|  |
| --- |
| **Regional Voltage Control** |
| OpenECA Analytic Design Document |
|  |
|  |
| **Duotong Yang**  **Zhijie Nie** |
| **11/08/2016** |
|  |

# **Statement of Work**

The continuing success of synchrophasors has ushered in several new subdomains of power system applications for real-time situational awareness, online decision support, and robust system control. In this paper, an adaptive decision tree based systematic method for open-loop regional voltage control is developed. This approach employs voltage security assessment method to generate voltage secure and insecure operating conditions for decision tree learning. Parallel decision trees corresponding to a set of control combinations are trained. To guarantee the robustness of decision trees and its capability of tracking the system condition after topology changes, the resulting trained trees are updated periodically in real time using an online boosting method. The alpha version of this analytic is evaluated using IEEE 118 bus system.

# **Introduction**

The study of voltage control for transmission grids has a long history, a number of different methodologies had been developed in the past century. To provide more cost-effective and more accurate system-wide voltage control, a control methodology based on parallel DTs is proposed in this study. With this methodology, each DT provides voltage security assessment (VSA) for each control decision. To guarantee the robustness of the proposed technique after possible topology changes or any unpredictable system conditions, the trained DTs should be updated periodically. Instead of retraining the DT from scratch, an adaptive DT training scheme using an online boosting method is introduced to update the trees in a timely manner.

The detailed concept of this analytic is illustrated in Figure 1. As shown, the analytic is divided into two parts: online and offline. The offline adapter is implemented to create/update decision trees based on EMS data snapshot. When the tree is created/updated, they are mapped into the online adapter which is running as a module in openECA. Synchrophasor measurements will fall into the tree and provide VSA for each control combinations.

