# An Interoperable and Scalable Web-based System for Classifier Sharing and Fusion

Grigorios Tsoumakas * Ioannis Vlahavas

*Department of Informatics,*
*Aristotle University of Thessaloniki,*
*54124 Thessaloniki, Greece*
*tel: +30 2310998418, fax: +30 2310998419*

**Abstract**

This paper describes CSF/DC, a Web-based system for classifier sharing and fusion. CSF/DC enables the sharing of classification models, by allowing the upload and download of such models expressed in the industry standard PMML language on the system's online classifier repository. CSF/DC also leverages the individual knowledge shared by such (potentially heterogeneous) classification models and offers quality decision support to any user with an Internet connection through a guided procedure. However, some organizations or individuals might want to share the predictive capabilities of their classification models without compromising their internal structure. This is accommodated by CSF/DC through the use of Web services. Specifically, CSF/DC allows the participation of Classifier Web services in the decision fusion process, by offering the necessary online mechanisms for the registration and invocation of such Web services developed and installed at remote sites.

*Key words:* Web Services, Classifier Fusion, Distributed Data Mining, Classification Models

* Corresponding author
*Email address:* greg@csd.auth.gr (Grigorios Tsoumakas).

## 1 Introduction

During the last decade, researchers and practitioners in the areas of Machine Learning, Data Mining and Knowledge Discovery from Databases have managed to develop robust methodologies and systems for the intelligent analysis of data and their transformation to new, interesting and useful models of knowledge.

Recently, the World Wide Web is increasingly being used as the platform for the deployment of such data mining models. For example, there are currently several data mining companies that offer customized business intelligence solutions as Web-based applications [1] . In addition, there are a lot of proposed architectures and systems for Web-based data mining that make the functionality of the resulting mining models accessible via the Web (1; 2; 3). Furthermore, *Web services*, the latest technology in distributed computing have recently been applied to data mining, leading to the design of system architectures that employ mining models as Web services on the internet (4; 5; 6).

The motivation towards this direction was the increased audience that could access the mining models though the information sharing power of the Internet. Other advantages are reduced overheads of software installations and updates and increased accessibility for the mobile user.

The availability of all these data mining models online brings up the opportunity of their integration in order to improve the quality of knowledge. For distributed classification models specifically, a variety of methods have been proposed for their combination, (7; 8; 9; 10).

This paper extends the idea of deploying the functionality of data mining models online by proposing: i) the sharing of classification models themselves on the Web and ii) the online fusion of distributed classification models.

This is accomplished through CSF/DC, a scalable and interoperable Web-based system for classifier sharing and fusion. The core of CSF/DC is an online repository for classification models, where users can upload and download classification models. The use of the industry standard PMML language for classifier representation ensures the interoperability of the system and models. In addition, CSF/DC offers the added value of using all applicable classifiers that are stored in the repository for the classification of a new example that is entered online through a guided procedure. The results of the classification models can then be combined through a variety of standard classifier fusion methods.

[1] http://www.kdnuggets.com/solutions/asp.html

However, some organizations or individuals might want to share the predictive capabilities of their classification models without compromising their internal structure. This is accommodated by CSF/DC through the use of Web services. Specifically, CSF/DC allows the participation of Classifier Web services in the decision fusion process, by offering the necessary online mechanisms for the registration and invocation of such Web services developed and installed at remote sites.

The benefits of CSF/DC system in comparison to other approaches are: i) The functionality of data mining models reaches a wider group of users due to the capability of centrally searching and finding a useful model, which can in addition be downloaded and then easily tested, examined and compared to other locally available classification models. In addition, communities can be informally created online by mining experts that work on the same domain. ii) The models themselves can be refined and enhanced as their strong and weak points will be uncovered through their exposure to multiple domain and mining experts. Domain experts will test the models on new data and can provide direct feedback to the mining experts that developed the models. Alternatively, model popularity as measured from the online interaction of users with the models (downloading and fusion), can give indirect feedback regarding the quality of models. iii) CSF/DC leverages the individual knowledge shared by many (potentially heterogeneous) classification models and offers quality decision support to any user with an Internet connection. The performance of the system is superior to approaches where classifiers are located in distributed sites, as it does not involve any communication costs. Coordination is a minor issue, as a newer version of a classification model can invalidate the old one through a few clicks. iv) The use of Web services technology adds substantially to the interoperability and scalability of the system. The XML-based standards of Web services combined with the widespread use of the Internet's http protocol are important technical advantages of CSF/DC, especially in comparison with systems that are based on outdated distributed computing paradigms such as CORBA and Java RMI.

