in predicting high, medium and low-demand product clusters in a region/locality. The predicted demand clustering results are used in RL based decision making model to distribute the low-demand products in to product high-demand supermarket. Thus, fog computing environment can enhance quality of service to end users' by analyzing the products information from same supermarket chain; compare and provide the high demand (best sale) product in the locality. For instance, in particular region r_j, a customer requests a set of popular or high-demand products H_p to the respected fog data center f_k^j. After receiving the request, preprocessing and analysis of data will be carried out in fog data center where the product information is retrieved from available multiple supermarket products data in the region f_k^j. From which, the analyzed results are sent to end user that contain top three high-demand products in supermarket s_i region r_j. Simultaneously, low-demand product distribution decision will be forwarded to store manager, as shown in Figure 2. By doing so, product wastage can be prevented and reduced by moving the low-demand products in supermarket source to high-demand products in destination supermarkets which have the same products either in the same or different regions.

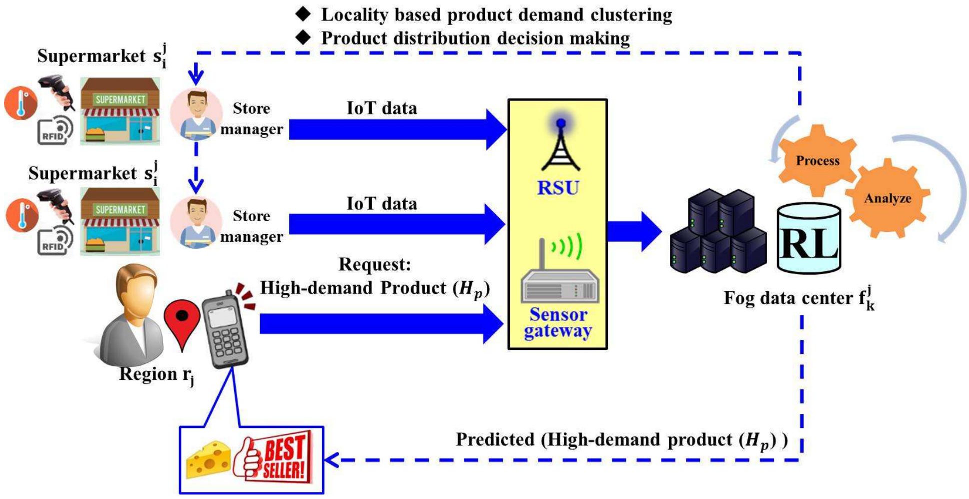


Figure 2. Prediction and decision making in fog computing environment.

3.2. Features Designing

To design the locality-based product demand clustering model, we designed and modeled some features, which are categorized as internal and external factors. These factors are extracted from the collected data at supermarket.

3.2.1. Internal and External Factors

Initially, store manager will order out-of-stock products to be supplied at time (t) in each store and region. Accordingly, stock supplier will supply and send all requested products. Right after, all products are arranged in the supermarket inventory, products are placed in the pre-designated shelfs. This information is recorded through the RFID sensors embedded in each product of a supermarket. For a particular product, let $RFID_p^{i,j}$ be the total number of available quantity of product p, where each supermarket s_i^j sells a set of products $p = \{1, 2, \ldots, |p|\}$. Thus, $|p|$ is the total number of products in a supermarket. It is assumed that each product p has several physical informations such as Size (Se) that includes Grams (Gr), Kilograms (Kg), Milliliters (Ml), and Liters (Lt), date of expiry (E) in terms of $YYYY/MM/DD$, and product price (Pr). Extracted from the BR sensors, we consider three internal features such as product line, product category, product price, which are denoted as a tuple $< C, N, Pr >$. For simplicity, we decided the predefined product line as a set of $C = \{HB, E, HL, FB, ST, F\}$, which stands for Health and Beauty (HB), Electronics (E), Health and Lifestyle (HL), Food and Beverage (FB), Sport and Travel (ST), and Fashion (F), respectively. For the product category, we consider perishable (PR) and non-perishable (NPR), and product price is defined by the unit price of a product, for instance unit/price = 1/300 NTD (i.e., New Taiwan Dollar). Furthermore, historical data of purchase is considered to be another internal factor, which can be modeled by considering the historical data from the supermarket transactions. Taking all considered features, we designed the internal factor as a row vector I, which can be seen in Equation (1).

$$I = \begin{bmatrix} D & s_i & r_j & C & N & Pr & RFID_p^{i,j} & Tot \end{bmatrix} \tag{1}$$

where D, s_i, r_j, C, N, P, Q, Tot are transaction date, supermarket ID, region ID, product line, product category, product price, total number of available quantity of product, and Total transaction, respectively. Moreover, in complex real-world scenario, internal factor features can be affected by customer behavior due to changing environmental conditions. Therefore, we considered an external factor as a row vector E of two elements such as calendar event X and weather information Y (Equation (2)) collected from the Temp sensor. Calendar event and weather information can be extracted from the transaction date D and Temp sensor in local region r_j, respectively. For calendar event, we considered public holidays and festivals that occurs on transaction date D, whereas for weather information, sunny and rainy were considered that happens in particular region r_j. Let X and Y be the binary value for calendar event and weather information, respectively. Where $X = 1$, if there is at least an event, otherwise, $Y = 1$, if it is sunny, else it is rainy. Hence, as shown in Equation (3), the union of vector I (Equation (1)) and E (Equation (2)) is defined as the final features row vector δ as given in Equation (4).

