

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Fuzzy Cognitive Maps for Identifying Fault Activation Patterns in Automation Systems

Salman Mohagheghi

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/59611>

1. Introduction

Modern automation systems are equipped with multitude of intelligent electronic devices (IEDs) – sensors and actuators that can be remotely accessed, monitored or controlled. These various components within the system form an interrelated network of elements. From a layered architecture standpoint, every device/application/module/sub-system (hereafter referred to as *component*) provides some services to a certain group of components, while receiving services from some others. These components are all prone to incidents (hereafter referred to as *failure*), either internally due to natural faults and malfunctions, or externally due to natural causes or cyber intrusions. Clearly, in a web of interconnected components, failure or degradation of each one can affect the performance of those receiving its services, and ultimately impact the availability and integrity of the underlying automation system.

Therefore, for efficient operation of such a system, the operator has to ensure that the impact of a component's degradation or failure on the rest of the system is determined as quickly as possible, so that he/she knows what part(s) of the system can continue operating as is, what parts are conditionally operational, and what others cannot continue operating reliably and should be terminated or disconnected from the rest of the system. This indicates a measure of *trustworthiness* for the overall system and its components. In small systems, with few interdependent components, this task is rather trivial. However, an increase in the number of system components or the dimensionality of interdependencies among them can exponentially complicate matters. Although experienced operators and/or automated mechanisms exist in automation systems that can handle individual applications, a uniform solution that can analyze the aftermath and range of impact of a failure does not exist.

The objective of this article is to devise a solution for system-wide situational awareness. Our goal is to identify how the impact of a failure is spread across the automation system, and what

components have been affected by it and to what extent. This capability should be a crucial aspect of modern automation systems, since system complexity often prevents the operators from visual and/or heuristic inspection.

We propose a scheme based on graph-theoretic fuzzy cognitive maps (FCM) that, in the event of internal or external failures in one or more components in the system, is able to identify the range of fault activation, and indicate what other components have been (or will be) affected by it most. This provides additional situational awareness for the system operator and allows him/her to isolate untrustworthy functions and applications as soon as possible. To make the solution less dependent on usually subjective human knowledge, the map of interdependencies is developed in two phases:

- a. Building one-step-ahead local relationships based on expert knowledge and qualitative data
- b. Graph-theoretic importance analysis for global impact assessment

The rest of this article is organized as follows: Section 2 provides a review of the literature on situational awareness in automation systems. The basic concepts behind FCM and design challenges are discussed in section 3. The proposed methodology and proof-of-concept simulation results are presented in sections 4 and 5 respectively. Section 6 discusses some practical considerations of the proposed methodology, while its adaptability aspect is briefly discussed in section 7. Finally, conclusions appear in section 8.

2. Situational awareness — State of the art

Situational awareness in a system may relate to various aspects such as spatial, mission, resource or crew awareness [1]. Whenever the system needs to process and interpret a large number of data points, situational awareness becomes more vital. In large-scale automation systems, the size and complexity of the network poses its own unique challenges. While during normal conditions, operators may show relatively high situational awareness, their awareness level drops following disturbance onset [2], which is when responsiveness and effective decision making is needed most. This has been attributed to elements such as attention narrowing, memory trap, workload fatigue, data overload and complexity creep [3].

Traditionally, the notion of situational awareness in automation systems has tied in closely with the concept of state estimation (SE). Using the measurements available from across the system, SE algorithm tries to find the set of system states (for example, voltages at various nodes in a power system) that statistically fits this set of measurements best [4]. In order to complete the picture of the system operating condition, extensive research was conducted on estimating the network topology through identification of topology errors [5, 6], which led to the introduction of the concept of generalized state estimation [7] that presented a unified solution for SE using both network measurements and system topology data. To efficiently deal with the large size of the problem, research has been conducted on developing distributed state estimation solutions for large multi-area systems [8-10].

In conjunction with SE techniques, situational awareness solutions have also been developed making use of alarms and signals. However, even the simplest failures in an automation systems could lead to tens to hundreds of alarms brought to the attention of the system operator [11], which echoed the need to convert this raw data to easily-interpretable information for the human operator. The early approaches for this purpose were in the form of organizing and prioritizing alarm messages [12]. Some authors suggested techniques for priority and grouping schemes [13], which were later improved by using Boolean algorithmic and rule-based expert system techniques [14, 15]. However, these modified approaches were still difficult to implement for large systems, and lacked adaptability and flexibility. The trend therefore took a turn towards more powerful intelligent techniques such as artificial neural networks [14, 16], fuzzy systems [17-20], neuro-fuzzy systems [21], genetic algorithm [22] and Tabu search [23] among others.

In parallel to these efforts, many researchers have focused on developing awareness solutions by improving system observability. For instance, in power system automation schemes this has been achieved by adopting additional system-wide dynamic measurements through Phasor Measurement Units (PMU) [24, 25], which can then be used for disturbance event detection and analysis [26, 27]. Also, data visualization tools have been proposed to assist system operators with emergency situations [28-30].

With the advent and large-scale deployment of advanced sensors, the focal point of some researchers shifted towards developing methodologies for sensor data fusion, sensor data interpretation and knowledge extraction, and linking them to operator situational awareness [31, 32].

Although many of the solutions proposed in the literature succeed in providing situational awareness and vigilance for the system operator, some aspects are missing and require more research:

- Many of these techniques assist the operator in narrowing down and interpreting events, but are unable to provide information on the breadth (range of impact) and depth (how severely other system components are affected) of the event,
- Event detection and identification is the first step towards awareness, but post-event mitigation requires a measure of "trustworthiness" of the rest of the system,
- Many of the methodologies proposed in the literature are application-specific, and cannot be readily applied to all systems,
- Some of the situational awareness solutions proposed in the literature use the concepts of supervised and unsupervised learning to build a logical model for the system under study, where historical data on the past incidents are used to construct and train the model. However, in most practical systems, training data related to past events are not available. Even if they are, they are likely to be recorded in a qualitative and application-specific format,
- Finally, most approaches for sensor data integration and fusion treat the data points at the same level of importance, whereas in large-scale systems different data from different

components may have dissimilar importance for the overall performance of the system or the underlying application.