The rest of this paper is structured as follows: Section 2 provides background information on the Web services framework and Section 3 on classifier fusion methodologies. Section 4 describes the CSF/DC system, presenting its architecture, functionality and implementation details. Finally, Section 5 concludes with an overview and future work directions.

## 2 Web Services

A Web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML[2]. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by internet protocols.

The use of the Web services paradigm is expanding rapidly to provide a systematic and extensible framework for application-to-application (A2A) interaction, built on top of existing Web protocols and based on open XML standards. Web services aim to simplify the process of distributed computing by defining a standardized mechanism to describe, locate, and communicate with online software systems. Essentially, each application becomes an accessible Web service component that is described using open standards. The basic architecture of Web services includes technologies capable of:

- *Exchanging messages.*
- *Describing Web services.*
- *Publishing and discovering Web service descriptions.*

### *2.1 Exchanging messages*

The standard protocol for communication among Web services is the *Simple Object Access Protocol*[3] (SOAP). SOAP is a simple and lightweight XML-based mechanism for creating structured data packages that can be exchanged between network applications. SOAP consists of four fundamental components: an envelope that defines a framework for describing message structure, a set of encoding rules for expressing instances of application-defined data types, a convention for representing remote procedure calls and responses, and a set of rules for using SOAP with HTTP. SOAP can be used with a variety of network protocols, such as HTTP, SMTP, FTP, RMI/IIOP, or a proprietary messaging protocol.

SOAP is currently the de facto standard for XML messaging for a number of reasons. First, SOAP is relatively simple, defining a thin layer that builds on top of existing network technologies such as HTTP that are already broadly implemented. Second, SOAP is flexible and extensible in that rather than trying to solve all of the various issues developers may face when constructing Web services, it provides an extensible, composable framework that allows

[2] www.w3c.org/TR/ws-arch/
[3] www.w3.org/TR/SOAP/

solutions to be incrementally applied as needed. Thirdly, SOAP is based on XML. Finally, SOAP enjoys broad industry and developer community support.

### *2.2 Describing Web services*

The standard language for formally describing Web services is *Web Services Description Language*[4] (WSDL). WSDL , is an XML document format for describing Web services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented (RPC) messages. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints may be combined into services. WSDL is sufficiently extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate. A complete WSDL definition of a service comprises a service interface definition and a service implementation definition, as depicted in Figure 1.

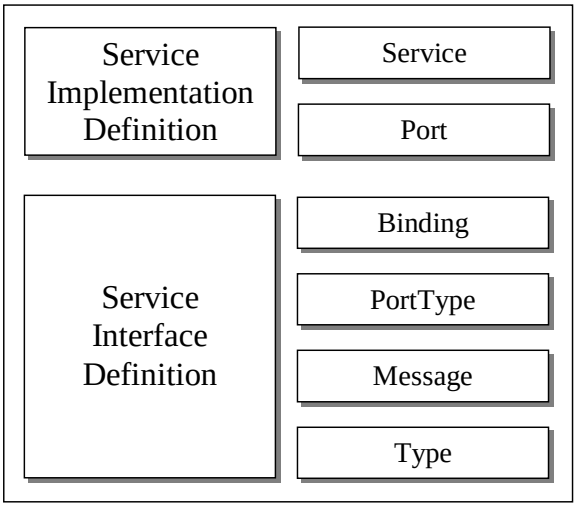


Fig. 1. WSDL service implementation and interface definitions

A service interface definition is an abstract or reusable service definition that may be instantiated and referenced by multiple service implementation definitions. A service interface definition can be thought of as an IDL (Interface Definition Language), Java interface, or Web service type. This allows common industry standard service types to be defined and implemented by multiple service implementers.

In WSDL, the service interface contains elements that comprise the reusable portion of the service description: *binding*, *portType*, *message* and *type* elements. In the *portType* element, the operations of the Web service are defined.

[4] www.w3c.org/TR/wsdl12/

The operations define what XML messages can appear in the input, output and fault data flows. The *message* element specifies which XML data types constitute various parts of a message. The *message* element is used to define the abstract content of messages that comprise an operation. The use of complex data types within the *message* is described in the *types* element. The *binding* element describes the protocol, data format, security and other attributes for a particular service interface (portType).

The service implementation definition describes how a particular service interface is implemented by a given service provider. It also describes its location so that a requester can interact with it. In WSDL, a Web service is modelled as a *service* element. A *service* element contains a collection of *port* elements. A *port* associates an endpoint (e.g. a network address location) with a *binding* element from a service interface definition.

