14th IMEKO TC10 Workshop Technical Diagnostics
New Perspectives in Measurements, Tools and Techniques
for system's reliability, maintainability and safety
Milan, Italy, June 27-28, 2016

Rules induction-supported random forest for the non-intrusive electrical appliances identification

Piotr Bilski[1], Wiesław Winiecki[1]

[1] *Institute of Radioelectronics, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, pbilski@ire.pw.edu.pl, W.Winiecki@ire.pw.edu.pl*

Abstract **– The paper presents the application of computational intelligence methods to the non-intrusive identification of electrical appliances in the household. The difficulty of the problem calls for implementation more than one algorithm. The presented two-staged methodology monitors the aggregated current level and includes the detection of the change in the state of appliances' configuration with the subsequent identification of the device responsible for it. The latter uses the random forest as the main classifier and the rules induction algorithm as the supporting method to increase the appliance identification accuracy. The tests conducted on the configuration of two-state appliances show the supremacy of such a combination of classifiers.**

I. INTRODUCTION

The increasing demand for suppressing the energy consumption in the typical households requires applications of sophisticated computing algorithms. The aim of the developed countries is to minimize the usage of energy in industrial installations and private households, responsible for the most of the energy consumption in the world [1].

Multiple approaches to create the energy consumption profiles of typical households were made during the recent years. They mainly include statistical and Artificial Intelligence (AI) methods, processing various features extracted from the current waveform. The most popular are Artificial Neural Networks (ANN) [2], Hidden Markov Models (HMM) [3] and dynamic programming [4]. The implemented approaches rely on various ranges of monitored frequencies, starting from single Hertz [1], through a couple of kHz [5], up to hundreds of kHz or even 1 MHz, based on the analysis of electromagnetic interference (EMI) related to the work regime of appliances [6]. To process the measurement data and conclude about the characteristics of particular appliances (average power consumption, or duration of operation) the information about the behavior of every device must be first collected, which is currently the main problem of the Non-Intrusive load Appliance Monitoring (NIALM). This methodology is aimed at discovering, which electrical appliances currently operate in the household based on the aggregated current and voltage patterns, collected by the module located near the energy meter. Currently, data sets from selected locations in Canada or United States are available [7], but they are limited to characteristics of electrical networks in these countries.

The paper presents the application of the Random Forest (RF) to identify appliances in the NIALM scheme based on the measurements taken in the medium range of frequencies. This classification approach requires the dedicated Data Acquisition (DAQ) hardware, able to collect the desired signals, from which features for the identification are extracted. The RF was identified as one of the most accurate algorithms for the task, but it is not flawless. To increase the identification accuracy, it must be supported by the second method. Even giving worse results in general, this additional algorithm may improve the overall efficiency of the NIALM system. The Rules Induction (RI) approach was used for this purpose. The implemented solution proves to be more flexible than any separate classifier, at the price of higher (but acceptable even for the embedded systems) computational cost.

The paper structure is as follows. In section II, the general architecture of the proposed NIALM methodology is presented. Here the principles of the appliances identification and features used for this purpose are discussed. Section III presents the idea of introducing the intelligent classifier to the task, which requires the machine learning algorithm to extract knowledge from the training data. In Section IV, the RF is introduced, with the parameters influencing its efficiency. Section V presents the supporting classifier, based on the rules induction algorithm. In Section VI, experimental environment (i.e. the laboratory household with selected appliances to distinguish) is described. Section VII contains experimental results of the combined approaches. In section VIII conclusions and future prospects of the proposed solution are presented.

II. THE NIALM SYSTEM ARCHITECTURE

The appliance identification system (Fig. 1) consists of the hardware and software part. The former is the DAQ module, located close to the location of the household current aggregation, and the computing unit, making the decision about the current configuration of appliances operating in the apartment. The DAQ module is responsible for probing the signal. In the presented methodology the minimum required sampling frequency is 2kHz, allowing for extracting all features needed to identify appliances.

