Acta Universitatis Apulensis,
ISSN 1582-5329, 2011,
Special Issue on Understanding Complex Systems,
Eds. Barna Iantovics, Ladislav Hluchý and Roumen Kountchev

KNOWLEDGE DISCOVERY IN A SCADA SYSTEM DATABASE

Maria Muntean, Honoriu Valean, Corina Rotar, Mircea Risteiu

Abstract. This paper presents three data mining techniques applied on a SCADA system data repository: Naïve Bayes, k-Nearest Neighbor and Decision Trees.

A conclusion that k-Nearest Neighbor is a suitable method to classify the large amount of data considered is made finally according to the mining result and its reasonable explanation.

The experiments are built on the training data set and evaluated using the new test set with machine learning tool WEKA.

Keywords: *Classification, Data Mining, SCADA System.*

2000 *Mathematics Subject Classification*: 68, 68T99.

1. Introduction

Knowledge discovery in databases (KDD) represents the overall process of converting raw data into useful information. According to the definition given in [1], KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. This process consists of a series of transformation steps, from data preprocessing to post-processing of data mining results.

Data mining, the central activity in the process of knowledge discovery in databases, is concerned with finding patterns in data. It consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns (or models) over the data [1].

Classification is one of the primary tasks in data mining. It represents the task of learning a target function (classification model) that maps each attribute set to one of the predefined class labels [2]. In other words it consists in assigning objects to one of several predefined categories.

The evaluation of the performance of a classifier is a complex process. The inducer's complexity, cost, usefulness, generalization error and success rate should be taken in consideration when evaluating the predictive performance for the learned model. The most well-known performance metric is the success rate, which is based on counting the test records correctly and incorrectly predicted by the classification model. These counts can be displayed as a two-dimensional confusion matrix, with a row and column for each class.

The most important examples of classifiers from literature are: Decision Tress, Naïve Bayes, Neural Networks, Association Rules, k-Nearest Neighbor and Support Vector Machines. For solving our problem we chosen three different classifiers: Naïve Bayes, k-Nearest Neighbor and Decision Trees.

A Naïve Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions.

Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In spite of their naive design and apparently over-simplified assumptions, Naive Bayes classifiers often work much better in many complex real-world situations than one might expect [3].

An advantage of the Naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

Instance-based (IB) learning methods simply store the training examples and postpone the generalization (building a model) until a new instance must be classified or prediction made. (This explains another name for IB methods – lazy learning – since these methods delay processing until a new instance must be classified).

K-nearest neighbor, an IB learning method, is a supervised learning algorithm where the result of new instance query is classified based on majority of K-nearest neighbor category. The purpose of this algorithm is to classify a new object based on attributes and training samples. The classifiers do not use any model to fit and only based on memory. Given a query point, we find K number of objects or (training points) closest to the query point.

K-nearest neighbor (k-NN) method assumes all instances correspond to points in the n dimensional space. The nearest neighbors of an instance are defined in terms of the standard Euclidean distance.

Decision tree learning represents one of the simplest, yet most popular methods for inductive inference. It has been successfully applied to a wide variety of problems from medical diagnosis to air traffic control or the assessment of credit risk for loan applicants. Its popularity is justified by the fact that it has some key advantages over other inductive methods. First of all, decision trees offer a structured representation of knowledge (as disjunction of conjunctive rules). As a direct consequence, decision trees may be rewritten as a set of " if-then" rules, increasing human readability. Secondly, decision trees are robust to errors, requiring little or no data preprocessing. Other important features include the capacity of handling both nominal and numeric attributes, as well as missing values and a good time complexity even for large data sets.

Structurally, a decision tree is a graph, whose inner nodes are " branching nodes" , because they contain some attribute test; the leaves contain the classification of the instance; the branches of the tree represent attribute values. The tree classifies an instance by filtering it down through the tests at the inner nodes, until the instance reaches a leaf.

The technique employed for building a decision tree is that of top-down induction, which performs a greedy search in the space of possible solutions. The first decision tree algorithm was introduced by J.R.Quinlan in 1986, and was called ID3. A large proportion of the decision tree learners that have been developed since are improved variants of this core method; the most successful of them was the C4.5 algorithm, also developed by Quinlan [4].