3. Graph-theoretic based approach

3.1. Basic idea

Cognitive maps were introduced in the 1970s to represent social scientific knowledge [33]. These were signed directional graphs (digraphs) where a positive link from vertex j to vertex i would indicate that j causally increases i , whereas a negative link would mean the reverse is true (Fig. 1). In this context, the *indirect effect* of vertex j on vertex i over a path P_q is negative if the number of negative signs (of the edges of P_q) is odd, and is positive otherwise. The *total effect* of j on i would then be negative if all the indirect effects (over all possible paths) are negative, positive if all indirect effects are positive, and indeterminate otherwise. It is safe to assume that in most real-world applications beyond a trivial level of complexity, the indeterminate case happens more frequently.

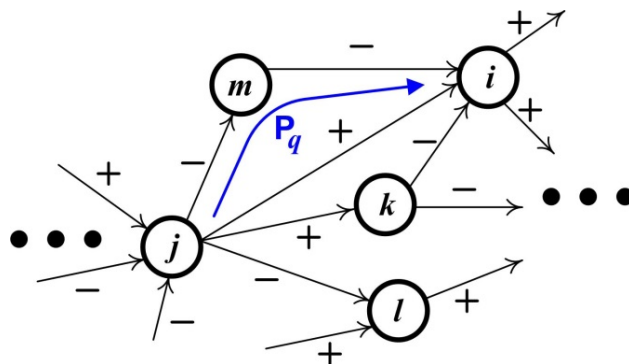


Figure 1. Schematic diagram of FCM with generic vertices and edges.

To further improve the computational capabilities of cognitive maps, Kosko [34] introduced fuzzy cognitive maps which assigned fuzzy weights to the edges connecting vertices. This allowed for determining the impact of vertices on one another based on the concepts of general fuzzy t -norm and t -conorm operators (minimum and maximum). Within this context, the indirect effect of vertex j on vertex i over a path P_q is defined as the minimum edge weight along that path, and the total effect of j on i would be the maximum of all indirect effects. In other words, the indirect effect represents the weakest causal link in a path connecting two vertices and the total-effect would be the strongest of the weakest links.

Later versions of FCM used algebraic equations to determine the value of each vertex based on the influence of the interconnected vertices (Fig. 2) [35].

This way, the state (value) x_i of vertex i at any point in time is defined as:

$$u_i(t) = \sum_{j=1, \neq i}^n w_{ij} x_j(t) \Rightarrow x_i(t+1) = f(u_i(t)) \quad (1)$$

where u_i represents the excitation level of vertex i , w_{ij} is the weight of the edge connecting j to i , and function $f(\cdot)$ is the threshold function defined, common choices being sigmoid or hyperbolic tangent. In matrix form, the state vector \mathbf{X} of the FCM can be expressed as:

$$\mathbf{X}(t+1) = f[\mathbf{W} \times \mathbf{X}(t)], \quad (2)$$

where \mathbf{W} is the weight matrix.

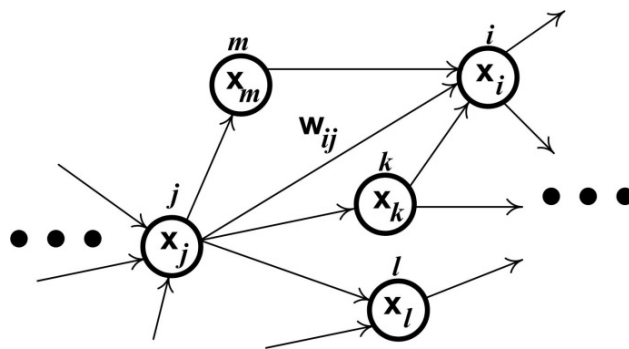


Figure 2. Schematic diagram of FCM with weighted edges.

3.2. Challenges with FCM

In a prior work, the author developed FCMs to model and analyze the interdependencies of typical power substations and to assess the impact of non-idealities in data quality on the overall performance of the system [36]. However, the maps were built manually based on expert knowledge, and it was shown that even for a simple system, the resultant FCM could become rather complex, which makes its manual derivation less feasible. One of the main challenges in this regard is how to assign weights to different edges of the graph. Traditionally, this has been done using expert knowledge. Some researchers have proposed modified techniques to measure the agreement between weights proposed by different experts [35], or assigning weights based on majority vote among experts [37]. To help automate the process further, some have used the notions of in-degree and out-degree of a vertex [38], the centrality (total degree) of the vertices [34], or concepts such as the key vertex [39] and the closeness (similarity) of relations between the states of two vertices i and j [40]. Alternatively, others have focused on methodologies for training the FCM and updating its weights [41].

While many of these techniques have proven effective in certain applications, they may fall short when applied to automation system situational awareness. First and foremost, it is often easy for human experts to identify and analyze *local* relations and interdependencies, but for large systems, an objective *global* system-wide analysis is difficult to obtain, and is likely to

become subjective. Moreover, in practice, an expert might be very knowledgeable in certain parts of the system, but not necessarily in all parts. Adopting training techniques to adjust the weights can solve the problem; however, its success largely depends on the availability of comprehensive sample data on system integrity related to past incidents, which is often not the case. Even if this data is available, it is not necessarily quantified and is likely to be qualitative, again subject to human interpretations.

4. Proposed methodology

4.1. Overview

Accordingly, the approach put forth in this article seeks a solution that takes advantage of the expert knowledge on local interrelations, but then modifies it towards a global perspective using the features of the abstract graph. The FCM is first formed based on the flow of data and communication services that exist between different components. The weights are then developed in a semi-automated fashion through three steps:

1. Using an expert system to determine the initial weights of the edges based on the short range (1-step ahead) dependencies of components,
2. Initial weights are then adjusted using the abstract centrality of the vertices (with no knowledge of the criticality/type of their services),
3. Weights are further tuned based on the relative number of paths from each vertex to the (user-defined) critical ones.

It should be noted that our proposed approach does not require conducting experiments on the system. Instead, it relies on the existing structure of the interdependencies and relations between different components as well as the expert knowledge (likely to be limited and local) on the criticality of these relations. As such, this methodology takes advantage of the graph-theoretic concepts that govern the automation system.

4.2. Assumptions

In developing the proposed solution, several assumptions have been made: Both external and internal faults have been considered; however, it is assumed that the faults are operational type (and not development faults [42]). Moreover, failure in service content is considered only, and not in its timing. Furthermore, dependability of components on one another has been analyzed in terms of availability and integrity.

