An Advanced Inventory Data Mining System for Business Intelligence

Qifeng Zhou*, Bin Xia§, Wei Xue†, Chunqiu Zeng†, Ruyuan Han*, and Tao Li†‡
*Automation Department, Xiamen University, Xiamen, Fujian, 361005 China
†School of Computing and Information Sciences, Florida International University, FL, 33199 USA
‡School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, 210023 China
§School of Computer Science Technology and Engineering, Nanjing University of Science and Technology, Nanjing, 210094 China

***Abstract*—Inventory management plays a critical role to track inventory levels, orders, and sales of the retailing business. Effective inventory management is a capability necessary to lead in the global marketplace. In the current retailing market, a huge amount of data regarding stocked items in inventory is generated and collected every day. Due to the increasing volume of transaction data and their correlated relations, it is often a non-trivial task to efficiently manage stocked goods, yet it is imperative to explore the underlying dependencies of the inventory items and give insights into implementing intelligent management systems. However, existing inventory management systems rely on statistical analysis of the historical inventory data, and have a limited capability of intelligent management. For example, they usually do not have the ability to forecast item demand and detect anomalous patterns of item inventory transactions. There is little work reported in implementing intelligent inventory management solutions to reveal hidden relations with integrated data-driven analysis. In this paper, we present an intelligent system, called iMiner, to facilitate managing enormous inventory data. We utilize distributed computing resources to process the huge volume of inventory data and incorporate the latest advances in data mining technologies. iMiner provides comprehensive support for conducting many inventory management tasks, such as forecasting inventory, detecting anomalous items, and analyzing inventory aging.**

I. Introduction

Inventory management is the process of monitoring the product storage. A good inventory management is critical to the successful operation of most businesses and supply chains. In operation, inventory management has functionalities to avoid product overstocks and outages thus to reduce the carrying costs. In marketing, inventory management affects customer satisfaction. In finance, inventory investment is a company's largest asset. With more demanding customers and rising operation costs, it is much more important for retailers to apply inventory management technology to manage business transactions and business decisions [3] [17]. Therefore, the ability to transform inventory data into meaningful and actionable insight is a significant factor of competitive advantage for large retailers.

Developing intelligent inventory management system is quite challenging. First, inventory management involves many different facets of the retailers' business, such as, warehouse management, retail loss prevention, and inventory count. Second, in the current retailing market, a large amount of data about stocked goods needs to be processed instantly. For example, in one of our studies, the retailing vendor has 251,874 items in total, 1321,400 transactions per day on average, and more than 600 million records per year, which leads to more than 1TB data.

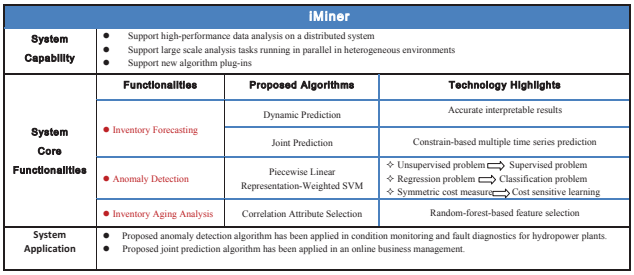

Fig. 1: iMiner Overview

A. Motivation for Developing iMiner

The evolution of market demand and business transaction influenced the inventory management systems to step beyond the basic and ERP inventory management, to adopt recent emerging cloud-based inventory management.

Most existing inventory management systems and methods, such as InFlow, Inventoria, Inform ERP, and Fishbowl Inventory[1], are demand-driven and cannot satisfy the needs in mining business big data. These traditional software tools only provide basic statistical functionalities, such as, tracking where products are stocked, which suppliers they come from, and how long they have been stored. They have limited capability and unable to support intelligent management, such as large-scale inventory data management, forecasting item demand automatically, and detecting anomalous patterns of item inventory transactions. It is a challenge for researchers to explore new methods to meet future requirements of inventory management.

To address the limitations of existing systems and assist large retailing business in efficiently performing inventory management, we design, implement, and deploy an intelligent inventory management system, named as **iMiner**. As shown in Figure 1, **iMiner** overcomes the aforementioned limitations

[1]http://www.softwareadvice.com/inventory-management/

978-1-5090-6318-5/17 $31.00 © 2017 IEEE
DOI 10.1109/BigDataService.2017.36

with a carefully-designed architecture and advanced data analysis techniques. More importantly, **iMiner** provides a set of key functionalities that facilitate the businesses convenience through efficient analysis of the large scale inventory data [11][13].

Main data analysis algorithms proposed in **iMiner** have also been extended to other application fields (e.g., the anomaly detection algorithm has also been applied in condition monitoring and fault diagnostics for hydropower plants, and the joint prediction algorithm has also been used in an online business management).