$$E = \begin{bmatrix} X & Y \end{bmatrix} \tag{2}$$

$$\delta = I \cup E \tag{3}$$

$$\delta = \begin{bmatrix} X & s_i & Y & C & N & Pr & Q & Tot \end{bmatrix} \tag{4}$$

4. Proposed Method

Based on the collected dataset, we designed and derived the features. We applied the data processing techniques including data conversion and standardization. In order to find the correlation of all preprocessed features, distribution of data sample, and influential parameters that have high variance, we selected PCA. Taking the selected principal components that have high data variance, locality-based product demand clustering is designed. Although SL algorithms for classification and regression tasks such as RF, SVM, etc., and UL algorithms for clustering and association tasks such as K-means, Apriori, etc., are available, the selection of algorithms needs to be chosen carefully based on the input data to get the desired outputs. Labeled data is not available as input data in our proposed model, and therefore UL algorithm is selected to find the patterns among the given input data. We designed a set of groups as outputs such as high, medium, and low-demand product clusters. Hence, we applied K-means UL algorithm to accomplish the clustering tasks. By doing so, K-means algorithm can assign which sample belongs to which cluster. This Integration of PCA and K-means algorithms is the solution for locality-based product demand clustering model that can reduce the dimension, decrease the computation cost, and optimize the number of clusters.

Moreover, high and low-demand product outcomes are used further for developing the product distribution decision making using SARSA. Overview of the proposed method is shown in Figure 3.

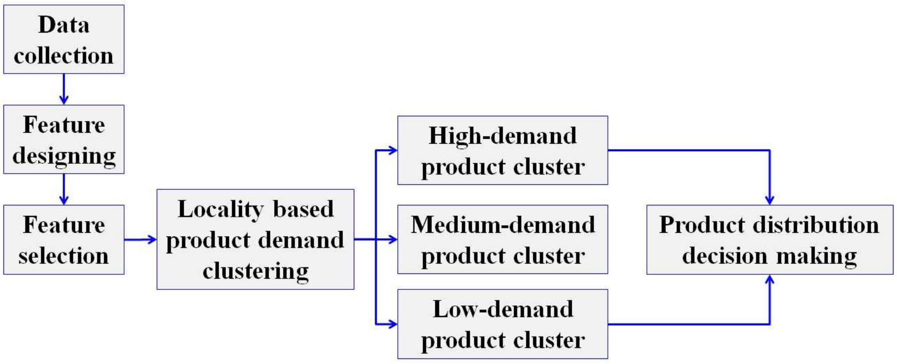

Data collection
Feature designing
Feature selection
Locality based product demand clustering
High-demand product cluster
Medium-demand product cluster
Low-demand product cluster
Product distribution decision making

Figure 3. Proposed method work flow.

4.1. Feature Selection Using PCA

As we deal with high dimensionality of the designed features, selection and usage of the most influential features are highly essential. Therefore, we adopted PCA feature selection technique. The formal description of PCA algorithm is given in Algorithm 1. Though this technique is mostly used to reduce the high dimensionality of dataset, the importance of each feature can be found. First, any categorical feature from the set of vector δ, needs to be converted into numerical features. For instance, we converted a set of product line $\{HB, E, HL, FB, ST, F\}$ into a set of numbers $\{1, 2, 3, 4, 5, 6\}$. Second, we applied standardization method in to the whole features. Third, mean and covariance matrix are calculated among the standardized features. Based on the correlation coefficient value, we calculated and found the eigenvalue ($Eval$) and eigenvectors ($Evec$), and then, we returned $\rho = \{a_1, a_2, \ldots, a_{|\rho|}\}$ the set of principal components. Accordingly, we calculated and analyzed the explained variance ratio. We selected the highest variance ratio of principal component and defined the most influential features. Hence, we used the selected features β for further analysis.

Algorithm 1 Feature selection using PCA.

```
Input: Matrix $\delta = \begin{bmatrix} X & s_i & Y & C & N & Pr & Q & Tot \end{bmatrix}$
Parameter:
$\rho$: A set of principal components, $\rho = \{a_1, a_2, \ldots, a_{|\rho|}\}$, $a_p \in \rho$
$\bar{x}, \bar{y}$: Mean value of each feature which is an element of Matrix $\delta$
$X_i, Y_i$: Data point $i$ that belongs to features $X$ and $Y$ which are elements of Matrix $\delta$
$N$: Total number of available features
$\beta$: A set of the most influential features
Process:
1: Data standardization (Matrix $\delta$)
2: Compute mean ($\bar{x}$, $\bar{y} \in$ Matrix $\delta$)
3: Compute covariance Matrix $cov(X, Y) = \frac{1}{N} \sum_{i=1}^{N} (X_i - \bar{x})(Y_i - \bar{y})$
4: Calculate $Eval$ (Matrix $\delta$)
5: Calculate $Evec$ (Matrix $\delta$)
6: Descending sort ($Eval$)
7: Calculate explained variance ratio ($\rho$)
8: Select $a_p$ which has the highest explained variance
9: Find the most influential features $\beta$ from selected $a_p$
10: Return Matrix $\beta$
```

