

**Southern California Edison (SCE)**  
**Model Documentation**  
**Prepared for 2026-2028 WMP**  
**Appendix B**

Combined Capacitor Sub-Model

5/16/25

## Table of Contents

1.	Executive Summary .....	3
1.1	Model Purpose and Intended Use .....	3
1.2	Model Description Summary .....	3
1.3	Model Risk Rating .....	4
1.4	Model Dependency and Interconnectivity .....	4
1.5	Model Assumptions .....	5
1.6	Model Limitations .....	5
1.7	Overall Model Performance Assessment.....	5
1.8	Contingency Plan for Vendor Model.....	5
2.	Model Framework and Theory .....	6
2.1	Model Inputs and Data Quality.....	6
2.2	Methodology.....	9
2.3	Suitability .....	12
2.4	Assumptions.....	12
2.5	Limitations and Compensating Controls .....	13
2.6	Model Outputs .....	14
3.	Model Performance and Testing.....	17
3.1	Model Specification Testing.....	17
3.2	Sensitivity Analysis .....	19
3.3	Outcome Analysis / Backtesting .....	21
3.4	Benchmarking Analysis .....	22
4.	Model Management and Governance .....	24
4.1	Ongoing Monitoring Plan .....	24
4.2	Security and Control .....	24
5.	References .....	24

## 1. EXECUTIVE SUMMARY

### 1.1 Model Purpose and Intended Use

The Combined Capacitor Model is a Probability of Ignition (POI) Sub-Model developed by Southern California Edison (SCE). At SCE, models are developed using Machine Learning (ML) algorithms for each asset—in this case, the combination of Overhead (OH) and Underground (UG) capacitor. The Combined Capacitor model is refreshed annually and used to predict the probability of failure (POF) for distribution OH and UG capacitors.

The calibrated outputs of the Capacitor model—i.e., failure events—are used by three programs described below:

1. Inspections and Remediations programs that consider POI as an element in prioritization and scoping.
2. Asset Class Strategies that are developed using the capacitor model to prioritize high risk capacitors for replacement strategies.
3. Risk analyses via SCE’s Multi Attribute Risk Scoring (MARS) Framework.

### 1.2 Model Description Summary

The Combined Capacitor model is a binary classification model using Extreme Gradient Boosting (XGBoost)—a ML technique. It predicts the probability of a capacitor igniting a spark due to equipment failure by considering available capacitor attributes and condition data (i.e., age) and other environmental and operational attributes (e.g., historical weather, number of switches).

The model is programmed in Python using libraries like scikit-learn and pandas and is connected to databases such as SAP, ADS Weather, etc. The model is run once a year manually by the Advanced Predictive Modeling team. The model is calibrated every year with the last 5 years of historical outage data.

Please refer to Section 2.1 for more information about the inputs used by the capacitor model along with data processing details.

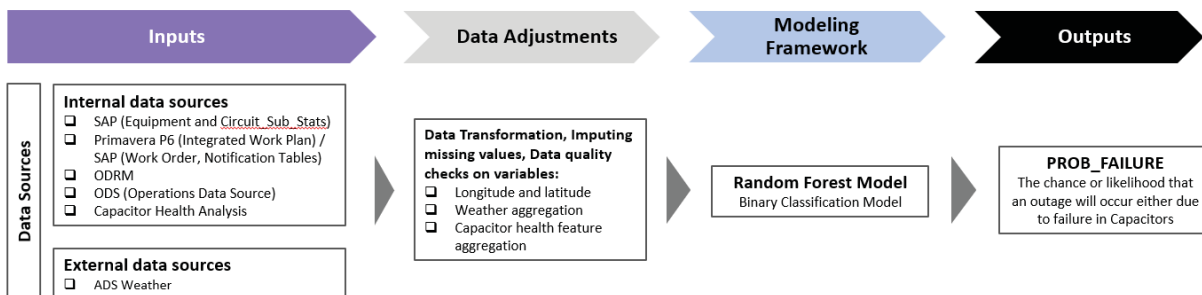


Figure 1: Combined Capacitor model framework

The combined capacitor model uses the XGBoost methodology. Since the prediction is a classified event (i.e., failure) and the XGBoost methodology can perform both classification and regression tasks, the XGBoost methodology is considered a good choice for the capacitor model. This methodology predicts output with high accuracy, runs efficiently on large datasets, and maintains accuracy with minimal adjustments for missing values and data treatments.

### 1.3 Model Risk Rating

There is no defined mechanism to identify model risk rating at SCE. However, certain factors—like frequency of risk events and use case—are considered when flagging model risk. Based on the Wildfire Mitigation Plan quarterly report, the frequency of outages in a year from capacitors averages around 338. This frequency is moderately low compared to other sub-drivers. Figure 2 provides a snapshot of the count of outages over the years by OH Capacitor equipment failures. In addition, the output of this model importantly informs the strategy of a few programs, discussed in section 1.1. Hence, the Combined Capacitor model is deemed to be a medium risk model.

					Number of risk events																Projected risk events							

Figure 2: Key recent and projected risk events due to capacitor damage or failure from SCE Q1 2022 Quarterly Data Report, Table 7.1

References: Refer to link [RF 1: SCE’s WMP 2022 Q1 Quarterly Data Report submission

] in Section 5 for SCE’s Wildfire Mitigation Plan Q1 2022 Quarterly Data Report submission.

### 1.4 Model Dependency and Interconnectivity

The Combined Capacitor model is an “Ignition Likelihood” model that uses Atmospheric Data Solutions (ADS) modeling output along with other data sources to calculate the probability of ignition.

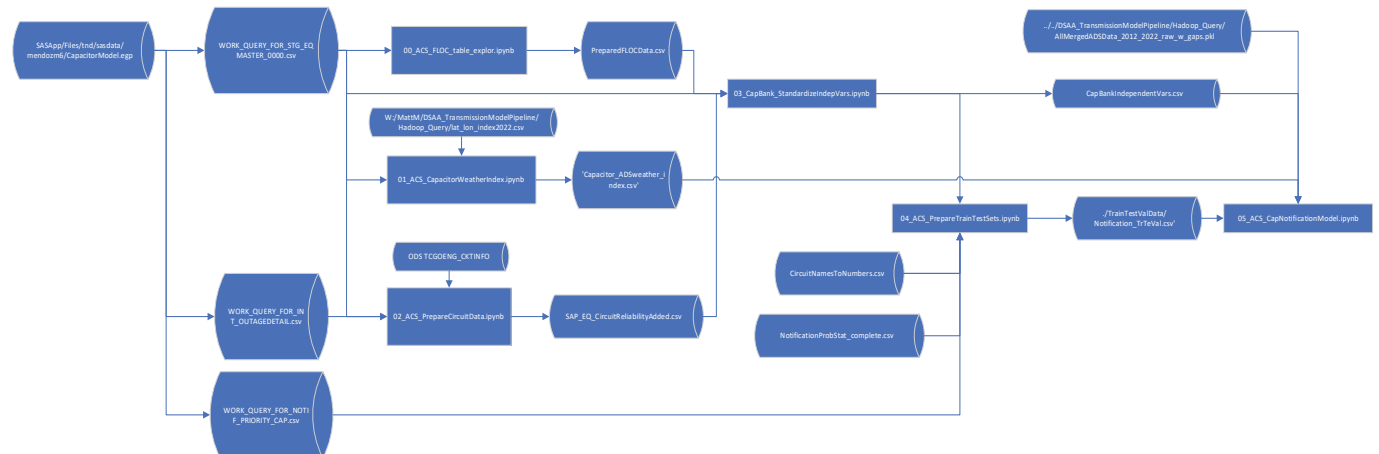


Figure 3: Model Interconnectivity Schema

ADS weather variables are used as one input in the Combined Capacitor model. ADS’ Next Generation Weather Modeling System (NGWMS) upgrades SCE’s in-house weather modeling capabilities and enhances SCE’s ability to make more targeted PSPS decisions. The ADS model generates 10 years of hourly weather data between 2012 and 2022. That information is then processed and aggregated to calculate statistical measures such as mean and standard deviation of wind, humidity, rain, snow, etc. These are used as locational measures and are matched to the capacitors by their latitude and longitude coordinates.