2. Data Analysis

We chose the well-known Weka environment as the data mining tool to implement the experiment. Originally proposed for didactic purposes, Weka is a framework for the implementation and deployment of data mining methods. It is also an open-source software developed in Java, released under the GNU General Public License (GPL), being currently available to Windows, MAC OS and Linux platforms [7]. Weka contains tools for classification, regression, clustering, association rules, data visualization and works with .arff files (Attribute Relation File Format) and also with files in .csv format (Comma Separated Values).

The classifiers are the most valuable resource that Weka provides, and can be used in a variety of ways, such as applying a learning method to a dataset and analyzing its output to learn more about the data; or using learned models to generate predictions on new instances; a third is to apply several different learners and compare their performance in order to choose one for prediction.

We chose three datasets that contains values of 84 parameters of a SCADA system returned in June of 2007 to implement the experiment (Figure 1).

A large amount of information, obtained by the data collection equipment, was recorded and accumulated in the database of the SCADA system used. The datasets used are presented in Table 1.

Dataset	Number of instances	Number of attributes
1-10.06.07	14403	84
11-20.06.07	14400	84
21-30.06.07	14397	84

Table 1. The datasets used in the experiment

Supervisory Control and Data Acquisition (SCADA) systems provide automated control and remote human monitoring of real world processes in many fields as: food, beverage, water treatment, oil and gas, utilities.

The SCADA system is used to monitor and control a plant or equipment and is a combination of telemetry and data acquisition. Data acquisition deals with the methods used to access information or data from the controlled

```
@RELATION 'SCADA'
@ATTRIBUTE  FeedwaterTempInlettoEcom NUMERIC
@ATTRIBUTE  FlueGasO2 NUMERIC
@ATTRIBUTE  BoilerExitFluerGasTemp NUMERIC
@ATTRIBUTE  WoodwasteSteamFlow NUMERIC
@ATTRIBUTE  TotalSteamFlow NUMERIC
@ATTRIBUTE  WoodwasteMoisturetoBoiler NUMERIC
@ATTRIBUTE  CombustionAirHeaterAirOutletTemp NUMERIC
@ATTRIBUTE  PostFDFanCombustionAirTemp NUMERIC
@ATTRIBUTE  SuperheaterOutletSteamPressure NUMERIC
@ATTRIBUTE  SuperheaterOutletSteamTemperature NUMERIC
@ATTRIBUTE  UndergrateColdAirCalc NUMERIC
@ATTRIBUTE  OutsideAirTemp10MinAverage NUMERIC
@ATTRIBUTE  GasBurnerCombustionAirFlow NUMERIC
@ATTRIBUTE  SuperheaterInterstageTemp NUMERIC
@ATTRIBUTE  Superheater1OutletTemp NUMERIC
@ATTRIBUTE  IntemperatorSprayWaterFlow NUMERIC
@ATTRIBUTE  InletCombustionAirFlow NUMERIC
@ATTRIBUTE  AirtoUnderGrateTemp NUMERIC
@ATTRIBUTE  CombinedAirtounderGrateFlow NUMERIC
@ATTRIBUTE  SecondaryAirFlow NUMERIC
@ATTRIBUTE  AirtoDryingGrateTemp NUMERIC
@ATTRIBUTE  TertirayAirFlow NUMERIC
@ATTRIBUTE  CombustionAirTemp NUMERIC
@ATTRIBUTE  GeneratorActivePower NUMERIC
@ATTRIBUTE  GasSteamFlow NUMERIC
@ATTRIBUTE  CombineAirFlowtoSlopingGrate NUMERIC
@ATTRIBUTE  OperatorMeasuredWWMoisture NUMERIC
@ATTRIBUTE  UndegrateDamper1Position NUMERIC
@ATTRIBUTE  UndegrateDamper2Position NUMERIC
@ATTRIBUTE  UndegrateDamper3Position NUMERIC
@ATTRIBUTE  UndegrateDamper4Position NUMERIC
@ATTRIBUTE  PrecipZone1KV NUMERIC
```

Figure 1: The parameters of the SCADA system

equipment while telemetry is a technique used in transmitting and receiving this information over a medium.

