Knowl Inf Syst
DOI 10.1007/s10115-016-1014-0

REGULAR PAPER

FIU-Miner (a fast, integrated, and user-friendly system for data mining) and its applications

Tao Li[1,2] · Chunqiu Zeng[1] · Wubai Zhou[1] · Wei Xue[1] · Yue Huang[2] · Zheng Liu[2] · Qifeng Zhou[3] · Bin Xia[1] · Qing Wang[1] · Wentao Wang[1] · Xiaolong Zhu[1]

Received: 8 May 2016 / Revised: 23 October 2016 / Accepted: 3 December 2016

Abstract The advent of Big Data era drives data analysts from different domains to use data mining techniques for data analysis. However, performing data analysis in a specific domain is not trivial; it often requires complex task configuration, onerous integration of algorithms, and efficient execution in distributed environments. Few efforts have been paid on developing effective tools to facilitate data analysts in conducting complex data analysis tasks. In this paper, we design and implement FIU-Miner, a *F*ast, *I*ntegrated, and *U*ser-friendly system to ease data analysis. FIU-Miner allows users to *rapidly configure a complex data analysis task* without writing a single line of code. It also helps users *conveniently import and integrate different analysis programs*. Further, it significantly *balances resource utilization and task execution in heterogeneous environments*. Case studies of real-world applications demonstrate the efficacy and effectiveness of our proposed system.

1 Introduction

As the data scale and complexity increase explosively, the tasks of discovering knowledge from data have far beyond the processing ability of human being. In many application domains such as health care, manufactures, finance, cloud service, disaster management, and social media, a typical data mining task often requires complex task configuration, integration of different types of data mining algorithms, and efficient execution on distributed computing environments [12,15,32,38,41]. Therefore, it is imperative to build tools for data analysts in such domains to efficiently perform data analysis tasks.

✉ Tao Li
taoli@cs.fiu.edu

1 School of Computing and Information Sciences, Florida International University, Miami, FL 33199, US

2 School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

3 Automation Department, Xiamen University, Xiamen 361005, China

Published online: 23 December 2016

Existing data mining products, such as Weka [11], SPSS, and SQL Server Data Tools, provide user-friendly interfaces to facilitate users to conduct the analysis. However, these products are designed for small-scale data analysis and do not allow users to plug in new algorithms easily. The data mining algorithm libraries, such as Mahout [23], MLC++ [21], and MILK [20], include a large number of data mining algorithms. However, it requires advanced programming skills to use these libraries for task configuration and algorithm integration in a complex data mining task. Integrated data mining frameworks, such as Radoop [27] and BC-PDM [36], provide user-friendly interfaces to quickly configure data mining tasks. However, these frameworks are Hadoop-based and have limited support for non-Hadoop implementations. Moreover, they do not explicitly address the resource allocation under multi-user and multitask scenarios.

To address the limitations of existing products, we develop FIU-Miner to facilitate data analysts to efficiently perform data mining tasks. FIU-Miner provides a set of novel functionalities that help data analysts conveniently and efficiently conduct complex data mining tasks. Specifically, the system has the following significant merits:

- *User-friendly rapid data mining task configuration* Following the Software-as-a-Service paradigm, FIU-Miner hides the low-level details irrelevant to the data analysis tasks. Through the interface, users can easily configure a complex data mining task by assembling existing algorithms into a workflow without writing a single line of code.
- *Flexible cross-language program integration* FIU-Miner is able to make full use of the existing state-of-the-art data mining tools by allowing users to import them into its system algorithm library. There is no restriction on the choice of implementation languages for the imported programs, since FIU-Miner is capable of correctly distributing the tasks to appropriate computing nodes with suitable runtime environments.
- *Effective resource management in heterogeneous environments* FIU-Miner supports the data mining tasks running in heterogeneous environments, including graphics workstations, stand-alone computers, and computing clusters. To optimize the resource utilization of the available computing resources, FIU-Miner schedules the tasks by considering various factors such as algorithm implementation, server load balance, and data location.

The remaining of the paper is organized as follows: Section 2 introduces the system architecture. In Sect. 3, we demonstrate the effectiveness of the system with real-world applications. In Sect. 4, we evaluate the performance of the system. Finally, we conclude this paper in Sect. 5.

2 System overview

FIU-Miner has been designed and developed by a research team consisting of 12 members for one year. A demo can be found in http://datamining-node08.cs.fiu.edu/FIU-Miner. Figure 1 presents an overview of the system architecture of FIU-Miner. The system is divided into four layers: *User Interface*, *Task and System Management*, *Abstracted Resource*, and *Heterogeneous Physical Resource*.

2.1 User interface layer

To maximize the system compatibility, the user interface is presented as a pure HTML5 web-based application. There are three major modules of user interface. Their functionalities and features are described as follows:

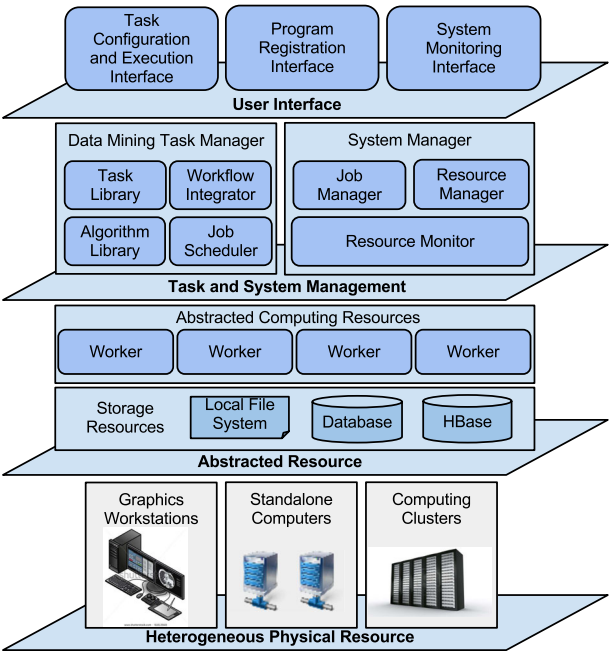


Fig. 1 System architecture

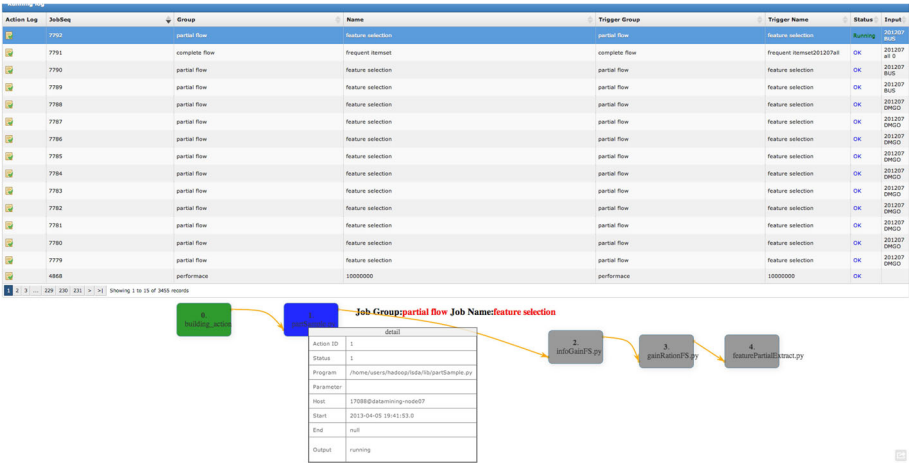


Fig. 2 Data mining task configuration. *Every row of table* denotes a workflow configuration, and its corresponding directed graph is displayed when clicking the *row*. The detail of algorithm is shown when placing the mouse over its corresponding node

– *Task configuration and execution (see Fig. 2)* This module supports workflow-oriented task configuration. The workflow of a data mining task is represented and visualized as a directed graph, where nodes denote specific algorithm and edges denote the data dependency among algorithms. A workflow can be quickly configured through GUI instead of programming. Moreover, users can set the execution plan of a data mining task that will be automatically executed whenever the time is up. This interface enables

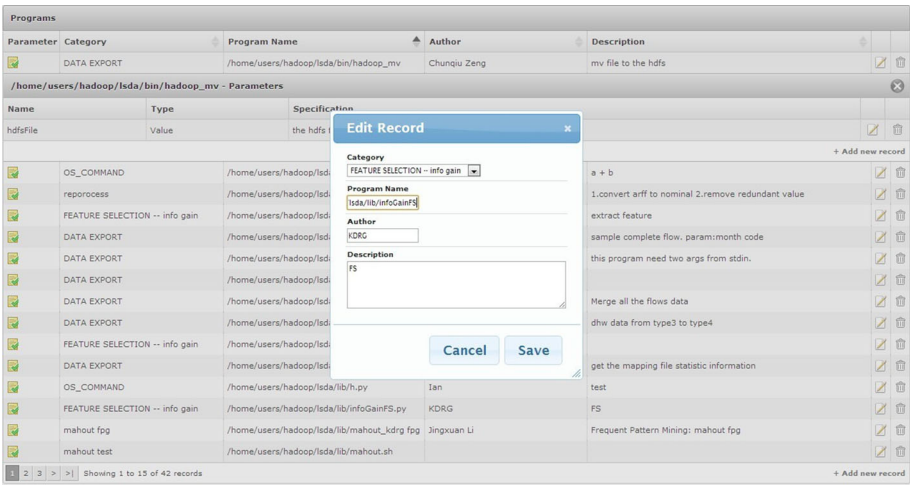


Fig. 3 Interface for data mining program registration. The pop-up dialog provides the interface to configure the executable program and its description. All registered programs are listed in *table*

users to conduct complex data mining tasks without programming. Users do not need to have any data mining background if the data mining task is predefined. The only thing they need to do is to set the input and output of the data. Moreover, the users can set the execution plan of the data mining tasks, so whenever the time is up, the scheduled task would automatically execute. For advanced users, they can design ad hoc data mining tasks by using existing implemented algorithms as the build blocks.

