

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

Combined Transformer 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	6
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	16
3.1 Model Specification Testing.....	16
3.2 Sensitivity	19
3.3 Outcome Analysis / Back testing	21
3.4 Benchmarking Analysis	22
4. MODEL MANAGEMENT AND GOVERNANCE	25
4.1 Ongoing Monitoring Plan	25
4.2 Security and Control	25
5. REFERENCES.....	26

1. EXECUTIVE SUMMARY

1.1 Model Purpose and Intended Use

The Combined Transformer 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) transformer. The Combined Transformer model is refreshed annually and used to predict the probability of failure (POF) for distribution OH and UG transformers.

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

1. Inspections and Remediations programs that consider POI as an element in prioritization and scoping.
2. Risk analyses via SCE’s Multi Attribute Risk Scoring (MARS) Framework.

1.2 Model Description Summary

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

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 five years of historical failure data.

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

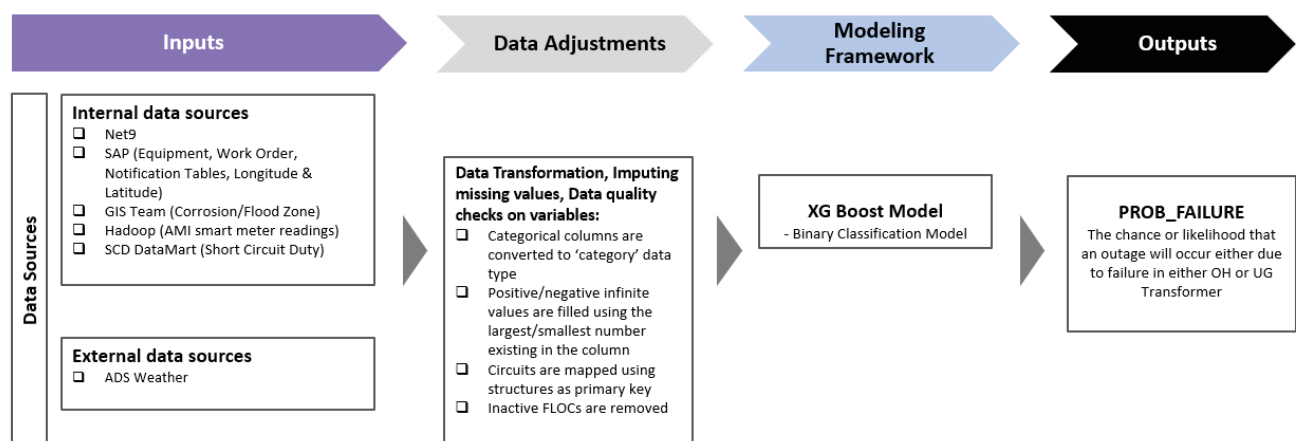


Figure 1: Transformer model framework

The combined Transformer 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 viable choice for the Transformer 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 OH transformers averages 2,311. This frequency is medium compared to other sub-drivers. Figure 2 provides a snapshot of the count of outages over the years by OH transformer 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 Transformer model is deemed to be a high risk model.

A	B	C	D	E	F										G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	
					Number of risk events																							Projected risk events										
Risk Event category	Cause category	#	Sub-cause	Are risk in	2015	2016	2017	2018	2019	2020	2020	2020	2020	2021	2021	2021	2021	2022	2022	2022	2022	2023	2023	2023	2023	Unit(s)	Comments											
Outage - Distribution		18.n.	Transformer damage or failure - Distribution	Yes																									522	603	1029	537	# risk events (excluding ignitions)					
					1889	1649	1978	2594	2489	416	559	1890	536	403	547	724	501	288	613	1053	556																	

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

References: Refer to link [RF 1 1] in Section 5 for SCE’s Wildfire Mitigation Plan Q1 2022 Quarterly Data Report submission.

1.4 Model Dependency and Interconnectivity

The Combined Transformer 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.

