International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 09 Issue 10 ǁ 2021 ǁ PP. 40-47

General Data Analytics Method Used In the Big Data Production Environment

Dr. N. SIVAKUMAR[1], Mr. J. MANIKANDAN[2]

[1] *Head of the Department, Department of Computer Science and Engineering, Panimalar group of institution, Chennai*

[2] *Assistant Professor, Department of Computer Science and Engineering, Panimalar group of institution, Chennai*

Abstract

The rise in the amount of available manufacturing information indicates that big data can be gathered and could be of great benefit to companies with proper deep analysis. Most small companies, however, cannot bear the overhead of a professional data analytics team. In this paper, a general data analysis method, the Generic Production Data Method (GPDS), is used to solve this issue. This framework can perform most data analytics activities in manufacturing, and users can easily perform data analysis, even though they have no previous skills or data analytics experience. We developed an abstract language, GPDS, to define the data analytics tasks of manufacturers to create such a framework. Several algorithms were picked, tuned, optimized for the factory data analysis. Some significant techniques have been developed in GPDS, such as a proper selection strategy for algorithms and an efficient algorithm for function evaluation. The case studies show the system's practicability and reliability.

Key words: manufactory; data analytics; data mining; optimization

---

Date of Submission: 06-10-2021 Date of acceptance: 20-10-2021

I. INTRODUCTION

Mining data stream is a succession of data arriving at a high speed and which is ordered by date and time and mining high speed data streams has recently come out as a growing field of study in the research. Data streams includes some research areas such as machine learning, database, artificial intelligence, statistics, Decision making, scientific inventions etc. For instance, high speed data stream applications including credit card system fraud, network intrusion, whether data forecasting, mobile applications and faster data volume applications. These applications are used to design the problems associated with high speed data.

The researchers recently focused on algorithms that fit large data sets. Mining high- stream is a way of picking up the hidden knowledge from increasingly arriving data streams, it is a way to obtain information from streams and discover a stream's development over time. The various range of approaches, such as sliding window, sampling, sketching, data structures for description, load shedding, etc., can be used to convert streaming data into a particular form for data analysis. The methods of streaming data processing such as grouping, regression, clustering and so forth are used.

In a small company data information is also the most important commodity. Nonetheless, a new manufacturing Revolution, called Industry 4.0, is emerging in an growing number of manufacturing enterprises to encourage smart manufacturing. Significant quantities of data are generated and gathered in these enterprises. Evaluation of these big data could offer tremendous opportunities for creativity, lower prices, better response to consumer needs, optimized solutions, intelligent systems, etc.[2] Similar scenarios are aimed at most current data analytics techniques used in manufacturing[3–10]. This method of data analysis can achieve a satisfactory result but is not practical, as it is not universal and requires a team of data analysis experts.

Problems also exist in commercial applications. Some making enterprises have used big data analysis platforms, e.g., Renold - Nissan combines SAP Company Suite and Sensor technology to early detect engine problems and the Mercedes implemented a Comprehensive modeler to improve efficiency by about 25%. Therefore, large-scale manufacturing enterprises are already benefiting from data analytics. However, it is impossible for small and medium making enterprises to honor the time and money required for big data analytic services, and neither do they need the complex data analytic services provided by other companies. A complete data mining process can be divided into six phases, i.e., business understanding, data understanding, data preparation, modeling, evaluation, and deployment. However, the complexity and diversity of making processes means that it is extremely difficult to make the whole data analytics process generic and suitable for a variety of

manufacturing processes and problems. At this point in the study we focus on the modeling phase and propose a data analytics system named GPDS(Generic Production Data Method), which has the following three major advantages.

We use the following three different use cases to describe GPDS more clearly. These three instances are common in manufacturing, covering most of the possibilities for data analytics in the industry. Each use case represents a type of data analytics task and shows some of the features of the system.