### *2.3 Publishing and discovering Web service descriptions*

While there are some established standards for Web service description and communication, the publishing and discovery of Web services can be implemented with a range of solutions. Any action that makes a WSDL document available to a requestor, at any stage of the service requestor's life-cycle, qualifies as service publication. In the same way, any mechanism that allows the service requestor to gain access to the service description and make it available to the application at runtime qualifies as service discovery.

The simplest case of publishing a Web service is a direct publish. This means that the service provider sends the service description directly to the service requestor. This can be accomplished using an email attachment, an FTP site, or even a CDROM distribution. Slightly more dynamic publication uses Web Services Inspection Language (WSIL) [5]. WSIL defines a simple HTTP GET mechanism to retrieve Web services descriptions from a given URL. Another means of publishing service descriptions available to Web services is through a Universal Description, Discovery and Integration (UDDI) [6] registry. There are several types of UDDI registries that may be used depending on the scope of the domain of Web services published to it. When publishing a Web service description to a UDDI registry, complete business context and well though out taxonomies are essential if the service is to be found by its potential consumers.

[5] www.ibm.com/developerworks/webservices/library/ws-wsilover
[6] uddi.org/pubs/uddi-v3.00-published-20020719.htm

## 3 Classifier Fusion

### *3.1 Classification and Classifier Types*

Classification is one of the most common machine learning and data mining tasks (11). It deals with the problem of identifying interesting regularities between a number of independent variables and a target or dependent categorical variable in a given data set. For example, given a set of training instances $(x_{i1}, x_{i2}, ..., x_{ik}, y_i), i = 1..N$, the task is to compute a classifier, or model, or concept that approximates an unknown function y=f(x) that correctly labels any instance drawn from the same source as the training set.

There exist many ways to represent a classification model and many more algorithms to generate it. Typical classifier learning approaches include concept learning, neural networks, decision trees, rule learning, Bayesian learning and instance-based learning. All these approaches construct models that share the common ability to classify previously unknown instances of a domain based on instances of the same domain that were used for their training.

The output of a classifier can be i) the label of a class, ii) rankings for all the classes and iii) measures of uncertainty such as belief, confidence, probability, possibility, plausibility or other for each class. Consider for example, a domain for predicting tomorrow's weather with three possible classes: *sunny*, *windy*, *rainy*. The corresponding output for the three types of classifiers could be: i) sunny, ii) 1 - sunny, 2 - windy, 3 - rainy and iii) 0.8 - sunny, 0.5 - windy, 0.1 - rainy. Classifiers that output labels are commonly called hard classifiers, while those that output measures of uncertainty are called distribution/soft classifiers.

Another distinction among classifiers is whether they are homogeneous or heterogeneous. There are two forms of classifier heterogeneity. According to the first, two classifiers are considered homogeneous if they are created using the same learning algorithm. For example a naive Bayes classifier and a decision list are heterogeneous classifiers, while two neural networks are homogeneous. Another form of heterogeneity is based on the schema of the training data of the two classifiers. For example, two decision trees that both predict tomorrow's weather, but one is based on temperature and wind speed, while the other on atmospheric pressure and humidity are considered heterogeneous. In this paper, the term *heterogeneity* will be used with the latter meaning.

### *3.2 Fusion Methods*

Classifier Fusion has been a very active field of research in the recent years. It was used for improving the classification accuracy of pattern recognition systems, as single classification learning algorithms were approaching their limits. It was also used as a method for scaling up data mining to very large databases, through combining classifiers trained in parallel from different parts of the database. Finally, it was used for learning from geographically distributed databases, where bandwidth or privacy constraints forbid the gathering of data in a single place, through the fusion of locally learned classification models.

There are two general groups of Classifier Fusion methods. The first one encompasses methods that combine the outputs of classifiers, while the second deals with the structure of a multiple classifier system. We will focus on the former group, as CSF/DC supports some of its methods.

Methods that fuse classifier outputs can be further categorized based on two properties. The first is the classifier output type on which they can operate, and the second is the need of training data for the fusion process. According to these, Table 1 presents the main methods. CSF/DC currently supports the simple method of Majority Voting and the Sum, Product, Min, Max and Median rules.

Table 1
Classifier fusion methods

| **Method** | **Output** | **Re-Training** |
|---|---|---|
| Majority Voting (12) | Label | No |
| Knowledge-Behaviour Space (13) | Label | Yes |
| Borda Count | Rankings | No |
| The Highest Rank | Rankings | Yes |
| Logistic Regression | Rankings | Yes |
| Intersection of Neighborhoods | Rankings | Yes |
| Union of Neighborhoods | Rankings | Yes |
| Sum, Product, Min, Max, Median rules (14) | Distribution | No |
| Stacked Generalization (15) | Distribution | Yes |
| Dempster-Shaffer Combination (16) | Distribution | Yes |
| Fuzzy Templates (17) | Distribution | Yes |
| Fuzzy Integrals (18) | Distribution | Yes |

The fact that CSF/DC is used for the fusion of heterogeneous classifiers forbids the use of most methods that require re-training, as they usually depend on re-training data with a common schema across classifiers.