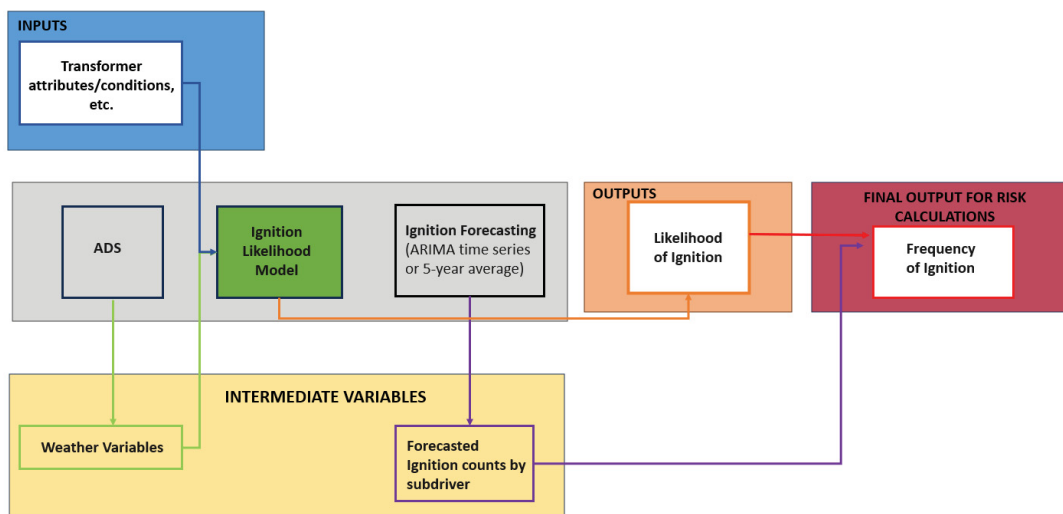


Figure 3: Model Interconnectivity Schema

ADS weather variables are used as one input in the Combined Transformer 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 transformers by their latitude and longitude coordinates.

The Combined Transformer model uses weather data, and historical attributes of transformers as input. The output data from the Combined Transformer model (i.e., POI) is used to inform the strategic decisions of the two categories of programs discussed in Section 1.1.

1.5 Model Assumptions

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

1. There is no change in the Transformer 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 Transformer 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 be reduced 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 Transformer 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 Transformer model was evaluated on test data using transformer replacement data.

- The AUC value is 0.9394
- Confusion matrix results capture the accuracy rate as 91.83%

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 Transformer model is a binary classification model pertaining to OH and UG transformer equipment failures. It employs an XGBoost algorithm to predict the likelihood of a transformer experiencing a failure that can result in an ignition event. The XGBoost approach was chosen for the classification task over other modeling approaches—such as logistic regression, random forest, and gradient boosting methods—because it predicts output with high accuracy, runs efficiently on large datasets, and maintains accuracy with default 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 circuit¹, structure, and equipment characteristics. It contains latitude and longitude information of the assets. SAP also provides failure targets and notification data via the nameplate of the transformer.
- **GIS Data** contains corrosion and flood zones from GIS team
- **Advanced Metering Infrastructure (AMI)** smart meter readings in HADOOP are used for loading information on the transformers.
- **Short Circuit Duty (SCD) DataMart** is a consolidated dataset built on Integrated Capacity Analysis and Net9. It provides segment-level SCD values across the territory. SCD represents the current that equipment experiences during a fault, indicating the stress on the equipment and its ability to clear faults quickly.

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 switches 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 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 Transformer model to ensure accuracy,

¹ 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.

validity, integrity, and consistency are provided below. Quality checks (QC) are incorporated and coded in Python.

The QC steps performed by automated Python code are as follows:

- SAP-provided data is checked for date issues, spelling issues, and duplication.

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 transformer failures, 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 60% in train data and 40% in 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.

- Categorical columns are converted to the 'category' data type.
 - Subtype
 - PRIMARY_CATEGORIES
 - User_Status_UDF
 - REGION_UDF
 - Corrosivity
 - FLD_ZONE
 - MANUFACTURER_GROUP
 - Model_GROUP
 - KVA_GROUP
 - OHUGInd_UDF
- Positive infinite values are filled using the largest number existing in the column.
- Negative infinite values are filled using the smallest number existing in the column.
- Inactive FLOCs are removed
- Circuits are mapped to FLOCs using structures as primary key

Data Assumptions

The accuracy of the predicted results is dependent on the accuracy of the data used to build the predictive models. Following are the data assumptions:

1. The assumptions for the data imputation uses SCE's Distribution Design Standard (DDS), engineering judgment, manufacturer data, and acceptable engineering practices.
2. 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 is refreshed.

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

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 labeling or missing data for a specific feature (predictive variable) that may impact model prediction power.

- Data labeling issues might be caused by manual errors during data entry. For example, when manufacturer/model is fed manually 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.
- Missing data for a specific feature might be due to unavailability of data. The data is imputed from other assets or location features if possible.

Independent variables

The Combined Transformer model uses multiple variables/features. Key features are provided below:

Feature	Original Data Source	Current Data Location	Description
AGE_UDF	SAP	Snowflake	Transformer age, calculated by referring to IN_SERVICE_DATE
CLIMATECODE_UDF	SAP	Snowflake	Climate Zone
Percent_Time_Overloaded	AMI Data	Snowflake	Percentage of time transformer overloaded
Peak_Loading	AMI Data	Snowflake	Maximum peak loading over last 5 years
KVA_UDF	SAP	Snowflake	KVA size of transformers
FLD_ZONE_X	GIS	GCP	Flood Zone
Subtype_BD-BURD	SAP	Snowflake	Transformer subtype, one hot encoded to represent BURD
Subtype_CV-CONVENTIONAL	SAP	Snowflake	Transformer subtype, one hot encoded to represent CONVENTIONAL
Model_GROUP_Group0	SAP	Snowflake	Identify model of transformer, one hot encoded, in this case Group 0
REGION_UDF_SanJoaquin	SAP	Snowflake	Region—geographic information from SAP, cleaned and one hot encoded for consistency and impute missing values, this variable was one hot encoded for San Joaquin
FINAL_SCD	SAP	SCD DataMart	Short Circuit Duty. Represents amount of fault current at that location

Table 1: Key Features in the Model

Transformer age, manufacturer, and models are provided through 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, cloud fraction, shortwave flux, rain, and snow.

