Reliability Forecasting

Sacramento Municipal Utilities District

&

University of the Pacific,
School of Engineering and Computer Science

In fulfilment of the requirements for the degree of Master of Science in Analytics

Analytics Team

Jeff Miller	J_miller42@u.pacific.edu
Kristen Guy	k_guy1@u.pacific.edu
Angela Cheng	a_cheng2@u.pacific.edu
Andrew Lee	a_lee95@u.pacific.edu

Academic Advisor: Dr. James Hetrick

Table of Contents

Introduction

The Sacramento Municipal Utilities District (SMUD) provides electric service to approximately 900 square miles in the heart of the California capital region. The University of the Pacific, founded in 1851, is California's first chartered institution of higher education, home of the School of Engineering and Computer Science and the John T. Chambers Technology Center.

With Sacramento expected to grow significantly by the year 2050, SMUD is particularly concerned with the impact that anticipated growth will have on peak demand which occurs during the summer when temperatures soar to 100 degrees or more. Today, SMUD customers drive the peak demand to near 3,000 megawatts. By 2050, peak demand is expected to be close to 5,000 megawatts. Energy efficiency is the best way to address the challenge for growing demand, including maximizing reliable energy delivery by reducing outages.

Background

The simplest definition for reliability is that electricity be available when it is needed. The North American Electric Reliability Council describes reliability as: "the degree to which the performances of the elements of [the electrical] system result in power being delivered to consumers within accepted standards and in the amount desired" (Hirst and Kirby 2000). Electrical outages are just one aspect of reliability as defined by the IEEE Standard 1366.

Historically, electric utility outages can be attributed to a number of different causes; however, the most significant generation and transmission disturbances can be attributed to weather, maintenance, and operations. More specifically, disturbances can be attributed to extreme weather, equipment failures, human error, vegetation interference, birds and other wildlife, and excess load.

The role of the SMUD Transmission and Distribution Maintenance Planning (TDMP) group is to develop maintenance and inspection plans as well as programs and procedures that ensure SMUD’s compliance with Federal and State regulations. SMUD and the TDMP group graciously accepted University of the Pacific’s request to host a reliability/outage forecasting capstone project for this team of graduate students in the Master of Science in Analytics program of the School of Engineering and Computer Science.

Project Objectives

Reliability is paramount to SMUD service and operations, and outages are the primary driver of reduced service reliability. The fundamental belief on which this project is based is that predictive analytics can provide insights that can be used to reduce SMUD's outage costs and improve service.

As such, our project's goals are to:

- Gain insights into collection and use of operational and other data relevant to outage prediction
- Use the collected data to create an analytic method for outage prediction
- Recommend future analytic methods as a basis for operational advances, cost reduction, and ultimately greater customer satisfaction.

Methodology

Data Analysis and Preparation for Modeling

Working with the TDMP team, we conducted a thorough analysis on data sets having features or events that could provide insights into the conditions related to outages in the network. The SMUD operational records we reviewed included data sets on outage events, non-outage events, peak feeder load amperage, vegetation management, and preventative and corrective maintenance at substations and distribution sites. Many of these data sets are housed in different systems or are managed by different groups within SMUD.[1]

Our first challenge was to understand the context behind each data set and the ways in which they intersected, and then determine how best to organize and connect them in a way that lends itself to predictive modeling.[2] We took an event-based approach, focusing on modeling the interplay of conditions that lead to an outage event versus those that lead to a non-outage event.

Some aspects of SMUD's data were not able to be included for this initial modeling project. For example, the vegetation management data proved to be too limited in its granularity, and the maintenance records did not have a clean way to hook together with the outage event data. We were also unable to obtain data on equipment attributes such as age or type of material.

To enhance the model in the face of those exclusions, we collected open-source data from the National Oceanic and Atmospheric Association (NOAA), pulling in features related to

[1] See Appendix: Data Discovery and Initial Findings: Operational Data

[2] See Appendix: Data Transformation

temperature, humidity, wind speed, visibility, and precipitation.[3] We also explored the idea of pulling in lightning data, but chose not to proceed due the lack of an available open-source data set at the level of granularity needed. What resulted from combining of these data sets was a large, single analytic file containing 9 years worth of time-stamped observations of outage and non-outage events associated with individual devices at specific geospatial locations on the grid.

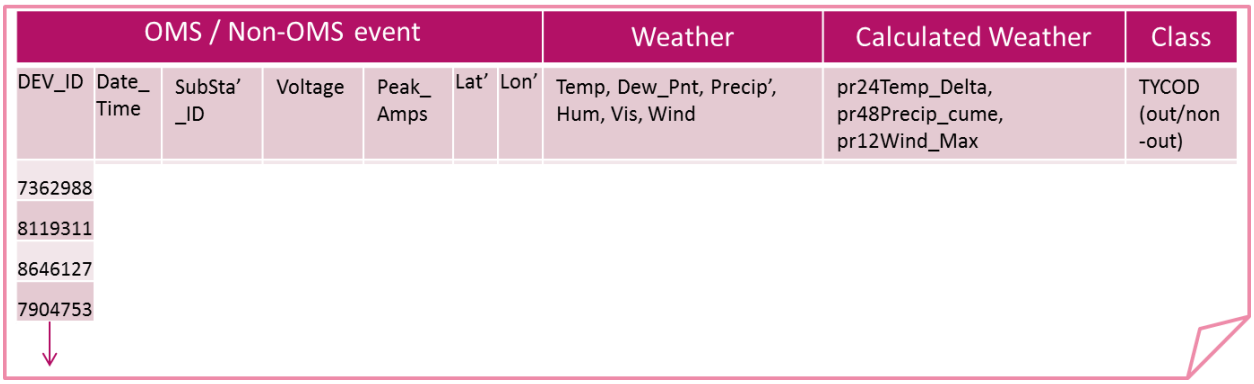


Figure 1 — Analytic File Structure

We then performed a series of statistical tests on the analytic file we had compiled, focusing on understanding how each feature contributes individually to our target (outage event versus non-outage event).[4] Features showing strong statistical significance or importance were tagged for preferential inclusion in the model. Features identified as being strongly correlated with each other were flagged to ensure both would not be included in the same model, so as not to overstate the movement they represent. A series of testable feature sets was generated, joining different features in a variety of combinations.

Predictive Modeling

To set up our predictive modeling, we chose to compare the performances of three different types of machine learning classification algorithms: logistic regression, random forest, and neural network.[5]

We conducted an iterative process of model testing[6] where we fed feature sets into the logistic regression and random forest algorithms, collected and compared the results[7] within each

[3] See Appendix: Data Discovery and Initial Findings: Weather Data

[4] See Appendix: Feature Selection

[5] See Appendix: Algorithms Selection

[6] See Appendix: Model Training and Testing Process

[7] See Appendix: Model Assessment Strategies

model type, honed the feature sets further and fed them back through each algorithm to add those results to the comparison, all while simultaneously adjusting our estimated parameters.[8]

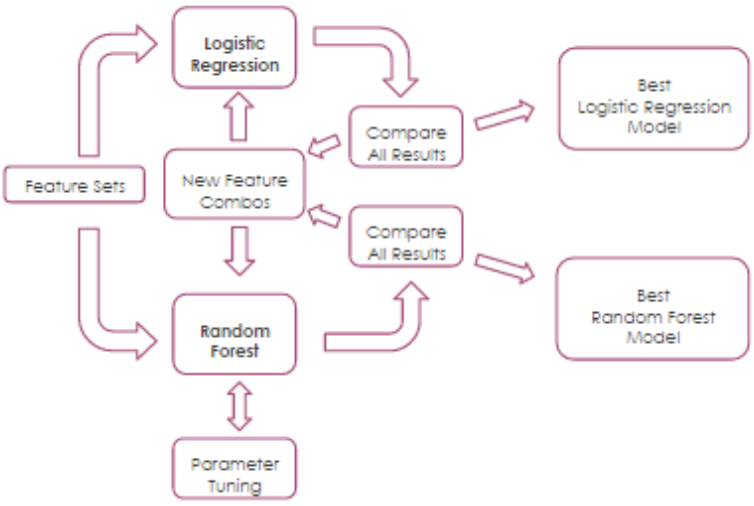


Figure 2 — Iterative Process of Model Testing

Through this process, we identified a hidden bias in the data; in the original outage and non-outage event data sets, the "System" feature (which identifies the system as being overhead, underground, etc.) was fairly consistently recorded in the outage events, but missing from almost all of the non-outage events. This caused the algorithms to treat it as a proxy for outage status, which led the initial round of models to have an unrealistically high predictive accuracy.[9]

Upon recognizing the bias, we removed that feature from our models and ran through the iterative testing process again. After identifying our top performing feature sets using the first two algorithms, we then set about training the more complex neural network model to get a similar series of model performance results. All model results were then compared within their algorithm groups, giving us the top performer of each model type to compare for our final model assessment.

