Fan Power Unit Fault Detection Analysis

Syefira Shofa & Snigdha Kanuparthy

Abstract

In the following investigation we aim to provide a comprehensive analysis of two variants of the fan-powered unit in heating ventilation and air conditioning systems conducted on behalf of Joulea. In analyzing the fan-powered unit data, we can identify patterns in inefficiencies that can lead to fan unit failures. By detecting patterns in inefficiencies as they correlate to failures, we may lead to significant savings by reducing the number of replacements and/or emergency services required for proper maintenance of these units.

Our work represents an opportunity for Joulea to augment their existing building energy model in order to enhance energy efficiency as it relates to fan-powered units. By leveraging these insights, Joulea can offer more comprehensive building anomaly detection and a higher caliber product, specifically augmenting the capabilities of the envelope assessment.

By integrating these findings in strategic planning, Joulea can position themselves as a market leader with comprehensive and thorough data analysis and industry leading machine-learning techniques as it comes to energy efficiency and cost savings within that market.

Background

A fan-powered unit (FPU) in heating ventilation and air condition (HVAC) system operates by utilizing a fan to distribute air, adjusting airflow and supply air temperature to meet the heating, cooling, and ventilation requirements of different zones within a building. The two types of fan operation in an FPU are Parallel fan-powered variable air volume (VAV) units (PFPU) and Series fan-powered VAV units (SFPU).

The Parallel Fan-Powered Variable Air Volume (PFPU) units work alongside the main air-handling system, utilizing a separate fan for continuous airflow. The primary system adjusts airflow based on temperature needs, providing individual zone control. The "static pressure set point" for PFPU is the desired pressure level within the system, measured in inches of water gauge (in.w.g.). In this case, the target pressure is 1.40 inches of water gauge, with an acceptable range of +/- 0.13 inches. This measurement indicates the force needed to effectively distribute air throughout the building.

Conversely, the Series Fan-Powered Variable Air Volume (SFPU) units operate in series with the main air-handling system. They draw air from the main ductwork and distribute it through a terminal box, maintaining a constant airflow rate. Temperature regulation is achieved through supply air temperature modulation. The "static pressure set point" for SFPU is also measured in inches of water gauge. In this case, the target pressure is 0.7

inches of water gauge, with an acceptable range of +/- 0.13 inches. This measurement reflects the force required for optimal performance in SFPU units, particularly in environments with stable occupancy levels where temperature control primarily relies on adjustments to air temperature.

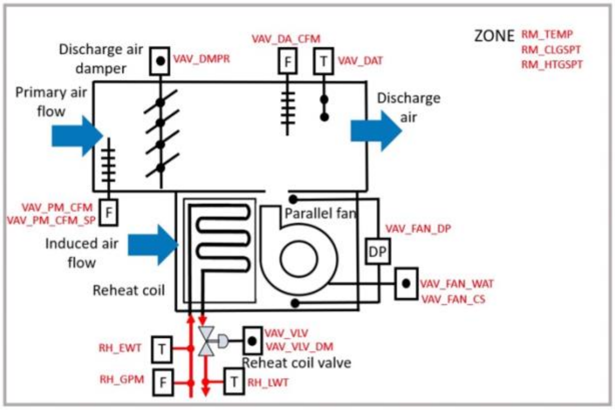


Figure 2: PFPU Schematics

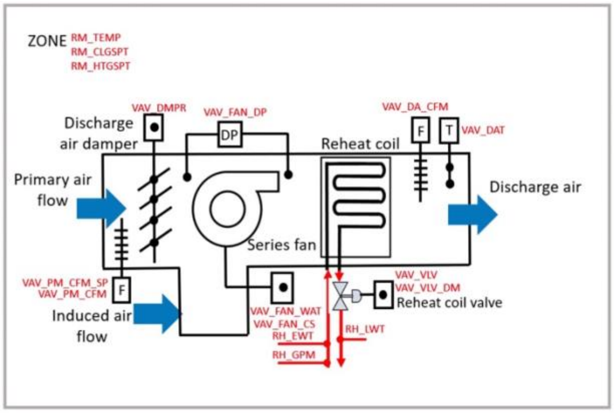


Figure 1: SFPU Schematics

Faults within these fan-processing units can lead to the following adverse outcomes for a building:

1. **Reduced Efficiency and Increased Operating Costs:** A faulty unit may lead to decreased operational efficiency, which in turn leads to an increased carbon footprint. The system will overcompensate for the fault by requiring more energy to overcome the decreased throughput. This increase in energy results in higher bills over the duration of the HVAC lifespan.
2. **Increased Wear and Tear:** Faults within units may also shorten the lifespan of the units themselves. This decreased lifespan within the HVAC unit will lead to more frequent repairs or replacements and a higher cost associated with building maintenance.
3. **Regulatory/Compliance Issues:** Inefficient HVAC systems may lead to non-compliance with energy efficiency standards (such as LEED standards) for buildings—this can lead to a loss of tax credits, a breach of zoning approval, higher utility rates, and a stigma as the value of the building decreases.
4. **System Reliability:** In highly regulated environments such as sterile rooms in laboratories, intensive care units in hospitals, and highly secure data centers, constant environmental conditions are vital—the fluctuations posed by a faulty unit can jeopardize critical functions.

5. **Quality of Life and Comfort:** Faults in the fan processing unit can lead to uneven temperatures, humidity increases, insufficient air circulation, and poor air quality due to decreased air circulation. These issues directly impact the comfort and well-being of building occupants and can result in the loss of productivity due to a less than ideal work environment.

The provided FPU data can be used to many ends, including predictive maintenance, fault classification, and anomaly detection. As such, many different predictive models can be used singly for a task or combined into an ensemble-type model. This project's main objective is to deploy a model capable of accurately identifying faults within a given data set. The project entails conducting a thorough data analysis on the provided dataset, developing models specifically designed to detect and categorize faults within the data, and organizing visual representations into a mock dashboard. This dashboard serves as a demonstration of the practical business utility of the models and their outputs. The project's focus is on enhancing fault detection accuracy and presenting the results in a comprehensible and visually accessible manner through the implemented dashboard.

Data Set

Data Set Description

There are two sets of data containing categorizations of a fault-free case or a faulty case representing a single fault type at a specific severity level that can be found at https://faultdetection.lbl.gov/dataset/simulated-pfpu-sfpu/. The test data was generated by simulating a variable air volume (VAV) heating ventilation and air condition system (HVAC) system. In the system, an air handling unit (AHU) and four associated fans powered VAV terminal units in (FPU) (four parallel FPUs (PFPU) or four series FPUs (SFPU)) in four separate zones were simulated in the HVACSIM+ software tool.

The control sequences were set according to the occupied operation hours (Mon-Fri 6:00AM-6:00PM) and unoccupied operation hours (Mon-Fri 6:00PM - 6:00AM, Sat-Sun 24-hour).

Occupied Hours

AHU (Air Handling Unit) fan control refers to the methods and systems used to regulate the operation of fans. Under occupied hours, a PI controller (a smart system that adjusts how fast the supply air fan (SAF) spins to keep the air pressure steady) works with a Varied Frequency Driver (VFD), which controls the fan's speed. For the Parallel Fan-Powered Variable Air Volume (PFPU) system, the desired pressure level is around 1.40 inches of water gauge, while for the Series Fan-Powered Variable Air Volume (SFPU) system, it's around 0.7 inches of water gauge. Also, the return air fan (RAF) is set to run at 80% of the supply fan's speed, ensuring balanced airflow throughout the system.

The AHU supply air temperature control system keeps the air coming out of the HVAC system at a comfortable level. It works in different ways depending on whether it needs to cool or heat the air. If it's cooling, it uses different methods based on the outside temperature and how much cooling is needed. For example, it might use just mechanical cooling or a mix of mechanical cooling and bringing in outside air. If it's cold outside, it might just be used outside air to cool things down. When it's heating, it adjusts things in a similar way, making sure the air isn't too hot or too cold. It's like having a smart system that knows just how to make the air feel right for the people inside the building.

The terminal (zone temperature) control in the system involves two control sequences for the FPU and SFPU, each using two PI control loops. The cooling and heating setpoints are 72 °F and 68 °F, respectively. Both FPUs employ PI control loops to determine reheat coil valve position, FPU airflow setpoint, and demand damper motor speed. The minimum airflow setpoint is 200 CFM, and the maximum varies for internal, parallel, and series FPUs. The PI outputs control reheat coil valve position and damper motor speed based on temperature and airflow setpoint differences.

The low-temperature protection control logic is designed to safeguard the coils in the Air Handling Unit (AHU) during extremely low outdoor air temperatures. If the AHU mixed air temperature falls below 35°F and remains at that level for 300 seconds, the system activates a shutdown mode to prevent coil freezing. The shutdown mode persists until the end of the current day, with the system restarting at the beginning of the next day.

Unoccupied Hours

During unoccupied hours, the HVAC system operates in two distinct modes: Setback mode and Shutdown mode.

In the Setback mode, the system activates if the air temperature in any of the four zones falls below the heating setpoint or exceeds the cooling setpoint. During this mode, the system runs for a duration of 30 minutes, like the occupied mode, with specific adjustments. The cooling setpoint is set to 85 °F, and the heating setpoint is set to 55 °F. Additionally, the economizer is disabled, and the outdoor air (OA) damper is fully closed.

On the other hand, the Shutdown mode is initiated when all zone temperatures align with their setpoints or after being in the setback mode for 30 minutes. In the Shutdown mode, both fans and valves cease operation, and the zone airflow demand comes to a complete stop, ensuring energy conservation during unoccupied periods.

Data Set Alteration

Data Set Vital Characteristics

The dataset given encompasses three hundred and sixty-five days of simulated data measured at one-hundred and six points, with measurements taken in one-minute intervals. This resulted in sixty-two files and seventeen gigabytes of data. As such, the analysis presented here is performed on a subset of data to accommodate consumer commercial hardware. Plans to scale this investigation through MLOps methodology and productize the data along with integrations for data streaming will be discussed in the future work section.

The EDA for the data and the data exploration is based on a subset of the data conducted with both random sampling and stratified sampling as grouped by time. This allowed us to appropriately perform our exploratory data analysis and our machine learning algorithms without the inadvertent bias inherent in using the later parts of a time series to predict faults, as faulty equipment tends to experience a greater deviation from a standard experience near the end of its lifecycle.

To this end, the data presented in this report was split into four portions for analysis:

1. Single-fault data: data and time-series analysis with a single scenario—primarily used for exploratory data analysis and relative comparative analysis within groups
2. Double-fault data: data with one fault and a control group, often sampled within-group (SFPU and PFPU) to increase homogeneity and reduce false conflation between scenarios.
3. Multi-fault data: data with one or more faults, sampled within-group to reduce false conflation.
4. Full-data sample: data sampled across all provided files.

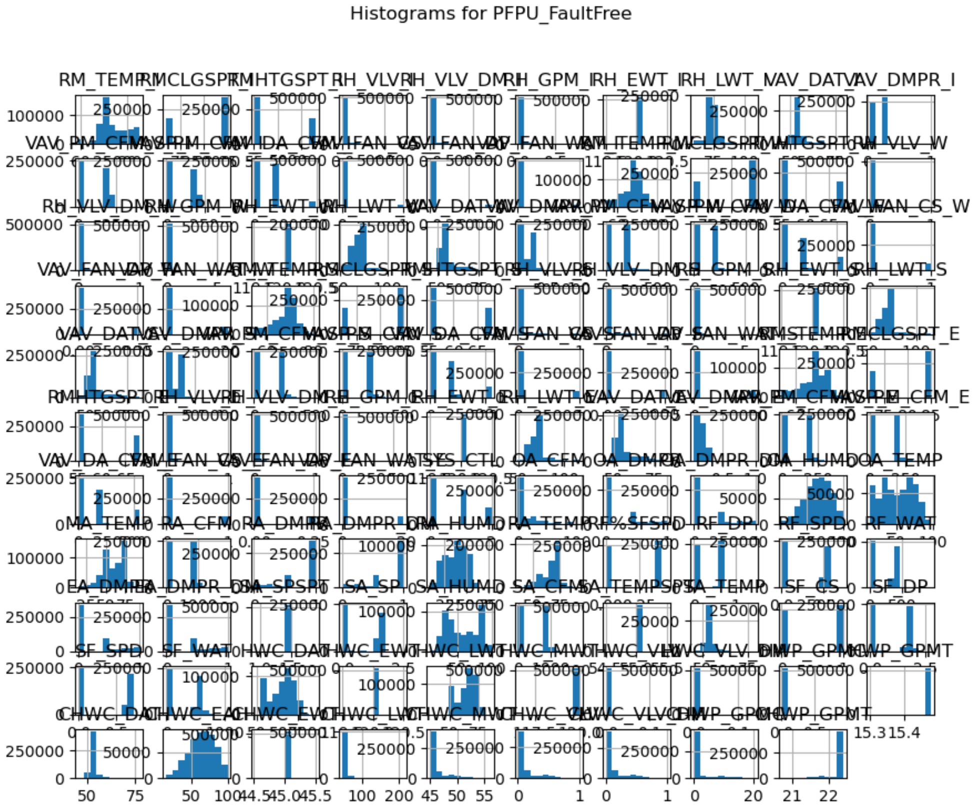
Exploratory Data Analysis

Here, exploratory data analysis attempted to gain a preliminary understanding of the faults themselves by attempting to look at both measures of centrality and dispersion within the data and contextualizing those measures within time. It is important to note that this data is structured and rooted in time, and our observations can be used to generate hypotheses about the data, which we can hope to answer and incorporate into findings for Joulea to use.