## 4 The CSF/DC System

This section presents the architecture of the CSF/DC system, its main functionality with respect to classifier management and classifier fusion and its implementation details.

### *4.1 Architecture*

The architecture of CSF/DC comprises the following main components as depicted in Figure 2:

- The Web Server hosts the Web pages that constitute the user-interface of CSF/DC . It accepts input from Web browsers and forwards it to the Application Server for further processing. It is also the place where the classification models are stored.
- The Application Server hosts the application logic of CSF/DC . It handles all dynamic processes that deal with the upload and download of classifiers, the registration of classifier web services and the invocation and fusion of all available classifiers. It accepts input from the Web Server and creates the dynamic content of the Web pages.
- The Meta-data Database is used for storing and retrieving important meta-data of the local and remote classification models.

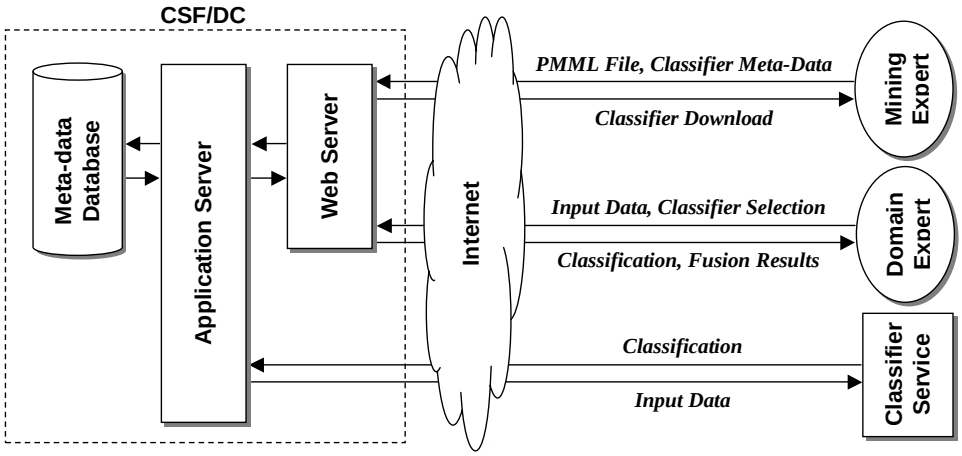


Fig. 2. The architecture of CSF/DC

Mining experts access the system to upload and download classifiers from the system or to register classifier web services. Domain experts access the system to enter data about a specific classification domain. They then obtain the results of the applicable classifiers that exist in the system. They can also select a subset of these models in order to get back the result of their combination through standard classifier fusion methods.

### *4.2 Classifier Management*

Classifier management deals with the upload and download of classifiers and with the registration of classifier web services. These are processes that are expected to be carried out by data mining experts or advanced data mining software users, as it will usually involve either importing/exporting models into/from data mining software or providing the meta-data of a data mining model.

#### *4.2.1 Upload and Download of Classification Models*

CSF/DC supports classification models expressed in the open industry standard PMML [7] (Predictive Modelling Markup Language) format. PMML is an XML-based language which provides a way for applications to define statistical and data mining models. The use of PMML ensures the interoperability of CSF/DC, as it can be used for sharing and fusing classifiers that were created with any PMML compliant application.

A classifier formatted in PMML can be added to CSF/DC through the system's corresponding Web page with a file upload form. The PMML document is first uploaded to the Web Server and subsequently sent to the Application Server where it is parsed using the DOM [8] (Document Object Model) interface in order to extract important meta-data about the classifier. These include the type (neural network, tree, Bayesian, etc.) and description of the classifier, along with the names, types and acceptable categorical values of the input and output variables. These are stored in the Meta-data Database of the system. At the end, the complete PMML document is stored on the server with a filename equal to *classifier_id.xml*, where *classifier_id* is the primary key of the classifier table in the meta-data database.

Classifiers stored in CSF/DC can be downloaded through the system's corresponding Web page that displays the recorded meta-data of all available classification models along with a hyperlink to the PMML file on the server.