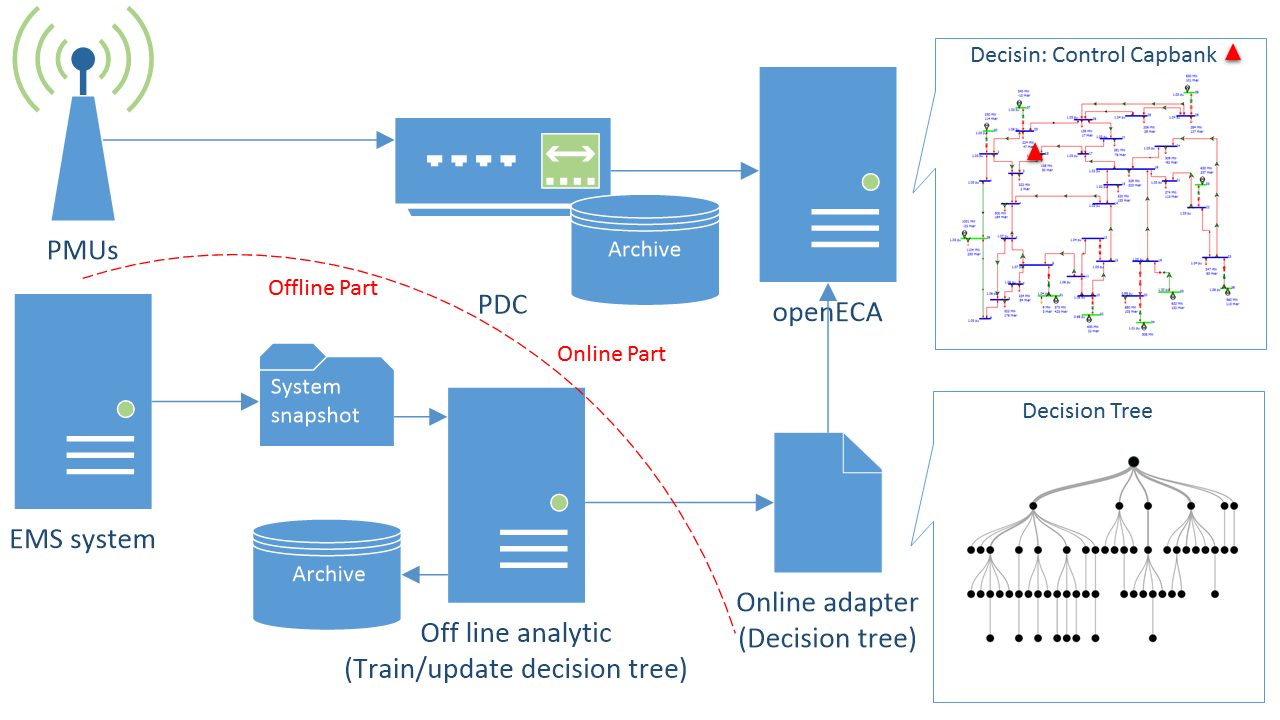


Figure 1 Regional Voltage Controller Analytic Concept

# **Voltage Security Assessment**

VSA is conducted to determine if the system OC is violating operation limits. Also it can be implemented to determine OC’s post-control status [1] [2]. A commonly used method for VSA is the PV curve method. The knee point of a PV curve indicates the maximum power transferred without incurring voltage instability issues. Further increasing the load over the knee point will cause the progressive voltage decline. These knee points are bifurcation points of the nonlinear power system model. Represented by mathematical terms, the loadability limit is corresponds to a scalar function of , where represents the load demand variables . We can consider it as an optimization problem:

(1)

*subject to*

where represents the a set of algebraic equations in algebraic variables, and denotes a set of state variables, for instance, and . To solve this problem, we are able to define the Lagrangian:

(2)

where is a vector of Lagrange multipliers. As a loadability limit, the Jacobian of of the steady state equations is singular, which means the system reaches its loadability limit. As discussed in [1] [3], in an unstable OC, we are left without any information regarding the nature and location of the problem. Therefore, it would be reasonable to create a new boundary called “security boundary” as shown in Figure 1, showing us that the system is approaching collapse. L-index proposed in reference [4] is able to give us a sign showing whether the system is becoming voltage-unstable or not. Reference [5] computes a parameter vector which is able to indicate how far away the system is from the secure operation limit. In this work, the boundary is approximated by finding the insecure OCs whose Euclidean distance to the secure operation limit is less than a predefined threshold value.

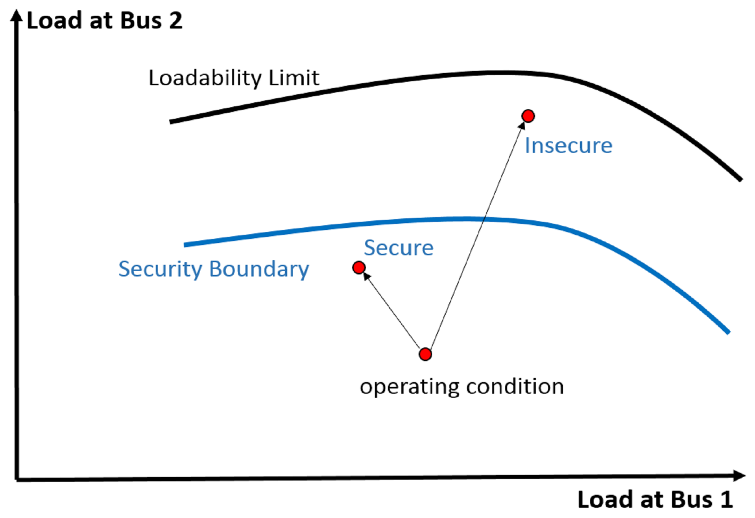


Figure 2 Loadability limit and security boundary

Consider a power system with two load buses, it is assumed that the current operating power at which the corresponding equilibrium is stable. As the load demands vary, the equilibrium varies in the state space. When the increasing load demand reaches a critical value , the system can lose stability. Such a set of critical load demands are denoted as a voltage stability margin . In this study, an index is created to measure the distance between the current OC to the voltage stability margin by computing the shortest Euclidean distance .

(3)

(4)

where . For the cases of , the OCs are considered as insecure OCs. To provide an accurate and timely VSA, the DT scheme is implemented to identify the secure/insecure status of OCs.