Comparing Centrality and Dispersion

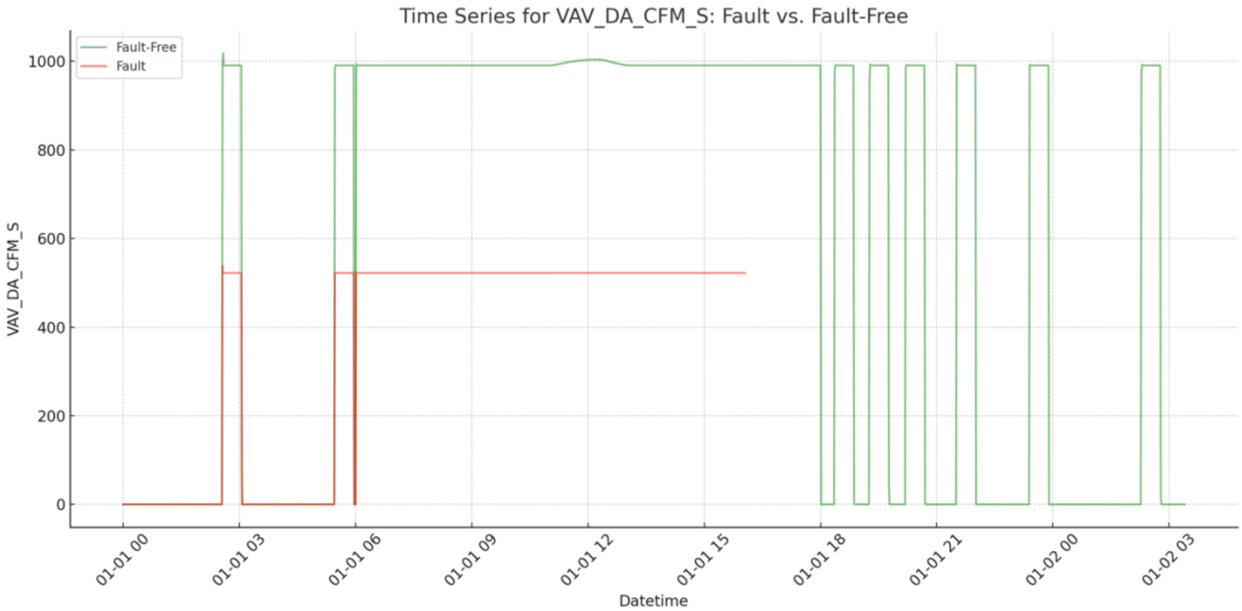
Ambient environmental factors such as room temperature, and fan heating/cooling set points are steady across data and groups, and do not have much variation between them.

Valve positions and flow rate have low means across groups but high standard deviations, and therefore much variability. These observations could indicate potential areas of critical impact for predictive algorithms when viewed through a lens of operational behavior.



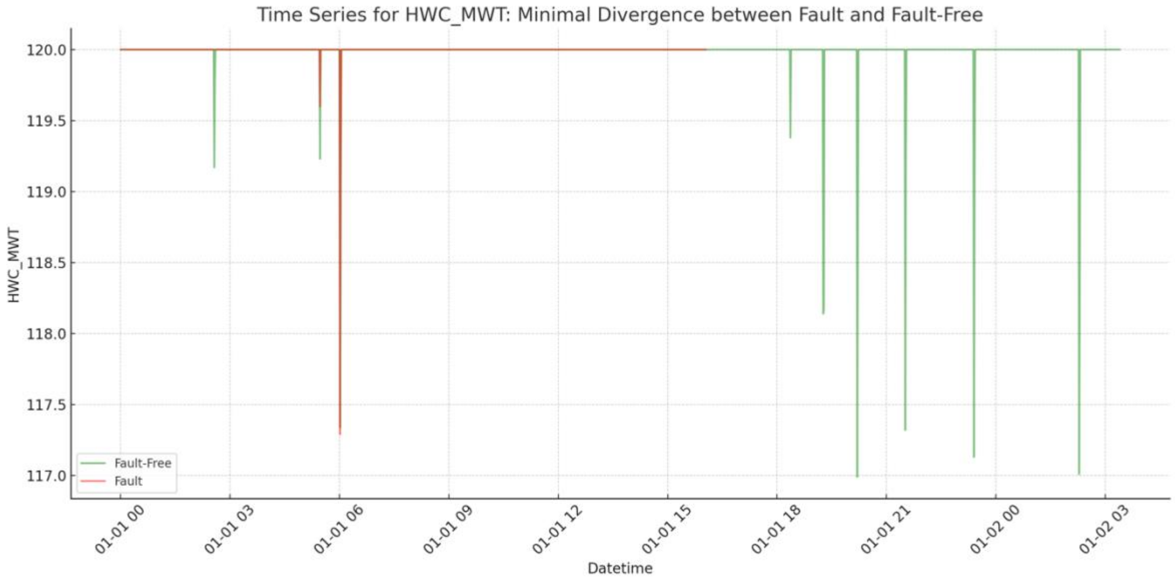
Here, we have included a diagram showing the data distribution for the PFPU-Fault Free dataset.

Our findings demonstrate that the distinct faults within the SFPU and PFPU datasets are attributable to variable conditions. In other words, specific fault types appear at first to be attributed to an interplay between specific factors.



Here, we have provided a visual of our data through time—by using the first subset of data mentioned earlier, the single-fault dataset, we can visualize the differences between a faulty and a fault-free system. Here, we have provided a computation for the SFPU VAV restriction as the faulty model, and the SFPU fault-free model.

When compared on the VAV primary air-flow set point, we observe a wide difference between the faulty and fault-free models. However, when observed on the axis shown below, namely the heating water coil mixed water temperature measurements we see more similar behavior.



As such, we can say that there may be some utility to contextualizing experiences within time, making sure to not falsely correlate faults observed in time (correlational data) to faults observed because of time based faults (false causational attribution.)

Based on the data shown above, we have decided to pursue two analysis paths with two distinct hypotheses: first, we believe that time is a neither a significant nor a distinct factor in our analysis and that we may simply use classification models that ignore time as an attribute. Second, that time is indeed a significant feature and predictor in our models, and thus all models must be contextualized within time. As such, we have tried to investigate a cohort of models that both involve and do not involve time to gain a holistic understanding of predictive capabilities.

Models

Category I: Time-Series Agnostic Models

Our first major category of investigation was the time-agnostic model. These models derived from the hypothesis that time is not a significant factor in fault—that is, a fan processing unit can be faulty at any point in time and is not caused by repeated exposure to the stimulus.

Support Vector Machine

Support Vector Machines (SVM) excel in managing high-dimensional datasets and discerning intricate patterns through effective margin maximization. Their ability to identify anomalies within datasets allows them to discern data points that deviate significantly from the majority. Moreover, SVMs exhibit robust capabilities in handling non-linear relationships within data, a crucial attribute when addressing the nuanced and non-linear behaviors associated with faults in fan-powered units. This quality enhances their applicability in fault detection scenarios. A key strength lies in SVMs' ability to determine the optimal hyperplane that maximally separates different classes in the feature space. This feature is particularly advantageous in fault classification for fan-powered units, where a clear boundary often exists between normal and faulty states. SVMs, through their precise separation of classes, prove to be effective in discerning and categorizing faults based on distinctive patterns within the data. For this project, we implemented linear SVM.

Data scaling was performed prior to training the SVM model. Scaling ensures that all features have a similar influence on the decision-making process, preventing features with larger scales from dominating the optimization process. By scaling the data to a common scale, the SVM algorithm can converge faster during training and make more accurate predictions. This preprocessing step enhances the model's performance, making it more robust and effective in handling diverse datasets.

We utilized Variance Inflation Factor (VIF) for feature selection due to its effectiveness in identifying multicollinearity among predictor variables. VIF offers a quantitative assessment of the inflation in the variances of regression coefficients caused by multicollinearity, with higher values indicating stronger multicollinearity. By eliminating features with elevated VIF values (VIF over 5), we aimed to alleviate multicollinearity issues, thereby enhancing the stability, interpretability, and predictive performance of our model. VIF-based feature selection ensures that redundant features are removed, leading to more reliable and accurate predictions while improving our understanding of the underlying relationships within the dataset.

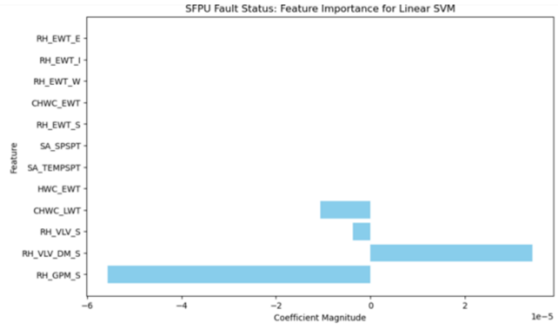
We conducted 5-fold cross-validation to evaluate the performance of the Support Vector Machine (SVM) model. This approach partitions the dataset into five subsets, with each subset serving as a validation set once while the remaining data is used for training. Repeating this process five times ensures that each data point is used for validation exactly once, providing a robust assessment of the model's generalization performance. For SVM, which relies on finding an optimal hyperplane to separate classes, cross-validation helps prevent overfitting by assessing the model's performance on multiple subsets of the data. Additionally, it provides a more reliable estimate of the model's accuracy and generalization capability, crucial for ensuring the SVM's effectiveness in handling diverse datasets.

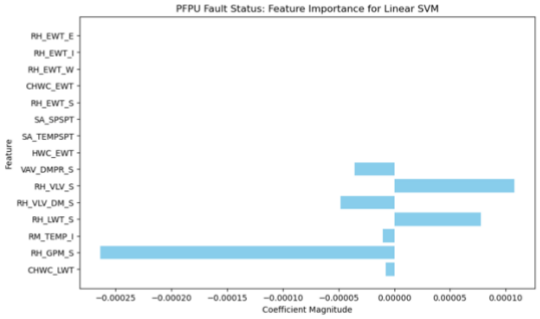
We provided visuals for feature importance, confusion matrix, and accuracy for each class to offer comprehensive insights into the performance and interpretability of our model, crucial for informed decision-making in business use cases. Visualizing feature importance aids in understanding which features contribute most significantly to predictions, facilitating strategic resource allocation and feature engineering efforts. The confusion matrix provides a clear overview of the model's classification performance, highlighting areas of strengths and weaknesses in class prediction, which can guide targeted improvements in product offerings or service delivery. Furthermore, visualizing accuracy for each class allows for a nuanced understanding of the model's performance across different categories, enabling businesses to prioritize areas requiring attention or intervention to optimize outcomes and enhance customer satisfaction. Collectively, these visuals empower business stakeholders to make data-driven decisions, optimize operational processes, and drive sustainable growth.

Dataset IV: Randomly sampling across all .csv files

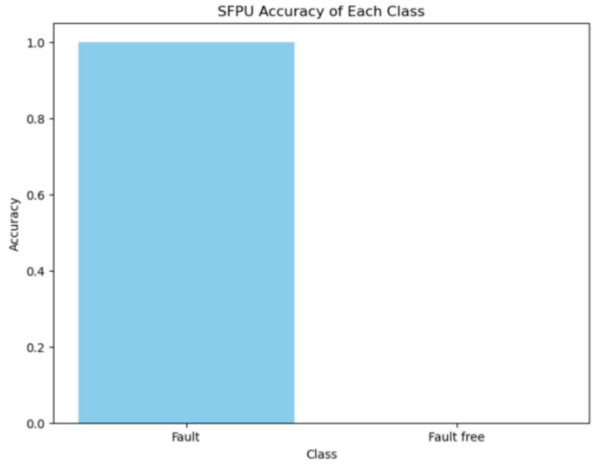
We separated the data into SFPU and PFPU subsets for SVM analysis to tailor the modeling approach to the distinct characteristics of each unit, enhancing model performance and interpretability. This separation allows the SVM to learn class-specific patterns and decision boundaries, potentially leading to more accurate predictions compared to combining the data. Additionally, due to limited computational resources, we randomly sampled the data to 100,000 rows while maintaining the original ratio of fault types, ensuring computational efficiency without sacrificing representativeness of the dataset. This approach enables us to leverage SVM effectively for fault prediction tasks within resource constraints, facilitating timely and informed decision-making in fault management and mitigation strategies.

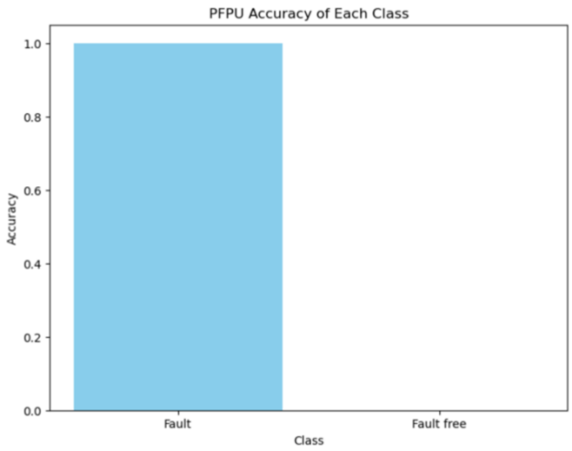
The results of the SVM led to an accuracy prediction of 97% for SFPU and PFPU. In our analysis, we found that the reheating coil water flow rate emerged as the most important feature for both SFPU and PFPU datasets. A likely hypothesis for this finding is that the water flow rate through the reheating coil significantly influences the thermal dynamics within the system. In PFPU, where parallel operation is prominent, variations in water flow rate can impact the distribution of heat across multiple units, potentially affecting the reliability and performance of individual components. Similarly, in SFPU, where series operation prevails, fluctuations in water flow rate may influence the overall efficiency of heat transfer, potentially leading to overheating or performance degradation in downstream components. Thus, maintaining optimal water flow rates through the reheating coil emerges as a critical factor for fault prevention and system reliability in both PFPU and SFPU configurations.

