Journal of Information Systems Engineering and Management
2025, 10(31s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Interactive Web Engineering System for Syndrome Classification in Social Networks

Alan Carlos Curiel-Consuelos[1], Sergio Cerón-Figueroa[3], Emmanuel Alejandro Ruiz-Fonseca[2], Rolando Gómez-Padilla[2], Cornelio Yáñez-Márquez[3*]

[1]*Escuela Superior de Ingeniería Mecánica y Eléctrica–Unidad Zacatenco, Instituto Politécnico Nacional, México*
[2]*Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, México*
[2]*Centro de Investigación en Computación, Instituto Politécnico Nacional, México*
**Corresponding author: cyanez@cic.ipn.mx*

ARTICLE INFO

Received: 26 Dec 2024

Revised: 14 Feb 2025

Accepted: 22 Feb 2025

ABSTRACT

Nowadays social networking has permeated through every human activity including, of course, web engineering. As social networks develop, a wide variety of social networking applications appear, which attract a lot of new users. In the current paper, an interactive web engineering system for syndrome classification is developed. The system is implemented in the context of social networks, using pattern recognition algorithms based on unconventional computing. The classification algorithm used is Morphological Autoassociative Max Memories, which belongs to the associative approach of Pattern Recognition. Throughout the experimental phase, the proposed system is applied to help diagnose several syndromes. The core proposal of this article is a web system that features an innovative characteristic: it interacts with social media users, enabling potential patients to utilize this novel system as a valuable tool for remote diagnosis of syndromes. It is crucial to emphasize, as demonstrated in the experimental results section, that the proposed interactive system achieves a high true positive rate compared to other state-of-the-art models. Experimental results confirm that the proposed web engineering system can be a valuable tool for knowledge discovery and social networking between different profiles of people involved in syndrome classification.

Keywords: Web Engineering, Syndrome, Pattern Classification, Associative Memories, Social Networks.

INTRODUCTION

The dawning of the third millennium has been accompanied in the developed societies, by the fast development of high impact technologies. It is also remarkable how quickly such technologies are adopted by wide sectors of modern society. Thus, social networking has permeated through every human activity including, of course, web engineering and cloud computing. As social networks develop, a wide range of social networking applications appear, which attract a lot of new users and developers [1].

The new generations of humans exhibit a curious phenomenon: children and youngsters fluently use the jargon which accompanies new technological developments, often teaching the concepts and operation associated with new devices to their parents and grandparents. It is commonplace to hear nowadays kids talk about programming languages, social networks, the Internet, websites, hackers, HTML, web applications, as well as a wide variety of bleeding edge technological terms. Such kinds usually evidence deep knowledge regarding the similarities and differences between different information and communication technologies, as well.

In this context, it is a well-known fact that web applications are essentially different form conventional web pages, in that the former generate their content dynamically; that is, the HTML code is generated on the fly and usually there is no physical version of the accessed resource [2].

On the other hand, cloud computing refers to distributing and consuming computing resources through an Internet connection in order to provide applications and services as commodities; as a result, data center virtualization has become increasingly popular, minimizing deployment times of production applications and thus costs.

In the current paper, an interactive web engineering system for syndrome classification is developed. The system is implemented in the context of social networks, using pattern recognition algorithms based on unconventional computing. Additionally, the proposal takes advantage of one of the most significant characteristics of web applications: they run on the web browser, but their business logic (where most of the data processing takes place) is hosted on the server, allowing the browser to act a light-weight terminal or thin client. In the particular case of the work presented in this paper, the pattern classification task happens on the cloud (server side), leaving the tasks of data bank uploading and system configuration to the client, in a social network environment.

The classification algorithm used in this paper is Morphological Autoassociative Max Memories, which belongs to the associative approach of Pattern Recognition. Throughout the experimental phase, the proposed system is applied to help diagnose several syndromes.

The web system is presented as a tool, in a social networks environment, that allows remote patients diagnosis with some kind of syndrome, providing a high rate of true positives in symptoms verification. Experimental results confirm that the proposed web engineering system can be a valuable tool for knowledge discovery and social networking between different profiles of people involved in syndrome classification.

Two key topics central to the primary proposal of this research are outlined, establishing both the conceptual and operational foundation for the study: Web Engineering and Social Networks. Following this, the core objective of the work is introduced: syndrome classification using web engineering within a social network environment. Subsequently, relevant scientific literature related to the proposed system is reviewed. The associative approach is highlighted as one of the viable paradigms for pattern recognition in classification tasks, and morphological associative max memories are described in detail. The main contribution of this article is then presented, alongside experimental results demonstrating the system's performance. Finally, the concluding section summarizes findings, discusses implications, and outlines directions for future work.

RELATED CONCEPTS AND WORKS

Although the field of web engineering —as well as the name coined for it— is relatively recent, it has already gained the interest of researchers, academics, developers, and clients, becoming significant and current from both theoretical and practical standpoints. Such importance becomes quite evident when looking at the emerging journals devoted to it, as well as all the currently increasing activities related to web engineering, be it in research, project developments, publications, international conferences and workshops, academic offerings by universities around the globe (either in bachelor programs and as individual courses), or practical applications in the software development industry. Clearly, the field is headed towards further advancement by means of research, education, and practice [3].

As discussed in [4], web engineering is a multidisciplinary field whose theoretical and practical bases lie in both computer science and information management systems, although it encompasses such diverse areas as: software engineering, information systems analysis and design, requirements engineering, project management, web programming, hypermedia engineering, human-computer interaction, artificial intelligence, and data mining. Based on such a wide variety of concepts and tools, web engineering has the goal of supporting the processes concerned in the lifecycle of web applications and web projects. To achieve this goal, web engineering makes use of different methodologies, processes, models, techniques, and technologies to deal with such activities as: web projects management, web systems development and maintenance, quality assessment, web resources management, and web intelligence.

