Intelligent - Miner: The Conceptual and Architectural Design of a Web Based Data Mining Service

Christos Fidas¹, Manolis Maragoudakis², Nikolaos Avouris¹

¹ Human-Computer Interaction Group, ECE Department
University of Patras
GR-26500 Rio Patras, Greece
fidas AT ece.upatras.gr,
avouris AT upatras.gr

² Department of Information and Communication Systems Eng.
University of the Aegean
mmarag AT aegean.gr

Abstract  The purpose of this paper is to present the conceptual and architectural design of a web enabled service that supports the implementation of data mining techniques. The particular service is addressed mainly to scientists who lack of data mining domain expertise but want to investigate hidden relationships among their experimental or observational data. A novelty of the proposed service relies on the fact that it proposes a common data presentation format which is scalable, robust and extensible, thus allowing scientists from different applications areas to adapt their raw data and use the proposed service. This particular service introduces the most proper data mining technique by investigating several complementary principles like Bayesian networks, neural networks, decision trees, support vector machines and other core algorithms from the field of machine learning and artificial intelligence. The service has been effectively used in a case study which is briefly described.

1 Introduction

Data mining is the principle of sorting through enormous data sets generated by experimental and observational methods and picking out relevant information. Data mining has been described as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data" and "the science of extracting useful information from large data sets or databases", 'as mentioned by Witten and Eibe [1]'.

The reason for the recent attention of business and academic interest in data mining grew as a direct consequence of the availability of large repositories of data especially through the internet. Data flood, i.e. the trend
of storing information into machine-readable format, is the key concept that initiates the need to extract hidden knowledge from it. Data mining is the process of finding patterns and relationships in data. At its core, data mining consists of developing a function (also called as model), which is typically a compact encoding of patterns found using historical data, and applying that model to new, previously unseen data. The application of the extracted model on new data can find certain forms of behaviour (classification and regression, as reported by Michie, Spiegelhalter and Taylor [2] and as mentioned by Salzberg [3]), segment a population (clustering), determine relationships within a population (association), as well as to identify the characteristics that most impact a particular outcome (attribute importance), as described by Hornick et al [4].

Data mining is vastly diverse in origin, in the sense that it borrows elements from a great variety of scientific domains: artificial intelligence, machine learning, statistics, neural computation and visualization. Thus, it is obvious that data mining is an inter-disciplinary concept that requires certain training for scientists which are not directly related to it. Simultaneously, scientists often have access to or own a plethora of available data from which useful knowledge could be extracted, either in a from of patterns or in a predictive manner. Nevertheless, the lack of generic data mining platforms discourages them from applying data mining, as described by Roodyn and Emmerich[5].

Given that building applications that require data mining methodologies is a challenging and complicated task, as stated by Wultrich[6], especially for non data mining experts, nowadays, two main approaches for tackling with this problem exist: a) using technological frameworks that are addressed to software developers in order to integrate data mining techniques on existing software application that are used by industry or scientists and b) the consulting support of data mining experts which in cooperation with application domain experts provide their assistance in order to apply certain data mining techniques on existing data. Both of the aforementioned approaches involve a huge amount of overhead to a user who wants to apply data mining techniques, as reported by Lui et al [7].

During the last years, researchers have attempted to standardize data mining tasks. The first contribution towards this direction was achieved via the introduction of WEKA as mentioned by Witten and Eibe [1] and JDM, as described by Hornick et al [4]. Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a data set or be called from another application. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. Although it initially appeared as an attractive platform, nowadays, due to the training it requires, it is used mostly by data mining experts. The WEKA API is easy to incorporate to existing software; however, since algorithms are developed by various
programmers, the final outcome may vary from other, commercial mainly, implementations. What modern data mining frameworks such as WEKA and JDM do, is that they establish a common programming language and platform for machine learning and software engineers. In the same context relies the JDM (Java Data Mining) standard. This software package benefits application developers (especially those who are dealing with Oracle servers) and requires special knowledge on the field. JDM also contains an XML schema for using it in web-based applications, which is a positive aspect, yet difficult to be adopted by people not belonging to the programmers sector.