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 Transformer represents the observation of a transformer equipment failure in terms of removals, replacements, and repairs. 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 transformer failure will occur. A function is used to specify the desired output (`PROB_FAILURE`) in probability values, rather than binary values. The probability value ranges from 0 to 1 where '0' represents the least likelihood for a failure and '1' represents the high chance for a failure.

The failure targets are those identified as Transformer failure in SAP. Then, Google Cloud's batch prediction is used to specify the desired output (`PROB_FAILURE`) in probability values, rather than binary values.

2.2 Methodology

SCE uses ML to identify patterns that may lead to failures causing sparks from transformers and uses the trained model to predict Probability of Ignition (POI) at the asset level. The Combined Transformer 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.

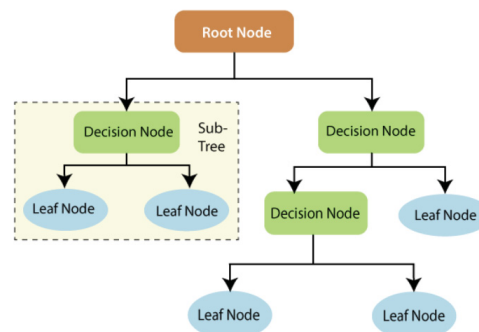


Figure 4: Decision Tree Structure

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.

Prior year model refresh was a Random Forest model, which uses bagging (bootstrap aggregating). Bagging is an ensemble technique that trains multiple models in parallel on different subsets of the training data and then combines their predictions to improve accuracy and reduce variance.

The latest model refresh uses the XGBoost algorithm. 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.

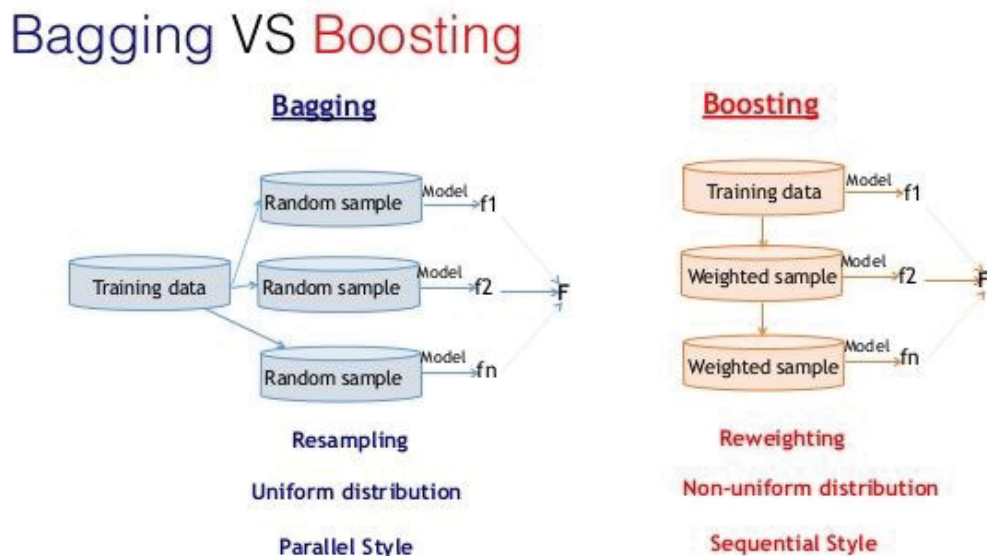


Figure 5: Structure of Boosting vs. Bagging

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 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 (60%) and testing dataset (40%) based on simple random sampling strategy with a split ratio of 3:2 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.

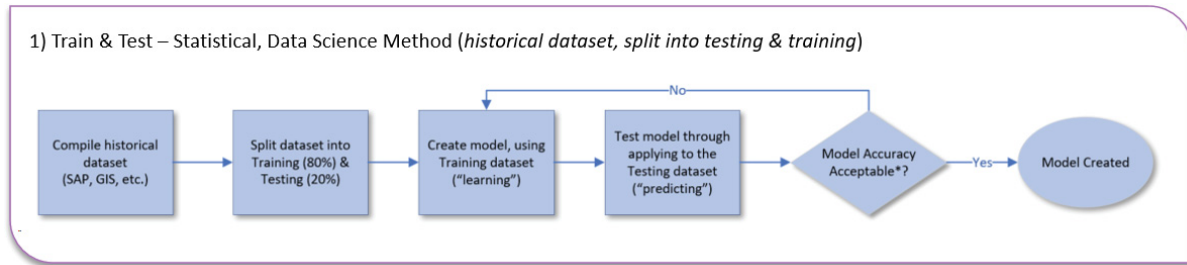


Figure 6: Train and Test data split logic

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 a failure or success. 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 to use is known as hyperparameter tuning, and the tuning process to achieve the best-defined performance statistic is known as hyperparameter optimization. Cartesian Grid search and Random Grid search are widely used strategies for hyperparameter optimization. However, in the latest model refresh, hyperparameter tuning was done manually to save time and computing resources. The optimal parameters are found below.

