An Enhanced Transmission Operating Guide Creation Framework Using Machine Learning Techniques

Jinye Zhao, Slava Maslennikov, Eugene Litvinov
Business Architecture & Technology, ISO New England
Holyoke, MA, USA
jzhao@iso-ne.com , smaslennikov@iso-ne.com ,
elitvinov@iso-ne.com

Xinbo Geng
Department of Electrical & Computer Engineering
Texas A&M University, College Station, TX, USA
xbgeng@tamu.edu

***Abstract*—Transmission Operating Guide (TOG) is a part of constraint management in power system real-time operations and short-term planning. TOGs contain instructions on interface limit values to be used as transmission constraints for specific operating conditions. Current TOG creation process is manual and time-consuming. It heavily relies on engineer's subjective judgment. This could lead to overlooking of some factors that are critical to interface limits as well as their inaccurate estimation, which might jeopardize the system security or underutilize available transfer capabilities. To address these issues, an enhanced TOG creation framework using machine learning techniques is proposed. The advantages of the proposed framework are demonstrated on two datasets of the ISO New England System. Simulation results show that (1) the proposed ensemble feature selection approach can effectively identify power system features that significantly impact interface limits, and (2) decision tree enables accurate prediction of interface limits while maintaining a simple representation of TOGs.**

***Index Terms*—interface limits, machine learning, feature selection**

I. Introduction

A. Background

Interface is defined as a pre-selected set of transmission lines and transformers that connecting two regions in a power system. An interface limit indicates the maximum power that can flow across the associated interface with the system meeting its security constraints (including thermal, voltage and stability constraints) for specific operating conditions (e.g. peak/light load, dispatch scenarios, topological conditions). The interface limit is a *proxy* for the security limit, allowing operators to monitor only one parameter – MW flow in the interface against the limit, instead of monitoring multiple individual thermal, voltage and stability constraints in a specific area of the system.

Interface limits depend on many features of operating conditions including but not limited to load levels, topological conditions, generation pattern, availability and status of Remedial Action Schemes (RAS), and status of reactive power source devices. Although there has recently been much interest in developing online dynamic interface limit calculation [1, 2], voltage- and stability-based interface limits are still widely calculated via offline studies in practice. The interface limits resulting from these studies are documented in *Transmission Operating Guides* (TOGs). The limits determined by TOGs are used for real-time operation and short-term planning.

B. Current TOG Creation Process at ISO-NE

At ISO New England (ISO-NE), the existing TOG offline creation process can be summarized as the following three main steps at the conceptual level:

Step 1: Engineers prepare a collection of study scenarios, which representatively reflect a variety of possible operating conditions during the future period of time for which a TOG is created. This set might consist of hundreds of scenarios to cover many possible realizations of preselected system features impacting the studied interface limit.

Step 2: Calculate the interface limit for each scenario through software such as PSSE, VSAT/TSAT. An interface limit is determined as the interface flow during increase of the power transfer from a pre-defined source to a sink until any of pre-determined security criteria is violated.

Step 3: Based on the results obtained in Step 2, engineers approximate the underlying relationship between the preselected system features and the associated interface limit. This step can be thought of as developing a predictive model for the interface limit. Depending on the complicity of the relationship, interface limits can be presented in the form of look-up tables, adders[1] or using natural language descriptions in TOGs.

There are many assumptions or choices made throughout the TOG creation process, which can greatly influence interface limits. For instance, the first step of the TOG creation process is very important as the selected features must

[1] In adder format, adders indicating MW impact of critical power system features on the interface limit, i.e. Interface limit = Base Limit + $\text{adder}_1 \times S_1 + \ldots + \text{adder}_n \times S_n$, where S_i represents the on/off status of a specific network element, such as a transmission line or generator, certain system load level, availability of a specific MVar source, etc.

978-1-5386-7703-2/18/$31.00 ©2018 IEEE

characterize properly a variety of power system operating conditions. Generally, the number of possible features is very large, but most of them have limited effect on the interface limit of interest. In the current practice, engineers select a few critical features mainly based on their experience. The subjectivity of engineers can result in overlooking certain critical features as well as makes it challenging to transfer the selection knowledge from one engineer to another, especially under the situation where time is limited. Furthermore, the existing process typically applies simple linear regression to approximate the relationship between an interface limit and the selected system features in Step 3. The drawback of this approach is that the linear model can be inaccurate because an interface limit and the corresponding features may not follow a linear relationship. Additionally, the lack of a uniform TOG format also makes it error-prone and inefficient for the users to look through multiple TOGs, which can be in different formats, to determine the appropriate interface limits during real-time operation.