Without loss of generality, notion of *quality* has been attributed to the services provided by different components in accordance with the quality attribute defined in IEC 61850 which is developed for utility automation systems [43]. Here, *validity* is defined as the main identifier of the quality attribute that determines the existence or lack of an abnormal condition of an acquisition function or source of information. Validity may be marked as good, invalid or questionable. An enumerated list *detail quality* further elaborates the reason for an invalid or

questionable value. This list of identifiers includes items such as overflow, out of range, bad reference, old data, inconsistent, inaccurate, etc. [43].

While the *quality* attribute provides valuable information for the operator, in many cases converting this information to useable data is not straightforward. For instance, the standard considers the client to be responsible for determining whether or not values marked questionable should be used. Moreover, the impact of each data value on the partial or overall performance of the substation in part depends on the interconnection of the protection, control and measurement functions within the system, which requires a methodology for measuring the propagation of bad data within these functions.

4.3. Algorithm

4.3.1. Step 0 – Initializing the FCM

To initialize the structure of the FCM, various components in the system are first modeled as the vertices of the graph. Any physical or logical component can be replaced by a vertex. The flow of data and services between these components determines the directional edges that interconnect the vertices. Data may refer to metering data, alarm signals, setting data, and suchlike. Services may refer to actuation signals, control commands, set-point adjustments, and data retrieval commands, etc. The directions of the edges are determined based on the notion of *client and server*. In other words, FCM edges connect servers (providers of data and services) to clients (users of data and services).

4.3.2. Step 1 – Local weight initialization

In the first step, an expert system is used to determine the initial weights. The weights derived this way would be short-sighted, i.e., the impact of each component on the very next one can be quantified relatively accurately, but the precision is expected to decline as the depth of impact of the component increases. The rationale adopted here is based on how a vertex B uses the services from vertex A (see Table 1).

Criticality Classification	Definition
Very critical (very high)	B only receives services from A
Critical (high)	B receives services from multiple components (including A), where no component can fully replace or replicate the services provided by A (i.e., no redundancy)
Semi-critical (medium)	B receives services from multiple components (including A), where services provided by A can be partially replaced or replicated by another component (i.e., partial redundancy)
Non-critical (low)	B receives services from multiple components (including A), where services provided by A can be fully replaced or replicated by another component (i.e., full redundancy), or when services provided by A are not essential for B to function

Table 1. Classification of edge relations $A \rightarrow B$

Without loss of generality, weights 1.0, 0.7, 0.4 and 0.1 have been chosen to indicate very critical, critical, semi-critical and non-critical edges respectively. The reader should note how the definition proposed here is purely local, with no system-wide impact in mind.

4.3.3. Step 2 — Weight update based on graph structure

Centrality of a vertex indicates how connected it is to the rest of the graph, and can be used as a measure of importance, since it shows the number of vertices in the graph that are related to this vertex. Centrality C of a vertex is defined based on the notions of its in-degree and out-degree. For a vertex i [38]:

$$\text{od}(i) = \sum_{j=1}^n |w_{ji}|, \quad \text{id}(i) = \sum_{j=1}^n |w_{ij}| \quad (3)$$

$$C(i) = \text{od}(i) + \text{id}(i) \quad (4)$$

However, in this article in order to decouple the expert system based weight derivation in step 1 (which can be subjective) from weight update in the current step, the notion of *abstract centrality* is used instead as shown in [44]. Here, the importance (centrality) of vertices will be derived based on abstract in-degree and out-degree as proposed in (5):

$$\text{aod}(i) = \sum_{s, s \neq i} h(w_{si}), \quad \text{aid}(i) = \sum_{s, s \neq i} h(w_{is}) \quad (5)$$

$$\text{where } h(u) = \begin{cases} 0 & u = 0 \\ 1 & u \neq 0 \end{cases}$$

$$C^*(i) = \text{aod}(i) + \text{aid}(i) \quad (6)$$

The previously derived weights will therefore be adjusted based on the abstract centrality $C^*(i)$ of each vertex i , in such a way that outgoing edges from vertices with high centralities will be further strengthened:

$$\forall j : w_{ji}^* = w_{ji} (1 + \beta \cdot \hat{C}(i)) \quad (7)$$

where β is a learning rate parameter (heuristically chosen here to be 0.20) and \hat{C} represents the normalized abstract centrality based on the range of $[C_{\min}^*, C_{\max}^*]$ values of all abstract centralities.

4.3.4. Step 3 — Weight update based on relation to critical vertices

Here, the severity of the relationship between vertices and the ones identified as critical (by the operator) are quantified. Definition of critical vertices is application specific. In automation systems, for instance, critical vertices could be defined as those taking an “action”, e.g., protection relays, switch controllers, etc. In a graph with k critical vertices, the severity of the relationship between vertex i and the set of critical nodes is defined as:

$$S_i = \sum_k \sum_p \frac{1}{L_{(i,k)}^p} \quad (8)$$

where $L_{(i,k)}^p$ is the length of the p^{th} path that connects vertex i to vertex k . To maintain decoupling from step 1, calculation of the length of the paths should ignore the weights of the edges in between. In this article, the length is defined as the number of intermediate vertices along the path. Severity is then used to update the weights:

$$\forall j : w_{ji}^* = w_{ji} (1 + \eta \cdot S_i) \quad (9)$$

where η is the learning rate parameter.

4.4. Discussion

It can be seen that relative independence is ensured between the three steps of the algorithm. This is necessary, because part of the information that is obtained from the expert system is subjective, and should not interfere with the other two steps that are graph-dependent.

Validation of the FCM output can be done using the historical data on past failures in the system. Of course, if desired, part of the historical data can be used for updating the weights, as shown in [44]. Here, the available data can be divided into independent training and testing batches to evaluate the efficiency of the training algorithm. However, training using historical data is not adopted here due to the fact that its availability in sufficiently large scale is often unlikely.

5. Case study

The system studied in this article is a modified version of the IEEE 34-bus test distribution system [45]. For the purpose of this study, the system has been expanded by adding fixed and switched capacitors, reclosing switches, distributed generation (DG) units, on-load tap

changer (OLTC) with line drop compensation (LDC), and various metering IEDs across the feeder and the laterals (Fig. 3).