Another way to explain web engineering is presented in [5], where it is described as successfully developing, deploying, and maintaining high quality web systems, by means of scientific, engineering, and management principles, as well as systematic approaches. In this sense, the authors of [6] point out that developing web engineering applications means to use sound methodologies, systematic techniques, quality assurance, best practices and tools, as well as rigorous, disciplined, and repeatable processes.

The importance of a holistic and proactive approach to developing successful applications for the web is increasingly underscored by the growing amount of applications that are either migrated or already born in the web environment, as discussed in [7]. Even more, such applications play an increasingly relevant role in many aspects of our lives, whether they pertain to government, education, healthcare, business, or common, everyday activities. Since web engineering is specifically directed at successfully developing, deploying, and maintaining large and complex web systems, the growing emphasis we place on the performance, correctness, and availability of such web applications will bring greater significance to the development and maintenance process employed, attracting more and more attention to this field.

Given this growing weight given by the public to the success of web projects in terms of performance, correctness, and availability, the need for developers to make their systems ore competitive is also stimulated. This in turns drives the adoption of emerging information gathering techniques relevant to defining web application requirements, since such methods are quite useful to projects whose competitive edge can substantially improve by some degree of innovation [8].

This practice-driven characteristic of web engineering induces a high relevance for the application of empirical research methods to it. As is discussed in [9], web engineering encompasses organizational issues, project management, aspects, as well as human behavior, besides the technical solutions. This makes empirical methods critically important for web engineering, since they enable the inclusion of human behavior aspects in the research done, which in turns can greatly improve the richness of the information used for decision making in the web engineering process when combined and coupled with other methods. Examples of such empirical methods include: controlled experiments, case studies, surveys, and post-mortem analyses.

Also, the environment and conditions intrinsic to web applications confer them with some particularly exacting requirements, such as the operational environment, their development approach, and a faster pace of development and deployment cycles, which are usually more demanding than traditional software development. Besides the latter, web systems have to account for and satisfy the needs of many different stakeholders besides its (potentially very diverse) user base, such as: maintenance personnel, the organizations that benefit from the system, and the parties which fund the development, deployment, and maintenance of the system [42]. These needs may in turn pose some additional challenges for web systems design and development, particularly in terms of responsiveness comprehension, security and integrity, evolution, growth, maintainability, testability, mobility, usability, interface design, and navigation.

Several research teams on web engineering, working from either an academic or an industrial point of view, have created a broad spectrum of architectures, specific development platforms, protocols, drivers, database engines, design patterns, services, libraries, and many other products that support web engineering practitioners to develop increasingly improved products every day, for the benefit of all users around the globe [3].

The proposed system operates in a social networking environment. Thus, a brief discussion of social networks, their characteristics, their relevance for the contemporary world, and the kinds of bleeding edge technologies underpinning their design and operation is included here.

A social network is a community of members who reciprocally interact with each other, thus increasing the cohesion among members and the boundaries with respect to outsiders. As part of these interactions, members may provide each other with such resources as: information, feedback, news, advice, or job opportunities, to name a few. When these social networks are implemented over computer networks, they acquire some interesting characteristics: the members may not be restricted to a specific location, nor interact at the same times.

It is becoming increasingly evident that social networking is an important part of our lives —both online and offline— showing diverse effects on us, which range from destructive to constructive behaviors of its users. As such, social networks are part of the trends that the Internet presents towards ubiquity. These aspects underpin the importance of taking advantage of all the information stored in these social networks, such as the composition of a person social circle, the way people communicate with their peers, and the degree of reciprocity between friends. Data mining, analysis, and visualization techniques are some of the tools that allow researchers to tap into the troves of information and relevant data that social networks have become.

One convenient way to model and study social networks is by means of graphs, where people are represented by nodes, and their mutual connections are indicated by edges linking their respective nodes. Based on such a definition of a social network, it can be measured and characterized by concepts such as ties, density, centrality, cliques and other relevant features. Using graph theory, the influence one person has on the whole network may be estimated by her connectivity: the amount of connected nodes and the number of paths they form throughout the graph. Using this tool, the impact an individual has on the whole network and its assessment has diverse applications; for instance, data mining can lead to where cliques are forming, measuring reputations and monitoring social events. Yet, since people exhibit a tendency to interact more closely and more frequently with people with similar interests or opinions [7], the former observations may be done at different levels. In this sense, the social capital value inherent to social networks manifests itself in how common interests help foster closer contact between social networks members with a higher affinity.

The advancement of World Wide Web-related technologies has greatly facilitated and fostered the development and growth of social networks, attracting the attention of researchers to this field. Their efforts of measurement and data collection in such environments are geared towards increasing the scientific study and understanding of the relationships among individuals immersed in social networks [7].

An example of such effort to formalize the study of interactions in a social network is the methodology presented in [10] to detect bridging nodes in a directed and weighted social network and their properties. Such bridging nodes, or bridges, are nodes in a social network that connect peripheral nodes and peripheral groups with the rest of the network. As part of this methodology, first the regular cliques, peripheral nodes and peripheral cliques from a network are extracted and then the bridging nodes identified. Then, the characteristic features of all nodes (such as social position and degree of nodes) are computed; finally, the correlation between the centrality and degree of the nodes, on one hand, and whether a given node is a bridge, on the other, is found.

