Development of a platform for the exchange of bio-datasets with integrated opportunities for artificial intelligence using MatLab

Christian Wansing[1], Oresti Banos[2], Peter Gloesekoetter[1], Hector Pomares[3], Ignacio Rojas[3*]

[1]Semicond & Bus Lab Dept.
University of Applied Sciences,
Steinfurt, Germany
* Corresponding author:
irojas@ugr.es

[2]Dept. Computer Engineering,
Kyung Hee University,
Suwon 446-701, Korea,

[3]Dept. Comput. Arch&Tech.,
CITIC-UGR,
University of Granada,
Granada, Spain

***Abstract*—This paper deals with the issue of automating the process of machine learning and analyzing bio-datasets. For this a user-friendly website has been developed for the interaction with the researchers. On this website it is possible to upload datasets and to share them, if desired, with other scientists. The uploaded data can also be analyzed by various methods and functions. The signals inside these datasets can also be visualized. Furthermore, several algorithms have been implemented to create machine learning models with the uploaded data. Based on these generated models new data can be classified or calculated. For all these applications the simplest possible handling was implemented to make the website available to all interested researchers.**

Keywords: M-health, machine learning, website

I. INTRODUCTION

The benefits arising from proactive conduct and subject-specialized healthcare have driven e-health and e-monitoring into the forefront of research, in which the recognition of motion, postures and physical exercise (i.e. fitness tracking) is one of the main subjects [9].

Fitness tracking is used in smartphones, stand-alone products and also more recently in smartwatches. On user demand, they are capable of recognizing a sporting activity and classifying it. In order that the app (short for application, a program that can be installed on a smartwatch or smartphone) is able to identify the current user's activity automatically, it must evaluate various sensor signals, including Global Positioning System (GPS) and acceleration sensors. These sensor signals are stored in a dataset for further processing and analysis. The data in this dataset is first filtered and then segmented (divided into several sections). A segment could be for example a short period of time.

The prediction of the current activity is then performed on one segment. In order to predict a sporting activity, the second step is to train a model. For this, the sensor signals of multiple sports activities are recorded, for example inline skating, jogging, hiking and cycling. Based on these sensor signals and the known sporting activity in which they were recorded the model is trained. The most common method that is used for this case is based on decision trees. In this method, a tree is built with different decision-making processes. An end of one decision path represents a sporting activity, for example hiking.

This simplified illustrated process is in reality more complex and requires professional knowledge to be performed successfully. Therefore, one aim of this paper is to automate this machine learning process to allow anyone getting started with it, without having detailed knowledge of signal filtering or feature extraction.

To do this, a website with a MatLab interface has been programmed and is presented in this contribution [1,2,3,5].

II. WEB PLATFORM DEVELOPMENT

To develop software today, it is necessary to reach as many people as possible with this software. Therefore, everybody has the opportunity to use the offered product. To make this possible, there must be at least one program available for all three major operating systems: Windows, Linux and Mac OS. Worthwhile would also be to have an app for the mobile operating systems Android, iOS and Windows Phone, to offer the mobile use of the software on these platforms. This would imply that six applications need to be developed and, more important, maintained. By selecting a suitable programming language such as Java or C++, the effort to develop several programs can be minimized but not completely lifted (the advantage of one of these languages is that they can be easily ported or just have to be recompiled). The downside of this is that it only works for the desktop operating systems and not for the mobile operating systems. For example, Android software is programmed in Java, and iOS apps in Swift(Objective-C). To share code between these two platforms is harder than for desktop systems.

A. Website

In order to develop an automated process, which is able to minimize the effort for machine learning, the solution chosen

978-1-4673-9669-1/15/$31.00 ©2015 IEEE

was to create a website. The reasons for this decision are the following ones:

1. Only one application must be programmed instead of six.
2. A website can be used by all of the above mentioned systems.
3. Simple scalability, the Webserver can be installed in a cluster environment
4. It is good for global software management, because if there is an error a rapid deployment of a fix is possible. The fix takes place immediately after publishing on the website. There is no need for the users of the website to perform an update on their own.
5. Accelerated deployment of features is possible.
6. Since it should be possible to share datasets and the results of their analysis with the publicity, a website is particularly suitable.

