

# Generating Software Agents for Data Mining: An Example for the Health Data Area

Reinier Morejón<sup>1</sup>, Marx Viana<sup>1</sup>, Carlos Lucena<sup>1</sup>

<sup>1</sup>Laboratory of Software Engineering (LES) - Pontifical Catholic University - PUC-Rio  
Rio de Janeiro, RJ - Brazil  
{rnoales, mleles, lucena}@inf.puc-rio.br

**Abstract**— Data mining has been a hot topic that attracts both database and machine learning researchers. Due to the constant growth of data volume there is an increasing need to obtain knowledge from these large data sets that are very difficult to handle and process with traditional methods. Software agents can play a significant role performing data mining processes in ways that are more efficient. For instance, they can work to perform selection, extraction, preprocessing and integration of data as well as parallel, distributed, or multisource mining. This paper proposes a framework based on multi-agent systems to apply data mining techniques to health data sets. For a first usage scenario, we use a data set for hypothyroidism and run two mining processes in parallel.

**Keywords.** *Multi-Agent Systems, Data Mining, Machine Learning*

## I. INTRODUCTION

Since its origins, computing science has always represented an accelerated path of constant evolution and transformation, at the same time implying a similar progression of information generation. Not only is this evidenced by the spectacular growth of traditional Internet usage [21] and the equally outstanding rise of the Internet of Things [26] [9], but also by the large volume of data generated by the healthcare industry [25].

Health care services are an expanding source of vast quantities of information about patients' health. Beyond the traditional approaches of clinical practice, databases in today's modern health institutions automatically collect "structured data relating to all aspects of care" [17] such as "diagnosis, medication, test results and radiological imaging data" [17]. These phenomena have a fundamental underlying element: the huge volume of data in constant expansion - data that can vary in types and sources [17], and need to be mined to produce significant knowledge. A task like this is very difficult to perform in traditional ways. Here is where Data Mining plays its role.

There are sources such as: (i) legacy systems like devices attached to, or inside, the human body to monitor and maintain human health and wellness; (ii) disease management and fitness tracking [26], and (iii) huge data sets of medical records for research studies [12]. These sources represent an astonishing volume of data available. Obtaining meaningful knowledge is very important in this area. As Durairaj said in [10], the application of data mining algorithms in the healthcare industry plays "a significant role in prediction and diagnosis of diseases."

Although there are already some solutions using data mining to deal with specific health data - for example, diabetes [12] or heart disease [24], it is unusual to see architectural models designed for knowledge discovery from medical data records that have been applied to more than one disease. Adjustable solutions can adapt from one disease to another simply by setting new instances. We can cite hypothyroidism [15] and its complications [7], which brings an extensive possibility of research. Therefore, in this research we will be mining a hypothyroidism data set.

In this context, we present the framework Java Agent Framework for Health Data Mining (JAF4HDM). This framework aims at providing support to build and operate different agents to run classification-mining methods. In order to take advantage of software agents to better implement data mining techniques over health data systems, such framework extends the JADE framework [2][3], which already provides support to building autonomous and pro-active agents. By using these new features, it is possible to build agents for classification-mining methods to: (i) work on one data set, (ii) perform the training process, and (iii) predict new unlabeled instances by using the rules obtained as a result of the training process.

This paper is organized as follows: Section II presents a background about the relationship between Data Mining, Machine Learning and Multi-Agent Systems. Section III has a brief background of WEKA. Section IV presents the JADE framework. Section V discusses some related work. Section VI presents JAF4HDM. Section VII describes a case study by showing how agents help to classify hypothyroidism data. Section VIII presents our conclusion and future work.

## II. THE RELATIONSHIP BETWEEN DATA MINING, MACHINE LEARNING AND MULTI-AGENT SYSTEMS

From the onset five important trends have been present [29] in computing: ubiquity, interconnection, intelligence, delegation and human-orientation. These trends have grown at a steady pace, which means more processing power; distributed and big scale architectures; more automation, and constant evolution for new solutions in the computing field. Likewise, the volume of information generated by computing systems has grown consistently and overwhelmingly.

