# Predicting Electricity Distribution Feeder Failures Using Machine Learning Susceptibility Analysis

**Philip Gross, Albert Boulanger, Marta Arias, David Waltz, Philip M. Long[1], Charles Lawson[2], Roger Anderson**

Columbia University
Center for Computational Learning Systems
New York, NY 10115
{philg | aboulanger | marta | waltz | plong | lawsonc | anderson}@ccls.columbia.edu

**Matthew Koenig, Mark Mastrocinque, William Fairechio, John A. Johnson, Serena Lee, Frank Doherty, Arthur Kressner**

Consolidated Edison Company of New York
New York, NY 10003
{koenigm | mastrocinquem | fairechiow | johnsonj | leese | dohertyf | kressnera}@coned.com

[1] Currently at Google Inc, Mountain View, CA.
[2] Consultant, Digimedia Corp, Austerlitz, NY

## Abstract

A Machine Learning (ML) System known as ROAMS (Ranker for Open-Auto Maintenance Scheduling) was developed to create failure-susceptibility rankings for almost one thousand 13.8kV-27kV energy distribution feeder cables that supply electricity to the boroughs of New York City. In Manhattan, rankings are updated every 20 minutes and displayed on distribution system operators' screens. Additionally, a separate system makes seasonal predictions of failure susceptibility. These feeder failures, known as "Open Autos" or "O/As," are a significant maintenance problem. A year's sustained research has led to a system that demonstrates high accuracy: 75% of the feeders that actually failed over the summer of 2005 were in the 25% of feeders ranked as most at-risk. By the end of the summer, the 100 most susceptible feeders as ranked by the ML system were accounting for up to 40% of all O/As that subsequently occurred each day. The system's algorithm also identifies the factors underlying failures which change over time and with varying conditions (especially temperature), providing insights into the operating properties and failure causes in the feeder system.

## Background

Electrical infrastructure has four main parts:

1. **Generation:** a prime mover, typically the force of water, steam, or hot gasses on a turbine, spins an electromagnet, generating large amounts of electrical current at a generating station

This work has been partly supported by a research contract from Consolidated Edison.


2. **Transmission:** the current is sent at very high voltage (hundreds of thousands of volts) from the generating station to substations closer to the customers
3. **Primary Distribution:** electricity is sent at mid-level voltage (tens of thousands of volts) from substations to local **transformers**, over cables called **feeders,** usually 10-20 km long, and with a few tens of transformers per feeder. Feeders are composed of many **feeder sections** connected by **joints** and **splices**
4. **Secondary Distribution:** sends electricity at normal household voltages from local transformers to individual customers

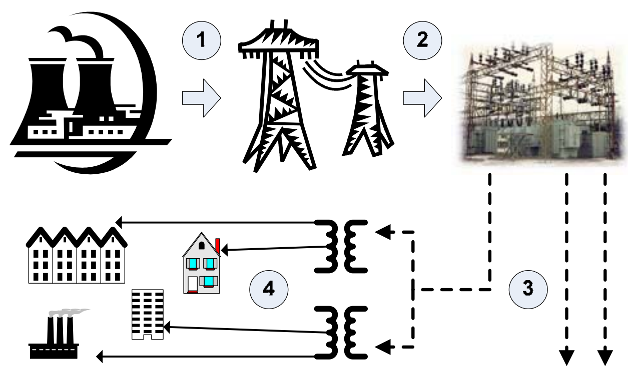


**Figure 1. Electrical Distribution**

The distribution grid of New York City is organized into **networks**, each composed of a substation, its attached primary feeders, and a secondary grid. The networks are largely electrically isolated from each other, to limit the cascading of problems.

The feeders of the primary grid are critical and have a significant failure rate (mean-time between failure of less than 400 days), and thus much of the daily work of the Consolidated Edison Company of New York (Con Edison) field workforce involves the monitoring and maintenance of primary feeders, as well as their speedy repair on failure.

In the specific case of the underground distribution network of Con Edison, transmission lines deliver electricity into the city from upstate New York, New Jersey and Long Island, as well as from in-city generation facilities. Substations reduce the voltage to 33kV or less, and underground primary distribution feeders then locally distribute the electricity to distribution transformers. From there, the secondary network, operating at 120V/208V, delivers electricity to customers. Our work focused on 941 underground primary feeders, distributing electricity to the New York City boroughs of Manhattan, Brooklyn, Queens, and the Bronx.