[8] See Appendix: Model Assessment Strategies: Random Forest: Parameter Estimations and Appendix: Model Assessment Strategies: Neural Network: Parameter Estimations

[9] See Appendix: Discovery of "System" Bias

Findings

The table below shows the comparison of the top performing models from each of our algorithm categories.

Table 1 — Final Model Performance Comparison

Model	Accuracy	Sensitivity	Specificity	Model Complexity
Logistic Regression	57.53%	68.00%	45.36%	Simple
Random forest	59.02%	54.78%	61.39%	Moderate
Neural network	57.45%	52.78%	62.17%	Complex

All models demonstrated similar performance in prediction accuracy (ranging 57-59%), indicating that the level of accuracy is reflective of the predictive quality of the event-based approach and available data. Sensitivity and specificity metrics[10] were similar as well, with the notable exception of the logistic regression; while it outperformed the other models in detecting potential outages (68% sensitivity), it was much less specific in its predictions (45% specificity), and therefore resulting in a much higher number of false alarms.

Overall, the random forest algorithm provided the best combination of accuracy, balanced sensitivity and specificity, and model complexity when trained on a combination of location, weather, and a small percentage of substation identifiers.

Conclusions

There are a multitude analytic options for approaching reliability forecasting. This project served as an initial foray into applying predictive analytics at SMUD, and we focused heavily on front-end analysis and data preparation. Given the data available, we chose an event-based approach that incorporated both spatial and temporal elements. Our analysis uncovered interesting biases inherent in the data collection methods between outage and non-outage events, differences in granularity in data tracking, and a limited number of shared attributes between data sets that can be used in predictive modeling. We confirmed that certain key features such as location (latitude and longitude) and weather (current relative humidity and yesterday's temperature) proved statistically significant across all models.

[10] See Appendix: Model Assessment Strategies

Our "event-centric" approach to the modeling provided limited predictive capabilities using the data currently available. However, refinement of the data and aggregation methods have the potential to produce greater performance.

Recommendations

Based on our findings, we have several key recommendations for adjustments SMUD can implement to improve the landscape in which to develop stronger predictive tools.

Enhanced Data Tracking

- Increase the location and time granularity of the vegetation management data.
- Standardize identifiers used for system equipment and devices.
- Improve consistency of shared attributes between data sets, such as standardizing feature names and definitions.
    - e.g. "Feeder" in one data set is actually a concatenation of "Substation" and "Feeder" in another data set
- Develop further resolution of "unknown" outage causes.
    - Currently ranked the top 3rd most frequent outage type recorded
- Develop further resolution of the "Broken/Malfunction" outage causes.

Business Implication Assessments

- Conduct an analysis of the tangible and intangible business costs of unpredicted and incorrectly predicted outages.
    - The influence of cost can be incorporated into prediction thresholds to better support business needs.
- Define the business impact of the difference between failures and outages. For example:
    - The impact of the load being shifted onto other portions of the grid.
    - Increased risk associated devices that lack backup/alternative power options during service interruptions

Future Work

There are several opportunities for continued development of the modeling work conducted in this project[11], as well as alternate analytic approaches[12] that can be explored in future projects. See Appendix for detailed recommendations.

✶

[11] See Appendix: Continued Development

[12] See Appendix: Future Analytic Opportunities

We would like to extend our gratitude to the TMDP team for their excellent support and invaluable domain expertise, and the SMUD executive team for the opportunity to perform this analytic project.

References and Resources

1. Hirst, E., and Kirby, B. 2000. Bulk-Power Basics: Reliability and Commerce. http://www.consultkirby.com/files/RAPBPBasics.pdf.
2. Lin, L., Dagnino, A., Doran, D., & Gokhale, S. S. 2014. Data Analytics for Power Utility Storm Planning. http://corescholar.libraries.wright.edu/knoesis/1041.
3. National Oceanic and Atmospheric Administration (NOAA). http://www.noaa.gov/.
4. SMUD. https://www.smud.org.
5. Wood, B. 2014. Predicting and Preventing Power Outages Across the National Electric Grid. https://www.mitre.org/publications/project-stories/predicting-and-preventing-power-outages-across-the-national-electric
6. Oak Ridge National Laboratory. "Predicting electric power outages before they happen." ScienceDaily. ScienceDaily, 26 September 2014. <www.sciencedaily.com/releases/2014/09/140926213548.htm>.
7. Kundo, A. 2012. How We Prevent and Predict Power Outages. http://www.tibco.com/blog/2012/06/20/how-we-prevent-and-predict-power-outages/
8. Nemati, H., Sant'Anna, A., & Norwaczyk, S. 2015. Reliability Evaluation of Underground Power Cables with Probabilistic Models
9. Alice, M. 2015. Fitting a neural network in R; neuralnet package. https://www.r-bloggers.com/fitting-a-neural-network-in-r-neuralnet-package/
10. DnI Institute. 2015. Predictive Model Performance Statistics. http://dni-institute.in/blogs/predictive-model-performance-statistics/
11. DnI Institute. 2015. Random Forest Using R: Step by Step Tutorial. http://dni-institute.in/blogs/random-forest-using-r-step-by-step-tutorial/

Modeling tool: R *version 3.2.4*

Primary Modeling Packages:

'Boruta'	https://cran.r-project.org/web/packages/Boruta/Boruta.pdf.
'Stats'	http://astrostatistics.psu.edu/su07/R/html/stats/html/00Index.html.
'randomForest'	https://cran.r-project.org/web/packages/randomForest/randomForest.pdf
'nnet'	https://cran.r-project.org/web/packages/nnet/nnet.pdf
'neuralnet'	https://cran.r-project.org/web/packages/neuralnet/neuralnet.pdf
'caret'	https://cran.r-project.org/web/packages/caret/index.html
'Hmisc'	https://cran.r-project.org/web/packages/Hmisc/index.html
'corrplot'	https://cran.r-project.org/web/packages/corrplot/index.html

Appendices

Data Discovery and Initial Findings

Through research and dialog with the engineering staff at SMUD, we confirmed that the primary expected contributing factors to distribution outages are related to operational components and weather. We undertook initial data exploration to understand the characteristics of the available data in these categories in order to choose an analytic method and devise appropriate data transformations.

Operational Data

SMUD data, along with additional reference documents for interpretation of the data files, were provided in five sets: outage management system (OMS), non-outage operations (Non-OMS), system peak load, vegetation management, and maintenance.

The OMS data consisted of approximately 24,000 records of outage events from January 1st, 2007 to September 23, 2016. Non-outage operations data held approximately 260,000 records and encompassed a date range of April 14th, 2004 to October 17th, 2016.

As we reviewed these and the other data sets, we began to devise a common structure into which they could be integrated. This structure relied on the presence of a device identifier and data-time information. Of the 260,000 Non-OMS records, approximately 72,000 had device identifiers, and 64,000 were in the data range of the OMS data.