The volume of data in the world and in our lives seems to increase steadily—and there is no end in sight [28]. Nowadays, information technologies (ITs) overwhelm us with data as ITs systems record every action we take when we use

them. This happens not only with personal actions but also with all kinds of decisions and operations that take place in the world of commerce and industry [28]. In addition, the huge volume of data that health care transactions generate are too complex and voluminous to be processed and analyzed by traditional methods [18].

Throughout recent decades, the magnitude of data volume and ubiquity, for example, has resulted in two of the most prominent, dynamic, and exciting research areas in information sciences: (i) autonomous agents and multi-agent systems (MASs), and (ii) data mining and knowledge discovery in databases (KDD) [8].

A software agent is a computer system situated in some environment, capable of autonomous action in order to meet its design objectives [29]. In other words, an agent can figure out by itself what needs to be done without the need to receive an explicit order of what to do at any given moment.

A multi-agent system is one that consists of a number of agents, which interact with one another, typically by exchanging messages through some computer network infrastructure. To interact successfully, these agents must be able to cooperate, coordinate, and negotiate between themselves [29].

On the other hand, we have data mining techniques that have different objectives from those of multi-agent systems. As in [28], data mining is the process of discovering patterns in data. The process must be automatic or (more usually) semiautomatic. The patterns discovered must be meaningful in that they lead to some advantage—usually economic.

Experience shows that in many data mining applications, the explicit knowledge structures acquired and the structural descriptions are as important as—and often much more important than—the ability to perform well on new examples [28]. Then, if MAS and data mining have evolved with well-defined, yet distinct, aims and objectives [8], how are they related?

Typical problems in agents that could find satisfactory solutions in data mining include multi-agent learning, adaptation, evolution, and behavior analysis. For instance, knowledge extracted through data mining could provide more stable, predictable, and controllable models for dispatching and planning, or it can assist in the self-organization and evolution of multi-agent systems in acceptable directions. On the other hand, software agents can support and enhance the knowledge discovery process in many ways. For example, agents can contribute to data selection, extraction, preprocessing, and integration and they are an excellent choice for parallel, distributed, or multisource mining [8].

### III. WEKA

Waikato Environment for Knowledge Analysis (WEKA) is a collection of machine learning algorithms for data mining tasks. WEKA provides several implementations of learning algorithms to apply to data sets. Furthermore, it includes a diversity of tools for transforming data sets; it also allows preprocessing a data set, feeding it into a learning scheme, and

analyzing the resulting classifier and its performance—all without writing any program code at all [11].

WEKA is easy to extend thanks to simple API and plug-in mechanisms and facilities that automate the integration of new learning algorithms by means of its graphical user interface [16].

The workbench includes methods for the main data mining problems: regression, classification, clustering, association rule mining, and attribute selection. In addition, it provides many data visualization facilities and data preprocessing tools. All algorithms take their input in the form of a single relational table obtained from a file or generated by a database query [11].

There are three ways of using WEKA: (i) to apply a learning method to a data set and analyze its output to learn more about the data; (ii) to use learned models to generate predictions on new instances (See Section VII), and (iii) to apply several different learners and compare their performance in order to choose one for prediction [11].

For a Java environment, it is possible to solve a learning problem without writing any machine learning code, just by accessing the available WEKA algorithms [11].

### IV. JADE

The Java Agent Development Framework (JADE) is a software framework to support the development of agent applications in compliance with the FIPA 2000 specifications for interoperable intelligent multi-agent systems [6]. Fig. 1 shows a standard model of an agent platform, according to FIPA.