As a result of applying this methodology, two types of bridging nodes can be distinguished: those that connect peripheral nodes with regular cliques on one hand, and those that connect peripheral cliques with the regular cliques on the other. Also, a correlation was found between the social position in the network and whether a node is a bridge or not, making social position a good measure to detect bridging nodes in a social network.

There is a relevant factor that has a clear influence in social networks dynamics, including social position and bridging nodes: trust. In fact, trust is one of the most fundamental factors in human interaction and communication. As such, trust becomes a crucial element for virtual space societies as more and more human activities migrate to the virtual world. Now, even though trust evaluation is a common place process in many every day activities (developing and intrinsic easy of evaluation for most people), there are no simple transformations of this typical human activity to the virtual world. The great differences between the virtual societies and the real world ones, in particular the level of anonymity and dynamism of virtual societies, have greatly influenced and changed previous trust relations. This situation evidences the need for now models and tools to describe and evaluate trust in virtual societies, such as social network over the web. In this regard, the current direction adopted by the software and information systems development community is related to the paradigm of service orientation.

Trust is therefore of utmost importance for our proposal, given that it involves the classification of syndromes in social networks. This represents a very serious responsibility, since the proposed task is related to one of the most precious gifts of human beings: their health.

In the context of syndrome classification based on web engineering in a social networking environment, it is important to note that the term syndrome originates from the Ancient Greek σύνδρομο (syndromo), meaning "concurrence of symptoms." Today, a syndrome is defined as the set of characteristic features that collectively identify a medical condition.

Throughout human history, different techniques have been developed for diagnosing illnesses, taking as basis the clinic history of the patient. Sometimes these techniques have shown poor performances, offering false negatives that lead in some cases to loss of lives [11]. The latter has driven the rising interest of the scientific community to improve and automate syndrome diagnostics and verification tools, in order make such methods more reliable and precise.

Classification is one of the tasks assigned to Pattern Recognition. As such, it is a powerful tool for the classification of syndromes, allowing the expert to confirm and verify the patient symptoms, which are associated to a specific illness. Pattern classification techniques can be automated by means of computer systems, which operate via an initial training phase —where the system is fit to known types of patterns— and later act on a set of syndromes whose types are unknown [2].

Pattern recognition also involves other tasks, such as pattern recalling, regression, clustering, or recommendation systems, to name a few. However, classification is strongly associated to pattern recognition in a large portion of scientific literature, and our work is related to pattern classification, when the patterns operated by our system represent syndromes.

The different models and algorithms of Pattern Classification belong to one of several approaches, such as the neuronal approach, the statistical-probabilistic approach, the fuzzy approach, the decision trees based approach, among others.

Several works have applied pattern classification methods to syndrome classification. For instance, several authors propose the use of neural networks with bispectral characteristics (QPC) for the identification of patients with Obstructive sleep apnea syndrome (OSAS), which is a condition present when the superior respiratory tract is repeatedly obstructed during sleeping. Also, an application of the Multilayer Perceptron (MLP) is proposed in [11] for the classification of Complex regional pain syndrome (CRPS), which is a condition that causes severe pain in an extremity (or part thereof), and may in turn cause severe deterioration of physical performance.

Similarly, Multiple Logistic Regression based on clinical factors, such as the homeostatic model assessment of Insulin resistance (HOMA-IR), is applied to predict a possible Metabolic syndrome (diabetes, hypertension, dyslipidemia, or arteriosclerosis).

Also, Bayesian networks are used to analyze risk factors for nasopharyngeal carcinoma, which a kind of cancer occurring in the superior portion of the pharynx. On the same classification model, in [12] an evolutive form of ordering is proposed for the Bayesian network features, which is then applied to the prediction of some metabolic syndromes. A related method which has been applied to detecting the Chronic fatigue syndrome (CFS) is the naïve Bayes classifier, with the assumption of each feature of the problem having independent probabilities distributions.

The authors of the current paper has developed during the course of more than a decade a new paradigm for pattern classification: the Associative Approach. The most representative and relevant models of this approach are the morphological associative memories.

It is precisely the latter classification model, working in one of its specific modes of operation, the one selected for the system proposed in this paper: the classification algorithm applied in our proposal is the Morphological Autoassociative Max Memories, which belongs to the associative approach to Pattern Recognition. Morphological associative memories use the dilation and erosion morphological operations in the learning phase, and the maximum and minimum usual operations in the recalling phase [13]. Below appear a description of the concepts and algorithms used by this model to operate.

Definition 1: Maximum product. Let D be a matrix of dimensions $m \times p$, and H be a matrix of dimensions $p \times n$; then the maximum product between D and H is denoted by $C = D \nabla H$, and is defined as follows.

$$C_{ij} = \bigvee_{k=1}^{p} (d_{ik} + h_{kj})$$

Definition 2: Minimum product. Let D be a matrix of dimensions $m \times p$, and H be a matrix of dimensions $p \times n$; then the maximum product between D and H is denoted by $C = D \Delta H$, and is defined as follows.

$$C_{ij} = \bigwedge_{k=1}^{p} (d_{ik} + h_{kj})$$

Based on the former definitions for the maximum and minimum products the operation of this model follows the algorithm depicted below, divided in two distinct phases: a learning phase (in which the associative memory is built), and a recalling phase (where the memory is operated by presenting it with a potentially unknown pattern).

Notice that the basic operations of this model are quite similar to the definitions for the fundamental morphological operations dilation and erosion, hence the name of morphological associative memories.

It is also worthy of mention that morphological associative memories may be of two kinds, while also having two modes of operation. On one hand, they may be autoassociative or heteroassociative, depending on whether the input pattern is equal to the output pattern for every association, or not, respectively. On the other hand, these memories can be of either the max or min kind, which determines the specific algorithm for both learning and classification phases to be used, as well as the resulting properties. However and since this work focuses particularly on the Morphological Autoassociative Max Memories, only this algorithm is presented below.