Peak load data provided the maximum amperages experienced on each of 550 unique distribution feeders under non-outage conditions, and included both substation and feeder identifiers.

SMUD's current vegetation management mapping includes 128 service grid sectors of the North and South urban routes which vary in area covered. The data set includes the last data on which work was undertaken in each area. The location and time granularity of the data captured within the vegetation management data was not sufficient to effectively integrate with the other data sets for use in model creation.

The maintenance data contained 138,847 records, each having date-time and work description, of which a majority had equipment number and street address. This data set had insufficient common features with OMS/Non-OMS to use in model creation.

Weather Data

Weather contributes to equipment wear and failure in energy distribution systems, and including weather data in outage prediction can identify equipment which is more likely to fail. Weather data analysis can be leveraged with geospatial and infrastructure data to forecast the likelihood of weather related outages.

The NOAA collects weather data from five stations in SMUD's service area. The data is reported on an hourly basis and is composed of over ninety features, including weekly and monthly summary data. We collected data for a date range of study matching that of the OMS data

(nine years) and consisted of 412,327 observations. The data had a significant number of missing elements; as not all stations reported in a given hour, and when reporting, not all features were recorded.

Data Transformation

Because different sources of data and knowledge domains were involved in this project, terms used for a given concept or element varied. Column, Feature, Field, and Variable referred to common element of a data set. Row, Record, and Observation referred to the unique elements of each data set.

OMS and Non-OMS Data

Performance of machine learning algorithms is frequently affected by the quantity of data used to train the models. At over 60,000 viable records, the quantity of the OMS and Non-OMS data was excellent. Fifteen of the features in these data sets were common, those of principal importance being device ID and data-time fields.

Some features, while maintained as separate fields in SMUD operational processes, had information content which was redundant once combined in preparation for use by a machine learning algorithm. In these cases, such as ADD_DATE_TIME and CREATE_DATE_TIME, one of the two was removed.

Frequently data features appeared to hold operator notes in free form text. In order to be of use to a machine learning algorithm, an input or "independent" variable, needs to be either a continuous value such as temperature, or of a discrete set of classes. These free form text features, while of operational value to SMUD, were not included as input for the algorithms. The remaining features were either continuous (e.g. peak load amperage), or categorical (e.g. Associated substation identifier for a given device).

Features that were not common to the OMS and Non-OMS data sets and for which a common merge key was not available were excluded. Some categorical fields had a large number of classes. In these cases, an informed decision was made to exclude or retain those fields.

We chose to use the TYCOD feature as our dependent (target) variable. The dependent variable is the variable from which an algorithm learns to distinguish between outage and non-outage classes. In this case, TYCOD was populated with thirty seven classes of event type which we translated into seven major classifications of 9,D,F,T,U,V and "O" for outage events. This simplification was made because our objective was to distinguish between outage and non-outage, and not between subclasses of outage.

Station ID is a discrete independent variable with 205 classes. This and other categorical features were converted to multiple Boolean class features, which has the effect of creating a number of discrete variables having a value of 1 or 0 for each class. This reformation was needed in order for the data to be ingestible by the machine learning algorithms.

The OMS data included location fields X_Cord, and Y_Cord which did not exist in the Non-OMS data. Rather than omit this potentially important field, we used the OMS data to create a

listing of device ID by X_Cord, and Y_Cord and merged that list back into the Non-OMS data using device ID as the merge key. This produced a fully populated Non-OMS set with location data associated with each event. If an event had a device ID for which there was no corresponding location data, it was excluded.

Events which did not have a potential role in any causal relationships precipitating outage, such as the cumulative number of customers affected by outage year-to-date or for the previous year-to-date, were excluded. Some features whose presence were a clear proxy for outage, such as CAUSE_DESC or OUTAGE_DESC, were excluded. Features that were not populated were excluded, and features that were so sparsely populated that they would impart no information to the learning model were also excluded.

Outages vs Non-Outages by Device ID

In addition to evaluating the rates to which each feature was populated or absent data, we also looked for any trend in outage related to device ID. The plot below is of frequency of outage vs non-outage for each device ID. We found no discernable trend on overall outage vs non-outage events by device ID; however, outliers did exist. Those outliers included one device associated with twenty four outage events, but only three non-outage events, and another device having twenty seven non-outage events and four outage events over nine years of data.

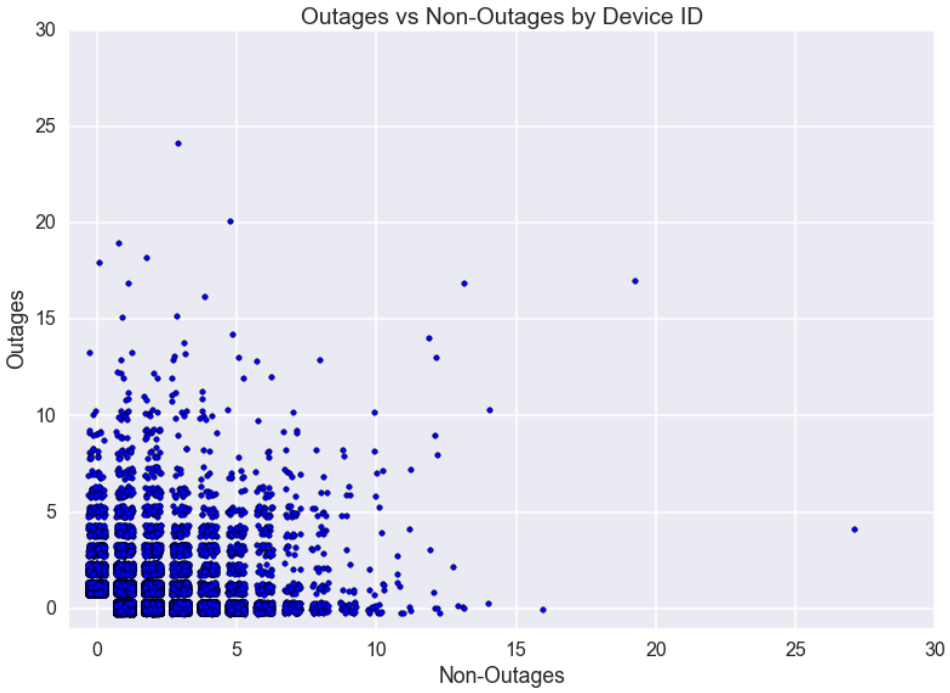


Figure a1 — Outage vs. Non-Outages by Device ID

Prevalence of Outage Cause

Also characterized was prevalence of outage cause. Over the nine year data set of outages, primary insulation failure (UG) - presumably underground insulation failures - was the dominant cause. Broken/Malfunction and Unknown followed as the second and third most prevalent causes of outage.