- **max_depth:** Specifies the maximum number of decision splits allowed within a tree. A lower value prevents overfitting by limiting the complexity of the model. In this case, max_depth=6.
- **min_child_weight:** Defines the minimum sum of instance weight (hessian) needed in a child. A higher value can prevent overfitting by making the algorithm more conservative. Here, min_child_weight=1.
- **gamma:** Controls whether a given node will split based on the expected reduction in loss after the split. A higher value makes the algorithm more conservative. In this case, gamma=0.
- **subsample:** Denotes the fraction of samples to be randomly selected for each tree. Lower values prevent overfitting by introducing randomness. Here, subsample=1.
- **colsample_bytree:** Specifies the fraction of features to be randomly selected for each tree. This helps in reducing overfitting. In this case, colsample_bytree=1.

- **learning_rate:** Also known as eta, it controls the step size at each iteration while moving towards a minimum of the loss function. Lower values make the model more robust to overfitting but require more trees. Here, learning_rate=0.3.

2.3 Suitability

During 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 sought for the latest model 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 use of XGBoost for the Combined Transformer model is deemed to be a suitable fit.

2.4 Assumptions

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

BA 01: There is no change in Transformer technical specification over time. The model assumes the type of transformers used in the model building process have the same characteristics in terms of build and quality. For example, transformer voltage is constant.

BA 02: The Calibration model assumes that fires are a subset of failures. Failures prompting the need for removals, replacements, and repairs are the representative failure targets used in place of a few ignition events. Some of these issues left unaddressed can potentially spark an ignition, but not all failures will 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 considered by 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 provides 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, highly correlated variables in XGBoost approach will impact 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 statistic.

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 the 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 for real time models as they would run more frequently. Since the Combined Transformer model is run only once a year with reasonable use cases, the impact of resource utilization is low. 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 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, 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 (60%) and test (40%) 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 Transformer model predicts the probability of ignition (POI) arising from equipment (transformer) failure. The model has a single output characterized by a continuous number between 0 and 1 for each Transformer asset.

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.

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

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 against 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 weight the model probabilities. The sum of probabilities from each calibrated model equals the forecast by sub-driver.

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

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

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

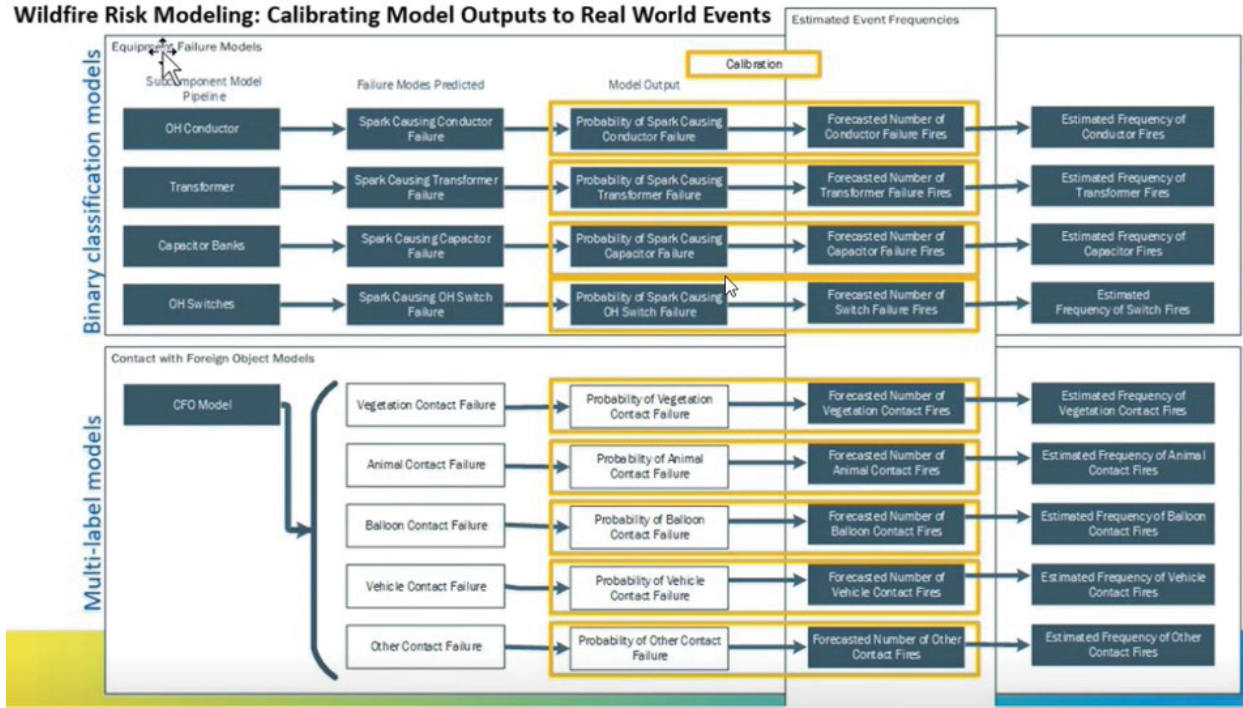


Figure 7: Calibration model schema

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 Transformer model is used to inform the programs mentioned in Section 1.1.