Algorithm for the Morphological Autoassociative Max Memories

Learning phase

1. Compute matrix $x^{\mu}\Delta(-x^{\mu})^{t}$, where $(-x^{\mu})^{t} = \begin{pmatrix} -x_1^{\mu} & -x_2^{\mu} & \cdots & -x_n^{\mu} \end{pmatrix}$ and $\mu = 1,2,\ldots,p$.
2. Apply the maximum operator to the p matrices previously computed

$$M = \bigvee_{\mu=1}^{p} [x^{\mu}\Delta(-x^{\mu})^{t}]$$

Recalling phase

1. Operate the associative memory **M** with an input pattern x^{ω} to obtain the recalled pattern

$$x = M\Delta x^{\omega}$$

Thus, the i-th component of the recalled pattern is

$$x_i = \bigwedge_{j=1}^{n} \left(m_{ij} + x_j^{\omega}\right)$$

MATERIALS AND METHODS

This section is the most relevant of the present work given that here is discussed the main proposal of the paper. First, the specific products of Web Engineering used to develop the proposed system are detailed; then, the architecture and data flow are presented. Finally, the design of the proposal is included.

In order to develop an interactive system able to maintain an active communication between the user and the server, the proposed system makes use of the Model-View-Controller (MVC) architecture [14] over the J2EE development platform [15, 16]. This allows the distribution of different technologies from the latter standard throughout the development of the tiered web application.

Given the flexibility of the MVC pattern, it is quite practical to implement it over a cloud computing architecture using an Infrastructure as a Service (IaaS). Iaas is a concept associated to cloud computing which involves consuming computing resources (usually virtualized) [17] and mounting over them the services required for application execution [18, 19].

One of the most well-known providers of cloud computing in recent years is Amazon Web Services (AWS), both for computing and storage services (aws.amazon.com). The costs of using such services are based on running time, used storage capacity, characteristics of the hardware platform to be used, and network traffic generated by the application.

In this context, Amazon offers services to consume Iaas resources [12], among which are Elastic Cloud 2 (EC2) [20], Relational Database Service (RDS) [21], and Storage 3 (S3) [22], having each service a specific use, depending on which tier belongs each service.

Model. This tier represents the entities that enable data persistence. In the case of the proposed system, patterns are mapped to the database through Hibernate and their use is facilitated by the driver acting as an Object Relation Mapping (ORM) [23].

The point of having data persistence lies in giving the pattern classification algorithm an efficient access to said data, as well as having convenient ordering operations. In our system, based on the Amazon cloud computing architecture capabilities, and through their RDS service, the MySQL database manager was selected for this task [24].

Of course, this is achieved by means of a service layer implemented inside the model. The Data Access Object (DAO) design pattern provides an abstraction of data query and update operations, regardless of the ORM framework used [25,26], giving way to an easier migration procedure to any other framework, such as the Java Persistence API (JPA) [27].

View. This tier basically represents the user interface of the application, whose components allow the user interactions in an as accessible manner as possible. In this regard, RichFaces by JBoss [28] is web interfaces library compatible with Asynchronous Javascript And XML (AJAX) [29], which implements a set of components that easy the development of the view layer.

RichFaces allow, among other things, to synchronize the algorithm execution and to render on the client only the HTML code portions showing the classification results and performance obtained. The components used for uploading the databanks link directly to the driver, easing the tasks of reading and writing on the S3 Amazon service.

Controller. This tier is usually devised to host most of the business logic of the application. In this sense, JavaServer Faces (JSF) [30] is a framework fully based on MVC, thus enabling appropriate abstractions of the actions taken by the user on the views, in order to run the processes that require access to the data stored by the model, as well as the algorithms stored by the Amazon EC2 architecture.

In the current instance of the classification system, the pattern recognition algorithms (Morphological Autoassociative Max Memories) reside inside each of the layers. However, these algorithms do not interact directly with the view; for that we have a JSF component known as ManagedBean. This class function is to capture the events generated by the view components —specifically when the user uploads a data bank or starts the classification— and provide a method to run the algorithms.

Additionally, the JSF ManagedBean class allows keeping the operation data in memory during the HTTP session, if necessary. This is relevant because it opens the possibility of setting up the system and running the classification in an independent manner.

Figure 1 shows the general architecture of the web application mounted on AWS, making use of the EC2 services for the virtualization of the JBoss application server, RDS for persistence of the MySQL patterns thru the DAO layer, and the Amazon S3 service for massive storage of the data Banks used for training and classification. Also, notice that HTTP communication is done through the AJAX controller implemented by RichFaces to the JSF-based business layer on the server, where the pattern classification algorithm resides.

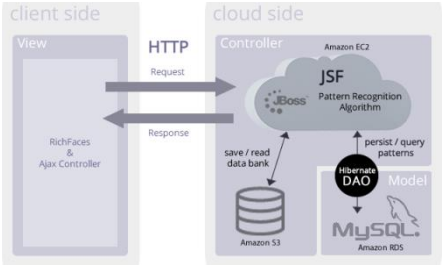

Figure 1. General architecture of the proposed system.

The process begins when the client uploads a training data bank to the system by means of the fileupload component in RichFaces, then the client sends the data bank to the server thru AJAX, and the file is stored by the Amazon S3 service in order to later retrieve it for future classifications. Once this file uploading has been done, some parameters of the classifier may be set, such as: the algorithm and validation technique to be used, as well as whether the classification will be done based on another data bank, whose instance classes are unknown.

