Abstract- The huge amounts of data required by Smart grids operation are impossible to be processed by human operators in a timely manner. New intelligent systems should provide a clear decision regarding the system state. This paper proposes a new methodology based on supervised learning using AdaBoost and CARTs as decision support system for power system state classification. The methodology proves to be time efficient and precise, with low false negative rates. This approach could help in Smart Grids design and deployment, as it could be easily integrated into the existing EMS/SCADA systems.

Index Terms-- Boosting, Decision support systems, Intelligent systems, Power system analysis computing, SCADA systems, Smart grids, Supervised learning.

I. INTRODUCTION

The evolution of power grids towards “smarter” ones is bringing after itself an increase in their complexity, both from the operational point of view and regarding the way they are managed.

New Intelligent Electronic Devices (IEDs) are being installed on expanding scales. Technological advances enable these IEDs to acquire and transmit very large amounts of data. However, data from IEDs have to be processed and analyzed quickly, so that human operators or automated self-healing structures can make decisions and act in due time.

The advantages of Smart Grids are tremendous, with function such as demand response, smart pricing or plug-in hybrid vehicles, but they also require an extensive and reliable communication and data processing infrastructure. Smart Grids should have data processing and analysis components able to handle the large amounts of on-line operational data (measurements and control) and also historical data for better equipment maintenance scheduling and load forecast.

The system changes over time, so data collected on-line have to be assessed in a suitable manner. Moreover, power system data consists of both analog (eg. voltages, currents, power flows) and digital signals (eg. status of circuit breakers, flags coding the operation of relays). But decision making is not done based on data. Smart Grids have to be able to transforme data subsequently into information and then into knowledge, in a timely manner. Therefore, instruments that use data as inputs and transform it internally into information, output only aggregated state reports that help operators understand the current status of the network.

The volumes of data gathered and sent in the communication infrastructure enable electrical networks to evolve to “smart”, but data could also become a threat to Smart Grids reliability, as they can become overwhelming. Making automated decisions within smart grids could represent a solution in processing power system data in appropriate time, as human operators do not have the ability to disseminate large amounts of data. Moreover, in smart grids data should be selected in an intelligent manner and sent in due time to users who need them, at different decision levels.

The decision making processes in power systems can employ several types of aggregated data, from topology analysis, bad data rejection and alarm processing to system operation state identification, most of these applications relying on classification tasks.

Applications concerning data processing in power systems have been approached so far by means of artificial intelligence techniques, like expert systems and neural networks [1] and more recently, rough sets [2].

The classification of the system state can be done either into two classes (safe/unsafe) or into multiple classes (for example, normal, alert, emergency and safe). [3]

Whatever the case, the classification technique has to process and summarize, and most importantly, to identify key information about the system operating conditions.

This approach would lead to a relief of the human operator, who would no longer have to cope with large amounts of data, some of which inconsistent, or with avalanches of alarms.

This paper proposes a new approach in power system classification tasks, based on machine learning with Boosting – namely the AdaBoost algorithm. This is outlined in Section II. Section III presents the proposed methodology and IV presents a short exemplification. Section V contains some concluding remarks.

II. ADABOOST AND VARIANTS

The concept of Boosting supervised learning has its origins in a popular machine learning technique, probably approximately correct learning (PAC learning) [4] and it is based on the idea that a set of weak learners can be combined so as to generate a single strong learner. Boosting can therefore be performed in classification learning tasks.

The first Boosting algorithm with provable polynomial time was proposed by Schapire [5]. Kearns and Vailant proved that learners performing only slightly better than random (weak learners that can classify examples better than random guessing) can be combined to form an ensemble hypothesis.
(or learner), provided there is sufficient data available [6]. The latter one is said to be a strong learner, i.e. it is arbitrarily correlated to the correct classification.

Let $H_1, H_2, \ldots, H_n$ be a set of hypotheses. The composite hypothesis can be written as:

$$F(x) = \sum_{i=1}^{n} \alpha_i H_i$$

where $\alpha_i$ is the weighting coefficient of hypothesis $H_i$ (learner) [5].

Both parameters from (1) have to be set by the Boosting procedure. The main variation between many boosting algorithms is their method of weighting training data points and hypotheses. Among these variations, the AdaBoost (Adaptive Boosting) is known as the first step of Boosting applications in real-world problems [7].