#### *4.2.2 Registration of Classifier Web Services*

CSF/DC supports classifier web services written in any programming language, as long as they comply with the interface description of the *classifier* Web service that the system expects, presented in Figure 3.

[7] www.dmg.org
[8] www.w3c.org/DOM

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://classifierws"
                xmlns="http://schemas.xmlsoap.org/wsdl/"
                xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
                xmlns:impl="http://classifierws-impl"
                xmlns:intf="http://classifierws"
                xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <types>
    <schema targetNamespace="http://classifierws"
            xmlns="http://www.w3.org/2001/XMLSchema">
      <complexType name="ArrayOf_SOAP-ENC_string">
        <complexContent>
          <restriction base="SOAP-ENC:Array">
            <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="xsd:string[]"/>
          </restriction>
        </complexContent>
      </complexType>
      <element name="ArrayOf_SOAP-ENC_string" nillable="true"
               type="intf:ArrayOf_SOAP-ENC_string"/>
    </schema>
  </types>
  <wsdl:message name="classifyResponse">
    <wsdl:part name="return" type="SOAP-ENC:string"/>
  </wsdl:message>
  <wsdl:message name="classifyRequest">
    <wsdl:part name="names" type="intf:ArrayOf_SOAP-ENC_string"/>
    <wsdl:part name="values" type="intf:ArrayOf_SOAP-ENC_string"/>
  </wsdl:message>
  <wsdl:portType name="classifier">
    <wsdl:operation name="classify" parameterOrder="names values">
      <wsdl:input message="intf:classifyRequest"/>
      <wsdl:output message="intf:classifyResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ws_classifierSoapBinding" type="intf:classifier">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="classify">
    <wsdlsoap:operation soapAction="" style="rpc"/>
      <wsdl:input>
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                       namespace="http://classifierws" use="encoded"/>
      </wsdl:input>
      <wsdl:output>
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                       namespace="http://classifierws" use="encoded"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

Fig. 3. The WSDL interface description of the Classifier Web Service

The interface indicates that the classifier Web service must have a *classify* function that expect as input two arrays of strings, *names* and *values*, and returns a string. The two arrays correspond to the names of the input attributes and the values that the user enters. The predicted class is the returned string.

In order for the system to know the address of the remote classifier web service and the meta-data of the remote classifier, CSF/DC offers a web page where the user can enter all this information. Specifically this page has input boxes for the URL, type and description of the Web Service and a file upload form, where the user must upload a part of the PMML file of the classification model containing just the *data dictionary* and *mining schema* elements. This file is parsed similarly to the classifier upload page using the DOM interface in order to extract the rest of the meta-data about the classifier.

The WSDL description of the classifier does not suffice for automating the above process. If we put the information on the input/output attributes of the classifier in the WSDL description, then the code of the Web service and its client will become overly complex. There exists a richer web service description language, called DAML-S [9] that is part of the semantic web vision and allows the inclusion of semantic content. However, currently CSF/DC does not support it, as it is still a standard under development.

### *4.3 Classifier Fusion*

Classifier fusion deals with the process of using the system to classify new data. This encompasses selecting an output variable, entering data for a new example and selecting the participating classifiers for the voting process. These tasks are domain specific and should be best performed by domain experts. Example users could be doctors in the case of medical classifiers or financial analysts in the case of credit risk assessment classifiers.

The first step in using the system for classifier fusion is domain selection. Users must here specify the output (target) variable that they want to predict. The selection is not arbitrary but they are given a list of all unique available output variables from which they can select only one. This list is retrieved through an SQL query to the Meta-data Database.

The second step involves entering data about the new example to be classified. The system presents all variables that could be used to predict the target variable that was selected in the previous step. The necessary information is again retrieved though an SQL query to the Meta-data Database. Users must here enter the values of these attributes with respect to the new example.

In the third step, the system presents the meta-data and predictions of each applicable classifiers. CSF/DC accomplishes this task by importing the PMML document of each locally available applicable classifier and instantiating it to a classifier object. This object is then used for predicting the output variable given the data of the new example that were entered in the previous step. It also sends the entered data to every remote applicable classifier Web service and collects the classification results.

The last step allows the selection of all or some of the applicable classification models for participating in the fusion process. By default all models are selected, but users can deselect any of the models to exclude them from the fusion. Users can also select here one of the available fusion methods. The

[9] http://www.daml.org/services/

fusion results Web page, is the final page of the fusion process, where the combined classification result is presented.

### *4.4 Implementation Details*

A prototype of CSF/DC has been developed and is available through the following address: http://lpis.csd.auth.gr/systems/csfdc. The current release of the system supports only the upload and registration of decision tree models and the fusion method of majority voting (12). The following subsections describe the technologies used for the development of the prototype and aspects of its user interface.