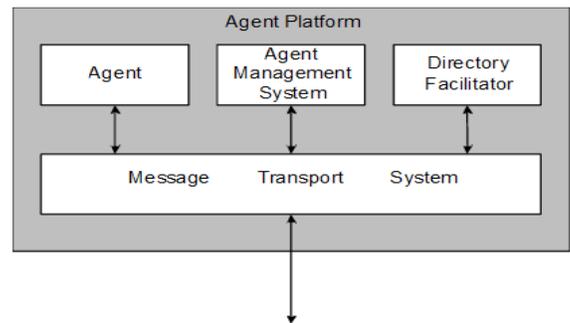


Figure 1. Reference architecture of a FIPA Agent Platform [3].

According to Fig. 1, three elements are essential in a FIPA compliant agent platform. First, the Agent Management System (AMS)—only one per platform—which is the agent that exerts supervisory control over access to, and use of, the Agent Platform. Each agent must register with an AMS in order to get a valid agent identifier (AID) [3]. Second, the Directory Facilitator (DF), the agent that provides the default yellow page service in the platform. Finally, the Message Transport System, (or Agent Communication Channel (ACC)), the software component controlling all the exchange of messages within the platform, including messages to/from remote platforms.

There are two different points of view for describing the JADE system: first, JADE is a runtime system for FIPA

compliant Multi-Agent Systems, supporting application agents every time they need to exploit some FIPA covered feature such as the life-cycle management of the agent. Second, JADE is also a Java framework for developing FIPA-compliant agent applications, making FIPA standard assets available to the programmer through object-oriented abstractions [6].

JADE includes both the libraries (i.e. the Java classes) required for developing application agents, and the runtime environment that provides the basic services. This environment must be active on the device before the agent’s execution moment. Each instance of the JADE run-time is called container. All containers taken as a whole form the platform, which provides a homogeneous layer that hides the complexity and the diversity of the underlying tiers (hardware, operating system, type of network, JVM) from agents and application developers [5].

### V. RELATED WORK

Some approaches proposed [12, 8, 14] perform the data mining process by using Multi-Agent Systems. For instance, in [14], the authors proposed a MAS Technology for Distributed Data Mining (DDM) and Distributed Classification (DC). They addressed as a key problem the fact that data sources are distributed, heterogeneous and, as a rule, of large scale [14]. In addition, the authors addressed as an issue that the Distributed Data Mining MAS design technology presumes collaborative activities of agent-mediated distributed users [14]. Therefore, they proposed an architecture of DDM and DC MASs and a technology for their design. They further proposed a number of well-developed protocols supporting agent-mediated design of applied DDM and DC MASs [14].

Gao, Denzinger and James [12] proposed a cooperative multi-agent data mining model to apply to medical data on diabetes. They developed CoLe, a model for cooperative agents for mining knowledge from heterogeneous data [12]. They designed that model having in mind the use of various agents to perform a data mining process over a completely heterogeneous data set by taking advantage of the cooperation between these agents to generate more significant and more useful structures of knowledge that no individual mining agent can produce alone [12]. This model works by making iterations of the mining process, and for each iteration, the discovered knowledge can be useful for the next one. Based on the proposed model, the authors implemented, and tested, a multi-agent system for mining diabetes related data, by having two agents running mining algorithms and another agent with methods to produce hybrid rules. As a result of the experiments that were carried out, the implemented system outperformed the individual mining algorithms, with better and high-quality rules. Another discovery was the confirmation of a close relation between diabetes and hypertension. An improvement to this work could be the creation of a generic framework according to the CoLe model. The authors implemented a solution to diabetes-related heterogeneous data sets, making it clear in [13] that the CoLe work is specific to Early Diabetes Detection.