1.1 Inventory forecasting instance

Inventory forecasting is one of the major tasks in making and includes raw material inventory forecasting. Precision and efficiency of inventories will guarantee a smooth production process. Here we assume that the forecasting target

1.2 Car Assessment instance

When manufacturing enterprises are going to launch a new product, it needs to match the public's product. We can use a rule extraction model to extract rules regarding the public's product that were hidden in previous user feedback data on various car models. Here we take a Car Evaluation Data Set the address is carEvaluation.csv and the target is car Acceptability.

1.3 Tool Condition observe instance

Manufacturing systems are becoming more complex and are subject to failures that poor impact their reliability, availability, safety, and maintainability. For example, in the high-speed milling process, a worn milling tool might permanent damage a work. In such a case, real-time monitoring of the condition of the tools can help the operator avoid tragic events. Here we take a Steel Plates Faults Data Set as training data and the target attribute is Faults.

Generic: GPDS cover most of the data analysis tasks, e.g., c4.5 algorithm, CART, APRIORI, the two typical approaches in the supervised machine learning i.e. classification and regression and Common manufacturing data analytics tasks. Section 2 describes the related work with our GPDS method. Section 3 describes the design and definition of the abstract language and presents the GPDS framework of a data analysis task. Section 3.1 describes the modules of GPDS framework. Section 5 shows the experimental results of GPDS framework. Finally, the whole paper is summarized in Section 6 and its future work.

II. RELATED WORK

In this section, we first discuss related research on manufacturing data analysis, then introduce three use cases to fully describe GPDS. Generally the company making data are typically noisy, highly correlated, and very often are randomly missing for various reasons such as faulty sensors and computer communication errors. The objects of analysis are often closely tied to real applications. Among many fields, the semiconductor industry is the one in which the applications are implemented most. The authors of Ref. implemented a framework with a neural network to extract hidden knowledge in production data and speed up quality control without any further knowledge of the manufacturing environment; two different methods were used to guarantee the correctness of product quality tests. An improved APRIORI algorithm based on rough set theory to extract the relationship among different production attributes was proposed. In this many regression models were established to model cold-rolled steel sheet heat treatment process.

Needless to say, data analytics can also be applied to many other making fields. The authors built a decision tree with a C4.5 algorithm to extract rules from carpet manufacturing data. In this, rough set theory was used to extract the rules for solder-ball defects. Multi-layer perception networks to estimate the quality of a template were established in neural networks and CART was used to predict the quality of a glass coating. To the best of our knowledge, current studies concerning making production are mainly aimed at specific data analysis scenarios, no one has yet proposed a generic data analytics system for non-expert users. Even the most popular statistical analysis software, which has visual interfaces, cannot perform data analytics tasks until a user manually chooses a model and sets some parameters. To make data analytics available to small and medium manufacturing enterprises and non-expert users, we aim to standardize the modeling phase method for manufacturing data analytics and establish the first highly automated and generic data analytics system, i.e., GPDS. Our goal is to develop a system that, instead of requiring users to select models and set parameters manually, only requires basic information about a task, e.g., task type, target attribute, and data address, to accomplish data analytics tasks.

It is refers that the several methods that are combined and learns a target function by training a number of individual learners and combining their forecasting. The Collection of individual partial knowledge could be comes as model.

There are two typical approaches in the supervised machine learning: 1. Classification: The classification is mainly relevant with class attribute as dependent variables; this can be divided into two levels: building and testing. The building model level could be used to estimate an output from learning algorithm and the testing model level estimate the quality of building model level.2. Regression: Regression is relevant with numerical attributes as its output. The different methods such as neural networks, decision tree, rule based etc are used for the classification. These methods are contrived to build classification model where distinct passes on the stored data is possible.