It is assumed that the system is equipped with advanced distribution automation (DA) system consisting of functions for conservation voltage reduction (CVR), voltage and var control (VVC), electric service restoration (RES), and DG dispatch. These interrelated functions share services from different devices:

- CVR: receives voltages from end-of-lateral meters (V_6 and V_7 in this example) and based on the current through the OLTC I_1 and the substation voltage V_1 regulates the LDC settings so that no voltage across the system drops below ANSI low. It also receives relevant data from VVC.
- VVC: controls the settings of the tap changers T_2, T_3 and switched capacitors C_{S1}, C_{S2} in order to minimize the losses. The capacitor switches are equipped with Point-on-Wave switching scheme. The function consults the network configuration, itself being updated based on the statuses of the reclosing switches among other things.
- RES: manages the reclosing switches R_1, R_2 and R_3 across the system in the event of a disturbance. For proper performance, the function is updated based on the output of the DG units DG_1 and DG_2 , and the power flow meters at certain locations of the feeder, i.e., I_1, I_2, I_3 and I_4 .
- DG Dispatch: regulates and manages the power generated by the DG units DG_1 and DG_2 based on economic and environmental considerations, as well as requirements from the RES function.

It can be seen that even this simple example portrays various levels of interdependencies between different DA applications, to an extent that they make visual assessment of system integrity very difficult, if not impractical. The initial structure of the FCM for the system in Fig. 3 has been developed based on the presumed interrelations (flow of data and communication services) between different components (Fig. 4).

The arrows shown in the diagram denote the direction of impact/dependence, i.e., $A \rightarrow B$ indicates that services of A have an impact on B. Each vertex in the FCM is assumed to have a linguistic state space of {failed (not trustable), questionable, low confidence, not verified, healthy}, which without loss of generality has been quantified as {1.0, 0.7, 0.4, 0.2, 0.0}. The choice of numbers might sound counter-intuitive to the reader, but it is necessary here, since the proposed FCM models the chain of “failures” in components. The weights of the FCM are first initialized based on the logic in Table 1. They are then adjusted based on the abstract centrality of the vertices according to (7). Finally, they are updated according to (9). For the purpose of this algorithm, critical vertices have been assumed to be VVC, DG Dispatch and RES.

A few examples have been provided to indicate the effectiveness of the proposed methodology. For all use cases, matrix \mathbf{W} is developed based on weights of the FCM (Fig. 4). Simulations have been performed in MATLAB® environment. The system starts from healthy state (all

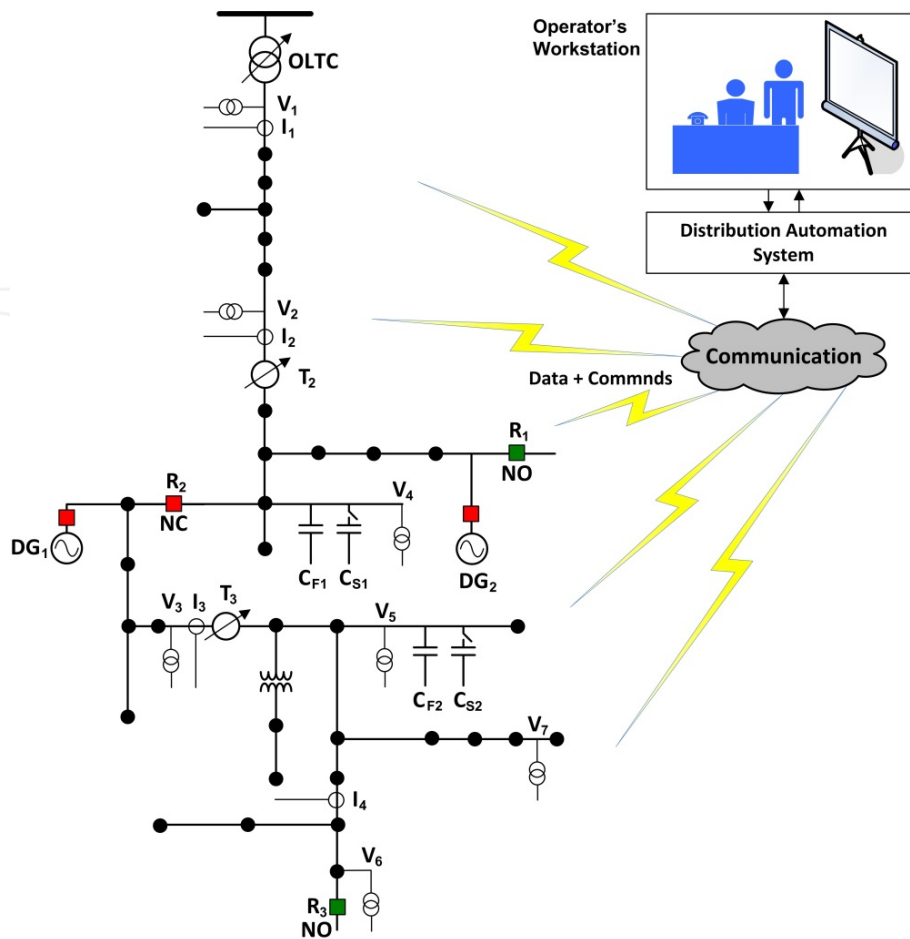


Figure 3. Modified 34-bus test distribution system. Switches denoted as NC and NO represent “normally closed” and “normally open” respectively. Measurement and status data are collected from the field and sent to the control center. Commands are sent from the control center to the field devices.

zeroes) except for the vertices that are considered to experience a failure. The state update equation (2) is repeated in a loop until the impact of the unhealthy vertices propagates across the whole system, i.e., the loop is iterated as many number of times as the length of the longest path originated from the unhealthy vertex.

5.1. Case study 1: Impact of failures on major functions

The following case studies have been analyzed:

- *Non-Verified Voltage Measurement V1:* voltage meter V1 at the distribution substation indicates a value that has not been updated (not verified).
- *Questionable Recloser Status R1:* the status of recloser R1 is indicated to have a questionable data quality as it has been oscillatory with relatively fast changes.
- *Outdated Tap Position T2:* the tap position of T2 indicates that the upper threshold is reached, which does not match the expected operation based on engineering judgment (low confidence).