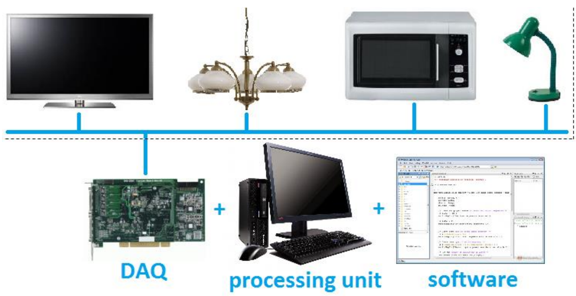

Fig. 1. Architecture of the NIALM system.

To ensure the correct identification, two stages are required in the software part. The first one is used to monitor the current level on-line and detect its abrupt changes, caused by the changed state of the selected appliance (for instance, because it was turned on or off). Subsequently, the identification algorithm is executed to determine, which device was responsible for this event.

The proposed methodology fits the wide range of NIALM applications, which include detection of operating appliances, estimating their power consumption, or modelling the behavior of the apartment residents. The appliance identification can be also used to detect the devices with malfunctions as long it influences their operation aspects visible in the voltage and current signals. This leads to creating the additional subcategories of the appliance, representing its states (the nominal and faulty ones). To detect faulty appliances, either models or actual devices must be verified.

Among four categories of appliances' work regime [8], the proposed architecture is suited to work with two of them: two state (turned on or off, like a kettle or lightbulb), and finite state machines (depending on the program, working in multiple modes, such as the washing machine, or the microwave). The presented research shows the efficiency of the architecture for the first group of devices. The monitoring system must be trained for every apartment separately.

The event detection is based on the analysis of the value of the current. Because the monitored patterns are sinusoidal, the searched change in the signal level is calculated based on the maximum of the thousand samples vector. The change event is checked twice a second, which is enough, assuming that no two appliances change their state during this time. Knowledge about all appliances operating in the household allows for representing then in the binary vector, where "1" means the device is turned on, and "0" is for the device switched off. For instance, in the six-appliance apartment the vector would have the following form:

$$\{0,1,0,0,1,1\}$$

The change in the device configuration would then be represented by the difference between two neighboring vectors, with "1" only on the position indicating the device changing the state since the previous event.

Turning the device on sometimes causes the spike in the current pattern, after which the signal level returns to the stable, new value (Fig. 2). The identification algorithm is executed after this transient state. The parameter of the event detection procedure is the minimum change in the current level triggering the appliance identification module. Too small values lead to the "false alarms", triggering the identification procedure too often. The threshold too high would mean missing some changes. Therefore the first solution, although more computationally demanding, is preferred. The selected threshold level was set to 100 mA, which is enough for most appliances (except some power-saving ones).

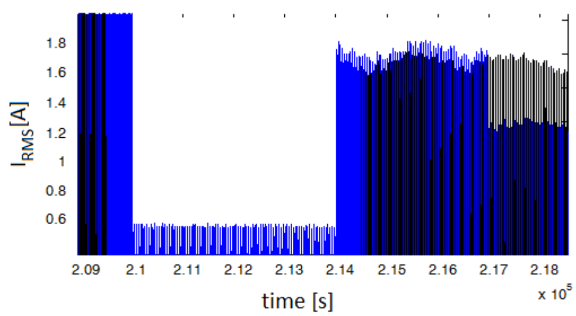

Fig. 2. State change of the appliance in the current pattern.

The second step of the software part requires the set of features extracted from the current waveform. Such a vector is calculated after detecting every event (during the on-line monitoring). Finally, the classifier is executed to determine the appliance responsible for the event.

III. ARTIFICIAL INTELLIGENCE CLASSIFICATION

The aim of the applied algorithms is to identify the particular appliance with knowledge gained during the training, based on features extracted from the waveform. This way the information about the behavior of each appliance is obtained. To train the classifier, the data set $\boldsymbol{L}$ must be prepared, thanks to experiments with each appliance to be identified. The set contains the number of feature vectors $\boldsymbol{f}_i$ for every device, supplemented by the integer number c_i, i.e. the appliance identifier. The pair $(\boldsymbol{f}_i, c_i)$, further called the example $\boldsymbol{e}_i$, is created by monitoring the selected appliance for some time, with all other devices shut down. During the process, samples' vectors are repeatedly acquired and examples generated. This way the single device is represented by multiple instances of its typical work regime characteristics. In our research each of six appliances was represented by 100 examples, with additional 100 for the “false alarm”, when no device changed its state.