The authors in [8] presented the synergies between Data Mining and Software Agents, creating the concept of Agent Mining. They generated a Disciplinary Framework - “a high-level research map of agent mining as a disciplinary area” [8]. They also gave an overview of what agent mining means: “a young scientific field that advocates and studies the integration of agent technology and data mining” [8]. In addition, they performed a deep analysis of agent technology and data mining, speaking about the advantages and flaws of each and about the unquestionable benefits of them working together. In short, the authors show that the potential of “agent mining for the mutual enhancement of both fields and for the creation of super-intelligent system” [8] is promising and optimistic.

Agent Academy (AA) [23] is an open-source framework and an integrated development environment to create Software Agents and Multi-Agent Systems. The authors developed by using JADE and WEKA APIs. The framework provides embedded intelligence to agents by inserting decision models generated by a previous data mining process over a background data-specific application. The authors applied AA only to a supply chain management industrial scenario. There is no evidence about the applicability of AA to a health-oriented scenario.

### VI. JAF4HDM: A SOFTWARE FRAMEWORK FOR HEALTH DATA MINING

This section describes the main elements required to understand the framework proposed in this paper. In addition, we will provide an overview of the JAF4HDM framework and discuss the different components, including the kernel (frozen-spots) and flexible points (hot-spots) [20].

#### A. The Framework

The JAF4HDM framework is implemented using software agents, as illustrated in Fig. 2. The JAF4HDM extends JADE [6], a FIPA compliant framework to implement Multi-Agent Systems developed in Java. In addition, it makes use of WEKA API algorithms to process health data. The framework comprises three main functions: (i) a workover of different data sources to preprocess the data and make it ready for training, (ii) a training process, by executing a desired mining method for the previously prepared data, and (iii) the execution of the prediction task, performed by agents using the rules generated by the training process over new unlabeled instances.

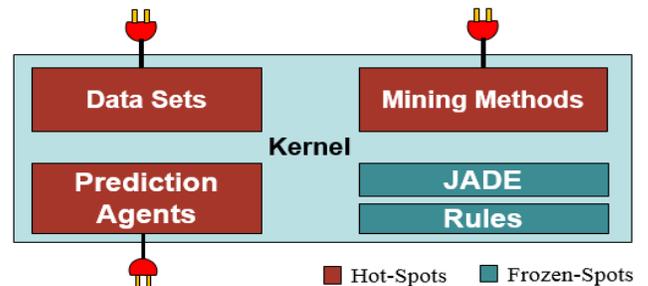


Figure 2. JAF4HDM framework architecture.

### B. The JAF4HDM Architecture

The JAF4HDM framework supports the creation of a health data mining process that shows how to classify a given data set, such as hypothyroidism, which we show in Section VII.

We based the JAF4HDM mining process on the model proposed in [12] nuanced with some conceptual elements proposed by [30], with the difference that we ran several mining methods on a single homogeneous data set by making predictions based on each method-generated rules instead of having an agent combining all the generated rules in one single set of rules. In addition, our idea is to apply our solution to several illnesses instead of just one.

Fig. 3 shows that the process begins with one agent that prepares a single, homogeneous data set for training. Then, several agents will train different mining methods over that data set, obtaining a trained model with generated rules for each method. Finally, by using these rules other agents will predict the classification of each instance in a provided unlabeled data set.

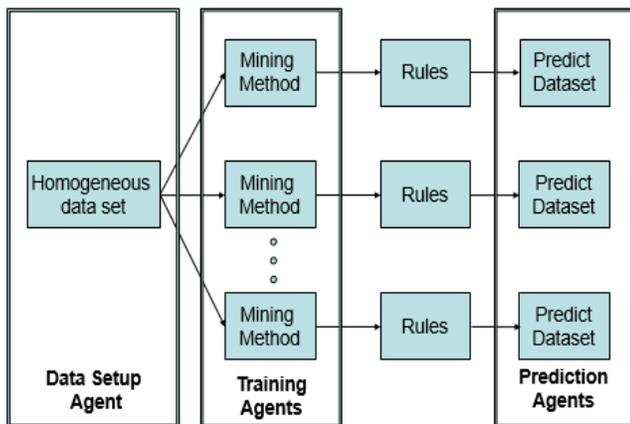


Figure 3. JAF4HDM mining process.