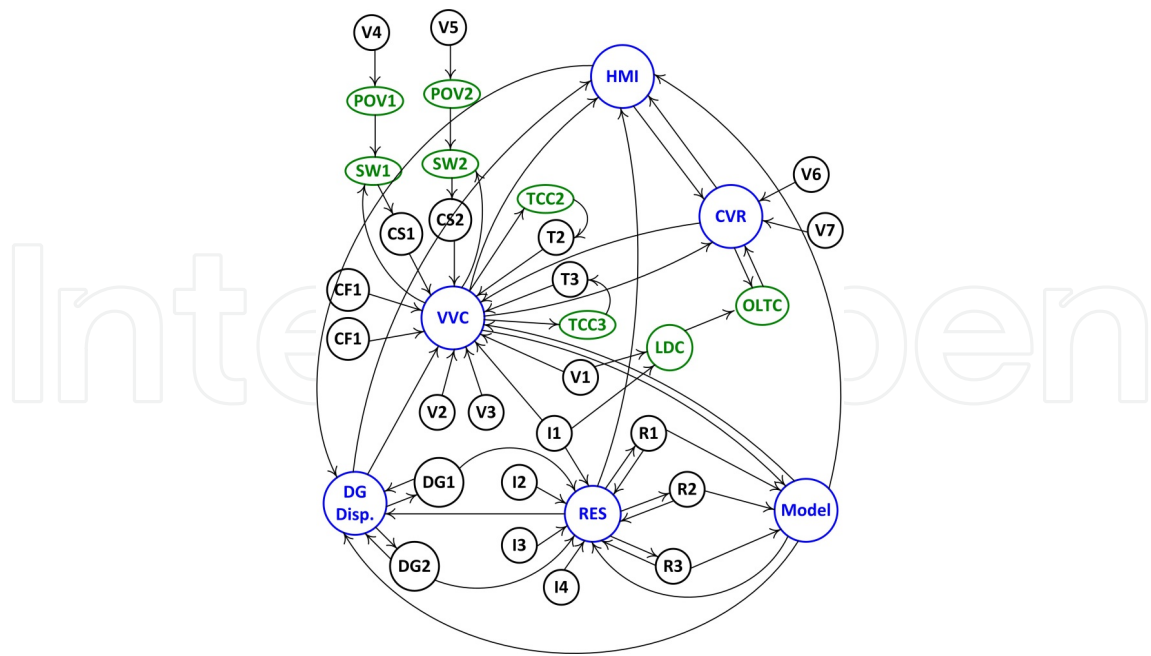


Figure 4. FCM developed for system in Fig. 3. Vertices in blue represent DMS applications that reside at the control center. Vertices in black represent devices and components out on the field. Those shown in green can be implemented either on the field or at the control center. Abbreviations: TCC: tap changer controller, CF: fixed capacitor, CS: switched capacitor, SW: capacitor switch, POV: point on voltage waveform. It is assumed that the DG sends all relevant information (e.g., active/reactive power, terminal voltage, injected current, connection status) to control center. Relationship between reclosers R and function RES is in the form of “send recloser status” and “receive control command”. TCC ensures that the upper or lower tap positions for each tap changer are not violated.

- *Failed Switch SW1:* switch responsible for controlling switched capacitor CS1 is failed with stuck blades.

Table 2 shows the impact of each case study on some of the main applications within the distribution automation system. The state of each application indicates where it lies in the range of healthy (0) to failure (1). It should be noted that the interpretation of values between 0 and 1 would be subjective; nonetheless, it would provide a qualitative measure of integrity for the application of interest. It can be seen that while some of the conclusions provided in Table 2 are easy to draw based on expert knowledge (e.g., RES would fail if R1 is questionable), others are not easy to infer from the local interrelations (e.g., the impact on DG dispatch and VVC). This is the advantage of the proposed methodology that converts local interrelations to global system-wide impacts.

Case Study	VVC	DG Dispatch	RES	Model	CVR
V1 not verified	0.26	0.07	0.07	0.12	0.17
R1 questionable	0.66	1.00	1.00	1.00	0.29
T2 low confidence	0.42	0.13	0.14	0.21	0.23
SW1 failed	0.70	0.12	0.07	0.16	0.25

Table 2. Impact of case studies on major DA applications

5.2. Case study 2: Range of impact of a failure

It is assumed in this case study that switch SW1 has failed. The goal is to provide an overall view of the system dependability. Colored vertices in Fig. 5 show the range of impact of the failure. It can be seen that the proposed FCM-based solution assists the human operator in detecting what functions/components have been affected by the failure and to what extent. For example, this case study indicates that as a result of failure in SW1, operation of VVC cannot be trusted, which in turn causes the information provided at the HMI to have low confidence or to be questionable. Such a visual overview of system trustworthiness can be extremely beneficial to the operator for making fast decisions during contingencies in order to save the rest of the system. For instance, in this failure scenario, the operator may decide to temporarily deactivate the CVR function so as to avoid instances of non-optimal voltage profile across the distribution system.

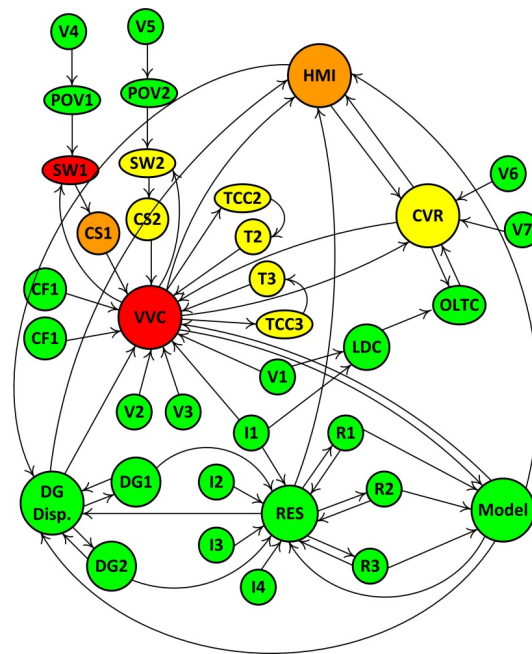


Figure 5. FCM developed for system in Fig. 3. The following color codes have been defined associated with case study 2: green (healthy to not verified), yellow (not verified to low confidence), orange (low confidence to questionable), and red (questionable to not trustable).

6. Practical considerations

6.1. Scalability

One of the main features in the proposed methodology lies in its ability to model various systems regardless of their size and number of components. In fact, any large system can be modeled as a hierarchical FCM. The key is to decompose the overall system into sub-systems and model each one as an FCM. This would represent the first layer of hierarchy. Each lower

layer FCM will then be modeled as an “equivalent vertex” for the FCM at the upper layer (Fig. 6). This allows for masking the dynamics and interrelations of lower layer sub-systems from the upper layers. This way, the operator can first pinpoint which sub-system has been affected by a failure and how this would impact the rest of the sub-systems. If desired, the operator can then study the individual components within that particular sub-system. In theory, there are no limits to the number of the layers defined.