SCADA has traditionally meant a window into the process of a plant and/or a method of gathering of data from devices in the field. Today, the focus is on integrating this process data into the actual business, and using it in real time. In addition to this, today's emphasis is on using Open Standards, such as communication protocols (e.g. IEC 60870, DNP3 and TCP/IP) and 'off-the-shelf' hardware and software, as well as focusing on keeping the costs down [6].

Concerning SCADA systems, there are at least two main issues: the reliability of the system and the optimal management of the huge amount of data being transferred to the SCADA server by the communication systems [7].

Our paper deals with the second issue and intends to contribute to a better using of communication lines ant to an economy of storing space. One can ask: all the time, all the acquired data are of the same importance for the plant control? Maybe a preprocessing at sensor level and some decisions taken at this level are better solutions than passing all the data to the server.

2.1. Data Preparation

The original data set included noisy, missing and inconsistent data. Data preprocessing improved the quality of the data and facilitated efficient data mining tasks.

Before the experiment, we prepared data suitable to next operation as following steps:

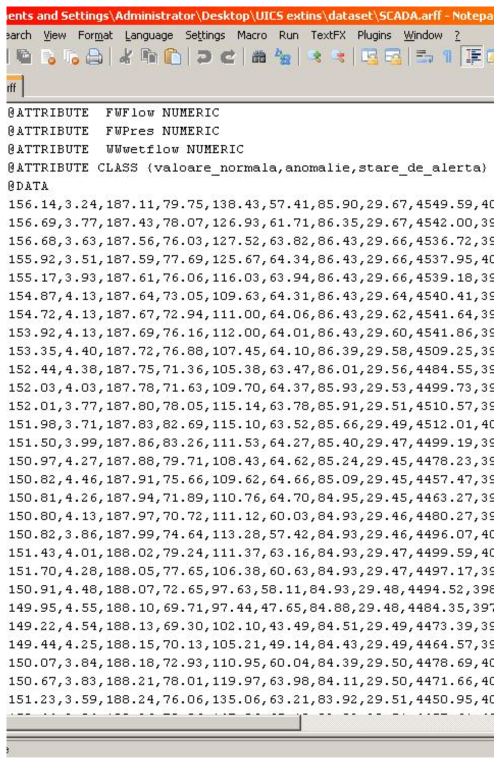
- Delete or replace missing values;
- Delete redundant properties (columns);
- Data Transformation;
- Data Discretization;
- Export data to a required .arff or .csv format file [11].

The original and modified formats of data set are shown in Figure 2 and Figure 3.

Data visualization is also a very useful technique because it helps to determine the difficulty of the learning problem. We visualized with Weka single attributes (1-d) and pairs of attributes (2-d). The figure 4 shows the variation of the temperature in time.

Feedwater Temp Inlet to	Flue Gas O2	Boiler Exit Fluer Gas Ter	Woo
156.139328	3.24443388	187.1074524	
156.6922302	3.773593664	187.4333954	
156.6774292	3.633202791	187.5584412	
155.9240112	3.51442337	187.585556	
155.1660919	3.925956726	187.6126709	
154.868927	4.125487804	187.6397858	
154.7238159	4.125487804	187.6669006	
153.9180756	4.133769035	187.6940155	
153.3471527	4.402714729	187.7211304	
152.4390717	4.383730412	187.7482452	
152.0284729	4.033183575	187.7753601	
152.0105743	3.76603508	187.802475	
151.9789734	3.710058212	187.8295898	
151.4999084	3.991338015	187.8567047	
150.9686279	4.266289234	187.8838196	
150.81604	4.464736938	187.9109344	
150.8099365	4.263769627	187.9380493	
150.803833	4.125068665	187.9651642	
150.8164063	3.861767769	187.9922791	
151.4267426	4.010254383	188.0193939	
151.7000732	4.281113625	188.0465088	
150.9064636	4.483417988	188.0736237	
149.9548187	4.550292969	188.1007385	
149.2224731	4.541767597	188.1278534	
149.4431915	4.251933098	188.1549683	
150.0674286	3.83922863	188.1820831	
150.6728973	3.82883811	188.209198	
151.2347107	3.590039492	188.2363129	
150.4357452	3.042577982	188.2634277	
151.5431519	2.841572046	188.2905426	
153.002243	2.709433317	188.3176575	
154.0943604	2.773037195	188.3447723	
154.4823456	2.581367016	188.3718872	
154.9624939	2.870390177	188.3990021	
155.2321167	2.688720703	188.4261169	