There are three kinds of agents (See Fig. 3) as part of JAF4HDM, described in Table 1:

TABLE I. AGENTS DESCRIPTION

Name	Description
<b>DataSetupAgent</b>	Prepares the data that has to be processed for training by selecting the source of the data, extracting the data and getting it ready for the training process.
<b>TrainingAgent</b>	Over the previous prepared data, and with a selected classifier algorithm, this agent will train and save a model, so that it can be used in future predictions.
<b>PredictionAgent</b>	This agent will take an unlabeled data set and will classify it with an existing trained model.

### C. Hot-Spots and Frozen-Spots

As previously mentioned, JAF4HDM is an extension of the JADE framework, and therefore, they share the same core and hot-spots. The process used for the communication between software agents, and the identifiers of agents are examples of hot-spots that JAF4HDM inherited from JADE. In addition, JAF4HDM defines other specific hot-spots, which are:

**Data Sets** (*DataSetupAgent* Class): There are several ways to interact with a data set, from the format of the source (data base, csv file, arff file, etc) to the way that data is prepared (divided by cross validation split, or by percentage split). An agent will perform this task to make it more dynamic.

**Mining Methods** (*TrainingAgent* Class): There are many methods to carry out data mining - all different in some ways. Each instance must use different algorithms to teach their models. *TrainingAgents* will train and generate models by performing mining methods. In this first stage of the work, we use WEKA API to perform the algorithms; in the next stages, JAF4HDM will be able to use other APIs.

**Prediction Agents** (*PredictionAgent* Class): They must produce knowledge, but the ways to achieve this can differ depending on the selected methods. Consequently, these agents can conduct the new knowledge generation process in different ways.

The frozen-spots in this context (also named Kernel) are composed of: (i) the JADE framework; (ii) classes that work with the mining methods, and (iii) a learned set of rules saved in models to make subsequent predictions. Fig. 4 shows the class diagram of JAF4HDM, and its hot- and frozen-spots.

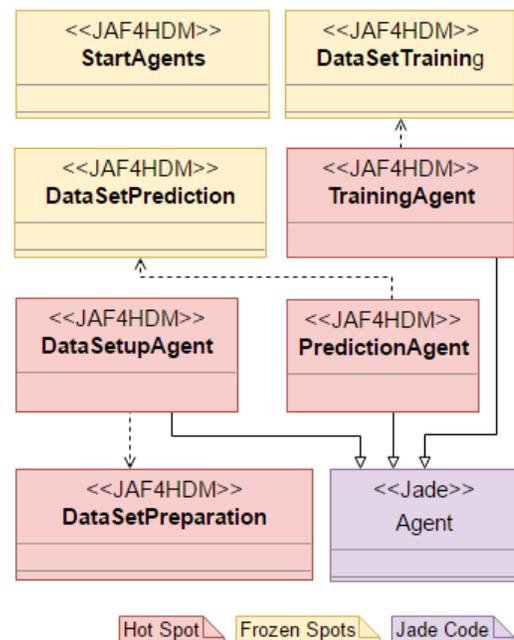


Figure 4. JAF4HDM Class Diagram.

To implement and deploy one application, based on the JAF4HDM framework, we need to implement at least one instance of each defined agent in order to successfully execute

the mining process in the usage scenarios. To build an application based on JAF4HDM, there must be at least one *DataSetupAgent*, one *TrainingAgent* and one *PredictionAgent* for an effective execution of the mining process.

## VII. USAGE SCENARIO: MINING ON HYPOTHYROIDISM DATA

A framework is classified according to its extensibility; it can be used as a white box or a black box [20]. In white box frameworks, an instantiation is only possible through the creation of new classes. A black box framework allows a developer to produce new instances using configuration scripts. This type of framework automatically creates the classes and source code for the chosen configuration, which facilitates the instantiation process [20]. For now, in its first version, JAF4HDM is only a white box framework.