Other commercial applications like KXEN®, Peltarion Synapse®, Salford Random Forests® are algorithm-dependent, meaning that they only exploit a certain data mining technique and are difficult to be parameterized. For example, KXEN is specialized in support vector machines, while Synapse is suitable for neural networks. The use of a generic, web-based, easy-to-use platform is not yet supported, so our proposed architecture is novel for the task at hand. As regards to academic solutions, Smart Archive (SA) is a component-based Data Mining application framework, ‘proposed by Laurinen et al [8]’. It is consisted of a conglomeration of data mining tasks, starting from filters and cleaners and extending to classifiers. However, as the authors state “Compromises had to be made as regards to the extent of details on each module and the components and architecture of SA were explained at such a level that people interested in experimenting with the architecture can implement it and adapt it to their applications and tools”. In other words, SA is mostly a proposition about how such a platform should be, rather than an explicit documentation about incorporating data mining dynamics to practical solutions.

The field of providing data mining services to no data mining experts is still rather unexplored, since only a limited number of approaches that describe a framework for addressing this need exist, as reported by Chan et al [9]. The most similar work to this one, although in a different domain, ‘is presented in Hsiung et al [10]’. The study presents a framework called VERTAF for the development of embedded real-time systems. The framework shares some of the motivation of the work ‘as described by Roodyn and Emmerich[5]’, offering components and interfaces for implementing embedded systems. Thus, we argue that there is a need for an integrated methodological and technological framework addressed to non data mining experts for performing data mining tasks.

This paper focuses on the conceptual model, the architecture and the main functionality of a comprehensive web based service, called Intelligent-Miner, which is available to scientist lacking of data mining domain expertise. The added value of the proposed service relies on the fact that it facilitates and supports the implementation of diverse data mining techniques by providing means for usable, straightforward setup of a data mining task, hiding the complicated parameters and having the miner focused only at the quality of the extracted knowledge. Thus, the estab-
lishment of a generic, standardized framework is significant, emphasizing at enabling individuals from different backgrounds and levels of expertise to come to a common goal, which is to discover knowledge from data and identify useful patterns that can be used to predict or classify newly arrived and previously unseen data. Producing a standard architecture in such a context is a challenging, yet fascinating task to accomplish. The advantages of providing a generic data mining framework are numerous, because a typical data mining application has to handle a large number of variables, parameters and transformations. For one thing, the time spent implementing functionality common to most data mining applications can be significantly decreased to the benefit of increased resources for application-specific development. The quality of the application is likely to be better, since the code of the framework has already been tested, and the application-specific code is called through well-defined interfaces.

The rest of this paper is structured as follows. In section 2, we describe the conceptual design of the proposed framework and in Section 3 we discuss the architectural design of the proposed web service. In section 4, a case study is presented, in which the proposed service has been used for applying data mining techniques in a medical application domain and finally, Section 5 provides some concluding remarks.

2 Conceptual and Architectural Design

2.1 Conceptual Design

At the conceptual level, the proposed service is designed to support all stakeholders (researchers, scientists etc.) who are not data mining domain experts but desire to use the framework in order to apply such techniques to their own data. The proposed service supports the users in a conceptual level by providing a common data presentation format for adapting their custom data in a readable and accessible format by the service and describing the different levels of inputs and outputs. On an operational level, users are supported by defining and refining inputs and outputs dynamically, according to the context of use and the applied data mining methodology. As usual, raw data are generally related to one experiment (this includes the nature of the data, its unit, etc.) and the experiment is performed on one structure (this includes the known properties, the research questions trying to answering why, who, how and when). Taking into consideration that each of the individual researchers has its own internal format relating with their data sets, different semantics and conceptual models relating with their research activities, one of the main goals of the proposed service is to achieve an easy adaptation of the custom data sets to a common format supported by the service.

The common data format is based on a data model which provides the ability to structure and organize data from various experiments which ne-
Intelligent - Miner: The Conceptual and Architectural Design of a Web Based Data Mining Service

cessitates providing enough metadata to effectively describe the research. This structuring of the data allows for novel searches to be conducted and to easily find the data sets of interest, since the dissemination of data sets to the broader community ultimately facilitates use of the research results by the users, 'as reported by Abiteboul [11] and Papakonstantinou et al [12]'. Throughout the data set uploading process, the researcher annotates the data following the proposed data model using the Intelligent-Miner interface. The goal is to effectively annotate data, such that users who are not directly participate in the research can still understand the data. Thus they could be able to either inherit data into their own or run different experiments through the data mining engine, which will be based on models build and trained from that borrowed data. The challenge is striking, since a balance between structuring the data, in order to provide insightful queries, and making data sets comprehensible and reusable by the broader domain community, has not been previously studied.