Figure 2: The Original Data Format




```
@ATTRIBUTE  FWFlow NUMERIC
@ATTRIBUTE  FWPres NUMERIC
@ATTRIBUTE  WWwetflow NUMERIC
@ATTRIBUTE CLASS {valoare_normala,anomalie,stare_de_alerta}
@DATA
156.14,3.24,187.11,79.75,138.43,57.41,85.90,29.67,4549.59,4
156.69,3.77,187.43,78.07,126.93,61.71,86.35,29.67,4542.00,3
156.68,3.63,187.56,76.03,127.52,63.82,86.43,29.66,4536.72,3
155.92,3.51,187.59,77.69,125.67,64.34,86.43,29.66,4537.95,4
155.17,3.93,187.61,76.06,116.03,63.94,86.43,29.66,4539.18,3
154.87,4.13,187.64,73.05,109.63,64.31,86.43,29.64,4540.41,3
154.72,4.13,187.67,72.94,111.00,64.06,86.43,29.62,4541.64,3
153.92,4.13,187.69,76.16,112.00,64.01,86.43,29.60,4541.86,3
153.35,4.40,187.72,76.88,107.45,64.10,86.39,29.58,4509.25,3
152.44,4.38,187.75,71.36,105.38,63.47,86.01,29.56,4484.55,3
152.03,4.03,187.78,71.63,109.70,64.37,85.93,29.53,4499.73,3
152.01,3.77,187.80,78.05,115.14,63.78,85.91,29.51,4510.57,3
151.98,3.71,187.83,82.69,115.10,63.52,85.66,29.49,4512.01,4
151.50,3.99,187.86,83.26,111.53,64.27,85.40,29.47,4499.19,3
150.97,4.27,187.88,79.71,108.43,64.62,85.24,29.45,4478.23,3
150.82,4.46,187.91,75.66,109.62,64.66,85.09,29.45,4457.47,3
150.81,4.26,187.94,71.89,110.76,64.70,84.95,29.45,4463.27,3
150.80,4.13,187.97,70.72,111.12,60.03,84.93,29.46,4480.27,3
150.82,3.86,187.99,74.64,113.28,57.42,84.93,29.46,4496.07,4
151.43,4.01,188.02,79.24,111.37,63.16,84.93,29.47,4499.59,4
151.70,4.28,188.05,77.65,106.38,60.63,84.93,29.47,4497.17,3
150.91,4.48,188.07,72.65,97.63,58.11,84.93,29.48,4494.52,39
149.95,4.55,188.10,69.71,97.44,47.65,84.88,29.48,4484.35,39
149.22,4.54,188.13,69.30,102.10,43.49,84.51,29.49,4473.39,3
149.44,4.25,188.15,70.13,105.21,49.14,84.43,29.49,4464.57,3
150.07,3.84,188.18,72.93,110.95,60.04,84.39,29.50,4478.69,4
150.67,3.83,188.21,78.01,119.97,63.98,84.11,29.50,4471.66,4
151.23,3.59,188.24,76.06,135.06,63.21,83.92,29.51,4450.95,4
```