The output data from the Combined Capacitor model (i.e., POI) is used to inform the strategic decisions of the three categories of programs, discussed in Section 1.1.

## 1.5 Model Assumptions

The business and model assumptions for the Combined Capacitor model are summarized below:

1. There is no change in the Capacitor technical specification over time.
2. The calibration methodology assumes that fires are a subset of failures.
3. The model is designed to work in both base weather and extreme weather conditions.
4. The feature variables in the dataset should have some actual values so that the classifier model can predict accurate results.
5. The predictions from each tree must have very low correlations.

A detailed explanation of these assumptions is available in Section 2.4.

## 1.6 Model Limitations

The model limitations for the Combined Capacitor model are summarized below:

1. Unavailability of linear/non-linear representation in the form of intuitive equation or correlation statistic.
2. Resource utilization in terms of system capacity and higher configuration for model execution is high.
3. Model accuracy may reduce if the dataset experiences covariate shift.

A detailed explanation of these data limitations is available in Section 2.5.

## 1.7 Overall Model Performance Assessment

The ML model used to build the Combined Capacitor model is the XGBoost algorithm. The model's overall performance is determined by the Area Under the ROC Curve (AUC) value and Confusion Matrix results.

The performance of the Combined Capacitor model was evaluated on test data using five years of notification data through 2023.

- The AUC value is 0.84.
- Confusion matrix results capture the accuracy rate as 80.2%.

The above metrics were derived at the time of the model refresh in August 2024 to capture an exhaustive set of statistical results for documentation purposes.

## 1.8 Contingency Plan for Vendor Model

A contingency plan is not applicable for this model as it is an in-house SCE model. This is not a vendor model.

## 2. MODEL FRAMEWORK AND THEORY

The Combined Capacitor model is a binary classification model pertaining to OH and UG capacitor equipment failures. It employs an XGBoost algorithm to predict the likelihood of a capacitor experiencing failure that can result in ignition event. The XGBoost approach was chosen for the classification task over other modeling approaches—such as support vector machines and random forest—because it predicts output with high accuracy, runs efficiently on large datasets, and maintains accuracy with minimal adjustments for missing values and data treatments.

### 2.1 Model Inputs and Data Quality

#### Data Sources

This model refers to multiple internal and external data sources. The internal data sources used by the model are:

- **SAP** houses circuit<sup>1</sup>, structure, and equipment characteristics. It contains latitude and longitude information of the assets. It also contains data gathered from inspections in the form of notifications.
- **ODRM** refers to Outage Database and Reliability Metrics. It contains the detail for all historical outages. This information is used in conjunction with data from ODS (Operational Data Store) containing information about devices like active underground and overhead switches which is used to identify locations impacted by an outage. It is also used to record historical reliability indices which are used at a circuit level to inform the models of historic stressors from outages or transients.
- **CHAD** refers to the capacitor health analysis dashboard and is a record of capacitor automations and MegaVolt Ampere Reactive (MVAR) readings.

The external data sources used by the model are:

- **ADS** model provides 10 years of hourly gridded weather data from 2012-2022. These are aggregated to individual locational measures and matched to the capacitors through spatial join to the nearest grid by the latitude and longitude as a part of the data engineering step.

#### Quality Checks

SCE has internal data management teams for ensuring data quality, including Enterprise Asset Data (EAD) and Master Data. They work on processing asset data corrections (E2 notifications) in SAP and fixing largely known data issues like missing or erroneous latitude and longitude information for assets in their territory. Some of the data quality checks performed in the Combined Capacitor model to ensure the accuracy, validity, integrity, and consistency are provided below. Quality checks (QC) are incorporated and coded in Python.

The QC steps performed by Python code are as follows:

---

<sup>1</sup> Circuit comprises segments that collectively form a path for electrical current floating from the power source (including, but not limited to, a substation) to another power source or circuit endpoint.

- The FLOC data file that is used to fetch the latitude and longitude values of the FLOCs from SAP has data quality issues. Coordinates are converted to floating point numbers and entries that are in degrees are converted to decimals. Missing or zero valued coordinates are imputed ordinarily by sorting FLOCs.
- ODRM provides information about all the outages encountered by SCE. Only the relevant information like failures specific to capacitors is loaded into this model. All the other non-relevant information, i.e., for equipment other than capacitors, is removed before loading the data into the model.

The manual QC steps are as follows:

- ADS weather data is validated against actual weather observations.
- Asset data obtained from SAP is validated and updated through inspections and other programs.

### **Data Sampling**

Since this is a classification model to predict the outages, there are no sampling strategies used in the model other than the random split strategy to bifurcate the train and test data. The dataset used for the model is randomly divided to have 67% train data and remaining 33% for test data.

### **Data Cleansing and Transformation**

The data cleansing and transformation activities that are incorporated in the Python code as a part of automation to ensure the completeness of data used for model training and estimation are provided below.

- Missing data for the below specified numeric variables are handled by imputing the mean value across the associated circuit.
  - GEN\_ALLOC
  - GEN\_QUE
  - GEN\_REMAIN
  - TOTAL\_GEN
  - NPL
  - CPL
  - PLL
  - MAX\_ACHIEV\_CAPACITY
  - PENETRATION\_LEVEL
- Data consistency is ensured by correcting formatting issues in date variables e.g., EQ\_StartUpDate variable can have different formats of data, format is corrected in python program code for data formats to be consistent.
- Reliability indices SAIDI, SAIFI, and MAIFI are summed to the circuit, for a year, then averaged over 5 years to get an average yearly circuit reliability which is applied to appropriate capacitors based on what circuits they are attached to.

### **Data Assumptions**

The accuracy of the predicted results depends on the accuracy of the data used to build the predictive models. The following are the data assumptions:

Southern California Edison (SCE) Model Documentation Prepared for 2026-2028 WMP Appendix B  
Combined Capacitor Sub-Model

1. The assumptions for the data imputation uses SCE's Distribution Design Standard (DDS), engineering judgment, manufacturer data, and acceptable engineering practices.
2. For performing the mean-by-circuit imputation for locations, it is assumed that distribution circuits do not cover more than a few miles of territory and since the locations are used to assign weather values, missing locations provide a reasonable estimate within the resolution of the weather data. Dates are used to calculate in service age, as such it is assumed that missing dates are the same as the median startup date making capacitors with missing dates the same age as most capacitors on the circuit.
3. Input data with respect to asset information, weather information, and engineering information are assumed to be stable and will not change over time until the subsequent data refresh. Example: If there is an update in the structure information specific to an asset, that updated information will be reflected only in the subsequent data refresh. So it is assumed that the updated structure information is not drastically different from the previous information and would not alter the model outcomes.

### Data Limitations

The following are data limitations across internal and external data sources:

- Some of the data used by the model faces accuracy issues in terms of consistency in data labelling, missing data for a specific feature (predictive variable) that may impact model prediction power.
  - Data labelling issues might be caused by manual errors during data entry. For example, when the type is manually fed into the system, different labels might be used in different data entries. This affects the consistency of the data and needs to be addressed before using the data in the model.
- With respect to Failure targets, the starting location of the outage is recorded at the FLOC and associated with a circuit. This combination of FLOC and circuit is used to identify which capacitor experienced the failure.