C. *Contributions of this paper*

The contributions of this paper are the following:

1) An iterative ensemble feature selection approach is developed to systematically identify critical power system features in Step 1, reducing errors and inconsistency of manual handling;

2) Decision tree is applied to Step 3 to provide a formalized process converting a large set of study results obtained in Step 2 into a simple format allowing accurate prediction of interface limits while maintaining a standardized and simple representation of TOGs.

II. An Enhanced TOG Creation Framework

A. *Overview*

Two machine learning approaches, namely ensemble feature selection and decision tree, are proposed to enhance steps 1 and 3 of the TOG creation process, respectively. More specifically, rather than purely relying on engineering knowledge in Step 1, a large set of features can be first selected by engineers, and then carefully reduced based on the proposed ensemble feature selection method, whose details can be found in Subsection II.B. The proposed ensemble feature selection not only provides a systematic way of finding critical system features, but also can help engineers build a simpler predictive model to approximate interface limits in the downstream step as well.

Given a set of study results containing various scenarios of critical system features and the corresponding interface limits, decision tree is proposed to model the relationship between the interface limit and critical system features in Step 3. Decision tree is a conceptually simple yet powerful machine learning model. It can be applied to both classification and regression problems. There is particularly a great interest in using decision trees in power system security assessment. Much of the existing literature in this area, e.g. [3-5], utilizes decision tree as a classification model to determine security regions in the space of system conditions. One of the primary contributions of this paper is employing a decision tree as a regression model to estimate interface limits. Because power system is a complex system, critical features typically interact in a complicated and nonlinear way to affect interface limits. It is difficult, if not impossible, to use a single linear predictive formula as does the existing practice to capture the interactions. Moreover, decision tree has advantages as compared to other machine learning techniques, such as artificial neural network [4] and support vector machine [7], because of its representation simplicity and rule induction as well as its good accuracy. A decision tree is essentially a series of “if-then” statements, which is well suited for traditional TOGs providing direct and clear instructions on managing interface limits. An introduction of decision tree is provided in Subsection II.C.2.

B. *Ensemble Feature Selection*

Feature selection is finding a subset of features which can efficiently describe the input data while still providing good prediction results. Feature selection involves a feature selection algorithm to perform a search in the space of features as well as a machine learning model to evaluate the significance of each feature measured by the predictive accuracy. The proposed ensemble feature selection includes two algorithms (forward selection and backward selection) and four machine learning models (linear regression, decision tree, decision tree with boosting and random forest). The reason of using the ensemble method rather than a particular one is the following: the performance of different feature selection algorithms and machine learning models vary depending on the size as well as linearity of the studied data. As shown in the numerical results in Section III.C, it is difficult to find a particular feature selection algorithm with a machine learning model that performs consistently. On the other hand, the proposed ensemble method provides more reliable and stable feature selection results by aggregating the feature rankings resulting from the combinations of multiple feature selection algorithms and machine learning models.

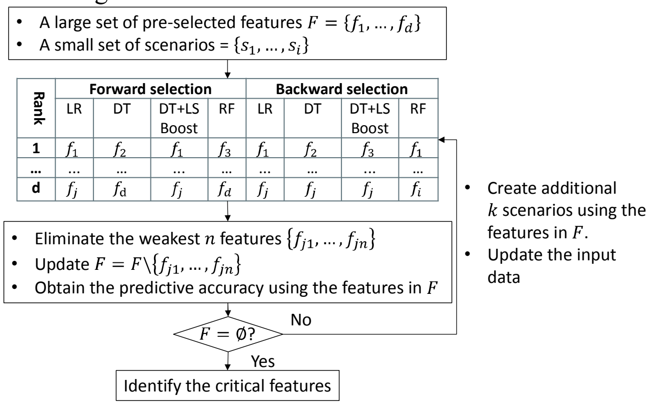
Fig. 1 illustrates the flow diagram of the iterative ensemble feature selection method. The iteration starts with a large set of features preselected by engineers and a small set of scenarios. The combinations of the two feature selection algorithms, which are introduced in Subsection II.C and four machine learning models, which are presented in Subsection II.D yield 8 sets of ranking for the features. The features satisfying the following conditions are considered to be weak among the remaining features thus are eliminated in the next iteration:

(1) The eliminated features belong to 10% of the most insignificant candidates in at least one ranking;

(2) The eliminated features do not belong to top 50% of the most significant candidates of any rankings.