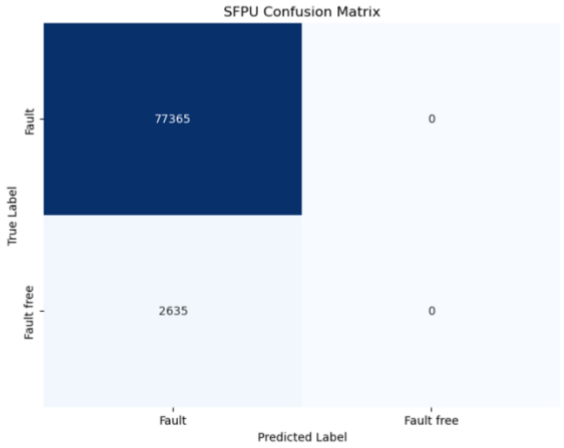


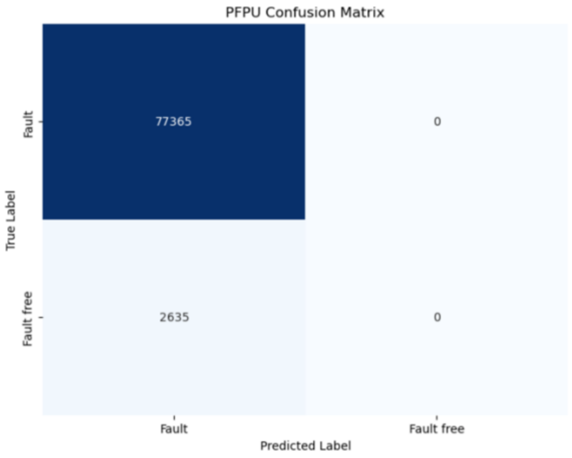


The SVM's classification was identified to bias towards faults without identifying fault-free instances indicates a significant performance flaw, undermining its utility for accurate predictions. To rectify this issue, strategies such as dataset rebalancing, feature refinement, and hyperparameter tuning could be employed. However, these solutions may demand substantial computational resources, which were unavailable in the current iteration. Without addressing this imbalance, the model's reliability and usefulness are compromised, potentially leading to detrimental consequences in real-world applications.



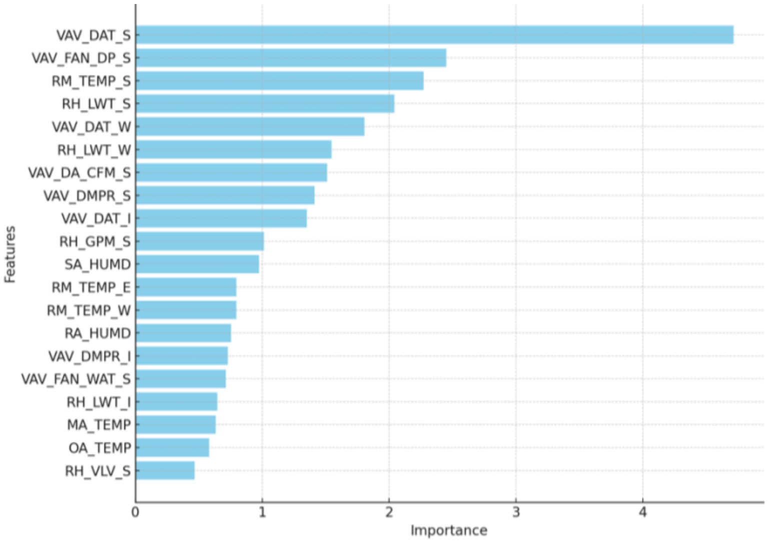




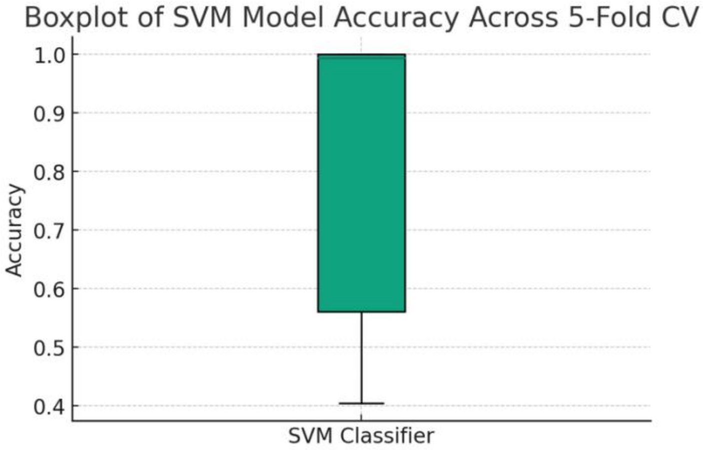


Dataset I: Sampling across one faulty SFPU dataset and the fault free dataset:

Here, we are presenting our single-fault dataset, in which we compare the SFPU Restricted Sample to a fault-free SFPU dataset. We do this to gain a baseline understanding of the data and create a binary model in the simplest context. As our prior exploratory data analysis revealed, our large variety of faults manifests in distinct and separate features in our data. This can muddy predictive accuracy if we claim to exist in a world in which a unit can only be "faulty" or "fault-free"



The primary purpose in presenting a stripped-down dataset in comparison to a more robust sampling is to show the difference in feature importances and predictive power in the SVM paradigm. As we can see, when compared to the more replete dataset the single-fault dataset is skewed and holds different feature importances; this is indicative of the variety in faults themselves as exposed by our EDA and data visualizations.



When we look at the accuracy across folds with this barer dataset, we see that a five-fold cross-validation results in an accuracy close to 100% while this is indicative of overfitting, when we look at the other datasets in comparison, we see that we still achieve high accuracy in a tuned SVM.

SVM Preliminary Conclusions

The accuracy of the SVM model remains low when using a sampling from all the data provided—if we are to remember business purposes, we must remember that this data is to be exported in a dashboard to be presented to a client. This client must make long-term business decisions with potentially detrimental consequences based on the data presented. As such, a 22% chance of the data being inaccurate, or the data inaccurately predicting a fault where there is none may result in undue spending and financial stress for the business.

From a model and data science point of view, based on the data/model split, we can claim that rather than fitting to detect faults in a binary manner, our model here optimizes for detecting the *fault-free* scenario for both the SFPU and PFPU dataset. We can conclude that based on the relative discrepancies in faulty and fault free data, if we score and rate data on a binary scale (faulty in opposition to fault-free) we can see that we tend to select and predict for fault-free data, and as we only have one sample we tend to bias towards a faulty view.

Random Forest

Random Forest is a powerful classification tool renowned for its effectiveness in handling a wide range of classification tasks. One of its notable strengths lies in its ability to handle complex datasets without the need for extensive preprocessing. Unlike many other machine learning algorithms, Random Forest can effectively handle missing values, outliers, and irrelevant features, reducing the need for extensive data preprocessing steps such as imputation and feature scaling. Additionally, Random Forest's ensemble nature, comprising multiple decision trees, enables it to capture intricate relationships within the data, leading to robust and accurate predictions. Moreover, its parallelized implementation allows for efficient computation, making it well-suited for large datasets and real-time applications. With its robustness, versatility, and fast computational time, Random Forest stands out as a reliable and efficient classification tool suitable for various domains and applications. Using the random forest paradigm, we can investigate and tune our data and understand which features and parameters within the dataset to see if there are certain cohorts of features which contribute the most to faults.

Dataset IV: Randomly sampling across all .csv files:

Due to the sheer size of the dataset, we had to trim it down to 100,000 observations. This downsizing was necessary to handle the data efficiently without compromising its integrity. Despite the reduction, we made sure to maintain the same balance of important categories as in the original dataset. By doing this, we ensured that our analysis and models remain reliable and applicable to our business needs, even with the scaled-down data.

To maximize the efficacy and specificity of our analysis, the Random Forest algorithm was individually applied to both the PFPU and SFPU datasets. This approach was chosen to account for potential differences in the operational characteristics and performance metrics between the two types of units. By analyzing each data set separately, we could uncover unique patterns, trends, and insights specific to PFPU and SFPU units, thereby facilitating more targeted and nuanced decision-making processes. Additionally, conducting separate analyses allows for a deeper understanding of the distinct factors influencing the performance and behavior of each unit type, ultimately leading to more accurate and tailored recommendations for optimizing their operation and efficiency. Overall, this segmented analysis strategy ensures that our findings are finely tuned to the intricacies of each unit type, enhancing the overall effectiveness and relevance of our insights in addressing specific operational challenges and objectives.

The Random Forest model underwent rigorous five-fold cross-validation, due to limitations posed by computational resources. However, if computational constraints allow, it's advisable for businesses to conduct additional cross-validation iterations to ensure robust model evaluation. By increasing the number of cross-validation folds, businesses can obtain more reliable estimates of the model's performance and better assess its generalization capability.

Our analysis includes a feature important plot, which succinctly highlights the significance of various features on model predictions. This visualization assists businesses in pinpointing key variables crucial for fault detection. By identifying these critical factors, stakeholders can prioritize resources effectively, streamline decision-making, and optimize operational processes. Additionally, the plot enhances transparency and trust in the predictive process, facilitating strategic initiatives aimed at improving system reliability and efficiency.

Our analysis includes class accuracy plots, providing a clear visualization of the model's accuracy in predicting each class type. These plots offer valuable insights into the performance of the classification model across different fault categories, enabling stakeholders to assess the model's effectiveness in detecting specific types of faults. The business use case and application of class accuracy plots lie in their ability to identify strengths and weaknesses in fault prediction, guiding targeted improvements in maintenance strategies and operational efficiency. By understanding the model's accuracy for each class type, businesses can prioritize resources, implement proactive maintenance measures, and optimize system reliability to minimize downtime and enhance overall performance.

The distribution of accuracy score plots from cross-validation provides valuable insights into the variability and reliability of the model's performance across different folds or iterations. This information is crucial for understanding the robustness of the model and assessing its generalization capability. In a business context, these plots help stakeholders gauge the stability and consistency of the model's predictions, enabling them to make informed decisions about its deployment and reliability in real-world scenarios. By identifying any potential fluctuations or inconsistencies in performance, businesses can refine their strategies for model deployment, optimize resource allocation, and ensure the effectiveness of predictive analytics initiatives.

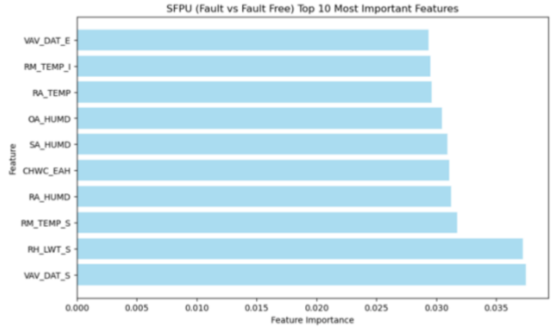
Fault vs Fault Free

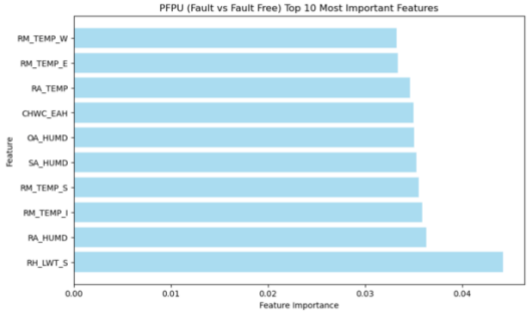
The Random Forest algorithm was employed to classify instances as either "fault" or "fault-free" within the dataset.

The feature importance plot below showcases that the most influential variables for distinguishing between fault and fault-free conditions were VAV discharge air temperature, reheating coil leaving water temperature, and room temperature for SFPU while reheating coil leaving water temperature, return air humidity, and room temperature were important for PFPU.

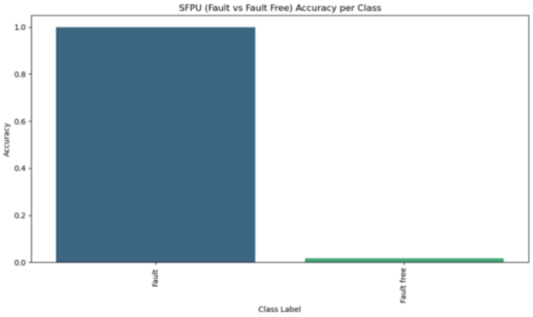
For SFPU units, VAV discharge air temperature is critical as it directly influences the air distribution within the space being conditioned. Any deviations from the expected discharge air temperature could indicate issues with airflow or temperature regulation, potentially signaling a fault. Similarly, reheating coil leaving water temperature is essential for SFPU units as it affects the efficiency of the heating process, particularly in maintaining comfort conditions during colder periods. Room temperature serves as a key indicator of occupant comfort and system performance, making it vital for fault detection in SFPU units.