4.2. Locality Based Product Demand Clustering

Based on the set of most influential features Matrix $\beta = \{b_1, b_2, \ldots, b_{|\beta|}\}$, we developed a locality-based product demand clustering using K-mean algorithm. Initially, we defined the $K = 3$ numbers of clusters for low (L_p), medium (M_p), and high-demand (H_p) product. Second, we randomly chosen the cluster centroid for each cluster. Third, using Euclidean distance, we calculate the distance between each centroid c_j with data point c_i where $i, j \in \beta$, as shown in Equation (5). Forth, we assigned each data point c_i into a cluster which has closets distance. Fifth, we updated the centroid. Until convergence where no more change in the cluster centroid, step two until fifth is repeated.

$$Dist(c_i, c_j) = \sqrt{\sum_{i=1}^{\beta} (c_i - c_j)^2} \tag{5}$$

4.3. Reinforcement Learning Based Product Distribution

In this section, we designed RL based product distribution model. RL is a learning process of an agent that takes an action a_t in the state s_t, where the agent is considered to be a fog data center. By doing so, agent will get a reward r_t and move into the next state s_{t+1}. These processes can be modeled using Markov Decision Process (MDP) and are presented as a tuple <s_t, a_t, r_t, s_{t+1}>. In this work, we adopted SARSA that is an on-policy algorithm, where an agent can learn from the environment by taking an action a_t in a state s_t and receives a reward r_t. Next, from the state s_t, agent again takes the next action a_{t+1} by following the current policy (ϵ-Greedy) used in a_t, and moves to the next state s_{t+1}. Thus, the tuples can be modified into <$s_t, a_t, r_t, s_{t+1}, a_{t+1}$> in SARSA. It is used to update and optimize the policy by learning and updating the Q-value, as shown in Equation (6).

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{6}$$

where α and γ are learning rate and discount factor, respectively in the range $[0,1]$. The locality-based product demand clustering model outputs include low and high-demand products, which are used in the product distribution decision making. As per our learning method hosted in fog data center, we designed the state s_t, action a_t, and reward r_t. Let s_t = $<r_j, s_i, L_p^{i,j}, \sigma>$ be a state, which is composed of region ID r_j, supermarket ID s_i, a set of low-demand products L_p in a supermarket i located in the region j, and a matrix σ, which is composed of high-demand product H_p of supermarket k. Based on the current state in a particular region, an agent will take an action, which is composed of product distribution decision for all supermarkets that belong to a fog data center located in that region. Let $A_{j,k} = \{D^{j,k}_{s_1}, D^{j,k}_{s_2}, \ldots, D_{s^{j,k}_{|S|}}\}$ be the set of actions, where fog node data center agent k sends an action D that contains low and high-demand product informations of the supermarket s_i in a region j. On forwarding and applying a set of actions $A_{j,k}$, respective supermarket s_i (i.e., Supermarket manager) will execute the decision by distributing those low-demand products to decide a supermarket s_k, where $k = \{1, 2, 3, \ldots\}$ that has the same high-demand product. By taking an action in state s_t, reward r_t is updated for an agent. We designed the reward based on the condition that occurs in the supermarkets of a smart city. For instance, if any supermarket that has low-demand product is updated to high-demand products, then a reward $r_t = 1$ is updated, otherwise $r_t = -1$. It is to be noted that the distributed low L_p to high-demand products H_p are the same product $L_p == H_p$. Equation (7) shows the designed reward function.

$$r_t = \left\{\begin{matrix} +1 & L_p == H_p \\ -1 & L_p! = H_p \end{matrix}\right\} \tag{7}$$

During the training, our goal is to optimize the policy π by taking a set of action $A_{j,k} = a_t$ in the current state s_t and maximizing the cumulative reward received in the state s_t in each episode $t = 1$ until total episode T, $Max\ \mathbb{E}[\sum_{t=1}^{T} r_t(s_t, a_t)]$. Overall, the formal description of proposed prediction model for product demand RL-based is described in Algorithm 2.

Algorithm 2 Product distribution decision making.

```
For each supermarket $s_i$
Input: High-demand product $H_p$ and Low-demand product $L_p$ clusters
Parameter: $\alpha$, $\gamma$, Policy $\pi$, Reward $r_t$
Initialize $Q(s_t, a_t)$
Process:
For episode 1 to M do
  Initialize state $s_t$
    Choose action $a_t$ from $\epsilon$-Greedy
  Repeat for each step of episode
    Observed $s_{t+1}, r_t$
    Choose next action $a_{t+1}$ based on policy $\pi$ derived from $\epsilon$-Greedy
    Calculate Equation (6)
    Update $Q(s_t, a_t)$
  Step END
Episode END
```

5. Performance Evaluation

5.1. Dataset, Hardware and Software

We considered supermarket transaction data from Kaggle open data [32]. The dataset was collected for three months from three branches of supermarkets. The collected data was composed of 16 features including Branch, City, Customer type, Gender, Product line, Unit price, Quantity, Tax 5%, Total, Date, Time, Payment, cogs, gross margin percentage, gross income, and Rating, and 1000 total number of data points without any missing value. Out of 16 features, six features are categorical and 10 features are numerical types of data. Moreover, for developing ML model, specific hardware and software were used. For hardware, GPU GeForce GTX 1070 Ti system, 418.56 NVIDIA-SMI, and 32 GB memory were used, and Ubuntu 18.04.2 Operating system, Scikit-learn 0.23.2 ML library including other python libraries Numpy 1.19.1, Pandas 1.0.5, Matplotlib 3.3.0, and Imageio 2.5.0.