Figure 3: The Modified Data Format

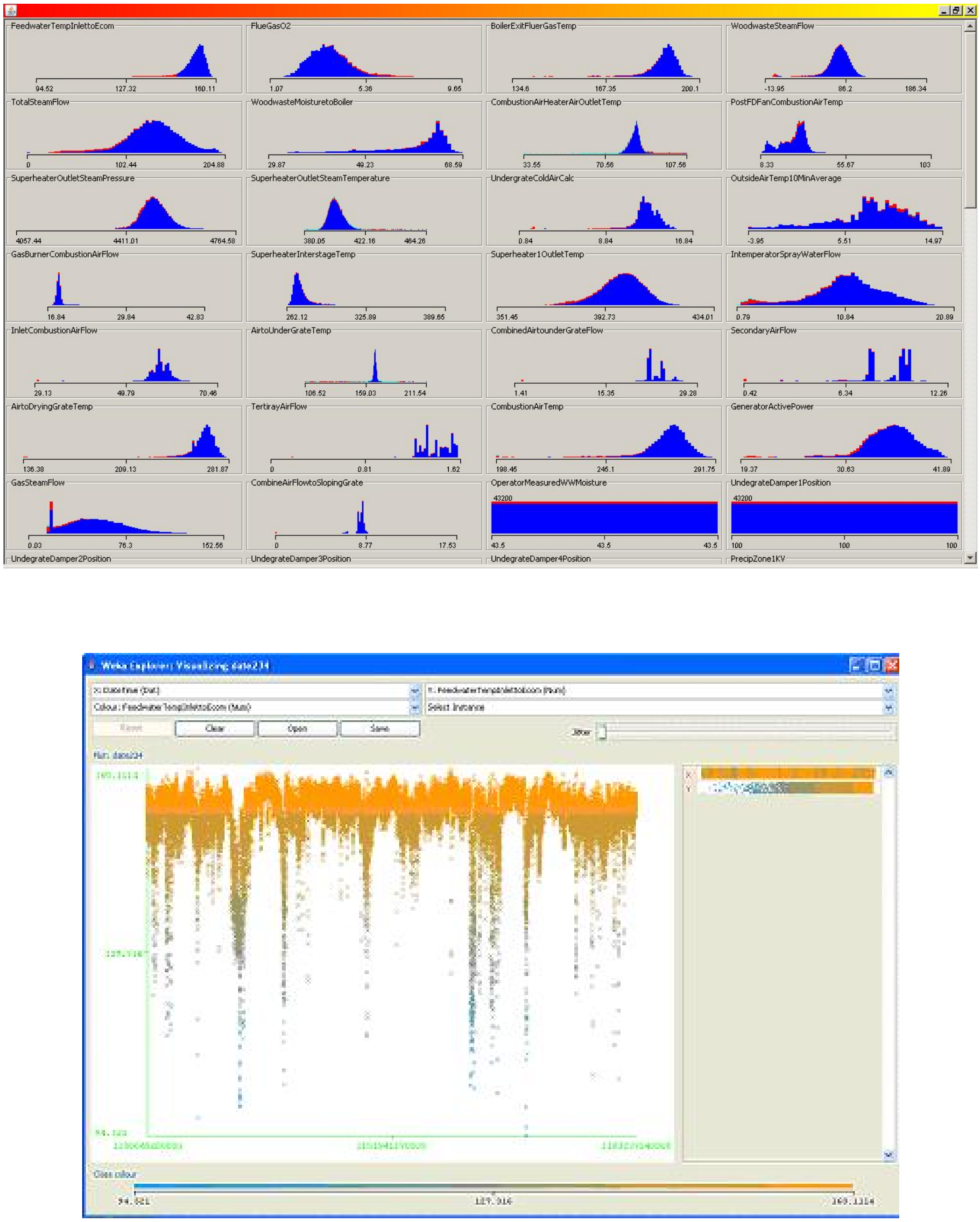


Figure 4: Data visualization

2.2.Data mining and interpretation of the results

A classification method was applied to assemble similar data points and to predict numeric quantities. In particular, we attempted to discover useful information and rules correlated to temperature values of the system in order to discard what could be regarded as irrelevant.