# **Decision Tree Based Voltage Security Assessment And Voltage Control**

A decision tree (DT) is an effective data mining tool to handle classification problems. DT is trained based on exhaustive searches on all the features, and it splits the ones whose child nodes have minimal entropy:

(5)

where represents the learning database for the decision tree while and indicates the secure and insecure proportions of. A more detailed concept of decision trees are covered in [6].

1. *Initial Learning Database Preparation*

Different OCs are generated randomly by scaling up bus loads from their original values given by the base case. Any OCs having power flow divergence, exceeding the voltage limits, or violating N-1 contingencies power flow stability are considered as unstable OCs. Besides, using VSA method mentioned in Section 2, those OCs whose Euclidean distance minimum to these unstable OCs are less than a threshold value are considered as insecure OCs. Concerning the aids of PMU based LSE), buses with a voltage base value higher than 230 kV are regarded to be fully observable. There are PMU voltage magnitude measurements for each OC. represents the number of observable buses.

1. *Parallel Trees for Post-Control*

All of the control options in system are treated as independent control decisions. Assuming the system having control options, there will be control combinations (including the one with no control). For system with 4 controlled capacitor banks with one capacitor bank operating at the beginning, the control combination is . The total number of decision combinations would be 16. For each control combination, a learning database is generated. The label for each OC is the post-control OC status, which is also determined by offline VSA. An example learning sample database format is shown as follows:

Table 1 Example learning database

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **OCs** | **Label** |  |  | **…** |  |
| 1 | Secure | 1.02 | 0.98 | … | 1.01 |
| 2 | Insecure | 0.91 | 0.96 | … | 1.03 |
| 3 | Insecure | 0.90 | 0.89 | … | 0.94 |
| … | … | … | … | … | … |
| N | Secure | 1.01 | 1.00 | … | 0.99 |

“Secure” indicates the control is acceptable for this OC with state reflected by the measurements. Therefore, for combinations, there could beparallel decision trees.

1. *Offline Training*

Adaboost is a scheme for improving learning algorithm accuracy by transforming a weak learning algorithm into a strong one. In this study, the adaboost method is utilized to train the ensemble decision tree offline. Each tree is a strong classifier as linearly combined with weak leaners or tree , i.e.

(6)

Given a training set , , with uniform distributed weight , a weak classifier can be trained based on and weights. Based on the training error of the , will be assigned with a voting factor which is computed by

(7)

The weight is increase when the corresponding sample is misclassified, and vice versa. At each boosting iteration, a new weak learner is added in to the strong classifier sequentially until a certain number of weak learners is met.

1. *Decision Tree Update*

Frequent topology changes in power system result in the difference between the actual system operating conditions and the initial learning sample database. To guarantee the reliability of trained decision tree, it is necessary to incorporate new available training cases. Re-training the whole decision trees from scratch might is not as a cost-effective way. In this paper, an ensemble method [7] widely used for computer vision is implemented to update the classifier in an online manner.

The online boosting algorithm is designed to correspond to the offline Adaboost method. In the online boosting algorithm pseudocode shown in Figure 2, represents the set of weak learners trained so far, denotes the latest arriving case, and is the algorithm that returns an updated weak classifier based on training sample and current hypothesis. In this case, the weak classifier is updated using methodology suggested in [8]. The new coming example’s weight is set as . denotes the sum of correctly classified example while represents sum of wrongly classified examples have seen so far at stage . is serving as a selector that picking the from the weak classifier pool based on the misclassification rate. The final strong classifier is a linear combination of selectors.