III. GPDS FRAMEWORK

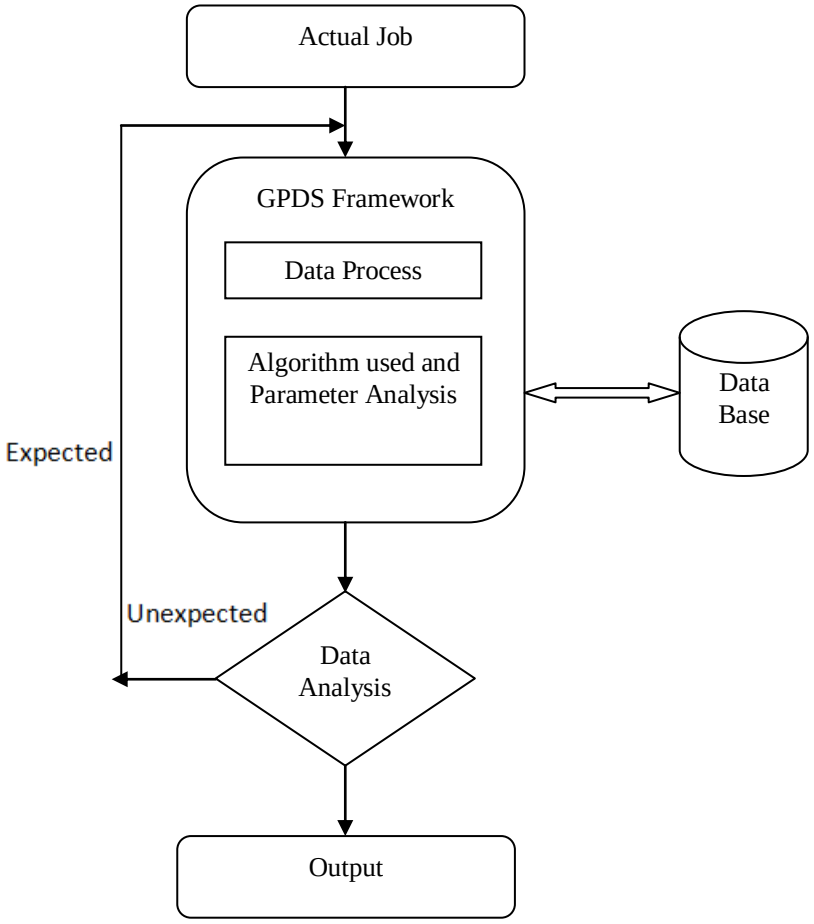

Figure 1: Architecture Diagram

GPDS is based on JavaScript. All the data analysis operations in GPDS are translated into JavaScript statements. GPDS is like an interlayer between data analysis requirements and data analysis tasks described by JavaScript, which can be processed on existing engine such as Rgui or Rstudio. The overall framework design of GPDS is shown in Fig. 1. To be highly automated and generic, GPDS mainly relies on three major system components, a data processor, an algorithm selector, and a parameter optimizer. In the next section we introduce these three components by demonstrating the GPDS workflow.

A complete data analytics process begins with a user's command. GPDS receives the task description and data processer automatically decides the attributes' type and displays it to the user. The user can adjust the attributes' type using commands if they wish. Then, algorithm selector selects the most appropriate algorithm according to the task type and data characteristics detailed in Section 4. If necessary, data processor will process the data to make sure that it is suitable for the selected algorithm. For example, multi-layer perceptron does not tolerate missing or discrete values and these must be removed. Next, if the algorithm has parameters to be determined, parameter optimizer will determine the optimum parameter values, as detailed in Section 2.

Following this, GPDS uses the selected algorithm and determined parameter settings to analyze the processed data and establish an analysis model. Finally, the analysis results are obtained and users can confirm if they are satisfied with it, if not, they can adjust the parameters or even change the algorithm until they are.

It's a basic graphical formalism that can be used to describe a system in terms of system input data, specific processing performed on that data, and this method produces the output data. It is one of the most important modeling methods, and is used to model components of the device. These elements are the system itself, the data it uses and an external entity that communicates with the system and the system's information flows. It shows how the knowledge is going through the system, and how a series of transformations change it. It is a graphical technique that represents the movement of information and the transformations applied as moves of data from input to output.

