December 18, 2018

WIRELESS MESH NETWORK FAULT DIAGNOSIS AND MANAGEMENT BASED ON BINARY TREE BASED SUPPORT VECTOR MACHINE

Lele Zhang

Chuanwei Li

Huimin She

Li Zhao

WIRELESS MESH NETWORK FAULT DIAGNOSIS AND MANAGEMENT BASED ON BINARY TREE BASED SUPPORT VECTOR MACHINE

AUTHORS:
Lele Zhang
Chuanwei Li
Huimin She
Li Zhao

ABSTRACT

Techniques are provided herein for using a Support Vector Machine (SVM) method for fault diagnosis and device management by data mining in the cloud. The result could provide valuable references for specialists to analyze and then perform operations to avoid loss. In current Connected Grid Mesh (CG-Mesh) implementations, many status profiles of nodes are collected for further use, but fault diagnosis is not performed. Preventive strategies for users are also provided.

DETAILED DESCRIPTION

Connected Grid Mesh (CG-Mesh) networks provide connectivity for Advantage Metering Infrastructure (AMI) and Distributed Automation (DA). It is usually composed of up to millions of nodes, which is a very large scale deployment. Some nodes have poor performance for various reasons. Currently, logging and status report of nodes for analysis is provided in case they are broken. These fault diagnoses depend on skillful engineers, who have rich experience to take over. But these engineers often cannot perform debugging work local to users due to the high expense of business travel. Thus, it may take an inordinate amount of time to communicate with customers (e.g., incorrect description, jetlag problem, etc.). For instance, as illustrated in Figure 1 below, there are two cases for the node in the bottom right corner.

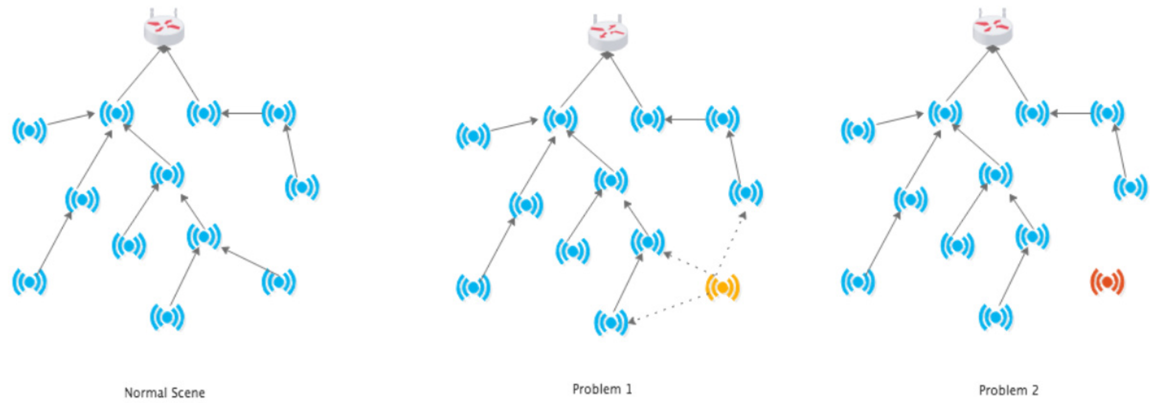

Figure 1

In problem 1, the trouble node is off-line occasionally, and users have to notify the engineering team due to lack of technical experience. Information is obtained from both the trouble node and its neighbors. In this example, it is determined that the node always changes its parent due to bad signal conditions. Finally, the suggestion is to put an additional node between the yellow node and its nearest neighbor as a range extender for forwarding messages.

In problem 2, the trouble node cannot be heard by any node, and has most likely been damaged.

While such problems are relatively simple for the development team because it has sufficient professional knowledge and skill, they are often difficult for non-specialist users. If a user realizes that there are existing problems in the Wireless Mesh Network (WMN), the network may be in big trouble because of a considerable number of unworkable nodes. This can lead to large losses on the user side. The engineering team may require more record details and accurate descriptions to resolve the problem if site commission practice is not possible. The whole process often takes a long time due to inefficient communication, and the users likely do not have enough skill and experience to provide desired elements. Sometimes, the user may provide inaccurate or incorrect information which misleads the engineering team resolving the issue.