### Independent variables

The Combined Capacitor model uses multiple variables/features. A subset of features is provided below.

Feature	Data Source	Description
InServiceYear	SAP	Equipment age, calculated by subtracting EQ_StartUpDate from year of subset.
PLL	SAP	Planned Loading Limit
EQ_NumberOfSwitches	SAP	Total number of switches on the capacitor bank
Structure_StartUpYear	SAP	Parsed from SAP EQ_StartUpDate, engineers identified that there were certain design standards that were changed in certain years.
Avg_air_temperature_2m	ADS	Average hourly air temperature



Feature	Data Source	Description
Stddev_pop_air_temperature_2m	ADS	Standard deviation of the hourly air temperature

Asset information is fetched from SAP. In addition to the data above, 10 years of hourly data fetched from ADS Weather model is processed and aggregated to calculate statistical measures like mean, max and standard deviation for wind, temperature, water vapor, turbulence kinetic energy, humidity, rain, and storm.

### Dependent Variable

In a typical classification risk model, defining the dependent variable is key for both model development and model performance assessment. The dependent variable in Combined Capacitor represents the observation of an outage occurring due to failure in capacitors. It is a binary status of failure or non-failure.

The final output of the model is PROB\_FAILURE, representing the chance or likelihood that a capacitor failure will occur. The probability value ranges from 0 to 1 where '0' represents the least likelihood for an outage and '1' represents the high chance for an outage.

## 2.2 Methodology

SCE uses ML to identify patterns that may lead to failures causing sparks from capacitors and uses the trained model to predict Probability of Ignition (POI)s at the asset level. The Combined Capacitor model employs an XGBoost algorithm to predict failure events.

XGBoost is a supervised ML algorithm that is constructed from many decision trees. It can be used to solve both classification and regression problems. This approach uses ensemble learning, which is a technique that combines many classifiers to achieve greater predictive accuracy than that of a single classifier. A decision tree is a decision support technique that forms a tree-like structure. It consists of three components: decision nodes, leaf nodes, and a root node. The following diagram shows the three types of nodes in a decision tree.

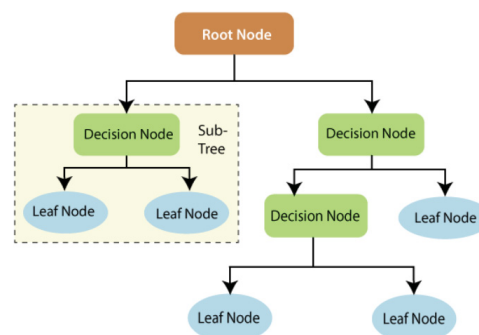


Figure 4: Decision Tree Structure

A decision tree algorithm divides observations of a dataset into branches, which further segregate into other branches. This sequence continues until a leaf node is attained. A leaf node cannot be segregated further. In more detail, the root node is the base of a decision tree, where the first of a chain of decisions is made. A branch is the connection path between nodes. A node is a potential splitting point on a tree.

Decision nodes provide a link to the leaves. On the other hand, leaves, also known as terminal nodes, are the ends of a tree, representing the resulting classification or value for the sample.

The 'forest' generated by the XGBoost algorithm is trained through boosting. Boosting is an ensemble meta-algorithm that fits multiple models sequentially, where each new model corrects the errors of the previous ones. The diagram below shows the contrast between bagging and boosting.

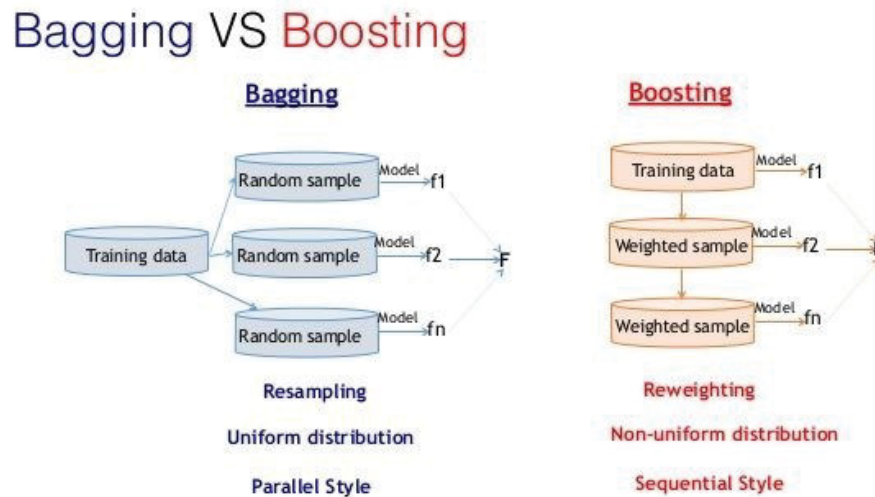


Figure 45: Structure of Boosting vs. Bagging

The selection of the final output in boosting follows a weighted voting system. In this classification model case, each decision tree contributes to the final output based on its accuracy, with more accurate trees having a greater influence. The XGBoost system combines the outputs of all the trees, correcting errors from previous trees to improve overall accuracy. The sequential addition of trees in the boosting process leads to higher accuracy and helps reduce bias, while also mitigating the risk of overfitting.

Train test split is a model validation procedure that simulates how a model would perform on new/unseen data. Figure 6 shows the logic for dividing the dataset into train data and test data. First, the data is consolidated and prepared for the train test split. Then the historical input datasets are split into a training dataset (67%) and testing dataset (33%) based on simple random sampling strategy with a split ratio of 2:1 without replacement. Simple random sampling is a technique that ensures each observation has an equal likelihood of being selected for a set. It is a fair strategy as it helps in avoiding any bias involved compared to other modeling techniques and it has no restrictions on the sample size which makes it suitable to handle vast amounts of input data. The predictive algorithm is developed using the training dataset and built by looking at the interactions between all the features to find patterns and predict the likelihood of equipment failure.

## Southern California Edison (SCE) Model Documentation Prepared for 2026-2028 WMP Appendix B Combined Capacitor Sub-Model

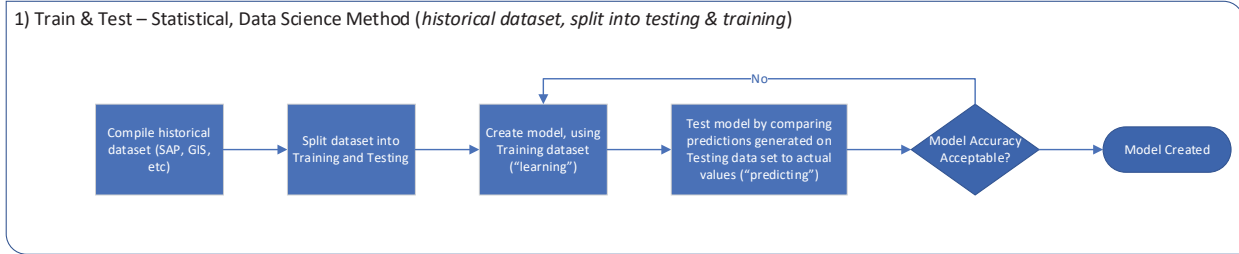


Figure 6: Train and Test data split logic

In the next step, the algorithm is tested on the ‘testing’ dataset. The model is run on the test dataset to make a prediction of failure or non-failure. Then an internal validation of the model is conducted by comparing the predicted results to the actual results which indicates the predictive capabilities of the features as well as the model. Area Under the Curve (AUC) is the metric used to assess the performance of the model on test data.