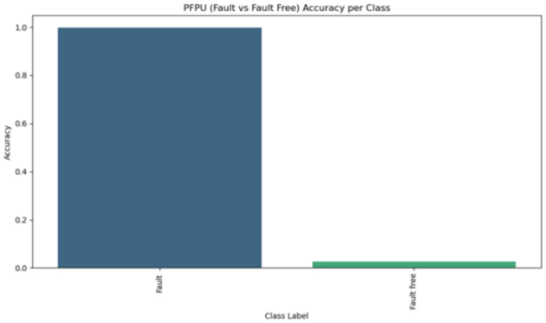
In the case of PFPU units, reheating coil leaving water temperature is crucial for ensuring proper heating capacity and comfort control within the space. Return air humidity is important as it reflects the moisture levels in the conditioned space, which can impact occupant comfort and indoor air quality. Room temperature remains significant for PFPU units as well, serving as a primary parameter for maintaining desired thermal conditions.



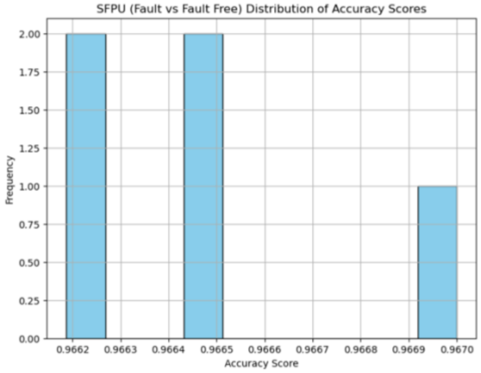


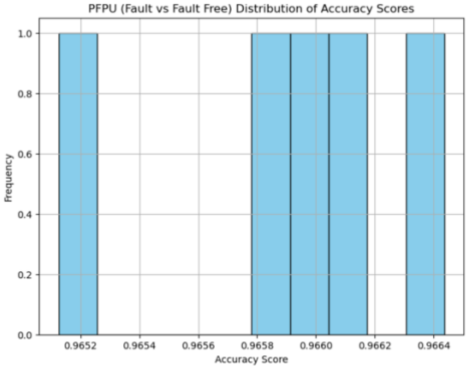
The class accuracy plots showcase the Random Forest algorithm is highly proficient at predicting faults, prioritizing their detection over fault-free instances. This isn't necessarily a negative as it's more important to detect when things go bad than when everything is good for this business use case.





Following five-fold cross-validation, the Random Forest model demonstrated a mean accuracy score of 96.64% for SFPU and 96.59% for PFPU. Analysis of the distribution of accuracy score plots indicate the model's exceptional accuracy in fault detection, showcasing reliability and minimal variance. However, it's bias towards classification towards Faults remains a liability in a business use case as the business cannot efficiently track where to send resources to fix the Faults.

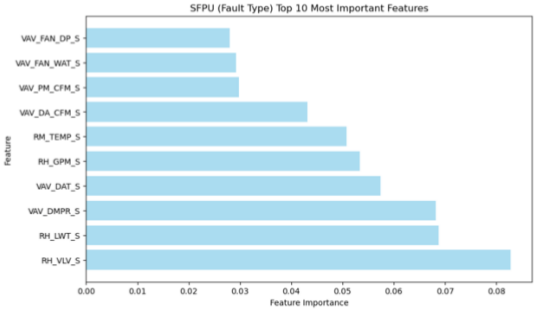


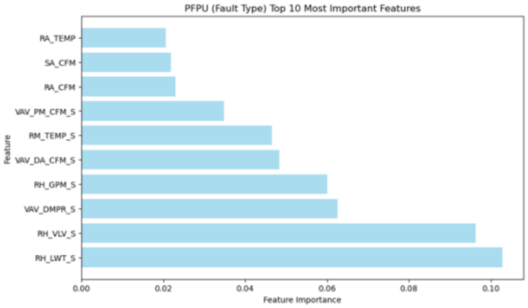


Fault Type

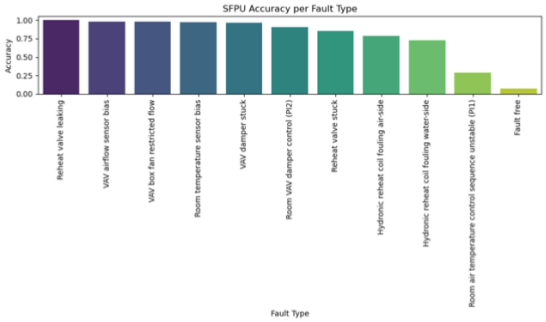
The Random Forest model was run to categorize the different fault types to provide a granular analysis, enabling detailed examination of fault characteristics.

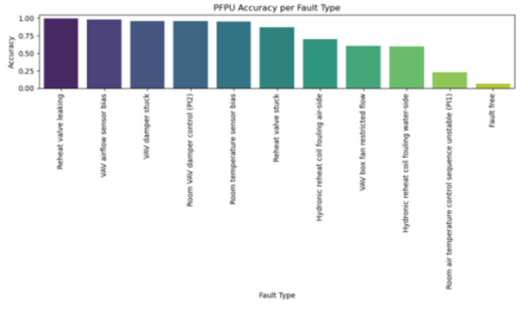
The most important features for detecting fault types in SFPU and PFPU units were identified as reheating coil valve position, reheating coil leaving water temperature, and VAV damper position. Reheating coil valve position and Reheating coil leaving water temperature impact the efficiency of heating processes, crucial for maintaining desired comfort levels within conditioned spaces. VAV damper position is instrumental in regulating airflow, ensuring optimal distribution of conditioned air. By prioritizing monitoring and management of these key parameters, businesses can enhance fault detection capabilities and optimize system performance across SFPU and PFPU units, ultimately improving operational efficiency and occupant comfort.



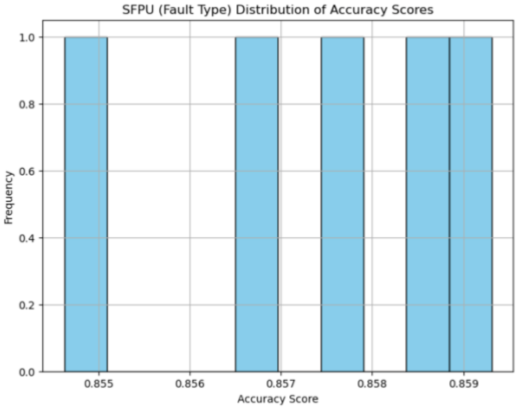


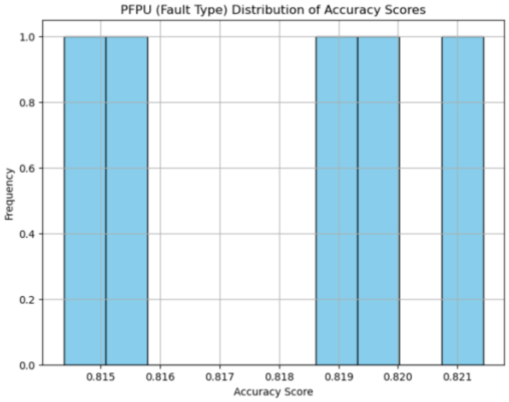
Both SFPU and PFPU models exhibit proficiency in detecting several common faults, including reheat valve leaking, VAV airflow sensor bias, room temperature sensor bias, and VAV damper stuck. The class accuracy chart plays a pivotal role for Joulea, providing insights to guide leaders in resource allocation based on the model's detection capabilities. Additionally, it highlights areas where the model may fall short in fault detection, enabling targeted efforts for further learning and improvement. One note to make is that the model detects zone S as a significant factor, showcasing a need to separate the model into different zones.





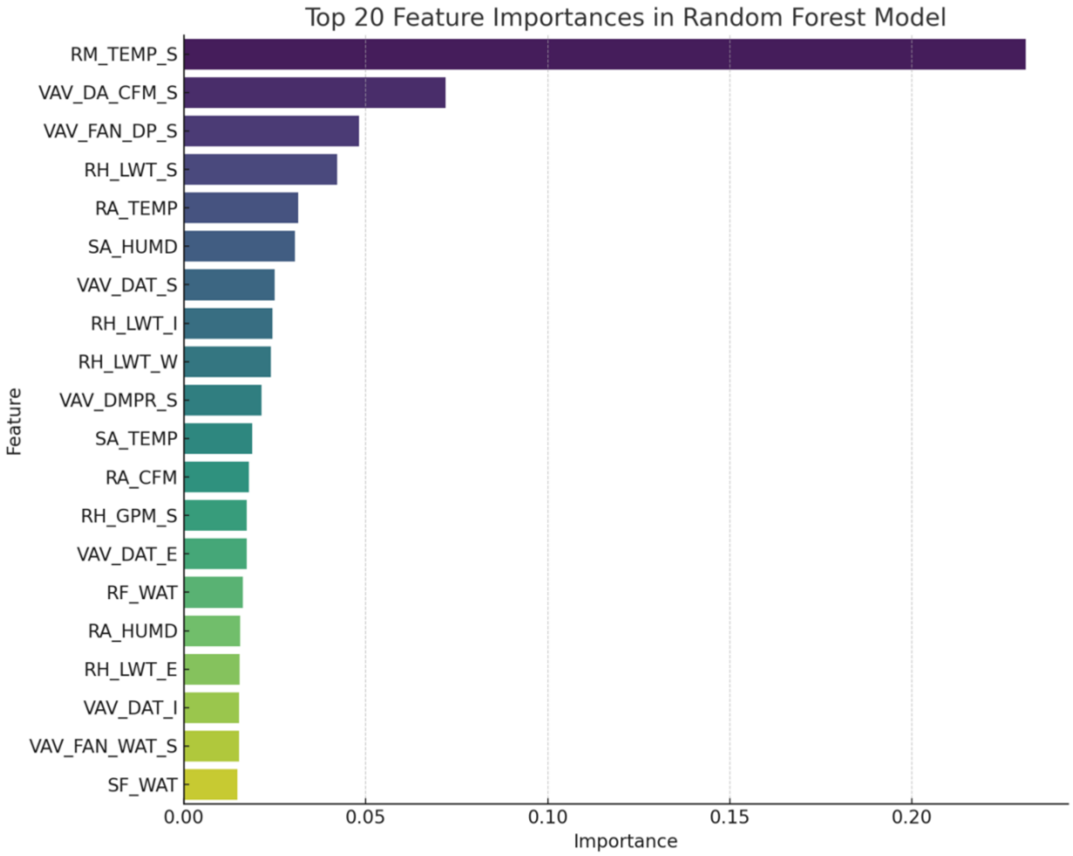
Following five-fold cross-validation, the Random Forest model demonstrated a mean accuracy score of 85.75% for SFPU and 81.80% for PFPU. Analysis of the distribution of accuracy score plots indicate the model's exceptional accuracy in fault detection, showcasing reliability and minimal variance. In its current form, the random forest can accurately detect certain fault types but not all and has a hard time detecting when something is fault free. Thus, it is still not the most reliable model.





Dataset III:

Binary classification using PFPU data:

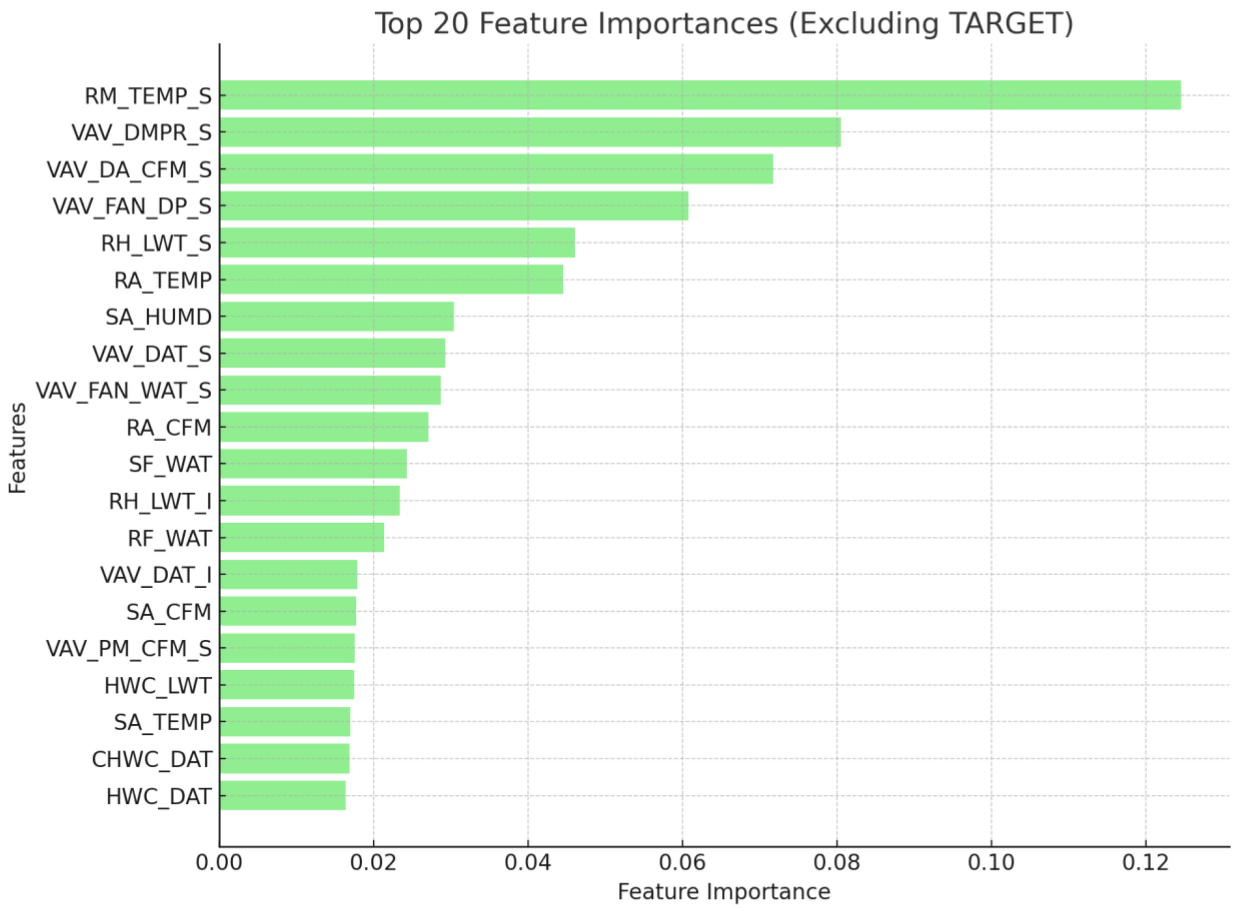


Here, we aimed to understand the discrepancy between binary and multi-class classification in a time-agnostic world. As we can see, a smaller sampling of fault sample types, including the Room temperature sensor bias at –2C, VAV box fan restricted flow, and VAV damper stuck at 100%. We have provided some summary statistics below:

Metric	Score
Accuracy	98.2%
Precision	98%
Recall	100%
F1 Score	99%

In this case, reducing the space needed to generate predictions has increased accuracy. However, the usefulness of a general-purpose detection is debatable, as knowing there may be a fault may not be sufficient to rectify the underlying issue, as the problem space has yet to be reduced.