B. Research Challenges and Solutions

Based on our long-term collaboration with retailing companies and the demand for intelligent business in big data environment, we have identified four key issues that need to be addressed in the traditional inventory management system.

1) **Business Big Data management.** Most existing inventory management software tools suffer from the following limitations: 1) They are memory-based and cannot efficiently support large scale analysis. 2) They do not support new algorithm plug-ins.
2) **Inventory forecasting.** Since forecasting inventory can avoid product overstock and greatly reduce the maintenance cost, an accurate and interpretable inventory forecasting is highly needed. Usually, inventory management process has two types of time series data, i.e., the amount of stock in and stock out evolving over time. With the increasing transaction scales and complex transaction types, the following features of inventory data should be handled carefully for accurate inventory forecasting: 1) *Large amounts of records*; 2) *Large amounts of attributes*; 3) *Item correlations*: e.g. when a customer is buying a TV, he or she may also choose other related products, such as TV mounts or DVD players. The correlations further increase the difficulty for efficient inventory management.
3) **Inventory anomaly detection.** Inventory anomaly detection helps retailers find the unmarketable products and abnormal stock. In traditional inventory management system, the inventory-to-sales ratio (i.e., the ratio of the inventory available for sale versus the actual quantity sold) is a key statistic to measure whether or not an item is overstocked. When the data scale increases dramatically, and the type of anomalies gets more complicated, the inventory-to-sales ratio cannot accurately and timely reflect the real anomaly stock. This introduces the challenge of anomaly detection on big time-variant inventory and identification of the unmarketable products.
4) **Inventory aging analysis.** Inventory aging analysis can help companies understand the inventory status anytime, prevent items from overstocking, and reduce the overstocked items. Aging analysis can also be applied to liability accounts to obtain a clear picture of the company's obligations. Commonly used statistics-based methods provide some basic inventory aging analysis functionalities such as inventory aging computing, comparison, and classification. However, how to further discover in-depth knowledge such as the primary factors correlated with overstock items, remains challenging for understanding stock issues in advance and making quick responses.

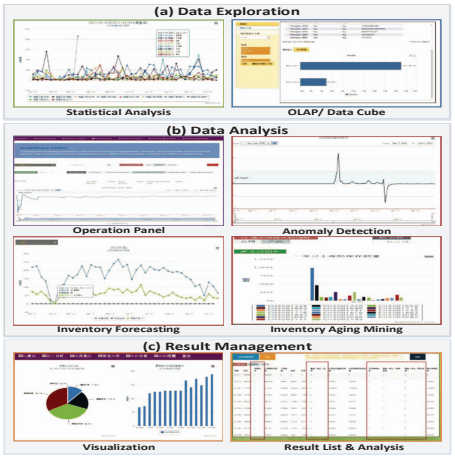

Fig. 2: Data Analysis Modules

To address the challenges mentioned above and fulfill the demand of intelligent inventory management for large retail company, a good inventory management system should be designed with the following principles: 1) The system should be able to handle large-scale data analysis; 2) The system should provide users with interactive functionalities; and 3) The system should have accurate, interpretable results.

Based on the above design principles, we develop **iMiner** on a powerful data analysis platform with advanced data analysis techniques. Specifically, to address challenge 1, we implement an integrated data analytic platform based on a distributed system to support high-performance data analysis. The platform manages all transactions in a distributed environment, which is capable of configuring and executing data processing and analysis in parallel. To address challenges 2, 3, and 4, we integrate a variety of data mining technologies to analyze inventory data. In particular, **iMiner** 1) adopts various regression models on time series data for inventory forecasting; 2) employs classification-based cost-sensitive learning algorithms to identify unusual items; and 3) utilizes statistical regression models to perform inventory aging analysis. **iMiner** provides various visualization tools and interpretable results for potential, complex transaction rules discovering. It helps users evaluate future inventory requests and make good market decisions.

C. Roadmap

The rest of the paper is organized as follows. Section 2 presents an overview of **iMiner**, starting from introducing the system architecture, followed by the system merits. Section 3 to Section 5 introduce three core functionalities of inventory management and the data-driven solutions adopted in proposed system. Specifically, in Section 3, we propose two models for

inventory forecasting: the dynamic model for single time series and the joint prediction model for multiple time series. In Section 4, we describe our proposed data-mining strategy to detect inventory anomaly from a large amount of inventory time series. Section 5 focuses on data-driven solutions for inventory aging analysis and attribute correlation mining on overstocked items. Section 6 presents the system deployment, including system performance evaluation and real results from actual usage. Finally, Section 7 concludes the paper.

II. System Overview

A. System Architecture

iMiner is built upon our previous developed large scale data analysis system, FIU-Miner, which is a Fast, Integrated, and User-friendly system to ease data analysis [19] [12].

The system is composed of four layers: *User Interface, Data Analysis Layer, Task and System Management Layer,* and *Physical Resource Layer*.