Therefore, the features that are eliminated earlier are less important than the ones eliminated later. Then, we record the predictive accuracy by using only the remaining features.

Before starting the next iteration, additional scenarios are generated by sampling on the remaining features, rather than the original large set of features. Note that the scenarios used in the proposed ensemble method are not pre-generated based on the original large feature set, but are generated on the fly as the important features are revealed. Because of the curse of dimensionality, the amount of scenarios needed to support the feature selection results increases exponentially with the dimension of the feature space. By incrementally generating scenarios as the dimension of the feature space reduces during the iteration, the new scenarios contain more information of the important features. In turn, it increases the accuracy of the feature selection results. The input data for the next iteration is updated by including the new scenarios and removing the eliminated features in the existing data. The iteration stops when there is no more remaining features. Base on the recorded predictive accuracy, a subset of the weakest features that has little or no impact on improving the predictive accuracy is considered to be unimportant, so the remaining features are the critical ones.


• A large set of pre-selected features $F = \{f_1, \dots, f_d\}$
• A small set of scenarios = $\{s_1, \dots, s_i\}$

Rank	Forward selection				Backward selection			
	LR	DT	DT+LS Boost	RF	LR	DT	DT+LS Boost	RF
1	f_1	f_2	f_1	f_3	f_1	f_2	f_3	f_1
…	…	…	…	…	…	…	…	…
d	f_j	f_d	f_j	f_d	f_j	f_j	f_j	f_i

• Eliminate the weakest n features $\{f_{j1}, \dots, f_{jn}\}$
• Update $F = F \backslash \{f_{j1}, \dots, f_{jn}\}$
• Obtain the predictive accuracy using the features in F
$F = \emptyset$?
No
Yes
Identify the critical features
• Create additional k scenarios using the features in F.
• Update the input data

Figure 1: The flow diagram of the ensemble feature selection algorithm

C. *Feature Selection Algorithms*

The forward selection algorithm is a greedy strategy [8]. It starts with an empty selection of features, then it adds each unused feature of the given input data set. For each added feature, its performance is evaluated by the preditive accuracy of a machine learning model defined by the root mean squared error (root MSE) as follows:

$$\text{root MSE} = \sqrt{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

where y_i is the actual interface limit determined in step 2 , and $\hat{y}_i$ is the predicted value by the machine learning model. To avoid overfitting and bias, we conduct 5-fold cross validation, and the resulting expected value of the root MSE is used to indicate the significance of the added feature. The smaller the expected root MSE, the more significant the corresponding feature is. Only the feature giving the highest performance is added to the selection. Then, a new iteration starts with the modified selection. At the end, the forward selection algorithm ranks each feature. Although its greedy nature might lead to suboptimal results, it has a few preferable characteristics such as low computational cost and small variance.

The backward selection algorithm works in the opposite direction to the forward selection algorithm [8]. The backward selection algorithm starts with the full set of features, and it removes the least useful features evaluated by a machine learning model at each iteration.

D. *Machince Learning Models*

We consider four machine learning models in the proposed ensemble feature selection method: linear regression, decision tree, decision tree with boosting and random forest. The following introduces the key idea behind each method and briefly discusses the pros and cons.

1) Linear Regression (LR)

LR model assumes a linear relationship exists in the input data. Although LR might be the most widely used model in practice due to its simplicity, the predictive accuracy can be poor if the underlying relationship of the data is non-linear.

2) Decision Tree (DT)

DT recursively partitions the feature space into many rectangles, and then fits a simple model in each one. There are two popular choices of decision tree algorithms: Classification and Regression Tree (CART) [9], which is being used in this paper, and C4.5 [10]. The basic idea behind these algorithms is “divide and rule.” A DT model consists of a set of rules to divide heterogeneous data into smaller and more homogenous subgroups of data with similar predicted outcomes. The rules mimic the way humans think and solve problems.