In figure 1, the data model adopted by the Intelligent Miner service is presented. As it can be easily observed, this model design focuses on a meta-data architecture, containing not only the raw data, available for performing classification and other data mining functionalities, but also containing essential information about the nature of data (i.e. characteristics of the research, etc), the research problem that is being confronted (i.e. any prior knowledge provided, related bibliography for assisting other researchers) and the specifications of the data mining algorithms and preprocessing tools.

The higher concept in the hierarchy is the Domain entity which expresses the thematic area in which the data mining tasks are applied. Properties of the Domain entity are its description, some additional information regarding the researcher who is working on this area and some optional references for allowing collaboration between scientists. Each Domain may contain many Research areas. By Research area, we denote any particular data mining procedure one may wants to model, i.e. classification, regression, error analysis, etc. This entity is governed by three significant questions that needed to be answered for a fully-functional and effective modeling.

- a) The first question deals with the problem of data completeness. Complete data means that the feature space is covered by a great number of instances. In case this is unknown the system assumes a middle state.

- b) The second question is how much noise lies within the data. Usually, noise is misleading the data mining algorithms from constructing representative models and needs to be removed. Moreover, noise is an important issue to know, in order to deploy special algorithms that have embedded tolerance in noisy data. Again, in the unfortunate case where someone does not know the noise level contained within a data set, the system can apply certain metrics based on
distance and density and isolate the extreme instances from the rest of the examples.

- c) The third question that is useful to know (thus it is optional for our approach) is the complexity of the specified problem. Although this is a very difficult question to be answered, there are certain types of problems that follow a known distribution, thus helping the data miner at choosing the appropriate algorithm. However, in case one is unable to specify the complexity of the problem, our framework performs a series of correlation and chi-square tests at the time of data input specification. If some attributes are found dependent among each other, the complexity levels fall by a varying factor that is from 10 to 25 percent. The refinement of a research topic to a specific data mining experiment, is a useful functionality that allows scientists to complete a series of mining procedures within the notion of a single research category.

Another attribute of the Research entity is related with prior knowledge. Domain experts have often empirical knowledge about the relations of certain input features with the class to be predicted. This information may be difficult to be expressed in a format the machine could interpret, especially when such experts lack of data mining technical knowledge. However, since such information is very helpful, our framework incorporates two techniques for inserting prior knowledge. A graphical tool where domain experts can draw arcs between nodes expressing the input features.
and nodes representing the class (i.e. an arc from an input variable node to the class node means that this feature directly affects the class) as well as draw arcs from an input node to another input node, or a cost table, containing weights (as continuous numbers or percentages) adjusting the degree of influence of features to the class.

The refinement (Run entity) of a research topic to a specific data mining experiment, is a useful functionality that allows scientists to complete a series of mining procedures within the notion of a single research category. For example, in a research related to classification, one may want to perform a more specific experiment involving only a subset of the available data. In the financial domain, suppose that the research question is the forecasting of the closing value of the NY stock market index, based on the values of certain stock quotes. A refinement of this to a specific experiment (also noted as run) could be the prediction of the value within a margin of \( \pm 2 \) percent only. The concept of the experiment is characterized by the specification of the subtask and the inclusion of some constraints with regards to computational (execution) time and tolerable error rate of the final outcome. The latter is very useful, since the data mining engine could run a plethora of trials before deciding which methods is most appropriate for the given run. Each refinement is directly connected to the notion of raw data Training Data that will be used for training. Additional information include the number of features that will be used as input, the data type and the size of each feature. The Training Data is the core of the engine since this history information is the basic source of knowledge, from where a predictive model will be derived. As often this set is not ready for data mining, a series of filtering tasks could be applied to purify the training set. Such filters are discretizers, wavelets, selectors of subsets, value transformations, etc.

The Training Data entity is related with the Evaluation Data entity. In this concept, the user define the outputs of the system (one or more) and specifies the type of output format. Currently, we are supporting three main actions: (a) the error analysis report, the classification (at run-time) module and the visualization of the outcomes.