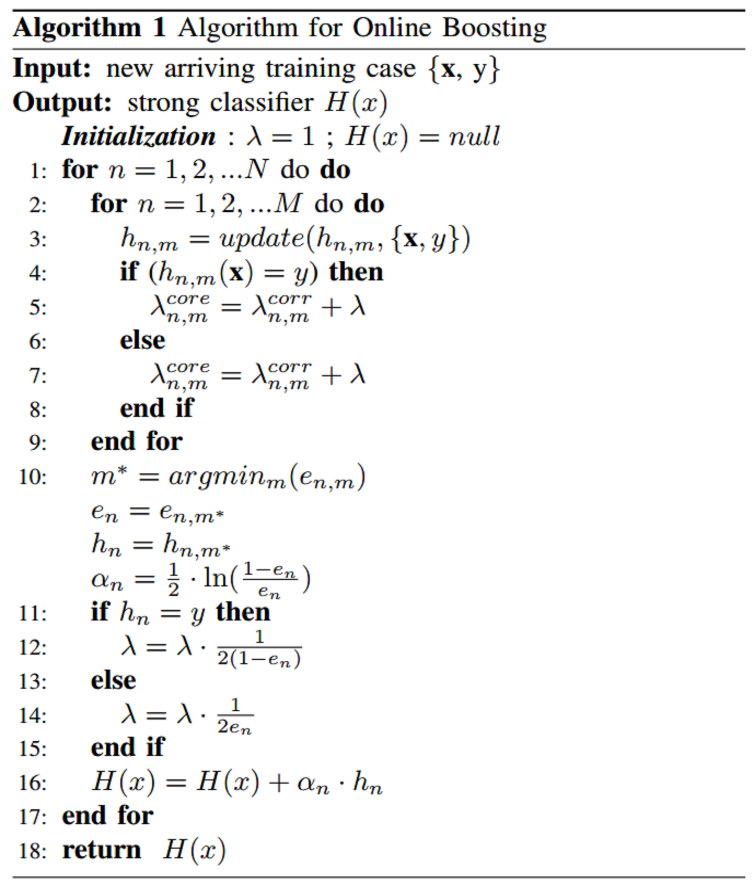


Figure 3 Online boosting algorithm

1. *Online Application*

Voltage measurements from PMUs and LSE collected in real-time continuously provide snapshots of system for each time stamp. Collected measurements will fall into the first tree for VSA only, and it will be compared with the critical splitting nodes inhere in the tree. If the terminal node indicates the current OC is “insecure”, the rest of the trees will be activated simultaneously. Otherwise, if the tree provides “secure” control decision, the control decision with the tree is selected; if more than one tree provide “secure”, the tree with fewer operations involved is selected. The concept of parallel-tree-based control is shown in Figure 3.

