PredicForex. A tool for a reliable market. Playing with currencies.

J. Aguilera Collar [1], R. González-Cebrián Toba [1], C. Cortés Velasco [2]

[1] *OpenLab*
[2] *European Virtual Engineering*

Abstract — **The Forex market is a very interesting market. Finding a suitable tool to forecast currency behavior will be of great interest. It is almost impossible to find a 100 % reliable tool. This market is like any other one, unpredictable. However we developed a very interesting tool that makes use of WebCrawler, data mining and web services to offer and forecast an advice to any user or broker. (October, 2009)**

I. INTRODUCTION

THE currency market also known as FOREX market stands for The Foreign Exchange Market is the biggest and most liquid financial market on earth , with a daily sales revenue of about US $ 1,5 $*10^{18}$ against the $ 25*10^2 traded on the New York Stock Market, leaving the stock Market in the second place. [2]

It is the market where currencies are being sold. In fact, it is the Market where money is being bought and sold. In other words, Forex is the market where a currency is sold and another one is bought. World currencies quote in a floating exchange rate and they are always handled in pairs, for instance EURO/USD or USD/YEN. This market was born in 1971, in the moment when currencies stop working in a fixed rate. This market works 24 hours per day and 5 days every week. It is a non stooping market. [2]

In the Forex Market, transactions are taken within central banks, governments, currencies gamblers, multinational companies, large organizations, small sized banks and so on. This provides the market of great liquidity and allows participants to execute their great value transactions without affecting significantly the prize of the coin.

The currency market does not operate from a given fixed

J. Aguilera Collar, computer engineer by the Pontifical Salamanca´s University at Madrid, Spain.
R. González - Cebrián Toba, computer engineer by the Pontifical Salamanca´s University at Madrid, Spain.
C. Cortés Velasco, computer engineer by the Pontifical Salamanca´s University at Madrid, Spain. She is currently working at European Virtual Engineering (EUVE), technological centre in Spain.

position, although there are several places of great importance like New York, London, Tokyo or Frankfort. It is a market that works through internet and the phone. [2]

Long time ago, in order to make a FOREX transaction it was necessary to have great quantities of money in order to operate and only the large companies and governments or central banks were capable of this. Nowadays everybody can take the advantages of the Forex system. Internet provides the necessary infrastructure to make this possible.

The exchange of currencies is a high value element in giving support to global economic commerce and as the currencies move ones against the others, great opportunities will arise. Although big dealers play with huge amounts of money such as millions of dollars, small players can also find opportunities in making money in this market.

Any given person has the opportunity to take part in this market and with a little bit of money and the necessary time to learn in order to operate in the currency markets, it is possible to enjoy of great revenues dealing with currencies.

The most common currencies in the Forex Market are the American Dollar USD $, the Euro €, the Japanese Yen JPY ¥, the English Pound GBP £, the Swiss Franc CHF and the Australian Dollar AUD $. [3]

Although it is possible to realize transactions with any given pair of currencies, 60 % of all transactions involve the Euro and the Dollar, what makes them the most used currencies with a big margin.

In the Forex market we can find gamblers which are basically betting against a coin. Hoping that some coins are going to increase their value while others will decrease.

There are many controversies about the job being done by the gamblers. Sometimes they can influence in the value of a currency or even affecting the whole economy of an entire country. [3]

Some people think they only move by their avarice and because they want to make profit, manipulating countries inside politics in their own benefit, being careless about the social consequences this can have.

However, defenders state that they help to adjust currencies all over the world.

II. WHY WE WANTED TO INVEST IN THE FOREX MARKET

Nowadays we live an uncertain moment about the world economy. The stock market is not reliable any more. Only the raw materials, gold, and public debit seem to be enough reliable or at least considered as save values. And there is a non existing tool capable of helping investors to take decisions weather to invest in the Euro/Dollar pair will be of great interest.

The economists use several theories about how the stock market changes, the market cycles and so on. They mix it with statistics and then they find the solution. The problem is that there are a lot of data to keep in mind at the same time and they can "forget" any of them, so that, *Predicforex* can be the solution, because it can make a prediction without missing any important thing. This is why we felt motivated to develop this tool.

III. WHAT WE AIMED AND HOW WE MADE IT

In the given project we had as objective to achieve the simulation or approximation of a program that can forecast or at least try to guess what the trend is going to be in the Dollar and the Euro.

Therefore we used the existing technologies: Netbeans as the Java platform where the code was developed and the data mining tool Weka. In addition to all these we used spiders or web crawlers to search for the information or data through the internet.

Our tool achieves to be suitable for investors or even gamblers helping them to have a better perception about the Forex Market and all the factors or conditions that can affect their investments and preventing lost in patrimony.