As a matter of fact, the hierarchical nature of DT fits well with the rule based TOGs. Therefore, DT not only is used in the proposed ensemble feature selection but is applied to step 3 of the TOG creation process to develop a predictive model for interface limits. More specifically, a DT model is presented as a standardized look-up table containing the conditions of critical features that define each homogenous subgroup of data as well as the corresponding predicted interface limits in each subgroup. An example of the proposed standardized TOG format can be found in Subsection III.D.

3) Decision Tree with Boosting

Boosting is one of the most powerful learning ideas introduced in the last twenty years. The key idea behind boosting is combining the outputs of many “weak” models to produce a powerful “committee” [13]. Here the weak model can be a two terminal-node DT. In this paper, we use “LSBoost” method proposed in [11], which is a gradient boosting method with least square loss function.

4) Random Forest (RF)

RF, developed by Leo Breiman [12] is a substantial modification of bagging that builds a large ensemble of de-correlated trees. Prediction is made by aggregating the predictions of the ensemble. RF generally exhibits a prediction accuracy improvement and variance reduction as compared to a single decision tree. However, the random forest model is difficult to interpret due to its ensemble structure. For this reason, it is not considered in TOG formation process, which often requires clear and intuitive rules.

III. Case Study

A. Datasets

Two interface limit datasets created by the ISO-NE engineers are used to demonstrate the effectiveness of the proposed ensemble feature selection. The first dataset contains 8 features, namely A1, B1,…, H1, which are preselected by the engineers as the critical features candidates for interface limit 1. Similarly, the second dataset contains 11 preselected features, namely A2, B2,…, K2 for interface limit 2.

B. Benchmark

In this subsection, the critical features datasets #1 and #2 are determined by using the best subset feature selection algorithm and the machine learning model with the best predictive accuracy. The critical features identified in this subsection will be used as benchmarks to evaluate the performance of the proposed ensemble feature selection in the next subsection.

The best subset algorithm is a brute-force method which exhaustively evaluates all possible combinations of the input features, and then finds the best subset [13]. Although this algorithm could provide optimal feature selection results, it is computationally expensive when the number of features is large. Therefore, the best subset algorithm is only used as the benchmark feature selection algorithm.

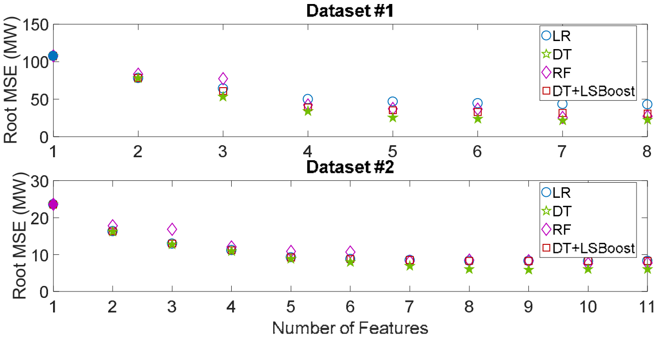


Figure 2: The results of the best subset algorithm with four machine learning models

To find the machine learning model with the best predictive accuracy for each dataset, we test all four machine learning models presented in Subsection II.C with the best subset algorithm. Unlike the proposed ensemble feature selection method where scenarios are generated on the fly, a large set of scenarios (1046 scenarios for dataset #1 and 860 scenarios for dataset #2) are pre-generated in the benchmark process. Fig. 2 shows the predictive accuracy of each machine learning model measured by the root MSE as the number of features considered in the subset increases from 1 to the maximum number of features. The shapes with solid fill highlight the most accurate machine learning models at different levels of number of features. It can be observed from Fig. 2 that DT outperforms others in most of the cases in both datasets, so the best subset of features determined by DT is used as the benchmark. More specifically, the root MSE of DT decreases significantly when the number of feature increases from 1 to 5 in dataset #1, but the incremental accuracy gain remains stable beyond five features. Therefore, the three weakest features are removed, and the remaining five features are considered to be critical. Similar observations can be obtained for dataset #2, so six features are retained as critical ones in dataset #2. The benchmark critical features for datasets #1 and 2 are summarized in Table 1.