Model Changes:

Previous models were separated into independent Underground and Overhead Transformer Models. The latest refresh combined both datasets into one Combined Transformer Model.

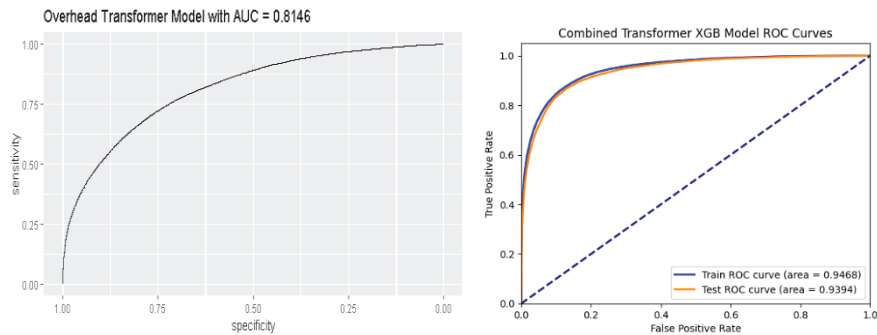


Figure 8: AUC results for 2023 and 2024 Transformer Model

The change in the AUC indicates the impact on model performance from the changes made as a part of the yearly refresh. For example, in Figure 8 the AUC value of the OH Transformer model was 0.8146 in 2023, before the refresh, and the AUC value of the Combined Transformer model was 0.9394 in 2024, after the refresh.

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 XGboost. The model is run once a year manually by the Advanced Predictive Modeling team with refreshed asset and weather data. The model is calibrated every year with the last 5 years of historical failure data.

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, mentioned in Section 2.4, are discussed below:

- 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 Transformer model employs several independent variables. Section 2.1 contains a list of the independent variables utilized in this model.

The variable importance test results for the Combined Transformer model, Figure 9, shows 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 Transformer Sub-Model