The procedure to attach the test data bank file is similar to that for the training data bank. Once the execution is run, the system parses the data in CSV format to the model entities, which are then persisted in the Amazon RDS database. The classification algorithm accesses the entities instead of the data bank, thus making the validation method the module with the greatest speed up gained by this characteristic. This is due to having the random ordering in the stratification phase done on the MySQL database engine, saving time for the application server processing.

It is noteworthy that the AJAX-based actions are run asynchronously [31]; yet, it is necessary to somehow block the execution of the view to avoid inconsistency in the results shown to the user. This requires a synchronous communication between browser and server. Figure 2 shows the timing diagram followed by communication when the client calls for running the pattern classification task. The flow begins when the classification action is run asynchronously, by means of the XmlHTTPRequest object (XHR) [32]. Then, the view must be blocked (at least partially) by a superposition of some element, and the pooling technique is used to verify each 100ms whether the classification results are ready. When the results are ready, they are sent by the server using a JSON encapsulation [33] to minimize the size of the transmitted data [61]. Finally, the view is unblocked and a part of the HTML code is updated with the results.

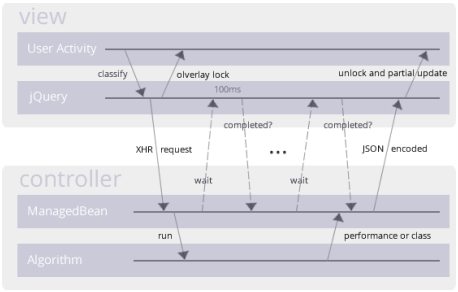

Figure 2. Timing diagram for synchronization of the pattern classification response.

As mentioned before, most of the business logic of the system is hosted on the controller tier, as shown in figure 3. The objects which interact during the execution of the classification algorithms, as well as which of these interact with the other two tiers of the MVC architecture, can be seen in this class diagram.

The view tier is managed by the JSF ManagedBean interface, thus all components of the view are linked to said interface. On the other hand, the DAO interface acts as a bridge towards the model tier, transparently obtaining the patterns which were persisted on the database.

In the end, the classification algorithm just sends a response to the AlgorithmController class, which in turn sends the results to the client since it extends the class in charge of this task. Notice also the S3Manager class, involved in reading and writing files on the Amazon S3 service; and the HibernateDao class, which is indirectly connected to the MySQL instance run on the Amazon RDS service.

Such indirect connection is due to the HibernateDao class having implemented the access methods to the patterns, but having the Hibernate library classes handle the actual communication to the database engine.

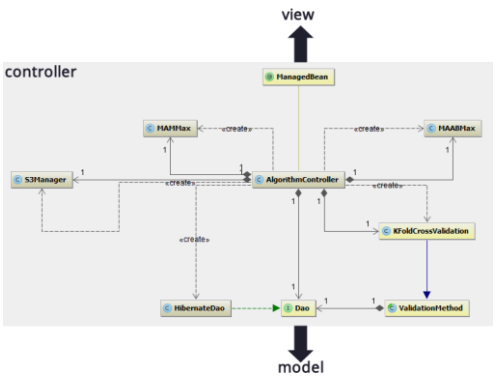

Figure 3. Simplified class diagram of the controller.

RESULTS AND DISCUSSION

The experimental results given by the system proposed herein are convergent with an emerging field known as translational medicine. This new area aims to increase the number of syndromes diagnosed and therapeutic insights derived, by means of an improved communication between basic and clinical science.

In the context of social networks, web engineering specialized applications —such as the proposed syndrome classification system— tend to show a remarkable characteristic. These applications usually set a bridge that goes, on one hand, from the drawing board to the bedside, where the effectiveness of preclinical research results is tested on patients; and on the other hand, from the bedside to the drawing board, where the patients data can be used to derive new insights into the biological and clinical principles of human diseases.

Regarding such valuable patients data availability, several repositories have arisen to store and offer public access to clinical information produced by scientific researchers all over the world. The availability and convenience of use of such public data sets promotes scientific research, in particular the design, creation, and testing of innovative data analysis mathematical models.

In order to ease the task of performance comparison by using well-known data banks, the research team decided to employ data sets related to syndrome diagnosis, which also were available in one of the most prestigious and famous public data set repositories in the scientific community: the UCI Machine Learning Repository. According to tis website (http://archive.ics.uci.edu/ml/about.html), this repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms; it has been widely used by students, educators, and researchers all over the world as a primary source of machine learning data sets. To get an idea of how prestigious this repository is, notice that it has over 1000 citations, making the paper where it is presented one of the top 100 most cited papers in all of computer science. Specifically, 5 data sets having a direct relationship with syndrome diagnosis were chosen from the former repository.

For performance comparison, the results given by some of the best-performing algorithms in the WEKA 3 datamining software [34] were used. More specific information on each of the selected methods, and their implementation in WEKA, can be found in the Data Mining: Practical Machine Learning Tools and Techniques book.

One of the goals of the comparative study presented here is to perform a consistent and trustworthy comparison between the classification performance of the proposed system and other well-known methods. The consistency and trustworthiness of such comparison rests on two issues to be addressed: selection of suitable test sets on one hand, and selection of an appropriate comparison method on the other. In order to assess the performance of a classifier on new, unknown instances, its success rate on a dataset disjoint from the one used to train the method must be evaluated. When the amount of available data is sufficiently large, there may be little problem in choosing such a test set: just build two large and separate datasets, one for training and another for testing. Yet, when only a small amount of data instances is available, the question of how to predict the performance with limited data is controversial. Although several validation techniques for experimental performance under such conditions have arisen, cross-validation has become the method of choice for most situations. In this sense, Kohavi [34] compared several variants of both cross-validation and bootstrap, showing that although the latter has usually a lower variance, it exhibits extremely large bias in some problems. Thus, the particular method of stratified 10-fold cross-validations is recommended. This technique has the advantage of dealing with both issues mentioned before (i.e. selection of suitable test datasets and performance comparison method), since its consistent and trustworthy performance evaluation comes from an appropriate management of limited data for both training and testing tasks, as is further explained in [34].