However, this decision also has a disadvantage. If there is more than one MatLab calculation performed on the server by several users, the server must be powerful enough to process the parallel computations. The required high performance also means higher running costs compared to a smaller and less powerful server. It must be also considered the following point: to enable a good user experience on mobile devices, the website must have a responsive layout, in order to use the page fluently on mobile devices. To store the datasets and their owner information on the website just a web server is not sufficient. There must be also a storage- or database engine available for this task.

B. Web server

The most important part, in order to host a website, is the web server. To choose the correct web server the following requirements must be considered and compared:

1. Security: If the web server is insecure, it is possible to attack the server itself, including all stored sensitive user data.
2. Scalability: If a website is successful and more and more people are connecting to the server, it must be possible to extend the server without reprogramming the whole site.
3. Performance: A good performance is required to provide a good user experience.
4. In order to call MatLab functions with an interface on the website, the web server must support a server-side programming language, which should also be compatible with MatLab.
5. Licensing: The licensing must allow the public use of the website.
6. Operating system: The web server must have an installation candidate for the server operating system.

After considering the above points, the decision was taken on the Microsoft web server, called "Internet Information Services" (IIS), in the most recent version 8.5 (June 2015). This server supports the .NET Framework programming language, which is also supported as a compatible MatLab language (Figure 1).

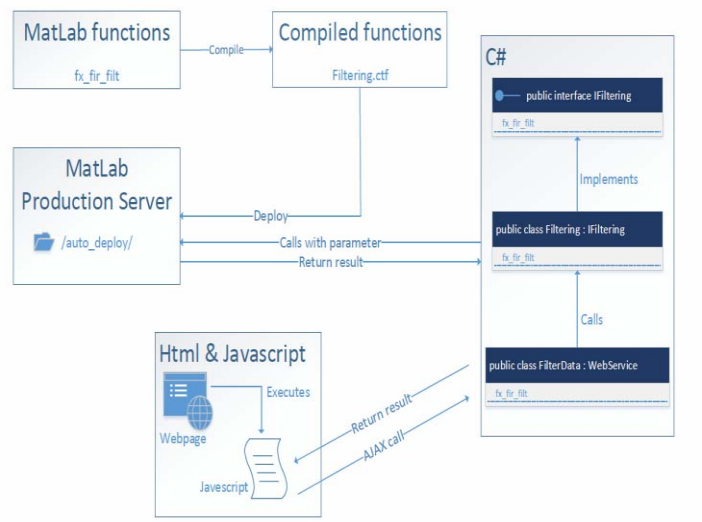

Figure 1 Overview of a MatLab function call with the website

The security of this web server is also very good. The server is directly integrated into every Windows installation, so the updates for this product are provided by the Windows Update mechanism, and there is no need to perform the updates manually. Furthermore, the worker process for the web server runs in its own isolated task, so that the web server can't affect the operating system. Even hosted websites on the same server are isolated from each other, so if there is an error in one site, it won't affect other websites. This is realized by the IIS feature Application Pools. Additionally, Application Pools allow setting different security levels for each website. In the case of the website this means that the security settings for the database server and MatLab can be exactly specified. This is a very important point for the security of the website.

The IIS-server also provides a good security in combination with a database server. The web server has a built-in request filtering function, which prevents for instance Distributed Denial-of-Service (DDoS)-attacks and SQL injection.

C. Database server

Choosing the database server is nearly as important as the web server. This server is responsible for storing all user information. Some of this information is security sensitive data, such as the email address and the login information. Therefore, it is very important to choose a secure server. Moreover, the SQL-server must be compatible with the web server, and the scalability and performance issues mentioned in section 4.1 for the web server also applies to the database. Another important point is the accessibility to the database server through the selected programming language. The programming language must have SQL support to access the data which is stored inside the server. Furthermore, it should be possible to call