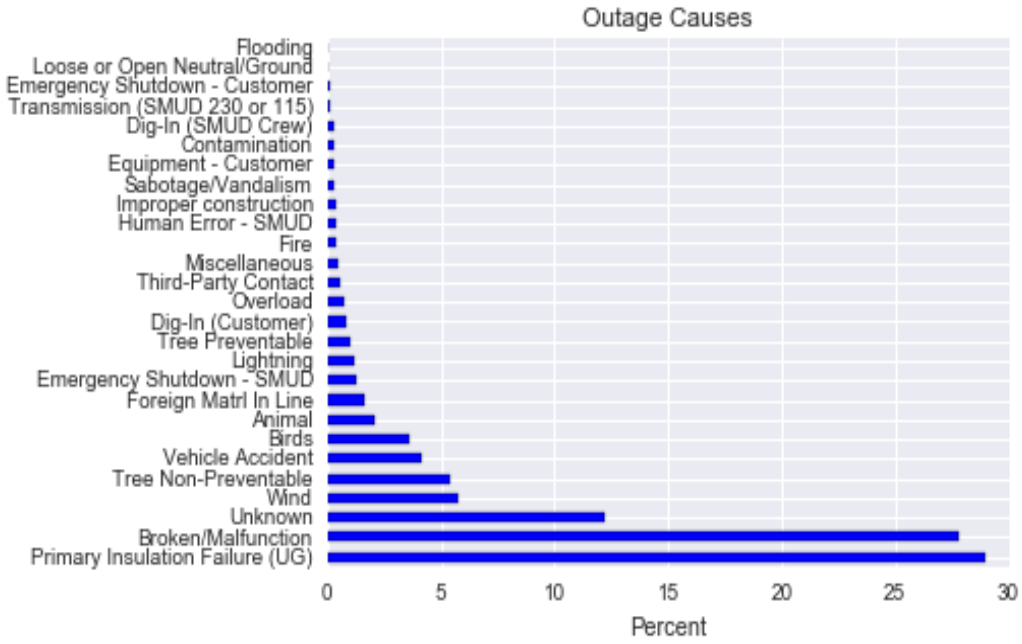


Figure a2 — Outage Causes by Percent

Location Conversion

While the operational data included location in the form of X and Y coordinates, the weather data were associated with the station location expressed in latitude and longitude. A conversion method was devised to provide latitude and longitude for each device ID appearing in the OMS and Non-OMS data.

Using the SMUD service area map, a plot of all outage device's X and Y coordinates, and a satellite imaging overlay of the SMUD service area, the true latitude and longitude for three failing devices were obtained. These devices were sufficiently distinct from other devices so as to conclusively identify their location, and were selected generally from the extremes of SMUD's service area. A conversion function, based on these three sets of X and Y and latitude and longitude coordinates, was created and applied to all device IDs in the OMS and Non-OMS data, thereby creating for each, their location expressed as latitude and longitude.

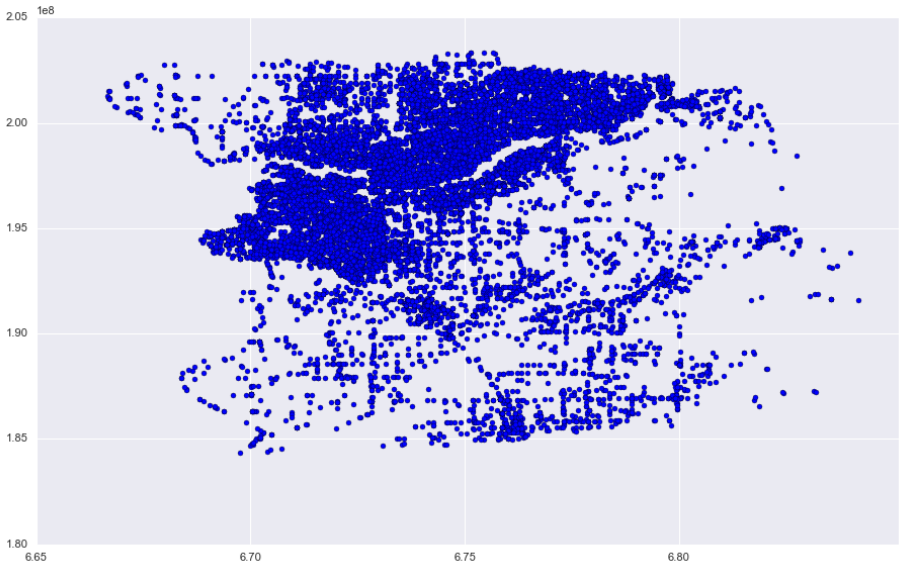


Figure a3 — Outage Device by X Y Location

Peak Load Data

The peak load data included substation and feeder identifiers. However, the data for these terms were incomplete or appear to have been used differently than in the OMS and Non-OMS data sets which necessitated creation of standardized versions of these two features to use as a merge key. The standardized alphanumeric identifiers used were those from the OMS and Non-OMS data sets. For each, the textual name in the peak load data was translated using the substation name and full feeder information in the OMS and Non-OMS data sets. The result was a peak load data set including alphanumeric substation and full feeder identifiers which were then used to merge the peak load data with the OMS and Non-OMS data sets.

Weather Data

Preparing the weather data for use in modeling required an effort of similar complexity to that for the restructuring of operational data. From SMUD's engineering team and our additional research we learned which weather features were believed to have significant impact on outages. We then used those in our initial selection of weather features. Those hourly weather features included the following:

- Visibility
- Temperature C
- Dew Point Temperature C
- Relative Humidity
- Wind Speed
- Precipitation

In addition to capturing hourly data features, we created a set of calculated features designed to capture preceding conditions with the potential for correlation with outages through some causal relationship. These calculated features are:

- Temperature change – prior 24 hours
- Wind speed maximum – prior 12 hours
- Cumulative precipitation – prior 48 hours

Analytic File Creation

The term *Analytic File* refers to a set of data that is cleaned and structured for use by a machine learning algorithm. In the case of our project, our analytic file includes operational event-based data and extrapolated weather data in which categorical variables were converted to Boolean class features and continuous variables normalized using min and max scaling.

Combining OMS and Non-OMS data was done by a random sampling the Non-OMS data set, up to the record count of the smaller OMS data set, which insured a combined balanced data set. Balance, in this case, refers to the approximately equal number of outage and non-outage events in the resulting combined data set. A balanced data set is favored when possible because machine learning algorithm performance degrades as class imbalance increases.

The event data included our initial selection of independent variables, and TYCOD (the dependent variable, consisting of a class label of outage or non-outage). However, the inclusion of weather data is not a simple merge. Each operational event included device ID, date-time, and location.

The initial step for merging weather for a particular event was to capture a list of weather observations that occurred at the approximately time of the operational event and then rank them by ascending time difference. This was predicated on availability of a weather feature (for example temperature) from a minimum of three of the five stations, as there were a non-insignificant number of null value in the weather data.

Once data for that weather feature from three observations closest to the event date-time was identified, a linear extrapolation was used to estimate that weather feature value at the device location associated with that event. As each of the weather feature datum was collected at the approximate time of the event, so too were data for the preceding 12, 24 or 48 hours. These were then used to create the calculated weather features. This was repeated for each weather feature and each operations event over nine years. In this way, the balanced operational data set was combined with weather data to produce analytic file having approximately 45000 records with 255 features.

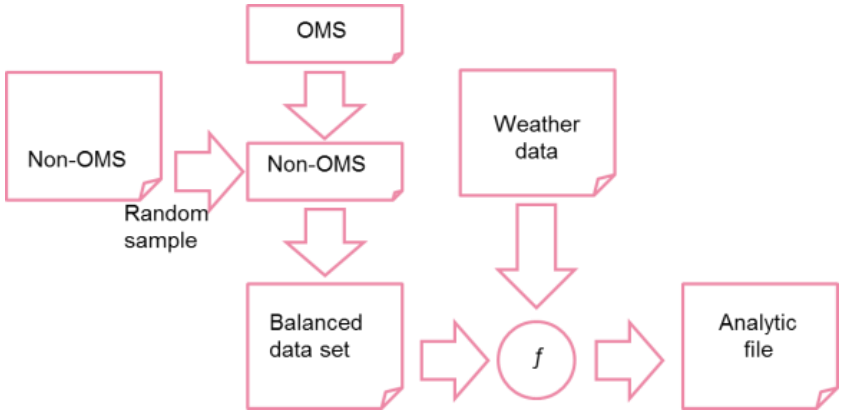


Figure a4 — Analytic File Creation Process

Feature Selection

Statistical Testing of Discrete/Categorical Input Features

For the 236 discrete input features, we performed a Chi-squared test of categorical independence. Of those, 78 proved to be statistically significant at a 95% confidence level when looked at individually in isolation with the target variable.