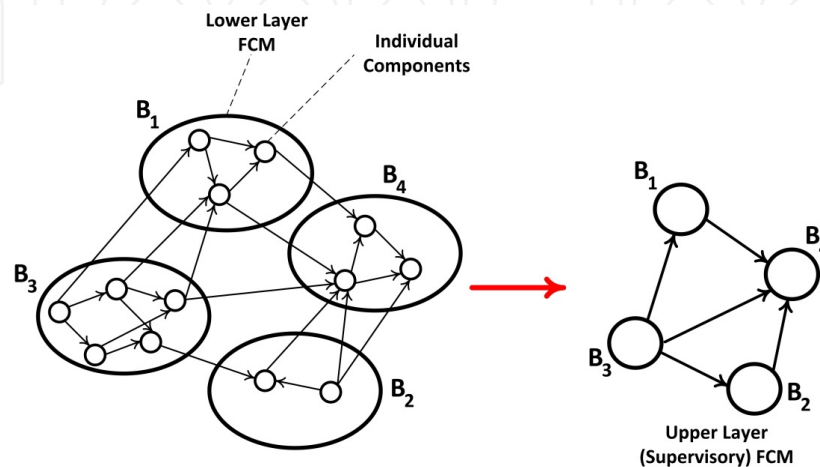


Figure 6. Hierarchical FCM comprising multiple lower layer FCMs that represent sub-systems.

Zhang *et al.* [46] proposed one such solution based on the notion of equivalence relation. In their approach, the set of vertices were first partitioned into blocks, where each block contained a portion of the original FCM. The authors referred to the upper layer as the quotient FCM which was responsible for providing global information about the original system. The sectional FCMs would then provide local information on the interactions and interrelations between vertices.

6.2. Reliability and accuracy of the results

The reliability and accuracy of the proposed FCM-based solution depends in part on the accuracy of the initial graph created (initial structure and initial weights). This, in turn, depends on the validity of the expert knowledge on local interrelations between components. Once the correct initial graph is in place, the graph theory rules will modify the “local” solution to a “global” one. However, in achieving that goal, learning rate parameters in equations (7) and (9) play an important role, and if chosen incorrectly may lead to solutions that overestimate or underestimate the impact of failures. These parameters should therefore be assigned carefully, typically starting from low values and gradually increasing them until reasonable outcomes are achieved. In the lack of historical data on the past failures in the system, heuristic selection of these parameters would be the only option. However, if this data exists (even if it is incomplete and partial) it can be used as training data to validate the output of the FCM and modify the learning rate parameters as necessary.

7. FCM adaptation

7.1. Structural changes to the FCM

Many automation systems may undergo structural changes and modifications as time passes. New components may be installed and/or current components may be upgraded with extended sensing and actuation capabilities. In the context of the logical interconnections mapped by the FCM, this indicates one or more additional vertices.

The following steps need to be undertaken to incorporate a new vertex into the FCM model:

- Create the new edges connecting the new vertex to the existing vertices. The direction of these edges will be determined based on the flow of data and services,
- Initialize the weights for all edges according to the definition in Table 1. All edges connecting the existing vertices to the new vertex will receive a new initial weight,
- Recalculate the abstract centrality for all vertices according to (6),
- Normalize the new abstract centrality values based on the new minimum and maximum values,
- Update the initial weights for all edges according to (7),
- Recalculate the severity index S_i for all vertices according to (8),
- Finally, update all the weights according to (9).

When components (or their sensing/actuation functionalities) are removed from the system, one can either remove the corresponding vertices from the graph and follow the steps above, or simply remove the corresponding entries of state vector \mathbf{X} and the rows/columns of matrix \mathbf{W} associated with that vertex.

7.2. Training the FCM

In this article, it was assumed that historical data on the past incidents in the system are not available and therefore the weights were only created and adjusted based on heuristic and graph-theoretic rules. However, it is possible that partial data be available on some past instances of failure in the system. This data may be obtained from post mortem analysis following a disturbance, and may be in the form of “when component x failed, components y and z , and applications a and b failed (completely or partially) as a result.” In all likelihood, this data will be qualitative. If sufficient expert knowledge is readily available to quantify the level and severity of these past failures then the data can be used for updating all or some of the weights of the edges. Two common scenarios are possible: supervised learning and unsupervised learning. For an overview of these two approaches, the reader is referred to [41].

7.2.1. Supervised learning

In this case, for the given input data, the actual values of the system states X_{actual} will be compared with the desired values X_{desired} . The difference between the two vectors will determine the direction of update in the weights. One algorithm may be expressed as:

$$w_{ij}(t) = w_{ij}(t-1) + \alpha \cdot \Delta x_j(t-1) \cdot [\Delta x_{i,\text{desired}}(t-1) - \Delta x_{i,\text{actual}}(t-1)] \quad (10)$$

where α is a learning rate parameter, which can be customized for each individual edge. For other variations of supervised learning, see [48-52].

7.2.2. Unsupervised learning

Unsupervised Learning methods adjust the weights of the FCM without any knowledge of the desired target values of the vertices. As such, this approach would be more appropriate for cases where desired system states are not known exactly (or at least not known in a quantified fashion). Many unsupervised learning schemes are based on the Hebb's learning law, where the change in the synaptic weight w_{ij} linking vertices j and i is expressed as a function of signal flows before and after the link:

$$\Delta w_{ij} = \alpha \cdot x_j \cdot x_i \quad (11)$$

In order to solve the instability issues, Oja modified the rule as [53]:

$$\Delta w_{ij} = \alpha \cdot x_j \cdot (x_i - w_{ij} x_j) \quad (12)$$

In essence, this method tries to strengthen the link between two vertices whose state values move in the same direction, and weaken the link should they move in opposite directions (e.g. one increases while the other decreases). For other variants of unsupervised learning, see [54-57].

8. Concluding remarks

Latest advances in sensor and actuator technology have made them attractive and, at the same time, affordable for large scale deployment across automation systems. The benefits of improved observability and controllability come at the price of having a more complex system of interrelated components, where failure of one may cause many others to fail or become degraded. System integrity assessment in this case requires a thorough analysis of the scope of impact of component failures. However, the potentially large number of components may prevent human operators from efficiently interpreting and identifying failure impact.