predefined functions in the SQL-server such as stored procedures. In order to store the datasets on the server, it must be possible to store large files (Binary large object <BLOB> data) in the database server alongside the relational user data. After reviewing the requirements for the SQL-server the choice felt on the Microsoft SQL-server in the most recent version 2014 (June 2015). This server is perfectly suited for the IIS-server. The selected programming language also fulfills all requirements completely, like storing relational and BLOB data. The Application Pool feature of the web server can also be used with the SQL-server to specify the security level. The advantage of this is that it is easier to manage one security configuration instead of separated ones for the web and the database servers. With this configuration it is possible to set the security settings in detail, such as whether the website should have read or write access to one specified database. Another important point for the security is the update mechanism. The updates for this product are rolled out with Windows Update, so there is no need to perform updates manually. To provide the best performance for this server the following point was considered: whenever possible a batch command is used instead of row-by-row commands. This has the advantage that the server has to perform the SQL command only once and not for each row. For instance, if *n* data points should be inserted in a table on the database server, the batch command runs only once on the server, and the row-by-row command *n* times. An example with an inserting statement of 1 to 2000 data points can be seen in Figure 2. This figure shows the time which is needed to insert the number of data points in the SQL-server. A small number of inserts is faster with a row-by-row command, but with increasing points, it is better to insert with a batch command.

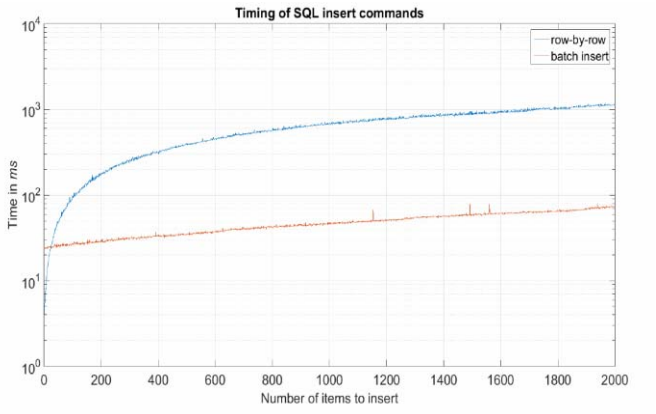

Figure 2 Comparison of different SQL insert commands

III. PROGRAMMING LANGUAGE

This section describes the programming languages which are used for the website. This section is divided into two parts: the server-side part, which describes the technologies used at the server, and the client-side part which expounds the client-side software.

A. Server-side

On the server-side an application framework called "ASP.NET", which was introduced in the .NET Framework version 1.0, is used. ASP.NET is an open source server-side framework which can be used for web development to build dynamic websites. ASP.NET allows it to write code in any language which is supported by the .NET Framework, because it is built on the Common Language Runtime (CLR).

The two most popular ones are C# (called C-sharp) and Visual Basic .NET. For the work of this paper the language C# is used for the server-side code. There are several reasons for this decision. The most important one is the compatibility with IIS and the SQL-server. Furthermore, it is possible to extend the functions of the Framework by different extensions. Another important point is the security. By default, ASP.NET has only read access to the web server's filesystem. There is no need to enable write access for storing the files of the datasets, because the SQL-server will store the files.

These security settings can also be specified with the Application Pool. Because ASP.NET is built upon the .NET Framework, it is possible to use all features which are available for it. This includes the use of the Framework Class Library (FCL) and the use of the Common Language Runtime (CLR). The Common Language Runtime is an application virtual machine [4] that provides extra security, because it runs in a virtual environment. An overview of this can be seen in Figure 3.

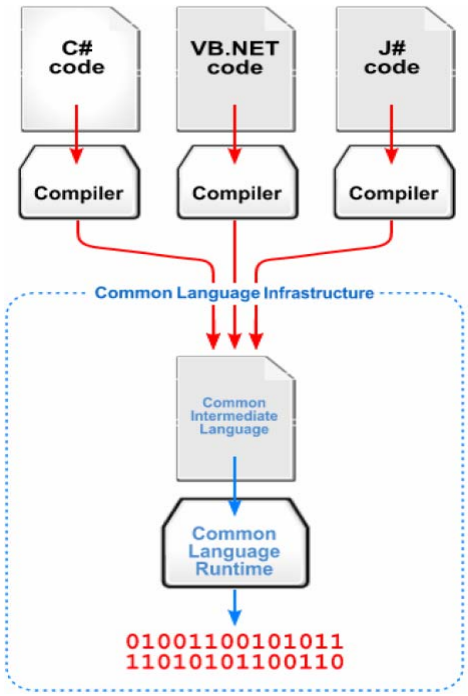

Figure 3 Overview of the Common-Language-Infrastructure. Source: https://en.wikipedia.org/wiki/.NET_Framework