AUC – Area Under the Receiver Operating Characteristic (ROC) Curve estimates the model discriminatory power (i.e., degree of separability) for the binary classification problem. The ROC curve plots True Positive Rate against different thresholds with False Positive Rate (FPR) or True Negative Rate (TNR). The higher the AUC, the better the model is at predicting True Negatives (non-events) and True Positives (events).

### Hyperparameter Tuning:

Hyperparameters are parameters that are explicitly defined by the user to control the learning process. The process of selecting the optimal hyperparameters for model performance is known as hyperparameter tuning, and the tuning process to achieve the best-defined performance statistic is known as hyperparameter optimization. The Cartesian Grid search and Random Grid search are widely used strategies for hyperparameter optimization.

- In the Cartesian grid search approach, the ML model is evaluated for a range of hyperparameter values, and it searches for the best set of hyperparameters from a grid of hyperparameters values. The disadvantage of grid search model is that it will go through all the intermediate combinations of hyperparameters which increases the time consumed by grid search computations.
- In the random grid search approach, the ML model is evaluated for a range of hyperparameter values like that in Cartesian Grid Search approach. However, search criteria parameters are added to control the type and extent of the search, and it moves randomly within the grid to find the best set of hyperparameters to achieve maximum performance in terms of the metric defined by the user. As search criteria, the user can set a maximum runtime for the grid, a maximum number of models to create, or metric-based automatic early stopping. If many of these requirements are supplied, the algorithm will end when the first of the criteria is met. This approach reduces the time taken for computation thereby solves the drawbacks of the cartesian grid search approach.

The Combined Capacitor model uses the Halving Random Grid Search method for Hyperparameter tuning. The reference literature link to understand the efficiency between Cartesian Grid search and Random Grid search is provided below. The criterion used for the hyperparameter tuning in Combined Capacitor Model are:

- N\_estimators: Total number of constructed by the ensemble model.

- **max\_depth:** Specifies the maximum size of the sample data drawn for training each tree. A higher value for this feature will make the model more complex but can lead to overfitting.
- **Scale\_pos\_weight:** This parameter assigns weight to the positive class to make these classifications more influential to the model. This is a way to handle imbalanced data to increase the recall of the model at the expense of type I and II errors.
- **Learning\_rate:** A parameter that controls the rate at which the optimizer updates weights. It's a value between 0 and 1. The learning rate affects the model's accuracy and speed of learning:
  - **Smaller learning rate:** Results in slower but more accurate updates. This can lead to a more optimal outcome.
  - **Larger learning rate:** Results in faster but less accurate updates. This can lead to underfitting.
- **Subsample:** The subsample parameter in XGBoost controls the fraction of observations used for each tree. By adjusting subsamples, you can influence the model's performance and its ability to generalize.

Halving Random Grid Search method uses random grid search methodology along with recursive halving and k-fold cross validation to find optimal hyperparameters.

Once the grid search completes, the grid object containing the list of models is queried, and models are sorted by a performance metric defined by the user. The model with better performance is chosen as the best model and it is validated on the test data.

References: Refer to [RF RF 1: SCE's WMP 2022 Q1 Quarterly Data Report submission

] in Section 5 to understand the efficiency between Cartesian Grid search and Random Grid search.

## 2.3 Suitability

During the development of the model in 2024, both Random Forest and XGBoost methods were tested. XGBoost yielded better AUC scores than Random Forest and ran more efficiently. XGBoost was tested in previous years and was close to outperforming Random Forest. The test results showed that the XGBoost methodology fits well with the data and the results desired from the latest refresh. See Section 3.4 for the AUC comparison of these two approaches.

XGBoost methodology can solve classification and regression problems and works well with categorical and continuous variables. Among the main advantages of the XGBoost methodology is that it runs efficiently for large datasets and maintains accuracy with minimal adjustments for missing values and data treatments. Theoretically, the XGBoost methodology exhibits a higher level of accuracy and stability and handles non-linear parameters and missing values more efficiently than other approaches. XGBoost also runs faster in Google Cloud compared to other models.

Hence, the usage of XGBoost for the Capacitor model is deemed to be a suitable fit.

## 2.4 Assumptions

The key business assumptions that were considered during model development are specified below:

**BA 01: There is no change in Capacitor technical specification over time.** The model assumes the type of Capacitors used in the model building process have the same characteristics in terms of build and quality.

**BA 02: The Calibration model assumes that fires are a subset of failures.** Capacitor failures do not always result in service loss to customers. As such, outages only represent a subset of capacitor failures. All failures are captured in the inventory of capacitors, the SAP equipment tables. These removal codes detail various reasons for removal. The removal codes used for the capacitor predictive model are related to failure, deterioration, and damage. Codes pertaining to removal due to circuit redesigns or idling are not used to quantify failures. Hence capacitor removals due to failure, damage, or deterioration are considered as events in model development. These removal conditions indicate capacitor conditions that can potentially spark an ignition as the failure target which can turn into a fire, but all failures will not result in a fire. Hence, fire can be treated as a subset of failure.

**BA 03: The model is designed to work in both base weather and extreme weather conditions.** The weather variables incorporated in the model are represented as various statistical aggregations like max, mean, and standard deviation on wind, wind speed, humidity, rain, and snow. Hence the model results can be used under both base weather and extreme weather conditions.

The functional/model methodology assumptions that were considered during model development are discussed in detail below:

**MA 01: The feature variables in the dataset should have some actual values so that the classifier model can predict accurate results.** In an ideal scenario, all variables would have actual, not estimated values. The current model can provide accurate results even after using estimates as they are derived through imputation using actual values from other variables.

**MA 02: The predictions from each tree must have very low correlations.** It is difficult to differentiate between a real interaction effect, marginal effects, and just random variations in XGBoost. Hence, the presence of highly correlated variables in XGBoost approach will have an impact on its ability to identify strong predictors.

## 2.5 Limitations and Compensating Controls

The key model limitations that would impact the accuracy and performance of the model are discussed in detail below:

**Limitation ID:** L01

**Limitation Title:** Unavailability of linear/non-linear representation in the form of intuitive equation or correlation statistics.

**Description:** The XGBoost algorithm does not explain any linear or non-linear relationship in the form of an intuitive equation or correlation statistic to enable measurement of the scalability of impact of independent variables on the dependent variable.

**Compensating Controls:** The XGBoost model is considered a black box as it is difficult to understand the relationship between independent and dependent variables and how independent variables influence

predictions. Since black box is a common limitation with most ML algorithms, usage of the model is considered appropriate as it provides better AUC results than other models.

**Limitation ID:** L02

**Limitation Title:** Resource utilization for model execution is high.

**Description:** Since XGBoost models process many decision trees, they need more resources with respect to system configuration and system capacity to store that data.

**Compensating Controls:** The resource utilization factor will have a major impact on real-time models as they would run more frequently. Since the Combined Capacitor model is run only once a year with reasonable use cases, the impact of resource utilization is low. Additionally, the usage of random grid search and stopping criteria like tolerance, maximum rounds, maximum run time, and performance improvement thresholds provide more control on the number of recurring instances run to identify the best fit hyperparameters to achieve optimal AUC. Since the model is not executed through computer program automatically at a defined frequency and is instead run only once a year manually, usage of the model is considered appropriate.