- *User Interface.* This layer contains various interactive interfaces for inventory operations. Specially, it provides Dashboard and Statistics Interface to allow users to have an overview of the current inventory status. In addition, several key indices of inventory, e.g., turnover rate and inventory-to-sales ratio, are presented in Inventory Index Interface, assisting users in promptly querying the status of a particular item.
- *Data Analysis Layer.* This layer is the heart of the system. Beside basic data processing and exploration functionalities, Data Analysis Layer consists of appropriate data mining solutions to the corresponding tasks of inventory management, including Inventory Forecasting, Anomaly Detection, and Inventory Aging Analysis. The main module functionalities will be discussed in Section 2.2 and the detailed algorithms will be discussed in Section 3, Section 4, and Section 5.
- *Task and System Management Layer.* Task and System Management Layer provides a fast, integrated, and user-friendly system to configure complex tasks, integrate various data mining algorithms, and execute tasks in a distributed environment. All the data analysis tasks in Data Analysis Layer can be configured as workflows and scheduled automatically.

B. Data Analysis Modules

1) Data Exploration: As shown in Figure 2(a), *Statistical Analysis* and *Data Cube* are capable of assisting data analysts in exploring inventory data efficiently and effectively. Statistical product analysis in different sizes and dimensions can quickly discover interesting time frame, product categories, and key indicators. Data Cube provides a convenient approach to explore high-dimensional data so that data analysts can have a better view of the characteristics of the dataset.

2) Data Analysis: In our system, the data mining approaches in *Algorithm Library* can be organized as a configurable procedure in Operation Panel. *Operation Panel* is a unified interface to build workflows for executing such tasks automatically. As shown in Figure 2(b), *Operation Panel* mainly contains three modules, *Inventory Forecasting* , *Anomaly detection*, and *Inventory Aging Mining*. These modules implement various mathematical models and advanced data mining algorithms to address the challenges of history data analysis in inventory management.

3) Result Management: As shown in Figure 2(c), *Result management* templates are designed to support automatic storage, update, and retrieval of discovered patterns. Results are recorded based on analysis tasks and can be organized according to different inventory time series, supply companies, brands, or data source. Dashboard, statistical graphs, and tables are produced to visualize the company's whole operations and analysis results. Also, for each result, customers and domain experts can refine and give feedbacks to the system.

III. Inventory forecasting

The primary goal of inventory forecasting is to minimize the inventory loading. A common practice of inventory forecasting is to predict the demand for a particular item in the future and reserve the appropriate amount of items, based on the forecasting results. However, inventory data is a type of time series with large volume, long time span, and fewer regularities. These features bring up two challenges to inventory forecasting: 1) implementing an accurate interpretable inventory prediction; 2) modeling the relationships among multiple time series data sets and predicting their future values simultaneously.

Although, in recent years, there has been an explosion of interest in mining time series [2] [14], traditional approaches such as auto-regression(AR), linear dynamical systems (LDS), Kalman filter(KF) cannot solve above challenge directly [1][8]. Hence, more accurate and optimal forecasting methods are expected. In **iMiner** we design and deploy two new inventory forecasting models: *dynamic prediction model* and *joint prediction model* to solve the prediction problems.

A. Dynamic Prediction Model

To implement an accurate and interpretable inventory forecasting, we propose a two-step dynamic prediction model. First, it adopts machine learning techniques combined with time series analysis methods to obtain a forecasting basis. Second, it takes into account multiple factors of inventory such as seasonality, trend, and special events for dynamic inventory forecasting.

The overall framework of dynamic forecasting model is shown in Figure 3.

1) Determining the Forecasting Basis: In this step, we employ machine learning algorithms to capture the hidden patterns in stock in/out time series. Each algorithm is used to build an inventory forecasting regression model based on the past inventory transaction data and update on a daily basis. These algorithms include : Linear Regression [16], Neural Network (NN) [7], Gradient Boosting Regression Tree [6],

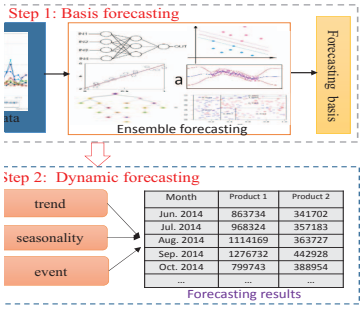

Fig. 3: The Framework of Dynamic Forecasting Model

Support Vector Machine (SVM) [18], and Gaussian Process (GP) [15].