– *Program registration (see Fig. 3)* This module allows users to easily import external data mining programs to enrich the *Algorithm Library*. To import a program, a user needs to upload the executable files and provide the detailed profiles of the program, including functionality description, required runtime environment, dependencies, and parameter specification. The imported program can be written in any form as long as its runtime environment is supported by the backend servers. FIU-Miner currently supports Java (including Hadoop environment), Shell, Python, C/C++, etc. Therefore, almost all the mainstream data mining algorithm implementations, such as Weka-based programs, Mahout-based programs, and MILK-based programs, can be imported into FIU-Miner. Users can also implement and import their own algorithms into the system.
– *System monitoring (see Fig. 4)* This module monitors the resource utilization of FIU-Miner in real time and also tracks the running status of the submitted tasks. Note that this module only displays the abstracted resources (e.g., the available workers, the tree structures of the file system) for users. The underlying physical resources are transparent to users.

2.2 Task and system management layer

This layer contains two major modules: *task management* and *system management*.

2.2.1 Task management

As previously mentioned, users are allowed to configure ad hoc data mining tasks as workflows to fulfill their analysis requirements. A user can pick the available algorithms registered

Fig. 4 Resource monitoring is displayed. The number of tasks executed on each computing node in the distributed environment is shown instantly

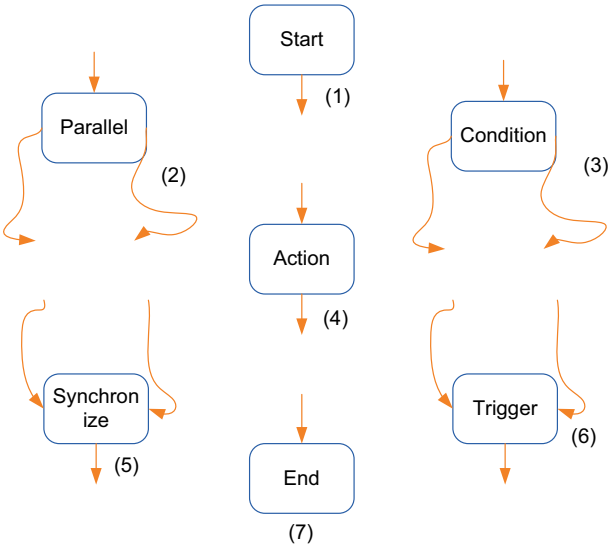


Fig. 5 Node types to compose a workflow

in the *Algorithm Library* as a building block when configuring a data mining task. The *Workflow Integrator* conducts task validation and reports invalid integration. Once the configuration is done, the new data mining task will be automatically added to the *Task Library* and is ready for scheduling.

A workflow is represented as a directed and connected graph consisting of nodes (denoting the sub-tasks) and edges (describing the dependencies among the sub-tasks). Data transmission between dependent sub-tasks is supported in our system.

In order to support different execution flows such as sequential execution, loop execution, and branching execution, seven types of nodes are defined as shown in Fig. 5.

Fig. 6 Workflow examples

1. *Start node* This type of node indicates the start of the workflow. There is only one such node in a valid workflow. This start node must link to one other node.
2. *Parallel node* This type of node has one input link and more than one output links. After the work is completed in the parent node, the parallel node triggers the sub-tasks in its children nodes. All the sub-tasks of its children are executed in parallel.
3. *Condition node* One input link and more than one output links are associated with this type of node. When the control flow reaches a condition node, it will check the input data and then move along one of its output links.
4. *Action node* One input link and one output link are associated with this type of node. It often accommodates the sub-tasks for data analysis. The data from the input link are fed into the sub-task, and the result data of this sub-task are forwarded along its output link.
5. *Synchronize node* This type of node has more than one input links and one output link. This node does not direct the control flow to its output link until all the sub-tasks in its parent nodes are completed.
6. *Trigger node* More than one input links and one output link are associated with this type of node. The node starts the sub-tasks in its output link once one of sub-tasks in its parent nodes is finished.
7. *End node* Any valid workflow should have one and only one end node. It indicates the end of the workflow.

A typical example workflow in data analysis is displayed in Fig. 6. This workflow describes an ensemble method for feature selection, where multiple feature selection algorithms are integrated. These multiple feature selection algorithms can be executed in parallel mode.

The *Job Scheduler* is responsible for assigning the computing jobs to the computing *Worker(s)* where the runtime environments are supported with the purpose of minimizing the running time. The scheduling in FIU-Miner is not trivial as it supports programs of different languages in a heterogeneous environment. It is possible that the programs in a configured job have different runtime requirements; hence, simply assigning the job to an arbitrary worker may render the job inexecutable. On the other hand, the I/O cost would increase when decomposing the job into different steps and running each step to a different worker. The scheduling would become even more difficult when considering the multi-user and multitask situation. In FIU-Miner, to address the aforementioned challenge, we implement the scheduler by considering the following factors: (1) runtime environment requirements of each step in a given job; (2) supported runtime environment of each computing worker; (3) current running status of each computing worker; and (4) estimated data size of the input.

2.2.2 System management

The *Job Manager* keeps track of the running status of an executed job. A user will be immediately notified the job status in real time.

Besides job monitoring, FIU-Miner also keeps track of the status of the workers and the underling computing resources. The *Resource Monitor* monitors the *Worker*s and provides the running status of the *Workers* to the *Job Scheduler* to help make scheduling decisions. The *Resource Manager* manages all the available *Worker*s. A unique feature of FIU-Miner is that it does not need the manual registration of available physical resources. Once a worker is deployed on a physical server, the worker will automatically register the server to FIU-Miner by sending server profile to the *Resource Manager*.

2.3 Resource abstraction layer

In FIU-Miner, the computing power of the physical servers is quantified by the number of *Worker*s, and the data mining tasks are scheduled to the *Worker*s. This mechanism, as a simplified version of system virtualization, is able to maximize the utilization of the computing resources.

To effectively manage the computing resources, each worker is associated with a profile containing the detailed specification of its capability, including the computing power, supported runtime environment, and running status. The storage of a physical server is shared by all the tenant *Worker*s, including the available database, HDFS, and local file systems.

3 Applications

FIU-Miner has been successfully applied into many real-world applications including advanced manufacturing, online spatial data analysis, and inventory data analysis.

3.1 Application I: advanced manufacturing

Advanced manufacturing such as aerospace, semiconductor, and flat display device often involves complex production processes and generates large volume of production data. In general, the production data come from products with different levels of quality, assembly line with complex flows and equipments, and processing craft with massive controlling parameters. The scale and complexity of data are beyond the analytic power of traditional IT infrastructures. To achieve better manufacturing performance, it is imperative to explore the underlying dependencies of the production data and exploit analytic insights to improve the production process.

Data analytics in advanced manufacturing, especially data mining approaches, have been targeting several important fields, such as product quality analysis [22,30], failure analysis of production [6,29], production planning and scheduling analysis [2,5], and analytic platform implementation [9,10]. However, few research and industrial efforts have been reported on providing manufacturers with an *integrated data analytical platform*, to enable automatic analysis of the production data and efficient optimization of the production process [42].

3.1.1 A concrete case: PDP manufacturing

Plasma display panel (PDP) manufacturing produces over 10,000 panels for a daily throughput in ChangHong COC Display Devices Co., Ltd (COC for short). The production line is near 6000 m, and the process contains 75 assembling routines and 279 major production equipments with more than 10,000 parameters. The average production time throughout the manufacturing process requires 76 h. Specifically, the workflow consists of three major pro-

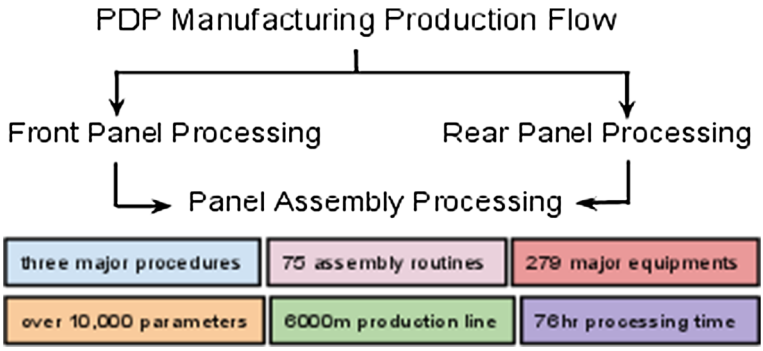


Fig. 7 PDP manufacturing production flow

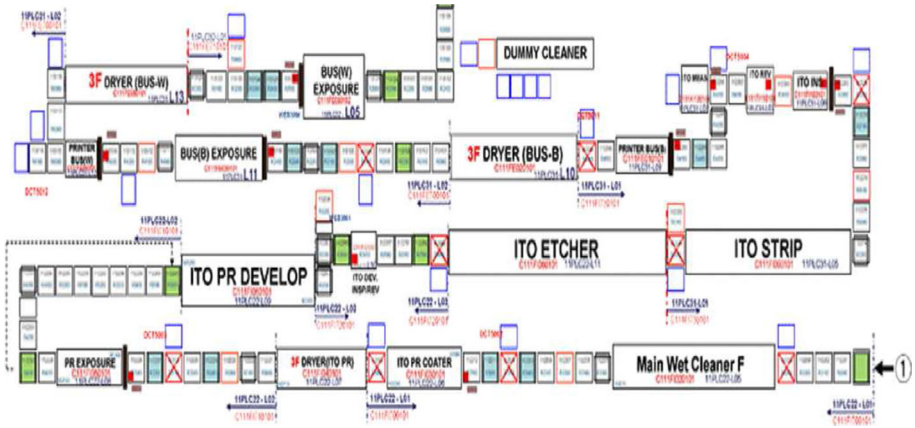


Fig. 8 An example routine in PDP workflow

cedures shown in Fig. 7, i.e., *front panel*, *rear panel*, and *panel assembly*. Each procedure contains multiple sequentially executed flows, and each flow is composed of multiple key routines. The first two procedures are executed in parallel, and each pair of front and rear panels will be assembled in the assembly procedure. Figure 8 depicts the real assembly line of one routine (Tin-doped Indium Oxide, ITO) in front panel procedure, which gives us a sense of how complex the complete production process will be.

There are 83 types of equipments in the PDP manufacturing process, each of which has a different set of parameters to fulfill the corresponding processing tasks. The parameters are often preset to certain values to ensure the normal operation of each equipment. However, the observed parameter values often deviate from the preset values. Further in the production environment, external factors, e.g., temperature, humidity, and atmospheric pressure, may potentially affect the product quality as the raw materials and equipments are sensitive to these factors. The observed values of external factors vary significantly in terms of sensor locations and acquisition time. The production process generates a huge amount of production data (10 Gigabytes per day with 30 Million records).