Table 1: The benchmark critical features

Dataset	Critical features[2]	Removed features
#1	A1, B1, C1, D1, E1	F1, G1, H1
#2	B2, C2, E2, G2, I2, J2	A2, D2, F2, H2, K2

C. Performance of Ensemble Feature Selection

In this subsection, we first demonstrate that using only a single feature selection method and machine learning model may not always yield correct critical features. Then, the proposed ensemble feature selection is applied to datasets #1 and 2 to demonstrate its effectiveness in determining the critical features.

Given that DT is shown to be the best benchmark machine learning model in the previous subsection, we employ the forward selection algorithm with DT to identify the critical features in datasets #1 and #2. The order of features shown in Fig. 3 indicates their importance. The feature in rightmost of the plot is the least important one, and the importance of the features increases as it moves towards left. Fig. 3 also reports the root MSE of DT, which is represented by blue dots. The root MSE increases as the number of features reduces from the maximum to 1 by removing the corresponding unimportant features. For dataset #1, the three least important features are G1, F1 and H1. The contribution of these three features to improving the root MSE is minimal. Therefore, they can be removed. This outcome is consistent with the benchmark result of dataset #1 in Table 1.

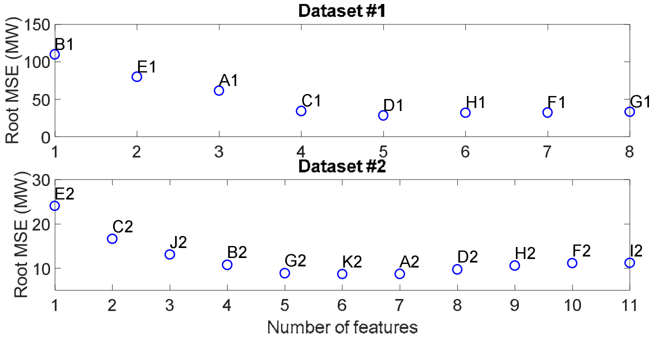


Figure 3: The results of the forward selection algorithm with DT

On the other hand, the forward selection algorithm with DT does not perform well on dataset #2. The bottom plot in Fig. 3 indicates that the six least important features, I1, F2, H2, D2, A2 and K2 should be removed. Compared to the benchmark results in Table 1, I2 is mistakenly identified as the least important features by the forward selection with DT whereas it is one of the critical features in the benchmark case. The unstable performance can also be observed from

[2] The order of the critical features does not represent the level of significance.

other combinations of the feature selection algorithms and machine learning models. This implies that there is no universal algorithm and model that can reliably identify critical features on any data sets.

Although an individual feature selection algorithm and machine learning model are not robust, aggregating them into the ensemble feature selection method can lead to consistent and effective results. This can be demonstrated by applying the ensemble feature selection to the datasets and the results are illustrated in Fig. 4. For dataset #1, the three least important features, G1, F1 and H1 are removed by the proposed method as they have little impact on the predictive accuracy. Similarly, the five least important features, A2, D2, H2, F2 and K2 are removed in dataset #2. The ensemble feature selection algorithm yields the identical outcomes to the benchmark results.

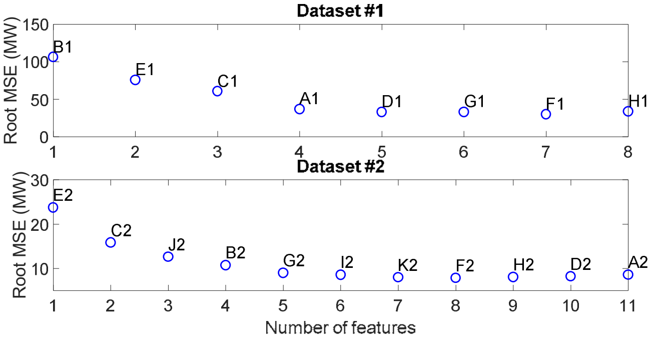


Figure 4: The results of ensemble feature selection method

D. A Simple TOG Format Using Decision Tree

After the critical features are identified in Subsection III.C, DT is applied to the dataset that contains only the critical features to model the relationship between the interface limit and critical features. In this subsection, dataset #1 is used to demonstrate the simple TOG format that DT generates.