To test the framework applicability, it is necessary to implement one instance, which in this case is a specific scenario to predict unlabeled instances of hypothyroidism related data.

To clarify it, the new system first will train two models, using PART and Decision Table classification algorithms (available in WEKA API) by mining a data set of Thyroid disease records. The Garavan Institute and J. Ross Quinlan, New South Wales Institute, Sydney, Australia supplied this data set [27]. Then, with the trained model, we will predict the classification of each instance in another data set in the same format as the first one, but unlabeled.

Next, an explanation about how the framework in the current usage scenario was instantiated. The steps are as follows:

A. Put both *arff* file data sets (labeled to train and unlabeled to predict) in the root folder of the project. As a good practice, both the labeled and unlabeled filenames are almost the same, only differentiated by the string “-test” added at the end of the unlabeled data set filename.

B. The current version is for running one or more mining methods over a single data set, Therefore, to put in context the application with the right data set, two attributes must be set up manually with default values. These are: (i) *dataset\_name* in *DataSetupAgent* Class with the name of the labeled data set, and (ii) *datasettest\_name* in *PredictionAgent* Class with the name of the unlabeled data set.

C. We built the current JAF4HDM instance to process the data set by using two classifier algorithms PART and Decision Table from WEKA. Therefore, it is mandatory for each algorithm to extend the *TrainingAgent* Class and the *PredictionAgent* Class.

D. For each extended *TrainingAgent*, it is required to set the variables *modeltotrain* and *receiver*, which means, respectively, the algorithm to use and the name of the extended prediction agent that will use its results as entry.

E. For each extended *PredictionAgent*, it is required to set the variables *predictmodel* and *datasettestname*, which means, respectively, the same algorithm used by the extended

*TrainingAgent* (whose results served as entry) and the name of the unlabeled data set to be predicted.

F. The agents must be defined in *StartAgents* Class before execution.

During and after the execution process, the system generates useful information such as the obtained rules after each training agent execution. Table 2 shows a fragment of the rules generated by the training agent that runs the PART algorithm over a hypothyroidism related data set. The system also generates a file containing the model obtained after the training process and a file containing a new data set already labeled as a result of the prediction process.

TABLE II. SAMPLE OF RULES GENERATED BY THE TRAINING AGENT THAT RUNS THE PART ALGORITHM

1	$(TSH \leq 6) \rightarrow \text{negative}$
2	$((FTI \leq 64) \wedge TSH\_measured \wedge T4U\_measured \wedge \neg \text{thyroid\_surgery} \wedge (T3 \leq 2.3)) \rightarrow \text{primary\_hypothyroid}$
3	$(\neg \text{on\_thyroxine} \wedge TSH\_measured \wedge \neg \text{thyroid\_surgery} \wedge (TT4 \leq 150) \wedge (TT4 > 48) \wedge TT4\_measured \wedge (T3 \leq 2.3)) \rightarrow \text{compensated\_hypothyroid}$
4	$\text{on\_thyroxine} \rightarrow \text{negative}$
5	$\neg TSH\_measured \rightarrow \text{negative}$
6	$(\text{thyroid\_surgery} \wedge (TT4 > 69)) \rightarrow \text{negative}$

As a sample of what a rule means in the data set context, we have in Table 2, line 1, the first generated rule, “ $(TSH \leq 6) \rightarrow \text{negative}$ .” For one instance that means that if the TSH (Thyroid Stimulating Hormone) measurement is less than or equal to 6, then the instance is classified as negative for hypothyroidism. The second line shows that if one instance has an FTI (Free Thyroxine Index) measurement less than or equal to 64 and both TSH and T4U (it is a test to measure Thyroxine) have been measured, and there has not been a thyroid surgery, and the T3 (Triiodothyronine) measurement is less than or equal to 2.3, then the instance is positive for primary hypothyroid. The same applies to the next rules. We use these learned rules after training to predict or classify new instances in real time. Table 3 shows the result of a prediction task for one hypothyroid-related unlabeled data set.