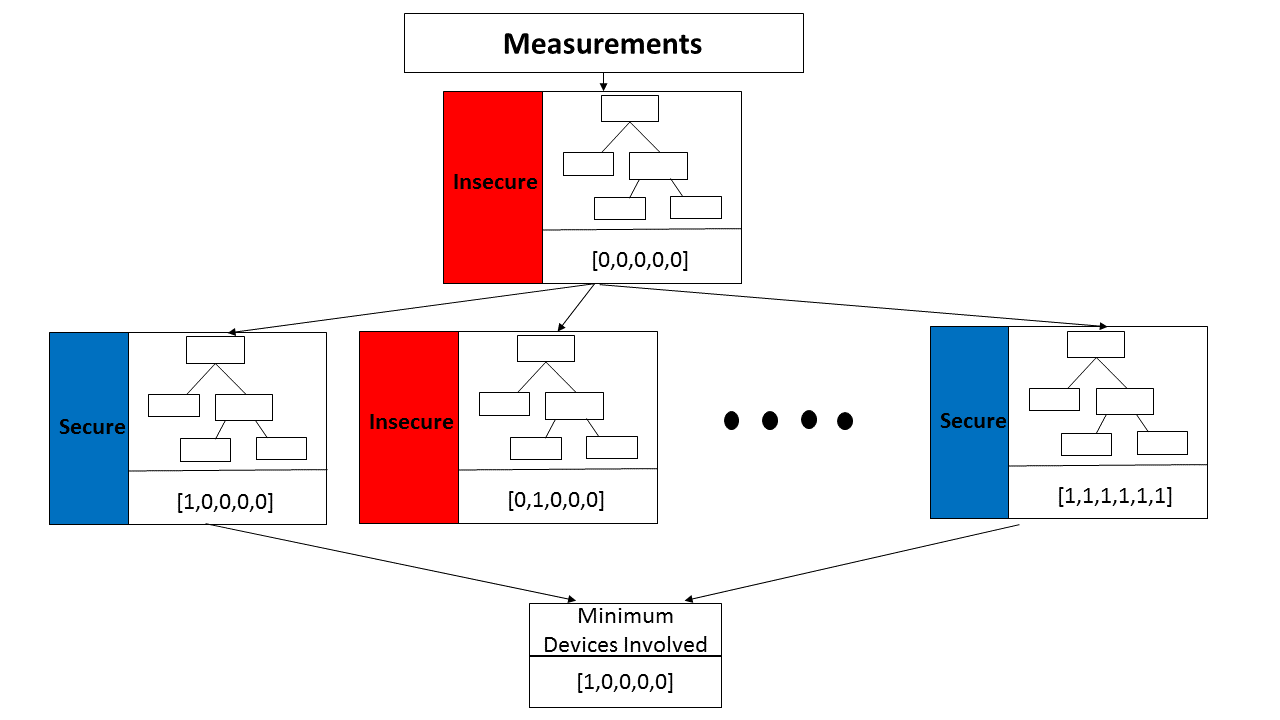


Figure 4 Logic of parallel tree based voltage control

# **Test**

*Test 1: Parallel Trees*

***Code***

*Under folder* ***Test1\ TrainingCaseCreation***

*Run* ***IEEE118\_voltage\_violation\_load\_genchange\_measurements.py*** *to create measurements:* ***databaseMeasurements***

*Run* ***IEEE118\_voltage\_violation\_load\_genchange\_LabelsOC\_CapbankSwitch.py*** *to create labels for all 64 control combinations (index 0 represents there is no control)*

*For example:* ***database\_OC\_PostControl\_0.csv***

*Copy 64* ***database\_OC\_PostControl\_#.csv*** *files to folder :* ***Test 1\ParallelDecisionTrees***

*Under Folder:* ***Test 1\ParallelDecisionTrees***

*Change* ***dir*** *for both MATLAB script* ***Main\_CreateDatabase.m*** *and* ***Main\_CrossValidationDecisionTree.m***

*Run* ***Main\_CreateDatabase.m*** *to create OCs*

*Run* ***Main\_CrossValidationDecisionTree.m*** *to do crossvalidation*

1. *Base Case and OC Generation*

The IEEE 118 bus system is used for case study. The system is divided into 3 areas. Load buses within each area are assumed to have the same loading pattern that the load is scaled up and down in the same percentage. The generator is re-dispatched the same amount of the load as the load changed within the same area. The base case is generated by scaled down 5 percent of the total load.

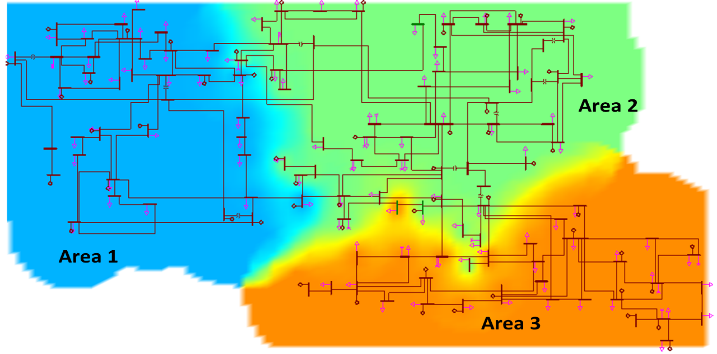


Figure 5 IEEE 118 bus system

Voltage magnitudes at all buses are selected for learning database generation. In this case, the control options are fixed capacitor banks only which are located at buses 34, 44, 45, 48, 74, and 105. In the initial condition, all selected fixed capacitor banks are switched off.

Table 2 Capacitor Bank Available for Control

|  |  |
| --- | --- |
| **Bus Number** | **Capacity of Capacitor Bank (MVar)** |
| 34 | 50 |
| 44 | 50 |
| 45 | 50 |
| 48 | 100 |
| 74 | 100 |
| 105 | 20 |

Overall, 25000 OCs are generated by scaling up the loads within 100% - 150% of their base case value for each area. The outputs of generators re-dispatch the same amount of load in the same area. VSA is implemented to determine the secure and insecure OCs. In this work, it is assumed that the load capacity limit and the secure operation limit mentioned in section 2 are overlapping. The unstable OCs are removed from the initial database, since they cannot provide useful information about the system condition. For all of these secure/insecure OCs, 60% of them are used for training while the rest of them are reserved for periodic update and testing. The initial trees are trained offline. For example, in the database for switching cap bank at 44 on, the number of secure and insecure OCs are shown in Table 3.