The Data Mining Model entity encapsulates different data mining methods, categorized according to their theoretical framework:

- function oriented like Multi-layer perceptron neural network, Radial Basis Function neural network, Linear Regression and Support Vector Machines

- statistics oriented like Naive Bayes and Tree-Augmented Naive Bayesian Networks

- trees oriented like C4.5 and Random Forests

- instance-based oriented like K-Nearest Neighbor
We consider the above selection of algorithms as the most representative since all of the above have presented state-of-the-art performance for numerous classification tasks of different nature. For example, in the medical domain, Bayesian networks and Decision Trees have found to be very promising, 'as reported by Maragoudakis et al [13]'. In classification of e-mails as spam or not, Naive Bayes and Support Vector Machines have proven more robust, 'as described by Joachims [14]'. However, in case a user in not able to decide, we have incorporated an autonomous agent which takes user constraints and nature of data into account and performs a series of experiments with every algorithm by using a small portion of training data and decides which methodology is most suitable for the task, thus performing a full training using it.

Since it is assumed that the Intelligent Miner service will be used by several users who wants to apply data mining techniques in several application domains we consider as well privacy and dissemination policy issues. In such cases, a user can decide to share the data with other users activating in the same domain (upon agreeing with the service policy), or obtain data from others as well. The entity dealing with this feature is the Dissemination concept, which encapsulates the ownership and accessibility levels given to other users of the community in relation with different parts of the experiment, such as the input data, the output model and the predictive model itself. Input data, denotes that upon cleaning and filtering, the data set could be available to other experts through the service. Output model describes situations where error analysis reports can be published to others. Model, means that classification scheme itself (built from the input data) could be used for future classification of new, unseen examples, provided by other users.

### 2.2 Architectural Design

As mentioned above, the added value of this service relies on the fact that it allows the collaboration and exchange of knowledge among various users within a domain community, in terms of sharing raw data or commenting on others’ results. A positive aspect of this approach is that since the system itself has a large repository of data for a plethora of domains stored in a central repository, it is able to provide more accurate and representative classification results on a predefined application domain. Taking into account the above implications, relating to the heterogeneous raw data formats, and the obvious need to provide scientists with access to other data within a given application domain of interest, a significant novelty of the proposed service is to introduce a novel concept of standardisation, exchange and publication of measured data models.

For reaching the aforementioned target two approaches can be identified: a) having all the data stored in a unique place which is commonly defined as a uniform database approach or b) having the data stored in distributed physical locations. The former approach is the most rational
approach when starting a services from scratch and data entry is though to be done by the services users it self. However, using this approach there is a significant loss of control by storing the data in a central database. It should be underpinned that there is a strong sense of ownership for the data produced and furthermore there is continuous need for duplicating the work by uploading in the central database. Using the latter approach, which is adopted by the proposed service, the primitive or raw data are maintained at the location of each domain expert, yet they can be exchanged by using a common data format. This particular approach allows data owners to develop, maintain and feed their data as usual. In parallel, by using a common data format converter which is based on a standardized XML and RDF scheme, it is possible to automatically transfer related data to the previously provided service, 'as it has been reported as well by Beckett [15]'.

At the architectural level, the proposed service is designed to provide an extensible and robust integration platform allowing custom data sets to be easily adapted.

From a technical point of view, the service consists of the following entities:

- An Application server, which is responsible for the collection and administration of relevant data related to the implementation of a data mining study using databases, applications etc.

- A number of components for the study preparation, execution and visualization of the results.

- A data mining engine which examines the most appropriate data mining technique for a given task among bayesian and neural networks or decision trees etc. (see figure 3 as well)

- A monitoring service which enables the setting up, execution and analysis of several studies, by supporting the definition of detailed

---

**Figure 2: The Intelligent Miner Architecture**
tasks like raw data sets adaptation, preparation of what if scenarios, visualization of data.

The architectural design of the data mining engine is shown in Figure 3. The main parts include an on-line and an off-line section. The on-line section deals with new, unobserved incoming data as well as with specification from the user’s point of view with regards to performance, time and computational cost. The off-line section contains archived data as well as models trained from them and used for the prediction of values in the on-line section.