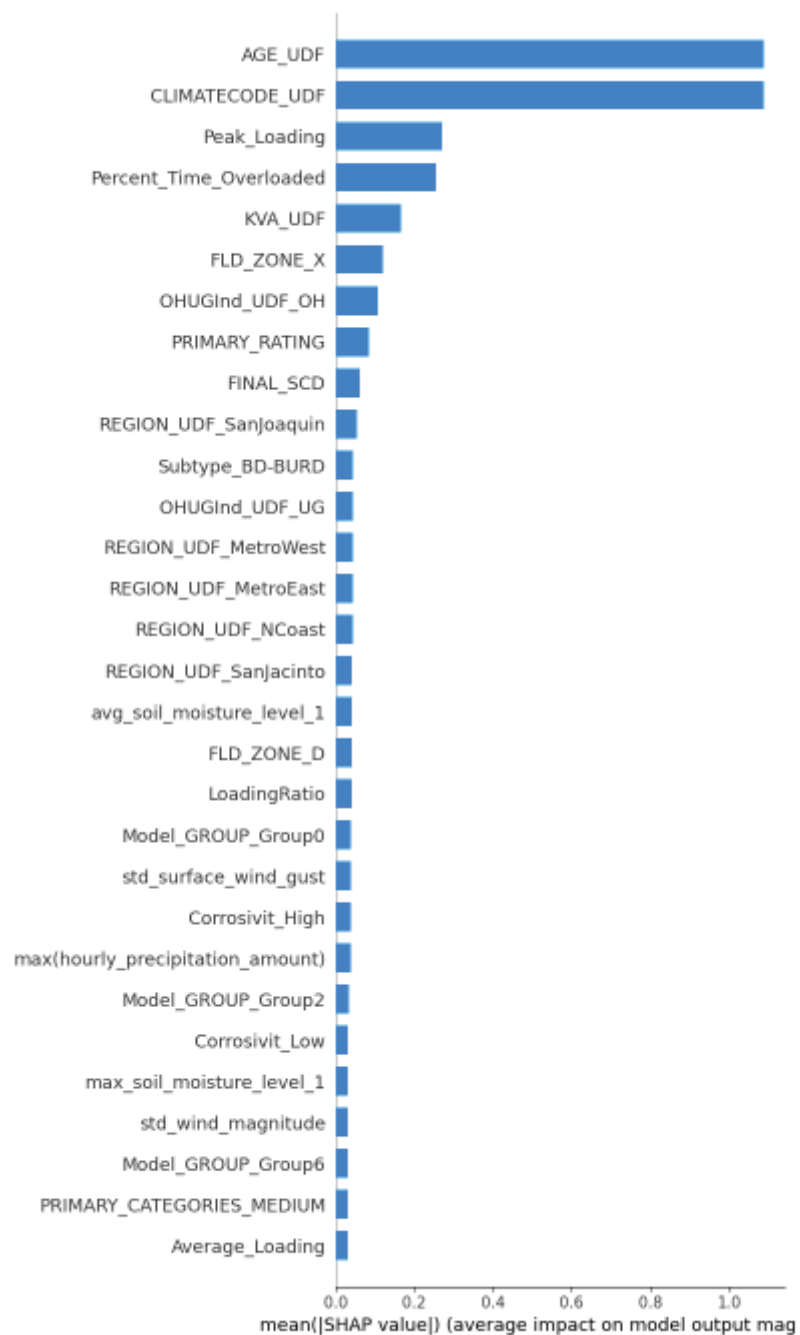


Figure 9: Variable Importance test results for Combined Transformer model

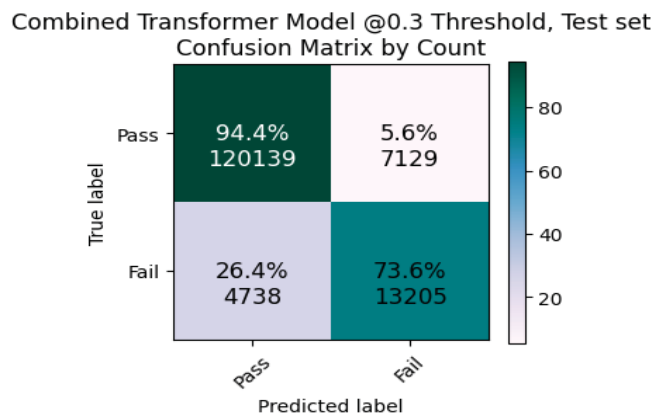
The results confirm that Age, CLIMATE_CODE, KVA and loading information exhibit high importance on the model output.

References: Refer to link **[Error! Reference source not found.]** in Section 0 for description on the methodology used to perform the Variable Importance for tree-based methods.

The Combined Transformer model had some parameter tuning to select the best set of hyperparameters to achieve maximum performance in terms of AUC as described in Section 2.2. The same parameters continued to be used in the refresh of the data. In terms of model convergence, the model runs to max_depth = 6 deep, with min_child_weight = 1 which is the minimum sum of instance weight needed in a child. A higher min_child_weight can prevent overfitting by making the algorithm more conservative. Making max_depth larger makes the model overfit the training data.

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.
- The threshold in a Confusion Matrix determines the cutoff probability for classifying an instance as positive. For example, in predicting transformer failure, setting a threshold of 0.3 means that any instance with a predicted probability of failure of 30% or more is classified as a failure.



- Table 1: Confusion matrix results for Combined Transformer Model**Error! Reference source not found.** provides the Confusion Matrix results for the Combined Transformer model. It captures the accuracy rate as 91.83%.
- 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.

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

The error rate for the transformer model is 8.17%.

All these test results are performed on test dataset with removal for failure data set from SAP.

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

3.2 Sensitivity

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 Transformer Sub-Model

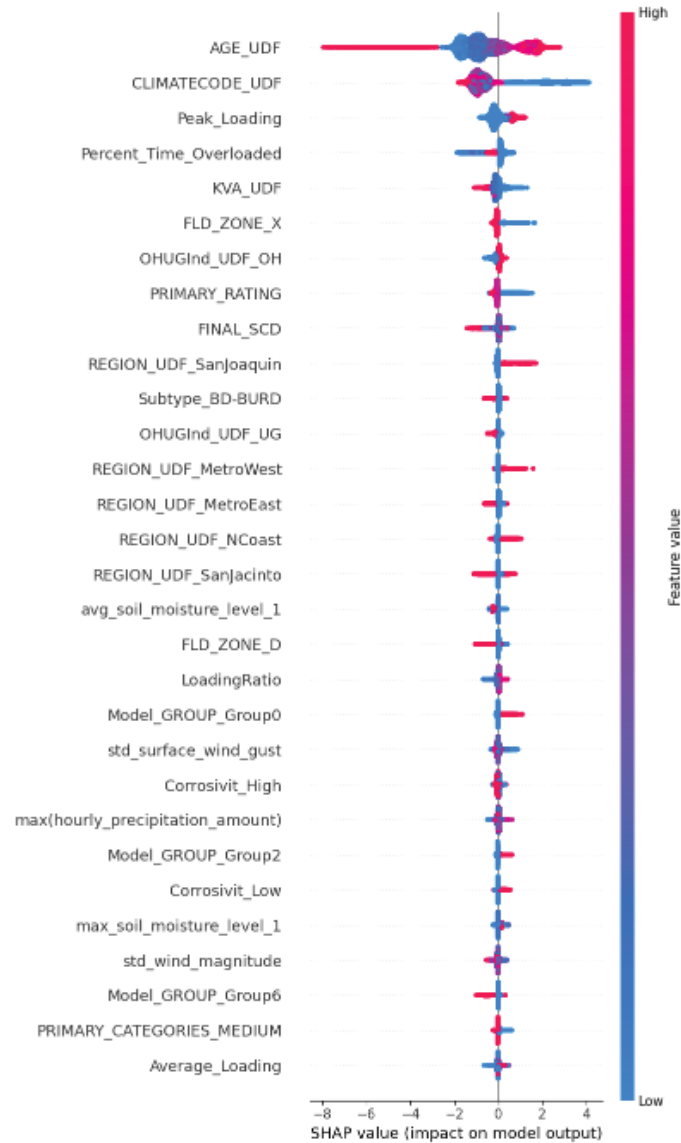


Figure 10: Variable Importance with SHAP values gives further insight into how the variables interact with the probability of failure results.

