A Multi-Agent System for Bushing Condition Monitoring

Christina B. Vilakazi, Tshilidzi Marwala

School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
t.marwala@ee.wits.ac.za

Abstract
The problem of automatic fault diagnosis has been an ongoing research in power system. This paper proposes a design of a multi-agent system for bushing condition monitoring system. The agents collect dissolved gases measurements from gas chromatography, analyze the measurements and perform fault diagnosis. The interpretation agents are encapsulated with artificial neural networks so as to enable the agents to perform classification. The rationale behind the use of multi-agent techniques, and the problems that can be overcome through this technology are described. In addition, the performance of intelligent interpretation techniques is detailed.

1. Introduction
Bushings are a critical component in the transmission distribution of electricity. Most of transformer failures are linked to bushing faults [1], which results in lengthy downtime that have economic consequences. Hence, there is a high demand of a cost effective and automated condition monitoring that can detect faults as early as possible. Early diagnosis and fault identification is an important activity for maximizing a plant’s lifetime, operational costs and levels of safety. Incipient faults in bushing transformers are either electrical or thermal in nature, can degrade the oil and cellulose insulation, leading to the formation of dissolved gases. These faults can be detected and monitored using dissolved gas-in-oil analysis (DGA). Engineers have introduced better decision support for bushing condition monitoring procedures through applications of centralized intelligent systems by using a variety of artificial intelligence (AI) techniques [2]-[8]. It is now widely recognized that problems due to the complexity of condition monitoring can be overcome with architectures that contain many dynamically interacting intelligent distributed modules, called intelligent agents [9]. Each agent is an autonomous system, which processes a selection of inputs, and the complete interpretation and diagnosis is accomplished through interaction with other agents.

This paper introduces a real-time bushing condition monitoring using multi-agent system to facilitate the task of monitoring the conditions for high voltage transformer bushings using dissolved gases. The rest of the paper is arranged as follows. Section 2 provides an overview of agents and multi-agent systems, followed by the proposed system in Section 3. The experimentation is discussed in Section 4 followed by the conclusions and discussions in Section 5.

2. Overview of Agents and Multi-Agents System
Agents are software systems that function autonomously to achieve desired objectives in their environment. Generally, intelligent agents must exhibit four characteristics: autonomy, social ability, pro-activeness and reactivity [10]. Autonomy means each agent will operate in an independent mode, continually performing its function while altering its behavior as required. Social ability implies each agent can cooperate and communicate with other agents. Reactivity means intelligent agents are able to perceive their environment, and respond in a timely fashion to changes that occur in it in order to satisfy their design objectives. Proactiveness is the ability of agents to solve problems and ensure that they deliver the correct information or initiate the required control activity. A multi-agent system (MAS) can be defined as a loosely coupled network of problem solvers that work together to solve problems that are beyond the individual capabilities or knowledge of each problem solver [11]. The increasing interest in MAS research is due to the inherent advantages in such systems, including their ability to solve problems that may be too large for a centralized single agent, provide enhanced speed and reliability and tolerance for uncertainty in data and knowledge. There are several factors that motivated the use of multi-agent system for bushing condition monitoring. Firstly, they have the ability to solve problems that are too large for a centralized single intelligent agent, due to resource limitations or the sheer risk of having one centralized system. Secondly, they enhance modularity, which reduces complexity, increase speed due to parallelism, improves reliability due to redundancy, provide flexibility such as new tasks are composed more easily from the more modular organization and offer reusability at the knowledge level hence sharing of resources [10]-[13]. Lastly, MAS offer the extensibility and flexibility framework for integrating the necessary data capture system, monitoring system and interpretation function. This technology permits the development of more intelligent and automated diagnostic and monitoring functions. In this study, intelligent agents are used to build the MAS. In addition, several researchers have used multi-agent system for various condition monitoring applications for power systems [14][15].

3. System Design
The design framework as described in [11] was followed in designing a multi-agent system for bushing condition monitoring. Figure 1 shows the architecture used for developing the multi-agent system. The three main phases, as depicted in the figure are; requirements capture and task decomposition; agent modelling and agent interactions modelling.

3.1. Requirements Capture and Task Decomposition
In the agent community, task decomposition is concerned with the partitioning of the problem domain into agents. The main focus of this phase is gaining an understanding of what the sys-
tem does in theory, which, serves as a starting point for the architecture development process. The activities undertaken in this phase pertain to understanding the application domain, identifying goals and boundaries of the system, and relating them to the agent design. The requirements for a bushing condition monitoring system were decided on, based on existing condition monitoring system by studying their shortcomings and strong points. The task hierarchy of the knowledge capture is shown in Fig. 2.