3.1 Modules used in GPDS framework

GPDS framework has 3 modules. They are Task Creation, Algorithm Selection & Predicted Result

3.1.1 Task Creation

GPDS receives the task description and data processer automatically decides the attributes' type and displays it to the user. In this module the user can fetch the training and testing dataset.

3.1.2 Algorithm Selection

Based on the given task the system can automatically select the algorithm. If user gives input task as car evaluation / Fault detection then this system can assign the algorithm as Decision tree. Else the user can give input task as Forecasting then this system can assign the algorithm.

3.1.3 Analysis results Prediction

GPDS uses the selected algorithm and determined parameter settings to analyze the processed data and establish an analysis model. In this model can perform the training and testing process. After training the dataset by using the selected algorithm then perform the result prediction for test dataset.

IV. GPDS ALGORITHM

4.1 Algorithm

Step 1: Initialize: *e* Min, *ParamList*, Null RangeList, *Rfm, ng, flag*;
Step 2: Input F( ), *Rfm, ng, P, C* *// Function starts*
Step 3: Output: *ParamList*;
Step 4: Loop

```
While RangeList is Not Null do
tempList Null;
if ParamList is Null then
e Min
    For Rfmi, nig2 range list Do
    Select value v
        if v > e then
        set True
        clear ParamList && Templist
        else if e =v then
        flag true, Rfmi
        if num =1 then     add num+ParamList
          else if 1< num < c then
          For k to Rfmi, nig do
            tempList Rfk, ng
          else divide Rfm, ng into c;
         add tempList=c
        End for
    End for
End While
```

Step 5: Return *ParmList*

4.1.1 Algorithm Description

Algorithm shows a method to determine the desired value for the parameter C. It starts with an initial *Rfm, ng*. Initial *Rfm, ng* is equally divided into *C* parts to create many number of new smaller R^0fm, ng. These

R^0fm, ng are stored in rangeList. Then, the algorithm will go through the suitable value by an iterative searching process. In every iterations, tempList is cleared, and minimum is assigned to *e* if *paramList* is empty. *e* records current highest *F(v)*. The algorithm choose a value *v* randomly for each Rfm_i , n_i *g* in rangeList . For each *v*, the algorithm gets an estimation of *F*, e.g., *F(v). If F(v)* is higher than *e*, the corresponding Rfm_i , n_i *g* will reserve, *F(v)* will be assigned to *e*, and *paramList* and *tempList* are cleared at the same time. Otherwise, its equivalent Rfm_i , n_i *g* are removed. For each Rfm_i , n_i *g*, the algorithm computes the number of available values in it depend on *P*. If only one value is available, this will be added into paramList. If more than one *C* values are accessible then Rfm_i , n_i *g* will be evenly divided into *C* parts and added into *tempList*. If there are no more than *C* available values in Rfm_i , n_i *g*, all the available values will be tried and added into *tempList* and then the *tempList* will be used as *rangeList* parameter in the next round. Iteration continues until *rangeList* is empty. All the appropriate values of the parameters are stored in *paramList*.

V. RESULTS AND DISCUSSION

The complexity of the algorithm is $\boldsymbol{A.NlOGN.a/T_{Single\,Tree}}$, where N is size of split, $\boldsymbol{a}$ is the number of available values in the parameter value range, which is partly influenced by the precision index. A parameter optimization algorithm can be also applied to the real type parameters as long as the precision index is given. We use the prediction accuracy to measure the model's performance. Improvement in model performance refers to the ratio of accuracy improvement from the parameter optimization algorithm to the original accuracy. The breast tissue data set was obtained from the UCI machine learning repository to examine the performance of the parameter optimization algorithm. We used one of the decision tree's parameters, i.e., minsplit, as an example. Figure 2 shows that decision tree will over-fit the training set if *minsplit* is too small, in this case, when the *minsplit* value keeps growing the accuracy first increases then decreases. We ran the parameter optimization algorithm to find the optimum value for minsplit, Figure 3 shows that the car making data where 500 cars have been made among them 300 cars are sold and its feedback finds 100 spares are faulty. Fig. 4 shows that the parameter optimization algorithm can improve the model performance by an average of 80.0% and at most 99%. Fig 5,6 and 7 shows that fault forecasting, Sales forecasting and manufacturing forecasting respectively.