From this Plot we extract features that were most important in prediction and some that had an interesting interaction with prediction for the Transformer model.

The Top 3 Contributors to Failure:

- AGE_UDF: This is the age of the transformer and looking at the partial plot of it, yields expected results—newer transformers have lower predicted levels of failure.
- Climate_Code: Sourced from Fire Sense team, categorizes different segments of SCE service areas by Climate.
- Peak_Loading: 75% of the data indicates 0% of Percent_Time_Overloaded. Consequently, it is reasonable to conclude that a significant number of failures fall into this category. This observation suggests that overloading is not the primary cause of these failures. However, for

instances where overloading does occur, there appears to be a linear relationship with failure, particularly as the overloading percentage approaches 100%.

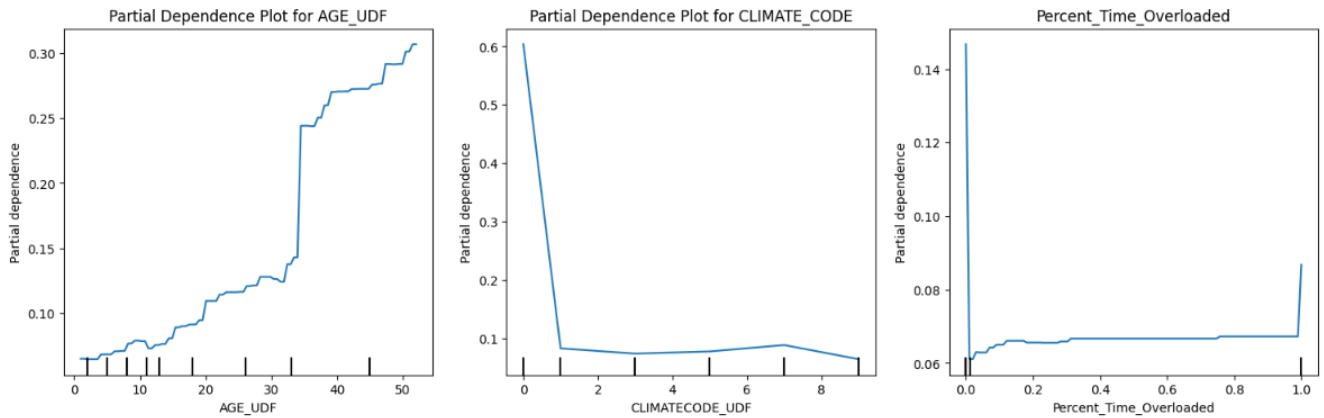


Figure 11: Partial Dependence Profiles for Age, Climate Code, and Percent Time Overloaded

3.3 Outcome Analysis / Back testing

The subset of historical data on which a model is trained and optimized is called 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 Transformer model uses a random sampling approach to split the dataset into Train (60%) and Test (40%) 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 model's performance is determined by the AUC value. Figure 12 shows the AUC value and ROC for the Combined Transformer Model based on the test dataset using transformer replacement due to failure data until 2023. The AUC value of 0.9394 implies that the model possesses high accuracy in terms of predicting the results, with room for improvement in future iterations.

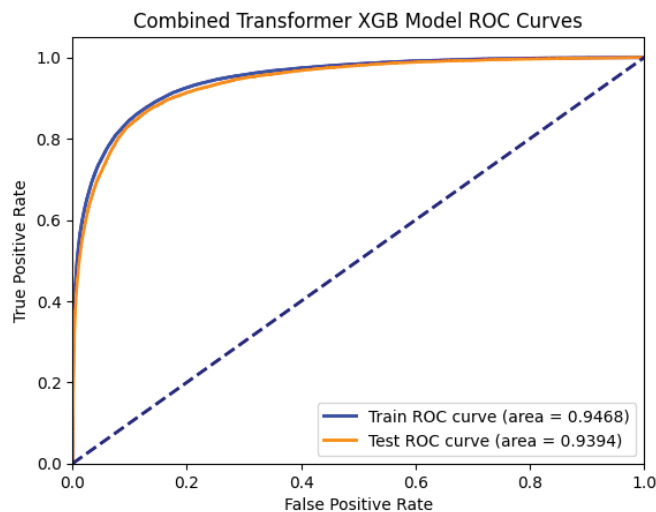


Figure 12: Out-sample back testing result for the Transformer Model based on test dataset

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 **Error! Reference source not found.**. 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 Transformer Model, different approaches like XGBoost, Gradient Boosting Machine (GBM) learning, Logistic Regression, and Random Forest were considered during the model development phase in 2024. 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.
- **Logistic regression** is used to solve classification problems. The three types of logistic regression available are Binary logistic regression (handles binary outcomes), Multinomial logistic regression (handles multiple outcomes, i.e., multi-classification variable), and Ordinal logistic regression (handles ordered outcomes). In contrast, linear regression solves regression problems where the outcome is continuous and can be any possible numeric value.
- **Random Forest** is a popular ML algorithm that can be used for both classification and regression problems. Random Forest is another ensemble method that combines the predictions of several decision trees to improve the predictive accuracy of the model. The individual decision trees are created based on a randomly selected subset of features at each node prior to determining the optimal split so each tree differs. The final output is determined by taking the majority vote of the predictions from the individual decision trees. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.
- **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, can handle missing values well, 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 ML tasks with reduced computation time and cost.