Based on the proposed framework, we chose *Naïve Bayes*, *% kNN* and *J48* algorithms to implement classification. We tried to obtain clear results by choosing a 20% split percentage, which means that about 20% records were used as test data in the pre-implemented training process before classification [11]. The classifiers will be evaluated on how well they predicted the percentage of the data held out for testing. We want to determine which classifier is suited for our data set. Running Nave Bayes algorithm in Weka is presented in Figure 5.

```
=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances        2608               90.5241 %
Incorrectly Classified Instances       273                9.4759 %
Kappa statistic                          0.589
Mean absolute error                      0.0632
Root mean squared error                  0.2476
Relative absolute error                 43.665  %
Root relative squared error             92.7555 %
Total Number of Instances             2881

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.953     0.186      0.974     0.953     0.964      0.936    valoare_normala
                 0.526     0.049      0.561     0.526     0.543      0.927    anomalie
                 0.744     0.029      0.259     0.744     0.384      0.955    stare_de_alerta
Weighted Avg.    0.905     0.169      0.921     0.905     0.911      0.935

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2418  116    2 |    a = valoare_normala
   64  161   81 |    b = anomalie
    0   10   29 |    c = stare_de_alerta
```

```
Correctly Classified Instances        2732               94.8611 %
Incorrectly Classified Instances       148                5.1389 %
Kappa statistic                          0.532
Mean absolute error                      0.0341
Root mean squared error                  0.1838
Relative absolute error                 68.7023 %
Root relative squared error            115.0133 %
Total Number of Instances             2880

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.96      0.07       0.997     0.96      0.978      0.969    valoare_normala
                 0.598     0.037      0.338     0.598     0.432      0.939    anomalie
                 0.889     0.013      0.387     0.889     0.539      0.996    stare_de_alerta
Weighted Avg.    0.949     0.069      0.971     0.949     0.958      0.969

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2656   99   11 |    a = valoare_normala
    8   52   27 |    b = anomalie
    0    3   24 |    c = stare_de_alerta
```

```
Correctly Classified Instances        2576               89.4755 %
Incorrectly Classified Instances       303               10.5245 %
Kappa statistic                          0.4762
Mean absolute error                      0.0695
Root mean squared error                  0.2608
Relative absolute error                 63.5695 %
Root relative squared error            113.309  %
Total Number of Instances             2879

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.924     0.238      0.976     0.924     0.949      0.932    valoare_normala
                 0.568     0.076      0.383     0.568     0.457      0.908    anomalie
                 0.769     0.014      0.328     0.769     0.46       0.992    stare_de_alerta
Weighted Avg.    0.895     0.223      0.925     0.895     0.907      0.931

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2430  197    4 |    a = valoare_normala
   59  126   37 |    b = anomalie
    0    6   20 |    c = stare_de_alerta
```

Figure 5: Running the Nave Bayes algorithm in Weka

The performance of the model was also evaluated by using split percentage technique and the results were presented as percentage of correctly classified instances (90,5241% for the first dataset, 96,8611% for the second dataset and 89,4755% for the third dataset) and incorrectly classified instances (9,4759%, 5,1389% and 10,5245%) and confusion matrix. After running the kNN algorithm in Weka on the same datasets we obtained the presented in figure 6.

```
Correctly Classified Instances        2787               96.7372 %
Incorrectly Classified Instances        94                3.2628 %
Kappa statistic                          0.848
Mean absolute error                      0.0219
Root mean squared error                  0.1475
Relative absolute error                 15.0983 %
Root relative squared error             55.2388 %
Total Number of Instances             2881

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.985     0.099     0.987      0.985     0.986      0.943    valoare_normala
                 0.856     0.019     0.84       0.856     0.848      0.918    anomalie
                 0.692     0.004     0.73       0.692     0.711      0.844    stare_de_alerta
Weighted Avg.    0.967     0.089     0.967      0.967     0.967      0.939

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2498   38    0 |    a = valoare_normala
   34  262   10 |    b = anomalie
    0   12   27 |    c = stare_de_alerta
```

```
Correctly Classified Instances        2847               98.8542 %
Incorrectly Classified Instances        33                1.1458 %
Kappa statistic                          0.8485
Mean absolute error                      0.0078
Root mean squared error                  0.0874
Relative absolute error                 15.6361 %
Root relative squared error             54.6866 %
Total Number of Instances             2880

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.995     0.14       0.994     0.995     0.995      0.927    valoare_normala
                 0.793     0.005      0.821     0.793     0.807      0.894    anomalie
                 0.926     0.001      0.926     0.926     0.926      0.963    stare_de_alerta
Weighted Avg.    0.989     0.135      0.988     0.989     0.988      0.927

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2753   13    0 |    a = valoare_normala
   16   69    2 |    b = anomalie
    0    2   25 |    c = stare_de_alerta
```

```
Correctly Classified Instances        2789               96.8739 %
Incorrectly Classified Instances        90                3.1261 %
Kappa statistic                          0.7976
Mean absolute error                      0.021
Root mean squared error                  0.1443
Relative absolute error                 19.1531 %
Root relative squared error             62.718  %
Total Number of Instances             2879

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.988     0.185      0.983     0.988     0.985      0.901    valoare_normala
                 0.766     0.014      0.817     0.766     0.791      0.876    anomalie
                 0.731     0.002      0.76      0.731     0.745      0.864    stare_de_alerta
Weighted Avg.    0.969     0.171      0.968     0.969     0.968      0.899

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2600   31    0 |    a = valoare_normala
   46  170    6 |    b = anomalie
    0    7   19 |    c = stare_de_alerta
```