Table a1 — Discrete Features in Order of Highest Significance

Feature	P-Value	Feature	P-Value
System: Overhead	0.00e+00	Voltage: SEC	1.25e-03
System: Underground	0.00e+00	25 Substation IDs	1.56e-03 - 1.30e-02
Voltage: 12kv	9.91e-145	System: Underground Network	1.30e-02
System: Substation	7.53e-60	25 Substation IDs	1.56e-03 - 1.30e-02
26 Substation IDs	2.01e-18 - 1.10e-03		

Statistical Testing of Continuous Input Features

We conducted a Wilcoxon rank sum test for each of the 13 continuous feature to determine any significance in the median of the difference between samples of the input and target features. The results indicated that all continuous features demonstrate statistical significance at a 95% confidence level when looked at in isolation with the target variable.

Table a2 — Continuous Features in order of highest significance

Feature	P-Value	Feature	P-Value
Hourly Relative Humidity	9.92e-73	48hr Prior Cumulative Precipitation	4.24e-16
Hourly Precipitation	7.85e-63	Hourly Wind Speed	2.73e-13
12hr Prior Wind Speed Max	7.19e-47	Hourly Dew Point Temperature	1.25e-09
Hourly Dry Bulb Temperature	1.14e-37	24hr Prior Wind Speed Variance	2.40e-08
Peak Amps	4.32e-26	Latitude	3.50e-08
Longitude	7.56e-26	Hourly Visibility	8.91e-08
24hr Prior Temperature Delta	7.50e-22		

Correlation tests

We conducted correlation testing with the Hmisc package in R, using a threshold of 0.5 to flag strongly positively or negatively correlated input feature pairs. Our goal with this step was to avoid including multiple features that describe similar movement within the same model (thus overstating their contribution to the target, and negatively influencing out-of-sample performance). The correlation plot below demonstrations the combinations most strongly correlated via the density of the color.

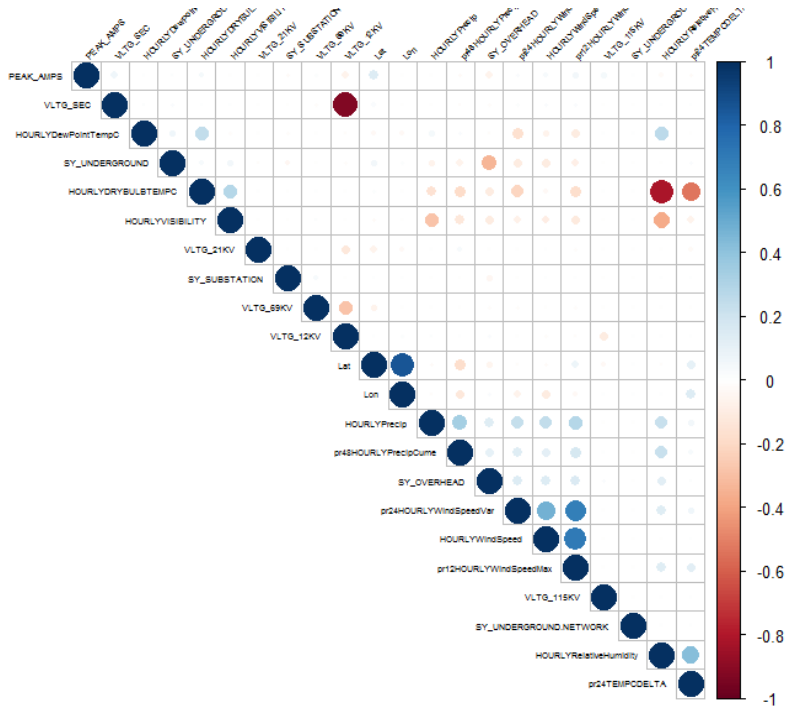


Figure a5 — Correlation Plot of Outages

Our testing returned a short list of feature combinations to avoid:

Table a3 — Feature Combinations and Coefficients

Feature 1	Feature 2	Correlation Coefficient
Voltage of 12kv	Voltage SEC	-0.924
Dry Bulb Temp	Humidity	-0.824
Wind speed	12hr prior wind speed max	0.625

When considering two highly correlated features for a given model, the one with the lower independent statistical significance with the target variable was excluded from use.

Significance testing

The ordering of features by significance is a common task in machine learning development and there are a number of methods that can be employed. Some of the more often used methods are available in development libraries. We used one such library package, called 'Boruta', for classification models which applies a random forest algorithm to perform a top-down search for relevant features and progressively eliminated irrelevant features to stabilize the selection weighting.

Feature Sets

Using the results of the testing methods outlined above, we generated a series of feature sets containing different combinations of input features. Records with missing values were dropped after the feature combination was set, resulting in different record counts depending on the feature set. After the initial rounds of model testing, additional feature sets were generated to optimize the combinations based on their performance with the different algorithms.

Feature set 0

- Size: 26149 x 250
- Description: "Baseline" feature set; includes all features from the analytic file

Feature set 1

- Size: Size: 26149 x 129
- Description: Excludes binary factors with less than 100 observations

Feature set 2

- Size: 36155 x 18
- Description: Excludes binary factors with less than 500 observations

Feature set 3

- Size: 36155 x 23
- Description: Excludes all substation ID binary factors

Feature set 4

- Size: 26149 x 20
- Description: Reduced to only the features marked significant during the random forest variable importance test.

Feature sets 4a-4k

- Description: Modifications of Feature Set 4, created after initial logistic regression modeling; each sub-feature-set was progressively reduced by 1 feature (removing the feature demonstrating the least significance in contribution towards the target).

Feature set 5

- Size: 35100 x 20
- Description: Reduced to only the features marked significant during the 2nd random forest variable importance test, where "System" features were excluded from consideration).

Feature set 6

- Size: 26149 x 14
- Description: Reduced to continuous features only, and only features that were flagged significant during wilcoxon testing.

Feature set 7

- Size: 26149 x 86
- Description: Comprised only of features flagged significant during Chi-squared and wilcoxon tests, and removing the smaller contributor of any correlated pairs.

Feature set 8

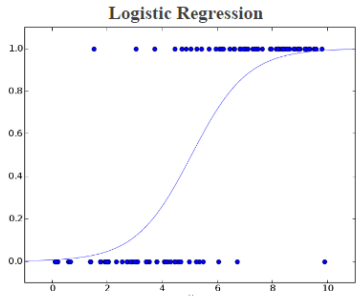
- Size: 26149 x 82
- Description: Identical to feature set 7, but excluding all "System" features.

Feature set 9 Size: 36155 x 12

- Description: Features consistently identified by early logistic regression models as being significant.

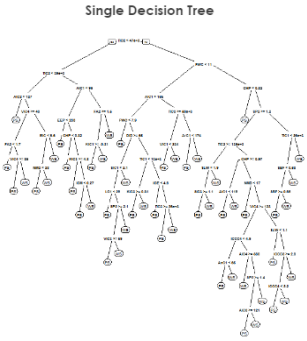
Algorithms Selection

We chose three classification algorithms to work with, each with a different mathematical approach and level of complexity.



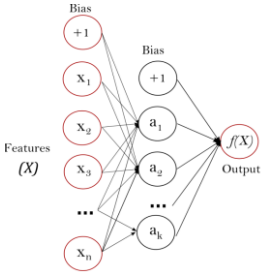
Logistic Regression is the simplest form of a binary classifier, which aims to estimate an S-shaped curve that represents the probability of shifting from one outcome (outage) to another (non-outage).

(Image from MathWorks.com; Shuang Wang, December 2012)



A **Random Forest** creates a series of semi-unique decision trees to use against each observation in the data set, and chooses the most common outcome seen across all of the trees in the "forest" to determine the likely classification for a given observation. Random Forest is considered to be a moderately complex algorithm.

(Image by Joseph Rickert, June 2013)



A **Neural Network** is an interconnected map structured somewhat like neurons in the brain. The "neurons" are laid out in a specific architecture containing an input feature layer, hidden calculation layer(s), and an output layer. Each layer is connected by "synapses" between each of the neurons, and each synapse has a specific weight associated with it. Neural networks are useful when working with nonlinear data, and are considered one of the more complex machine learning algorithms.