In daily operations, the manufactures are concerned with how to improve the yield rate of the production. To achieve this goal, several questions need to be carefully addressed, including

– What are the key parameters whose values can significantly differentiate qualified products from defective products?
– How the parameter value changes affect the production rate?
– What are the effective parameter recipes to ensure high yield rate?

Answering these questions, however, is a non-trivial task due to the scale and complexity of the production data and is impossible for domain analysts to manually explore the data. Hence, it is necessary to automate the optimization process using appropriate infrastructural and algorithmic solutions.

3.1.2 Challenges and proposed solutions

The massive production data pose great challenges to manufacturers in effectively optimizing the production workflow. Over the past three years, we have been working closely with the technicians and engineers from COC to investigate data-driven techniques for improving the yield rate of production. During this process, we have identified two key challenges and proposed the corresponding solutions to each challenge as follows.

In general, highly automatic production process often generates large volume of data, containing a myriad of controlling parameters with the corresponding observed values. The parameters may have malformed or missing values due to inaccurate sensing or transmission. Therefore, it is crucial to efficiently store and preprocess these data, in order to handle the increasing scale as well as the incomplete status of the data. In addition, the analytics of the production data is a cognitive activity toward the production workflow, which embodies an iterative process of exploring the data, analyzing the data, and representing the insights. A practical system should provide an integrated and high-efficiency solution to support the process.

Challenge 1 Facing the enormous data with sustained growth, how to efficiently support large-scale data analysis tasks and provide prompt guidance to different routines in the workflow?

Challenge 2 Facing various types of mining requirements, how to effectively adapt existing algorithms for customized analysis tasks that comprehensively consider the domain characteristics?

To address the aforementioned challenges, we design and implement `PDP-Miner`, an integrated *Data Analytics Platform* based on FIU-Miner, to support high-performance analysis. The platform manages all the production data in a distributed environment, which is capable of configuring and executing data preprocessing and data analysis tasks in an automatic way. The platform has the following functionalities: (1) cross-language data mining algorithms integration, (2) real-time monitoring of system resource consumption, and (3) balancing the node workload in clusters.

3.1.3 System overview

The overall architecture of `PDP-Miner` is shown in Fig. 9. The system, from bottom to top, consists of two components: *Data Analytics Platform* (including *Task Management Layer* and *Physical Resource Layer*) and *Data Analysis Modules*.

Data Analytics Platform is built based on FIU-Miner and provides a fast, integrated, and user-friendly system for data mining in distributed environment, where all the data analysis tasks

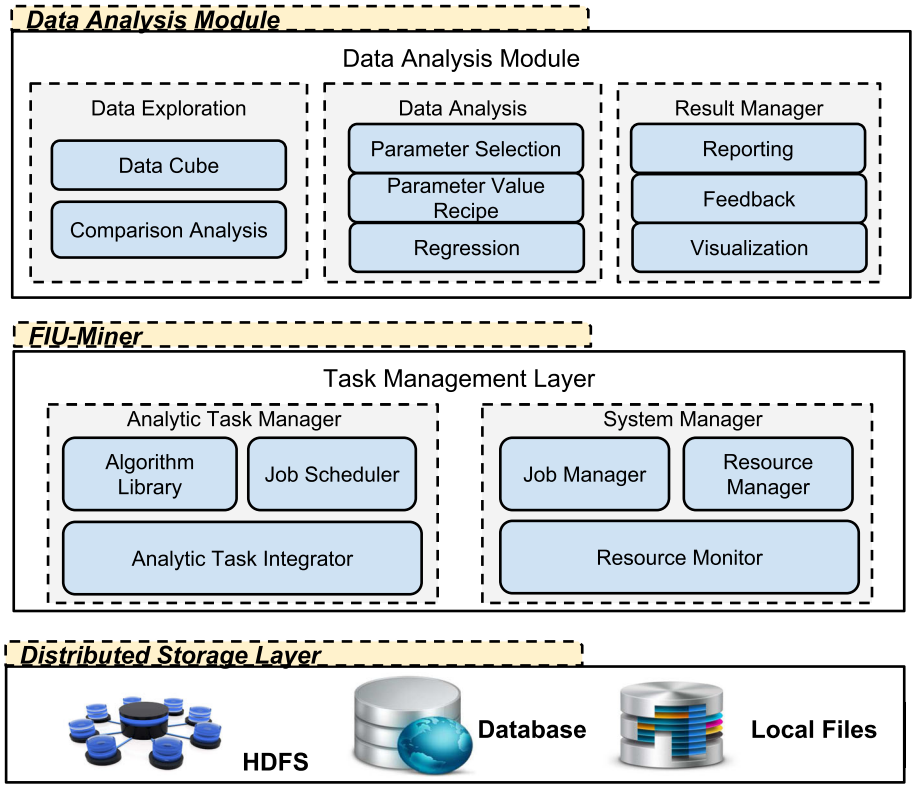


Fig. 9 System architecture

accomplished by *Data Analysis Modules* are configured as workflows and also automatically scheduled.

Data Analysis Modules provide data mining solutions and methodologies to identify important production factors, including controlling parameters and their underlying correlations, in order to optimize production process. These methods are incorporated into the platform as functions and modules toward specific analysis tasks. In `PDP-Miner`, there are three major analytic modules: *data exploration*, *data analysis*, and *result management*. Figure 10 presents the screen shots and the analysis modules of `PDP-Miner`.

3.1.4 Case study

`PDP-Miner` has been deployed as the manufacturing process analysis platform at ChangHong Corporation, one of the world's largest display product manufacturing companies located in China. This system is currently running in a distributed environment containing one 64-node computing cluster and several graphics workstations. Each node of the cluster consists of 4 Intel Xeon E5645 CPU, 32GB main memory, and 1T hard disk space.

One common task in the PDP process optimization is to extract important feature combinations that are highly related to the yield ratio. To accomplish this task, a workflow and its associated components are depicted in Fig. 11. The workflow involves the following steps:

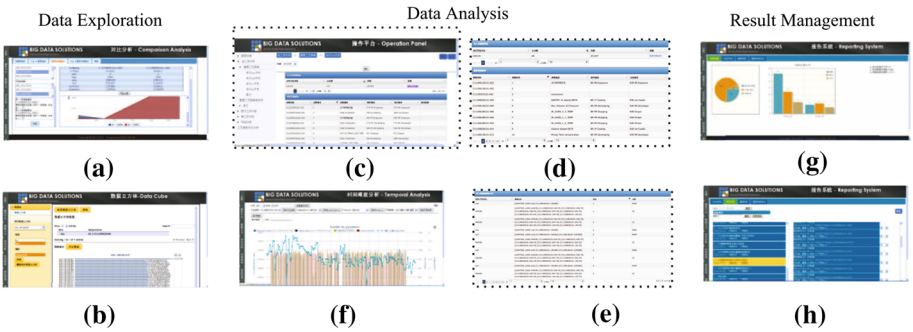


Fig. 10 `PDP-Miner` analysis modules are depicted. The Data Exploration module applies data cube and comparison analysis to explore the data distribution. The Data Analysis module integrates parameter selection, regression analysis, and discriminative analysis in the operational panel. The Result Management module provides the functionalities of visualizing the analysis result and collecting the feedbacks from the domain experts

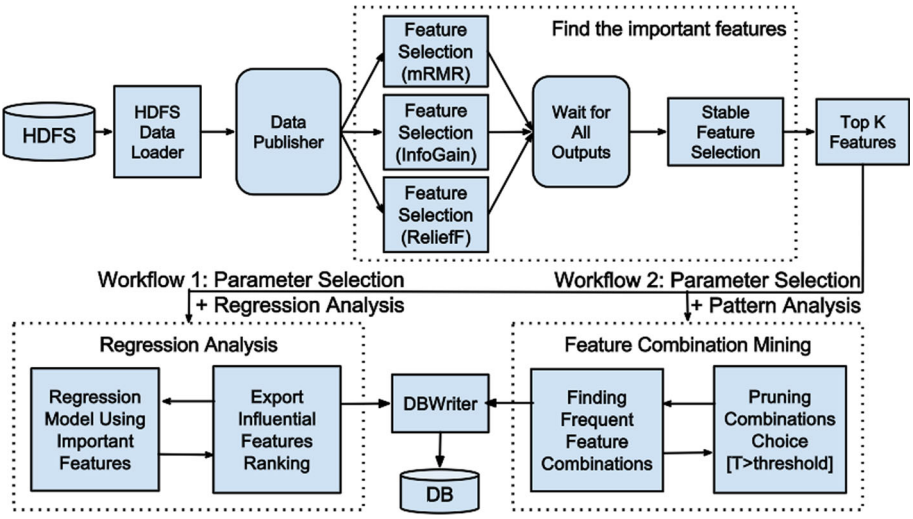


Fig. 11 Workflow for PDP manufacturing case study

1. PDP dataset is loaded from HDFS by *HDFS Data Loader*, and *Data Publisher* dispatches the dataset to three different *Feature Selection* algorithms including *mRMR* [25], *InfoGain* [31], *ReliefF* [17].
2. Important features are first extracted by the *Feature Selection* algorithms and then combined by the *Stable Feature Selection* component to output the stable features which are highly ranked by all the feature selection algorithms [35].
3. Based on the selected stable features, the *Feature Combination Mining* and *Regression Analysis* component can then be conducted based on the discovered top-K features.

Generally, three major phases (Component Preparation, Task Configuration and Task Execution) are needed to compose this complex task from the scratch. Table 1 compares the data analysis with and without FIU-Miner and demonstrates the advantages of FIU-Miner.

Table 1 Advantage of FIU-Miner is summarized by comparing data analysis with and without FIU-Miner

	Phase I: component preparation	Phase II: task configuration	Phase III: task execution
Without FIU-Miner			
Tasks	Implement the data loading and storing procedures	Writing code to manage component dependence; manual scheduling	Implement the monitor module; difficult to locate failure
Costs	A few hours for coding	Up to days	Up to days
Require data analysts to have advanced programming skills with experience in distributed environments			
With FIU-Miner			
Tasks	Rapid data loading and storing using customized interfaces	Manage dependence using GUI; scheduling transparent to users	Monitor and track mining process with user-friendly GUI; support failure diagnosis
Costs	A few clicks	Within 1 h without programming	A few clicks without programming
No programming requirements for data analysts			

During Phase I, without FIU-Miner, users can use existing tools at each component; however, they have to implement the data loading and storing procedures between components. By contrast, FIU-Miner provides customized user interfaces for data loading and storing. Users can easily prepare each component and integrate multiple algorithms without writing a single line of code.