As such, techniques are provided for an improved Support Vector Machine (SVM) based on a binary tree to resolve such problems by using a cognitive cloud. In particular, an SVM learning mechanism may identify the most likely user problem and provide effectual advice before manual intervention is required.

With the development of advanced machine learning techniques, more and more difficult problems have been settled by using Machine Learning (ML) algorithms. Support Vector Machine (SVM) is a well-known ML method widely used in fault diagnosis application for large industrial equipment and may provide smarter suggestions and successfully reduce repair costs.

A set of valuable properties may be used as training samples for the SVM algorithm. The cloud collects and updates these information elements from all available nodes at any time. These properties may include neighbor information, bidirectional frame statistics, topology information, modulation, symbol rate, deployment place, frequency band, and remote logging.

Neighbor information provides signal conditions from neighbors (e.g., bidirectional Received Signal Strength Indication (RSSI) and Link Quality Indicator (LQI), quantity of visible neighbors, path cost such as Expected Transmission Count (ETX) or Rank, etc.). Bidirectional frame statistics provides the number of frames the node receives or transmits. Different kinds of frames represent different meanings. For instance, discovery beacon request / discovery beacon is associated with a problem relating to the beginning of joining a Personal Area Network (PAN), and sync beacon request / sync beacon means the node has trouble finding an available parent node. Topology information provides the depth of the node, which bundles with problem about route path or parent selection. Modulation may be 2 Frequency-Shift Keying (2FSK) or Orthogonal Frequency-Division Multiplexing (OFDM). Symbol rate may be 50 kbps or 150 kbps. Deployment place may be indoor or outdoor (e.g., wild field or campus). Frequency band may be 900 MHz (adopted in U.S.) or 800 MHz (adopted in India). Remote logging may be worth being mined using ML.

SVM multi-class classification may be leveraged for better fault diagnosis. Figure 2 below illustrates an example infrastructure wherein the cognitive cloud leverages an SVM algorithm to detect whether there is a problem on some nodes, and furthermore, to diagnose what kind of error it is.