TABLE III. RESULTS AFTER PREDICTING UNLABELED DATA SET INSTANCES

Category	Amount
Total number of instances:	19
Instances classified as ‘negative’	13
Instances classified as ‘compensated hypothyroid’	4
Instances classified as ‘primary hypothyroid’	2
Instances classified as ‘secondary hypothyroid’	0

Table III shows how a prediction agent classifies each element of the unlabeled data set. For example, we demonstrate the prediction according to the use of one already trained model (based on PART algorithm) over one unlabeled data set containing 19 instances of hypothyroid-related data.

Out of 19 instances, the prediction classified 13 as negative, 4 as compensated hypothyroid, 2 as primary hypothyroid and none as secondary hypothyroid.

### VIII. CONCLUSION AND FUTURE WORK

This paper proposes the JAF4HDM framework for Health Data Mining, with the objective of providing support to build and operate different agents to run classification-mining methods to: (i) work on one data set, (ii) conduct the training process, and (iii) predict new unlabeled instances by using the rules obtained as a result of the training process. We can verify the applicability of the JAF4HDM framework by using the scenario presented in Section VII, where instantiated agents are responsible for running a mining process over a hypothyroidism data set to predict new instances later.

For future work our aim is to build features where: (i) the framework must be able to work with heterogeneous data sets using different methods on different types of data. It is necessary to conduct studies about heterogeneous data sets; (ii) the *DataSetupAgent* (See Table I) must be able to manage dynamically the extraction and preparation of data from its source in order to deliver to each *TrainingAgent* the adequate data set to perform the training process and, thus, avoiding to run on the whole data set directly, and (iii) The prediction process must have one agent that can receive the learned models from each training agent, find the way to mix the rules of these models, evaluate them and put the better ones into a final model. This agent will obtain meaningful information from models and will provide a feedback to trainers to improve their own work. In addition, it will help the *DataSetupAgent* to form the slices of data to process in the next iteration.