The phase of constructing the predictive models incorporates a verification process, in which evaluation of all data mining algorithms take place and whichever performs more robustly and effectively on the archived data is selected as final. The exact data mining process as depicted in the above figure is as follows: The arrows denote data streams and control messages, the rectangles denote active data mining modules and the other graphical objects denote different format of data and repositories.

The user provides the system with data, in a machine readable format (XML or in a table). This is denoted as ”Gathered Data”. It is a conceptual data abstraction for storing, manipulating, and accessing multidimensional data sets. The basic component of this module is a software programming interface that supports a device-independent view of the provided data model. The user is insulated from the actual physical file format for reasons of conceptual simplicity, device independence, and future expandability. CDF files created on any given platform can be transported to any other platform onto which CDF is ported and used with any CDF tools or layered applications.

The system asks the user to set which parts of information will be considered as input (also called as input features) and which as output (in the literature, it is usually referred to as the class variable). Note that users can set more than one feature to be the class. This step is denoted by ”Data/Problem Definition”. Upon completion of this stage, the system prompts the user to set the degree of complexity of the problem (how difficult he/she believes that is to find a representative predictive model), how confident that the provided data is error/noise free and how representative (as regards to the feature space) they are. Provided that such information is known by the user and given to the system, the ”pre-processing” phase takes place. The responsibility of the data pre-processor is to perform elementary operations, such as data cleansing and integration, to ensure that only high quality data is fed into the subsequent components.

The feature extractor filters the data to extract information that is not directly measurable but may improve the performance of the model. During pre-processing, a series of filtering processes is executed, including normalization, removal of noisy data (i.e. data examples which deviate from the standard data distribution), tests on feature dependencies and feature selection. In the case that users cannot answer to the data com-
Intelligent - Miner: The Conceptual and Architectural Design of a Web Based Data Mining Service

Figure 3: The Intelligent Miner Engine Architecture

plexity prompts, the system will attempt a more modest pre-processing phase where only features with high degree of correlation will be removed from the set.

The cleaned data are forwarded to the "classifier". The purpose of this component is to build a function that takes new, previously unobserved data and predict the value of the class(es). In order to perform such a task, a training phase has to be done. The classifier communicates with the "model repository", found in the off-line module. In case it is empty or not suited for the new task, the model builder is initiated. This component is instantiated by the "user constraints", in the sense that the user defines the time of execution and the performance threshold. The model builder then starts an iterative approach various algorithms of different origins are selected from the "algorithm repository". The "performance evaluator" estimates the precision and recall measurements from each training/test run using the 10-fold cross validation method, in which the data set is separated into train and test set in a 9/1 ratio, and repeats the procedure 10 times using different examples each time.

The most accurate model is preserved to the "model repository" for future classification tasks. The classifier uses this model in combination with any available "prior knowledge" (optional) that can be forwarded to the model builder and help constructing a more representative classification model. A final component is the "results" section, which presents to the user the final predicted outcome, either as single values or vector of values in textual or graphical form.

The advantages of the proposed service is demonstrated in the next section by presenting a case study which deals with classifying patients of
a hospital with certain symptoms into pre-defined categories of illnesses.

3 A case study of patient classification to pre-defined categories of illnesses according to certain symptoms

In this section we present a case study in which a prototype of the aforementioned service has been used by a group of scientists for modeling medical knowledge. In particular, the goal has been towards building a web-based classification system for the treatment of nosocomial and community acquired pneumonia, without requiring technical knowledge about data mining procedures. The term pneumonia, either nosocomial or community refers to the inflammation of the lung parenchyma, due to bacterial, viral, fungal, parasitic causative microorganisms, that is accompanied with clinical and radiological features of one or more lung densities, ‘as reported by Jameson [16]’. In the past, several attempts have been made as regards to medical informatics and computational intelligence using several data mining techniques, ‘as reported by Sacha et al [17]’ and Buchanan et al [18].

Scientists from this domain, have to take the following issues into consideration:

– The pathophysiologic characteristics of a patient,
– Maintenance of a priority list for handling pneumonia cases concerning their clinical picture and indications
– Maintenance of a list of chemotherapeutic-antimicrobial agents, along with their contra-indications
– Choice of the appropriate antibiotic in a percentage scales of priority, regarding the appropriateness of matching between the microorganisms, the pathophysiologic condition of the patient and the drug.