During Phase II, without FIU-Miner, users need to manage the dependence among different components. As the number of components increases, their dependence would become more complicated. It is difficult for users to schedule the tasks in a distributed environment. On the contrary, FIU-Miner provides a user-friendly GUI to build the dependence among all the components. It can also take full advantage of distributed environments to improve the efficiency of task execution, and the scheduling is transparent to users.

During Phase III, without FIU-Miner, users need to implement the monitor to track the mining process. There is no efficient mechanism for users to locate the failure components. Comparatively, FIU-Miner provides automatic functionalities to track the running status of each component in the workflow. In addition, users can easily find out the details of the failure components.

Stable feature selection As observed in Fig. 12, the three feature subsets share only one common feature ("Char_020101-008"). Such a phenomenon indicates the instability of feature selection methods, as it is difficult to identify the importance of a feature from a mixed view of feature subsets. In general, the selected features are the most relevant to the labels and less redundant to each other based on certain criteria. However, the correlated features may be ignored if we select a small subset of features. In terms of knowledge discovery, the selected feature subset is insufficient to represent important knowledge about redundant features. Further, different algorithms select features based on different criteria, which renders the feature selection result instable.

Information gain(top10)	mRMR(top10)	ReliefF(top10)
Char_110101-004	Char_020101-016	Char_110101-009
Char_110101-003	Char_020101-004	Char_110101-008
Char_110101-005	Char_020101-008	Char_100102-079
Char_110101-002	Char_020101-009	Char_100101-199
Char_110101-006	Char_020101-010	Char_100101-208
Char_110101-001	Char_020101-007	Char_100101-212
Char_020101-008	Char_020101-006	Char_100102-013
Char_020101-017	Char_020101-003	Char_100101-213
Char_100101-168	Char_020101-014	Char_100102-081
Char_020101-013	Char_020101-013	Char_020101-008

Fig. 12 Selected features by three different algorithms: information gain, mRMR, and ReliefF

The stability issue of feature selection has been studied recently [7,13] under the assumption of small sample size. The results of these works indicate that different algorithms with equal classification performance may have a wide variance in terms of stability. Another direction of stable feature selection involves exploring the consensus information among different feature groups [18,35,37], which first identifies consensus feature groups for a given dataset and then performs selection from the feature groups. However, these methods fail to consider the correlation between selected features and unselected ones, which might be important to guide us for feature selection.

In our system, inspired by ensemble clustering [33], we employ the ensemble strategy on the results of various feature selection methods to maintain the robustness and stability of feature selection. The problem setting of stable feature selection is defined as follows. Given a dataset with M features, we employ N feature selection methods which for an arbitrary feature i return a N-lengthed vector $\mathbf{y}_i$, $i = 1, 2, \ldots, M$. Each entry of $\mathbf{y}_i$ is 1 or 0 indicating whether the feature i is selected or not by the corresponding feature selection method. Since we are concerned with whether to select a feature or not, we assume a feature i, in the form of results of N feature selection methods, $\mathbf{y}_i$, is generated from a mixture of two multivariate components, indicating selected features and unselected features, i.e.,

$$p(\mathbf{y}_i) = \sum_{j=1}^{2} \pi_j \, p(\mathbf{y}_i | \boldsymbol{\theta}_j), \tag{1}$$

where π_j denotes the mixture probability of j-th component parameterized by $\boldsymbol{\theta}_j$, in which the n-th entry θ_{jn} means the probability of the output of n-th feature selection method equals to 1. We further assume conditional independence between feature selection methods. Therefore,

$$p(\mathbf{y}_i | \boldsymbol{\theta}_j) = \prod_{n=1}^{N} p(y_{in} | \theta_{jn}). \tag{2}$$

As the result of a feature selection method in the vector $\mathbf{y}_i$ is either selecting (1) or not selecting (0) the feature i, the probability of the feature i being selected by the n-th feature selection method, i.e., $p(y_{in}|\theta_{jn})$, could be represented by a Bernoulli distribution

$$p(y_{in} | \theta_{jn}) = \theta_{jn}^{y_{in}} \left(1 - \theta_{jn}\right)^{1 - y_{in}}. \tag{3}$$

In addition, we assume that all the features are *i.i.d*. Then, the log likelihood of the unified probabilistic model is

$$\mathcal{L} = \sum_{i=1}^{M} \log \sum_{j=1}^{2} \pi_j \, p(\mathbf{y}_i | \boldsymbol{\theta}_j). \tag{4}$$

To learn the parameters π_j and $\boldsymbol{\theta}_j$, $j \in \{1, 2\}$, we use expectation–maximization (EM) algorithm. To this end, we introduce a series of hidden random variables $z_i, i = 1, 2$ to indicate $\mathbf{y}_i$ belonging to each component, i.e., the parameters of the random variable $z_{i1}, z_{z2}, z_{i1} + z_{i2} = 1$.

The iterative procedure of EM will be terminated when the likelihood of the mixture model does not change too much constrained by a predefined threshold. The hidden variable z_i indicates the probabilities of membership of feature $\mathbf{y}_i$ with respect to all mixture components. It is in some sense similar to the situation in Gaussian mixture models. The feature is assigned to the j-th component that the corresponding value z_{ij} is the largest in z_{ij}, $j \in \{1, 2\}$. As a feature selection method will eventually generate two subsets of features (selected or not), it is reasonable to make two mixture components.

After obtaining the assignments of features to components, say $\phi(z_i)$, we group features into two categories, i.e., selected/unselected. In practice, the number of selected features is significantly less than the unselected ones, and hence, the features that are not selected by any feature selection method are put into a large category. The features in the other category are final feature selection results. Specifically for each component j, we pick the features that have the membership assignment, i.e., z_{ij}, greater than a predefined threshold τ and then put these features into the selected category. In this way, we can discard features with low probabilities for selection, and hence, the stability of feature selection can be achieved by assembling different feature selection results using the mixture model.

Conclusion By taking advantage of our system, the overall PDP yield rate increases from 91 to 94%. Monthly production capacity is boosted by 10,000 panels, which brings more than 117 million RMB of revenue improvement per year.[1] Our system plays an revolutionary role and can be naturally transferred to other flat panel industries, such as liquid crystal display (LCD) panels and organic light-emitting diode (OLED) panels, to generate great social and economic benefits.

3.2 Application II: spatial data analysis

FIU-Miner has been successfully applied in TerraFly GeoCloud to support various online spatial data analysis [39,40]. With the rapid advancement in technology of geographic information system, online spatial data analysis becomes increasingly essential in various application domains such as water management, crime mapping, disease analysis, and real estate. As a consequence, many geographic applications from different domains emerge recently in the form of web applications or mobile applications. Miscellaneous requirements from diverse application domains strongly dictate efficient support for spatial data analysis.

However, the inherently complex and dynamic nature of GIS applications gives rise to great challenges for users to efficiently analyze and quickly understand the spatial data. Spatial data analysis conducted on a typical geographic application tends to involve a series of complicated interactions and may be fussy with a lot of low-level details. In addition, users from different domains want GIS systems to dynamically create map applications on their own spatial data sets. Moreover, massive spatial data analysis conducted on GIS applications is

[1] http://articles.e-works.net.cn/mes/article113579.htm.

```
SELECT
        '/var/www/cgi-bin/house.png' AS T_ICON_PATH,
        r.price AS T_LABEL,
        '15' AS T_LABEL_SIZE,
        r.geo AS GEO
FROM
        realtor_20121116 r
WHERE
        ST_Distance(r.geo, GeoFromText('POINT(-80.27, 25.757228)')) < 0.3;
```

Fig. 13 A MapQL query on real property data is given, where POINT(−80.27,25.757228) is the location of Florida International University

very resource-consuming. These big challenges require many GIS applications to be designed with innovative approaches to gain competitive advantages.

3.2.1 TerraFly GeoCloud

Recently, TerraFly GeoCloud is designed and developed to support spatial data analysis and visualization [19,40]. Point and polygon spatial data can be accurately visualized and manipulated in TerraFly GeoClould. It allows users to visualize and share spatial data related to different domains such as real property, crime, and water resources. Online analysis of spatial data is supported by the spatial data analysis engine of TerraFly GeoCloud as well. In order to efficiently support complex spatial data analysis and flexible visualization of the analysis results, MapQL, a SQL-like language, is implemented to represent the analysis queries in TerraFly GeoClould. A MapQL statement is capable of defining an analysis task and customizing the visualization of analysis results. According to the queries, the spatial data analysis engine completes the analysis task and renders the customized visualization of analysis results. For instance, given the real property data, a user may want to explore the house prices near Florida International University. The corresponding MapQL statement for such an exploration is shown in Fig. 13.

A MapQL statement extends the semantics of traditional SQL statements by introducing new reserved key words. As shown in Fig. 13, *T_ICON_PATH*, *T_LABEL*, *T_LABEL_SIZE* and *GEO* are four additional reserved words in a MapQL statement. These four reserved keywords are used in the "*expression AS <reserved word>*" clause, which provides the expression with additional semantics. In particular, *GEO* describes the spatial search geometry; *T_ICON_PATH* customizes the icon resource for the spatial search geometry; *T_LABEL* provides the icon label to be shown on the map; and *T_LABEL_SIZE* gives the size of label in pixels. The corresponding spatial query results for the MapQL statement in Fig. 13 are presented in Fig. 14.

3.2.2 Challenges and proposed solutions

Comparing with using GIS application programming interface (API), MapQL provides a better interface to facilitate the use of TerraFly map for both developers and end users without any functionality limitation. Similar to GIS API, MapQL enables users to flexibly create their

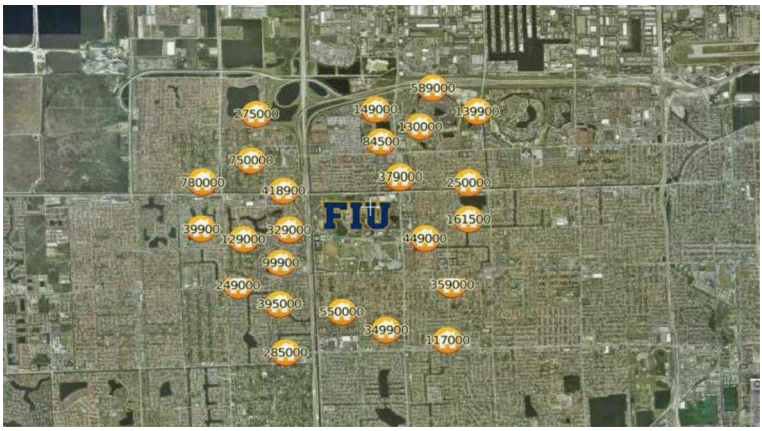


Fig. 14 MapQL query result on real property data is displayed on the map

own maps. However, our further study of TerraFly GeoCloud reveals three interesting and crucial issues which present similar challenges in other online spatial analysis systems.