Everything inside the Common Language Infrastructure (CLI) runs in a virtual environment. The Common Intermediate Language (CIL) is a platform neutral language; therefore, it can run on all desktop platforms. The platform-specific Common Language Runtime compiles the CIL code to machine code that can only run on the current platform. Furthermore, the CLR provides memory management and exception handling. So there is no need to allocate memory like in plain C. This is automatically done by the CLR. The Framework Class Library

provides among other things libraries for data access and database connectivity, to communicate with databases. Furthermore, there are libraries available especially for web application development, cryptography and network communications. The following section will describe the data access to the SQL-server through the selected programming language C#.

B. *Client-Side*

To structure the texts, pictures and other elements for each web page in the website, HyperText Markup Language (HTML) is used. HTML is the standard markup language used to create websites. These HTML files will be interpreted and rendered by web browsers into the browser window to show the content of the page. But just with HTML it is impossible to create a presentable and responsive website. Therefore, the layout for this website is based on Bootstrap [6], which was created by Twitter in 2010 and is now developed by Mark Otto, Jacob Thornton and Bootstrap contributors. Bootstrap offers ready-to-use Cascading Style Sheets (CSS) to easily make a beautiful responsive layout. It automatically adapts websites for various screen sizes, so that the website can be used with a mobile device or a desktop computer. These CSS-files just have to be implemented into the HTML code to use the layout Bootstrap provides.

In order to establish a communication between the client and the server, two HyperText Transfer Protocol (HTTP) request methods, GET and POST are used.

Another programming language used on the client side is Javascript. Javascript is a dynamic programming language which is used at the client and not at the server-side. It is supported by all modern browsers, including mobile browsers. Javascript is used for the website to display the signals of datasets in graphs or to show an overview of the submitted datasets in a table with sorting and filtering functions. Together with Asynchronous Javascript and XML (AJAX), Javascript is capable of loading data asynchronously without reloading the browser window. These two technologies are for example used to implement the filtering feature in the datasets overview. With a search term it is possible to filter the datasets. This search term is passed to the server without reloading the window. The response for this request is sent by the server back to the client which will render this response to display the filtered results.

The technique is also used to start a MatLab calculation on the server. The user can start the calculation with a button click in the browser, and AJAX submits all parameters for this function to the server which will perform the calculation.

IV. BIO-DATASETS

The bio-datasets are the most important part for the work of this contribution. The goal of this paper is to analyze the datasets and implement artificial intelligence features with the data inside them. All this should be done by an easy-to-use interface and without the requirement for the user to program MatLab functions on their own.

A. *Privacy and security*

As declared above, the security and privacy of the user data was a very important point during developing the website. The same also applies to the datasets and the data inside them. Because not everyone is willing to share their datasets or the results of the analysis, it is possible on the website to mark datasets as private. Any datasets which are marked as private can be seen only by the creator. Any third-party users don't have access to them. The same goes also for the files. All files which belong to a private dataset are also marked as private and can't be viewed or downloaded by others. So the complete data and analysis is stored safely, according to the privacy, inside the website.

B. *Displaying the signals*

To display the signals, a line chart was implemented into the website. For this chart the Javascript library "Highcharts" is used (http://www.highcharts.com/). Highcharts is available for all modern browsers, including mobile browsers. This makes it possible to plot signals also on mobile devices, with native pinch-to-zoom functionality. Because Highcharts is written in Javascript, the signal is plotted on the client-side, by the browser of the user. The server just submits the data to the client. This is implemented through AJAX.