(Image by Issam Laradji, August 2014)

Model Assessment Strategies

Our primary model assessment metrics included the confusion matrix (calculating accuracy, sensitivity, & specificity) and the ROC curve (receiver operating characteristic). We utilized in-sample, validation, and out-of-sample performance to compare each of the models and identify cases of overfitting.

We chose to give preference to

a) higher accuracy scores and

b) higher and more balanced specificity and sensitivity scores.

The latter decision sought to attain a balance between models that would be sensitive enough to catch likely outages, yet specific enough to reduce the likelihood of generating false positive predictions.

Model Training and Testing Process

We tested the predictive efficacy of the various feature sets using each of the three algorithms. The sections below describe in greater detail some of the specific training and assessment strategies, as well as the approach for parameter estimations used for the algorithms that involve structural components.

Logistic Regression

Logistic regression, based on linear regression, is the simplest form of a binary classifier. It produces probabilities between 0 and 1, (0-100%), which are then subject to a threshold. Events with probabilities above the threshold are labeled as one of two binary classes, and those with probabilities below are classified as the second. The threshold was set to 0.5 for our testing.

As is common practice, we selected logistic regression as a baseline model with which to compare the performance of other models. *Our implementation of logistic regression was done using the Generalized Linear Models command (glm) of the stats package in the R open source programming language.*

The initial learning trial using logistic regression was done on feature set 0, the null data set, followed by feature set 4, the features set produced by the use of a random forest algorithm to perform an automated search for relevant features. Feature sets 4a through 4k were created by progressively removing the feature of set 4 having the least significance. Those retained, feature set 4K, are shown with their relative significance (lower p-value indicates greater significance) in the Variable Importance plot below.

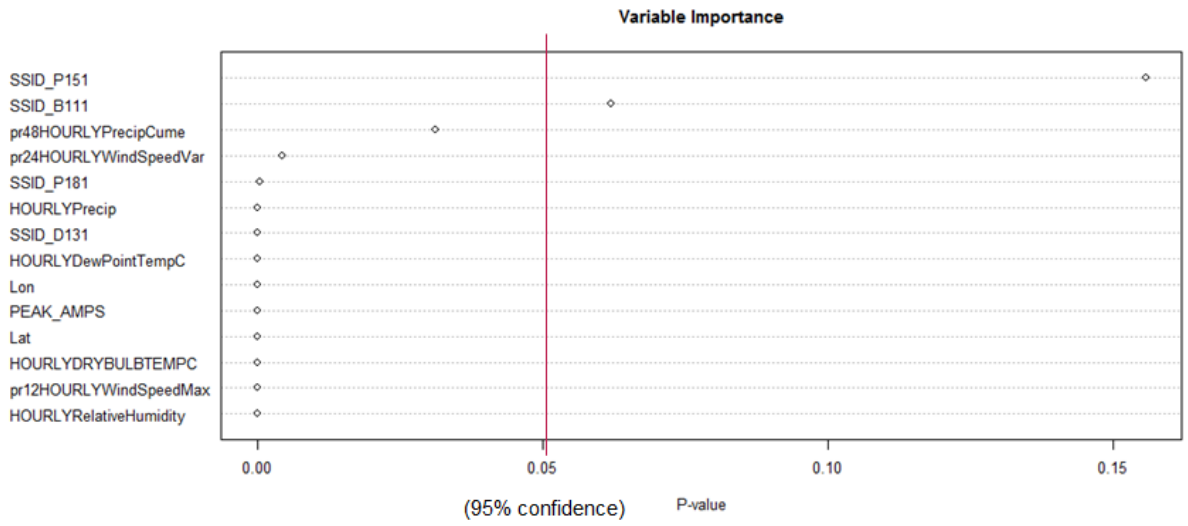


Figure a6 — Revised Logistic Regression Results

Final Model and Results - Logistic Regression

Feature set 4k was chosen for the final logistic regression model due to the highest test accuracy rate in comparison with other logistic regression models. The test model accuracy rate indicates that 57.53% of the cases were correctly classified into class 0 and class 1. The similar accuracy rates for the training, validation, and test models provide further evidence that the model will produce very similar results under varying conditions. The sensitivity rate of 68.00% indicates that of all the class 1's, 68% were correctly classified as class 1. The specificity rate of 45.36% indicates that of all the class 0's, 45.36% were correctly classified as class 0.

Table a3 — Feature Set 4k after Removal of Feature Set 4 Features by Descending P-value

	Accuracy	**Sensitivity**	**Specificity**
Training	56.01%	70.82%	39.01%
Validation	57.03%	66.55%	45.44%
Test	57.53%	68.00%	45.36%

Random Forest

Model training was conducted using the randomForest package in R.

There were three rounds of model training and testing. In the first round, each of the original feature sets was used to generate a separate model. When the initial results proved to be unrealistically good and the source of the bias was identified[13], a second round of training was conducted focusing just on feature sets that excluded any of the biased features. That round of results identified some of the most likely predictive feature sets, and we began to experiment with parameter estimations to test whether performance could be enhanced.

Parameter Estimations

There are two structural parameters being estimated in a random forest: the number of trees, and the number of variables to be sampled at each split (m_{try}). We started with a baseline of 50 trees, and used the tuneRF() function to identify the m_{try} with the lowest in-sample out-of-bag (OOB) error.

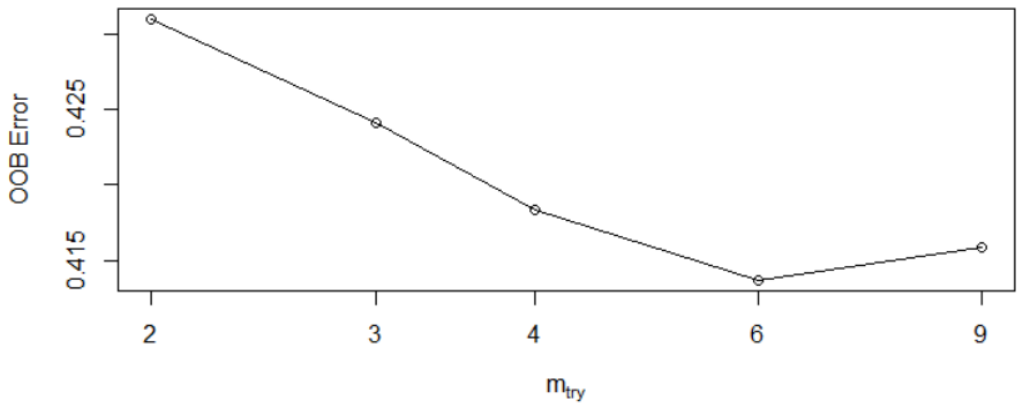


Figure a7 — Tuning the Number of Variables per Split

We then used the plot(model) function to visually estimate the behavior of the in-sample OOB error alongside the true positive and false negative errors. Using a maximum of 500 trees, we identified the point at which errors were no longer decreasing across all three metrics as the appropriate tree count for that feature set. Variable splits were then recalculated using tuneRF() for the optimized tree count. The plot below demonstrates the behavior of a model with an optimized tree count and m_{try}.