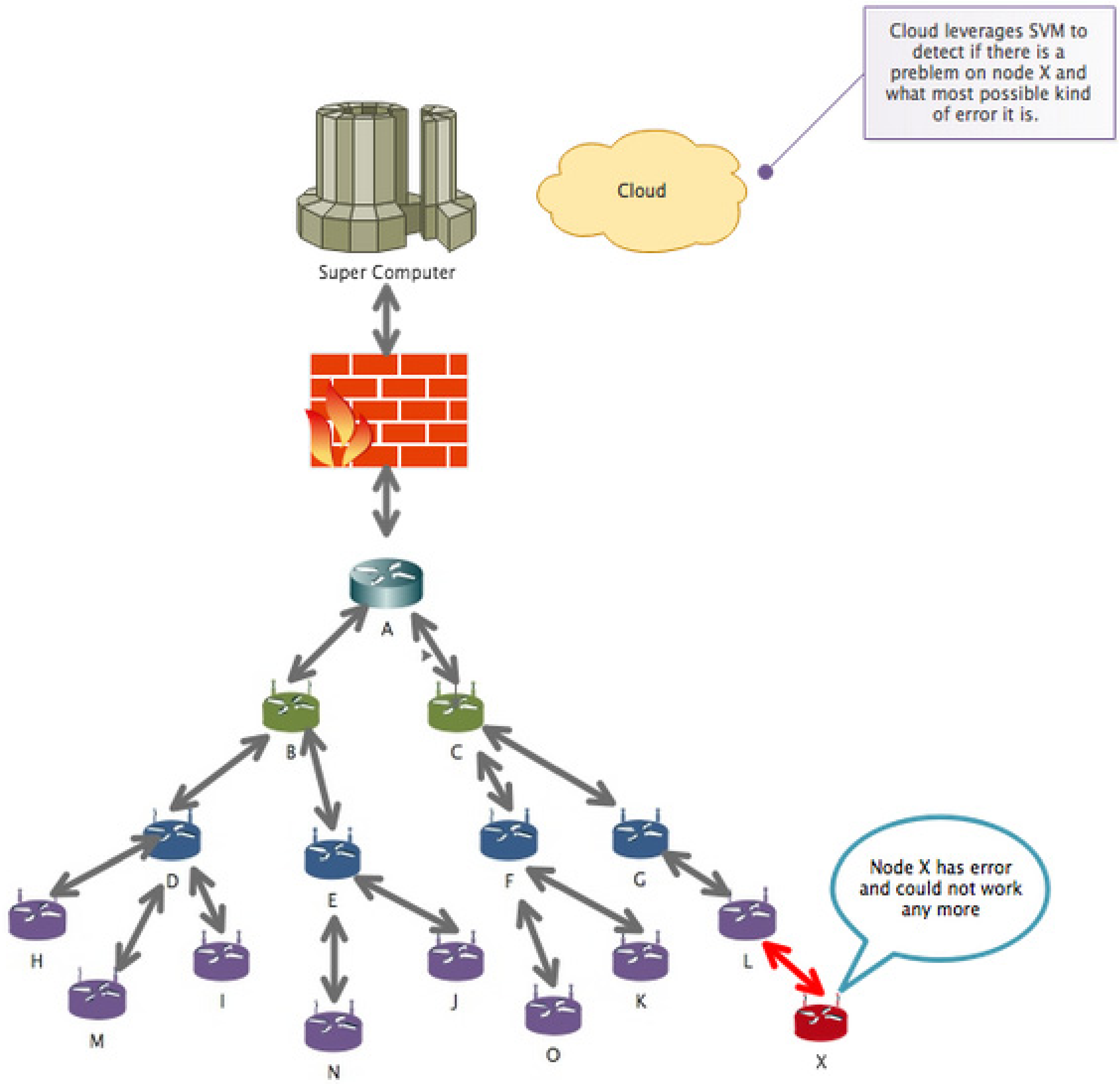

Figure 2

Some possible classes for identifying problems may include the following: hardware problem, poor signal condition, security problem, and software problem. Hardware problems may include power off, key component damaged such as Radio Frequency (RF), etc. Poor signal condition may include strong interference, too many collisions because of dense deployment, etc. Security problems may include wrong certificate, expired key, hacker attack, etc. Software problems may include software bugs.

Classical SVM was originally designed for a two classification problem, and may be divided as linear-based or non-linear-based. As illustrated in Figure 3 below, SVM works well for determining a perfect boundary which has a maximum marine to distinguish one class from another class.