#### *4.4.1 Technologies*

The CSF/DC implementation makes use of open and free technological solutions: The Apache Tomcat server [10] was used as both the Web and Application Server. MySQL [11] was used as the relational database. Java Server Pages were used to implement CSF/DC 's functionality on Tomcat. The Xelopes data mining library [12] was used for instantiating classifier objects from their XML representation and using them for the classification of new examples.

#### *4.4.2 User Interface*

We here provide screen-shots of the Web pages that constitute the user interface of CSF/DC, through a specific case study that uses one of the *Wisconsin Breast Cancer Databases* from the UCI Machine Learning repository (19). It concerns data about benign and malignant breast cancer cases from patients of the University of Wisconsin Hospitals, Madison.

The scenario involves as users the mining experts and doctors of the hospitals. The mining experts develop a classification model from the data and provide it as a Web service. They also register this service with CSF/DC . Figure 4 depicts the Web page for the registration of the classifier. Upon pressing submit users are directed to a Web page that displays the registration result, either success or failure.

Doctors directed to the main page of the system can click the fuse link, which leads them to the domain selection page. There, they select the *breast-cancer*

[10] jakarta.apache.org/tomcat/
[11] www.mysql.com
[12] www.prudsys.com/Produkte/Algorithmen/Xelopes/

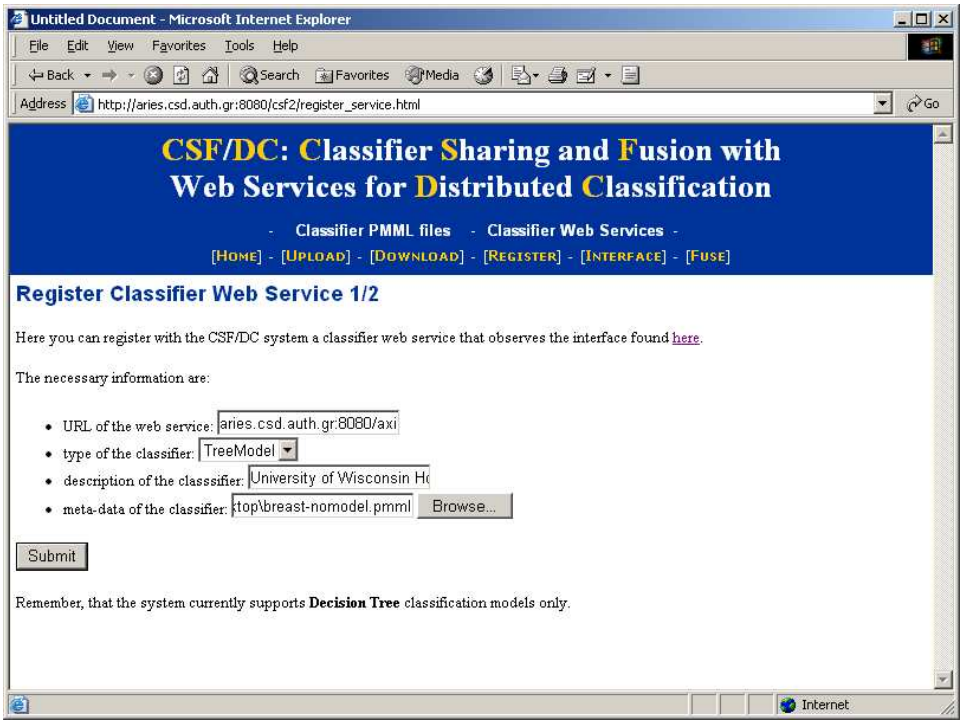


Fig. 4. Classifier Web service registration Web page

output attribute and click submit, to be directed to the data entry page, which is depicted in Figure 5.