[13] See Appendix: Discovery of "System" Bias

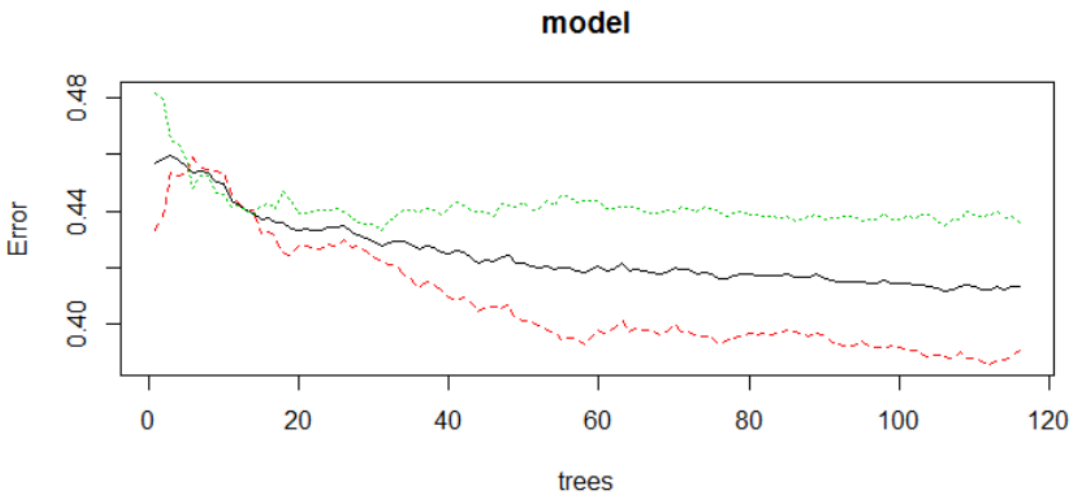


Figure a8 — Error Rate Changes by Number of Trees

Final Model and Results - Random Forest

After completing all of the testing, the final feature set that proved to have not only the highest accuracy but also the highest and most balanced sensitivity and specificity scores was Feature set 5 (excluding "Dry Bulb Temperature"), using 116 trees and an m_{try} of 6.

Table a4 — Revised Random Forest Results

	Accuracy	**Sensitivity**	**Specificity**
Training	58.18%	55.15%	61.18%
Validation	58.09%	54.68%	61.39%
Test	59.02%	54.78%	63.30%

The test model accuracy rate indicates that 59.02% of the cases were correctly classified into class 0 and class 1. The similar accuracy rates for the training, validation, and test models provide further evidence that the model will produce very similar results under varying conditions. The sensitivity rate indicates that of all the class 1's, 54.78% were correctly classified as class 1. The specificity rate indicates that of all the class 0's, 63.30% were correctly classified as class 0.

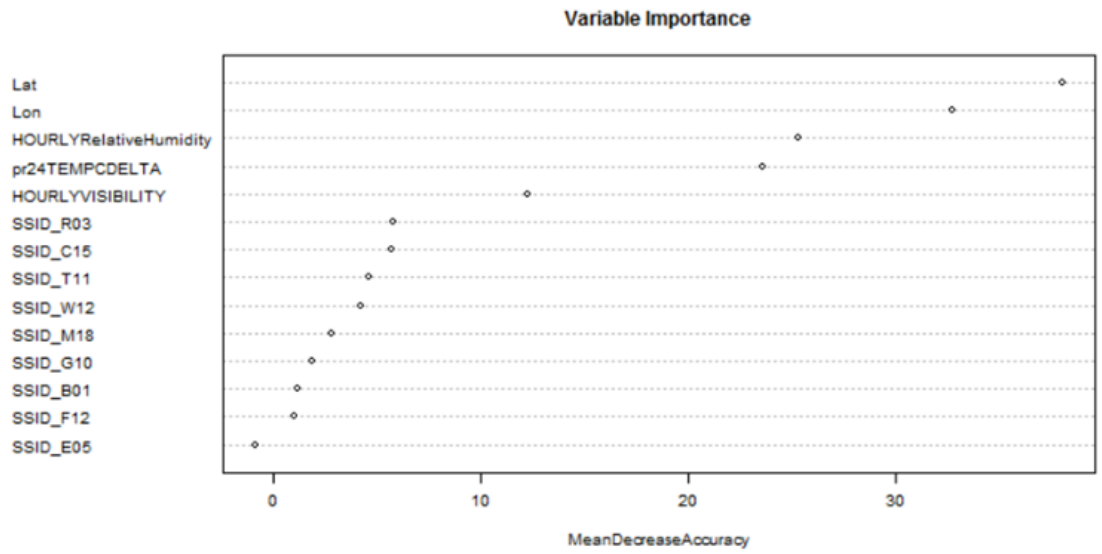


Figure a9 — Variable Importance for Revised Random Forest Results

Neural Network

Model training was conducted using two separate packages in R (neuralnet and nnet) in order to test out different training strategies and compare their performance.

The training for the neural network was postponed until high performing feature sets had been identified by the other algorithms.

With the neuralnet package, we conducted an iterative feature testing approach which involved adding in a single feature at a time, and only retaining it in the model if it improved the performance of the overall performance of the model. This required the build of a separate model during each step of the training process. The approach proved computationally prohibitive with the full data set, leading us conduct this training on a small slice (5%) of the observations instead. The models produced via this strategy exhibited extremely poor results (accuracies consistently less than 50%).

A switch to the nnet package produced models much more in line with what we had seen in the previous algorithms, and proved to be more robust and able to run full feature sets without encountering convergence or computational problems. Measurements of accuracy ranged between 50% and 57% for each data set.

Parameter Estimations

Neural networks are highly sensitive to the number of input features and number of neurons in the hidden layer, due to the large influence those factors have on the number of weights that need to be calculated. During our feature combination testing (which causes variation in the

number of input features), we adopted two separate methods for identifying the optimal number of neurons in the hidden layer.

With the **neuralnet** package, we worked with an iterative process of adding in one feature at a time on each run, and testing each feature combination with a variety of hidden layer sizes, starting with one neuron and building up from there, ceasing neuron testing once the validation accuracy, specificity and sensitivity either stabilized or degraded. The manual nature of the testing process limited the number of feature and neuron size combinations able to tested.

With the **nnet** package, we were able to streamline parameter estimation by running a subgroup of the highest-performing full feature sets through a loop that created a series of models ranging in hidden layer size from 1 to slightly above half the number of inputs of the feature set. The loop produced a table of results with which to determine the best performing feature set and hidden layer size combinations.

In assessing the neural network performance, model complexity was taken into account. For example, feature sets 5 and 8 performed similarly well at 8 hidden layer neurons (with the latter slightly outperforming the former). However, the number of input nodes required for feature 8 jumped the number of weights to be calculated from 145 to 673, and we determined that the minor increase in accuracy did not justify the exponential increase in parameters to be calculated.

Additional testing was performed on the best model to identify whether setting a maximum number of steps or a stopping criteria on validation error increases would improve the performance. The model was set to iterate through several step maximums of 1 through 500, with the criteria of stopping if the validation error increased three times in a row. The validation error proved to hold steady early on, with no indication that the larger step size was overfitting to the training data. This was also supported by the in-sample results, which proved comparable to both validation and out-of-sample findings.

Final Model and Results - Neural Network

Feature set 5 was chosen as the final neural network model due to the most accurate test accuracy rate relative to other neural network models. The test model accuracy rate indicates that 57.45% of the cases were correctly classified into class 0 and class 1. Much like the logistic regression and random forest models, the neural network model's similar accuracy rates for training, validation, and test models indicate that the model will produce very similar results under varying conditions. The sensitivity rate indicates that of all the class 1's, 52.78% were

correctly classified as class 1. The specificity rate indicates that of all the class 0's, 62.17% were correctly classified as class 0.

Table a5 — Feature Set 5, 8 hidden layer neurons, 145 weights

	Accuracy	Sensitivity	Specificity
Training	58.09%	53.58%	62.42%
Validation	52.12%	50.58%	53.64%
Test	57.45%	52.78%	62.17%

Summary - Model Training and Testing Process

The similarity in accuracy rates for all 3 models indicates uniformity in predictive accuracy across modeling methods and that the models will produce similar results if given new input data. The logistic regression model has the highest sensitivity rate in the test model of 68.00%, which indicates that it is the most successful model in classifying the target event outcome of outages. Both the random forest and neural network models have a stronger tendency in predicting non-outages with both models having over a 60% specificity rate as opposed to a sensitivity rate in the low 50% range. Ultimately, it was decided that the random forest model was the best model due to having the highest overall accuracy rate of 59.02%, a medium level of model complexity, and more balanced sensitivity and specificity rates.