5.2. Simulation Environment

We deployed a smart city environment into a grid world with a size of $Row \times Column$ = 500 × 500, where each element of grid world can be denoted as a set of region IDs $\{R = r_1, r_2, \ldots, r_{Row \times Column}\}$ and each supermarket is located randomly in each region ID. Based on our dataset, we considered three supermarket branches, where one supermarket acts as source supermarket and the rests are destination supermarkets. We randomly assigned the location of these supermarkets and other neighbor supermarkets in to different regions. Figure 4 shows the example of our designed simulation environment.

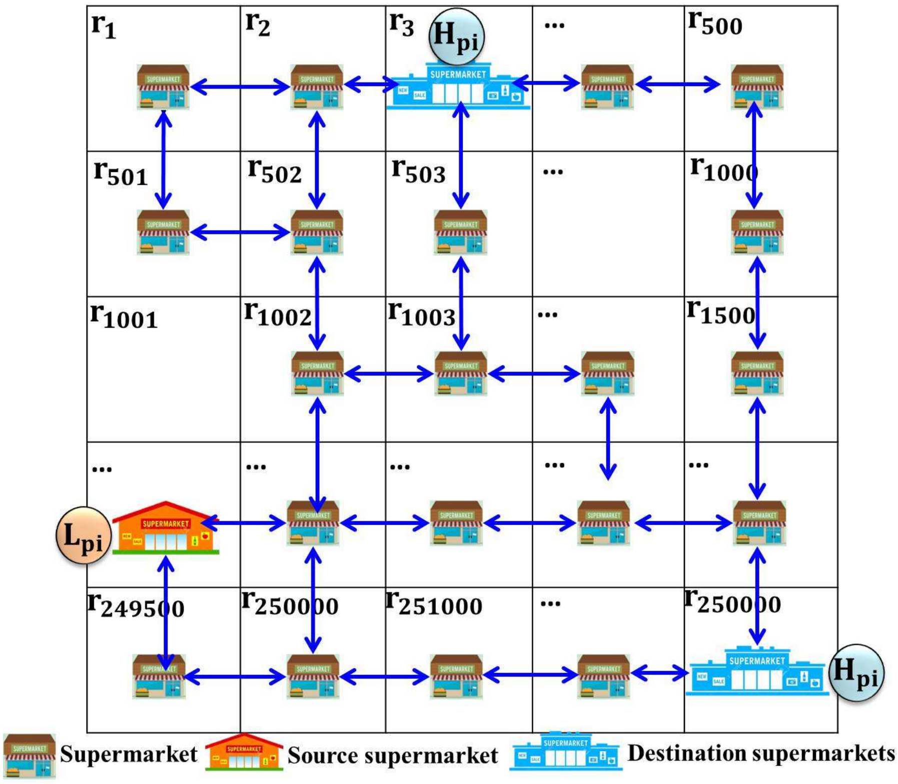
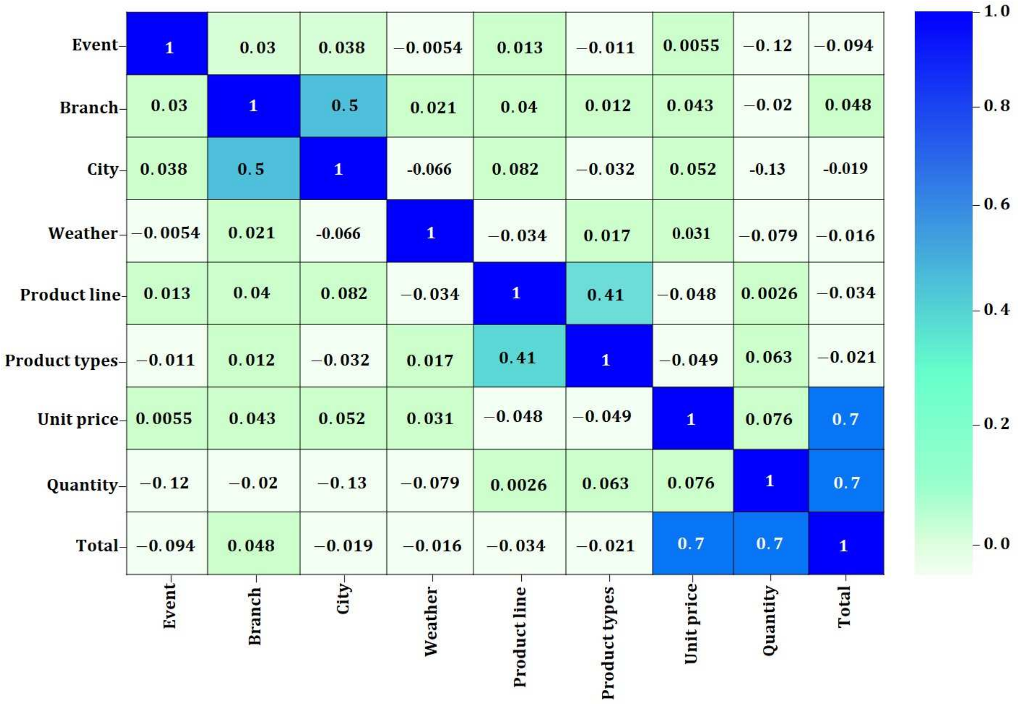


Figure 4. Simulation environment for RL.