Performing this analysis in a multi-class forward way, we see the following feature importances:



We also see an accuracy of 99.52%. We see a precision of 88.63%. We have a high score despite a train/test split indicating that there is a strong correlation between time and fault.

Preliminary Conclusions from Random Forest:

We can see that a smaller dataset causes extreme overfitting in the model case, often trending towards false positives in a binary representation of the data. As such, we cannot fully align behind the random forest methodology

Gradient Boosting Analysis

Gradient Boosting is a technique by which a strong classifier is built from many weak classification models to boost model performance; the weak classifiers are shallow decision trees which are then compounded—typically a gradient boost classifier will observe the residuals in a single shallow tree or learner and subsequently fits decision trees iteratively to minimize the residuals from previous fittings. A gradient boosted classifier—here, a gradient boosted tree offers a highly flexible approach to both classification and regression in a dataset composed of heterogeneous features.

In this approach, due to the volume and size of the data and the relative limitations of commercial hardware, we attempt to randomly sample the data for investigations and provide relative comparative analysis of various datasets.

Dataset I: Prediction with two categories: the first investigation was completed by randomly sampling from the SFPU dataset—both faulty data from the VAV fan restricted flow and fault-free data was presented to the gradient boosting classifier after being manually labelled as faulty or fault free. To enhance the sample's quality and avoid biases and overfitting, the random sample for this set was taken from the same period, and an extra column was added to indicate a faulty or non-faulty unit.

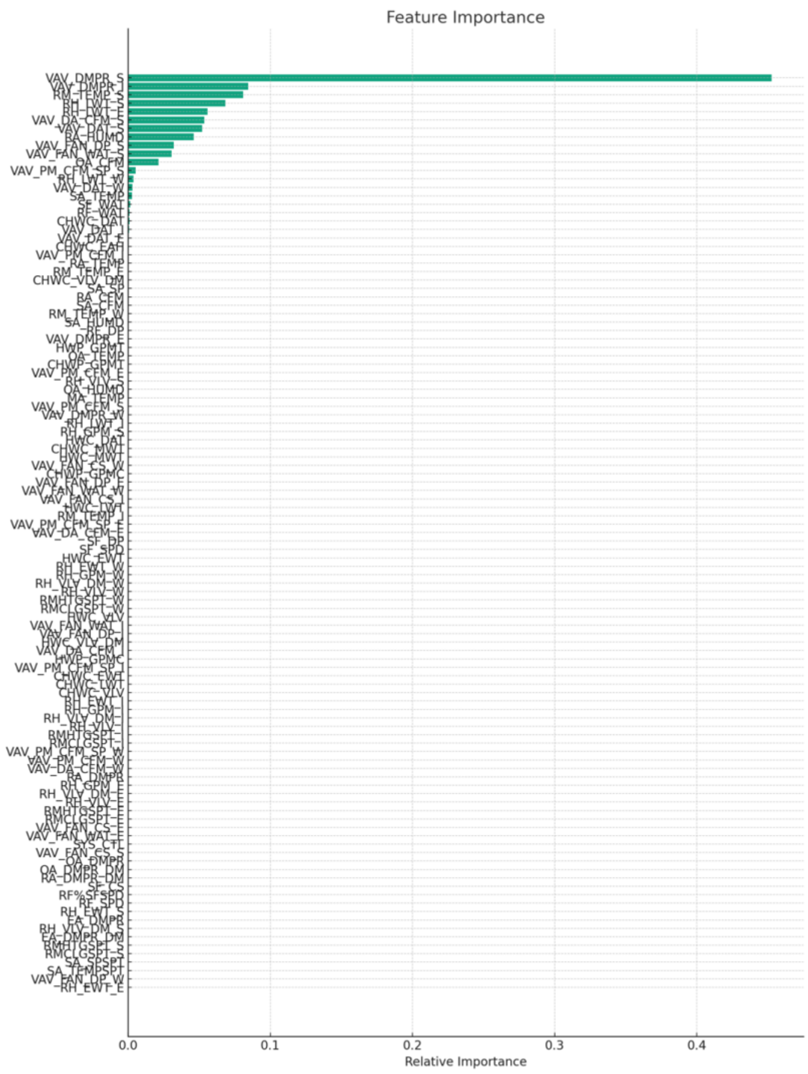
The gradient boost classifier was used to predict the incidence of a faulty or non-faulty unit.

We attempted to attain parity with other tests: our primary aim in choosing a smaller dataset, as with other tests was to create a more manageable dataset to be able to compute our data locally.

Prediction Results

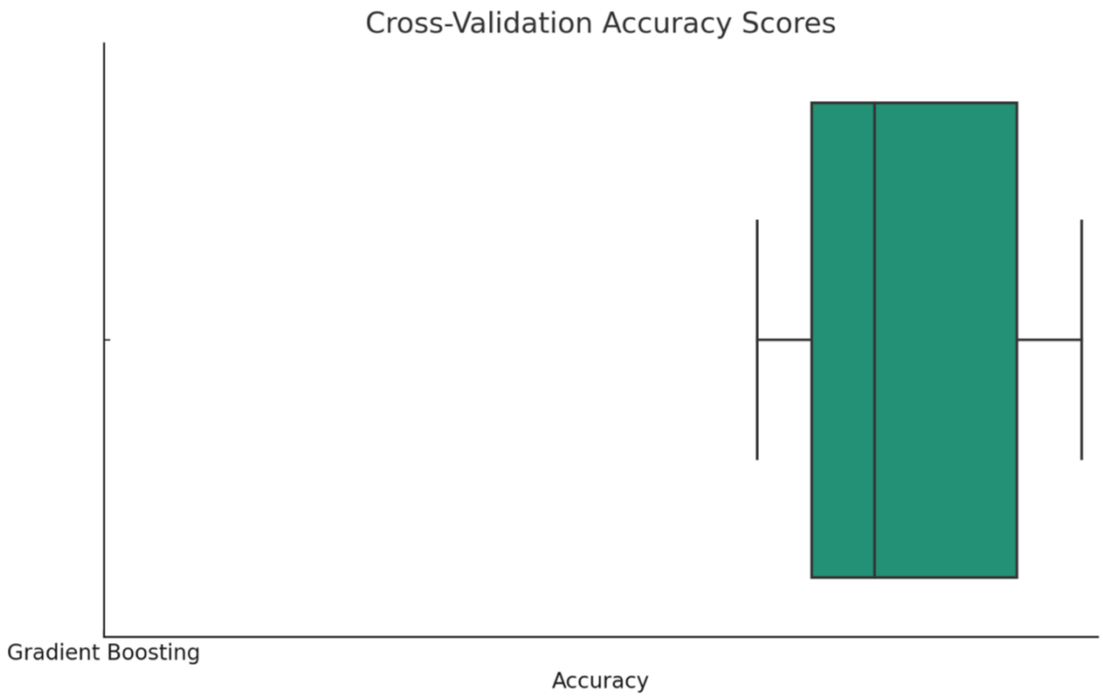
When this prediction was done without cross-validation we observed overfitting, namely, we had a 100% accuracy observed. With cross-validation, we were able to mitigate the overfitting observed and saw an average accuracy of around 79% across folds.

Feature Importance



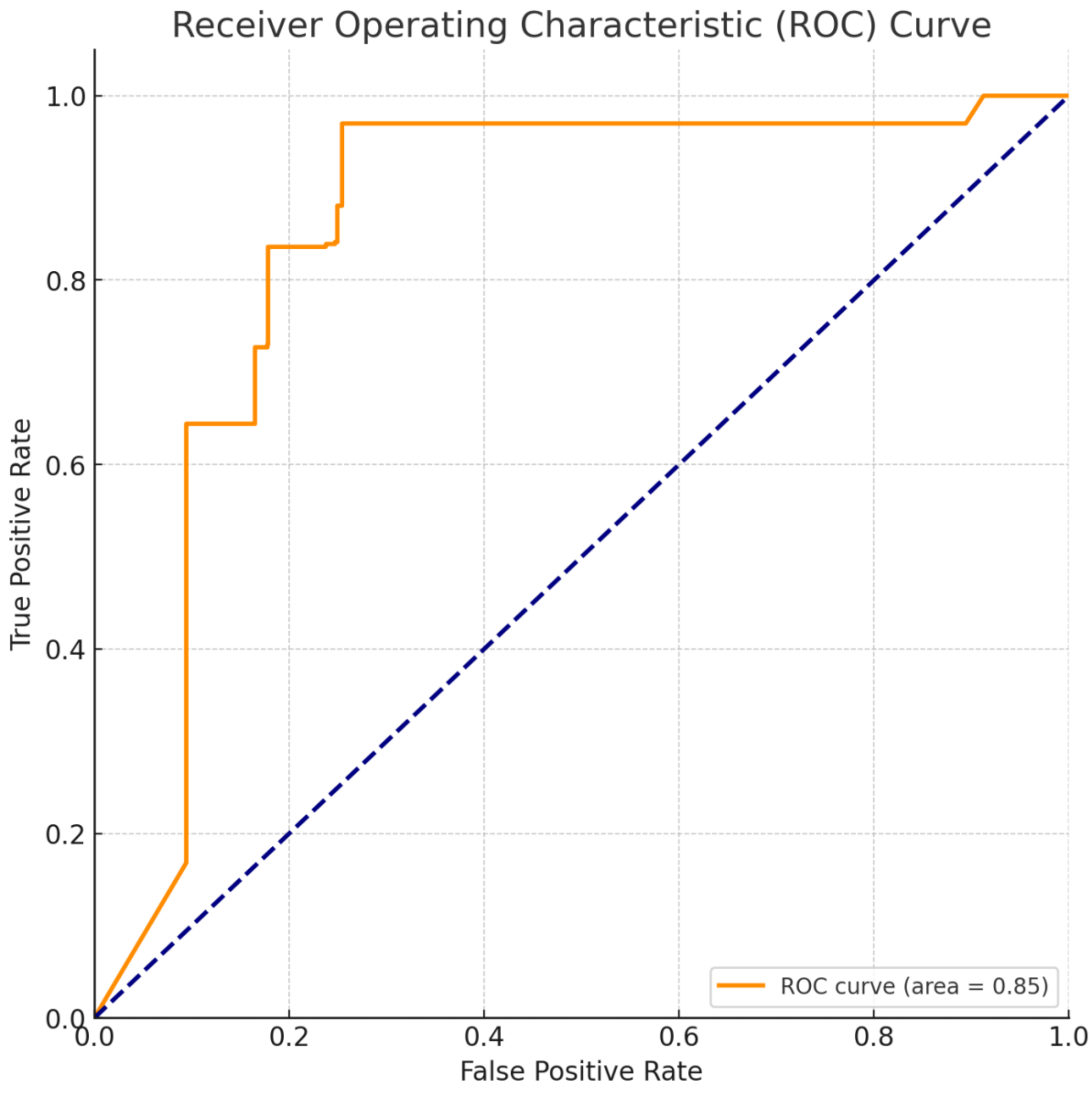
As we can see, the VAV fan-restricted unit differs from the fault-free unit in the damper position. Extrapolating to a more general set of data, we can assume that most faults are caused by the tuning and adjustment of a few unit variables.