Following we present an example case of history of the required information: Patient personal data and medical history:

– The patient is male, 26 years old,
– He had heart transplantation 18 months ago,
– He has already done influenza virus vaccine,
– He has been under therapy using cortisol, cyclosporine and cellosept.

Clinical picture:

– Bilateral pneumonia of interstitial type,
Intelligent - Miner: The Conceptual and Architectural Design of a Web Based Data Mining Service

- WBC: 7,000/mm³, PMNs: 65,000/µL, Ly: 28,000/µL.

Diagnosis:
- PCR(+) for CMV (Cytomegalovirus):

Suggested therapy:
- Gancyclovir (DHPG): 5mg/kg q12h i.v. (for 12-21 days) + 5mg/kg q24h i.v. (maintenance)

Taking into account the application domain, the Intelligent Miner service should be capable of transforming the available history data sets into the proposed data model and classify new patients according to predefined sets of suggested therapy. To perform such task, the relational concepts depicted in our data model have to fill their corresponding information fields, as it can be seen in figure 4. The exact process is described below and it is fully supported by the web service at its whole. As a consequence, the goal is to create a classification service, where data from future patients that describe personal information and medical history could be used as input and the service would predict the type of therapy (if needed) the patient should take (not the exact dosage of each substance, only the name, e.g. "Gancyclovir").

In order to achieve that, the user has to specify the meta data model by providing a short profile with personal information, the domain application in which this attempt takes place, meaning that the system should know which the main thematic category is, in order to group similar tasks and enable future collaborations between researchers. Optionally, some bibliographic reference could be introduced, enriching the scientific background of the task.

The case study was performed by a medical expert on the domain of pneumonia treatment, while data was available from 300 patients of the "Evangelismos" hospital (located in Athens, Greece), and they were gathered under the purposes of a research project called DIKTIS, 'as reported by Maragoudakis et al [19]. Data was transformed to a relational database, in Microsoft® Excel® format and they contained physiological and medical data. As it has been mentioned previously, the goal of pneumonia experts is to create a classification service, where data from future patients that describe personal information and medical history could be used as input and the service would predict the type of chemotherapy (if needed) the patient should take (not the exact dosage of each substance, only the name, e.g. "Gancyclovir").

Therefore, in our example, the domain is "Pneumonia" and it is the parent conceptual category of the case study. Since in each domain, several research tasks may be carried out, the doctor will have to continue by filling up fields for the research concept.

In this case study the data were incomplete, but there were no estimation regarding the degree of incompleteness. Thus, by using an easy-to-use graphical tool such as a slider, the user can provide an initial estimate
about how good the examples of his data (i.e. his patients) describe the notion of pneumonia treatment. In case this is unknown the system assumes a middle state.

Since noise is almost anywhere in real data sets, the user may also quantify the level of noise his data believes they contain. It is a very informative step since the data mining engine can use this feedback to perform certain tasks such as removal of noisy examples by estimating distances or densities among data instances. Provided that such knowledge was much more technical to tackle with, the data mining engine was responsible for eliminating noisy examples, since that was dictated by the expert. The third important question one should specify deals with the complexity of the final goal. Since it was also not known to the user, the appropriate slot labeled "machine specified" instructed the pre-processing modules to perform a series of correlation and chi-square tests at the time of data input specification.

The refinement of a research topic to a specific data mining experiment is a useful functionality that allows scientists to complete a series of mining procedures within the notion of a single research category. For example, in our case, the research question is still the classification of a patient according to the proposed chemotherapy, but in a more specific set-up, a user could use only the male data examples and without considering the attribute (or feature) of age. In our conceptual framework, this is denoted as a 'run'. Before proceeding to analyzing the configuration of each run, the notion of prior knowledge has to be specified. Prior knowledge is any
information that a domain expert could insert on the system, knowing that it would be helpful in finding a more accurate model which describes the task. Usually, it involves clarifications on the influence of certain input features to the final outcome, or the class. In our case, such knowledge is provided graphically as regards to the correlation of input and output features (via a graphical tool). The user knows that age is not very important to the task, thus he adjusts the weight to a negative value, again by using a slider. Furthermore, he also knows that the existence of an influenza virus vaccine is almost always affecting the suggestion of chemotherapy. Thus, he can draw an arc from this feature to the class.