5.3. Simulation Results: Locality Based Product Demand Clustering

We performed a combination of PCA and K-mean (KPCA) clustering to group locally the product demand based on the designed features. Initially, we calculated the features correlation matrix of all designed features as shown in Figure 5. From the matrix, we analyzed that the relationship between any pair of features, for instance the correlation between branch and city features is 0.5. Then, we performed the PCA and selected the principal component. The selection criteria of principal component is defined by the "elbow" in scree plot and ratio of data variance. As shown in Figure 6a, we plotted the scree plot that comprises the eigenvalues along the Y-axis and total number of principal components along the X-axis. It is observed that the "elbow" is at the 2nd component. Besides, we also analyzed the variance ratio of all components as shown in Figure 6b. Finally, we selected total number of principal component = 1, which has 30% of data variance in our locality-based product demand clustering.



	Event	Branch	City	Weather	Product line	Product types	Unit price	Quantity	Total
Event	1	0.03	0.038	−0.0054	0.013	−0.011	0.0055	−0.12	−0.094
Branch	0.03	1	0.5	0.021	0.04	0.012	0.043	−0.02	0.048
City	0.038	0.5	1	-0.066	0.082	−0.032	0.052	-0.13	-0.019
Weather	−0.0054	0.021	-0.066	1	−0.034	0.017	0.031	−0.079	−0.016
Product line	0.013	0.04	0.082	−0.034	1	0.41	−0.048	0.0026	−0.034
Product types	−0.011	0.012	−0.032	0.017	0.41	1	−0.049	0.063	−0.021
Unit price	0.0055	0.043	0.052	0.031	−0.048	−0.049	1	0.076	0.7
Quantity	−0.12	−0.02	−0.13	−0.079	0.0026	0.063	0.076	1	0.7
Total	−0.094	0.048	−0.019	−0.016	−0.034	−0.021	0.7	0.7	1

Figure 5. Coefficient matrix of designed features.

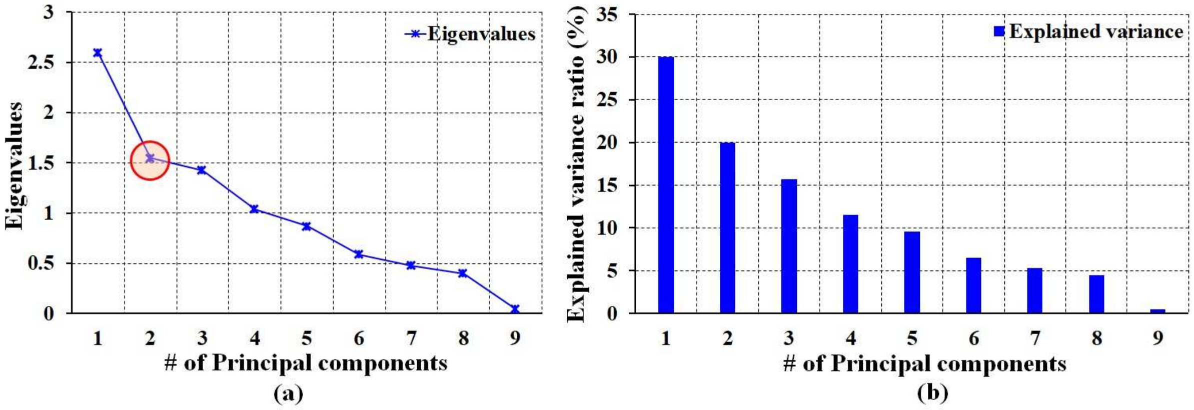


Figure 6. Principal component selection: (**a**) Scree plot; (**b**) Explained variance ratio graph.

We conducted our clustering model using K-means. As per our requirement, we set $K = 3$ as total number of clusters and random state = 0. By applying Euclidean distance, we calculated the distances among all data points and formed the clusters. To justify the quality of formed clusters, we measured and evaluated those clusters using Silhouette score. Our proposed method KPCA can well separate the data points into each cluster with higher similarities. As shown in Figure 7, quality of the clustering is closer to 100%, which is 63% as compared to K-means without PCA, which is 20%.