The first issue is the difficulty in authoring MapQL queries. Though most of developers who are familiar with SQL can pick up MapQL quickly, the learning curve for end users who have no idea about SQL before is very steep. Authoring MapQL queries remains a challenges for the vast majority of users. As a result, it is difficult for those end users to utilize MapQL to complete a spatial analysis task from scratch.

The second issue is the complexity of a spatial analysis task. A typical spatial analysis task tends to involve a few sub-tasks. Moreover, those sub-tasks are not completely independent from each other, where the outputs of some sub-tasks are used as the inputs for other sub-tasks. According to the dependencies, a spatial data analysis task can be naturally presented as a workflow. The complexity of building such a workflow turns out to be a great obstacle for the users during the online spatial data analysis.

The third issue is the inefficiency of sequentially executing the workflow of a spatial analysis task. Even though the sub-tasks in a workflow are not linearly dependent on each other, the sub-tasks can only be sequentially executed by end users one by one. As a consequence, it fails to take advantage of the distributed environment to optimize the execution of independent sub-tasks in parallel.

The above three issues pose big challenges for users to freely and flexibly explore spatial data using online spatial analysis system. To address the issues, we develop effective solutions based on FIU-Miner. We first employ sequential pattern mining algorithms to discover the sequential query patterns from the MapQL query logs of TerraFly GeoCloud. With the help of discovered sequential query patterns, the workflows of spatial data analysis tasks are first constructed. FIU-Miner is then employed to optimize the execution of the spatial data analysis tasks by maximizing the parallelization of sub-tasks in the corresponding workflow.

3.2.3 System overview

The overview of the integrated system is given in Fig. 15. The system consists of four tiers: User Interface, Geo-Spatial Web Service, and Computing Service and Storage.

In the layer of User Interface, Map Rendering Engine is responsible for rendering the geo-objects on the map nicely based on the visualization customized by users. The component of

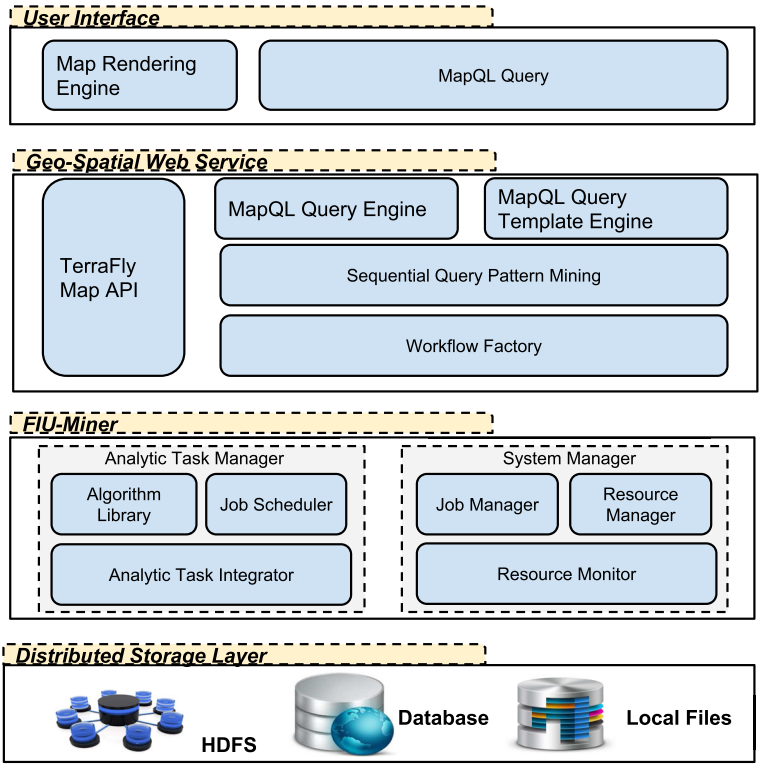


Fig. 15 Spatial analysis system overview

MapQL accepts MapQL statements that describe the spatial analysis task and the required elements for map rendering.

The second layer is Geo-Spatial Web Service. In this layer, TerraFly Map API provides the interface to access the spatial data for other components in the same layer and Map Rendering Engine in User Interface layer. MapQL Query Engine is responsible for analyzing the MapQL statements and guarantees their syntactic and semantic correctness. Sequential Query Pattern Mining is utilized to discover the sequential query pattern from the MapQL query log data. The discovered sequential query pattern can be used to generate the query templates by MapQL Query Template Engine. Users are able to rewrite the MapQL query template to construct new MapQL statements in User Interface layer. A sequential query pattern contains a sequence of MapQL queries and is used to form a workflow by Workflow Factory. Each query in a sequential pattern corresponds to a sub-task in the corresponding workflow.

For instance, a template is generated by MapQL Query Template Engine from the query logs shown in Fig. 16. The template consists of two queries about street search and hotel search. Two simple workflows are constructed in Fig. 17. These two workflows accomplish the same spatial data analysis task described in Fig. 16. In the subfigure (1) of Fig. 17, the two sub-tasks (i.e., SearchStreet and SearchHotel) are executed sequentially. However, SearchStreet needs the template parameter #arg1# as its input, while SearchHotel needs all three parameters. Provided with the three parameters, both sub-tasks can be executed independently. Thus, in the sub-figure (2) of Fig. 17, a parallel workflow is introduced to complete the spatial data analysis task. Since our data analysis tasks are scheduled by FIU-

Fig. 16 Example of a generated template

```
Template(#arg1#, #arg2#, #arg3#):

  SELECT      geo
  FROM        street
  WHERE
              name LIKE #arg1#;

  SELECT      h.name
  FROM        street s
  LEFT JOIN   hotel h
  ON          ST_Distance(s.geo,h.geo) < #arg2#
  WHERE       s.name LIKE #arg1#
  AND         h.star >= #arg3#;
```

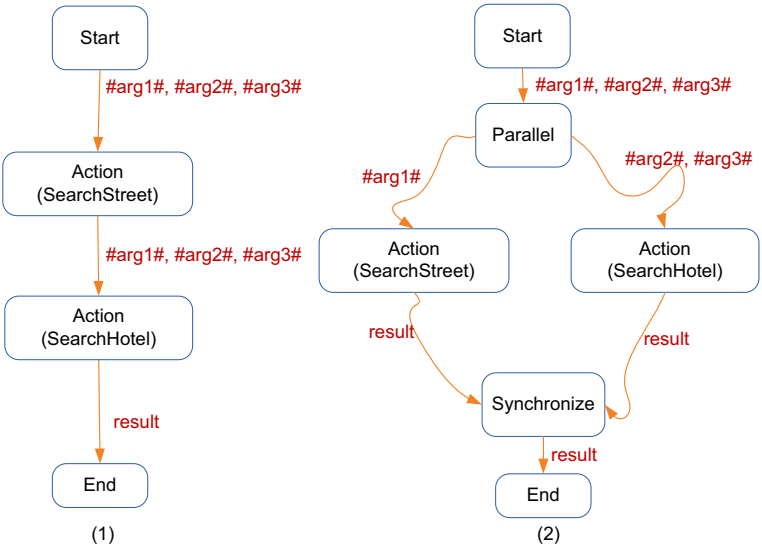


Fig. 17 Workflow constructed from query template

Miner, which takes full advantage of the distributed environment, the parallel workflow is more preferable to our system in terms of efficiency.

The third layer is Computing Service. FIU-Miner Framework takes a workflow from the second layer as an input. FIU-Miner takes the load balance of distributed environment into account to schedule the sub-tasks of a workflow for execution. The spatial data analysis library is deployed in the distributed environment. The library can be extended by developers. The computing resource is used to support the spatial data analysis tasks.

The last layer is mainly responsible for storing and managing the spatial data. All the spatial data in TerraFly are stored in the distributed file system, where replica of data guarantees the safety and reliability of system.

3.2.4 Case study

In this section, the empirical study is conducted to demonstrate the efficiency and effectiveness of our system.

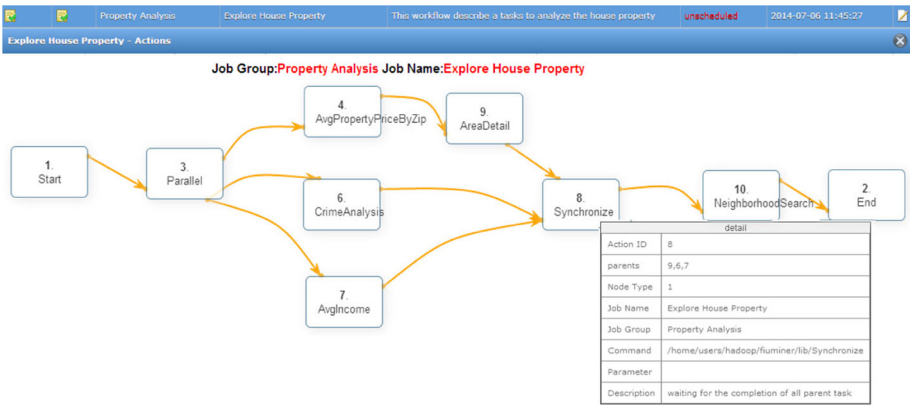


Fig. 18 Workflow of searching for house properties with a good appreciation potential. All the sub-tasks in the workflow are scheduled by FIU-Miner and are executed in the distributed environment

The integrated system provides an optimized solution to spatial data analysis problem by explicitly constructing a workflow. It supports many different applications by analyzing the corresponding datasets. One typical application scenario is to locate the house property with a good appreciation potential for investment. Intuitively, it is believed that a property is deserved for investment if the price of the property is lower than the ones of surrounding properties. Our system is capable of helping users (e.g., investors) to easily and conveniently identify such properties. According to the historical query logs collected in our system, the sequential query patterns are extracted. Based on the discovered sequential query patterns, the query templates are then generated automatically. The templates related to the house property case study are assembled to build a workflow for house property data analysis. The workflow is presented in Fig. 18.