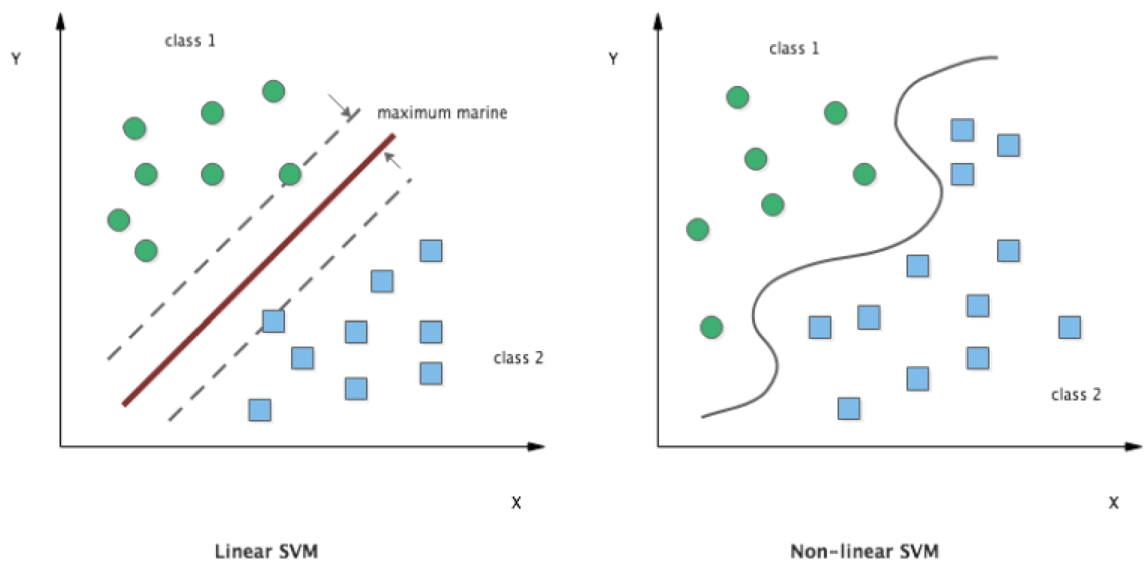

Figure 3

As shown, the boundary line with SVM may be obtained for either the linear case or non-linear case. But the practical problems involve multi-class classification rather than simple two-class classification. Many solutions have been proposed to help address this problem, such as One-versus-the-rest method (1-v-r), one-versus-one method (1-v-1), Decision Directed Acyclic Graph (DDAG), etc.

As illustrated in Figure 4 below, these problems cannot be solved with these existing multi-class SVM solutions perfectly, although they are good enough for other user cases in practice.

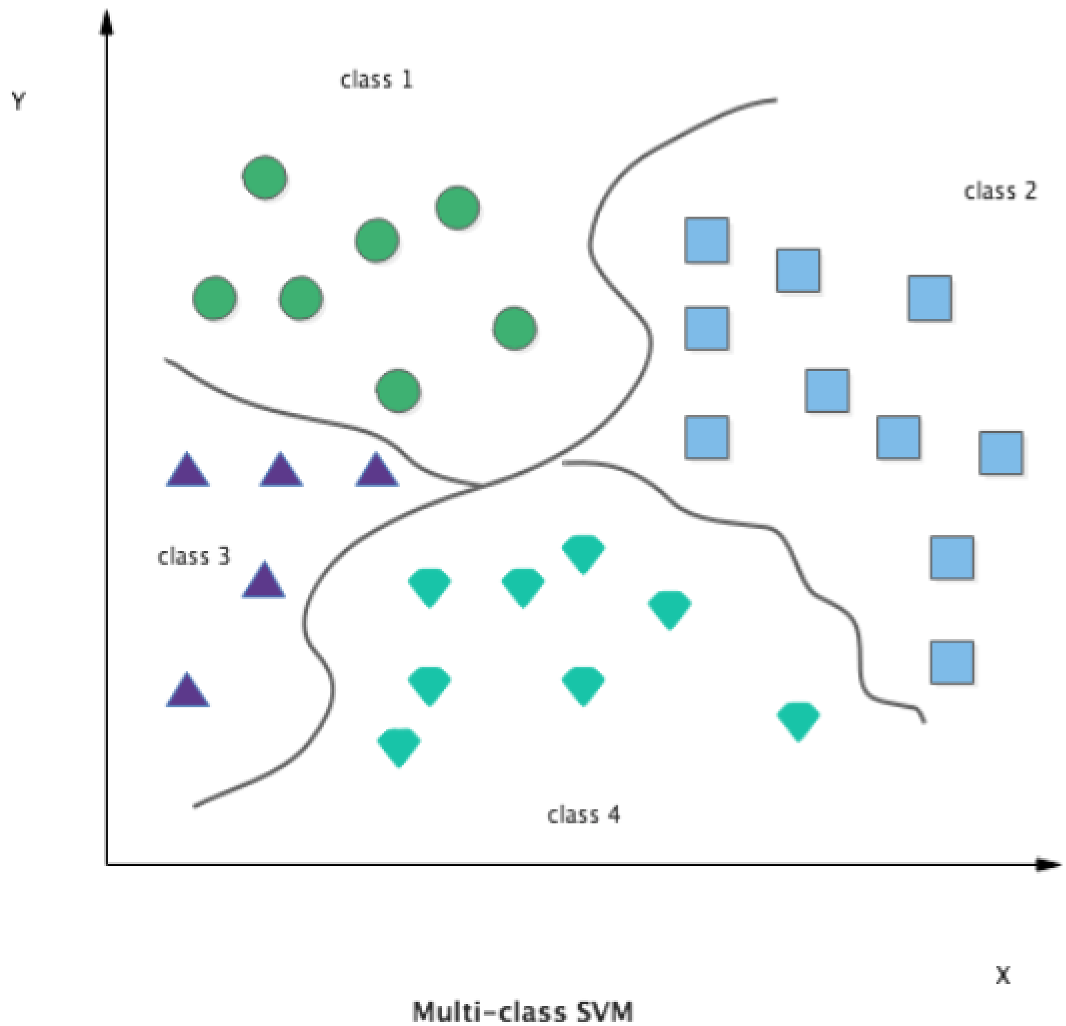

Figure 4

For the 1-v-r method, classical SVM just supports distinguishing classes between positive and negative, and as such this method treats class i as a positive class when it is desired to support class i, and the rest of the classes will be marked as negative classes. Figure 5 below illustrates the weakness of this method as typically having a big residual value.