3.2. Agent Modelling

Based on the analyzed goals, agents need to be identified and their relationships need to be modelled. In any system, there is a choice to be made about the number of functions that can be combined within a single agent versus function becoming autonomous. The requirements lead to the design of the condition monitoring system where functional modules are grouped by their overall goal. The bushing condition monitoring system consists of four layers; data monitoring; interpretation; diagnostic and information layer as shown in Fig. 3. Agents for each layer are discussed below.

3.2.1. Data Monitoring Layer

The data monitoring layer allows only relevant information to come into the system. Raw data from the sensors and associated condition monitoring systems is received and all necessary pre-conditioning takes place. The gases received from the bushing operation by the data monitoring layer are; methane, ethane, ethylene, acetylene and hydrogen, carbon monoxide, carbon dioxide, nitrogen and oxygen.

The data monitoring layer consists of one agent, the data preprocessing agent. The data preprocessing agent was designed to provide the necessary conditioned data to the interpretation layer agents. The interpretation agents are encapsulated with artificial neural network, this to ensure that the agent are able to learn different conditions of the bushing. Data conditioning is an integral part of neural network architecture as it makes it easier for the network to learn. There will be cases where, one or two variables are missing due to sensor failure. To accommodate inputs with missing variable, a zero imputation method is used. Markey et al. [15] showed that this method is effective in replacing missing variable. The main reason for retaining these blank entries is to ensure that no information about other dissolved gases that may still give some information about the condition of the equipment is lost. Another function of this agent is to normalized data, this is done using the min-max normalization. The equation for a min-max normalization for a single feature is:

$$ x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} $$

where $x_{\text{min}}$ and $x_{\text{max}}$ are the minimum and maximum value for that feature, respectively.

3.2.2. Interpretation Layer

The interpretation layer turns the data into information that can be easily interpreted by plant operator. This module uses artificial neural network, coupled with codified knowledge and expertise in the area of condition monitoring, to perform diagnosis. It supports more that one interpretation technique. Data interpretation is achieved through three agents, kernel-based classifier, backpropagation neural network, and a fuzzy neural network. As the interpretation layer agents are classifiers, they require to be trained before being used in the multi-agent system. Experimentation was done using DGA data collected from transformer bushings over a period of 2.5 years. The data was analyzed to produce output using the IEC60599 and IEEEc57.104 standard. The data was divided into three sets, the training, validation and testing data set. The training data set consists of 1000 instances, while the validation and testing data set contains 900 and 2000 instances, respectively. The classifiers are discussed below.

Kernel-Based Agent-Kernel-based classifiers have recently been used as popular and powerful tools for classification, due to their strong theoretical origin from statistical learning the-
ory as well as their high performance in practical applications [16]. Support Vector Machine (SVM) classifiers are kernel-based learning algorithms which determines the optimal hyperplane decision boundary in the feature space. A detailed description of SVM is found in [16]. A polynomial kernel function was used to train the support vector machines. The overall classification accuracy of the SVM is 95.63%, which is used as the voting weight in the diagnostic layer.

*Backpropagation Neural Network Agent* - The backpropagation neural network implemented is the multi-layer perceptrons (MLP). It employs supervised learning in the training of a neural network. The input data vector is presented to the network input layer while the output layer is presented with the "target" output. The network is refined through a process of error backpropagation, where the resultant error between the actual and target output is minimized. A detailed description of the multi-layer perceptrons can be found in [17]. An MLP that consists of 10 inputs layer nodes, 16 hidden layer nodes and 5 output layer nodes is implemented. The backpropagation neural network gave an overall classification accuracy of 91.47%, which is used as the voting weight for an MLP in the diagnosis stage.

*Fuzzy neural network Agent* - Fuzzy ARTMAP was developed by Carpenter et al. [18] and is a generalized ARTMAP system that learns to classify inputs by a fuzzy set of features or a pattern of fuzzy membership values between 0 and 1, which indicate the extents that each feature is present. A detailed description of the fuzzy ARTMAP can be found in [18]. A vigilance parameter of 0.75 was used; the learning rate was chosen to be 1, which is the fast learning mode. The overall classification accuracy of the fuzzy ARTMAP is 88.7%, which is used as a voting weight of fuzzy ARTMAP in the diagnosis layer.