In the workflow, there are nine sub-tasks, denoted as rectangles, to constitute the complete house property analysis task. A user can view the detailed information of each sub-task from a pop-up layer as long as the mouse hovers on the corresponding node. The workflow begins with a start node, which is used to prepare the required setting and parameters. The start node links to the parallel node with three out links. The parallel node indicates that the three sub-tasks along its out links are able to be executed simultaneously.

The AvgPropertyPriceByZip node in the workflow calculates the average property price. The overview of the analysis results is presented in Fig. 19. Note that the property prices of red regions are higher than those of blue regions. From the overview, users often interested in the regions marked with a green circle since the average property price of the region is lower than the ones of its surroundings.

In the next step, users check more detailed information on the region in the green circle by conducting the data analysis in the AreaDetail node. The spatial auto-correlation analysis on the average property prices by zip code data in Miami is conducted in this node, and the analysis results are shown in Fig. 20. Each point in the scatter plot corresponds to one zip code. Moran's I measure is applied during the auto-correlation analysis [1,14]. The points in the first and third quadrants show positive associations with its surroundings, while the points in the second and fourth quadrants indicate negative associations. Herein, users are generally interested in the points of second quadrants, having lower property price than the ones of its surrounding regions. The interesting points are marked with yellow circles. The

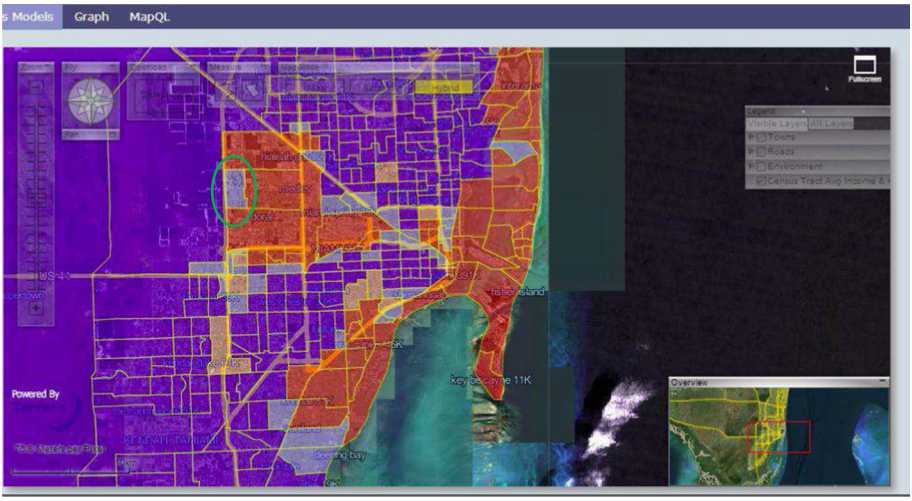


Fig. 19 Average property prices by zip code in Miami

Fig. 20 Detailed properties in Miami

analysis leads to the result that most of the cheap properties with good appreciation potential are along the Gratigny Pkwy.

In order to make sure that the areas with cheap properties have good appreciation potential, spatial data analysis to investigate the crime rate and average income of these areas is conducted. The two data analysis sub-tasks are described in CrimeAnalysis and AvgIncome nodes, respectively. These two sub-tasks are executed in parallel with the properties analysis. The Synchronize node waits for the completion of all three sub-tasks along the in links. Parallel execution accelerates the whole spatial data analysis and reduces the time cost.

Without discovering any abnormalities in the crime rate and average income, users proceed to acquire more detailed property information along the Gratigny Pkwy by executing the sub-task in the NeighborhoodSearch node. The MapQL query listed in Fig. 21 is executed in the NeighborhoodSearch node by passing the "Gratigny Pkwy" as the input parameter. The MapQL statement employs different colors to mark the regions with various property

Fig. 21 A template for searching the neighborhood, given the partial name of street

```
SELECT
  CASE
    WHEN   h.pvalue >= 400000
      THEN  '/var/www/cgi-bin/redhouse.png'
    WHEN   h.pvalue BETWEEN 200000 AND 400000
      THEN  '/var/www/cgi-bin/bluehouse.png'
    WHEN   h.pvalue BETWEEN 100000 AND 200000
      THEN  '/var/www/cgi-bin/bluehouse.png'
     ELSE    '/var/www/cgi-bin/darkhouse.png'
   END AS T_ICON_PATH,
  h.geo AS GEO
FROM
    osm_fl o
  LEFT JOIN
    south_florida_house_price h
  ON
    ST_Distance(o.geo, h.geo) < 0.05
WHERE
     o.name = #arg1# AND
      h.std_pvalue < 0 AND
      h.std_sl_pvalue > 0;
```

Fig. 22 Final analysis results

prices. The final analysis results are presented in Fig. 22. The regions painted in dark have the cheapest property prices and good appreciation potential.

With the completion of the sub-task in the NeighborhoodSearch node, the control flow reaches the end node of the workflow. Comparing to the analysis procedure without workflow, where sub-tasks can only be executed sequentially, our system takes full advantage of FIU-Miner to schedule multiple tasks simultaneously in the distributed environments. It greatly reduces the time consumed by a complex spatial data analysis task and increases the throughput of our system.

3.3 Application III: intelligent inventory data analysis

Inventory management refers to tracing inventory levels, orders, and sales of a retailing business. In the current retailing market, a tremendous amount of data regarding stocked goods (items) in an inventory will be generated everyday. Due to the increasing volume of transaction data and the correlated relations of items, it is often a non-trivial task to efficiently and effectively manage stocked goods [16].

3.3.1 Introduction

Inventory management is often an indispensable process for most retailing companies, as it has the functionalities to avoid product overstock and outages. It refers to the general process of monitoring the fluctuant flow of goods (items) into and out of an existing inventory. This process usually generates two types of time series data, representing the amount of stock in/out evolving over time. Complete inventory management also seeks to control the costs associated with inventory operations [43].

In current retailing businesses, a huge amount of items are often stored at different locations of a supply network to precede the planned course of sales. For example, in our demo case, the retailing vendor has 251,874 items in total, with 132,140 transactions daily on an average. The increasing volume of such data renders it difficult to manually manage the inventory, for example, to specify the shape and percentage of stocked items. In addition, some items may have correlations with others. For example, when a customer is buying a TV, he/she may also choose multiple auxiliary products, e.g., TV mount or DVD players. Such correlations further increase the difficulty of efficient inventory management.

3.3.2 Challenges and proposed solutions

Existing inventory management softwares, such as inFlow[2] and Inventoria,[3] provide functionalities to facilitate businesses to manage the inventory. However, most of these systems only rely on statistical analysis of the existing inventory data and have very limited capability of intelligent management, e.g., forecasting item demand and detecting abnormal patterns of item inventory transactions.

To address the limitations of existing systems, we design and develop `iMiner` to assist retailing businesses in efficiently performing inventory management. `iMiner` provides a set of key functionalities that help businesses conveniently and effectively manage huge volume of inventory data. Specifically, the system has the following merits:

- *Efficient support of large-scale inventory data analysis* The analytical platform is built on FIU-Miner to support high-performance data analysis. The platform manages all the transaction data in a distributed environment, which is capable of configuring and executing data preprocessing and data analysis tasks in an automatic way.
- *Effective management of complex analysis tasks* `iMiner` integrates appropriate data mining algorithms and adapts them to the problem of analyzing inventory data. In particular, the system (1) adopts various regression models and combines them with time series data analysis to fulfill the task of *inventory forecasting*; (2) employs context-aware anomaly detection algorithms to *identify abnormal items*; and (3) utilizes statistical regression models to perform *inventory aging analysis*.

[2] http://www.inflowinventory.com.

[3] http://www.nchsoftware.com/inventory.

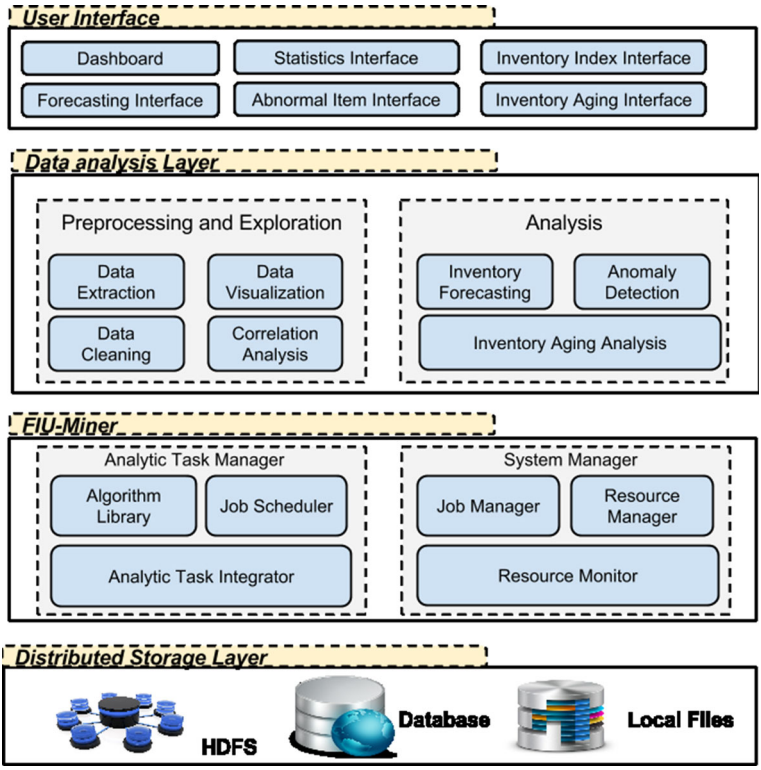


Fig. 23 An overview of the system architecture

3.3.3 System overview

`iMiner` has been designed under the following principles: (1) The methods used in the system should provide accurate, and more importantly, interpretable results; (2) the system should provide users with interactive functionalities; and (3) the system should be able to handle large-scale data analysis. Figure 23 presents an overview of the system architecture of `iMiner`. The system is composed of four layers: *User Interface*, *Data Analysis Layer*, *Task and System Management Layer*, and *Physical Resource Layer*.