Figure 6: Running the kNN algorithm in Weka

We observed that from the 20% of the instances representing the test set (2879, respectively 2880 and 2881 instances), 90 respectively, 33 and 94 instances of the three datasets were incorrectly classified (3,1261%, respectively 1,1458% and 3,2628% of instances). Figure 7 below shows three snap shots of a Run information in Weka for parameters values on June 2007 which used split percentage test mode for J48 classifier algorithm.

```
Correctly Classified Instances        2878               99.8959 %
Incorrectly Classified Instances         3                0.1041 %
Kappa statistic                          0.9951
Mean absolute error                      0.0008
Root mean squared error                  0.0263
Relative absolute error                  0.5196 %
Root relative squared error              9.8696 %
Total Number of Instances             2881

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        valoare_normala
                 1         0.001      0.99      1         0.995      0.999    anomalie
                 0.923     0          1         0.923     0.96       0.962    stare_de_alerta
Weighted Avg.    0.999     0          0.999     0.999     0.999      0.999

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2536    0    0 |    a = valoare_normala
    0  306    0 |    b = anomalie
    0    3   36 |    c = stare_de_alerta
```

```
Correctly Classified Instances        2879               99.9653 %
Incorrectly Classified Instances         1                0.0347 %
Kappa statistic                          0.9954
Mean absolute error                      0.0002
Root mean squared error                  0.0152
Relative absolute error                  0.4669 %
Root relative squared error              9.5209 %
Total Number of Instances             2880

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.009     1         1         1          0.996    valoare_normala
                 0.989     0         1         0.989     0.994      0.994    anomalie
                 1         0         1         1         1          1        stare_de_alerta
Weighted Avg.    1         0.008     1         1         1          0.996

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2766    0    0 |    a = valoare_normala
    1   86    0 |    b = anomalie
    0    0   27 |    c = stare_de_alerta
```

```
Correctly Classified Instances        2878               99.9653 %
Incorrectly Classified Instances         1                0.0347 %
Kappa statistic                          0.9978
Mean absolute error                      0.0003
Root mean squared error                  0.0152
Relative absolute error                  0.2608 %
Root relative squared error              6.6058 %
Total Number of Instances             2879

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0         1         1         1          1        valoare_normala
                 1         0         0.996     1         0.998      1        anomalie
                 1         0         1         1         1          1        stare_de_alerta
Weighted Avg.    1         0         1         1         1          1

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2630    1    0 |    a = valoare_normala
    0  222    0 |    b = anomalie
    0    0   26 |    c = stare_de_alerta
```

Figure 7: Run Information using J48 Tree Classifier Algorithm

From the "Classifier output" we found that just 1 value from the first dataset, 1 from the second one and 3 from the third one were incorrectly classified (0,0347% for the first and second dataset, and 0,1041 for the third dataset).