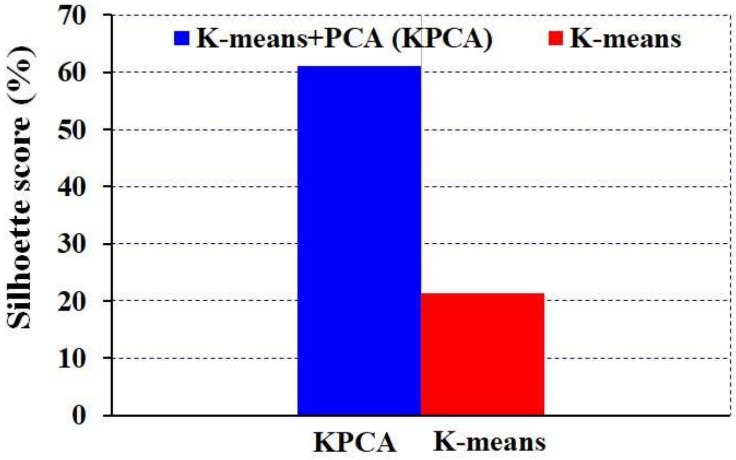


Figure 7. Silhouette score comparison between a combination of k-means and PCA and k-means.

5.4. Simulation Results: Product Distribution Decision Making

We performed the product distribution decision making by considering two clusters such as low and high-demand data points. We took one to many scenario for our environments, where one source supermarket will distribute the low-demand products to many destination supermarkets, which have high-demand products. As described in our simulation environment, we conducted the simulation using proposed adopted SARSA and have compared with two related works such as inventory management using Actor-Critic algorithm [30] and products replacement within a store using DQN [31]. In addition, we also compared our results with other RL algorithms including Q-Learning and Monte-Carlo. As per our goal is to maximize the cumulative reward received by an agent in each episode, we analyzed the maximum rewards in different total number of episodes. As shown in Figure 8, all algorithms are having similar maximum cumulative total rewards at initial number of episode, for instance at episode number = 600. However, when the number of episodes is increased particularly from 700 total number of episodes, total maximum reward received by the agent using Monte-Carlo and Actor-Critic starts decreasing. Whereas, our proposed adopted SARSA gradually outperforms over Q-Learning and DQN, when the total number of episodes is increasing. It is also observed that Q-learning and DQN share similar performance as both algorithms have the same calculation method for updating the Q-value functions.

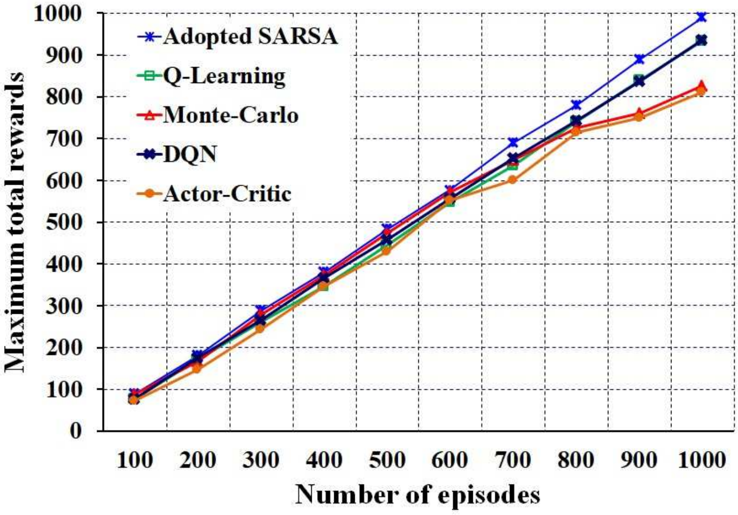


Figure 8. Maximum cumulative reward comparison.

From the maximum cumulative rewards, we calculated the average reward with different number of episodes. As shown in Figure 9, our proposed adopted SARSA has higher average cumulative rewards as compared to other algorithms. For all episodes, our proposed method achieved reward with an average value of 0.454, whereas reward value of 0.377, 0.418, 0.3952, and 0.351 achieved by Q-Learning, Monte-Carlo, DQN, and Actor-Critic algorithm, respectively.

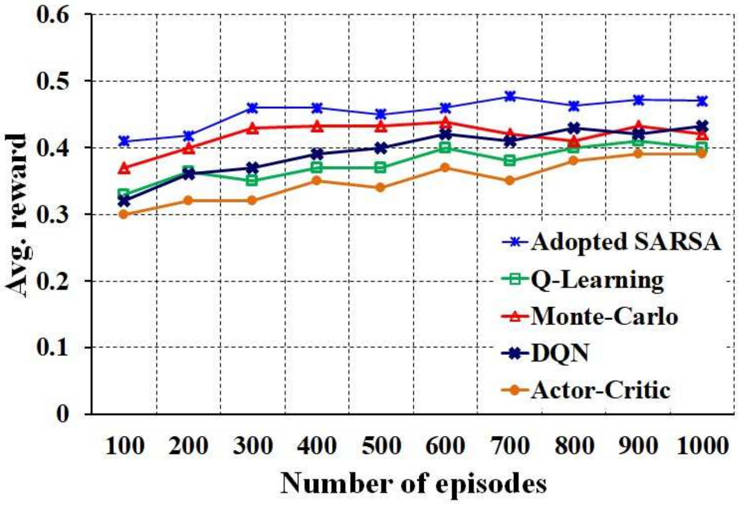


Figure 9. Average cumulative reward comparison.

We also observed the total execution time in unit of minutes with different total number of episodes. It can be observed that the learning process using our proposed adopted SARSA has less execution time as compared to Q-Learning, Monte-Carlo, DQN, and Actor-Critic as shown in Figure 10 although total number of episodes is increasing. It is also observed that our proposed technique has higher execution time in 700 total number of episode as compared to Monte-Carlo.