We stored the most suitable algorithm for the data instead of each performance on it. The reason is that, when evaluating algorithms, GPDS randomly divides the data into training set and test set, and the results of the parameter optimization varied each time. Therefore, it is undependable to record the accuracy of each data set in an algorithm. GPDS adopts an idea of KNN to select the most appropriate algorithm to analyze the data. When new data arrives, GPDS extracts the features from it, and then selects the three most similar records in knowledge base based on the KNN algorithm. These three records vote for the most appropriate algorithm for the data or knowledge of data analytics.

Figure 2: Execute the Decision tree

Figure 3: Car making result

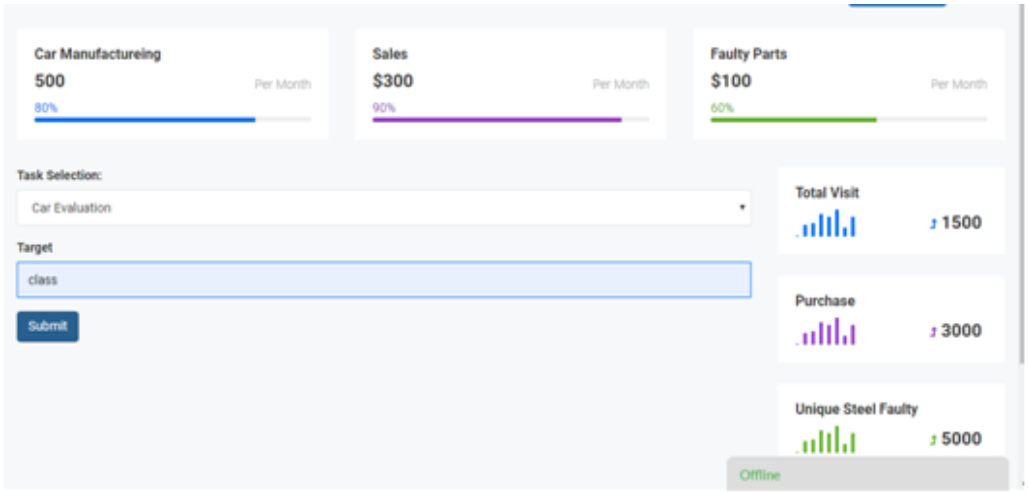

Figure 4: Parameter Optimization

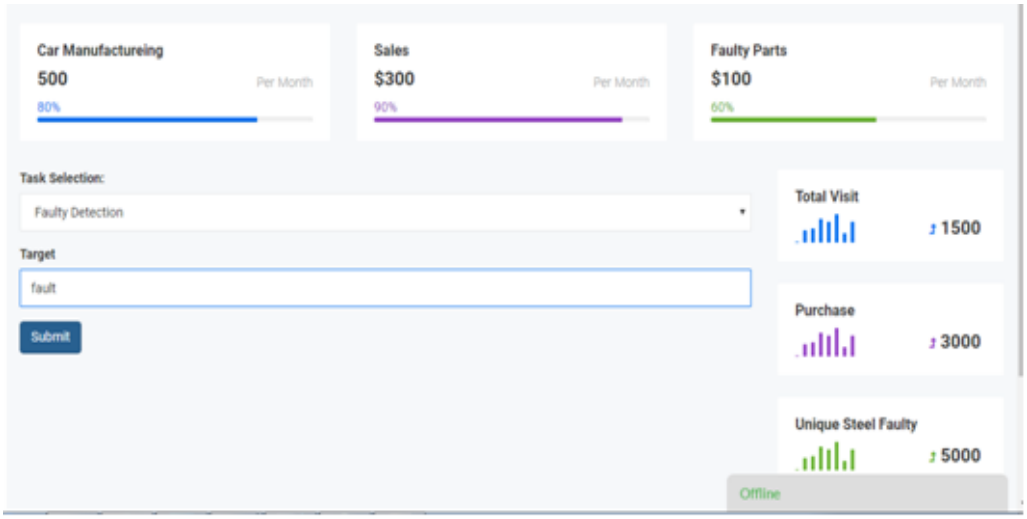

Figure 5: Fault Forecasting

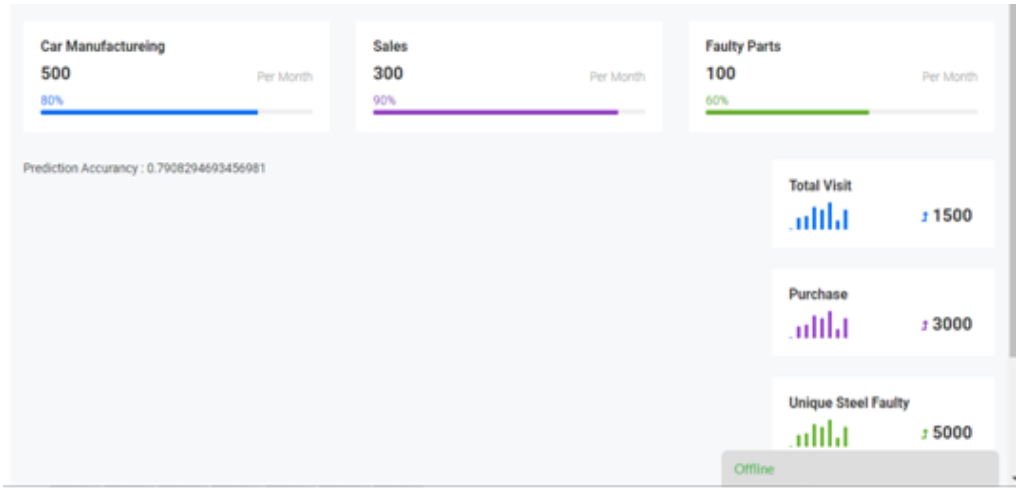

Figure 6: Sales forecasting

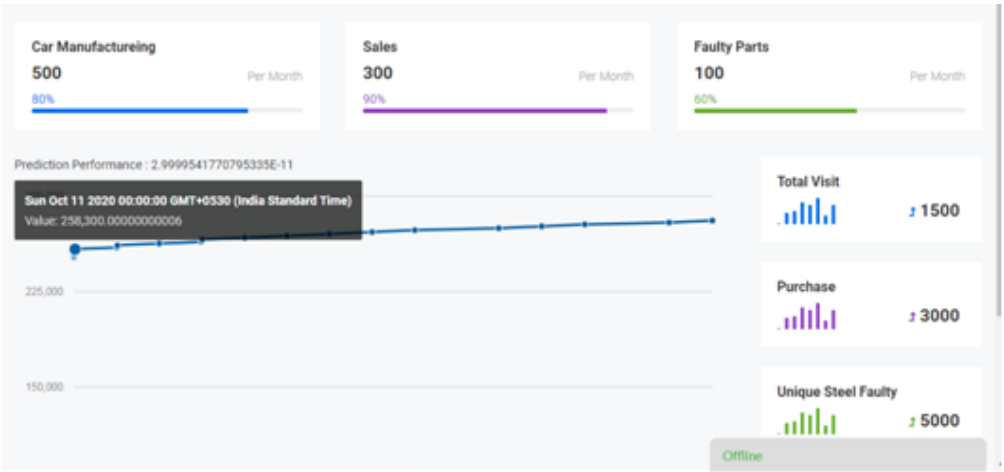

Figure 7: Manufacture forecasting

VI. CONCLUSION

A general data in the manufacturing data analytics system, GPDS, is proposed in this paper. This will make possible for small and medium manufacturers to conduct data analysis tasks using their own data and to benefit from it, even if they have no knowledge and experience of data analytics. To establish such a system, a expressive language was designed, through which the user can illustrate analysis task easily. A data base was established so that our system could select, based on the KNN algorithm, the most appropriate algorithm for the data. Several algorithms, including