### 3.2.3. Diagnostic Layer

The diagnostic layer is composed of one agent, the diagnosis agent. This agent takes the outputs of the interpretation layer agents and builds an overall diagnostic conclusion. It is known that the performance of ensemble is often much better than that of individual classifiers, because of independently-trained classifiers and their uncorrelated errors [19]. Since several independent agent classifiers are used, we need to aggregate them in an appropriate manner. A number of decision fusion techniques exist such as weighted majority voting, majority voting and trained combiner fusion. The majority voting is the simplest and widely-used aggregation method [19]. This voting scheme treats all agents with equal weights. Prediction errors of agent are often different, thus, it is more reasonable to give them different weights, in proportion to their prediction performance. In this study, the weighted majority voting is used. In the weighted majority voting, the predicted class label of the diagnosis agent is given by:

$$C_m = \arg \max_k \sum_{k=1}^{K} W(k,m) I_{km}$$

Where $W(k,m)$ is the weight when the predicted class label of the kth agent is $C_m$. The weights can be determined by calculating appropriate performance measure for each classifier as mentioned in previous section. The weights of the classifiers must add up to one. A table is built with classifier tabulated against conclusion, the cell data is then populated with various results that have been submitted to the diagnosis agent. Each time new results are added, the conclusion is recalculated. After the table is fully populated, the weighted majority voting is used to determine an outcome of the event.
This layer contains an Operator Assistant Agent. This is designed to present information to the relevant operator. And to handle diagnostic information from a number of different transformers and bushing the operator is responsible for. The information layer will give diagnosis and recommend a course of action to the operator. The complete design process for the bushing condition monitoring is shown in Fig. 3

3.3. Agent Interaction
Agent interaction defines communications and exchange of information between different agents. Agent-based systems require a communication infrastructure. Agents in the condition monitoring system communicate with each other by sending messages in the Agent Communication Language specified by FIPA [20]. Agent interaction is facilitated by the registry agent using subscriptions, query, inform and confirm command. These messages correspond to registration requests, information requests and other agent actions. A brief overview of agent actions and the corresponding messages is given below.

Registration of an agent with the registry agent: Each agent must register with the registry agent. A registration message includes the registering agent’s agent-id, list of published variables. The registry agent issues a confirmation message upon successfully entering the new agent in its database.

Information request by an agent about other agents: To find agents capable of providing required input data, the agent sends a search request to the agent registry. The search request includes the requester’s agent-id, and the required variables.

Registry agent’s reply to an information request: Upon receiving a search request, the registry agent verifies that the request is legitimate before searching its database to determine which agents can supply the requested variables and the status of these agents. The message from the registry agent to the requester includes the requested variable name, the agent-id of the agent publishing the variable and the status of the requested agent.

Request for belief subscription: Upon receiving the list of agents capable of providing the required input from the registry, the subscribing agent sends requests directly to these agents. A subscription request consists of the requester’s agent-id, requested input variable name, the duration of subscription time, the desired time interval between subsequent updates, and a request-id.

Belief-update messages: Upon receiving a belief subscription request the publishing agent sends regular updates within the agreed intervals and duration of the subscription. The message contains the request-id, the sender’s id and the requested information.

Figure 4 depicts a sample agent interaction for the bushing condition monitoring system.

4. Experimental Results
Having designed condition monitoring system using the most appropriate features of existing MAS design methodologies, the next stage was to implement the prototype. To achieve this the multi-agent building toolkit, JADE (Java Agent DEvelopment Framework) was used. JADE is a middleware that could be used to develop agent-based applications in compliance with the FIPA specifications for inter-operable intelligent multi-agent systems. JADE is java-based and provides the infrastructure for agent communication in distributed environments, based on FIPA standards. For experimental purposes, one data instance is passed to the data preprocessing agent and processed as described. The preprocessed data is now sent, via agent subscription and messaging, to each of the interpretation layer agents. The interpretation layer agents work on data simultaneously and upon completion the agents will forward their results to the diagnosis agent to perform diagnosis. From these, the first conclusions are generated by the multi-layer perceptron agent and this passes the result to the diagnosis agent using the agent interactions described. The agent returns results detailing the confidence or probability due to one of the possible conditions. The diagnosis table is as shown in Table 1. The table shows that the fault is of a thermal nature with a confidence level of 88.7%. Next, the kernel-based agent provides its result. This yields the results shown in Table 2 which indicate that there is a thermal fault with a confidence level of 92.2%.