Table 3 Number of secure/insecure OCs

|  |  |  |
| --- | --- | --- |
| **OC** | **Training** | **Testing and Update** |
| **Secure** | 9948 | 6615 |
| **Insecure** | 2199 | 1483 |

Their cross validation accuracies for all 63 control combinations are shown Figure 5. As it can be seen, the low error rates of cross validation indicate that the trained tree is able to provide accurate VSA for each control decision.

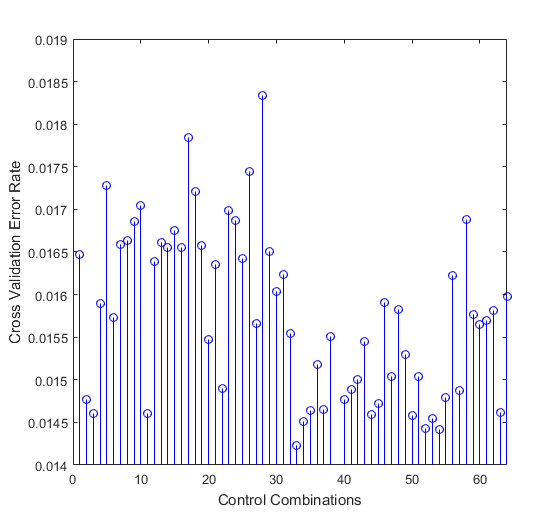


Figure 6 Cross-validation error rate

*Test 2: Online Boosting*

***Code***

*Under folder* ***Test2\ TrainingCaseCreation***

*Run* ***IEEE118\_voltage\_violation\_load\_genchange\_measurements.py*** *to create measurements*

*Run* ***IEEE118\_voltage\_violation\_load\_genchange\_LabelsOC\_CapbankSwitch.py*** *to create labels for all 64 control combinations (index 0 represents there is no control)*

*Copy the csv files:* ***database\_OC\_PostControl\_16.csv*** *and* ***databaseMeasurements\_topologyChange.csv*** *to folder:* ***Test2\OnlineBoosting***

*Copy the .mat files:* ***xtrain.mat, xtest.mat,ytrain.mat,and ytest.mat*** *from folder* ***Test1\ParallelDecisionTrees\case\_16*** *to folder* ***Test2\OnlineBoosting***

*Under folder* ***Test2:OnlineBoosting***

*Run* ***Main\_CreateDatabase.m*** *to create post-topology change test and training data as* ***xtest\_afterTopologyChange.mat, xtrain\_afterTopologyChange.mat,***

***ytest\_afterTopologyChange.mat,*** *and* ***ytrain\_afterTopologyChange.mat***

*Run* ***Main\_initializeTreeStumps.m*** *to create* ***TreeStumps.mat***

*Run* ***Main\_onlineBoost.m*** *to update the Adaboost trees*

*(To shorten the simulation time, the variable in* ***Main\_onlineBoost.m****:* ***NumberofDataforTreeUpdate*** *is set as 1000, The maximum you can set is 4000)*

1. *Periodic Update Using Online Boosting*

In this section, control decision by switching on capacitor bank at bus 44 is selected for classifier performance evaluation. The initial tree is trained based on the offline Adaboost method [9] incorporated with 30 weak learners, and the number of selectors is also 30.

Transmission line between bus 15 and 33 is tripped on the test system. New training cases and test cases are created using the proposed approach in the previous sections but with a different system topology. 4000 of these new cases are used for the periodic update, and another 4000 of them are reserved for online validation. Among these new cases, 82% of them are secure OCs while the rest of them are insecure OCs. The performance of online boosting approach is evaluated by comparing it with single decision tree training using default MATLAB tree training. The computation time and misclassification error rate are recorded and illustrated in Figure 6. The online boosting scheme turns out to be more accurate than single DT training while the computation time spent by online boosting for tree update is much less than re-training tree from scratch. The computation is run under the environment of MATLAB on a workstation with Intel Core i7-4790 3.6 GHz CPU and 32 GB memory.