APRIORI, decision tree, c4.5, were integrated into the system, according to data analytics tasks familiar in the manufacturing industry, to ensure that the system covered most tasks. A number of new methods were implemented to enable the automatic accomplishment of the whole data analysis process even when the user was inexperienced. Experiments show that the system is practical and reliable enough to accomplish common data analytics tasks in manufacturing and can easily be used by non-professional. There is no doubt that such a system is a powerful tool for small and medium-sized manufacturers and could benefit thousands of manufacturing industry. However, with the restrictions of the R language, GPDS cannot handle big datasets at present. In the future, we plan to replace the R part in GPDS SparkR to make it available for use with big data.

REFERENCES

[1]. Sivakumar, N. and Anbu, S., "A Dynamic Random Multiple Decision Tree Algorithm For Mining High-Speed Data Streams", Australian Journal of Basic and Applied Sciences (AJBAS), Vol. 9, No. 20, pp. 412-417, 2015.

[2]. Sivakumar, N. and Anbu, S., "An efficient clustering algorithm for mining high-speed data streams", Journal of Chemical and Pharmaceutical Sciences (JCHPS), Vol. 9, No. 4, pp. 51-56, 2016.

[3]. Sivakumar, N. and Anbu, S., "Evaluation of a new incremental Classification tree algorithm for mining High-speed data streams", Machine Learning and Applications: An International Journal (MLAIJ), Vol. 3, No. 3, pp. 15-24, 2016.

[4]. J. A. Harding, M. S. Srinivas, and A. Kusiak, Data mining in manufacturing: A review, J . Manuf. Sci. Eng., vol. 128, no. 4, pp. 969–976, 2005.