The benchmarking results of GBM and Logistic Regression shared in this section were developed using the sklearn library in Python on the Test data with targets based on the transformer replacement due to failure data until 2023. Since benchmark results were not saved during the model development phase, the benchmark models were executed in August 2024 for documentation purposes. Figure 13 provides the AUC values for the Combined Transformer model using the GBM, Logistic Regression, and Random Forest methodologies.

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

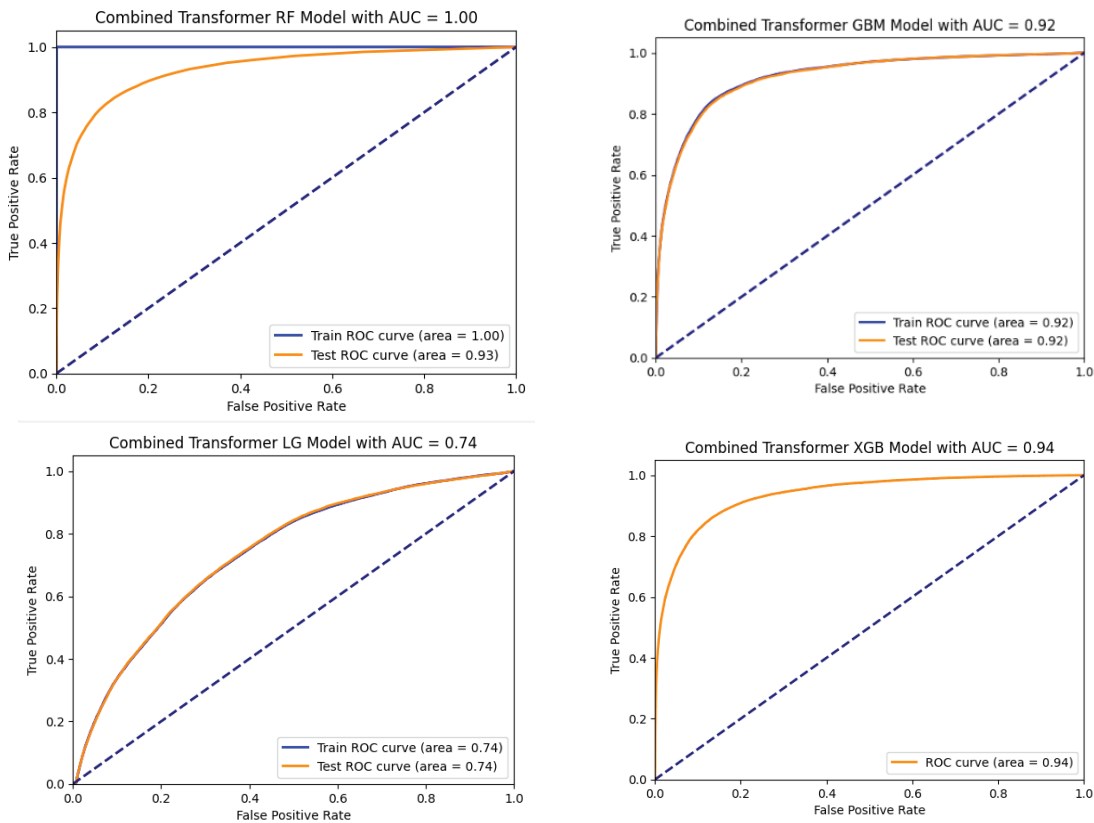


Figure 13: AUC Comparison for the Transformer Model using Random Forest, GBM, Logistic regression and XGB methodologies. GBM is the gradient boosting method, LG is the logistic regression, XGB is the XGBoost, and RF is the Random Forest method.

SCE chose XGBoost for the Combined Transformer model as it aligns with the modeling approach for SCE's other predictive asset failure models and achieved a slightly higher AUC than GBM. While Random Forest had a testing AUC of 0.93, its training AUC was 1 which is completely overfit. SCE will continue to consider GBM as a methodology for use as part of the annual refresh of the model. Some additional advantages of using XGBoost over GBM and Logistic Regression 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.

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 Transformer 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 actions are taken 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 model's performance 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 access to the model outputs upon request but cannot update or modify the code.

The model is run using Python and is executed in Google Cloud Platform. Current model versioning is labeled by year of refresh (e.g., 2024 refresh) within the repository `combined_transformer_model`. 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

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: Boosting vs. Bagging Visual Reference

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

RF 3: Variable Importance methodology for tree-based methods

[Variable Importance — SHAP Documentation](#)