Existing inventory management softwares have two limitations when applied to the practice: (1) They do not support handy algorithm plug-in and (2) they do not support large-scale analysis tasks running in parallel in heterogeneous environments. *Task and System Management Layer* provides a fast, integrated, and user-friendly system to address the aforementioned limitations, where all the data analysis tasks in *Data Analysis Layer* can be configured as workflows and automatically scheduled. This system is built on our previous large-scale data mining system, FIU-Miner [38].

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*. In Sect. 3.3.4, more details are provided by presenting our data mining solutions customized for inventory management tasks. The system also provides basic data processing and exploration functionalities.

User Interface contains various interactive interfaces for inventory operations. Specifically, 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 stock-to-use ratio, are presented in *Inventory Index Interface*, assisting users in promptly querying the status of a particular item.

3.3.4 Case study

We utilize distributed computing resources to process the huge volume of inventory data and incorporate the latest advances of data mining technologies into the system to perform the tasks of inventory management, e.g., forecasting inventory, detecting abnormal items, and analyzing inventory aging. Since 2014, `iMiner` has been deployed as the major inventory management platform of ChangHong Electric Co., Ltd, one of the world's largest TV selling companies in China.

3.4 Inventory forecasting

The primary goal of inventory forecasting is to minimize the inventory loading. Excess inventory is sub-optimal as it requires additional maintenance and cost. Hence, in inventory management, it is imperative to perform accurate forecasting so as to reduce the inventory investment and risk of obsolescence. A common practice of inventory forecasting is to predict the demand of a specific item in the future and reserve the amount of item based on the forecasting result. This often refers to as demand forecasting. We design and implement a forecasting mechanism in `iMiner` based on the past inventory transaction data. It takes as input the past transaction data of an item (as a real-valued time series) and processes the data into a list of instances. In each instance i, the amount of stock out of the item in timestamp i is treated as the label, and the values of stock out in i's previous k timestamps are regarded as the features. We then utilize various regression algorithms, such as neural network, linear regression, SVM [4], Gaussian process [28], regression tree [3], and gradient descent regression tree [8], to build regression models with optimized parameters for each item. These models are updated in a daily basis. An ensemble method [34] is employed to aggregate the predictions of these models. To enhance the interpretability of forecasting, we treat the ensemble result as the forecasting basis and further propose a dynamic model that takes into account various factors of inventory data, including seasonality, trend, and special events. Seasonality refers to the portion of item demand fluctuation accounted for by a reoccurring pattern [26]. These factors are integrated into an interactive interface shown in Fig. 24 to provide dynamic prediction for demand forecasting.

3.5 Inventory anomaly detection

Monitoring inventory index for anomaly detection is a very important task in inventory management. This problem becomes further difficult in the Big Data era as the data scale increases dramatically and the type of anomalies gets more complicated. In our system, in addition to providing traditional statistical methods (e.g., parametric and nonparametric methods) and proximity-based methods [24], we also design and develop a context-aware anomaly detection method to identify abnormal items in inventory data. Our system builds the unsupervised model of context clusters 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

周期	需求	周期因素	去周期因素影响	平滑预测	趋势	平滑	覆盖（修正）趋势	考虑趋势调整的预测	考虑周期因素的预测	事件影响指标	覆盖（修正）预测指标	覆盖（修正）固定预测	最终预测结果
201405	1078766	1	1078766	213768	213768	128261	0.8	534421	534421	1	1	0	534421
201406	860558	1	860558	559767	345999	258904	0	1207027	1207027	1	1	0	1207027
201407	973563	1	973563	680083	120316	175751	0	1119461	1119461	1	1	0	1119461
201408	1114430	1	1114430	797475	117392	140736	0	1149315	1149315	1	1	0	1149315
201409	1286661	1	1286661	924257	126782	132364	0	1255167	1255167	1	1	0	1255167
201410	792209	1	792209	1069219	144962	139923	0	1419027	1419027	1	1	0	1419027
201411	1062247	1	1062247	958415	-110804	-10513	0	932133	932133	1	1	0	932133
201412	1458783	1	1458783	999948	41533	20715	0	1051736	1051736	1	1	0	1349156

Fig. 24 Dynamic forecasting of item demand

Fig. 25 An example of anomaly detection is illustrated. The normal value range is set for the time series. An anomaly is detected if the value falls out of the normal value range

learned model, and the model can also be updated with the new data. Figure 25 shows an example of the anomaly detection interface.

3.6 Inventory aging analysis

The main purpose to monitor and analyze 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., 6 months) old, where x and y can be set by users. We provide in the system both basic tools and advanced tools to analyze the inventory aging. The basic tools allow users to visualize inventory aging distributions

Fig. 26 Attributes in inventory aging analysis

of a given item and compare current and historical inventory aging changes among different items. The advanced tools are able to help users find attributes of items correlated with overstocking. This further allows users to monitor the related attributes and especially pay attention to those items of which the value of the related attribute is above or below a predefined threshold. Figure 26 shows the interface of querying items using multiple attributes. A query is a logical conjunction of a set of literals, each of which corresponds to an attribute.

4 Resource utilization evaluation

In this section, we design two sets of experiments to investigate how FIU-Miner utilizes static and dynamic computing environments. The experiments are conducted on a testbed cluster separated from the real production system. The cluster consists of eight computing nodes with different computing performances.

Exp I Each node in the cluster is deployed with one *Worker*. We configure ten tasks with different running times in FIU-Miner. To mimic the random job submission and task execution, we schedule these jobs to start at time 0 and repeat with a random interval ($<$1 min). The system is observed in a 80-min period, and the running status of the system is recorded. Figure 27 shows how FIU-Miner balances the workloads for the underlying infrastructures, where the x-axis denotes the time and the y-axis denotes the average number of completed jobs for each *Worker*. As is shown, the accumulated number of completed jobs (the blue bars) increases linearly, whereas the amortized number of completed jobs (the white bars) remains stable. These phenomena show that FIU-Miner achieves a good balance of the resource utilization by properly distributing jobs.

Exp II We use the same job setting as in *Exp I*. To investigate the resource utilization of FIU-Miner under a dynamic environment, we initially provide four nodes, each with 1 *Worker*, and then add the other four nodes 10 min later. To emulate the nodes with different computing powers, the newly added nodes are deployed with 2–5 *Workers*, respectively. Each *Worker*

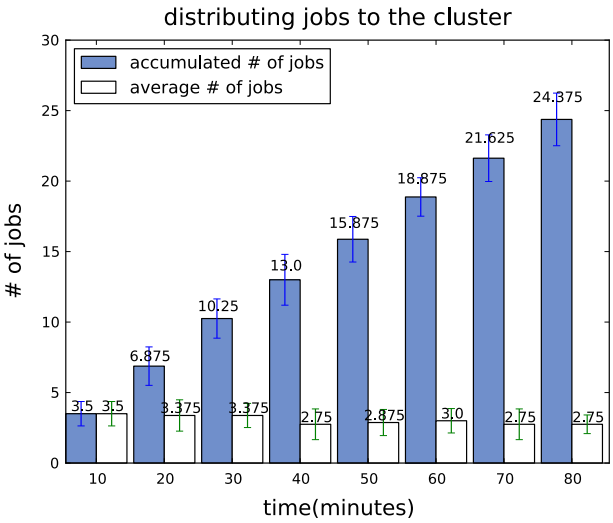


Fig. 27 Load balance in static environment

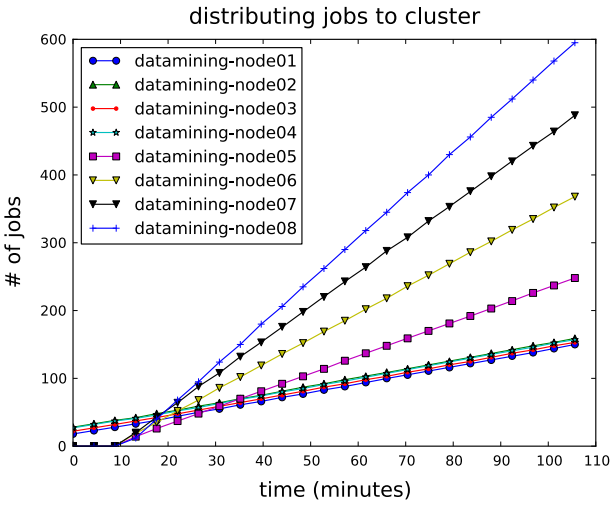


Fig. 28 Load balance in dynamic environment

is restricted to use only 1 CPU core at a time, so the node deployed with more *Workers* can have more powerful computing ability. We observe the system for 110 min, and Fig. 28 shows the number of jobs completed by each node over time. As is shown, the first four nodes with 1 *Worker* have approximately the same number of completed jobs at each observation timestamp. The slopes of the newly added nodes are higher than those of the first four nodes, since they have more computing powers. We also observe that the number of completed jobs is proportional to the number of *Workers* on each node. This experiment illustrates that FIU-Miner can balance the workloads under a dynamic environment.

5 Conclusions

In this paper, we present FIU-Miner, an integrated system that facilitates users to conduct the ad hoc data mining tasks. FIU-Miner provides a user-friendly GUI to allow users to rapidly configure complex data mining tasks. It also allows users to conveniently import and integrate arbitrary data mining programs. Internally, FIU-Miner leverages *Job Scheduler* to effectively schedule the data mining jobs and leverages *Resource Manager* to manage the underlying resource management. The three real-world applications demonstrate that FIU-Miner is effective in configuring complex task, integrating various algorithms, and executing tasks in distributed environments. In particular, since 2014, `PDP-Miner` has been deployed as the production data analysis platform of COC. By using our system, the overall yield rate has increased from 91 to 94%, which has brought more than 117 million RMB of revenue per year.[4]

Acknowledgements We would like to thank the following former members of Knowledge Discovery Research Group (KDRG) at FIU: Dr. Li Zheng, Dr. Lei Li, Dr. Yexi Jiang, Dr. Liang Tang, Dr. Chao Shen, and Dr. Jingxuan Li, for their contributions to the FIU-Miner project. We would also like to thank the High Performance Database Research Center at FIU for the cooperation on spatial data analysis. This project was partially supported by the National Science Foundation under Grants HRD-0833093, CNS-1126619, IIS-1213026, and CNS-1461926, the US Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI0001, Nanjing University of Posts and Telecommunications under Grants NY214135 and NY215045, Scientific and Technological Support Project (Society) of Jiangsu Province No. BE2016776, Chinese National Natural Science Foundation under Grant 91646116, and an FIU Dissertation Year Fellowship.