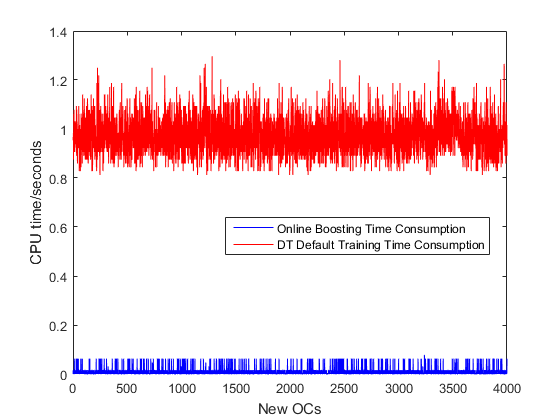


Figure 6 (a) Computation time for tree update

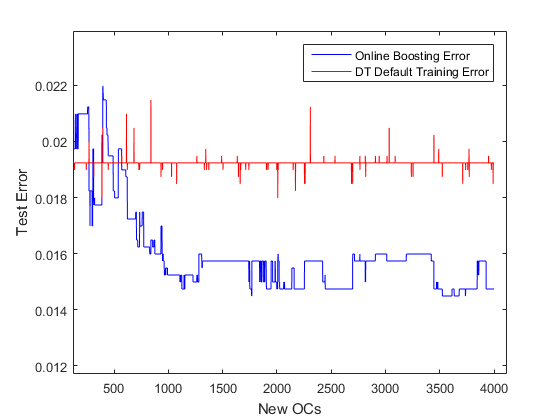


Figure 6 (b) Test error rate for online boosting and single DT training

# **Conclusion**

In this study, an adaptive decision-tree-based systematic method for open-loop regional voltage control is developed. The proposed scheme employs proposed the VSA approach to generate a DT training sample database for each control combination. The online-boosting method is implemented to adaptively update the trained decision trees. This approach is evaluated based on IEEE 118 bus system. The cross validation scheme shows that the parallel trees trained for different control combinations can predict the post-control security status for OC in a low error rate. Finally, a topology change scenario is created to test the online boosting scheme. Simulation result shows that the proposed method is able to reduce the computation burden and have a lower misclassification error compared with the default decision tree training method.

# **Reference**

[1] E. E. Bernabeu, J. S. Thorp and V. Centeno, "Methodology for a Security/Dependability Adaptive Protection Scheme Based on Data Mining," *IEEE TRANSACTIONS ON POWER DELIVERY ,* vol. 27, no. 1, pp. 104-111, 2012.

[2] L. Chengxi, "A systematic approach for dynamic security assessment and the corresponding preventive control scheme based on decision trees," *Power Systems, IEEE Transactions,* vol. 29, no. 2, pp. 717-730, 2014.

[3] D. Ruisheng, S. Kai, V. Vijay, O. R. J, R. M. R, N. Bhatt, S. Dwayne and S. S. K, "Decision Tree-Based Online Voltage Security Assessment Using PMU Measurements," *IEEE TRANSACTIONS ON POWER SYSTEMS*, vol. 24, no. 2, pp. 832-839, 2009.

[4] P. Kessel and H. Glavitsch, "Estimating the Voltage Stability of a Power System," *IEEE Transaction on Power Delivery,* Vols. PWRD-1, no. 3, pp. 346-354, 1986.

[5] S. Shukla and L. Mili, "A hierarchical decentralized coordinated voltage instability detection scheme for SVC," in North American Power Symposium , Charlotte, NC, USA, 2015.

[6] B. Leo, J. Friedman, C. Stone and R. Olshen, Classification and regression trees, CRC press, 1984.

[7] H. a. H. B. Grabner, "On-line boosting and vision," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[8] M. He, Z. Junshan and V. Vijay, "Robust online dynamic security assessment using adaptive ensemble decision-tree learning," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4089-4091, 2013.

[9] D.-J. Kroon, "MathWorks," 01 Jun 2010. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/27813-classic-adaboost-classifier. [Accessed 1 11 2016].