Cross Validation


Cross-Validation Accuracy Scores
Gradient Boosting
Accuracy

ROC

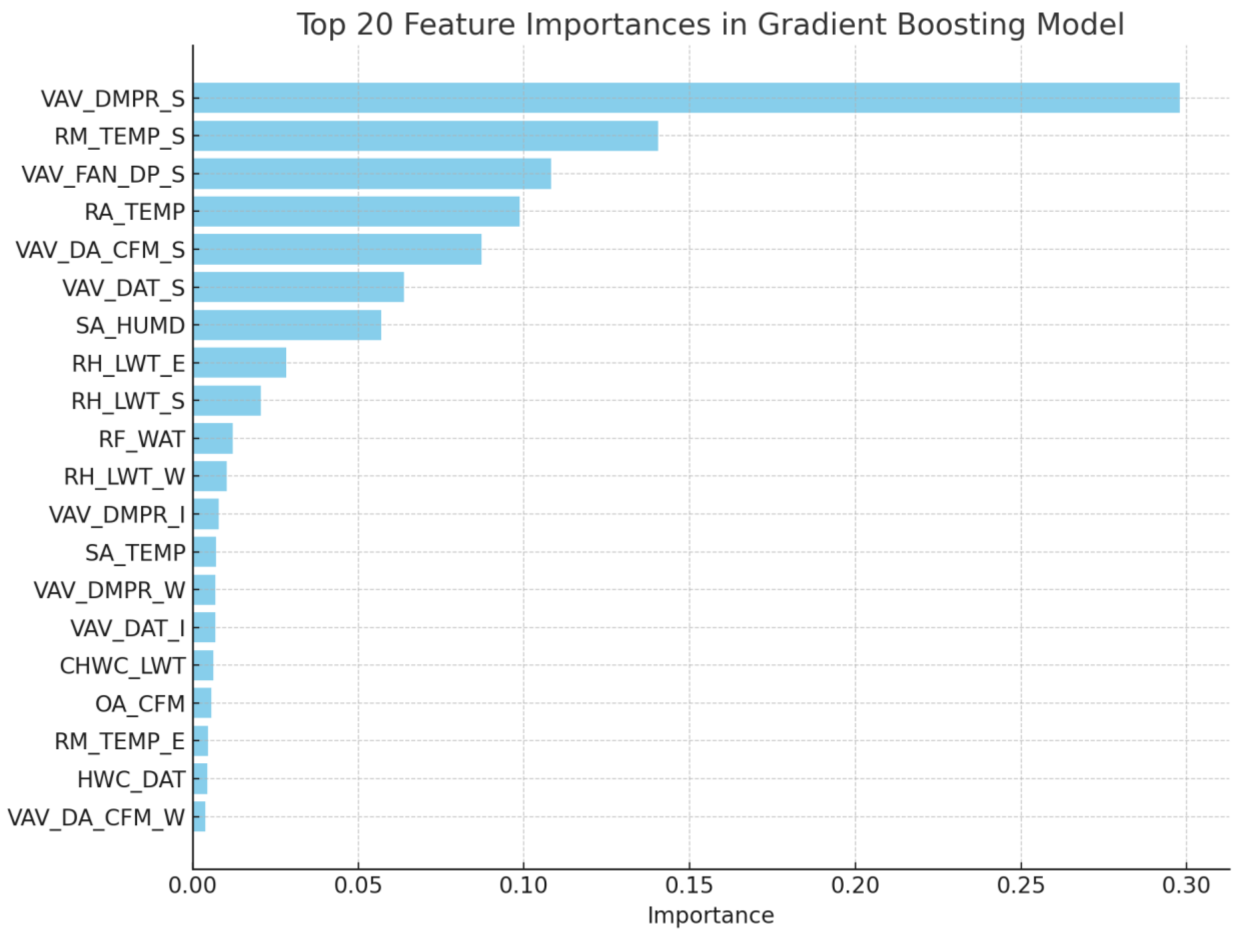
Curve



The ROC curve indicates a good ability to distinguish between the two classes presented.

It is important to note that for the gradient boosted analysis, classification was done independently from the timestamp, meaning that while the time was used as a feature, the data was not presented as sequential or ordered in any way.

Dataset III:



As with other iterations of this dataset, we performed a cross-validated gradient boost, and found that the feature importances varied across models. We have also provided a summary of the metrics here.

Metric	Score
Accuracy	99.4%
Precision	97.1%
Recall	100%
F1 Score	99.6%

The high outputs for each success metric indicate that this dataset is in some way overfit.

If we are to repeat this data analysis in a binary fashion, we see that even given a train/test split, we result in the following metrics, meaning that we continue to see the overfitting.

Metric	Score
Accuracy	100%
Precision	100%
Recall	100%

F1 Score	100%

Dataset IV:

The sample of 100k data points in a multi-class classifier resulted in the following metrics.

Metric	Score
Accuracy	87.9%
Precision	91.0%
Recall	94.3%
F1 Score	89.8%

Conclusions:

As we can see, the gradient boosted method is highly prone to overfitting, and the time and space-based correlation will potentially cause conflation within zone and within date. A gradient-boosted decision tree will attempt to minimize entropy at each step, and therefore use correlational data between zone and between time in each subset of data to create predictions which may not be fully accurate and instead focusing on micropatterns or local minima.

Category II: Time-Series-Based Models:

In our second worldview, we assume that time does play a significant role in the generation of faults—in other words, as time increases and goes on, the divergence between a faulty unit and a fault-free unit also increases. This apparent difference can be seen as a correlation between data, resulting in overfitting in gradient-boosting methods in our earlier assessment. By correcting this assumption through modeling for time, we hope to gain a model with an increase in predictive accuracy.

Logistic Regression for Time-Series:

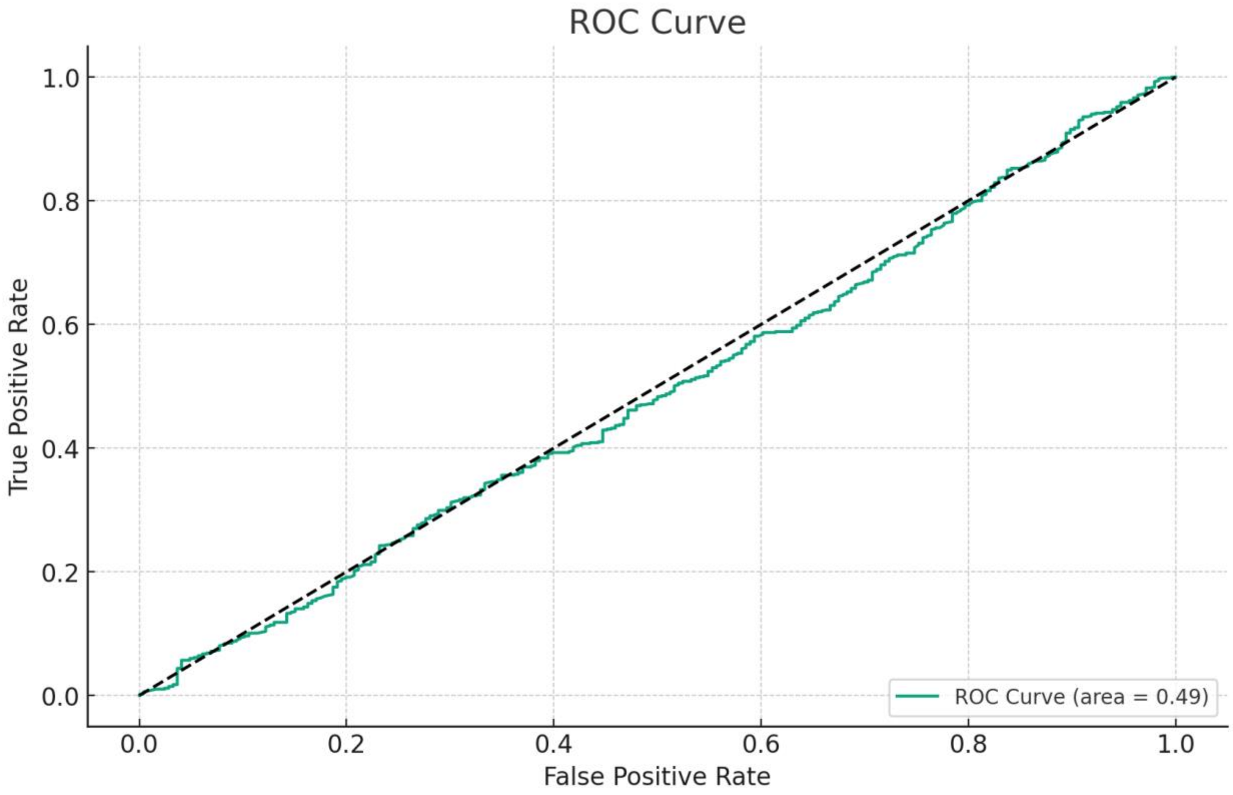
Logistic regression itself is a binary classification model in which instances are classified into a “faulty” or “fault-free” version. Here we assume the underlying probability distribution resembles something like the binomal distribution without a heavy class imbalance.

A logistic regression model for time-series data is fundamentally a logistic regression model with an allowance for correlated data. A logistic regression model assumes a linear but discrete relationship between input and output, or in this case, factors of influence and the presence of faulty or fault-free data. A traditional logistic regression assumes that the data is independently drawn, but a time-series based logistic regression allows for the correlation of data through the presence of legged terms for autocorrelated variables, wherein autocorrelation describes and refers to a periodic signal that can be observed within a certain interval.

In this paradigm, we have decided on logistic regression over a multi-class classification method because of processing limits on our local infrastructure. We will provide a more comprehensive holistic view on architecture in our future work section.

It is important to note that because of the additional autocorrelation features, we were unable to process all 100k responses from the full dataset sample (IV) and have instead shown an example analysis on the subsets of data used earlier for additional analytical depth.

Dataset III PFPU Multi-Fault Dataset:

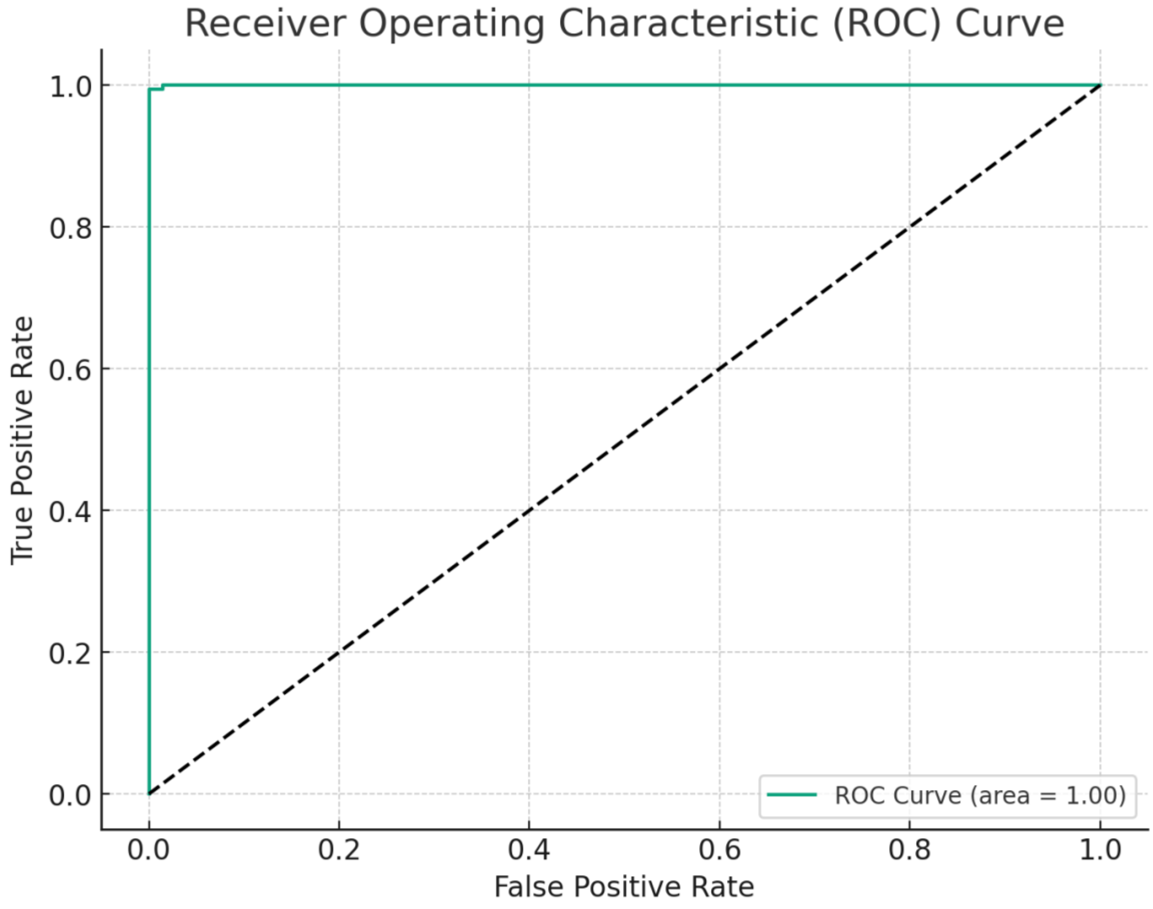


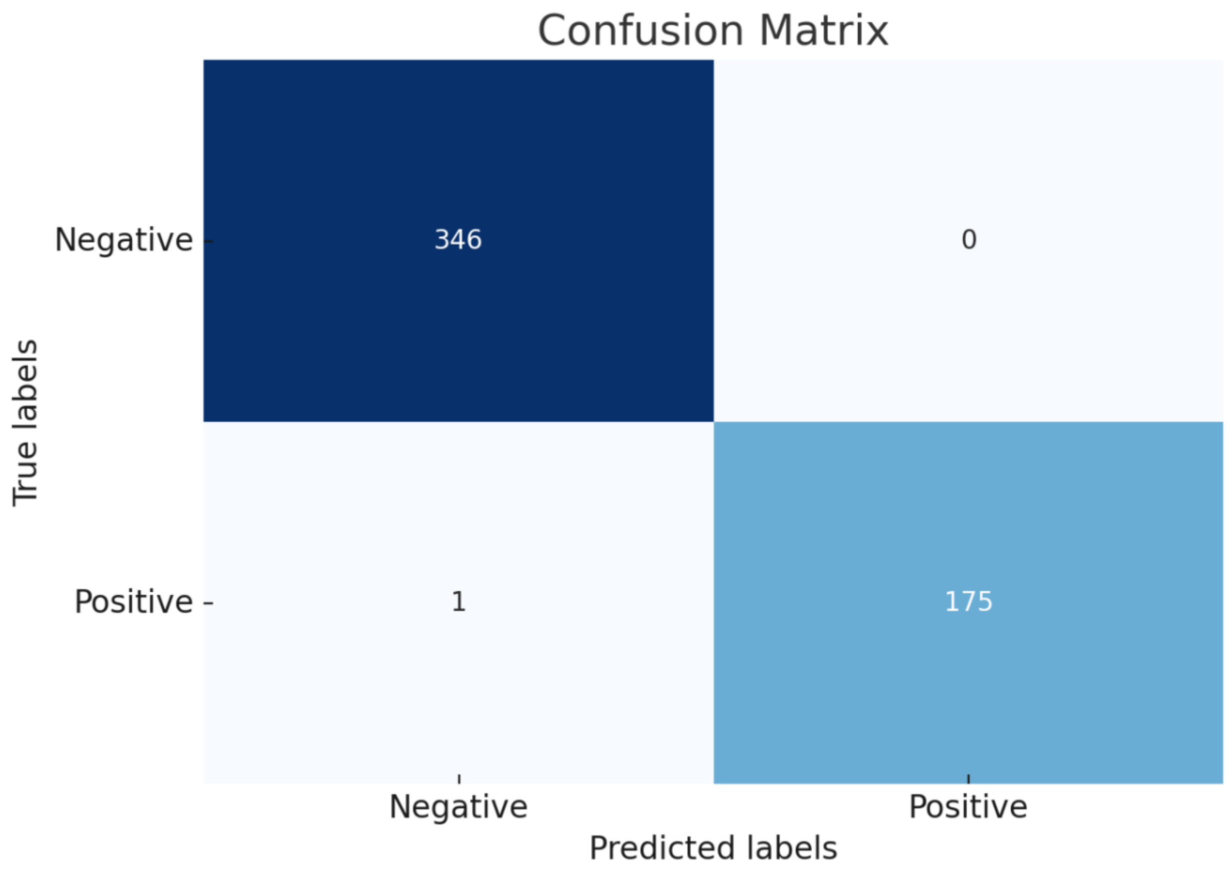
Metric	Score
Accuracy	73.6%
Precision	73.6%