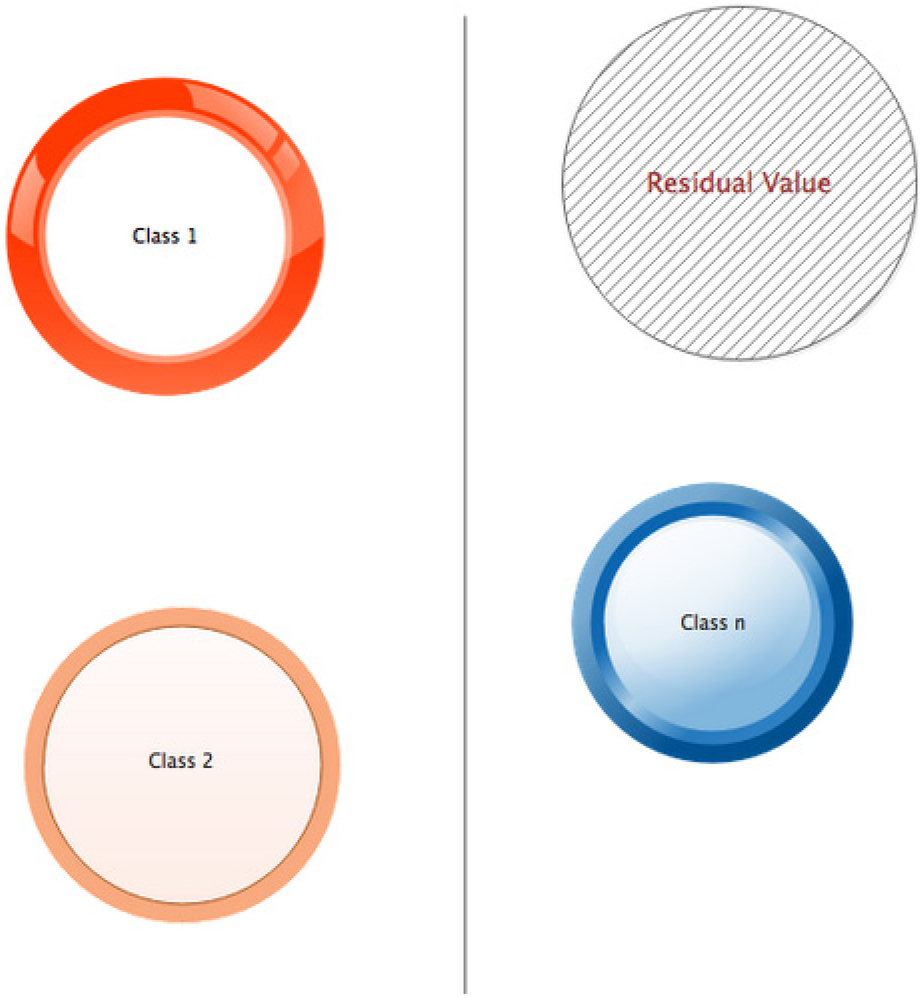

Figure 5

The 1-v-1 method uses the concepts of permutation and combination to build k(k-1)/2 possible situations if there are k classes. Each possibility is tested, and the class having the highest score is selected as the final optimal result. As illustrated in Figure 6 below, like the 1-v-r solution, its weakness is having uncertain residual value.