Some of these include power systems applications, but there is very little literature on the subject. As it seems, only two applications of Boosting in power systems have been proposed prior to this paper. The first was a non-intrusive technique for monitoring the power consumption of various household appliances [8]. The other one concerns power system security assessment and uses AdaBoost [9].

The pseudocode of a generalized AdaBoost algorithm, proposed by Schapire and Singer is given in [10]. More detailed information about AdaBoost can be found in [11].

Several variants of the algorithm have been proposed and tested. In the following, we will only briefly present three of them, which were used during studies.

A. Real AdaBoost

The Real AdaBoost, optimizes $E[\exp(-y(H(x) + H_m(x)))]$, as function of $H_m(x)$ and it is given in Figure 1.

B. Gentle AdaBoost

The Gentle AdaBoost algorithm uses weighted least-squares regression to minimize the function $E[\exp(-yH(x))]$. It is an improvement to the Real AdaBoost, by using Newton stepping, providing a more reliable and stable ensemble. The pseudocode for the Gentle AdaBoost is given in Figure 2.

1. Initialize the weights $w_i = 1/N, \ i \in \{1, \ldots, N\}$
2. For $m=1$ to $M$
   a) Train $H_m(x)$ by weighted least-squares of $y_i$ to $x_i$, with weights $w_i$
   b) Update $H(x) = H(x) + H_m(x)$
   c) Update $w_i \leftarrow w_i \exp(-y_i H_m(x_i))$ and renormalize to $\sum w_i = 1$
3. Output $H(x) = \text{sign}\left(\sum_{m=1}^{M} H_m(x)\right)$

Fig. 2. The Gentle AdaBoost [12].

C. Modest AdaBoost

The Modest AdaBoost algorithm, given in Figure 3, has been demonstrated to have better generalization error, compared to the previous variants presented here, but has a higher training error.

1. Initialize the weights $w_i = 1/N, \ i \in \{1, \ldots, N\}$
2. For $m=1$ to $M$ and while $H_m \neq 0$
   a) Train $H_m(x)$ by weighted least-squares of $y_i$ to $x_i$, with weights $w_i$
   b) Compute “inverted” distribution $\overline{w}_i = (1 - w_i)$, and renormalize to $\sum \overline{w}_i = 1$
   c) Compute:
      \begin{align*}
      P_m^{-1}[y = +1, H_m(x)] = P_m(y = +1, H_m(x)) \\
      P_m^{-1}[y = -1, H_m(x)] = P_m(y = -1, H_m(x)) \\
      P_m^{-1}[y = +1] = P_m(y = +1, H_m(x)) \\
      P_m^{-1}[y = -1] = P_m(y = -1, H_m(x))
      \end{align*}
   d) Set $H_m(x) = (P_m^{-1}[y = +1] - P_m^{-1}[y = -1])$
   e) Update $w_i \leftarrow w_i \exp(-y_i H_m(x_i))$
      and renormalize to $\sum w_i = 1$
3. Output $H(x) = \text{sign}\left(\sum_{m=1}^{M} H_m(x)\right)$

Fig. 3. Modest AdaBoost [13].

The update in step 2.d decreases a weak classifier’s contribution if it performs “too well” on data that was previously classified correctly with high margin.

The impressive generalization capabilities of boosting are due to the classifier having large margins on the training data.

The three variants briefly described above were implemented as part of the methodology proposed in Section III. In our studies, classification and regression trees (CART) were used as weak learners.

III. PROPOSED METHODOLOGY

The methodology proposed here, addressed in the following as SMA_RT, aims at providing the power system operator a quick decision about the system state. As mentioned in Section I, this could reduce the impact of faults over the network, provided it is timely and reliable.

The core of the methodology is the AdaBoost learning module, which uses CART’s as weak learners. The system is
provided with a snapshot from the state estimator and it uses the classifier to output the system state. The current implementation of SMA_RT only performs bi-dimensional classification (“normal” or “unsafe”) and it was built with sole purpose of demonstrating AdaBoost capabilities in Smart Grid applications. The module can be replaced by any other AdaBoost variant, for example AdaBoost.M1 [14], which was designed for multi-class tasks.

The functional structure of SMA_RT is given in figure 4. The inputs of the decision system (power flows, voltages, network topology etc.) are received from the state estimator, which detects and rejects bad measurement data and outputs a snapshot of the current system state by estimating unknown field values.