Then, to combine the forecasting results of these models effectively, we adopt a weighted linear combination strategy [9]. Suppose the forecasting value of the learning model $p \in P$ at time t is $\hat{v}_p^{(t)}$, and its weight at time t is $\hat{w}_p^{(t)}$, then the weighted ensemble forecast value for stock out at time t is

$$\hat{v}^{(t)} = \sum_p \hat{w}_p^{(t)} \hat{v}_p^{(t)}, s.t. \sum_p \hat{w}_p = 1. \quad (1)$$

Initially (t=0), all the learning model have the same contribution to the forecasting result, i.e. $\hat{w}_p^{(0)} = \frac{1}{|P|}$. The relative error between each forecasted value $\hat{v}_p^{(t)}$ and the real value $v_p^{(t)}$ are used to update the weights

$$w_i^{(t+1)} = \frac{c_i^{(t)}}{\sum_p c_p^{(t)}} w_i^{(t)}, \quad (2)$$

where $c_i^{(t)}$, $c_p^{(t)}$ is the predictive error of predictor i and p computed by regression cost measure function, such as mean average error (MAE), least square error (LSE), and mean absolute percentage error (MAPE).

2) Constructing Dynamic Forecasting Model: The ensemble models do not consider the characteristics of inventory forecasting. To enhance the accuracy and the interpretability of forecasting, we take the ensemble result as the forecasting basis, and then, take into account various practical factors of inventory data for providing final predictions.

- *Long-term trend factor*: It refers to a portion of item demand has a trend of increase (growth) or decrease (decline) for a long period (e.g. monthly amount of stock out over past 3 to 5 years). For instance, a trend may show a period of growth followed by a leveling off.
- *Seasonality factor*: It refers to the portion of item demand fluctuation accounted for by a reoccurring pattern but different intensity and frequency.
- *Event factor*: It refers to some special events, such as holidays and sales promotion, and these events may have a great impact on the demand for items.

Considering the past sales are positively correlated with current sales, after obtaining the forecasting basis, we combine it with current sales to obtain the final forecasting results.

This two-step dynamic prediction model can capture the characteristics of the inventory transaction data including the trend in time series (reflected in the forecasting basis) and the impact of monthly periodicity, trend and events (reflected in the dynamic forecasting process), thus, give an interpretable prediction result.

B. Joint Prediction

1) Multiple Time Series Prediction: Existing inventory management systems [2] often forecast the two time series stock in and stock out separately. Both of them are treated as independent ignoring their relationship.

In practice, the amounts of stock in and stock out in an inventory are often dependent on each other. The amount of stock out (S_{out} for short) is usually subject to the amount of stock in (S_{in} for short) at the same or near time periods to prevent the situation that an item is out of stock. Also, the scheduled S_{in} primarily depend on the past S_{out} to avoid the situation that a unit is in excess of demand. So two time series of S_{in} and S_{out} bear some interdependencies according to the characteristics of inventory management. The existing single time series predictive methods lack the capability of capturing the dynamic relationships between multiple time series or predicting their future values simultaneously. Little research attention has been paid to predicting the movement of a collection of related time series.

In **iMiner**, we model the interdependencies of time series data and integrate them into the process of time series prediction. The model can capture the dynamic relationships between multiple time series data set and predict their future values simultaneously. Specifically, in the domain of inventory management, the aggregated amount of S_{in} is often larger than the aggregated amount of S_{out} in a given period to avoid "out of stock". In addition, S_{in} and S_{out} should be close to each other to prevent a unit from "excess demand". Based on such an intuition, we transform the requirement of inventory management into model constraints and perform time series prediction under the constraints.

2) The Joint Predictive Model: In multiple time series prediction, we have L different time series, and for all time series, we have N available examples $\{x_i^{(l)} : i \in \{1, 2, \ldots, N\}\}$. The goal is to learn a function $f_l N \rightarrow X$ for each l-th time series based on the available data, and in total, we have L such functions. Note that the input space of all L time series might share common representation, or their input space could be totally different.

Different from independent L single time series predictions, in joint prediction, the output space of different time series might correlate with each other, i.e., they may have to satisfy some specific constraints naturally embedded in the applications. In the case of inventory management, the storage capacity should always be greater than the volume of sales to avoid the situation that supply falls short of demand. Therefore, we add constraints into the model to capture the relationships between different time series.

$$\frac{1}{NL} \sum_{l=1}^{L} \sum_{i=1}^{N} L(x_i^{(l)}, f_l(i)) + \gamma \sum_{l=1}^{L} \|f_l\|_F \quad (3)$$

where $\|f_l\|_F$ could be any kind of regularization term for the function f_l.

In our case, the constraint is

$$\begin{aligned} &\|f_{\text{in}}(i) - f_{\text{out}}(i)\| < \delta, \quad \forall i \\ &\sum_{i=1}^{n} f_{\text{in}}(i) + C > \sum_{i=1}^{n} f_{\text{out}}(i) + M, \quad \forall n \end{aligned} \tag{4}$$

where $\delta > 0$, C is the initial amount of storage, M is the safety margin between the amount of stock in and stock out.