Recall	100%
F1 Score	84.78%

Given the data and summary statistics above, we see that we have some degree of accuracy and precision within our system, and have maintained the balance of false positives and false negatives through the F1 score, however our AUC is close to 0.5, which means that some dimensions of our prediction are no better than random chance—we can explain this by realizing that when decomposed to a two-outcome binary system, we have a fairly imbalanced dataset; we have more data in a faulty scenario than we do in a fault-free scenario and therefore are likely to overpredict the faulted case given a logistic regression model.

Dataset I SFPU binary single-fault dataset:





Given the metrics and confusion matrix, while we can claim the victory of an improved performance when compared to a larger dataset, we see that even given the decoupling of the time series term with a lagged autocorrelator, we still have a high degree of correlation within the dataset, which is causing overfitting.

Preliminary Conclusions:

Ultimately while we were able to take a step in the right direction by reducing correlation among the predictor data using a time-series analysis we did not reduce the full scope of correlation and were not able to preserve the integrity of the per-class outcomes.

Spatial Models:

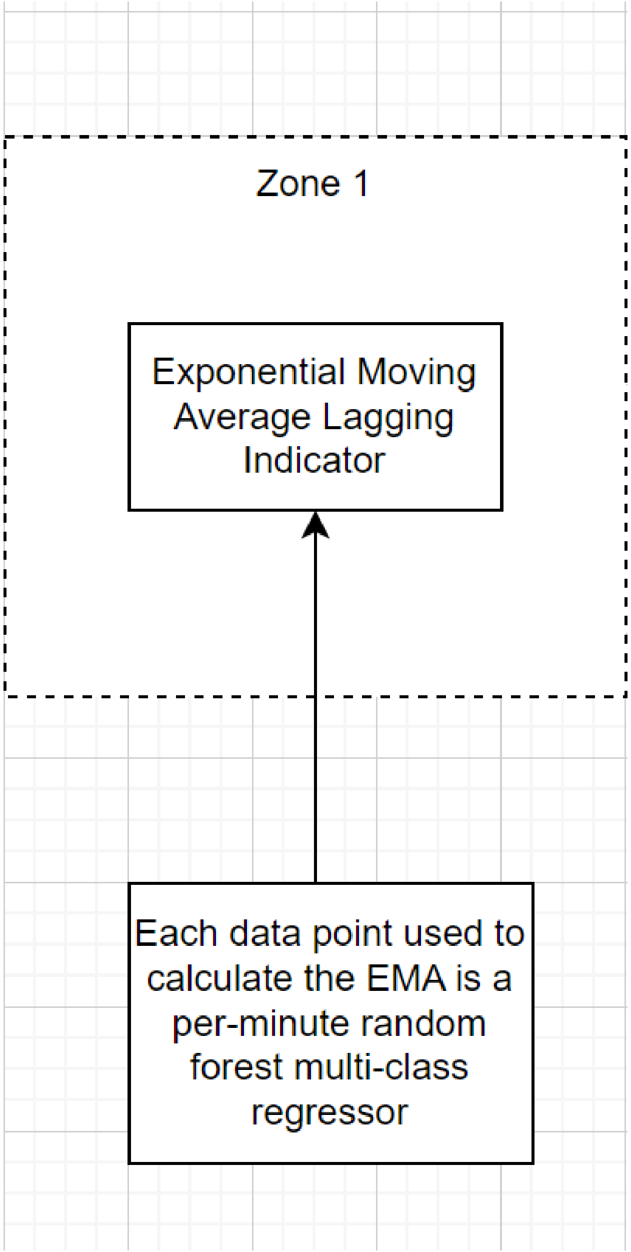
When attempting the analysis, we saw that models were highly correlated per-zone, indicating the need for a temporal assessment. Unfortunately, commercially available software has little performant models for spatially correlated data. Industry standard for such models is through GIS software.

Final Model Suggestions:

As previously mentioned, a single-purpose model ignores the conflation between date and zone, and as such a spatio-temporal model would incorporate both date and zone data as part of a stratified sample, and train models specific to the date and zone.

The ideal state of this architecture involves each zone

A proposed architecture is shown below:

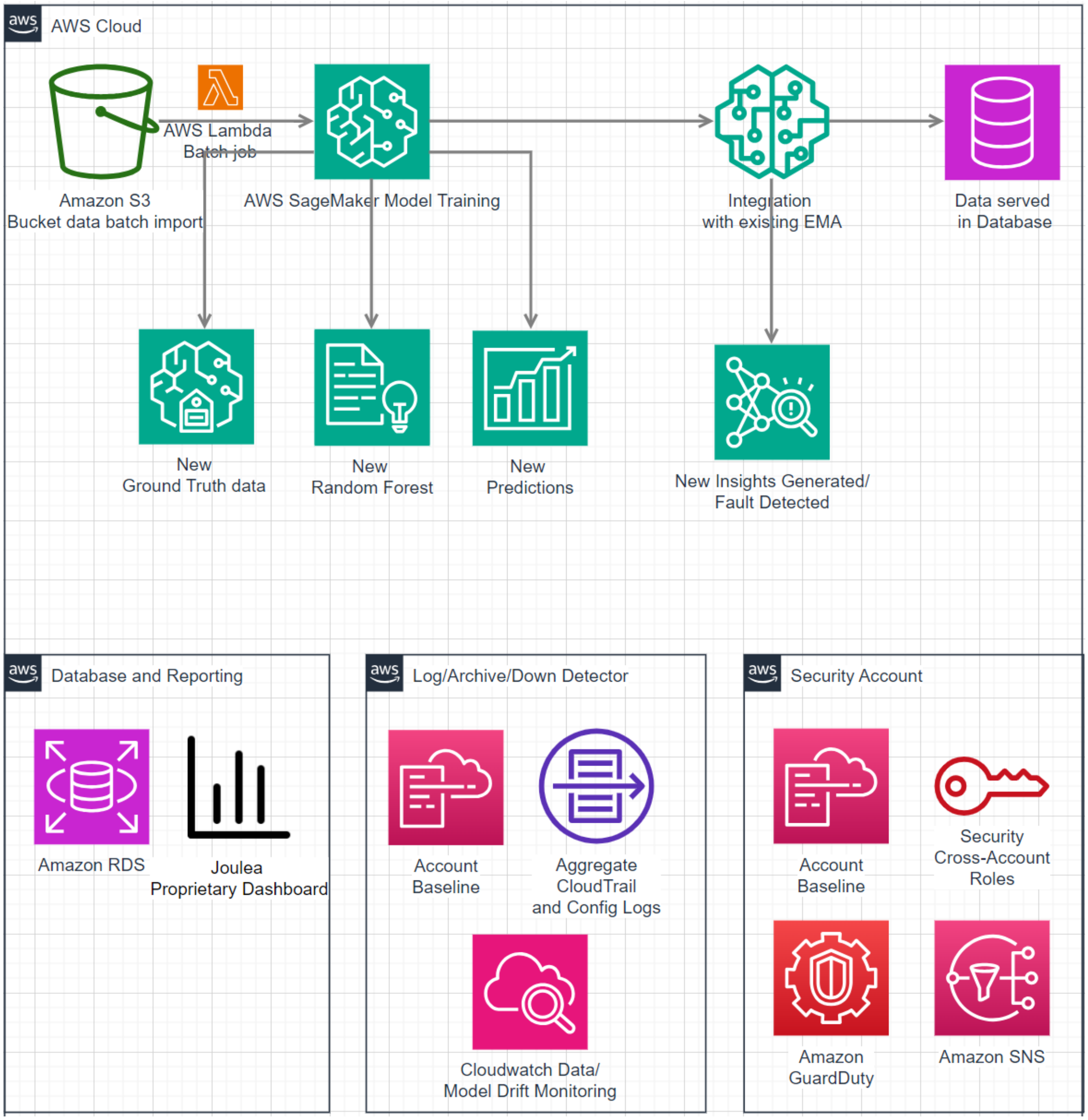

Zone 1
Exponential Moving Average Lagging Indicator
Each data point used to calculate the EMA is a per-minute random forest multi-class regressor

Each model is trained per zone per minute and final metrics are displayed per zone and updated in a batch.

The output of each daily model would be a class similarity score calculated by the random forest model, averaged across a period with the EMA.

On Scale:

As previously stated in our report, we see that we have a significant delta between computational capabilities and overall requirements for a fully robust project. As such, here we propose a scalable cloud architecture that will allow Joulea to break dependency on commercially available hardware and instead be able to massively scale and provide value for clients who may have more than four zones and a reporting cadence that is more granular than the per-minute scale.



Here, we assume that all FPU (Fan Powered Unit) data is stored in an S3 bucket or is otherwise loaded with an ETL. We have suggested an AWS Lambda batch job to reduce costs overall as streaming data would require per-minute or per-second calculations which would incur additional costs for no conferred benefit; that is a streaming solution is expensive, and a fault although detected cannot be remedied and monitored in a minute.

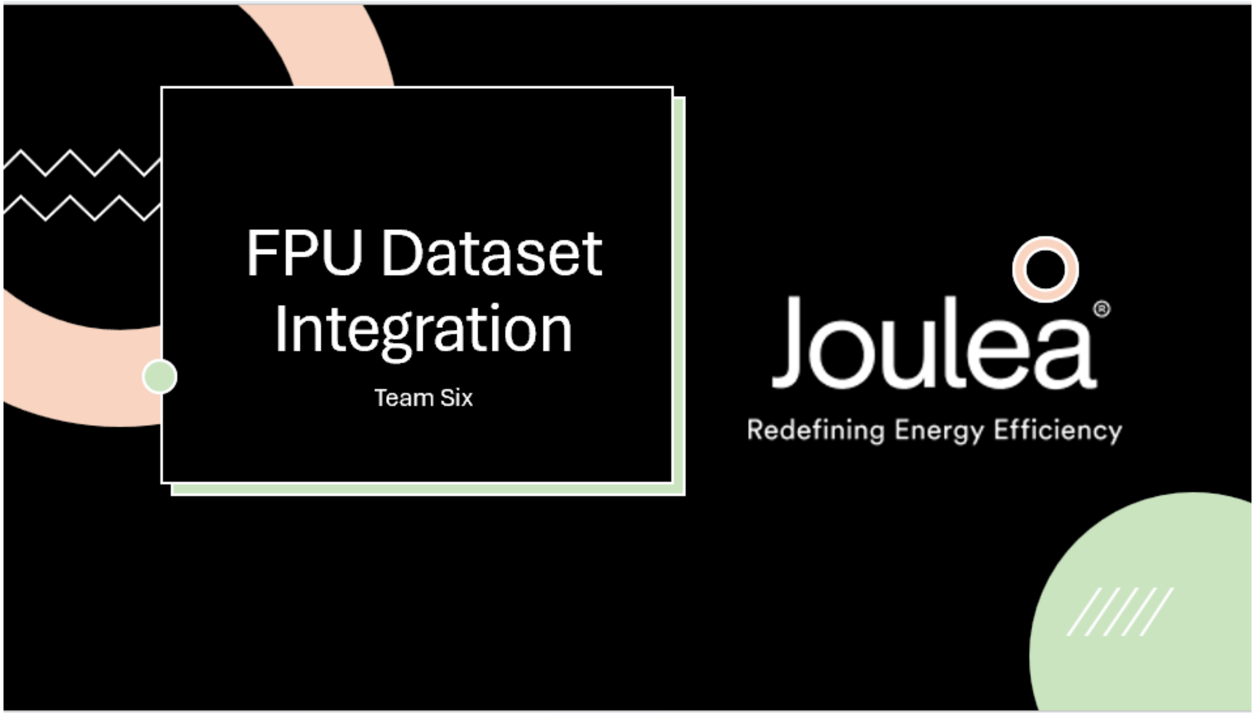
The overall architectural diagram suggests that we use AWS SageMaker services to train a new Random Forest model per day per zone. This per-day, per-zone Random Forest will predict and align a class for our new data and confer it within the context of the day and zone information that we have provided. The model training itself can integrate new ground truth data to allow the use of new predictions as each new day of data can produce a new model version which can provide us with the most up-to-date insights. Each day's model is then rolled into an EMA (exponential moving average) to provide smoothing and capture periodic trends. This EMA also allows the benefit of a periodic predictor that can potentially act as a superior lagging indicator to capitalize on trend data and allow clients to perform unit maintenance when a downward trend in unit health is observed.