Discovery of "System" Bias

The results of the first round of modeling with the random forest and logistic regression algorithms were unrealistically good, with each measure of model assessment coming out at close to 100%.

Table a6 — Initial Model Results

	Logistic Regression *Feature Set 4*			**Random Forest** *Feature Set 4* *50 trees, m_{try} of 6*		
	Accuracy	**Sensitivity**	**Specificity**	**Accuracy**	**Sensitivity**	**Specificity**
Training	97.93%	96.13%	99.99%	99.97%	99.95%	100.00%
Validation	97.57%	95.47%	100.00%	97.99%	96.29%	99.96%
Test	98.03%	96.33%	100.00%	98.06%	96.39%	100.00%

An investigation into the variable importance plot for the random forest models identified the majority of factored “System” features to be exponentially more significant than other features in the model.

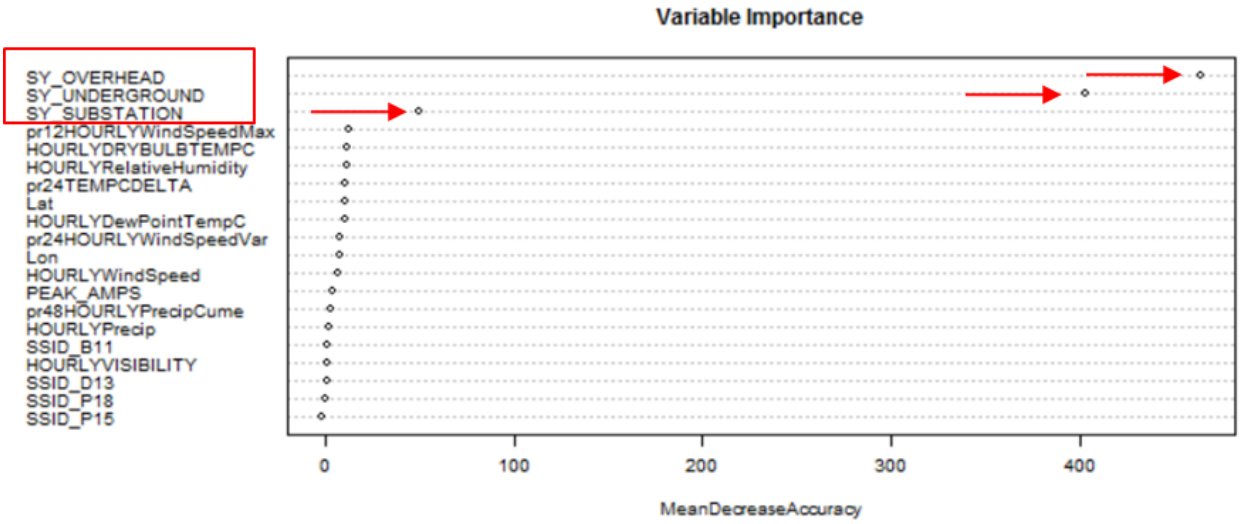


Figure a10 — Bias Identified: “SY_” Skew

Upon revisiting the raw data files, we uncovered an extreme distortion in the distribution of empty values between the outage and non-outage event records. While most outage events were populated with a system category, nearly the entirety of the non-outage events were left blank for this feature, with the exception of a single observation. During the binary factoring process for this feature for the analytic file format, all empty values were treated as category 0. This caused any system variable set to 1 to act as a proxy for outage events, hence the highly unrealistic accuracy of our models.

Once this skewing effect was identified, we entered round two of the training process, and removed the system features from any model under consideration in order to produce more balanced and realistic results.

Continued Development

The following are thoughts we would like to share with those who may choose to build upon what we have accomplished:

Higher resolution weather data

There are limitations to the use of linear extrapolation of weather data. Continued development of this project’s event-based analysis using the NOAA’s global forecast system can be expected to more finely resolve the impact of weather on the machine models and improve performance. Integrate preventive and corrective maintenance data

A comparison of the device IDs feature of the combined OMS, Non-OMS data, and the Equipment field in the Preventative/Corrective maintenance data found no common features on which to connect the data. However, assuming the terms are synonymous and a translation is available, the maintenance data can be integrated with operations data.

Include cost functions

The costs of false positive outage predictions and missed outage predictions (false negatives) are complex and distinctly different in their elements. However, static cost weighting can be added to the prediction models to enhance current performance and serve as a basis for including more sophisticated costing in the future.

Lightning data

Lighting data could be integrated into the combined analytic data; however, the sparsity of lighting in SMUD's service area would suggest this would be a lower priority than some of the other recommendations listed here.

Prediction tool

The time available for this project and the model performance based on the data available did not permit the full completion of an operational tool for outage predicting. However, after implementation of the continued development suggestions outlined above, it is expected that a completed tool can produce the desired reporting of outage probability under prescribed weather scenarios which could then be used in the prioritization of outage mitigation.

Future Analytics Opportunities

Future potential for the use of analytics by SMUD within the context of systems reliability are expansive. We have attempted to capture, and would like to share, the following ideas encountered during the development of our project which could be of value in future analytic work.

Predict failures and failure types rather than outages

The distinction between failure and outages is not inconsequential when architecting a machine learning strategy. Failures may occur, which do not result in outages due to system resilience or redundancy. And so the prediction of failures which precipitate outages, rather than the outages themselves, would enhance predictive performance.

System element rather than Device ID

Using device identifiers as the key data element by which to aggregate data for algorithm training excludes some operational events that are associated with failures.

Using system elements rather than device ID alone could include other equipment as well as conductor segments. Adopting the idea of system element on which to aggregate data will allow additional data sets and data types to be included. Using this scope of inclusion, any system element with which a failure can be associated would enhance the learning of the algorithm as well as performance of the prediction models.

Including circuit, feeder or line conductor elements in the data raises a question as to how to best associate their linear construct with other spatial data features such as population data, vehicle traffic density, or types of foliage.

Vegetation management data collection

Vegetation management activities are believed to reduce the likelihood of outages sufficiently to justify the cost of revising data collection. The collection of higher granularity data of both location and time can be expected to provide improved model performance. Such revised vegetation data collection could include Date-Time work began and ended, circuit, feeder or line identifier adjacent to tree trimming activity, location (latitude / longitude), and the type of vegetation (tree/shrub).

System element characteristics

Different classes of system elements (meaning those with which a failure could be associated) have characteristics which may correlate with failure. Including these characteristics in data aggregation will allow modeling of those correlations and their use in predicting failures. For example, such characteristics for a transformer could include electrical and thermal design capacities, above or underground installation, in-service date, and expected lifetime.

Characteristics of circuit or feeder conductors could include energy capacity, cross sectional area, tension rating, material type (Steel, Aluminum, and hybrid), insulation material, and in-service date.

Wear metrics

Including the characteristics of system elements in the data sets would enable the creation and use of wear metrics. A wear metric for a transformer might include cumulative Amp-Hour loading as a percent of temperature de-rated thermal capacity. This cumulative metric could be based on the load-time data reported from smart meters connected to that particular transformer.

Other analytic methods

Our project has been based on machine learning methods with select weather features calculated at fixed intervals of 12, 24, and 48 hours prior to an outage event. This allowed inclusion of specific temporal features which domain experience indicated are correlated with outage.

Time series analyses are a highly developed set of analytic methods which are different from machine learning. While distinct, they do offer more effective means for assessing real-time data, such as loading, weather, or power quality conditions presaging outages.

Multiple models

Our initial modeling has been based on a single classification of outage and non-outage. While we chose this classification for simplicity in this initial development, multiple models can be explored.

A model can be created for each of any set of outage types represented in the operational data that are deemed significant by measure of cost or service impact. Different models can be expected to produce better performance when individually optimized for advantage over a combined model.