Given a specific loss function, we can carry out a joint prediction by solving the above optimization problem. In **iMiner** we adopt SVM regression as learning machine which uses a type of loss function called ϵ-insensitive loss function [18]:

$$L(x_i, f(i)) = \begin{cases} 0 & \text{if} \quad |x_i - f(i)| \leq \epsilon \\ |x_i - f(i)| > \epsilon & \text{otherwise} \end{cases} \tag{5}$$

Thus combining Equation 3 and Equation 5, we can get an SVM-based joint prediction model to perform multiple time series prediction.

Dynamic forecasting model and joint prediction model are both integrated into an interactive interface to provide multiple prediction of demand forecasting.

IV. Inventory Anomaly Detection

Monitoring inventory index for anomaly detection is a critical task of inventory management. This problem becomes difficult in the Big Data era since the data scales substantially, and the types of anomalies gets diversified.

In our system, in addition to providing traditional statistical methods, we also design and develop a classification-based anomaly detection method to identify abnormal items. Here, we assume abnormality is big fluctuation of stock or sale data.

We implement a precise and practical anomaly detection method. The fundamental techniques and novelties are highlighted as follows:

- *Labeling anomalous points automatically.* Original inventory time series do not contain the label information of anomaly. We search for anomalous points from the data automatically and annotate them with different labels, so that the problem is modeled as a supervised problem.
- *Adopting classification-based method instead of regression-based method to reduce the computing effort.* After labeling the data with class labels, anomaly detection can be formulated as a classification problem. Compared with common regression-based methods, the computing effort is reduced significantly.
- *Changing the cost measure to be cost-sensitive.* Existing anomaly detecting methods consider the equal costs for different types of detecting errors. However, in practice, anomaly detection is a cost-sensitive learning problem. The costs for different class instances are different. Our model minimizes the detecting costs instead of minimizing detecting errors.

Based on above ideas, **iMiner** implements a classification based anomaly detecting model named PLR-WSVM, which combines Piecewise Linear Representation with Weighted Support Vector Machine. Piecewise Linear Representation (PLR) method is used to capture the fluctuated points to form the training dataset, and the weights of the fluctuated points are also determined according to the changing trend. Finally, Weighted Support Vector Machine (WSVM) is adopted to build the anomaly detection model.

A. Problem Transformation

1) Generating Anomaly Training Points by PLR: PLR is used to automatically generate the different classes for anomalous points. The time series is then represented by

$$T_L = \{L_1(i_1, i_2), L_2(i_2, i_3), ..., L_{m-1}(i_{m-1}, i_m)\}, \tag{6}$$

where the function $L_{m-1}(i_{m-1}, i_m)$ represents the linear fitting function at the interval $[i_{m-1}, i_m]$. Because PLR represents the time series as a sequence of linear functions, the value of each point in every interval can be obtained by linear interpolation. Then, the fitting sequence is expressed as $T'(i) = (x'_1, x'_2, ...x'_n)$.

There are three types of time series segmentation algorithms: Sliding Windows, Top-down, and Bottom-up [10]. In our model, the Top-down algorithm is selected to segment the inventory time series and the linear interpolation is adopted to generate the approximation line. The segmentation objective is to produce the best representation such that the maximum error for any segment does not exceed the given threshold.

We divide all the samples produced by PLR into three classes: peak points(inventory suddenly increased points), trough points (inventory suddenly decreased points) and stable points (inventory changes a little), labeled classes 1, 2, and 3, respectively.

Furthermore, anomaly detection is cost sensitive, the anomalous points obtained from PLR are assigned different weights according to the change rate between the current anomalous point and the next one, therefore, the weight reflects the relative importance of each anomalous point.

B. Constructing Anomaly detection Model by WSVM

A three-class weighted classification problem can be constructed for samples. Since SVM has the excellent generalization performance as well as all solutions of SVM model are globally optimal, we incorporate PLR and Weighted Support Vector Machine (WSVM) [21][5] to model this three-class classification problem.

The main idea of SVM is to generate a classification hyperplane that separates two classes of data with the maximum margin. The standard SVM model is as follows:

$$\begin{aligned} &\min_{w,b,\xi_i} \ \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{l} \xi_i, \\ &s.t. \quad y_i(\langle w, \phi(x_i) \rangle x_i + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0, \quad i = 1, 2, ...l, \end{aligned} \tag{7}$$

where $x_i \in R^n$ and $y_i \in \{-1, 1\}$ are respectively the training sample and the corresponding class label, ϕ is a nonlinear map from the original space to a high dimensional feature space, w is the normal vector of hyper-plane in the feature space, b is a bias value, ξ_i is the slack variable, $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors, and C is a penalty coefficient to balance the training accuracy and generalization ability. The dual form of model (7) is:

$$\begin{aligned} &\min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{l} \alpha_i \\ &s.t. \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \\ &0 \leq \alpha_i \leq C, \quad i = 1, 2, ...l, \end{aligned} \tag{8}$$