### REFERENCES

- [1] BAOYUN, Wang. Review on internet of things [J]. Journal of electric measurement and instrument, 2009, vol. 12, p. 1-7.
- [2] BELLIFEMINE, Fabio, et al. Jade administrator's guide. TILab (February 2006), 2003.
- [3] BELLIFEMINE, Fabio, et al. Jade programmer's guide. Jade version, 2002, vol. 3.
- [4] BELLIFEMINE, Fabio Luigi; CAIRE, Giovanni; GREENWOOD, Dominic. Developing multi-agent systems with JADE. John Wiley & Sons, 2007.
- [5] BELLIFEMINE, Fabio, et al. JADE: A software framework for developing multi-agent applications. Lessons learned. Information and Software Technology, 2008, vol. 50, no 1, p. 10-21..
- [6] BELLIFEMINE, Fabio; POGGI, Agostino; RIMASSA, Giovanni. JADE: a FIPA2000 compliant agent development environment. Proceedings of the fifth international conference on Autonomous agents. ACM, 2001. p. 216-217.
- [7] BERBER, Eren. Complications Of Hypothyroidism. EndocrineWeb, 2016. Web. 20 Feb. 2017.
- [8] CAO, Longbing; GORODETSKY, Vladimir; MITKAS, Pericles A. Agent mining: The synergy of agents and data mining. IEEE Intelligent Systems, 2009, vol. 24, no 3.
- [9] COLUMBUS, Louis. Roundup Of Internet of Things Forecasts And Market Estimates, 2015. Forbes, December, 2015, vol. 27.
- [10] DURAIRAJ, M.; RANJANI, V. Data mining applications in healthcare sector: a study. Int. J. Sci. Technol. Res. IJSTR, 2013, vol. 2, no 10.
- [11] FRANK Eibe , et al. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques," Morgan Kaufmann, Fourth Edition, 2016.
- [12] GAO, Jie; DENZINGER, Jörg; JAMES, Robert C. A cooperative multi-agent data mining model and its application to medical data on diabetes. International Workshop on Autonomous Intelligent Systems: Agents and Data Mining. Springer Berlin Heidelberg, 2005. p. 93-107.
- [13] GAO, Jie; DENZINGER, Jörg; JAMES, Robert C. CoLe: A cooperative data mining approach and its application to early diabetes detection. Data Mining, Fifth IEEE International Conference on. IEEE, 2005. p. 4 pp.
- [14] GORODETSKY, Vladimir; KARSAYEV, Oleg; SAMOILOV, Vladimir. Multi-agent technology for distributed data mining and classification. Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on. IEEE, 2003. p. 438-441.
- [15] HALL, Reginald; SCANLON, Maurice F. Hypothyroidism: Clinical features and complications. Clinics in Endocrinology and metabolism, 1979, vol. 8, no 1, p. 29-38.
- [16] HALL, Mark, et al. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 2009, vol. 11, no 1, p. 10-18.
- [17] JENSEN, Peter B.; JENSEN, Lars J.; BRUNAK, Søren. Mining electronic health records: towards better research applications and clinical care. Nature Reviews Genetics, 2012, vol. 13, no 6, p. 395-405.
- [18] KOH, Hian Chye, et al. Data mining applications in healthcare. Journal of healthcare information management, 2011, vol. 19, no 2, p. 65.
- [19] KUMAR, Sujeet; KUMAR, Utkarsh. Java agent development framework. International Journal Research, 2014, vol. 1, no 9, p. 1022-1025.
- [20] MARKIEWICZ, Marcus Eduardo; DE LUCENA, Carlos JP. Object oriented framework development. Crossroads, 2001, vol. 7, no 4, p. 3-9.
- [21] MINIWATTS Marketing Group. Internet World Stats: Usage and population statistics. Retrieved from Internet World Stats: <http://www.internetworldstats.com/stats.htm>, 2016.
- [22] MITCHELL, Tom M. Machine learning and data mining. Communications of the ACM, 1999, vol. 42, no 11, p. 30-36..
- [23] MITKAS, Pericles A., et al. A framework for constructing multi-agent applications and training intelligent agents. International Workshop on Agent-Oriented Software Engineering. Springer Berlin Heidelberg, 2003. p. 96-109.
- [24] PALANIAPPAN, Sellappan; AWANG, Rafiah. Intelligent heart disease prediction system using data mining techniques. Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on. IEEE, 2008. p. 108-115..
- [25] RAGHUPATHI, Wullianallur. Data mining in healthcare. Healthcare Informatics: Improving Efficiency through Technology, Analytics, and Management, 2016, p. 353-372.
- [26] SZEWCZYK, Paweł. Impact of the Internet of Things on the economy and society. Zeszyty Naukowe. Organizacja i Zarządzanie/Politechnika Śląska, 2016, no 93, p. 461-470.
- [27] UCI KDD archive. Irvine, University of California, Department of Information and Computer Science, <http://kdd.ics.uci.edu>. Last access January 2017.
- [28] WITTEN, Ian H., et al. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.
- [29] WOOLDRIDGE, Michael. An introduction to multiagent systems. John Wiley & Sons, 2009.
- [30] BRIOT, Jean-Pierre; DE NASCIMENTO, Nathalia Moraes; DE LUCENA, Carlos José Pereira. A multi-agent architecture for quantified fruits: Design and experience. En 28th International Conference on Software Engineering & Knowledge Engineering (SEKE/2016). SEKE/Knowledge Systems Institute, PA, USA, 2016. p. 369-374.