In order to make this possible we need three different subsystems. One subsystem is to acquire the information from web pages, another subsystem is to process all these information to make the desired recommendations and the last one is to show the advice on the given currencies on a web page. How we made it?

- We decided to recollect the information with the help of web crawlers.
- Process Data with mining tool Weka.
- Show information using a JEE server.

We needed all the information stored in *arff* files. This is because Weka works with this kind of files. We had to translate the data into this format. All this is string conversion code. From a Java API and with its correspondent code we call the necessary Weka processes. [1]

The *arff* files have the following structure:
@RELATION forex
@ATTRIBUTE DJ numeric
@ATTRIBUTE SyP numeric
@ATTRIBUTE USDGBP numeric
@ATTRIBUTE AMEX_Oil numeric
@ATTRIBUTE NASDAQ100 numeric
@ATTRIBUTE decision {compra,vende,nose}
@DATA

8331.72,891.2,0.6575,932.97,1359.54,compra
5000,891.2,0.6575,932.97,1359.54,vende

"Compra" is the Spanish term for sale while "vende" is for buy. "Nose" is the doubt.

At the end of the file, in the last two lines, you can find some quantities (five to be exact) separate per commas and followed for one of the words mentioned above. These quantities are the value of each factor which takes part on the process. The factors are: "Dow Jones" (DJ), "S&P 500" (SyP), Dolar/Pound price (USDGBP), "Oil Barrel" (AMEX_Oil), "Nasdaq" (NASDAQ100) respectively. These values are used to train the neuronal net. We introduce some of the data and we classify them in the three possible places: "compra", "vende", and the last one "nose". Now, the neuronal net has a model to classify the following situations, the ones the crawlers will choose.

But, how the crawlers work? In a very easy way. Each one minute (or the time you want) the crawlers surf the net (some given URLs, like "www.eleconomista.es", http://finance.yahoo.com, and so on) looking for new data. We introduce these data at the end of our file with the prediction to increase it. Whereas, our tool uses this file for training the neuronal net with all the historic data of previous rates and another one with a single line working as the instances that wants to be predicted. Here is shown the output of the net training:

=== Run information ===

Scheme:
weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R
Relation: forex
Instances: 3
Attributes: 6
        DJ
        SyP
        USDGBP
        AMEX_Oil
        NASDAQ100
        Decision

Test mode: user supplied test set: size unknown (reading incrementally)

=== Classifier model (full training set) ===

Sigmoid Node 0
  Inputs   Weights
  Threshold   -2.7531661267028182
  Node 3   2.0176783647722734
  Node 4   2.122999469333026
  Node 5   1.7215560277261523
  Node 6   1.0053816557295714
Sigmoid Node 1
  Inputs   Weights

```
   Threshold    2.7567411013790055
   Node 3    -2.066370442548119
   Node 4    -2.0495607009915946
   Node 5    -1.7471105876114683
   Node 6    -1.012914205526177
Sigmoid Node 2
   Inputs    Weights
   Threshold    -2.5074816470045764
   Node 3    -0.7720623255127751
   Node 4    -0.849179639879635
   Node 5    -0.8813380163659676
   Node 6    -0.8949421547493384
Sigmoid Node 3
   Inputs    Weights
   Threshold    -0.13860864232277748
   Attrib DJ    2.862352542187435
   Attrib SyP    -0.03533983423534818
   Attrib USDGBP    -0.039550931374902835
   Attrib AMEX_Oil    0.049072278571478306
   Attrib NASDAQ100    0.03172970714093244
Sigmoid Node 4
   Inputs    Weights
   Threshold    -0.13028474378381705
   Attrib DJ    2.8951004190015275
   Attrib SyP    -0.04767618775161106
   Attrib USDGBP    0.012514636346555932
   Attrib AMEX_Oil    -0.0012767152969857185
   Attrib NASDAQ100    0.033889035004701834
Sigmoid Node 5
   Inputs    Weights
   Threshold    -0.15659666445987427
   Attrib DJ    2.6007965004089377
   Attrib SyP    0.004673975719846554
   Attrib USDGBP    -0.008920380450893828
   Attrib AMEX_Oil    0.024624140532233052
   Attrib NASDAQ100    0.002669943460486615
Sigmoid Node 6
   Inputs    Weights
   Threshold    -0.14999743751166328
   Attrib DJ    1.9011025605260115
   Attrib SyP    0.04644868606768501
   Attrib USDGBP    0.02763122912749326
   Attrib AMEX_Oil    0.02331520701949938
   Attrib NASDAQ100    0.039933501161149354
Class compra
   Input
   Node 0
Class vende
   Input
   Node 1
Class nose
   Input
   Node 2
```