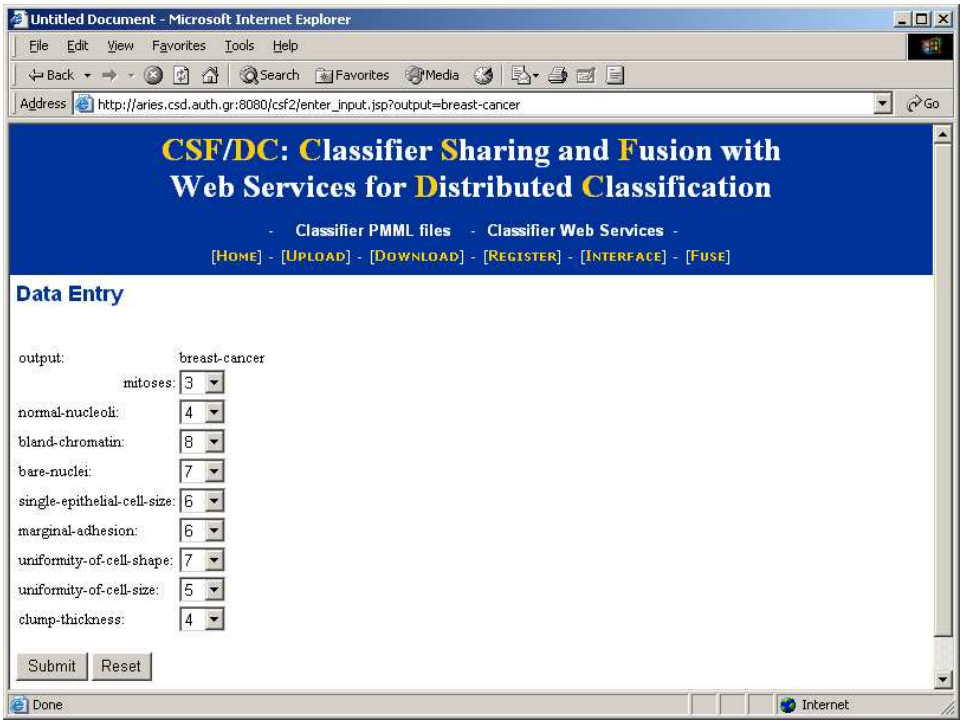


Fig. 5. Data entry Web page

Doctors enter here the data for a new patient and upon pressing submit they are directed to the classifier results page that is depicted in Figure 6. There the doctors can see the results of the hospital's classifier.

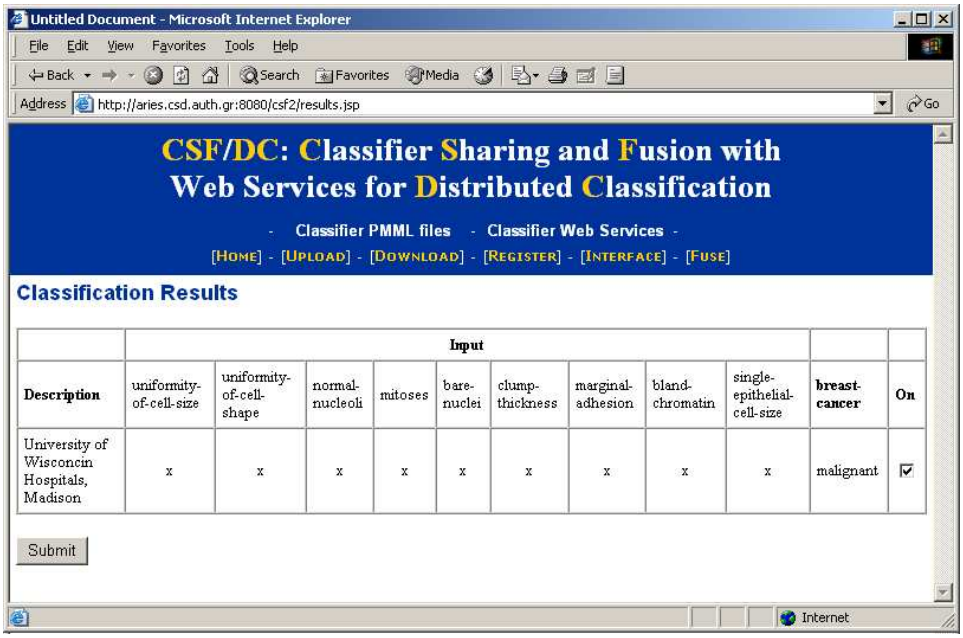


Fig. 6. Classification results Web page

## 5 Conclusions and Future Work

This paper has presented the CSF/DC system for sharing and fusing classifiers on the Web. Amongst the main benefits of the system are wide accessibility of the knowledge that stems from the application of classification algorithms, through the use of the PMML industry standard for classifier representation and an online repository that simplifies the discovery and exploitation of classification models. The system can lead to the evolution of user communities that are involved with the modelling of the same domain. In addition CSF/DC offers a platform that can lead to the enhancement and refinement of classification models themselves through their exposure to many users of various expertise.

The use of Web services technology has upgraded the interoperability and scalability of the system, as organizations or individuals will be able to easily share the predictive power of their already existing classification web services. No installation of software is required and no maintenance costs for the distributed classifiers. In addition the system and the distributed classifiers are always accessible from the mobile user, thanks to the Internet-based communication. Web services is the upcoming standard of tomorrow in distributed Web-based computing.