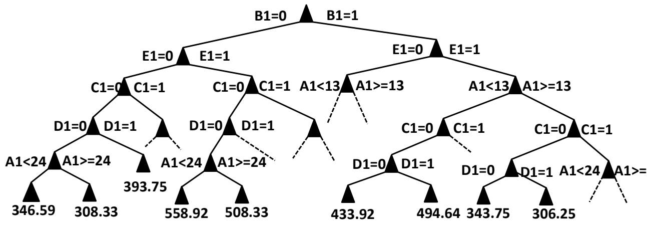


Figure 5: Graphic representation of DT for dataset #1

Fig. 5 illustrates the graphic representation[3] of the resulting DT for dataset #1. Each path of the tree is a series of "if-then" statements. The nodes at the bottom of the tree return the prediction for the interface limit #1.

Table 2: Standardized TOG format for dataset #1

A1	B1	C1	D1	E1	Interface limit #1 (MW)
<24	0	0	0	0	346.59
>=24	0	0	0	0	308.33
<24	0	0	0	1	558.92
…	…	…	…	…	…

The proposed TOG format is a look-up table, as shown in Table 2. It is an alternative representation of the DT. Each row in Table 2 represents a branch of the tree in Fig. 5. The look-up table is simple and straightforward. For example, the the first row of Table 4 indicates that if A1 is less than 24 MW, B1, C1, D1 and E1 are unavaiable, then the interface limit #1 is set to be 364.59 MW. The proposed TOG format also makes it easy to develop an automatic process to transfer TOGs into real-time production software, increasing efficiency and reducing the errors resulting from the users manually looking up the existing TOGs. Additionally, compared to the existing TOG, where LR is used to estimate interface limits, DT provides a more accurate approximation. The root MSE of the existing interface limit #1 TOG is 46.87 MW using the LR approach. The root MSE is 25.92 MW by using DT, improving accuracy by 45%. This indicates that DT can potentially decrease the inefficent usage of interface #1 by almost half.

IV. Concluding Remarks

In this paper, we propose an enhanced TOG creation framework using machine learning techniques. The proposed framework improves the TOG creation process in the following aspects: (1) effectively identify the critical power system features; (2) develop simple standardized TOG format to more accurately estimate interface limits. The proposed framework is a more systematic and reliable approach to creating TOGs. It can improve the efficiency of the process of creating and using TOGs, reduce human errors of manual handling, and improve interface limit quality.

References

[1] http://www.powertechlabs.com/software-modeling/dynamic-security-assessment-software/ [Available online].

[2] http://www.vrenergy.com/index.php/powersystemsoftware/rose.html [Available online].

[3] L. Wehenkel, M. Pavella, E. Euxibie, and B. Heilbronn, "Decision tree based transient stability method a case study," IEEE Transactions on Power Systems, vol. 9, no. 1, pp. 459-469, 1994.

[4] S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown, "Decision trees for real-time transient stability prediction," IEEE Transactions on Power Systems, vol. 9, no. 3, pp. 1417-1426, 1994.

[5] I. Genc, R. Diao, V. Vittal, S. Kolluri, and S. Mandal, "Decision tree-based preventive and corrective control applications for dynamic security enhancement in power systems," IEEE Transactions on Power Systems, vol. 25, no. 3, pp. 1611-1619, 2010.

[6] N. Amjady, "Dynamic voltage security assessment by a neural network based method", Electric Power Systems Research, 2003.

[7] S. Kalyani, and K. S. Swarup, "Classification and assessment of power system security using multiclass SVM," IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews, vol. 41, no. 4, pp. 753-758, 2011.

[8] I, Guyon and A. Elisseeff. "An- introduction to variable and feature selection." Journal of machine learning research3, pp. 1157-1182, 2003.

[9] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression tree," CRC Press, 1984.

[10] J. R. Quinlan, "C4.5: Programs for machine learning," San Mateo: Morgan Kaufmann, 1993.

[11] J. Friedman, "Greedy function approximation: a gradient boosting machine." Annals of statistics, pp. 1189-1232, 2001.

[12] L. Breiman, "Random forest," Machine learning, 2001.

[13] J. Friedman, T. Hastie, and R. Tibshirani, "The elements of statistical learning," New York: Springer series in statistics, 2001.

[3] Some braches of the DT are not explicitly plotted for the purpose of the simplicity of the illustration.