We concluded that *J48 Tree Classifier* model has a higher level of classification accuracy than the *Naïve Bayes Classifier* model, but the *IBk* algorithm is more adequate to our data set. The final results of the classification techniques are presented in the table below:

Dataset	**Classification accuracy**		
	Naïve Bayes Classifier(%)	IBk Classifier(%)	J48 Tree Classifier(%)
1-10.06.07	90,52	96,86	89,48
11-20.06.07	96,74	98,85	96,87
21-30.06.07	99,90	99,97	99,97
AVERAGE	**95,72**	**98,56**	**95,44**

Table 2. The accuracy of the classification methods

3. Conclusions and future works

Classifier performance evaluation is an important stage in developing data mining techniques.

Our goal was to find the classifier that is suitable to the data set provided by SCADA system. The highest level of accuracy was matched in *IBk Classifier*. The three classes obtained after running the model allows a better optimization of the transmitted data traffic and of the necessary data storing space and projects the large amount of data to a lower dimensional space.

On the data acquisition system level we can program the transmission of warning and anomaly values and discarding normal values.

A future approach consists in a high sampling rate of data transmission from the three classes.

We also propose to develop one program that makes difference between acquisition system level and local storage of the functioning modes.

References

[1] J. Quinlan. Boosting first-order learning. Proceedings of the 7th International Workshop on Algorithmic Learning Theory, 1160:143–155, 1996.

[2] C. Nadal, R. Legault, and C. Y. Suen. Complementary algorithms for the recognition of totally uncontrained handwritten numerals. In Proceedings of the 10th International Conference on Pattern Recognition, volume A, pages 434–449, June 1990.

[3] Hand, DJ, & Yu, K. (2001). "Idiot's Bayes - not so stupid after all¿' International Statistical Review. Vol 69 part 3, pages 385-399. ISSN 0306-7734.

[4] J. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.

[5] http://www.dayton-knight.com/Projects/SCADA/scada_explained.htm

[6] http://www.bin95.com/certificate_program_online/control-systems-technology.htm

[7] I. Stoian, T. Sanislav, D. Căpăţînã, L. Miclea, H. Vălean, S. Enyedi, "Multi-agent and Intelligent Agents' Techniques Implemented in a Control and Supervisory Telematic System" , 2006 IEEE International Conference on Automation, Quality and Testing, Cluj-Napoca, 25-28 May 2006, pp. 463-468.

[8] E. K. Cetinkaya, "Reliability analysis of SCADA Systems used in the offshore oil and gas industry" , 2001.

[9] S. Wang, "Research on a New Effective Data Mining Method Based on Neural Networks" , 2008 International Symposium on Electronic Commerce and Security, Guangzhou City, 3-5 Aug. 2008, pp.195-198.

[10] B. Zheng, J. Chen, S. Xia, Y. Jin, "Data Analysis of Vessel Traffic Flow Using Clustering Algorithms" , 2008 International Conference on Intelligent Computation Technology and Automation, pp. 243-246.

[11] G. Wang, C. Zhang, L. Huang, "A Study of Classification Algorithm for Data Mining Based on Hybrid Intelligent Systems" , Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 371-375.

[12] P. Du, X. Ding, "The Application of Decision Tree in Gender Classification" , 2008 Congress on Image and Signal Processing, pp. 657-660.

[13] S. B. Shamsuddin, M. E. Woodward, "Applying Knowledge Discovery in Database Techniques in Modeling Packet Header Anomaly Intrusion

Detection Systems" , Journal of Software, vol.3, no. 9, December 2008, pp. 68-76.

[14] Zengchang Qin, "Naive Bayes Classification Given Probability Estimation Trees" , Proceedings of the 5th International Conference on Machine Learning and Applications, 2006.

Maria Muntean, Ioan Ileană, Corina Rotar, Mircea Rîşteiu
Department of Mathematics and Informatics
"1 Decembrie 1918" University of Alba Iulia
email: *maria_munt2006@yahoo.com*

Honoriu Valean
Automation Department,
Technical University of Cluj Napoca,
Romania
e-mail: *Honoriu.Valean@aut.utcluj.ro*