In each run, users have to specify the sub-task which is to be executed along with some constraints, regarding the available time the data mining engine has for training and the error percentage that the user considers as tolerable. As a last adjustment, the user has to provide feedback about the data used for training and evaluation, as well as the type of output he considers as useful. This deals with our conceptual models of training and evaluation data. The training data is actually the knowledge source from which the engine is called to perform the data mining question. This historical data have to be given to the engine, along with information on the number, the type and the size of features that will be actually used. Since data sets are not always in the correct format, one may involve a series of filters, such as discretizers, value transformers, wavelets, selector of subsets, etc., to further improve the quality of the set. In our case, if the user lacks of any particular knowledge on data pro-processing, a special wizards uses the information given in the research step and attempts a basic pool of filters.

Upon specification of the training set, the evaluation set is also configured, meaning that the expert selects which feature is going to be used as the prediction class and what is the desirable type of output. In our case, suggested therapy is the only class. There are three different ways to obtain the outcome. Either in a form of an error analysis report, or in a single module that will accept new, unseen examples and will predict the value of the class, or by using visualization tools. The first case, deals with the construction of a confusion matrix, where actual and predicted data are tabulated in a table, along with the standard metric used in information retrieval, i.e. precision and recall. This task assumes that the training set will be split into training and test in a 9:1 ratio and this process will run for 10 times, using different samples for training and test. In the bibliography, this is known as the 10-fold, cross-validation approach. This is useful in case the user evaluates the task in general, obtaining a basic idea on how data mining is performing. However, a more useful type of outcome is the predictor, which is actually a model where the user can insert data of a new patient and the model would return the value of the class i.e. the type of chemotherapy needed. Another type of output, supported by our framework is the visualization of the problem, which includes the ROC curve (receiver operating characteristic) and the
visualization of the classifier errors. The ROC curve portrays the performance of the various data mining algorithms used for classification. Our specific user had selected all three types of outcomes. Moreover, in our study, the user could not decide on the data mining algorithm prior to any execution, thus in the corresponding conceptual model all methods were filled with label "any", meaning that an autonomous agent which takes user constraints and nature of data into account should perform a series of experiments with every algorithm by using a small portion of training data and decide which methodology is most suitable for the task, thus performing a full training using it. Finally, the dissemination concept was specified to model only, meaning that the service should hide the historical and evaluation data from other users and create a web interface where others could just fill the feature values of new patients and obtain the final prediction on the "suggested therapy" class.

4 Conclusions

In this paper we presented a novel, component-based application service framework called "Intelligent-Miner". Intelligent-Miner provides functionalities to researchers and scientists who are not data mining experts but are interested to find hidden relationships among their observational or experimental data. The particular service adopts a distributed management of data sets, thus allowing its users to manage their data as usual, trying not to interfere with the actual processes of gathering and storing data sets. For achieving inter-operability, robustness, extendability and reusability of data and extracted knowledge, the Intelligent-Miner service proposes a data model which is abstract enough in order to be able to encapsulate in a common format the raw data of diverse research areas. The proposed data model incorporates raw data information, meta data research description and functional attributes of data mining assessments covering in this way the complementary views of a data mining task.

Furthermore, the proposed service provides and implements diverse data mining methodologies and components for predicting which algorithm is best suited for the user-specified data and needs, as they are initially expressed. The service supports a straightforward approach, without having to build everything from scratch and without worrying about data preparation, preprocessing, cleaning and filtering. A novel extension of the proposed service, yet not evaluated with real users, relies on the fact that it allows users to access, review and comment in several levels the assessments of other researchers in situations where owners publish their research either in terms of input and output data or in the form of functional classifications models.

Nevertheless, only an early prototype of the proposed system has been applied to a certain domain which has been briefly described. The whole service needs to be evaluated with long term evaluation studies which
Intelligent - Miner: The Conceptual and Architectural Design of a Web Based Data Mining Service

will provide us with improvements and new features. Taking into consideration that there is a continuous trend to provide service oriented frameworks through the web, services like the proposed one are of general value for data mining tasks.


Intelligent - Miner: The Conceptual and Architectural Design of a Web Based Data Mining Service