$$\boldsymbol{L} = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} f_1 & c_1 \\ \vdots & \\ f_n & c_n \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1m} & c_1 \\ \vdots & \ddots & & \vdots \\ a_{n1} & \cdots & a_{nm} & c_n \end{bmatrix} \quad (1)$$

The trained classifier is used in the second step of the software module, triggered by the event detection. The sequence $\boldsymbol{S}$ of events is used as the testing data, verifying the accuracy c_a of the implemented approach, defined as:

$$c_a(\boldsymbol{S}) = \frac{|f_i : d(f_i) = c(f_i)|}{|\boldsymbol{S}|} \quad (2)$$

where $d(\boldsymbol{f}_i)$ is the identification of appliance made by the classifier based on the feature vector $\boldsymbol{f}_i$, $c(\boldsymbol{f}_i)$ is the actual identifier of the appliance related to the event and $|\boldsymbol{S}|$ is the number of events in the sequence to identify.

Because during the on-line operation multiple appliances work simultaneously, the module identifies only the device recently changing its state. The sequence of correct identifications allows for determining, which appliances worked in the household in the particular order. The features extracted after the event detection are calculated for the group of currently operating appliances and cannot be used instantly for the identification. Instead, the system also stores in the memory the set of features prior to the analyzed event. This way it is possible to calculate the features for the particular appliance as the difference between two vectors, calculated after the previous $\boldsymbol{f}(t-1)$ and the last event $\boldsymbol{f}(t)$:

$$\boldsymbol{f}_i(t) = \boldsymbol{f}(t) - \boldsymbol{f}(t-1) \quad (3)$$

To calculate the vector $\boldsymbol{f}_i(t)$ for the *i-th* appliance (which identifier is yet unknown), the features must be additive. Multiple sets are considered (mainly focused on the analysis of harmonic components of the 50Hz signal) [8]. The following time and frequency attributes were extracted from the current and voltage waveforms:

- Amplitudes of the first sixteen harmonic components of the current
- Phase shifts of the first sixteen harmonic components of the current
- Root Mean Square (RMS) of the current
- Mean value of the current
- Maximum current value
- Value of the DC component
- Mean power (calculated based on the voltage and current waveforms)
- Values of the active power in the first sixteen harmonics
- Values of the reactive power in the first sixteen harmonics

They were calculated on the 1024 samples vectors. The set of overall 67 features (real numbers) is calculated for the training set and during the on-line operation.

The considered AI algorithms applied in this work included the rule-based approaches (such as the Rules Induction – RI, Decision Tree – DT and Random Forest - RF). The knowledge is represented in the form:

$$\textit{IF premises THEN conclusions} \quad (4)$$

where *premises* is the conjunction of conditions to meet by the analyzed set of features assigned to the particular category (defined by *conclusions*). The conditions for the continuous features are of three types:

$$a_j > \theta \quad (5a)$$

$$a_j < \theta \quad (5b)$$

$$a_j \in (\theta_1, \theta_2) \quad (5c)$$

The thresholds θ, θ_1 and θ_2 indicate the value to compare with the *j-th* feature’s value.

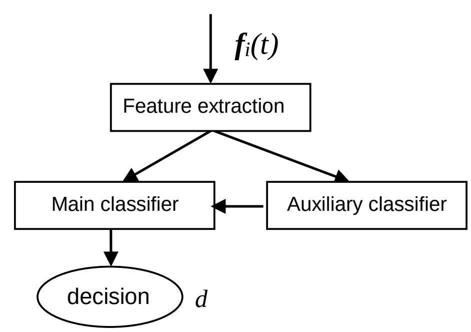

Fig. 3. Architecture of the decision making module.

In contrast to statistical and numerical methods, knowledge extracted during the training has the form readable by humans and may be modified. The greatest accuracy was obtained so far for RF considered the optimal classifier in the uncertainty conditions and successfully used in a number of applications (such as technical diagnostics [9]). Analysis of errors made by all algorithms showed that each is wrong in different cases. Therefore combining them would increase the system accuracy. The classifier fusion may be used for this purpose, but it is the most computationally expensive and requires tuning for each apartment separately [10]. Therefore the simpler approach is also proposed (Fig. 3), where two methods work in parallel. In the following subsections both classification method are introduced.