The model (8) is an linear SVM method, and it can be easily generalized to non-linear decision rules by replacing the inner products $\langle x_i, x_j \rangle$ with a kernel function $k(x_i, x_j)$. When each training sample has a weight, the standard SVM can be extended to weighted SVM (WSVM) [5], the model (8) is transformed to

$$\begin{aligned} &\min_{\alpha} \frac{1}{2} \sum_{i=1}^{l} \sum_{i=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{i=1}^{l} \alpha_i \\ &s.t. \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \\ &0 \leq \alpha_i \leq C\mu_i, \quad i = 1, 2, ...l, \end{aligned} \tag{9}$$

where $\mu_i (i = 1, ..., l)$ represents the weight of instance x_i. The decision function for WSVM is the same as the standard SVM.

iMiner implements the above techniques and builds a classification model based on a training set and then applies the model for new test instances. New test instances arriving every day can be cross-checked with the learned model and the model can also be updated with the new data.

V. Inventory Aging Analysis

The main purpose of monitoring and analyzing inventory aging is to prevent items from overstocking and reduce the overstocked items. In our system, an overstocked item at the time t is an item with the amount more than x% (e.g., 30%) over y (e.g., six months) old, where x and y can be set by users. We provide in the system both basic and advanced tools to analyze the inventory aging.

A. Basic Inventory Aging Analysis

The basic tools allow users to visualize inventory aging distributions of a given item. Users can also analyze inventory aging changes among different items. Our system mainly contains the following modules:

- Category and index: grouping inventory data according to the type of items (e.g. Color TV, PDP, LCD TV), setting inventory aging segmentation (e.g. 0-3 months, 4-6 months, etc.)
- Inventory aging computing: based on Fist In First Out rule, computing and visualizing the inventory aging distributions.
- Current and historical inventory aging compare: finding out and ranking the items with inventory aging data over a particular period.

B. Advanced Inventory Aging Analysis

The advanced tools help users find potential overstocking and adjust inventory automatically. The aging mining is to identify attributes of items correlated with overstocking. This allows users to monitor the related attributes and pay attention to the items whose attributes are not consistent with predefined thresholds. To find out the item attributes that result in overstocking, we model this task as a feature selection problem. The system integrates two types of feature selection methods: filter model and wrapper model, to rank the attributes. The filter model is first used to preselect feature candidates from the original set of features. Then a wrapper model is used to identify attributes that are greatly correlated with overstocking.

We assume that for each item in stock and each timestamp, we have a data sample, (x_i, y_i), where x_i represents the attributes of the item and y_i is the label. When an item is overstocked, $y_i = 1$; otherwise $y_i = 0$. The label information can divide the entire item collection X into overstocked items X_1 and non-overstocked items X_0.

1) Feature Selection based on Filter Model: Inventory data has an enormous amount of records and attributes. The time required by a classification algorithm often grows dramatically with the number of features. In a filter model, a feature selector serves as a filter to differentiate the irrelevant and redundant features. It is usually less computationally intensive than wrapper models. Hence, our system first adopts a filter model to find a candidate feature subset from the original set of attributes to speed up the processing of inventory aging mining.

The system integrates several filter feature selections methods including Information Gain, mRMR, and ReliefF to rank the attributes, and allows users to configure them.

2) Feature Selection based on Wrapper Model: After candidate feature subset is generated by the filter models, we use a wrapper model, i.e., random forest proximity matrix [20] to obtain the final results.

In general, inventory items have lots of noise, and their attributes contain various types, such as numeral, text, and time. On the other hand, Random Forest (RF) as an ensemble learning algorithm well suited for inventory data [4].

The original random forest returns the measures of attribute importance. This measure is based on Out-of-bag (OOB) error and can be used as feature selector. We improved this method by using a more sensitive measure: samples proximity matrix. We take the proximity as a kernel measure, and use the differences between samples proximity, instead of OOB error, as the criterion to select the informative attributes. Compared with the original variable importance analysis of random forest, our method is more sensitive to correlated features,

and yields smaller sets of informative features and preserve predictive accuracy.

Proximity matrix. Random forest can produce a proximity matrix, which measures the input based on the frequency that pairs of data points are in the same leaf nodes. Specifically, if two samples are in the same leaf node, their proximity is increased by one. This step is iterated in all trees for all samples. Then we normalize the counts by dividing the number of all trees to get the proximity matrix. Proximity matrix is symmetric, where the diagonal elements are 1 and the elements off the diagonal are from 0 to 1 . It is useful for similarity measure between cases and can be taken as a special kernel measure for feature selection.

Feature selection approach. Our previous studies show that in comparison to OOB error or margin measure, proximity measure is more sensitive to the change of features in the context of random-permuted-based variable importance measure. Our feature selection algorithm, named as Feature Selection Random Forest Proximity (FS-RFP), utilizes the similar mechanism with random forest variable importance measure to calculate the importance of all features, leading to a feature rank. By permuting the values of one feature for all samples, the variation of proximity matrix can be used to calculate the importance scores for features. The entry in the rank signifies the index of the feature. The top-ranked features have high importance.