A solution was proposed in this article for providing situational awareness and dependability assessment of components within an automation system that is exposed to partial failure. A combination of expert rule-based and graph-theoretic approaches were used to develop a cognitive map of global interdependencies in the system. Expert knowledge was first used to create an initial directional and weighted graph model of the automation system based on the interrelations between components and their local (one-step ahead) impacts on one another. In this model, a component is modeled as a vertex of the graph, and an edge between two vertices would indicate the existence and direction of flow of data/services between them. This model was then modified based on the assessment of centrality and criticality of different vertices in the graph. This way, the local solution that was developed by a human expert would be converted into a global solution that considers the overall system performance.

By reproducing the fault activation pattern, the proposed solution can provide additional situational awareness for the system operator during or in the aftermath of a fault, and can be used for static as well as dynamic verification of system dependability. It was shown through simulation results that the proposed solution allows the system operator to identify the range of impact of a failure on the overall system and determine what parts of the system remain trustworthy and dependable.

Author details

Salman Mohagheghi*

Electrical Engineering and Computer Science Department, Colorado School of Mines, Golden, CO, USA

References

- [1] R.W. Pew, "The state of situation awareness measurement: circa 1995," in *Proc. Intl. Conf. Experimental Analysis and Measurement of Situation Awareness*, Daytona Beach, FL, USA, Nov. 1995, pp. 7-16.
- [2] B.P. Hallbert, "Situation awareness and operator performance: results from simulator-based studies," in *Proc. IEEE Annual Human Factors Meeting*, Orlando, FL, USA, Jun. 1997.
- [3] M.R. Endsley and E.S. Connors, "Situation awareness: state of the art," in *Proc. IEEE PES*, Pittsburgh, PA, USA, Jul. 2008.
- [4] A. Monticelli, "Electric power system state estimation," *Proc. IEEE*, vol. 88, no. 2, pp. 262-282, Feb. 2000.
- [5] F.F. Wu and W.H.E. Liu, "Detection of topology errors by state estimation," *IEEE Trans. Power Systems*, vol. 4, no 1, pp. 176-183, Feb. 1989.

- [6] I.S. Costa and J.A. Leao, "Identification of topology errors in power system state estimation," *IEEE Trans. Power Systems*, vol. 8, no 4, pp. 1531-1538, Nov. 1993.
- [7] O. Alsac, N. Vempati, B. Stott and A. Monticelli, "Generalized state estimation," *IEEE Trans. Power Systems*, vol. 13, no 3, pp. 1069-1075, Aug. 1998.
- [8] Z. Liang and A. Abur, "Multi area state estimation using synchronized phasor measurements," *IEEE Trans. Power Systems*, vol. 20, no 2, pp. 611-617, May 2005.
- [9] Y. Li, X. Zhou and J. Zhou, "A new algorithm for distributed power system state estimation based on PMUs," in *Proc. Intl. Conf. on Power System Technology*, Oct. 2006.
- [10] A.P. Sakis Meliopoulos, G.J. Cokkinides, F. Galvan, B. Fardanesh and P. Myrda, "Delivering accurate and timely data to all," *IEEE Power & Energy Magazine*, vol. 5, no. 3, pp. 74-86, May/Jun. 2007.
- [11] D. Droucher, "Langage: an expert system for alarm processing", presented at the 11th Biennial IEEE Workshop on Power Systems Control Centers, Montréal, BC, Canada, Sept. 1990.
- [12] D.S. Kirschen and B. Woolenberg, "Intelligent alarm processing in power systems", *Proc. IEEE*, vol. 80, no. 5, pp. 663-672, May 1992.
- [13] H. Amelink, A.M. Forte and R.P. Guberman, "Dispatcher alarm and message processing", *IEEE Trans. Power Systems*, vol. PWR5-1, pp. 188-194, Aug. 1986.
- [14] E.H.P. Chan, "Application of neural network computing in intelligent alarm processing", in *Proc. Power Industry Computer Application Conf.*, San Jose, CA, May 1989, pp. 246-251.
- [15] C.A. Protopepes, K.P. Psaltiras and A.V. Machias, "An expert system for substation fault diagnosis and alarm processing", *IEEE Trans. Power Delivery*, vol. 6, no. 2, pp. 648-655, Apr. 1991.
- [16] R. Khosla and T. Dillon, "Learning knowledge and strategy of a neuro-expert system architecture in alarm processing", *IEEE Trans. Power Systems*, vol.12, no. 4, pp. 1610-1618, Nov. 1997.
- [17] S.S. Choi, K.S. Kang, H.G. Kim and S.H. Chang, "Development of an on-line fuzzy expert system for integrated alarm processing in nuclear power plants", *IEEE Trans. Nuclear Science*, vol. 42, no. 4, pp. 1406-1418, Aug. 1995.
- [18] C.L. Nelson and D.S. Fitzgerald, "Sensor fusion for intelligent alarm analysis", *IEEE AES Systems Magazine*, pp 18-23, Sept. 1997.
- [19] E.M. Meza, J.S.S. de Souza, M.T. Schilling and M.B.D.C. Filho, "Exploring fuzzy relations for alarm processing and fault location in electrical power systems", in *Proc. IEEE Power Tech*, Sept. 2001.
- [20] H.J. Cho and J.K. Park, "An expert system for fault section diagnosis of power systems using fuzzy relations", *IEEE Trans. Power Systems*, vol. 12, no. 1, pp. 342-348, Feb. 1997.