A. Random Forest

This is the DT-based classifier of the increasing popularity. Its advantages include the legible form of the stored knowledge and the ability to work with uncertain data (suffering from the additive noise and limited measurement accuracy). During the training, multiple trees (differing in structure) are constructed on the same data set ***L***. Every DT starts with the root, connected to nodes of the lower level (offsprings), which in turn are also connected to their offsprings, until the lowest level (with leaves) is reached. Every node contains a test of the form (5a) or (5b). The leaf contains the category (identifier of the appliance). Making decision by the DT is made by going from the root to one of leaves, according to the tests' results in the visited nodes. This way the tree structure is considered as the set of rules (4) where the premises part implements the sequence of visited nodes, while the conclusion is the leaf with the device identifier. Only one rule may be active at the time.

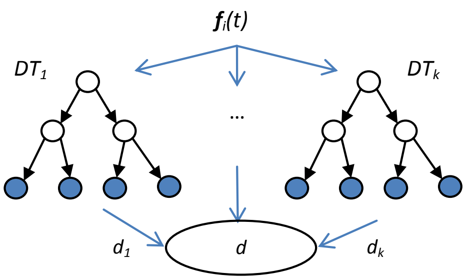

Fig. 4. Random forest structure.

The generation of every tree in the RF consists in creating subsequent nodes and selecting the tests for them to separate examples belonging to different categories as quickly as possible. The typical method of selecting the feature and its value for the test is to use the minimum entropy criterion. In the RF the test is randomly selected from the set of *h* candidates with the smallest entropy values. Therefore each DT is different, although generated on the same set ***L***. This enables generalization and avoids overlearning. Another parameter is the number of generated trees *k*. The compromise between the value of *k* and the processing time must be made.

The operation of the RF (Fig. 4) relies on using every tree to process the set of feature and produce *k* decisions, which then take part in voting to produce the overall decision *d*. Among multiple voting mechanisms, the simplest one was used, where every tree has equal weight. Therefore the appliance identifier pointed by the maximum number of trees is the response of the RF. To avoid a draw, when two equally numerous groups of trees support the same category, the value of *k* is odd.

B. Rules Induction

The implemented method is the authors' version of the AQ algorithm, originally used to create rules for discrete data sets. The proposed modifications include adjustment to the continuous data, which required changing the structure of premises into the form as in (5).

The RI [11] uses the sequential covering scheme, creating at least one rule to cover every example. The rule must be fully accurate, covering only examples from ***L*** belonging to the single category. In each step the new example, not covered so far, is selected and combinations of conditions like in (4) are generated to cover it and as many examples of the same appliance as possible. Initially, the set of conditions is universal, covering all examples from ***L***. It is subsequently specialized to exclude examples describing other devices, until none of them are covered by the premises. Because multiple sets of conditions are generated in every iteration, the criteria to select only some of them are introduced. They are the number of the candidates for the premise (α) and the significance of covering by the rule (i.e. the number of potential examples that can be covered by it) $\beta \in (0, 1)$. Also, to avoid the overlearning, the thresholds separating values too close to each other are not considered.

Contrary to RF, the induced rules are independent of each other and may have different complexity (i.e. the number of conditions in the premises part of (4)). This makes possible activating (firing) multiple rules at the same time for the analyzed vector of features, but also firing no rule at all. The RI is time consuming and demands large amount of memory to generate rules.

IV. EXPERIMENTAL RESULTS

The proposed scheme was tested on the configuration of selected appliances working in the isolated environment. Six two state devices were considered (brackets contain the appliance identifiers): power-saving lightbulb ("1"), dryer ("2"), vacuum cleaner ("3"), mixer ("4"), juicer ("5") and kettle ("6"). They were turned on and off in various sequences, during which the DAQ module was acquiring the current and voltage waveforms. The hardware responsible for the signal acquisition was the PC with the Advantech PCIe 1816 card, characterized by 16-bit resolution and 2MHz of sampling frequency. Fig. 5 presents the example of the 30 minutes sequence (lower part) with the measured current (upper part).

Several sequences (each lasting for 30 minutes) as in Fig. 5 were recorded. They were then processed by the event detection procedure and both classifiers to determine their abilities in the appliance identification. AI algorithms were adjusted to maximize their accuracy. Next, the strategy of their combined application was proposed to increase the classification outcome.