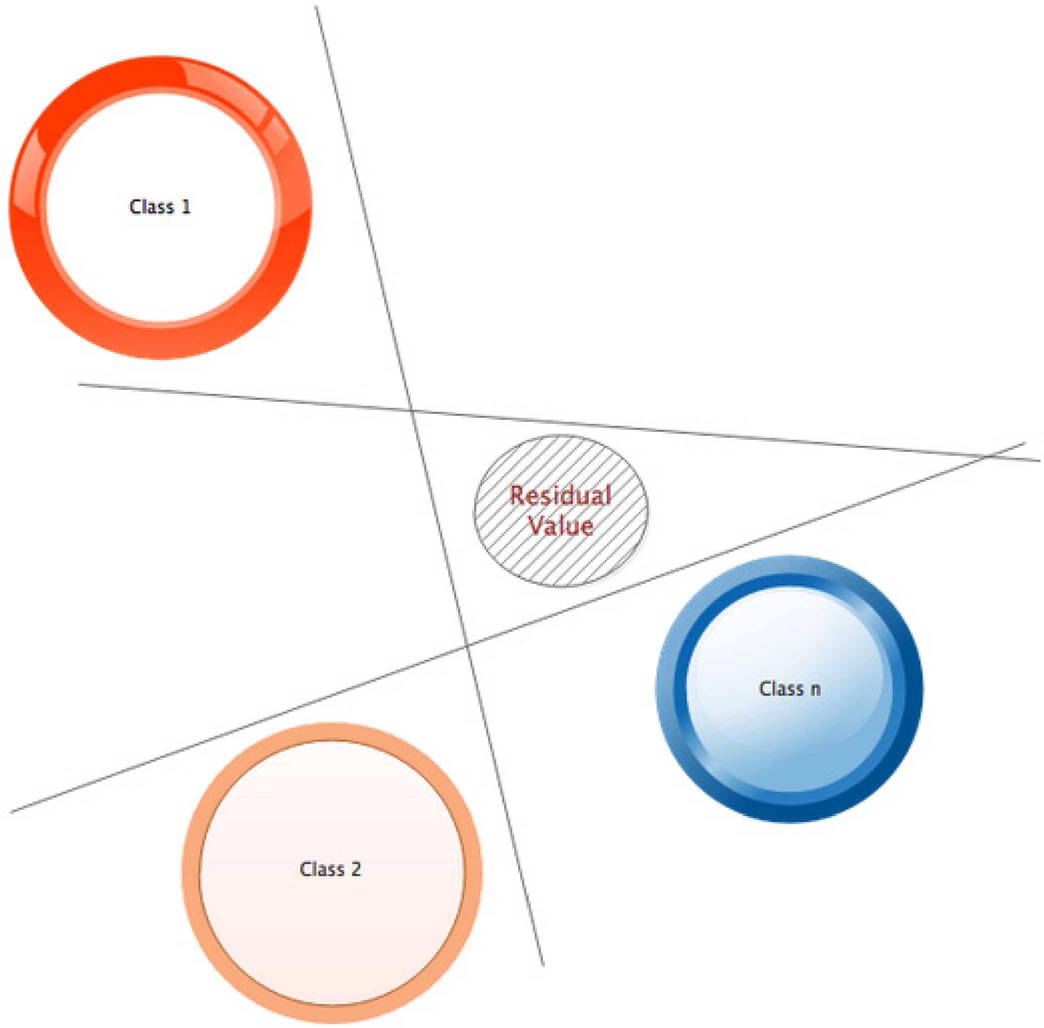

Figure 6

The DDAG method generates k(k-1)/2 possible situations, as does 1-v-1 method at the beginning, then splits the classes of both ends per round to form a two-class classification solution. As illustrated in Figure 7, it is much better than the aforementioned methods because its residual value only depends on the maximum margin of each class. But it is not an optimal solution, because the ends are not the optimal split class in practice.