Our system adopts FS-RFP model to mine the most correlated attributes with overstocking. For each attribute, users can further compare its histogram on X_1 (overstocked items) and X_0 (non-overstocked items) to obtain a more intuitive sense of the relationship between attributes and overstocking. Using the related attributes, users can impose queries to retrieve the items that are likely to overstock and put them on the monitoring list.

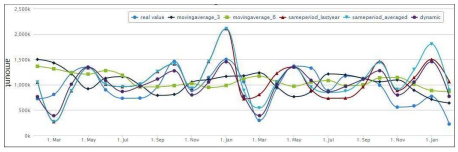

Fig. 4: The Forecasting Results for Product A

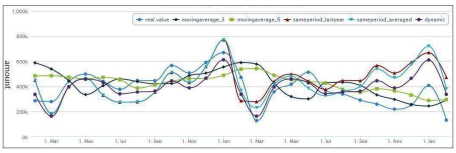

Fig. 5: The Forecasting Results for Product B

VI. System Evaluation

We evaluate our proposed system on two aspects: the system performance and the practical application. The evaluation demonstrates that our system successfully addresses the aforementioned challenges (cf.§1.2). Our system offers an effective and efficient solution for large-scale inventory data analysis, because of various data-driven techniques and its customization ability.

A. System Performance

We first evaluate the performance of core functionalities of our system on a real world inventory big dataset. Compared with existing inventory analysis techniques, **iMiner** can perform large-scale data management and data analysis. **iMiner** shows a powerful decision support performance by distilling real inventory data.

(1) Prediction Accuracy. To illustrate the performance of proposed inventory prediction models, we conduct two experiments for dynamic prediction model and joint prediction model respectively. After the deployment of **iMiner**, we weekly collected the real-world data of stock out for two products: Product A and Product B, from January 2011 to December 2013.

EXP A. Dynamic prediction model. We predict the amount of stock out from March 2014 to December 2014. As shown in Figure 4 and Figure 5, compared with common statistical analysis approaches, i.e., Moving average and Contemporary comparison, our proposed dynamic forecasting model obtains the highest fitting degree with the true values of both sales trends and periodicity. Moreover, our model can give a reasonable interpretation to the analysis results.

EXP B. Joint prediction model. We also compare our model with commonly used single time series approaches: Moving average and SVM regression, on the real business dataset. We select three types of metric: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) to evaluate the performance of prediction models. Our proposed joint prediction model is able to capture the dynamic relationships between multiple time series and predict their future values simultaneously. As shown in Table I, the precision of joint prediction is higher than the other learning methods, meanwhile, the joint prediction results can conform to the practical constraints, such as predicted S_{in} great than predicted S_{out}.

TABLE I: Performance Comparison between Joint Prediction and Single Time Series Prediction

	Model	S_{in}	S_{out}	Average
MAE	Joint[a]	**12130**	8247	**10188.5**
	SVM[b]	14051	7391	10721
	Moving Average[c]	15429	7613	11521
RMSE	Jiont	**16017**	9458	**12737.5**
	SVM	17756	9311	13533.5
	Moving Average	17756	**8450**	13103
MAPE	Jiont	**0.18**	**0.11**	**0.16**
	SVM	0.21	0.13	0.17
	Moving Average	0.25	0.14	0.20

[a] Joint prediction model with constraints.
[b] Single prediction model based on SVM regression without constraints.
[c] Single prediction model based on Moving average approach without constraints.

(2) Real findings. iMiner has been playing an important role in revealing deeper and hidden relations behind inventory Big Data in practice. For example, **iMiner** finds some real overstock items from inventory time series data. On February 3, 2014, our system first detected that a peak point appeared on two kinds of product: Product A and Product B. Traditional inventory-to-sales ratio method indicates these two product are overstocked. However, our system further analyzed the overstocking issue, and showed that these two peaks are planned inventory storage, instead of an overstocking anomaly.

As to inventory aging analysis, according to the data analysis of January 2011 to February 2014, our system found out some valuable information. Such as the attributes of "time since last in-put-warehouse" and "time since last out-put-warehouse" have strong correlation, and these two attributes are suitable for predicting material overstock. We also obtain some potential overstock knowledge, for example, the safety inventory aging threshold of one product is 4 to 6 months. If the inventory aging exceeds a threshold, the likelihood of overstock is high.

B. Deployment Practice

From January 2014 to now, **iMiner** has been applied in a big company in China to reach the goal of simplified, efficient inventory management and sound economy.

In summary, our system brings great benefits in intelligent inventory management:

1) Efficient support of large-scale inventory data analysis. The platform manages all the transaction data in a distributed environment, which is capable of configuring and executing data preprocessing and automatic data analysis.