## Problem

The underground distribution network effectively forms a 3-edge-connected graph – in other words, any two components can fail without disrupting delivery of electricity to customers in a network. Most feeder failures result in automatic isolation – called "Open Autos" or O/As – and many more occur in the summer, especially during heat waves when power use for air conditioning adds to the load. When an O/A occurs, the load that had been carried by the failed feeder must shift to adjacent feeders, further stressing them. O/As put networks, control centers, and field crews under considerable stress, especially during the summer, and cost millions of dollars in Operations and Maintenance (O&M) expenses annually.

One of Con Edison's primary goals has always been reliability, and its distribution network is the most reliable in the United States, and among the best in the world. The goal is to maintain this standard of excellence even as electricity consumption continues to grow. Over the years, they have made enormous investments in manpower and money to collect and analyze vast amounts of data from their systems. The ROAMS system described in this paper is a continuation of Con Edison's striving towards the most reliable and efficient system possible, and it is built on the foundation of the data and insights they have amassed over decades.

We have initially focused on the specific problem of ranking primary distribution feeders according to their susceptibility to failure. Con Edison has made considerable efforts to identify, test and replace failure-prone feeders in the fall and spring of each year, but overall numbers of failures (normalized to weather conditions) has remained fairly constant. A number of possible explanations have been offered for this: the increasing loading and continuous aging of the grid are neutralizing the improvements from the maintenance program; and/or the selection of reliability improvement measures needs to be more effective.

The goal of the ROAMS system described here is to rank the feeders most susceptible to impending failure with sufficient accuracy so that timely preventive maintenance can be taken on the right feeders at the right time. Con Edison would like to reduce feeder failure rates in the most cost effective manner possible. Scheduled maintenance avoids risk, as work is done when loads are low, so the feeders to which load is shifted continue to operate well within their limits. Targeting preventive maintenance to the most at-risk feeders offers huge potential benefits. In addition, being able to predict incipient failures in close to real-time can enable crews and operators to take short-term preventative actions (e.g. shifting load to other, less loaded feeders).

The work reported here is a key step in Con Edison's long term goal: system-wide evolution from reactive to condition-based preventive maintenance processes and procedures.

## Application Description

We are not aware of previous AI-based systems applied to power distribution problems. There have been a number of efforts to improve the efficiency of complex systems by having a computer interpret a stream of sensor data. However, these systems generally use human-constructed

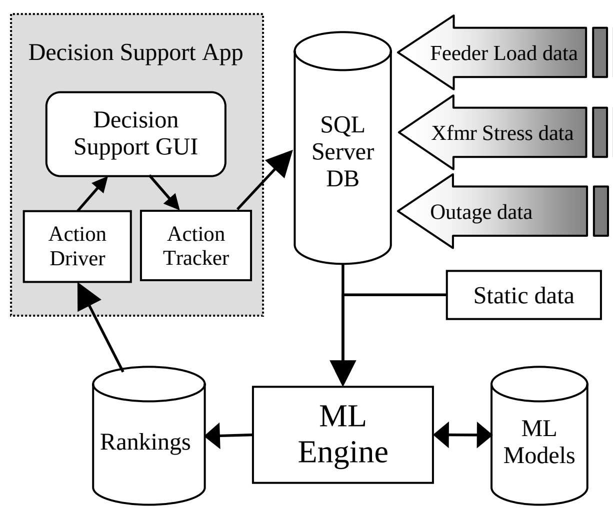


**Figure 2: System Diagram. Incoming dynamic system data at upper right is stored in the main database. The ML Engine combines this with static data to generate and update models, and then uses these models to create rankings, which can be displayed to the operator via the decision support app. Any actions taken as a result are tracked and stored back in the database.**

expert or rule-based systems [7, 8, 9]. In contrast, we have opted for a machine learning system that learns its models entirely from data and hence does not include any human biases. The overall structure of the ROAMS system is illustrated in Figure 2.

A mixture of static data (e.g. age and composition of each feeder section) and dynamic data (e.g. electrical load data for a feeder and its transformers, accumulating at a rate of several hundred megabytes per day) is combined into a large data table. The data table currently has roughly 150 attributes for each feeder, and new ones are still being added.