Another aspect of the utmost importance for a performance comparison is that of choosing appropriate measures of performance. Those selected to be used here are the three main performance indicators for binary classification tests: sensitivity, specificity, and classification accuracy. These indicators are computed from the confusion matrix, which is a statistical tool usually used in supervised learning for evaluating a classifier performance. To do so, a confusion matrix presents the actual outcome of classification arrayed in rows and columns. In general terms, a classification task with L classes yields a confusion matrix of size LxL [34], with the diagonal elements representing

the correctly classified instances, and the rest of the elements representing misclassifications. Table 1 shows the usual confusion matrix for a binary classification test (i.e. two class classifier, such as “sick” and healthy”).

In such a confusion matrix as the one depicted in table 1, True Positive (TP) refers to those instances correctly classified as positive, while True Negative (TN) indicates those instances for which both the condition and outcome are negative. On the other hand we have the misclassified instances: those instances whose actual condition is negative but the test outcome is positive are termed False Positive (FP), while False Negative (FN) means that their condition is positive but their test outcome is negative.

Table 1. Confusion matrix for a binary classification test.

		TEST	
		P	**N**
REAL	**P**	True Positive	False Positive
	N	False Negative	True Negative

Both sensitivity and specificity are statistical indicators of performance for a binary classification test and, from a medical point of view, they help to assess the results of diagnostic and screening tests. On one hand, sensitivity or True Positive Rate (TPR) indicates the proportion of actually diseased patients in a screened population who are detected by the test and being diseased. Thus, sensitivity is a measure of the probability the test has of correctly diagnosing a condition, and is computed as follows.

$$sensitivity = \frac{TP}{TP + FN}$$

On the other hand, specificity or True Negative Rate (TNR) is the proportion of healthy people identified as so by the screening test. Then, specificity measures the probability of correctly identifying a healthy person, and is computed as follows.

$$specificity = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}}$$

Additionally to these indicators, a classifier performance may be measured by its success rate. The classifier outputs a class for each test instance, if the class is correct, then it is counted as a success: the success rate (or classification accuracy) is the proportion of correctly classified (or successful) instances over the whole test set. Thus, the classification accuracy (for a binary classification test) may be calculated using the following expression.

$$accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{TP + FN + TN + FP}$$

In order to illustrate the application of the selected comparison method (i.e. stratified 10-fold cross validation) along with these performance indicators, the Hepatitis Disease Dataset from the UCI Machine Learning Repository was used. This data bank contains clinical results data of 155 hepatitis patients belonging to two classes: 32 instances on the first class (die), and 123 instances on the second class (alive). Each instance consists of 20 attributes, including the class attribute. This dataset has multiple missing values. Due to the small size of the dataset and the considerable number of missing values, these cannot be discarded. In this case the missing values were substituted by the class mode for categorical features and by the class mean for continuous values.

According to [36] the standard way of predicting the classification accuracy of a learning technique is to use stratified 10-fold cross-validation. This method divides the dataset into 10 parts in which each class is represented in approximately the same proportion as in the full dataset. The classification algorithms will be executed 10 times, in each execution one different part will be used as the test set and the classification algorithm will be trained with the remaining nine parts. The success rate will be calculated for each execution. Finally, the 10 success rates are averaged to yield an overall success rate. To perform the comparison of our proposal with other pattern classification algorithms, we used the 10-fold cross-validation approach.

Table 2 shows the results of our proposal compared against the 10 best performers of WEKA on classification accuracy, when applied to the Hepatitis Disease Dataset; results for sensitivity and specificity are also included.

Notice that the proposed system exhibits the best performance in two of the three indicators: classification accuracy and specificity. Although its value is not the highest for sensitivity, it is also among the highest values.

Table 2. Comparative results on the Hepatitis Disease Dataset, for classification accuracy, sensitivity, and specificity (ordered by accuracy, best performance indicated in **bold**).

	Accuracy	Sensitivity	Specificity
Proposed system	**82.65**	0.81	**0.83**
DecisionTable	80.32	0.81	0.61
RBFNetwork	77.72	0.79	0.59
BFTree	76.10	0.80	0.43
ConjunctiveRule	75.78	**0.84**	0.37
NBTree	69.67	0.77	0.61
FT	69.03	0.77	0.60
JRip	68.39	0.74	0.61
PART	67.74	0.71	0.64
REPTree	62.58	0.81	0.40
SimpleCart	61.94	0.75	0.46

According to published classifiers comparative studies, oftentimes the classification accuracy is the most relevant indicator (particularly if the other two give good performance). Taking this into consideration, the research group opted to perform an experimental comparative study of the proposed system against the classifier methods included in WEKA (in the context of social networks), using classification accuracy as the sole performance indicator, based on the stratified 10-fold cross-validation approach. Thus, the compared algorithms were applied to four additional datasets, all related to syndrome classification, and all taken from the UCI Machine Learning Repository. The selected data banks are: Wisconsin Breast Cancer Dataset, Pima Indians diabetes dataset, Heart disease dataset, and Liver Disorders Dataset. Below are included brief descriptions of each dataset.