The last result passed to the diagnosis agent is the conclusion from the fuzzy neural network. When added to the table, yield the result shown in Table 3.

At this stage, the bushing diagnosis agent concludes its calculation of the combined result. It determines that there is a thermal fault with a confidence level of 62.14%. This is now passed to the Operator Assistant Agent for display. Due to the fact that the different classifiers learn the data differently, this means they will make errors differently. In this case, the MLP and SVM agree on the nature of fault, although the fuzzy ARTMAP gives a different result, hence the reduced probability.

5. Discussion and Conclusions
This paper demonstrated a design of bushing condition monitoring system using dissolve gas-in-oil analysis. The bushing condition monitoring system captures and condition the data automatically. Furthermore, the system is able to learn the plant behavior over a period of time and based on the learned information it is able to offer diagnosis on the state of the transformer and transformer bushing. The system also offers clear and concise information about the state of the transformer and transformer bushing to the operator.

A multi-agent approach for the development and implementation of dissolve gas-in-oil analysis based condition monitoring systems to assist engineers with transformer and bushing diagnosis was adopted. This paper has explored the viability and benefits of multi-agent approach for bushing condition monitoring of bushings and transformer. The bushing condition monitoring architecture using multi-agent proves that the partitioning of the domain knowledge makes it easy to maintain and understand the system, and improve performance of the system. Furthermore, the distribution of responsibilities among agents, achieves modularity and reduced complexity. The adoption of a MAS architecture also provides a basis for continual improvement of the engineering decision support due to the flexibility and scalability achieved by the use of common communication mechanism. This system enables integration of further data sources and interpretation agents.

6. Acknowledgements
The financial assistance of the National Research Foundation (NRF) of South Africa, the Carl and Emily Fuchs Foundation and the Council of Scientific and Industrial Research towards this research is hereby acknowledged. Opinions and conclusions arrived at, are those of the authors and are not necessarily to be attributed to the NRF.
Figure 3: The bushing condition monitoring system

Figure 4: Sample agent interaction for bushing condition monitoring system

Table 1: Diagnosis results for the backpropagation neural network

<table>
<thead>
<tr>
<th>AGENT</th>
<th>NORMAL</th>
<th>CORONA</th>
<th>LOW ENERGY</th>
<th>HIGH ENERGY</th>
<th>THERMAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>0.0283</td>
<td>0.0283</td>
<td>0.0283</td>
<td>0.0283</td>
<td>0.8870</td>
</tr>
</tbody>
</table>
Table 2: Diagnosis results for the backpropagation neural network and kernel-based classifier

<table>
<thead>
<tr>
<th>AGENT</th>
<th>NORMAL</th>
<th>CORONA</th>
<th>LOW ENERGY</th>
<th>HIGH ENERGY</th>
<th>THERMAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>0.0283</td>
<td>0.0283</td>
<td>0.0283</td>
<td>0.0283</td>
<td>0.8870</td>
</tr>
<tr>
<td>SVM</td>
<td>0.0107</td>
<td>0.0107</td>
<td>0.0107</td>
<td>0.0107</td>
<td>0.9560</td>
</tr>
<tr>
<td>Results</td>
<td>0.019415</td>
<td>0.019415</td>
<td>0.019415</td>
<td>0.019420</td>
<td>0.92200</td>
</tr>
</tbody>
</table>

Table 3: Diagnosis results for the backpropagation neural network, kernel-based classifier and fuzzy neural network

<table>
<thead>
<tr>
<th>AGENT</th>
<th>NORMAL</th>
<th>CORONA</th>
<th>LOW ENERGY</th>
<th>HIGH ENERGY</th>
<th>THERMAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>0.0283</td>
<td>0.0283</td>
<td>0.0283</td>
<td>0.0283</td>
<td>0.8870</td>
</tr>
<tr>
<td>SVM</td>
<td>0.0107</td>
<td>0.0107</td>
<td>0.0107</td>
<td>0.0107</td>
<td>0.9560</td>
</tr>
<tr>
<td>Fuzzy ARTMAP</td>
<td>0.0213</td>
<td>0.0213</td>
<td>0.0213</td>
<td>0.9147</td>
<td>0.0213</td>
</tr>
<tr>
<td>Results</td>
<td>0.0201</td>
<td>0.0201</td>
<td>0.0201</td>
<td>0.3179</td>
<td>0.6214</td>
</tr>
</tbody>
</table>

7. References