A. Sequence processing

Examples of identification results for the selected events (out of 84 overall detected) is in Table 1. The values in every row are identifiers of particular appliances pointed by the classifiers. The value "0" means the detected event was not caused by any device (false alarm). The actual value "5+6" means that the event detection module missed one turning on of the appliance (therefore the change related to the previous event includes two additional appliances that changed their state). The value "-1" means the rule-based method was unable to activate any rule cover the calculated features, making no decision.

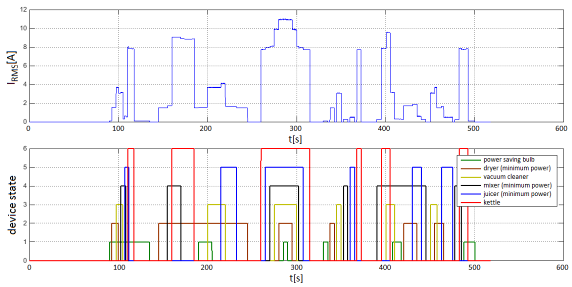

Fig. 5. Example of the appliances' operation sequence.

The analysis of the classifiers' operation shows their following advantages and drawbacks:

- The RI has the greatest problems with detecting the "false alarm" event (when no appliance in fact changed its state). In the fifty percent cases no rule or the incorrect one was fired for the actual event encoded with "0". During the training the greatest number of rules was generated for this state, suggesting this category is the most difficult.
- The RF has the highest accuracy (for instance, 89.28% of all identifications in the presented sequence), therefore it is the most reliable and its responses are more important during the voting.

The most difficult to detect is the event related to the low-power device, such as the power-saving bulb. In this case both classifiers identify only the less energy efficient appliance (such as is in the event No. 65 in Table 1).

Table 1. Example of appliance identification sequence.

Event No.	**RF**	**AQ**	**Actual**
13	5	6	**5+6**
14	1	1	**1**
15	2	2	**2**
16	4	*-1*	**4**
18	4	*-1*	**4**
19	6	6	**6**
20	*3*	*3*	**4**
21	0	0	**0**
47	5	6	**5+6**
65	3	3	**3+1**

B. Comparison of classifiers

The efficiency of both approaches was tested and compared. First, the RF of various sizes *k* was generated. Because the structure of every DT in the forest is different, the experiment was repeated ten times. The accuracy results in Table 2 are mean values μ, additionally the standard deviations σ are given to illustrate their repeatability. The highest score for the single forest was 89.28% for 11 trees. In the mean sense the best is the RF with 11 or 13 trees (depending on the analyzed sequence). The optimal results are for the highest score with the smallest deviation. Training of the classifier requires repeating the process several times for the predetermined size of the forest and keeping the best configuration for tests with other sequences.

Table 2. Classification accuracy of RF for various sizes.

k	$\mu(c_a)$	$\sigma(c_a)$
3	79,16	2,47
5	84,52	3,19
7	84,32	2,43
9	86,90	1,30
11	86,11	1,62
13	87,69	0,97
15	86,50	1,79

The evaluation of RI included generating the set of rules using the modified AQ algorithm for various values of parameters. The overall classification accuracy (Table 3) is low because for many events no rule is fired. The most important is the parameter β, its lower values ensure the higher accuracy. This also causes increasing the number of rules detecting the false alarm state. All other appliances are identified by only one rule. The training duration is high (compared to the RF), increasing with the increasing value of α (which does not influence the accuracy otherwise). Therefore the optimal AQ training configuration was obtained for $\alpha=10$ and $\beta=0.05$.

Table 3. Classification accuracy of RI for various parameters.

α	β	c_a	**No. of rules**
10	0.7	59.52	7
20	0.7	59.52	7
40	0.7	59.52	7
10	0.5	63.09	15
20	0.3	64.28	24
40	0.3	64.28	22
10	0.05	65.47	21
20	0.05	64.28	24
10	0.1	64.28	24

C. Implementation of the clasdification strategy

The following characteristics justify adding the RI auxiliary method to the RF (which is the most efficient standalone algorithm of all tested ones).