The system is comprised of two major components that work in parallel: an active component that gives the classification verdict of the system state and displays the results and a separate thread that automatically activates on a timed basis, which updates the classifier model accordingly to previous states of the monitored power system.

The snapshot is taken from the working memory and the current system attributes are inputted into the classifier, whose output is displayed on the human operator console. In the same time, the current attributes are stored in the long-term memory where they are accessible to the update module. From time to time (for example every six hours), the update module is automatically activated. The current classifier is replicated into a temporary memory and the one in the working memory is modified using AdaBoost with CARTs by learning the examples in the database. After the learning
process, the new classifier model is tested. If it performs better than the previous one, it is stored and will be used for the classification tasks. Otherwise, the replicated model is reinstated.

IV. CASE STUDY

In order to test the performances of AdaBoost with CARTs in the classification of power system states, the three AdaBoost variants described in section II were tested on IEEE 39-bus network.

The precision, false positive, false negative rates and running time were computed for each variant. It should be noted that false positive results, i.e. false alarms, may be acceptable in power system control as long as they do not waste resources (for example trying to discover the causes of an inexistent fault), but false negative results may lead to faults, as unsafe states are classified as safe. This is why, in general, new classification methods seek to minimize the false negative rates. The results of the three variants are given in the following.

The test system contains 10 generators, 19 loads, 36 branches and 12 transformers. The characteristics publicly available at [15,16].

The database used for testing comprised 434 examples, whereas the testing one contained 97 examples. Each example has 153 attributes (voltages, real and reactive powers etc.). The performances of each variant are presented in tables I and II for the training and, respectively, for testing of the proposed methodology.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Modest AdaBoost</th>
<th>Gentle AdaBoost</th>
<th>Real AdaBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy [%]</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Mean square error</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>False negative rate [%]</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>False positive rate [%]</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Computational time [s]</td>
<td>0.502</td>
<td>0.502</td>
<td>0.502</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Modest AdaBoost</th>
<th>Gentle AdaBoost</th>
<th>Real AdaBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy [%]</td>
<td>97.79</td>
<td>97.07</td>
<td>97.87</td>
</tr>
<tr>
<td>Mean square error</td>
<td>0.06</td>
<td>0.08</td>
<td>0.05</td>
</tr>
<tr>
<td>False negative rate [%]</td>
<td>2.2</td>
<td>2.9</td>
<td>2.1</td>
</tr>
<tr>
<td>False positive rate [%]</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Computational time [s]</td>
<td>0.0156</td>
<td>0.0156</td>
<td>0.0156</td>
</tr>
</tbody>
</table>

Figure 5 represents the comparative analysis of the false positive rates and the accuracy of the three variants during the testing stage.
V. CONCLUSIONS

Even though today’s electrical networks do not fully comply with the requirements of Smart Grids, it is expected that, in a foreseeable future, most of them will become “smarter”. There are several working groups spread around the world that study intelligent electrical networks from various points of view, like, for example, SmartGrid, Intelligrid, Grid2030, Intelligent Utility Network and others. All of them define Smart Grids as the electrical transmission and distribution networks which, by incorporating information technology and communication capabilities, become able to predict, self-heal and self-adjust to changes.

In this context, incorporating ICT and power networks infrastructures into one interconnected architecture is a promising foundation for better automated control – a feature required but not sufficient in order to develop Smart Grids. The standardization in communication protocols for Smart Grids is also imperative in order to enable interoperability through network components (be it hardware or software, power installations, communication or data processing devices). Moreover, this integration of the two, initially separated, infrastructures provides real-time capabilities for both network control/operation and power markets. Whatever the case, timely information and in a discernible format for its receiver (control devices or human operators) is the milestone for achieving Smart Grid functionalities.

A new methodology based on supervised learning with AdaBoost and CARTs as weak learners was proposed in this paper for power system state classification. This classification, in due time and with high accuracy, could help human operators avoid faults. Moreover, after proving its reliability, the method could be implemented into closed-loop control routines in order to autonomously overcome system faults. The methodology, in its current development, is far from achieving this goal, but it is clear that it is suitable for such applications and more attention should be given to state of the art machine learning applications in power systems.

All in all, it is our belief that Boosting and machine learning could provide the support needed to develop the functionalities and performances required by a power grid to become a Smart Grid.

REFERENCES