**Limitation ID:** L03

**Limitation Title:** Model accuracy might reduce if the dataset experiences covariate shift.

**Description:** Covariate shift is a type of model drift that occurs when the distribution of independent variables changes between the training environment and live/test environment. Since the XGBoost cannot extrapolate (i.e., predict outside the training space), the model performance might decrease if there is covariate shift in the dataset.

**Compensating Controls:** The covariate shift affects most ML models to some degree, as test data is never going to be the same as training data. Detecting and addressing covariate shift is therefore a key step to the ML process. The current model is run only once a year along with data refresh. It uses a random sampling mechanism to split the dataset into train (67%) and test (33%) data whenever it is run. The random sampling mechanism is used to resolve covariate drift and maintain the accuracy of model results. Hence the XGBoost methodology and the random sampling mechanism to split train/test data are considered appropriate.

## 2.6 Model Outputs

The Combined Capacitor model predicts the probability of failure (POF) arising from equipment (capacitor) failure. The model has a single output characterized by a continuous number between 0 and 1 for each Capacitor asset. For the wildfire risk, only overhead (OH) capacitors are contributors to wildfire risk and are the only capacitor assets included in the following calibration.

The probabilities across different asset failure predictive models cannot be aggregated or compared and hence are calibrated to derive frequencies of ignition. The sum of the resulting frequencies of ignition for a sub-driver equals the total expected ignitions for the specified year.

$$Frequency\ of\ Ignition = Probability\ of\ Ignition \times \frac{Calibrated\ Targets}{\sum Probability\ of\ Ignition}$$



where Calibrated Targets = Forecasted Ignitions for that sub-driver

The output from this calibration exhibits the following features:

- Frequency: Each value can be specified as the frequency of fires per year.
- Comparability: The frequencies are comparable with sub-drivers and models.
- Additivity: The frequencies can be added across models to derive the aggregated fire forecast in a year.

This is achieved by forecasting fires by sub-driver and using these forecasts to further weight the model probabilities. The sum of probabilities from each calibrated model equals the forecast by sub-driver.

Figure 7 provides the calibration steps that are performed using the failure probability results from the OH components of the Capacitor model. The methodology followed in the calibration model is provided below:

- Aggregate the probability output from each sub-driver model.
- Based on the forecast logic selected, find the forecast results (i.e., expected fires) for each sub-driver.
- Generate the calibration factor for each sub-driver based on the values calculated in the above steps (B/A).
- Multiply each model probability by its calibration factor to arrive at the estimated frequency of fires from each sub-driver.

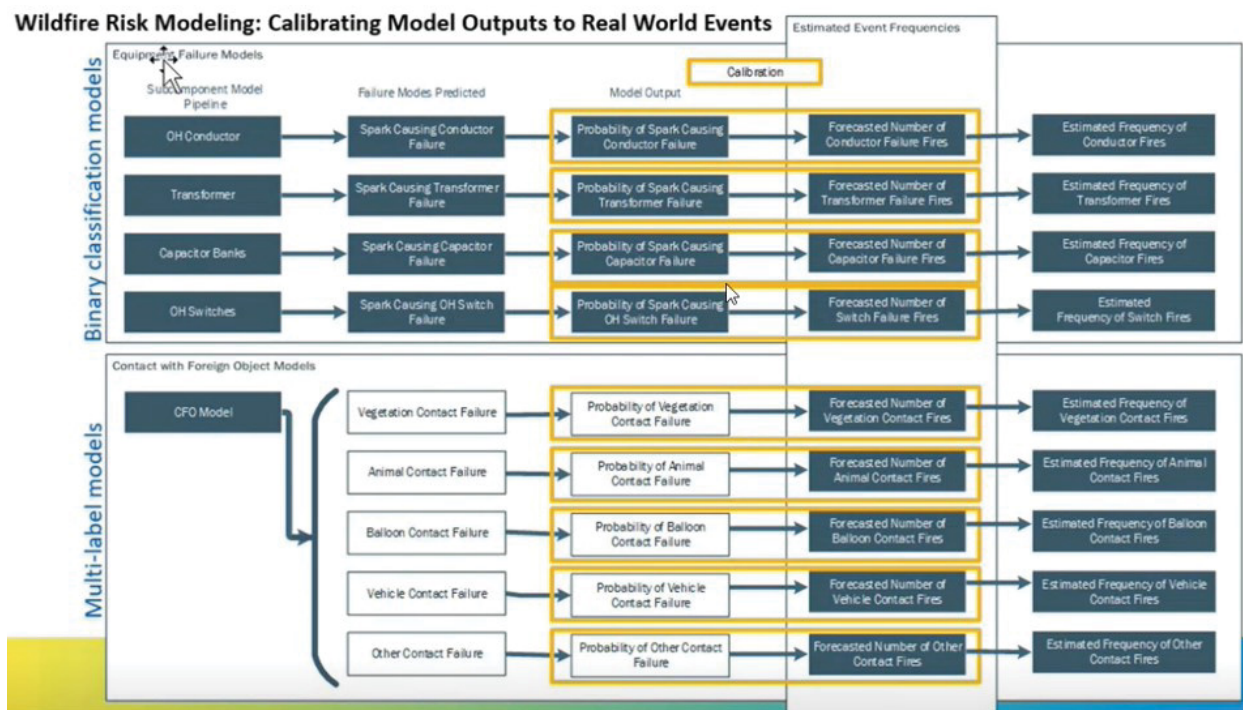


Figure 7: Calibration model schema

This estimated frequency of fires from each sub-driver can be added across the models to derive the expected frequency of ignition for each location.

The data from the calibrated probabilities—frequencies of events—based on the output from the OH Capacitor model is used to inform the programs mentioned in Section 1.1.

### Model Changes:

Till September 2021, the Combined Capacitor model only used SAP REMOVAL CODE to signify a failure. As part of the greater asset class strategy effort, two other ways of identifying failures were added: Outages from ODRM and Notifications from SAP. Removal code is a record in the capacitor inventory that records when specific capacitors are removed from the system due to deterioration, damage, or failure. Outages are records of failures that result in customer loss of service. Notifications are records from inspections teams to identify capacitors of concern. Although the three options exist, for wildfire risk assessment, notifications were ultimately used for failure indicators due to the nature of failing capacitors often degrading “silently” (without an outage signature). Engineering experts advised that inspections would be the best way to identify failing capacitors.

Model explanatory data was enriched with the addition of Capacitor Health and Analytics Dashboard data (CHAD) which was undergoing development over the last 2 years. This data contains hourly voltage and MegaVolt-Ampere Reactive (MVAR) data as reported by automated capacitors and is aggregated to describe average changes in values over time.

The AUC value of the Combined Capacitor model was 0.84 after refresh.

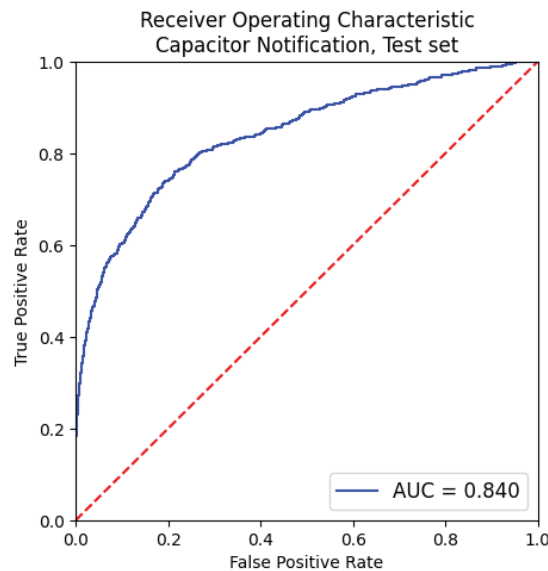


Figure 8: AUC results for Combined Capacitor Model



### 3. MODEL PERFORMANCE AND TESTING

For each ML model developed, SCE tries to select the best algorithm based on the model train/test performance, which can be measured by Area Under the Curve (AUC) and other metrics from the Confusion Matrix.