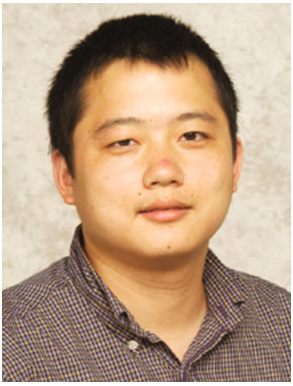
References

1. Anselin L (1995) Local indicators of spatial association—LISA. Geogr Anal 27(2):93–115
2. Belz R, Mertens P (1996) Combining knowledge-based systems and simulation to solve rescheduling problems. Decis Support Syst 17(2):141–157
3. Breiman L, Friedman J, Stone CJ, Olshen RA (1984) Classification and regression trees. CRC Press, Boca Raton
4. Chang C-C, Lin Chih-Jen (2011) Libsvm: a library for support vector machines. TIST 2(3):27
5. Chen Injazz J (2001) Planning for ERP systems: analysis and future trend. Bus Process Manag J 7(5):374–386
6. Chen W-C, Tseng S-S, Wang Ching-Yao (2005) A novel manufacturing defect detection method using association rule mining techniques. Exp Syst Appl 29(4):807–815
7. Davis Chad A, Gerick Fabian, Hintermair Volker, Friedel Caroline C, Fundel Katrin, Küffner Robert, Zimmer Ralf (2006) Reliable gene signatures for microarray classification: assessment of stability and performance. Bioinformatics 22(19):2356–2363
8. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 1189–1232
9. Groger C, Niedermann F, Schwarz H, Mitschang B (2012) Supporting manufacturing design by analytics, continuous collaborative process improvement enabled by the advanced manufacturing analytics platform. In: CSCWD, pp 793–799. IEEE
10. Gröger C, Niedermann F, Mitschang B (2012) Data mining-driven manufacturing process optimization. Proc World Congr Eng 3:4–6
11. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD explorations newsletter 11(1):10–18
12. Jiang Y, Perng C-S, Sailer A, Silva-Lepe I, Zhou Yang, Li Tao (2016) CSM: a cloud service marketplace for complex service acquisition. ACM TIST 8(1):8
13. Kalousis A, Prados J, Hilario M (2007) Stability of feature selection algorithms: a study on high-dimensional spaces. Knowl Inf Syst 12(1):95–116

[4] http://articles.e-works.net.cn/mes/article113579.htm.

14. Li H, Calder CA, Cressie N (2007) Beyond Moran's I: testing for spatial dependence based on the spatial autoregressive model. Geogr Anal 39(4):357–375
15. Lei L, Wei P, Saurabh K, Tong S, Tao L (2015) Recommending users and communities in social media. ACM Trans Knowl Discov Data 10(2):17:1–17:27
16. Li L, Shen C, Wang L, Zheng L, Jiang Y, Tang L, Li H, Zhang L, Zeng C, Li T, Tang J, Liu D (2014) Iminer: mining inventory data for intelligent management. In: Proceedings of the 23rd ACM international conference on conference on information and knowledge management, CIKM '14, pp 2057–2059, New York, ACM
17. Liu H, Motoda H (2008) Computational methods of feature selection. Chapman & Hall, London
18. Loscalzo S, Yu L, Ding C (2009) Consensus group stable feature selection. In: SIGKDD, pp 567–576. ACM
19. Lu Y, Zhang M, Li T, Guang Y, Rishe N (2013) Online spatial data analysis and visualization system. In: Proceedings of the ACM SIGKDD workshop on interactive data exploration and analytics, pp 71–78. ACM
20. MILK. http://pythonhosted.org/milk
21. MLC++. http://www.sgi.com/tech/mlc
22. Oh S, Han J, Cho H (2001) Intelligent process control system for quality improvement by data mining in the process industry. In: Dan B (ed) Data mining for design and manufacturing, pp 289–309. Springer, Berlin
23. Owen S, Anil R, Dunning T, Friedman E (2011) Mahout in action. Manning, New York
24. Pang-Ning T, Steinbach M, Kumar V et al (2006) Introduction to data mining. Pearson Education, USA
25. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE PAMI 27(8):1226–1238
26. Pindyck RS, Rubinfeld DL (1998) Econometric models and economic forecasts. Irwin and McGraw-Hill, New York
27. Prekopcsak Z, Makrai G, Henk T, Gaspar-Papanek C (2011) Radoop: analyzing big data with rapidminer and hadoop. In: RCOMM
28. Rasmussen CE (2006) Gaussian processes for machine learning. MIT Press, Cambridge
29. Shen L, Francis EHT, Liangsheng Q, Yudi S (2000) Fault diagnosis using rough sets theory. Comput Ind 43(1):61–72
30. Skormin VA, Gorodetski VI, Popyack LJ (2002) Data mining technology for failure prognostic of avionics. TAES 38(2):388–403
31. Tan P-N, Steinbach M, Kumar V (2006) Introduction to data mining. Pearson Education, USA
32. Tao L, Chunqiu Z, Wubai Z, Qifeng Z, Li Z (2015) Data mining in the era of big data: from the application perspective. Big Data Res 1(4):1–24
33. Topchy A, Jain AK, Punch W (2004) A mixture model of clustering ensembles. In: SDM, pp 379–390. doi:10.1137/1.9781611972740.35
34. Unger DA, van den Dool H, O'Lenic E, Collins D (2009) Ensemble regression. Month Weather Rev 137(7):2365–2379
35. Woznica A, Nguyen P, Kalousis A (2012) Model mining for robust feature selection. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining ACM, New York
36. Yu L, Zheng J, Wu B, Wang B, Shen C, Qian L, Zhang R (2012) Bc-pdm: data mining, social network analysis and text mining system based on cloud computing. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 1496–1499). ACM, New York
37. Yu L, Ding C, Loscalzo S (2008) Stable feature selection via dense feature groups. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, pp 803–811. ACM, New York
38. Zeng C, Jiang Y, Zheng L, Li J, Li L, Li H, Shen C, Zhou W, Li T, Duan B, Lei M, Wang P (2013) FIU-Miner: international conference on knowledge discovery and data mining, pp 1506–1509
39. Zeng C, Li H, Wang H, Guang Y, Liu C, Li T, Zhang M, Chen S-C, Rishe N (2014) Optimizing online spatial data analysis with sequential query patterns. In: Joshi J, Bertino E, Thuraisingham BM, Liu L (eds) IRI, pp 253–260. IEEE
40. Zhang M, Wang H, Lu Y, Li T, Guang Y, Liu C, Edrosa E, Li H, Rishe N (2015) Terrafly geocloud: an online spatial data analysis and visualization system. ACM Trans Intell Syst Technol 6(3):34:1–34:24
41. Zheng L, Shen C, Tang L, Zeng C, Li T, Luis S, Chen S-C (2013) Data mining meets the needs of disaster information management. IEEE Trans Hum-Mach Syst 43(5):451–464
42. Zheng L, Zeng C, Li L, Jiang Y, Xue W, Li J, Shen C, Zhou W, Li H, Tang L, Li T, Duan B, Lei M, Wang P (2014) Applying data mining techniques to address critical process optimization needs in advanced man-

ufacturing. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '14, pp 1739–1748, New York, ACM
43. Zipkin PH (2000) Foundations of inventory management, vol 2



Tao Li received the Ph.D. degree in Computer Science from the Department of Computer Science, University of Rochester, Rochester, NY, in 2004. He is currently a professor with the School of Computing and Information Sciences, Florida International University, Miami, FL. He is also a professor with the School of Computer Science, Nanjing University of Posts and Telecommunications (NJUPT). His research interests include data mining, computing system management, information retrieval, and machine learning. He received the US National Science Foundation (NSF) CAREER Award and multiple IBM Faculty Research Awards.

Chunqiu Zeng is currently a Ph.D. student in the School of Computing and Information Sciences at Florida International University. He received his B.S. and M.S. degrees in Computer Science from Sichuan University in 2006 and 2009, respectively. His research interest includes event mining, system management, recommender system, and large-scale data mining.

Wubai Zhou is currently a Ph.D. student in the School of Computing and Information Sciences at Florida International University. He received his B.S. degree in Computer Science and Technology from Wuhan University in 2012. His research interest includes system-oriented data mining, system management, and machine learning.

Wei Xue is currently a Ph.D. candidate in the School of Computing and Information Sciences at Florida International University. He started his Ph.D. studies at FIU in 2012. Before joining FIU, he received his Bachelor and Master degrees both in Computer Science at Zhejiang University in China. He mainly focuses on text mining, machine learning, and natural language processing.

Yue Huang is currently a Ph.D. candidate at Nanjing University of Posts and Telecommunications, China. His current research interests include big data, data mining, and disaster information management.

Zheng Liu is currently an assistant professor in the School of Computer Science, Nanjing University of Posts and Telecommunications, China. His research interests include summarizing and querying large graph data and mining large-scale network management data. He has published research papers in several major conferences, including ICDE, ICDM, and DASFAA.

Qifeng Zhou received the Ph.D degree in control theory and control engineering from Xiamen University, China, in 2007. She is currently an associate professor in the Institute of Pattern Recognition and Intelligent System at Xiamen University. Her research interests include data mining, machine learning and its applications.

Bin Xia is currently a visiting Ph.D. student in the School of Computing and Information Sciences at Florida International University. He is working toward the Ph.D. degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His research interests include recommender system and sequential data mining.

Qing Wang is currently a Ph.D. student in the School of Computing and Information Sciences at Florida International University. She received her B.S. and M.S. degrees in Computer Science from Zhengzhou University and Xidian University in 2009 and 2013, respectively. Her research interest includes recommender system, knowledge graph, and large-scale data mining.

Wentao Wang is currently a Ph.D. student in the School of Computing and Information Sciences at Florida International University. He received his B.S. degree in Computer Science from Sichuan University in 2014. His research interest includes data mining and machine learning.

Xiaolong Zhu is currently a Ph.D. student in the School of Computing and Information Science at Florida International University. He started his Ph.D. studies at FIU in September 2015, after receiving the MS degrees in Computer Science and Management in Information System. He received his Bachelor degree in Management in Information System from Shandong Economic University in China. His research interests include data mining, data visualization, and information retrieval.