[5]. C. Sassenberg, C. Weber, M. Fathi, and R. Montino, A data mining based knowledge management approach for the semiconductor industry, in Proc. 2009 IEEE Int. Conf. Electro/Information Technology, Windsor, Canada, 2009, pp. 72–77
[6]. A. A. F. Saldivar, Y. Li, W. N. Chen, Z. H. Zhan, J. Zhang, and L. Y. Chen, Industry 4.0 with cyber-physical integration: A design and manufacture perspective, in Proc. 21st Int. Conf. Automation and Computing (ICAC), Glasgow, UK, 2015, pp. 1–6.
[7]. M. Moghimi, M. H. Saraee, and A. Bagheri, Modeling of batch annealing process using data mining techniques for cold rolled steel sheets, in Proc. 2011 Int. Conf. Mechatronics (ICM), Istanbul, Turkey, 2011, pp. 277–281.
[8]. Q. F. Zhou, R. Y. Han, and T. Li, A two-step dynamic inventory forecasting model for large manufacturing, in Proc. 14th Int. Conf. Machine Learning and Applications (ICMLA), Miami, FL, USA, 2015, pp. 749–753.
[9]. Semeion Research Center of Sciences of Communication, http://www.semeion.it, 2017
[10]. Fazel, M. Saraee, and P. Shamsinejad, Mining time series data: Case of predicting consumption patterns in steel industry, in Proc. 2nd Int. Conf. Software Engineering and Data Mining (SEDM), Chengdu, China, 2010, pp. 501–505.
[11]. S. B. Keser and U. Yayan, A case study of optimal decision tree construction for RFKON database, in Proc. 2016 Int. Symp. INnovations in Intelligent Systems and Applications (INISTA), Sinaia, Romania, 2016, pp. 1–6.
[12]. C. Y. Chen, J. M. Hu, Q. Meng, and Y. Zhang, Short- time traffic flow prediction with ARIMA-GARCH model, in Proc. 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 2011, pp. 607–612.