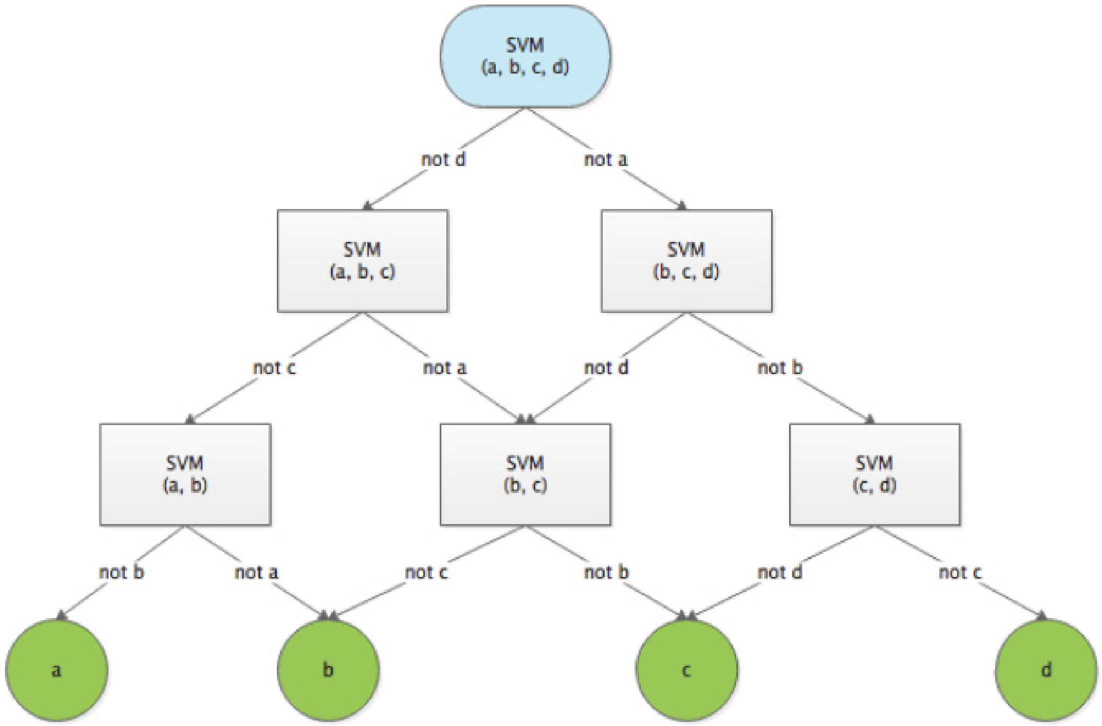

Figure 7

As described herein, an improved SVM method is based on a binary tree to reduce the residual value as much as possible. Binary tree based SVM adopts desirable properties from the binary tree. All samples may be continuously divided into two classes and then again to either sub-class. The loop ends when there is only one class left. This method is easy to understand and also has rapid convergence. The only weakness to be overcome is avoiding error accumulation because of the incorrect split node at the beginning. Based on the above, Euclidean distance may be used to overcome this weakness.

At a first example step, it is assumed that training set X contains k classes. X_i represents the sub-training set and only includes class i samples. If the number of samples is n for class i, the center of class i samples is:

$$C_i = \frac{1}{n} \sum_{x \in X_i} x$$

According this formula, every class center may be obtained. The centers may be placed into a new set $Y = \{C_1, C_2, ..., C_k\}$. If C_i and C_j represent the center for class i and class j respectively, the Euclidean distance between them may be obtained with the following formula:

$$Edis_{(i,j)} = \| \; C_i - \; C_j \; \|$$

At a second example step, the aforementioned formulas may be leveraged to calculate all the Euclidean distance for each pair, and determine which pair has the maximum Euclidean distance. Assume two classes, class i and class j, are found. Class i is placed into empty set X1, and class j is placed into another empty set X2. A SVM training algorithm is set on this split node at that time. Then, set X is updated with the following formula:

$$X = X - (X1 \; \cup X2)$$

The Euclidean distance from one of remaining class k to class i ($Edis_{(i,k)}$) and class j ($Edis_{(j,k)}$) is calculated. Class k may join X1 if $Edis_{(i,k)}$ <= $Edis_{(j.k)}$, and otherwise it will join X2. The set X may be updated with the aforementioned formula, then another remaining class may be continued to be picked up unless set X is empty.

At a third example step, the same actions may be repeated with both set X1 and X2, and two smaller subsets may be obtained every time. At last, each subset has only one class, and then the whole algorithm is finished. Figure 8 below illustrates the whole process.