We then apply a boosting-based machine learning technique to the large data table, which outputs a learned model. We evaluate these generated models by applying them to subsequent test data using a performance metric that captures how high actual feeder failures are positioned in our ranked list of feeders (feeders predicted to have bad susceptibility will be higher in the list). More concretely, our performance metric, which we refer to as the *normalized average rank of failures*, is:

$$1 - \frac{\sum_i rank(failure_i)}{\# failures * \# feeders}$$

For example, suppose that there are 1000 feeders in our system. If there are 3 failures on a given day, and the feeders that correspond to these failures were ranked at 23, 65 and 147 in our list, then the resulting performance is:

1 – ( (23 + 65 + 147) / (3 * 1000) ) = 0.92

Note that a performance close to 1.0 (the ideal result) indicates that almost all actual outages were accounted for at the very top of our worst-feeders ranking. A performance close to 0.0 (the worst possible result) indicates that all actual outages were at the bottom of our ranking. A performance of 0.5 would indicate that our ranking is no better than random.

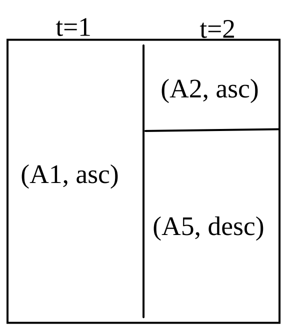


**Figure 3: An example of a Marti model with 2 levels (i.e., T=2): the first level (left) shows the attribute A1 with an ascending sort direction; the second level (right) shows the other two attributes A2 and A5 that are part of the model. Notice that the split of the two lists on which A2 and A5 operate is implicitly described by the height of the limiting line.**

**inputs:** list $L$ of attribute-value descriptions of feeders with associated nr. of failures; nr of boosting rounds $T$
**output:** marti-model $M$

1. **let** M be the empty model
2. **for** each round t=*1,..,T* **do**:
   - partition $L$ into t sub-lists $L_1$, .., $L_t$ s.t. each $L_j$ has same nr. of failures; let $th_2$, .., $th_t$ be the location of the splits in terms of the normalized fraction of feeders that fall above the split.
   - **for** each sub-list *i=1,..,*t **do**:
     i. compute quality of $L_i$ sort
     ii. **for** each attribute $A$ **do**:
        1. sort $L_i$ according to $A$ in ascending order, compute quality of resulting sort
        2. sort $L_i$ according to $A$ in *descending* order, compute quality of resulting sort
   - **if** there exists attribute A and polarity P that improves $L_i$'s sort, **then**:
     i. **if** $i > 1$, add $th_i$ to M at level t, position i
     ii. add A to M at level t, position i.
     iii. sort $L_i$ according to (A,P)
   - **else**:
     i. **if** $i > 1$, add $th_i$ to M at level t, position i
     ii. add "NOP" to M at level t, position i.
3. **output** M

**Figure 4: MartiRank Algorithm.**

When developing seasonal predictions used to guide replacement planning, we gather large amounts of historical data and train a few models based on different random subsets of this data. The final ranking is computed as the average of the rankings produced.

To create the control center operators' near-real-time display, we select a recent, well-performing model as the "current" model. Every 15 minutes, we apply this model to the latest dynamic data, and generate a new ranking. This is the first time that our new MartiRank algorithm, described below, has been used for real-time operational decisions.

An additional feature of this display is action tracking, which presents suggested actions that can be taken by field crews to improve system reliability, and tracks these operator actions. This tracking data is collected in a database and is available as features for further machine learning analyses, to evaluate the effectiveness of field crew intervention. The optimal actions learned will be used to design better preventive maintenance and operational policies for managing the electric grid of New York City.

## Use of AI Technology

The core of the ROAMS system is a machine learning ranking engine whose aim is to produce in real-time a list of the network's feeders sorted from most to least susceptible to failure. To train our models, we use Martingale Boosting [2], a recent variant of boosting. Boosting [3] is a very successful machine learning technique that combines several "weak learners" or simple classification rules, each with low individual accuracy, into a powerful single model with high predictive accuracy. In our case, the weak learners are functions defined by the ranking produced by sorting on a single attribute. A high-level description of our learning algorithm, MartiRank, is given in pseudocode in Figure 4. MartiRank greedily selects in each of its rounds the attribute that is most correlated with the failures in the given training data set. The model records the selected attribute along with the direction of its sort (ascending or descending), which we call "polarity" in Figure 4. In round t (from t=1 up to t=*T*), it splits the total data set into t sub-lists, on which it applies its greedy attribute selection procedure; the list is partitioned so that each sub-list contains the same number of failures. Figure 3 shows an illustration of an example Marti model.