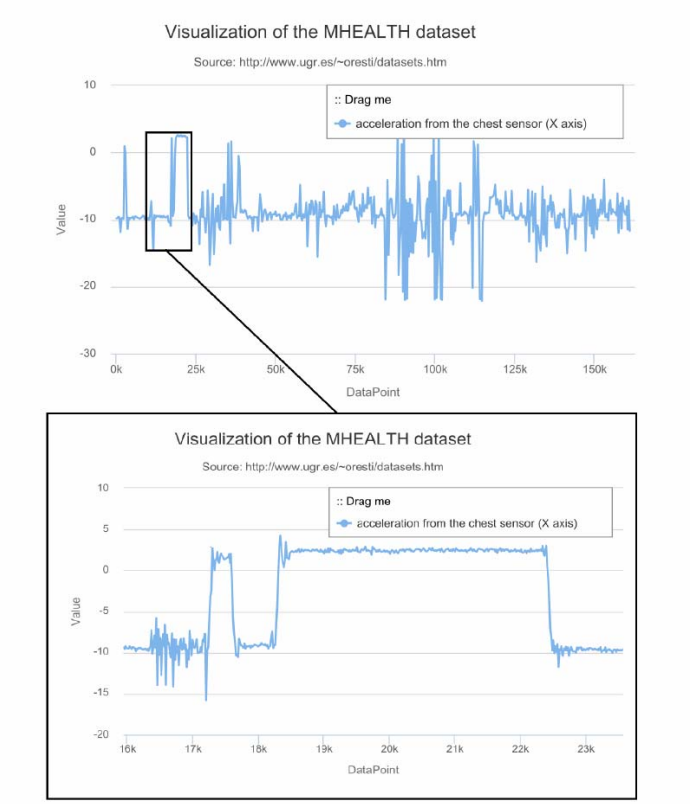

Figure 4 Exported chart of a signal

The client sends a request to the server to plot the data. The user submits the parameters, for example which file and which signal should be plotted. The response from the server is a Javascript Object Notation (JSON) formatted string with the x

and y values. The Highcharts library interprets this string and displays the result. To plot the graph on the client-side has several advantages: the server can save resources, because it just sends the data and don't plot the chart itself. Another important advantage of this feature is that there is more responsiveness for this solution. For instance, it is possible to implement a zooming feature. When the user zooms into the graph, to show a more detailed part of the plot, the browser will send a request to the server to load the more detailed data points. The library will render the response and show the zoomed signal. To share the plot of the signal, an export function is available for the chart. It is possible to download the chart as an image, including vector images, or as a pdf file. An example can be seen in Figure 4.

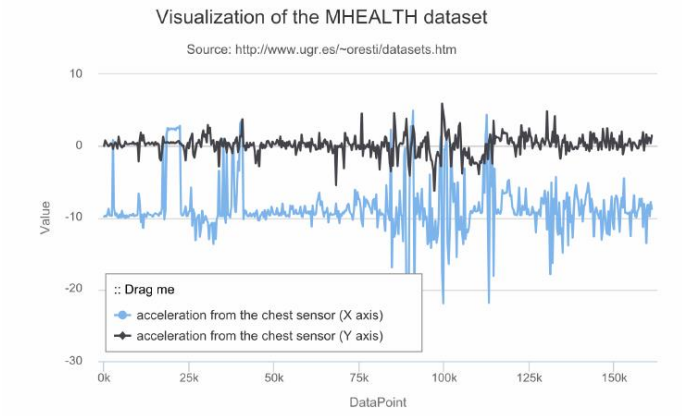

Figure 5 Exported chart of multiple signals

Another good advantage of the chart is the possibility to display more than one signal into the plot. With this feature it is possible to compare signals, which are measured at different conditions. For example, in Figure 5 the signal of the acceleration sensor on the chest in x and y direction is shown.

V. MACHINE LEARNING MODEL

In order to train a machine learning model there must several steps performed first.