Time taken to build model: 221.59 seconds

=== Predictions on test set ===

 inst#,   actual, predicted, error, probability distribution
   1   3:nose 1:compra   + *0.901 0.092 0.007

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances	0	0	%
Incorrectly Classified Instances	1	100	%
Kappa statistic	0		
Mean absolute error		0.6617	
Root mean squared error		0.7758	
Relative absolute error		119.1142 %	
Root relative squared error		130.7858 %	
Total Number of Instances	1		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0	1	0	0	0	?	compra
	0	0	0	0	0	?	vende
	0	0	0	0	0	?	nose
Weighted Avg.	0	0	0	0	0	0	

=== Confusion Matrix ===

```
 a b c   <-- classified as
 0 0 0 | a = compra
 0 0 0 | b = vende
 1 0 0 | c = nose
```

Finally, you can see the tree which has been generated as the result. However, we used Weka through Weka libraries. In the following URL we found all the information necessary to make use of the correspondent libraries. It's necessary to import *weka.jar* to our Netbeans project.
   http://weka.sourceforge.net/doc.dev/

There it's stated the class used with its most important methods during the development of our project. We found the class *MultilayerPerceptron* inside *weka.classifiers.functions* package [5], the one we used to implement a part of *Predicforex*.

```
   java.lang.Object
      weka.classifiers.Classifier
         weka.classifiers.functions.MultilayerPerceptron
```

We decided to use this algorithm because we thought this is the one who fixed the best with the aim we were looking for. As you can read in the web, the algorithm is "A Classifier that uses backpropagation to classify instances. This network can be built by hand, created by an algorithm or both. The network can also be monitored and modified during training time. The nodes in this network are all sigmoid

(except for when the class is numeric in which case the output nodes become unthresholded linear units)." [5], so that, as you can, it could be the better one. Inside our code, the first line that refers to the *MultilayerPerceptron* algorithm is as follows:

Classifier miclassifier = new MultilayerPerceptron();

Here we use the *MultilayerPerceptron* constructor:

public MultilayerPerceptron();

Later on, you find the following sentence:

miclassifier.buildClassifier(instancias);

With this sentence we build the neuronal network. We make it by calling *buildClassifier*:

public void buildClassifier(Instances i);
throws java.lang.Exception;

The following line is very important. It calls this function to predict the class of an instance once a classification model has been built with the *buildClassifier* call.

A = miclassifier.distributionForInstance(i);

It returns a double array filled with the probabilities of each class type.

The broker, once started the application will see the following window with the *expected* advice.
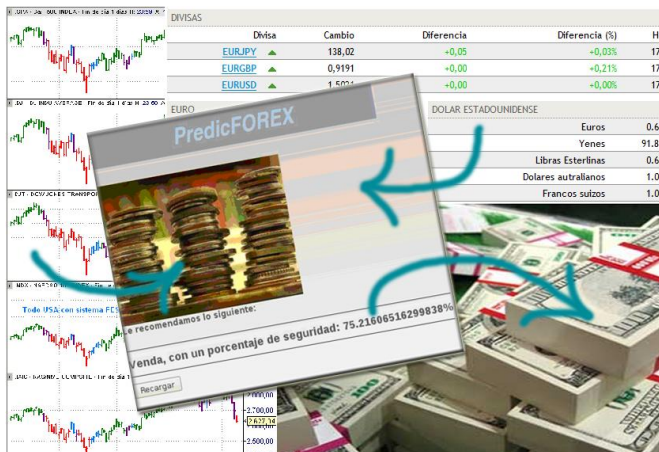

Fig. 4: PredicForex interface

All the application layers, subsystems and processes are hidden, the broker cannot see them.

IV. CONCLUSION

With this project we acquired a lot of knowledge in the neuronal-network field. We learned both theoretical knowledge and practical knowledge. Through this process we aimed and achieved the goal of forecasting currencies. *Predicforex* was made successfully. Anyone interested in making investment in the currency markets can use this application.

We consider this tool to be of great interest to anyone. We believe it is reliable and trustable.

If the broker can access to a large enough data base will benefit of more reliability and will have enough proved information to gamble against competitors.

How do we evaluate our project? Well, it depends on what you are looking for. As we said, *Predicforex* is a very good tool to predict the mentioned data and it works successfully. But, as our poor knowledge in the economy field, the data we used to train the neuronal net are not real. If you want real values, you'll have to contact to a professional on it to make the tool a very reliable one.

V. ACKNOWLEDGMENT

REFERENCES

[1] Class Notes from PhD. J. Soto
[2] http://www.forex.es
[3] http://es.dailyforex.com/
[4] http://www.cs.waikato.ac.nz/ml/weka/
[5] http://weka.sourceforge.net/doc.dev/