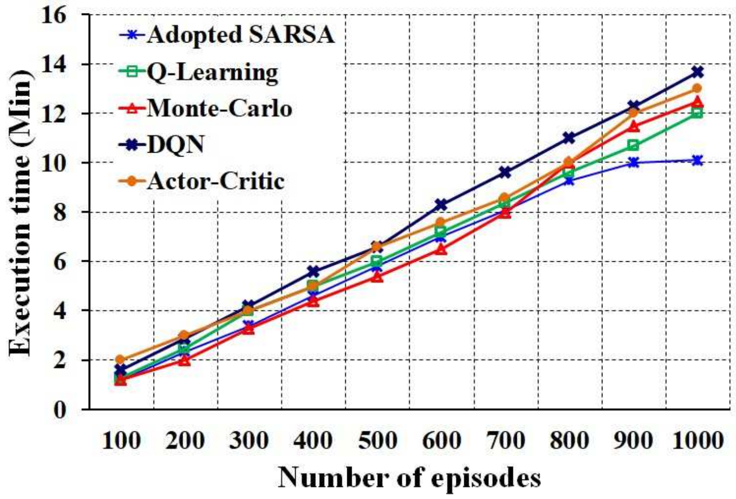


Figure 10. Execution time comparison.

6. Conclusions

In this paper, locality-based product demand prediction and decision making in fog computing enabled smart city supermarket system is designed by applying UL and RL techniques, where IoT data processing, analysis and decision making are carried out in the fog computing data centers. We applied PCA and K-means algorithms of the unsupervised learning, and SARSA of the reinforcement learning. The data generated by the IoT devices is integrated with the supermarket system. The internal and external supermarket environment are collected and analyzed in the distributed fog computing environments in each regions of smart city. The primary goal is to minimize the wastage of the perishable and non-perishable products by predicting and distributing the low-demand product in to high-demand product. By doing so, we proposed clustering model and decision making model. We clustered the low, medium, and high-demand products. Based on low and high-demand products cluster data, we made a decision making model to decide how agent can move the low-demand products from the single source supermarket to multiple destination supermarkets which have high-demand products. Based on our simulation environment and results, we proved that a combination of UL and RL method works well in supermarket scenario, where we achieved Silhouette score $\geqq$ 60% and highest maximum cumulative reward and average reward with less execution time as compared to Q-Learning, Monte-Carlo, DQN, and Actor-Critic.

Author Contributions: G.N., D.D.O. and P.K.S. conceived the idea and designed the algorithms, experiments, analyzed the data, and wrote the paper. G.N. performed the experiments on fog computing. D.D.O. performed the artificial intelligence related experiments. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grant 109-2221-E-182-014 and in part by Chang Gung Medical Foundation, Taiwan under Grant CMRPD 2J0141 and CMRPD 2J0142.

Acknowledgments: We are thankful to Healthy Aging Research Center, Chang Gung University for supporting the GPU platform for our analysis from the Featured Areas Research Center Program within the Framework of the Higher Education Sprout Project (EMRPD1I0491) by the Ministry of Education (MOE), Taiwan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lin, C.; Zhao, G.; Yu, C.; Wu, Y.J. Smart city development and residents' well-being. *Sustainability* **2019**, *11*, 676. [CrossRef]
2. Wang, W. Everyday practice in the high density, volumetric Hong Kong: Ambiguity, intensity and life between interfaces. *Cities* **2020**, *96*, 102462. [CrossRef]
3. Wu, S.M.; Chen, T.C.; Wu, Y.J.; Lytras, M. Smart cities in Taiwan: A perspective on big data applications. *Sustainability* **2018**, *10*, 106. [CrossRef]
4. Global Agricultural Information Network. Available online: https://www.fas.usda.gov/ (accessed on 27 September 2020).
5. Food and Agriculture Organization, United Nations. Extent of Food Losses and Waste. Available online: http://www.fao.org/3/mb060e/mb060e02.pdf (accessed on 27 September 2020).
6. Huber, J. Alexander Gossmann, and Heiner Stuckenschmidt. Cluster-based hierarchical demand forecasting for perishable goods. *Expert Syst. Appl.* **2017**, *76*, 140–151. [CrossRef]
7. Parada, R.; Palazón, A.; Monzo, C.; Melià-Seguí, J. RFID Based Embedded System for Sustainable Food Management in an IoT Network Paradigm. *Future Internet* **2019**, *11*, 189. [CrossRef]
8. Mondal, S.; Wijewardena, K.P.; Karuppuswami, S.; Kriti, N.; Kumar, D.; Chahal, P. Blockchain inspired RFID-based information architecture for food supply chain. *IEEE Internet Things J.* **2019**, *6*, 5803–5813. [CrossRef]
9. Pan, S.; Zhong, R.Y.; Qu, T. Smart product-service systems in interoperable logistics: Design and implementation prospects. *Adv. Eng. Inform.* **2019**, *42*, 100996. [CrossRef]
10. Lee, J.; Lee, J. Hierarchical mobile edge computing architecture based on context awareness. *Appl. Sci.* **2018**, *8*, 1160. [CrossRef]
11. Martinez, I.; Jarray, A.; Hafid, A.S. Scalable Design and Dimensioning of Fog-Computing Infrastructure to Support Latency Sensitive IoT Applications. *IEEE Internet Things J.* **2020**, *7*, 5504–5520. [CrossRef]
12. Singh, P.; Nayyar, A.; Kaur, A.; Ghosh, U. Blockchain and Fog Based Architecture for Internet of Everything in Smart Cities. *Future Internet* **2020**, *12*, 61. [CrossRef]
13. Thakkar, H.K.; Dehury, C.K.; Sahoo, P.K. MUVINE: Multi-Stage Virtual Network Embedding in Cloud Data Centers Using Reinforcement Learning-Based Predictions. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1058–1074. [CrossRef]
14. Vairagade, N.; Logofatu, D.; Leon, F.; Muharemi, F. Demand Forecasting using Random Forest and Artificial Neural Network for Supply Chain Management. In Proceedings of the International Conference on Computational Collective Intelligence, Hendaye, France, 4–6 September 2019; Volume 11683, p. 328.