1)Acquisition of data: The data must be collected by sensors, or created by a mathematical description.
2)Split the data into a training and a testing part. To test the model in the last step, the input data must be split. The trained model can be afterwards tested with the testing data.
3)Preprocessing: The sensor signals can be filtered in this step to eliminate measuring errors.
4)Segmentation: Segment the data into different blocks. Example: A time-discrete signal can be split into multiple intervals, e.g. split every few seconds.
5)Feature Extraction: Replace the data by different features. Example: A 10-row, 5-column interval, which will be replaced by 2 features.
6)Train the model: Train the model with the training part of the data.
7)Test the trained model: Test the model with the testing data to calculate the misclassification rate.

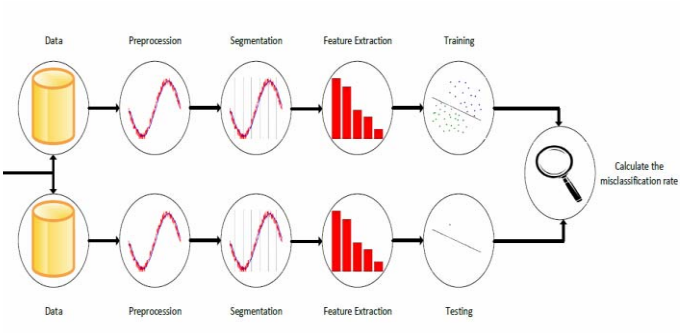

Figure 6 Flow chart to train a machine learning model

This process is displayed in Figure 6.

The following methods for training a machine learning model are supported by the website:

- Naive Bayes classifier: Naive Bayes classifiers are simple probabilistic classifiers based on applying Bayes' theorem [7] with naive independence assumptions between the features.
- Support Vector Machine (SVM): A Support Vector Machine divides a set of objects into classes so that the class boundaries around the widest possible area are maximized. Support Vector Machines can be used both for classification and for regression.
- Multiclass Support Vector Machine: Implementation of normal Support Vector Machine with multi class assistance.
- Decision trees: Decision trees are ordered, directed trees from which the representation of decision rules. The graphical representation of a tree diagram illustrates a hierarchical consecutive decision. The end of a decision tree represents a decision.
- k-nearest neighbor classifier: The k-nearest neighbor's algorithm is a nonparametric method for estimating probability density functions. The resulting k-nearest neighbor classification algorithm is a method in which a class assignment is made in consideration of its k-nearest neighbors.
- Discriminant analysis classifier: Discriminant analysis is a generalization of Fisher's discriminant [8]. A method used to find a combination of features that characterizes or separates two or more classes of objects or events.

ACKNOWLEDGMENTS

This work was partially supported by the Junta de Andalucia Project P12-TIC-2082.

REFERENCES

[1] The MathWorks Inc. MathWorks - MATLAB and Simulink for Technical Computing. url: http://www.mathworks.com

[2] The MathWorks Inc. Parallel Computing Toolbox - MATLAB. url: http://www.mathworks.com/products/parallel-computing/index.

[3] The MathWorks Inc. MATLAB Documentation. url: http://www.mathworks.com/help/matlab/.

[4] Andrew Kennedy and Don Syme. "Design and implementation of generics for the. net common language runtime." In: ACM SigPlan Notices. Vol. 36. 5. ACM. 2001, pp. 1–12.

[5] Microsoft Corporation. aspnet/EntityFramework. url: https://github.com/aspnet/EntityFramework (last visited on 06/29/2015).

[6] Mark Otto, Jacob Thornton, and Bootstrap contributors. Bootstrap The world's most popular mobile-first and responsive front-end framework. url: http://getbootstrap.com/

[7] Dennis V Lindley. "Fiducial distributions and Bayes' theorem." In: Journal of the Royal Statistical Society. Series B (Methodological) (1958), pp. 102–107.

[8] Leo H Chiang, Evan L Russell, and Richard D Braatz. "Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis." In: Chemometrics and intelligent laboratory systems 50.2 (2000), pp. 243–252.

[9] Oresti Banos, Miguel Damas, Hector Pomares, Alberto Prieto, and Ignacio Rojas. "Daily living activity recognition based on statistical feature quality group selection." In: Expert Systems with Applications 39.9 (2012), pp. 8013–8021.