#### 3.1 Model Specification Testing

The model is developed and tested in Python using the library scikit-learn. The model is run once a year manually by Advanced Predictive Modeling team.

SCE performs verification of the model implementation by checking the variable importance results (see Model Estimation section below for a detailed explanation of variable importance results). The performance of the model is validated through the AUC, defined in Section 2.2 and provided in Section 3.3.

The validity and impact of the Model Assumptions are discussed below:

- The features used in the model are expected to have some actual values so that the model results can be accurate. In an ideal scenario, all the variables would not have estimated values, and they would instead use actual values. Missing numerical values are estimated by imputation using average values shared with like equipment on the same circuit. After using these estimates, the data quality is enhanced to support reliability of the current model in terms of improved predictive accuracy.
- XGBoost is a powerful method for variable selection in high-dimensional data. It can handle variables with high correlation due to its tree-based structure, which considers dependencies hierarchically. However, highly correlated variables can still pose challenges, as they might lead to redundancy and reduce the model's interpretability. While XGBoost can capture interaction effects, distinguishing between real interactions, marginal effects, and random variations can be complex. Therefore, it is often beneficial to filter out highly correlated features to improve the model's performance and interpretability.

#### Model Estimation:

The Combined Capacitor model employs several independent variables. Section 2.1 contains a list of the independent variables utilized in this model.

The feature importance results for the Combined Capacitor model (Figure 9) show the order of which features provide the most information gain in informing the correct prediction of failure or non-failure. The variable importance features test estimates the relative influence of each variable by calculating whether that variable was chosen to split during the tree building process and how much the squared error over all trees improved, or decreased, as a result.

## Southern California Edison (SCE) Model Documentation Prepared for 2026-2028 WMP Appendix B Combined Capacitor Sub-Model

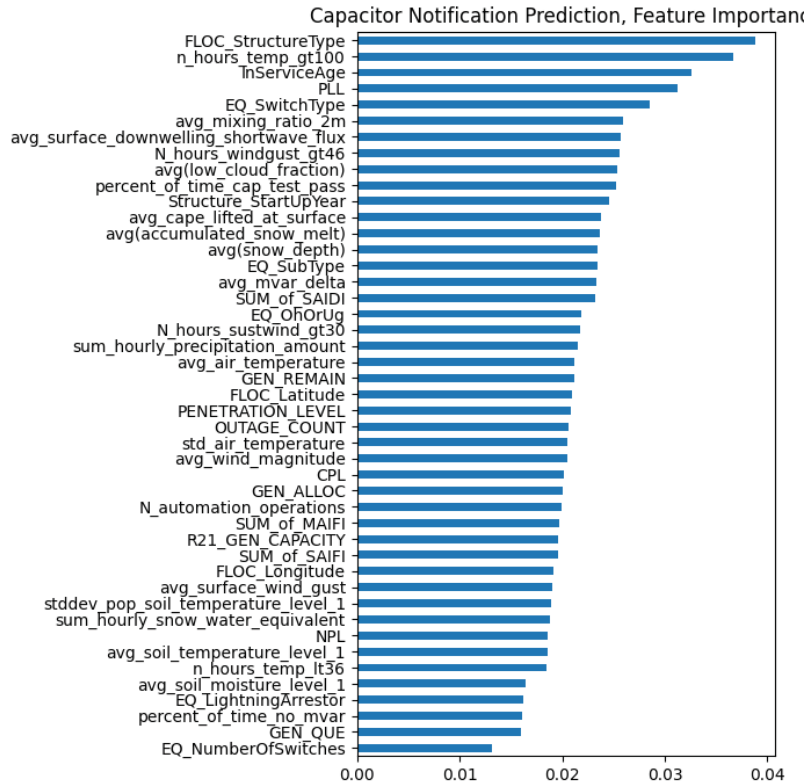


Figure 9: Variable Importance test results for Combined Capacitor model

References: Refer to link [RF 3RF 1: SCE's WMP 2022 Q1 Quarterly Data Report submission

] in Section 5 for description on the methodology used to perform the Variable Importance for tree-based methods.

The Combined Capacitor model uses the halving random grid search approach for hyperparameter optimization to select the best set of hyperparameters to achieve maximum performance in terms of AUC and recall as described in Section 2.2. Once the grid search is completed, a list of models with their associated hyperparameter values is obtained. The acquired models are then sorted based on the AUC values for the model while also considering that maximizing capturing failures and minimizing misclassifications provide more operational benefit than classifying nonfailures. To this end, precision, recall, and f1 score are also considered as tradeoffs to AUC to find the optimal values for all. The best model is run on the respective test data, and the AUC metric is used to evaluate the performance of the model.

The AUC is used to estimate the model discriminatory power to predict the results in a binary classification problem. A higher AUC means the model can predict the results more accurately. Figure 11 shows the AUC ROC for Combined Capacitor based out of test dataset ran with holdout SAP notification data through 2023. The AUC value for the optimal model is 0.84 Optimal model hyperparameters are:

Hyperparameter	Optimal value
N_estimators	1000
Max_depth	15

<b>Learning rate</b>	0.03
<b>Class_weight</b>	75

In terms of model convergence, the random grid search for hyperparameter tuning uses a stopping criterion based on a specified tolerance in AUC. This means that the additional efforts involved in hyperparameters tuning and training are not likely to improve the model performance beyond the specified threshold.

The accuracy of the model prediction, in addition to AUC, can be determined using the Confusion Matrix and Classification Error Rate results.

- A Confusion Matrix presents a tabular layout of the different outcomes of the predicted and actual values of a classifier model.

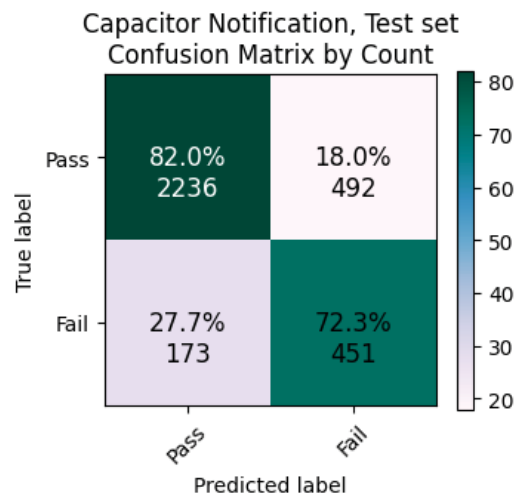


Figure 10: Confusion matrix results by count and normalized

- Table 1 provides the Confusion Matrix results for the Combined Capacitor model. It captures the accuracy rate as 80.2% on the test data.
- Classification error rate is used to estimate the proportion of instances misclassified over the whole set of instances. It is estimated using the formula below.

$$Error\ Rate = \frac{False\ Positives + False\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negatives} * 100$$

The error rate for the capacitor model is 19.8%.

All these test results are performed on test dataset with holdout SAP notification data through 2023.

A detailed assessment of the model limitations and associated compensating controls is available in Section 2.5.

### 3.2 Sensitivity Analysis

Shapley Additive Explanations (SHAP) is a method that provides an explanation of a model's output by attributing the contribution of each feature to the model's prediction. SHAP is based on the concept of Shapley values, which is a method for distributing the contribution of each player in a cooperative game. In the context of a ML model, the players are the features, and the game is to predict the output. For sensitivity analysis, SHAP values were calculated for the input features to quantify how each feature impacted model predictions.