- In the case of missed events (when two appliances should be identified, because one event of the state change was missed) the RI is able to point at the second device, while RF identifies the first one (event No. 13 in Tab. 1).

- Although RI has low efficiency when detecting the false alarms, its response "0" should be treated as the most reliable, event in RF provides the other identifier.
- The greatest number of mistakes made by RI is caused by activating no rule for the particular features vector. But when the particular category is produced, it usually supports the RF, increasing the confidence of the classifier.

The following code of conduct should be adopted to increase the accuracy of the RF supported by RI:

- If the RF points at the particular appliance and the RI produces no output ("-1"), the proper system's response is the one from RF.
- If both classifiers identify different appliances, their responses should be considered as correct. This allows for detecting change of state in two devices, when the event of one of them was missed by the event detection procedure.
- If RF points at "0", while RI points at some appliance, the correct response should be "0" (as RI is not good in detecting the "false alarm" state).

The presented rules enable increasing the accuracy of the identification system by a few percent. For instance, in the presented sequence, the combined approaches give 91.66% accuracy, which is the best result obtained so far.

V. CONCLUSIONS

The implemented strategy allows for detecting and correctly identifying most of the appliance changes. The RF is the best standalone classifier, but the additional algorithm is able to increase its accuracy. The main problems during processing the sequence of two-state appliances are related to the identification of low energy devices, such as the power-saving lightbulb. When working alone, it is easy to detect, but during the operation with multiple devices in the background, its characteristics may remain hidden. In all other cases, including the false alarm detection, the combination of implemented classifiers is able to make the correct decision about the appliance that changed its state.

The future works require implementing other classifiers to the system and compare their efficiency. Also, the set of features should be verified, as more characteristics can be incorporated as well. Finally, the finite state machines should be introduced to the configuration of appliances to check if it is possible to determine not only the switch on/off event, but also changing the program of the washing machine etc.

ACKNOWLEDGMENT

This work has been accomplished within the project PBS2/A4/0/2013 "The Non-invasive System for Monitoring and Analysis of Electricity Consumption in the Area of the End-user" financially supported by the National Centre for Research and Development

REFERENCES

[1] G.. W. Hart, „Nonintrusive Appliance Load Monitoring," Proceedings of the IEEE, Vol. 80, No. 12, Dec. 1992, pp. 1870-1891.

[2] A.R. Rababaah and E Tebekaemi, "Electric Load Monitoring of Residential Buildings using Goodness of Fit and Multi-Layer Perceptron Neural Networks," IEEE International Conference on Computer Science and Automation Engineering (CSAE), 25-27 May 2012, Zhangjiajie, China, pp. 733-737.

[3] Z. Kolter and T. Jaakkola "Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation," Proc. 15th International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands.

[4] M. Baranski, J. Voss, "Detecting Patterns of Appliances from Total Load Data Using a Dynamic Programming Approach," Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)

[5] M. Figueiredo, A. Almeida, B. Ribeiro, "Home electrical signal disaggregation for non-intrusive load monitoring (NILM) systems," Neurocomputing, 96 (2012), pp. 66–73.

[6] S. Gupta, M. S. Reynolds and S. N. Patel, "ElectriSense: Single-Point Sensing Using EMI for Electrical Event Detection and Classification in the Home," Proc. 12th ACM international conference on Ubiquitous computing, 2010, pp. 139-148.

[7] J. Z. Kolter, M. J. Johnson, "A public data set for energy disaggregation research," Proc. SustKDD workshop on Data Mining Applications in Sustainability, 2011.

[8] M. Zeifman, and K. Roth, "Nonintrusive Appliance Load Monitor-ing: Review and Outlook," IEEE Trans. Consumer Electronics, vol. 57, No. 1, pp. 76-84, February 2011.

[9] P. Bilski, "Application of random forest to the fault detection in analog circuits," Proc. XXI IMEKO World Congress, 30 Aug.-4 Sept. 2015, Prague, Czech Republic

[10] P. Bilski, W. Winiecki, „Non-Intrusive Appliance Load Identification with the Ensemble of Classifiers," NILM Workshop 2016, Alberta, Canada.

[11] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac, "The AQ15 inductive learning system: An overview and experiments, Report 1260," Department of Computer Science, University of Illinois at Urbana-Champaign, 1986.