2) Effective management of complex analysis tasks. The system integrates a variety of data mining technologies to tasks of inventory forecasting, anomaly detection, and inventory aging analysis accurately.

3) Intelligent decision support of potential knowledge mining. The system also plays an important role as an intelligent decision support system for inventory management. It can discover some valuable knowledge from inventory data. For example, the system can determine the relation between the inventory change and the product external data. This knowledge can facilitate the strategic and sales plan improvement, and can contribute in achieving long-term organizational goal.

In addition, some major data analysis algorithms proposed in our system have also been applied to other fields. For example, the proposed anomaly detection algorithm has also been used for condition monitoring and fault diagnostics in hydropower plants in Fujian province of China. The proposed joint prediction algorithm has also been adopted in an online Tea sell company as a core part of inventory forecasting.

VII. Conclusions

iMiner has been deployed at a big Chinese company as an intelligent inventory management system since 2014. It improves the inventory management from demand-driven to data-driven, and addresses the challenge of big data and complicated transaction process.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China under Grant No. 61503313 and No. 91646116, by Scientific and Technological Support Project (Society) of Jiangsu Province (No. BE2016776), by Ministry of Education/China Mobile joint research grant under Project No. 5-10. The authors would also like to thank the former members of Florida International University KDRG research team (Dr. Jingxuan Li, Dr. Lei Li, Dr. Chao Shen, Dr. Liang Tang, and Dr. Li Zheng) for their support and contributions to this work.

References

[1] Y. Baraud, F. Comte, and G. Viennet. Model selection for (auto-) regression with dependent data. *ESAIM: Probability and Statistics*, 5:33–49, 2001.

[2] G. E. Box, G. M. Jenkins, and G. C. Reinsel. *Time series analysis: forecasting and control*, volume 734. John Wiley & Sons, 2011.

[3] C. C. Bozarth and R. B. Handfield. *Introduction to operations and supply chain management*. Prentice Hall, 2015.

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[6] J. Elith, J. R. Leathwick, and T. Hastie. A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4):802–813, 2008.

[7] M. T. Hagan, H. B. Demuth, M. H. Beale, et al. *Neural network design*. Pws Pub. Boston, 1996.

[8] A. Jain, E. Y. Chang, and Y.-F. Wang. Adaptive stream resource management using kalman filters. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 11–22. ACM, 2004.

[9] Y. Jiang, C.-S. Perng, T. Li, and R. N. Chang. Cloud analytics for capacity planning and instant vm provisioning. *Network and Service Management, IEEE Transactions on*, 10(3):312–325, 2013.

[10] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 289–296. IEEE, 2001.

[11] L. Li, C. Shen, L. Wang, L. Zheng, Y. Jiang, L. Tang, H. Li, L. Zhang, C. Zeng, T. Li, et al. iminer: Mining inventory data for intelligent management. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 2057–2059. ACM, 2014.

[12] T. Li, C. Zeng, W. Zhou, W. Xue, Y. Huang, Z. Liu, Q. Zhou, B. Xia, Q. Wang, W. Wang, et al. Fiu-miner (a fast, integrated, and user-friendly system for data mining) and its applications. *Knowledge and Information Systems*, pages 1–33, 2016.

[13] T. Li, C. Zeng, W. Zhou, Q. Zhou, and L. Zheng. Data mining in the era of big data: From the application perspective. *Big Data Research*, 1(4):2015041, 2015.

[14] Y. Matsubara, Y. Sakurai, W. G. van Panhuis, and C. Faloutsos. Funnel: automatic mining of spatially coevolving epidemics. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 105–114. ACM, 2014.

[15] C. E. Rasmussen. Gaussian processes for machine learning. 2006.

[16] G. A. Seber and A. J. Lee. *Linear regression analysis*, volume 936. John Wiley & Sons, 2012.

[17] B. Tan and S. Karabati. Retail inventory management with stock-out based dynamic demand substitution. *International Journal of Production Economics*, 145(1):78–87, 2013.

[18] V. Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[19] C. Zeng, Y. Jiang, L. Zheng, J. Li, L. Li, H. Li, C. Shen, W. Zhou, T. Li, B. Duan, et al. Fiu-miner: a fast, integrated, and user-friendly system for data mining in distributed environment. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1506–1509. ACM, 2013.

[20] Q. Zhou, W. Hong, F. Yang, and L. Luo. Feature selection of random forest-based proximity matrix difference. *Journal of Huazhong Univ. of Sci.& Tech.: Natural Science Edition*, (4):58–61, 2010.

[21] Q. Zhou, B. Xia, Y. Jiang, Q. Li, and T. Li. A classification-based demand trend prediction model in cloud computing. In *Web Information Systems Engineering–WISE 2015*, pages 442–457. Springer, 2015.