Wisconsin Breast Cancer Dataset: This dataset was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. The dataset has information of clinical cases of breast cancer. The dataset contains 699 instances belonging to two classes, 458 instances belong to the first class (benign) and 241 belong to the second class (malignant). Each instance consists of 10 attributes, including the class attribute. The dataset has 16 pattern with one missing values. The instances with missing values were deleted from the original dataset and the resulting data set was used for the experimental phase.

Pima Indians diabetes dataset: This database was originally owned by the National Institute of Diabetes and Digestive and Kidney Diseases, U.S. This dataset contains cases from a study that was conducted on female patients at least 21 years old of Pima Indian heritage. This dataset consists of 768 instances belonging to two different classes (500 “the patient tested positive for diabetes” cases, 268 “the patient tested negative for diabetes” cases). Each instance consists of 9 attributes, including the class attribute.

Heart disease dataset: This database comes from the Cleveland Clinic Foundation and was supplied by Robert Detrano, M.D., Ph.D. of the V.A. Medical Center, Long Beach, CA. The purpose of the dataset is to predict the presence or absence of heart disease given the results of various medical tests carried out on a patient. This dataset consists of 270 instances belonging to two different classes: presence and absence (of heart-disease). Each instance consists of 14 attributes, including the class attribute.

Liver Disorders Dataset: The Liver Disorders dataset was created by BUPA Medical Research Ltd. This dataset presents the results of a study of liver disorders that might arise from excessive alcohol consumption. It contains 345 instances belonging to two classes, 145 instances belong to the first class and 200 instances belong to the second class. Each instance consists of 7 attributes, including the class attribute.

Table 3 shows the comparison of experimental results obtained by the proposed system, as well as several algorithms included in WEKA, when applied to the four datasets mentioned before, measured by the classification accuracy indicator. The results are ordered by the accuracy on the Wisconsin Breast Cancer Dataset, and include

the 10 classifiers from WEKA which exhibit the best performance on this dataset. Notice that the performance of the proposed system is quite competitive; yet, no one classification method is the best performer on all datasets (as proved by the No-Free-Lunch Theorems).

As can be seen, the proposed system clearly outperforms the other methods on the Wisconsin Breast Cancer Dataset, closely followed by the DTNB, and somewhat farther along by two of the most popular and widely used algorithms around the world (and specifically on the WEKA platform): BayesNet and RotationForest.

Our proposal also has the best performance on the Heart disease dataset, but now tied with other three WEKA algorithms: RandomForest, Logistic, and MiltiClassClassifier. Even though this is not a clear-cut outperformance, this shared first place is valuable to classifying syndromes in social network environments.

In the other two datasets used for this comparative study, the proposed system came our further away from the first place; yet, it never showed the worst performance. This means that, although the proposal exhibits suboptimal performance, it still is amongst the 8 best classifiers of more than 70 classification algorithms included in WEKA, which is not a negligible result.

Table 3. Comparative results for classification accuracy on the Wisconsin Breast Cancer Dataset (Breast), Pima Indians diabetes dataset (Diabetes), Heart disease dataset (Heart) and Liver Disorders Dataset (Liver) (ordered by accuracy on the Wisconsin Breast Cancer Dataset, best performance indicated in bold).

	Breast	Diabetes	Heart	Liver
Proposed system	**97.58**	74.34	**83.70**	68.69
DTNB	97.51	73.82	82.59	57.97
RotationForest	97.21	76.82	82.59	**73.04**
BayesNet	97.21	74.34	82.22	56.81
RandomForest	97.07	72.39	**83.70**	70.72
SMO	96.92	77.34	83.33	57.97
FT	96.92	77.34	82.22	70.43
Dagging	96.77	74.08	82.22	57.97
SimpleLogistic	96.63	**77.47**	82.22	71.01
Logistic	96.63	77.21	**83.70**	68.69
MultiClass	96.63	77.21	**83.70**	68.69

CONCLUSIONS AND FUTURE WORK

In this paper, a web engineering system embedded in a social networks environment has been introduced. The proposed system employs the Morphological Associative Max Memories to perform the task of syndrome classification. In particular, the proposed syndrome classification web system was tested on several data banks taken from a well-known public dataset repository: Hepatitis Disease Dataset, Wisconsin Breast Cancer Dataset, Pima Indians diabetes dataset, Heart disease dataset, and Liver Disorders Dataset; all accessed from the UCI Machine Learning Repository.

Also included in this paper is a comparative study of the experimental performance shown by the proposal, and by several classification algorithms included in the WEKA platform. These results indicate a competitive performance by the proposed system, clearly outperforming the 10 methods with best performance on three of the five datasets (Hepatitis, Breast Cancer, and Heart disease), based on the classification accuracy (success rate) indicator and the stratified 10-fold cross-validation approach. Regarding sensitivity and specificity, the results suggest a competitive performance by the proposal.

Such competitive results exhibited by the tool developed and presented here suggest a positive impact on translational medicine, improving the communication between basic and clinical science so that more diagnostic of syndromes and therapeutic insights may be derived. Of particular interest to the authors of this proposal are the implications of deploying such tool in a social network environment, which could potentiate the insights reached thanks in part to the access to more numerous and rich data. The study of such impacts, as well as the application to other syndromes, remain as open problems to be worked in the future.

REFERENCES

[1] Barnes, M. R., Nicosia, V., & Clegg, R. G. (2025). Measuring social mobility in temporal networks. Scientific Reports, 15(1), 5941.

[2] Hossain, S. M., Rao, Y., Hossain, J. O., Pritchard, J. R., & Zhao, B. (2025). goloco: a web application to create genome scale information from surprisingly small experiments. BMC bioinformatics, 26(1), 1-8.