To calculate the Shapley values for a feature, SHAP generates a set of all possible feature combinations, called coalitions. For each coalition, SHAP calculates the model's output and the difference between the output of the coalition with and without the feature. These differences are averaged over all possible coalitions, giving a measure of the feature's contribution to the model's prediction. This process is repeated for each feature in the model.

The result is a set of Shapley values that describe the contribution of each feature to the model's prediction. Positive Shapley values indicate that the feature increases the model's prediction, while negative values indicate that the feature decreases the prediction. The magnitude of the Shapley value indicates the importance of the feature. These values can be used to provide an explanation of the model's output, either by showing the contribution of each feature for a specific prediction or by calculating the average contribution of each feature over the entire dataset. By transforming variables into additive factors that drive probability, SHAP can analyze the sensitivity of a model to different variables, which can help identify which features are the most important in making predictions. Overall, SHAP provides a powerful method for understanding and interpreting the behavior of complex ML models.

## Southern California Edison (SCE) Model Documentation Prepared for 2026-2028 WMP Appendix B Combined Capacitor Sub-Model

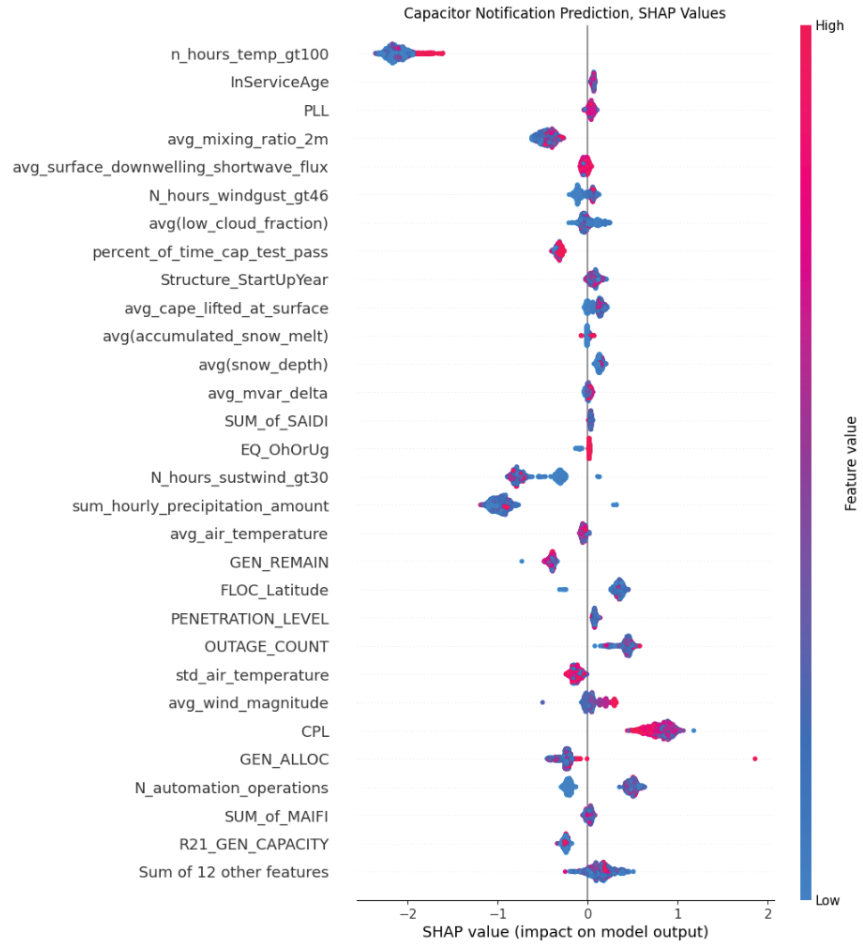
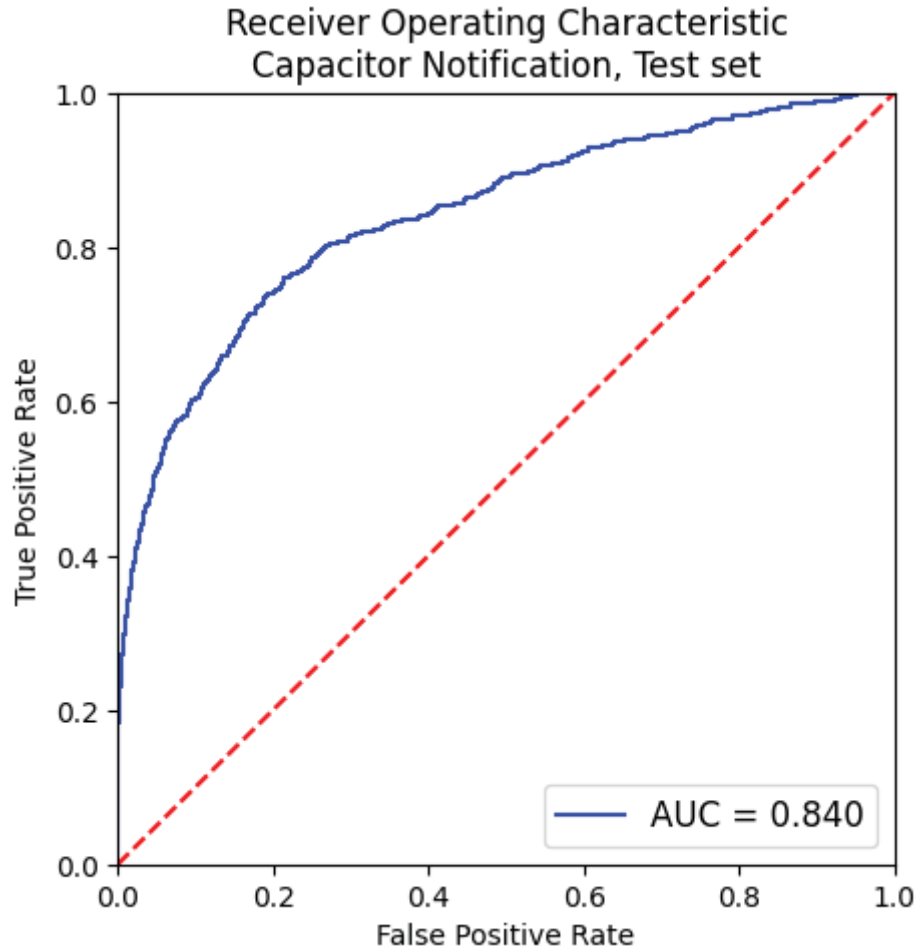


Figure 11: SHAP for most important features

### 3.3 Outcome Analysis / Back testing

The subset of historical data on which a model is trained and optimized is referred to as the in-sample data, while the subset of the dataset that has been reserved to test the model is known as the out-of-sample data. The Combined Capacitor model uses a random sampling approach to split the dataset into Train (67%) and Test (33%) data. The results of the train data are considered in-sample back testing and the results of the test data are considered out-of-sample back testing.

Once the ML model is built with the training data, it is evaluated using a separate test dataset that has not yet been studied. The performance of the model is determined by the AUC value. Figure 11 shows the AUC value and ROC for the capacitor model based on the test dataset using historical notification information through 2023. The AUC value of 0.84 implies that the model possesses moderate accuracy in terms of predicting the results.



*Figure 12: Out-sample back testing result for Combined Capacitor model based on test dataset*

The impact of uncertainty in model inputs and parameters on model outputs are tested as a part of the sensitivity analysis and the results are captured in Section 3.2. In addition, the data imputations that are incorporated to address missing values before running the model are defined in Section 2.1.