We generate susceptibility rankings from a real-time feeder description list and a trained model by repeatedly sorting the list according to the attributes chosen by the trained model. Intuitively, it is as if we walk the initial list through the model from left to right, reshuffling the list according to the attributes at that level of the model. Using the example model shown in Figure 3, we first sort the whole list according to A1 in ascending order. Then, we split the list into two parts with the top part containing (roughly) 1/4 of the total number of failures and the bottom part containing the rest and we sort the top sub-list following attribute A2 in ascending order and the bottom list using the values of attribute A5 in descending order.

For the sake of comparison, we have been running two other methods: SVM [4] and linear regression. For both methods we use the predictions for each feeder (the predictions are real values) to get a final ranking by sorting on the predictions. As Figure 5 shows, MartiRank clearly outperforms both methods. A further advantage of using MartiRank over the other methods is that our models are easily readable and interpretable by human users. This is absolutely crucial for two reasons: (1) one of the main goals of this application is to learn about root causes of failure so that preventive maintenance procedures can be designed, and (2) operators using our system would not trust a prediction system that they cannot interpret. Notice also that MartiRank has been specifically designed to produce ranking models, whereas most machine learning methods need some kind of ad-hoc postprocessing (such as the ones used with the regression methods mentioned earlier) since the methods are not designed specifically for ranking. Notable recent exceptions are RankBoost [5], and RankNet [10]. Both of these methods address the problem of ranking by learning a real-valued function (used to rank future test examples) from pairs of instances (x,y) s.t. x is ranked higher than y in a given training dataset; RankBoost [5] uses boosting and RankNet [10] uses neural networks to learn the regression function. As no implementation is available for these systems, comparison to these alternatives is currently not feasible within the time constraints. Moreover, RankBoost and RankNet are "black-box" methods, i.e., it is very difficult to interpret and understand the models they produce. The interpretability of its model and the excellent prediction performance obtained so far have made MartiRank ideal for this application.


**Figure 5: This plot shows the performance of three competing methods (MartiRank, SVM, and linear regression) during the second half of 2005. Each day, we compute each algorithm's performance in terms of the normalized average rank of failures that occurred during that day. MartiRank clearly outperforms the other methods. Further, the periods that are critically important are the peak summer months, when the system is near capacity and feeder failures are more frequent, and these are the periods where MartiRank's accuracy is best. A histogram of daily failure counts is shown below the line graph. All curves in the upper graph have been smoothed using a 5-day moving average for clarity of presentation.**

## Application Use

Control center operators use a web-based interface to view the current ranking of feeder susceptibility. It shows the current ranking, and highlights feeders that have changed ranking since the last refresh 15 minutes ago. Additionally, it integrates and displays useful dynamic data collected in the process of assembling training data sets. This includes information on how many of the feeder's sections and transformers are operating close to their limits. The operator's display is shown in Figure 6.

The susceptibility display is designed to work in conjunction with three other existing applications, so operators can take proactive actions based on worst case

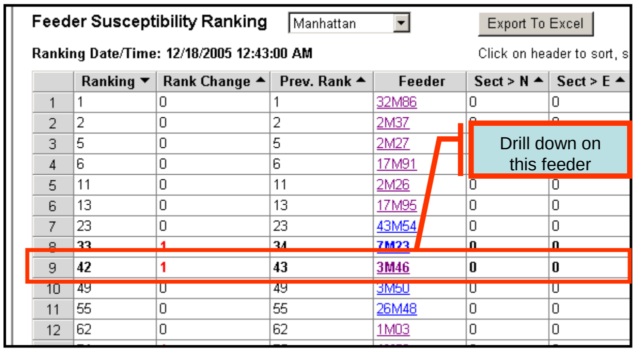


**Figure 6: Operator's display is used with 3 other web applications to act on feeder susceptibility rankings.**

analyses. Action tracking features have been added to these existing applications, enabling the tracking of activities taken on susceptible feeders, and thus allowing the quantification and measurement of resulting network performance improvements. Two of these applications are shown in Figures 7 & 8.