[3] Murugesan, S. and Ginige, A. Web Engineering: Introduction and Perspectives. In Su, W. ed. Web Engineering Principles and Techniques, Idea Group Publishing, 2005, 1-30.

[4] Su, W. Web Engineering Principles and Techniques. Idea Group Publishing, 2005.

[5] Mendes, E., Mosley, N., & Counsell, S. (2006). The need for web engineering: An introduction. Web Engineering, 1-27.

[6] Mendes, E. The Need for Empirical Web Engineering: An Introduction. in Rossi, G., Pastor, O., Schwabe, D. and Olsina, L. eds. Web Engineering: Modelling and Implementing Web Applications, Springer-Verlag, 2008, 421-448.

[7] Murugesan, S. and Ginige, A. Web Engineering: Introduction and Perspectives. In Su, W. ed. Web Engineering Principles and Techniques, Idea Group Publishing, 2005, 1-30.

[8] Standing, C. The Requirements of Methodologies for Developing Web Applications. in Su, W. ed. Web Engineering Principles and Techniques, Idea Group Publishing, 2005, 261-280.

[9] Wohlin, C., Höst, M. and Henningsson, K. Empirical Research Methods in Web and Software Engineering. in Mendes, E. and Mosley, N. eds. Web Engineering, Springer-Verlag, 2006, 409-430.

[10] Musia, K. and Juszczyszyn, K. Properties of Bridge Nodes in Social Networks. in Nguyen, N.T., Kowalczyk, R. and Chen, S.M. eds. Computational Collective Intelligence: SemanticWeb, Social Networks and Multiagent Systems, Springer-Verlag, 2009, 357-364.

[11] Yang, M., Zheng, H., Wang, H., McClean, S., Hall, J. and Harris, N. A machine learning approach to assessing gait patterns for Complex Regional Pain Syndrome. Medical Engineering & Physics, 34. 740-746.

[12] Park, H.S. and Cho, S.B. Evolutionary attribute ordering in Bayesian networks for predicting the metabolic syndrome. Expert Systems with Applications, 39. 4240-4249.

[13] Ritter, G. X., Diaz-de-Leon, J. L., & Sussner, P. (1999). Morphological bidirectional associative memories. Neural Networks, 12(6), 851-867.

[14] Li, S. and Sun, L., Advantages analysis of JSF technology based on J2EE. in International Conference on Computer Science and Service System (CSSS), (Nanjing, China, 2011).

[15] Prajapati, H. and Dabhi, V., High Quality Web-Application Development on Java EE Platform. in IEEE International Advance Computing Conference, (Patiala, India, 2009).

[16] Core J2EE Patterns - Data Access Object. Sun Microsystems. [Online]. Available: http://www.oracle.com/technetwork/java/dataaccessobject-138824.html

[17] Lee, B.S., Yan, S., Ma, D. and Zhao, G., Aggregating IaaS Service. in Annual SRII Global Conference, (San Jose, California, USA, 2011).

[18] Fan, P., Chen, Z., Wang, J. and Zheng, Z., Online Optimization of VM Deployment in IaaS Cloud. in IEEE 18th International Conference on Parallel and Distributed Systems, (Singapore, 2012).

[19] Mauch, V., Kunze, M. and Hillenbrand, M. High performance cloud computing. Future Generation Computer Systems, 29 (6), 1408-1416.

[20] Amazon Elastic Compute Cloud (Amazon EC2). [Online]. Available: http://aws.amazon.com/es/ec2/

[21] Amazon Relational Database Service (Amazon RDS). [Online]. Available: http://aws.amazon.com/es/rds/

[22] Amazon Simple Storage Service (Amazon S3). [Online]. Available: http://aws.amazon.com/es/s3/

[23] O'Neil, E. J., Object/relational mapping 2008: hibernate and the entity data model (edm). in Proceedings of the 2008 ACM SIGMOD international conference on Management of data, (New York, USA, 2008).

[24] Widenius, M. and Axmark, D. MySQL Reference Manual. O´Reilly Community Press, 2002.

[25] Brambilla, M., Ceri, S., Fraternali, P. and Manolescu, I. Process modeling in Web applications. ACM Transactions on Software Engineering and Methodology, 15 (4), 360-409.

[26] Cardoso, J., Hepp, M. and Lytras, M.D. The Semantic Web: Real-World Applications from Industry. Springer, 2008.

[27] Keith, M. and Schincariol, M. Pro JPA 2: Mastering the Java Persistence API. Apress, 2009.

[28] Richfaces. [Online]. Available: http://www.jboss.org/richfaces

[29] Zakas, N.C., McPeak, J. and Fawcett, J. Professional Ajax. Wrox, 2007.

[30] Leonard, A. JSF 2.0 Cookbook. PACKT Publishing, 2010.
[31] Hanakawa N. and Ikemiya, N., A Web Browser for Ajax Approach with Asynchronous Communication Model. in Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, (Washington, DC, USA, 2006).
[32] W3C. (2012, December) Xmlhttprequest. World Wide Web Consortium. [Online]. Available: http://www.w3.org/TR/XMLHttpRequest/
[33] Rossi, G. and Schwabe, D. Modeling and Implementing Web Applications with OOHDM. in Rossi, G., Pastor, O., Schwabe, D. and Olsina, L. eds. Web Engineering: Modelling and Implementing Web Applications, Springer-Verlag, 2008, 109-156.
[34] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann P. and Witten, I.H. The weka data mining software:An update. SIGKDD Explorations, 11. 10-18.
[35] Kohavi, R. and Provost, F. Glossary of terms special issue on applications of machine learning and the knowledge discovery process. Machine Learning, 30. 271-274.
[36] Witten, I.H. and Frank, E. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, 2005.