- [21] E.M. Meza, J.S.S. de Souza, M.T. Schilling and M.B. Filho, "Alarm processing in electrical power systems through a neuro-fuzzy approach", *IEEE Trans. Power Delivery*, vol. 19, no. 2, pp 537-544, Apr. 2004.
- [22] F. Wen, C.S. Chang and D. Srinivasan, "Alarm processing in power systems using a genetic algorithm", in Proc. *Intl. Conf. on Evolutionary Computation*, Perth, WA, Nov./Dec. 1996.
- [23] F.S. Wen and C.S. Chang, "Tabu search approach to alarm processing in power systems", *IEE Proceeding*, vol. 144, no. 1, pp. 31-38, Jan. 1997.
- [24] R.B. Bobba et al., "Enhancing grid measurements," *IEEE Power & Energy Magazine*, pp. 67-73, Jan./Feb. 2012.
- [25] S.E. Widergren, Z. Huang and J.E. Dagle, "Electric system-wide measurements: North American directions," in Proc. 40th Hawaii Intl. Conf. on System Sciences, Waikoloa, HI, USA, Jan. 2007.
- [26] G. Zhang, K. Sun, H. Chen, R. Carroll and Y. Liu, "Application of synchrophasor measurements for improving operator situational awareness," in Proc. IEEE PES, Detroit, MI, USA, Jul. 2011.
- [27] H. Qi et al., "Increasing the resolution of wide-area situational awareness of the power grid through event unmixing," in Proc. 44th Hawaii Intl. Conf. on System Sciences, Kauai, HI, USA, Jan. 2011.
- [28] T.J. Overbye, "Visualization enhancements for power system situational assessment," in Proc. IEEE PES, Pittsburgh, PA, USA, Jul. 2008.
- [29] J. Wang, Y. Hu, A. Johnson, H. Tram and R. Nasri, "System requirements of visualization platform for wide area situation awareness system," in Proc. IEEE PES, Minneapolis, MN, USA, Jul. 2010.
- [30] C. Zheng, Y. Dong, O. Gonen and M. Kezunovic, "Data integration used in new applications and control center visualization tools," in Proc. IEEE PES, Minneapolis, MN, USA, Jul. 2010.
- [31] R. Laughery et al., "Decision-centered design as the basis of defining the human role in systems," in Proc. *IEEE Human Factors Meeting*, Scottsdale, AZ, USA, Dec. 2002.
- [32] Y. Liang, H.Q. Wang, H.B. Cai and Y.J. He, "A Novel stochastic modeling method for network security situational awareness," in Proc. *IEEE Conf. Ind. Elec. Appl.*, Singapore, Jun. 2008, pp. 2422-2426.
- [33] R. Axelrod, *Structure of Decision: the Cognitive Maps of Political Elites*, Princeton, NJ: Princeton University Press, 1976.
- [34] B. Kosko, "Fuzzy Cognitive Maps," *International Journal Man-Machine Studies*, vol. 4, pp. 65-75, 1986.

- [35] C.D. Stylios and P.P. Groumpos, "Modeling complex systems using fuzzy cognitive maps," *IEEE Trans. Systems, Man and Cybernetics – Part A*, vol. 34, no. 1, pp. 155-162, Jan. 2004.
- [36] S. Mohagheghi, "A Fuzzy cognitive map for data integrity assessment in a IEC 61850 based substation," in Proc. *IEEE PES*, Jul. 2010.
- [37] B. Lazzerini and L. Mkrtychyan, "Analyzing risk impact factors using extended fuzzy cognitive maps," *IEEE Systems Journal*, vol. 5, no. 2, pp. 288-297, Jun. 2011.
- [38] D.B. West, *Introduction to Graph Theory*, Prentice Hall, 2001.
- [39] Y. Miao and Z.Q. Liu, "On causal inference in fuzzy cognitive maps," *IEEE Trans. Fuzzy Systems*, vol. 8, no. 1, pp. 107-119, Jan. 2000.
- [40] M. Schneider, E. Shnaider, A. Kandel and G. Chew, "Automatic construction of FCMs," *Fuzzy Sets Syst.*, vol. 93, pp. 161-172, Jan. 1998.
- [41] E.I. Papageorgiou, "Learning algorithms for fuzzy cognitive maps – a review study," *IEEE Trans. Systems, Man and Cybernetics: Part C*, vol. 42, no. 2, pp. 150-163, Mar. 2012.
- [42] A. Avižienis, J.C. Laprie, B. Randell and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11-33, Jan/Mar. 2004.
- [43] *Communication Networks and Systems in Substations*, IEC Standard 61850 (parts 1-10), 2010.
- [44] S. Mohagheghi, "Integrity assessment scheme for situational awareness in utility automation systems," *IEEE Trans. Smart Grid*, Vol. 5, No. 2, pp. 592-601, March 2014.
- [45] IEEE Dist. Planning Working Group, "Radial distribution test feeders," *IEEE Trans. Power System*, vol. 6, no. 3, pp. 975-985, Aug. 1991.
- [46] J. Y. Zhang, Z. Q. Liu, and S. Zhou, "Quotient FCMs—A decomposition theory for fuzzy cognitive maps," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 5, pp. 593-604, Oct. 2003.
- [47] Y. Boutalis, T.L. Kottas and M. Christodoulou, "Adaptive Estimation of Fuzzy Cognitive Maps with Proven Stability and Parameter Convergence," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 874-889, Aug. 2009.
- [48] T.L. Kottas, Y.S. Boutalis and M.A. Christodoulou, "Fuzzy Cognitive Network: A General Framework," *Intelligent Decision Technologies*, vol. 1, pp. 183-196, 2007.
- [49] X. Luo, X. Wei and J. Zhang, "Guided Game-Based Learning Using Fuzzy Cognitive Maps," *IEEE Transactions on Learning Technologies*, vol. 3, no. 4, pp. 344-357, Oct/Dec. 2010.
- [50] H. Song, C. Miao, W. Roel, Z. Shen and F. Catthoor, "Implementation of Fuzzy Cognitive Maps Based on Fuzzy Neural Network and Application in Prediction of Time Series," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 2, pp. 233-250, Apr. 2010.

- [51] W. Stach, L. Kurgan, W. Pedrycz and M. Reformat, "Genetic Learning of Fuzzy Cognitive Maps," *Fuzzy Sets and Systems*, vol. 153, pp. 371–401, 2005.
- [52] W. Stach, L. Kurgan and W. Pedrycz, "A Divide and Conquer Method for Learning Large Fuzzy Cognitive Maps," *Fuzzy Sets and Systems*, vol. 161, pp. 2515–2532, 2010.
- [53] E. Oja, "Neural Networks, Principal Components and Subspaces," *International Journal of Neural Systems*, vol. 1, no. 1, pp. 61–68, 1989.
- [54] A.V. Huerga, "A Balanced Differential Learning Algorithm in Fuzzy Cognitive Maps," in Proc. *International Workshop on Quantitative Reasoning*, Sitges, Spain, Jun. 2002.
- [55] E.I. Papageorgiou, C.D. Stylios and P.P. Groumpos, "Active Hebbian Learning Algorithm to Train Fuzzy Cognitive Maps," *International Journal of Approximate Reasoning*, vol. 37, pp. 219–249, 2004.
- [56] E.I. Papageorgiou and P.P. Groumpos, "A Weight Adaptation Method for Fuzzy Cognitive Map Learning," *Soft Computing*, vol. 9, no. 11, pp. 846–857, 2005.
- [57] W. Stach, L. Kurgan and W. Pedrycz, "Data-Driven Nonlinear Hebbian Learning Method for Fuzzy Cognitive Maps," in Proc. *IEEE International Conference on Fuzzy Systems*, Hong Kong, Jun. 2008.