In terms of prediction performance, we found that as summer 2005 became increasingly hotter, our ROAMS System learned the mix of attributes most closely correlated with impending failures, so that by July, we were approaching 0.8 out of a possible 1.0 performance.

Figure 9 shows the ranks of feeders that failed on July 27th, a very hot day with an unusually-high number of failures (28 in total).[3]

During the month of July 2005, none of the 500 feeders ranked lowest for susceptibility failed. More precisely, the worst ranked actual failure during the month of July was 442 (out of a possible of 941). There were a total of 184 failures during the month of July so that probability of this happening by chance is less than $2^{-184}$.

We have a number of theories as to why performance was better during the summer. The first is that many of the input features to our machine learning algorithm were developed by Con Edison with a specific focus on modeling the electric distribution system during heat waves. The second is that distribution system failures may have more deterministic causes during heat waves, as the load and stress contribute directly to cable, joint, and transformer problems, while in the cooler months, failures tend to be more random and difficult to model. However, these are just conjectures, and we are continuing to investigate this question.

[3] Complete statistics with results for the summer can be found at **http://www.cs.columbia.edu/~marta/roger**

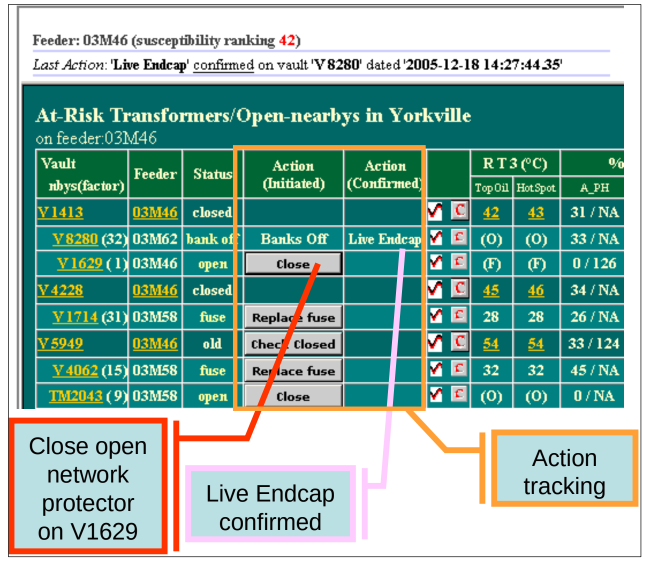


**Figure 7: Action tracking is used to track the actions taken on susceptible feeders to measure their effectiveness**

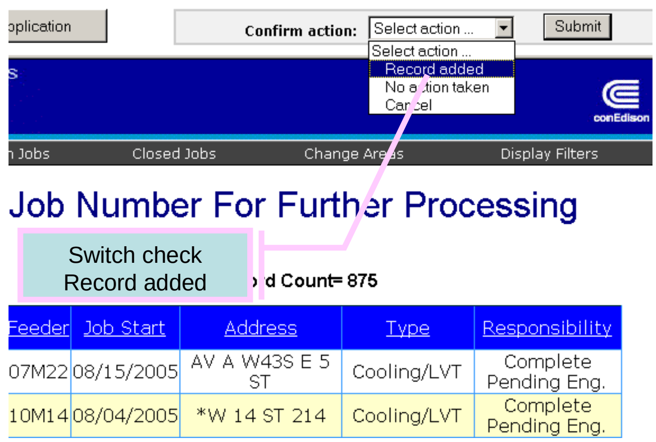


**Figure 8: To close the network protector on the transformer in vault V1629, the Switch Check application is used to open a job to dispatch crew to close the switch. Action Tracking records that the operator confirmed that he added a new switch check job to close the network protector.**

## Payoffs

Reacting to an unscheduled feeder failure has a significant cost in manpower and materials, as well as potential risk of customer power outages. Our ROAMS system is helping to mitigate those consequences in a number of ways.

Our long-term recommendations of feeder susceptibility are helping to decide which feeder components should be replaced during the off-season. Although the results will only become apparent over the coming years, we hope to see failure rates dropping again, resulting in a system that operates in a more cost effective manner.