The prototype implementation of CSF/DC proves the feasibility of the proposed system. However, there are a number of small extensions that can enhance the value that the system offers to its users. For example, providing attribute explanations and statistics for numeric attributes in the data entry Web page, recording the popularity of usage for the models, or keeping statistics with respect to model performance. We intend to extend the system with these features and other suggestions that will come up through feedback of users/experts on the field.

An important future development of the system is user management. Storing user data and handling user sessions, will allow users to manage their own classification models only, and make their models available either to specific users or to the whole public. This will allow the use of the system by communities of mining experts for specific applications, and it will promote it to a safe repository and classification system for mobile users. In addition, it will allow the development of added value services such as providing recommendations to users based on their past interactions with the system or those of similar users.

An important research problem that we also aim to explore in the future, is that of syntactic and semantic inconsistencies amongst the distributed classification models that are added in the system. This is not an issue for classifiers of a closed user group, but when it comes to taking advantage of open distributed heterogeneous classifiers then it becomes a serious problem. A solution could be to have the classifiers checked across domain-specific ontologies that would be embedded in the system and reject the addition of those classifiers that do not comply with those ontologies.

## References


[1] J. Chattratichat, J. Darlington, Y. Guo, S. hedvall, M. Khler, J. S. A. Saleem, D. Yang, Deploying enterprise data mining on the Internet, in: Proceedings of the PAKDD, 1998.

[2] S. Krishnaswamy, A. B. Zaslavsky, S. W. Loke, An architecture to support distributed data mining services in e-commerce environments, in: WECWIS 2000, 2000, pp. 239–246.

[3] S.-T. Li, A web-aware interoperable data mining system, Expert Systems with Applications 22 (2002) 135–146.

[4] S. Sarawagi, S. H. Nagaralu, Data mining models as services on the internet, SIGKDD Explorations 2 (1) (2000) 24–28.

[5] G. Tsoumakas, N. Bassiliades, I. Vlahavas, A knowledge-based web information system for the fusion of distributed classifiers, in: D. Taniar, W. Rahayu (Eds.), Web Information Systems, 2004, pp. 271–308.

[6] O. B. Kwon, Meta web service: building web-based open decision sup-

port system based on web services, Expert Systems with Applications 24 (2003) 375–389.

[7] A. Prodromidis, P. Chan, S. Stolfo, Meta-learning in distributed data mining systems: Issues and approaches, in: H. Kargupta, P. Chan (Eds.), Advances in Distributed and Parallel Knowledge Discovery, MIT Press, 2000.

[8] G. Tsoumakas, I. Vlahavas, Effective stacking of distributed classifiers, in: Proceedings of the 15th European Conference on Artificial Intelligence, 2002, pp. 340–344.

[9] L. Hall, N. V. Chawla, K. W. Bowyer, Decision tree learning on very large data sets, in: Proceedings of the IEEE SMC Conference, San Diego, California, 1998, pp. 2579–2584.

[10] H. Kargupta, B. Park, D. Hershbereger, E. Johnson, Collective data mining: A new perspective toward distributed data mining, in: H. Kargupta, P. Chan (Eds.), Advances in Distributed Data Mining, AAAI/MIT Press, 1999, pp. 133–184.

[11] L. Saitta, Machine learning: A technological roadmap, Tech. rep., University of Amsterdam (2000).

[12] L. Lam, C. Y. Suen, Optimal combinations of pattern classifiers, Pattern Recognition Letters 16 (1995) 945–954.

[13] Y. S. Huang, C. Y. Suen, A method for combining multiple experts for the recognition of unconstrained handwritten numericals, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (1995) 90–93.

[14] J. Kittler, M. Hatef, R. P. W. Duin, J. Matas, On combining classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (3) (1998) 226–238.

[15] D. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.

[16] G. Rogova, Combining the results of several neural network classifiers, Neural Networks 7 (1994) 777–781.

[17] L. Kuncheva, R. K. Kounchev, R. Zlatev, Aggregation of multiple classification decisions by fuzzy templates, in: Proceedings of the Third European Congress on Intelligent Technologies and Soft Computing EUFIT'95, Aachen, Germany, 1995, pp. 1470–1474.

[18] H. Tahani, J. Keller, Information fusion in computer vision using the fuzzy integral, IEEE Transaction on Systems, Man and Cybernetics 20 (1990) 733–741.

[19] C. L. Blake, C. J. Merz, UCI repository of machine learning databases, http://www.ics.uci.edu/∼mlearn/MLRepository.html (1998).