15. Lu, X.H.; Mamiya, H.; Vybihal, J.; Ma, Y.; Buckeridge, D.L. Guiding public health policy by using grocery transaction data to predict demand for unhealthy beverages. In Proceedings of the International Workshop on Health Intelligence, Hawaii, HI, USA, 27 January–1 February 2019; Volume 843, p. 169.
16. Neelakantam, G.; Onthoni, D.D.; Sahoo, P.K. Reinforcement Learning Based Passengers Assistance System for Crowded Public Transportation in Fog Enabled Smart City. *Electronics* **2020**, *9*, 1501. [CrossRef]
17. Rouhani, S.; Ravasan, A.Z. ERP success prediction: An artificial neural network approach. *Sci. Iran.* **2013**, *20*, 992–1001. [CrossRef]
18. Tavana, M.; Hajipour, V.; Oveisi, S. IoT-based enterprise resource planning: Challenges, open issues, applications, architecture, and future research directions. *Internet Things* **2020**, *11*, 100262. [CrossRef]
19. Fikar, C.; Mild, A.; Waitz, M. Facilitating consumer preferences and product shelf life data in the design of e-grocery deliveries. *Eur. J. Oper. Res.* **2019**. [CrossRef]
20. Agbaeze, E.; Chiemeke, M.N.; Ogbo, A.; Ukpere, W.I. Impact of Pricing Practice Management on Performance and Sustainability of Supermarkets in the Urban Area of Enugu State, Nigeria. *Sustainability* **2020**, *12*, 6019. [CrossRef]
21. McColl, R.; Macgilchrist, R.; Rafiq, S. Estimating cannibalizing effects of sales promotions: The impact of price cuts and store type. *J. Retail. Consum. Serv.* **2020**, *53*, 101982. [CrossRef]
22. Dobson, G.; Pinker, E.J.; Yildiz, O. An EOQ model for perishable goods with age-dependent demand rate. *Eur. J. Oper. Res.* **2017**, *257*, 84–88. [CrossRef]
23. Pal, A.; Kant, K. Internet of perishable logistics: Building smart fresh food supply chain networks. *IEEE Access* **2019**, *7*, 17675–17695. [CrossRef]
24. Li, G.; Wu, J.; Li, J.; Wang, K.; Ye, T. Service Popularity-Based Smart Resources Partitioning for Fog Computing-Enabled Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4702–4711. [CrossRef]
25. Tange, K.; De Donno, M.; Fafoutis, X.; Dragoni, N. A systematic survey of industrial Internet of Things security: Requirements and fog computing opportunities. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2489–2520. [CrossRef]
26. Arisdakessian, S.; Wahab, O.A.; Mourad, A.; Otrok, H.; Kara, N. FoGMatch: an intelligent multi-criteria IoT-Fog scheduling approach using game theory. *IEEE/ACM Trans. Netw.* **2020**, *28*, 1779–1789. [CrossRef]
27. Huber, J.; Stuckenschmidt, H. Daily retail demand forecasting using machine learning with emphasis on calendric special days. *Int. J. Forecast.* **2020**, *36*, 1420–1438. [CrossRef]
28. Kilimci, Z.H.; Akyuz, A.O.; Uysal, M.; Akyokus, S.; Uysal, M.O.; Atak, Bulbul, B.; Ekmis, M.A. An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain. *Complexity* **2019**, *2019*, 9067367. [CrossRef]
29. Karb, T.; Kühl, N.; Hirt, R.; Glivici-Cotruta, V. A network-based transfer learning approach to improve sales forecasting of new products. *arXiv* **2020**, arXiv:2005.06978.
30. Sultana, N. N.; Meisheri, H.; Baniwal, V.; Nath, S.; Ravindran, B.; Khadilkar, H. Reinforcement Learning for Multi-Product Multi-Node Inventory Management in Supply Chains. *arXiv* **2020**, arXiv:2006.04037.
31. Jenkins, P.; Wei, H.; Jenkins, J.S.; Li, Z. A Probabilistic Simulator of Spatial Demand for Product Allocation. *arXiv* **2020**, arXiv:2001.03210.
32. Supermarket Sales. Available online: https://www.kaggle.com/aungpyaeap/supermarket-sales (accessed on 27 September 2020).