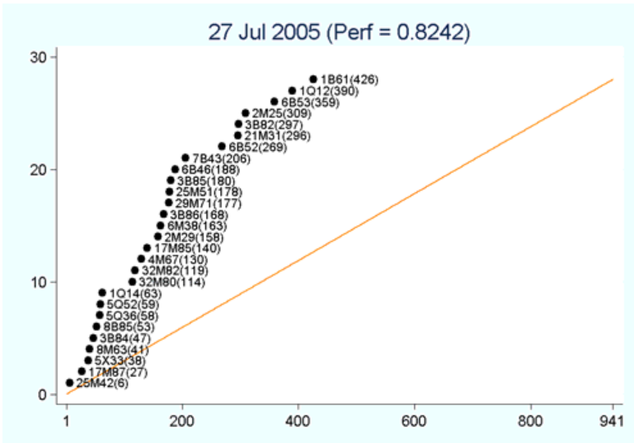


**Figure 9: Failed feeders of 27th July 2005. The x axis shows the rank of the failures, the y axis shows the cumulative number of failures up to that rank.**

Operators are using the short-term prediction capabilities of the current system in a number of ways. In particular, they can now augment their existing information about the *structural* risks of a failure, i.e. what loads will increase and by how much due to the loss of the feeder, with an estimate of the *likely* risks of a failure, i.e. which feeders are actually most likely to fail under the new circumstances.

Con Edison is investigating designs for a new generation of equipment that will allow dynamic switching of current flows. We are working together to develop new machine learning techniques integrated with system-wide models so that as this new equipment comes online, more advanced and proactive responses will be possible for risks identified by our ROAMS System.

The most impressive benefit from our work so far has come from an analysis of the preventive maintenance that Con Edison already does on its feeders. Con Edison has a program to ensure the reliability of its feeders through an in-field test. High DC voltage is put across a feeder for a sustained period (the tests are known as "Hi-Pots"), and any resulting faults are immediately fixed. The tests are effective for "shaking out" latent problems, but are expensive and can put substantial stress on the feeder.

We discovered that our susceptibility ranking was extremely good at predicting which feeders were likely to fail their Hi-Pot. More importantly, those "good feeders" at the low-risk end of our ranking were virtually guaranteed to pass the Hi-Pot, and could safely be skipped.
Con Edison is now including our susceptibility ranking in their evaluation of candidate feeders for Hi-Pot tests this season, and we expect to see substantial savings from a reduction in unnecessary tests, as well as increased reliability from feeders that are tested and pass.

## Development, Deployment, and Maintenance

There were a number of development challenges that had to be overcome before the system could be deployed. Con Edison is a large company, and its data is spread over many repositories. Simply identifying the data that would be useful for the system and then arranging access mechanisms took substantial time (and is still continuing).

As Con Edison has always been active in using advanced technology, many of their software systems have been in use for a considerable amount of time, and retain their original orientation towards human-readable output. We wrote programs, sometimes of substantial complexity, to convert this data back into a computer-usable form.

A later challenge was the volume of data. Our software processes were initially unable to keep up with the gigabytes of data being accumulated on a daily basis. A full reorganization of our underlying database table structure and the addition of a preprocessing stage solved the problem.

Having solved these problems, screens displaying the continuously-updated failure susceptibility rankings for feeders have been in use since mid-2005, about seven months prior to writing this paper. Failure susceptibility rankings to support maintenance scheduling are being used with Fall 2005/Winter 2006 maintenance planning. Results from use of our ROAMS System to-date have given Con Edison the confidence to incorporate our findings into their current preventive maintenance program and to continue the development and deployment of this tool in 2006.

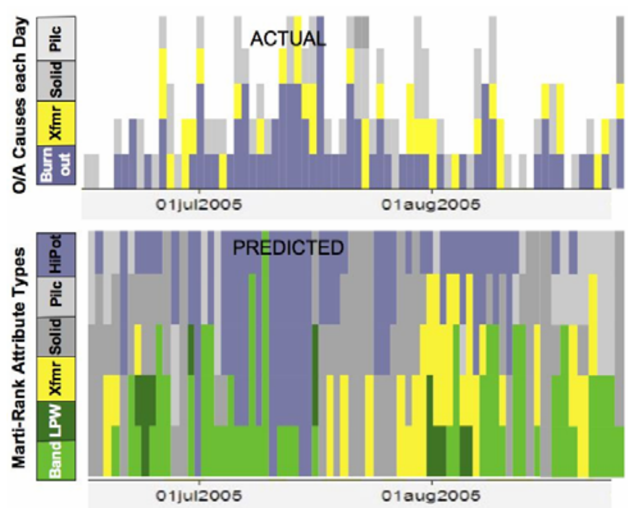


**Figure 10: Feature categories ranked in the top five of the first pass of MartiRank (below) are leading indicators of actual feeder failure causes (above). Categories are shown stacked by type.**