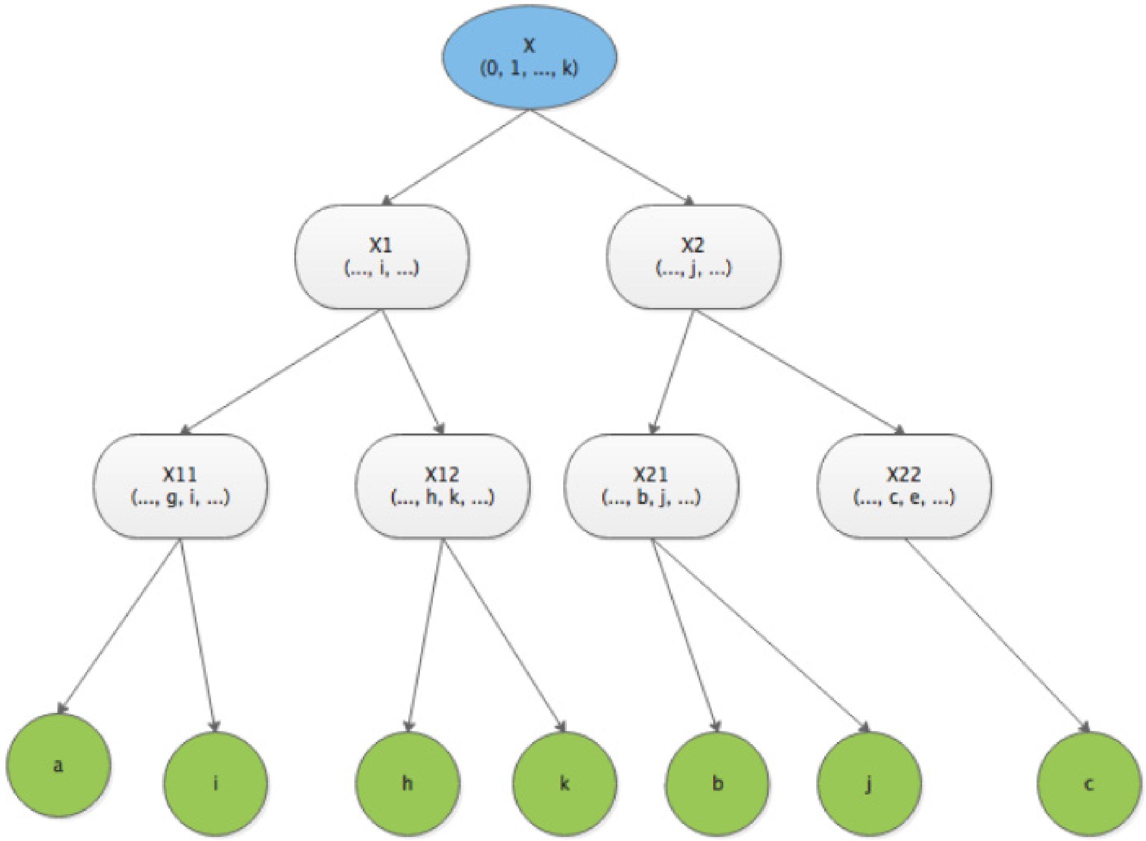

Figure 8

Some preventive strategies are also provided to reduce the penalty because of WMN fault after obtaining the result from SVM. There may be three levels to mark the serious degrees with low, middle, and high for each aforementioned class. The cognitive cloud may adopt with different strategies according to the defect level.

The low level is for tiny errors that have very small-scale bad impacts. Thus, the cloud only needs to generate fault logging and then integrate it into the existing logging. These logging files are required to report to the engineering team for further analysis periodically.

The middle level can affect the middle-scale of the network and reduce performance correspondingly. Therefore, the cloud needs to show a warning for users and provide some advances for user adopting except reporting the result to the engineering team.

The high level faults are most serious in the WMN, which means there are considerable nodes could not connect with the cloud. The cloud needs to provide to the engineering team all the detailed logging and analysis reports. It is required to update the latest valuable information from current reachable nodes intelligently as well.

Rather than using an ML algorithm, each of these three levels may be defined by engineers using their rich experience. The reason is that the method may provide more accurate results. When experienced engineers make use of the result generated by SVM, they may discover the root cause faster and more accurately than before.

In summary, techniques are provided herein for using a SVM method for fault diagnosis and device management by data mining in the cloud. The result could provide valuable references for specialists to analyze and then perform operations to avoid loss. In current CG-Mesh implementations, many status profiles of nodes are collected for further use, but fault diagnosis is not performed. Preventive strategies for users are also provided.