### 3.4 Benchmarking Analysis

For the Combined Capacitor model, different approaches like Gradient Boosting Machine (GBM) learning, Support Vector Machines (SVM), and XGBoost were considered during the model development phase in 2019. The analysis on these supervised ML approaches and the results are provided below.

- **Gradient Boosting Machine (GBM)** is one of the most popular forward learning ensemble methods in ML. It is a powerful technique for building predictive models for classification and regression tasks. GBM sequentially combines the predictions from various weak learner decision trees and builds a final predictive model with more accurate predictions by minimizing a defined loss function.
- **Support Vector Machines (SVM)** attempts to calculate mappings in multivariate space that make for differentiation between failure classes by calculating a hyperplane that best separates the classes.

- **Extreme Gradient Boosting (XGBoost)** is an advanced implementation of gradient boosting designed for speed and performance. It is highly efficient and scalable, making it suitable for large datasets and complex models. XGBoost includes regularization to prevent overfitting, handles missing values gracefully, and provides detailed feature importance scores. Additionally, it is optimized to run efficiently on cloud platforms like Google Cloud, leveraging distributed computing to handle large-scale machine learning tasks with reduced computation time and cost.

The benchmarking results of GBM and SVM shared in this section were developed using scikit learn library on the Test data with targets from the last 5 years of historical failure data (2019-2023). Since benchmark results were not saved during the model development phase, the benchmark models were executed in August 2024 for documentation purposes. Figure 12 provides the AUC values for the Combined Capacitor model using the GBM, SVM, and XGBoost methodologies.

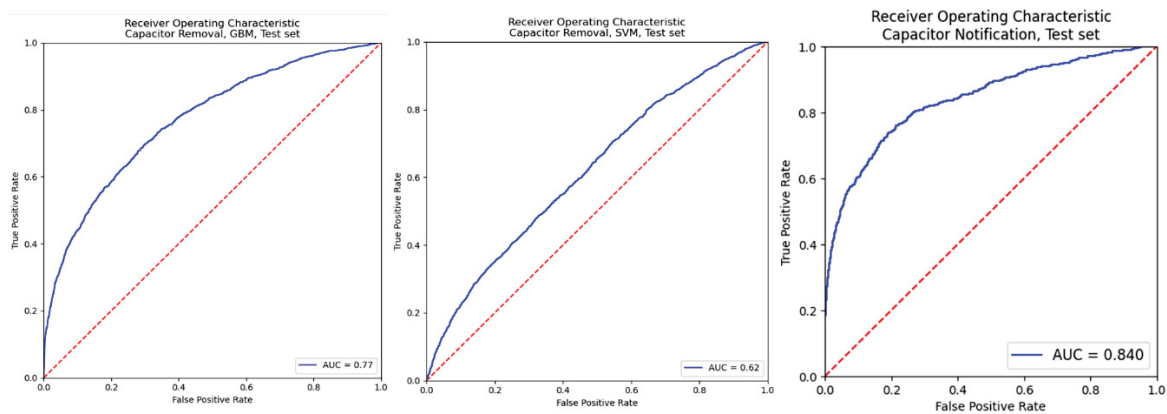


Figure 13: Gradient boosting, Support vector machines, and XGBoost ML algorithm performance on the capacitor predictive model data set compared

For the Combined Capacitor model, the AUC results for GBM, SVM, and XGBoost were 0.77, 0.62, and 0.84 respectively.

SCE chose XGBoost for the Combined Capacitor model as it aligns with the modeling approach for SCE's other predictive asset failure models and achieved a slightly higher AUC than GBM. Some additional advantages of using XGBoost over GBM and SVM are provided below:

- **Efficiency and Speed:** XGBoost is designed to be highly efficient and can handle large datasets faster than traditional GBM. It uses advanced optimization techniques and parallel processing to speed up training.
- **Regularization:** XGBoost includes L1 (Lasso) and L2 (Ridge) regularization, which helps prevent overfitting and improves model generalization. This is not inherently available in traditional GBM.
- **Handling Missing Values:** XGBoost has a built-in mechanism to handle missing values, making it more robust and easier to use with real-world datasets that often have missing data.
- **Tree Pruning:** XGBoost uses a more sophisticated tree pruning algorithm, which helps in reducing overfitting and improving model performance.
- **Cross-Validation:** XGBoost has built-in cross-validation capabilities, allowing for more straightforward model evaluation and tuning.

- **Scalability:** XGBoost is highly scalable and can be distributed across multiple machines, making it suitable for large-scale machine learning tasks.
- **Efficiency on Google Cloud Platform:** XGBoost is optimized to run efficiently on Google Cloud Platform, leveraging cloud infrastructure to handle large datasets and complex models with reduced computation time and cost.
- **Feature Importance:** XGBoost provides detailed feature importance scores, helping with understanding the impact of each feature on the model's predictions.
- **Better Performance than SVM:** Decision tree ensembles have several advantages over support vector machines. They are less computationally intensive, more robust to outliers and non-linear data, and benefit from better explainability thanks to metrics like feature importance.

## 4. MODEL MANAGEMENT AND GOVERNANCE

### 4.1 Ongoing Monitoring Plan

Ongoing monitoring is important for ML models especially when used to make predictions or when they are run on datasets with high volatility in variable values. The Capacitor model is run manually once a year, incorporating updated input datasets to reflect the latest available data and implementing any specific model enhancements, e.g., inclusion/replacement/removal of a feature, optimization of the code, evaluation of a new performance metric, etc. During the model refresh, the limitations and assumptions of the model are also revisited by the model developers and necessary action items are conducted to address them.

Performance monitoring is required only after running the model. The AUC and accuracy rate from Confusion Matrix results obtained after model refresh are compared against a threshold of 70%. If the value drops below this threshold, the reason behind the performance dip is investigated. Post-investigation, the steps required to improve the performance of the model will be carried out. To monitor the model performance more thoroughly, the developers of the model plan to additionally evaluate metrics like Precision and Recall. Precision is the positive predictive value which represents the proportion of predicted failures that were predicted correctly. Recall is the true positive rate which represents the proportion of actual failures that were predicted correctly.

The model documentation and the performance results are updated once a year immediately after the model refresh.

### 4.2 Security and Control

The Advanced Predictive Modeling team has access to the data inputs, code, and implementation for the model. Other business units, like the Grid Hardening Strategy team, are provided with access to the model outputs upon request but cannot update or modify the code.

The model is run using Python programming and it can be executed in Python 3. Current model versioning is labeled by year of refresh (e.g., 2024 refresh). The code is saved on GitHub, a platform that facilitates version control by tracking changes to the source code. Users with write or admin privileges to the repository can review proposed changes and approve them.

A contingency plan is not applicable for this model as it is an in-house model for SCE.

## 5. REFERENCES



Southern California Edison (SCE) Model Documentation Prepared for 2026-2028 WMP Appendix B  
Combined Capacitor Sub-Model

RF 1: SCE's WMP 2022 Q1 Quarterly Data Report submission

<https://www.sce.com/sites/default/files/AEM/Data%20Requests/2022/SCE%20Q1%202022%20Tables%201-12.xlsx>

RF 2: Literature reference on grid search vs random search approach for hyperparameter tuning.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(1), 281-305.

RF 3: Variable Importance methodology for tree-based methods

[Variable Importance — H2O 3.38.0.3 documentation](#)

RF 4: Boosting vs. Bagging Visual Reference

Odibat, O. (n.d.). Boosting Algorithms. SlideShare. Retrieved May 2, 2025, from [URL]