## Future Work

Within the machine learning module there exist several directions for improvement. Our system uses, at any given time, the most recently trained model to make its predictions. It is currently up to the operators to decide when to re-train a model if they observe that performance degenerates. We are currently designing a dynamic online algorithm that will select optimal or quasi-optimal models based on recent past performance of competing models. This next-generation ML engine aims for full automation of the system and removal of assumptions that might be hurting the current performance. Further, it may allow for an interesting knowledge discovery process, as we learn to relate environment and system states to optimal model characteristics.

Additionally, we are starting to investigate the use of our ML system to identify root causes for failures. The first pass of the MartiRank algorithm evaluates each feature individually for its ability to predict outages across the entire set of feeder training data. We have found that if we group these features into general categories (electrical characteristics, transformer stress, cable type, etc.), the top-ranked feature categories from the first MartiRank pass are effective leading indicators of corresponding actual failure causes, with a lead time of around 6 days.

For example, we see a rise in Hi-Pot test related features in our analysis around a week before seeing a corresponding rise in actual feeder failures caused by burnouts, as shown in Figure 10.

We believe that further study of the relationship between ML-identified attributes and actual causes may lead to further improvements in fault management processes and system reliability.

A related prediction problem that arises in the context of our collaboration with Con Edison is that of making quantitative predictions about the time to failure of each feeder and relevant component in the network. This very hard problem could potentially be solved by a mixture of machine learning and statistical techniques such as survival analysis [6]. Although we are not currently focusing on this line of research it is part of our longer-term research plans.

## Acknowledgements

Too many people at Consolidated Edison contributed to the success of this project to name individually. We wish to thank all of those who gave so generously in time, advice, and effort. We would particularly like to recognize the following: MECC Operations: Matt Sniffen; Distribution Engineering: Elie Chebli; Cable and Splice Center of Excellence: George Murray, Bill McGarrigle, and the Van Nest team; Statistical Consulting: Dr. David J. Allen.

Additionally, a number of Columbia faculty, students, staff, and researchers contributed towards this effort: Rocco Servedio, Samantha Cook, Hila Becker, Samit Jain, John Ioannidis, and Sergey Sigelman.

## References

[1] Hanley, J. A. and B. J. McNeil. *The meaning and use of the area under a receiver operating characteristic (ROC) curve*. Radiology 143: 29-36, 1982.

[2] P. Long and R. Servedio. *Martingale Boosting*. Eighteenth Annual Conference on Computational Learning Theory (COLT), 2005, pp. 79-94.

[3] Y. Freund and R. Schapire. *A decision-theoretic generalization of on-line learning and an application to boosting*. Journal of Computer and System Sciences, 55(1):119–139, 1997.

[4] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.

[5] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. *An efficient boosting algorithm for combining preferences*. Journal of Machine Learning Research, 4:933–969, Nov. 2003.

[6] D. Kleinbaum and M. Klein. *Survival Analysis: A Self-learning Text*. Springer, 1997.

[7] Rob Wylie, Robert Orchard, Michael Halasz, François Dubé. *IDS: Improving Aircraft Fleet Maintenance*. AAAI/IAAI 1997, pp. 1078-1085.

[8] R. Helfman, E. H. Baur, J. Dumer, T. Hanratty and H. Ingham. *Turbine Engine Diagnostics (TED): An Expert Diagnostic System for the M1 Abrams Turbine Engine*. AAAI/IAAI 1998, pp. 1032-1038.

[9] G. S. Semmel, S. R. Davis, K. W. Leucht, D. A. Rowe, K. E. Smith and Ladislau Bölöni. *NESTA: NASA Engineering Shuttle Telemetry Agent*. AAAI 2005, pp. 1491-1498.

[10] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton and G. Hullender. *Learning to Rank using Gradient Descent*. Proceedings of the 22$^{nd}$ International Conference on Machine Learning (ICML), 2005, pp. 89 - 96.