It is important to note that while AWS has been depicted as the platform of choice, the underlying architecture is platform-agnostic and can as easily be ported to Azure or GCP.

Business Recommendations & Deliverables for Joulea:

While we have proposed a model for Joulea, this model is without benefits if it cannot be used in a manner benefiting clients. As such, we recommend that this model is used to power insights in an executive level dashboard, as mocked up in a section below.

We have also created a small slide deck to be shared with the executive team regarding full-project implementation, including a Gantt chart and timeline, with key KPIs for each product phase to determine overall business value. We hope this will allow Joulea to understand the true scope of integrating FPU data with the organization.

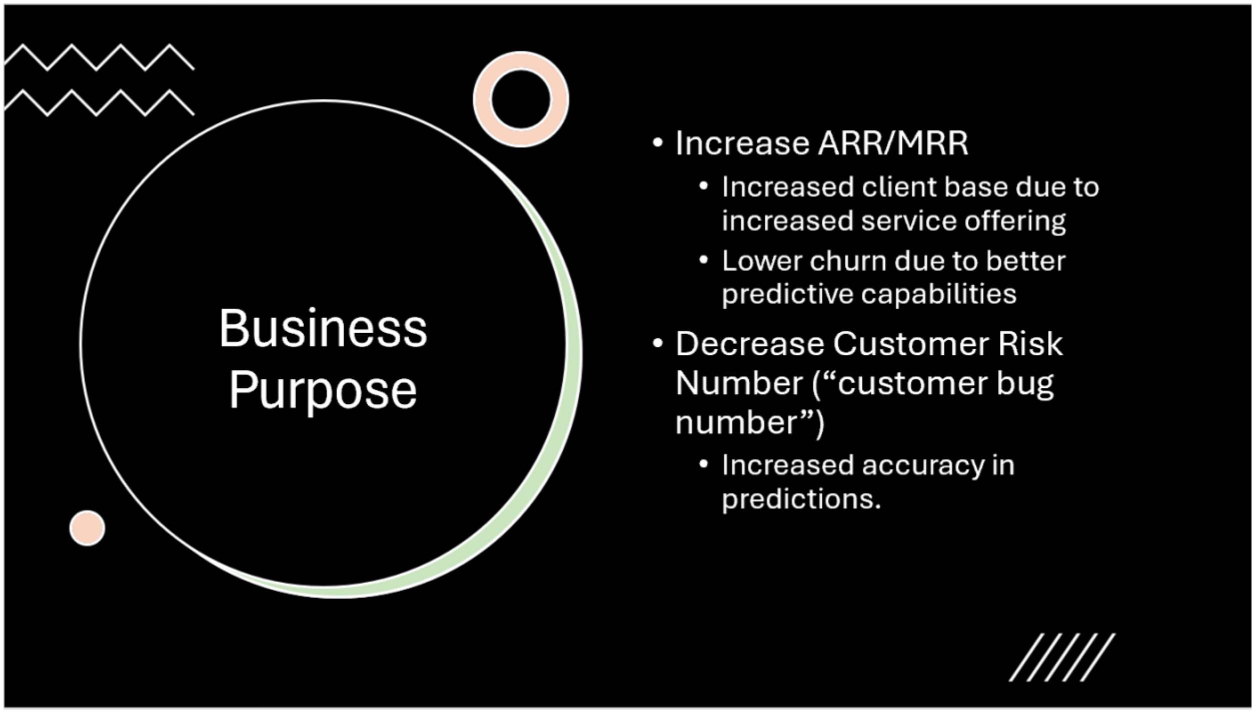


FPU Dataset Integration

Team Six

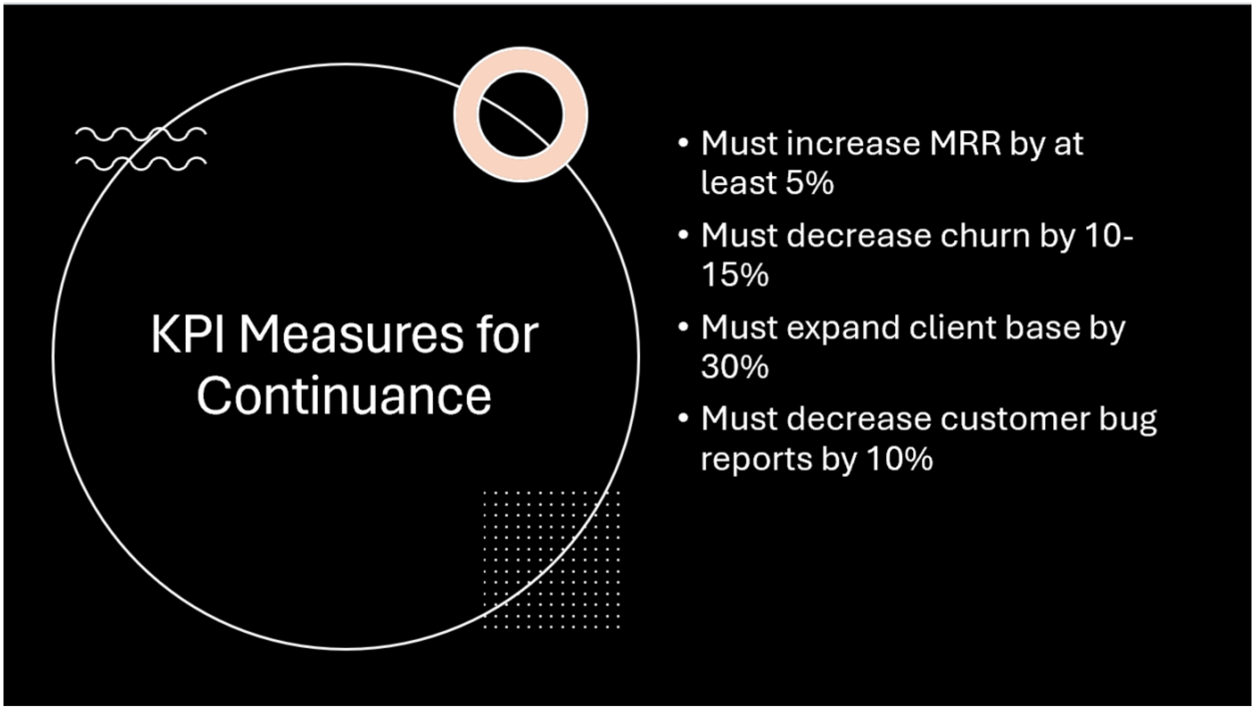
Joulea®

Redefining Energy Efficiency



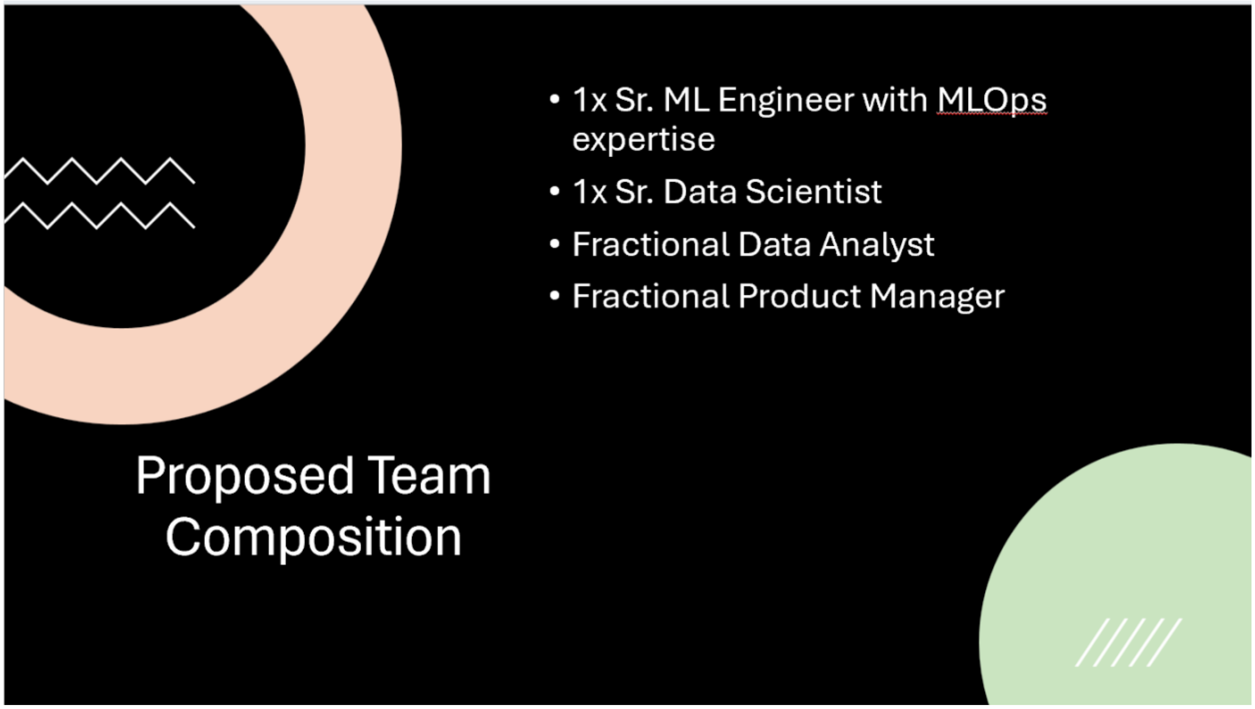
Business Purpose

- Increase ARR/MRR
  - Increased client base due to increased service offering
  - Lower churn due to better predictive capabilities
- Decrease Customer Risk Number (“customer bug number”)
  - Increased accuracy in predictions.



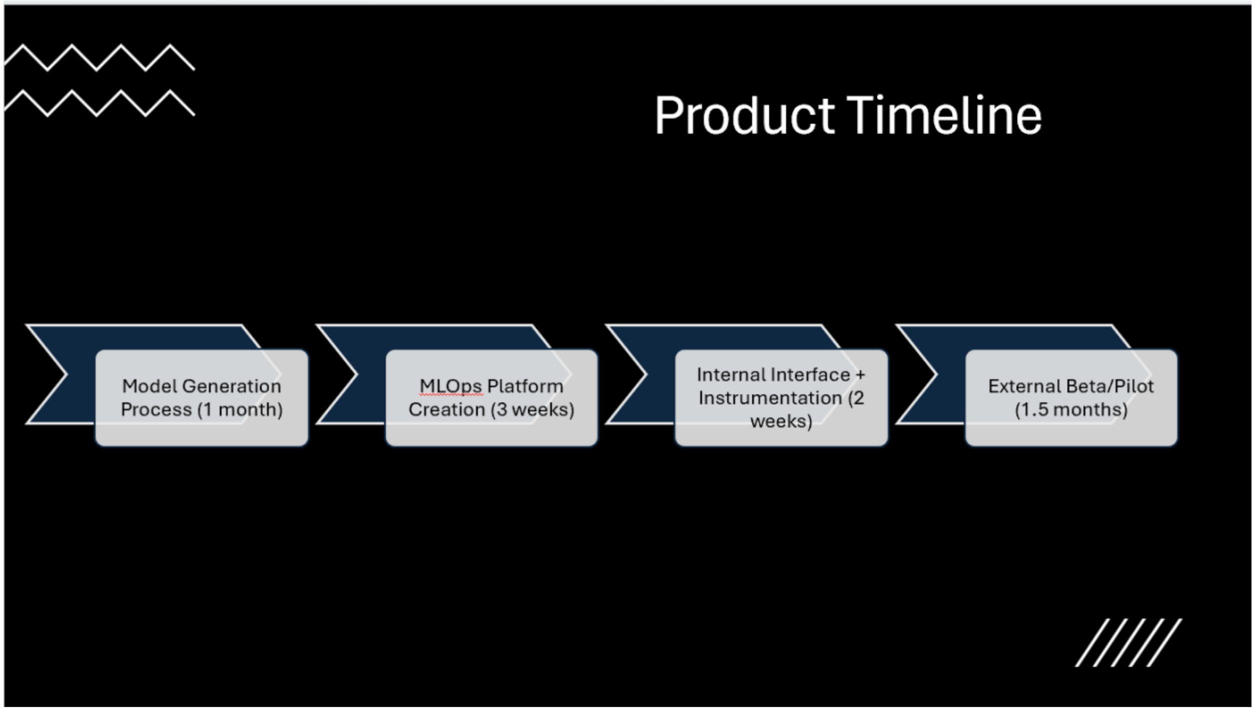
KPI Measures for Continuance

- Must increase MRR by at least 5%
- Must decrease churn by 10-15%
- Must expand client base by 30%
- Must decrease customer bug reports by 10%



Proposed Team Composition

- 1x Sr. ML Engineer with MLOps expertise
- 1x Sr. Data Scientist
- Fractional Data Analyst
- Fractional Product Manager



Product Timeline

Model Generation Process (1 month)

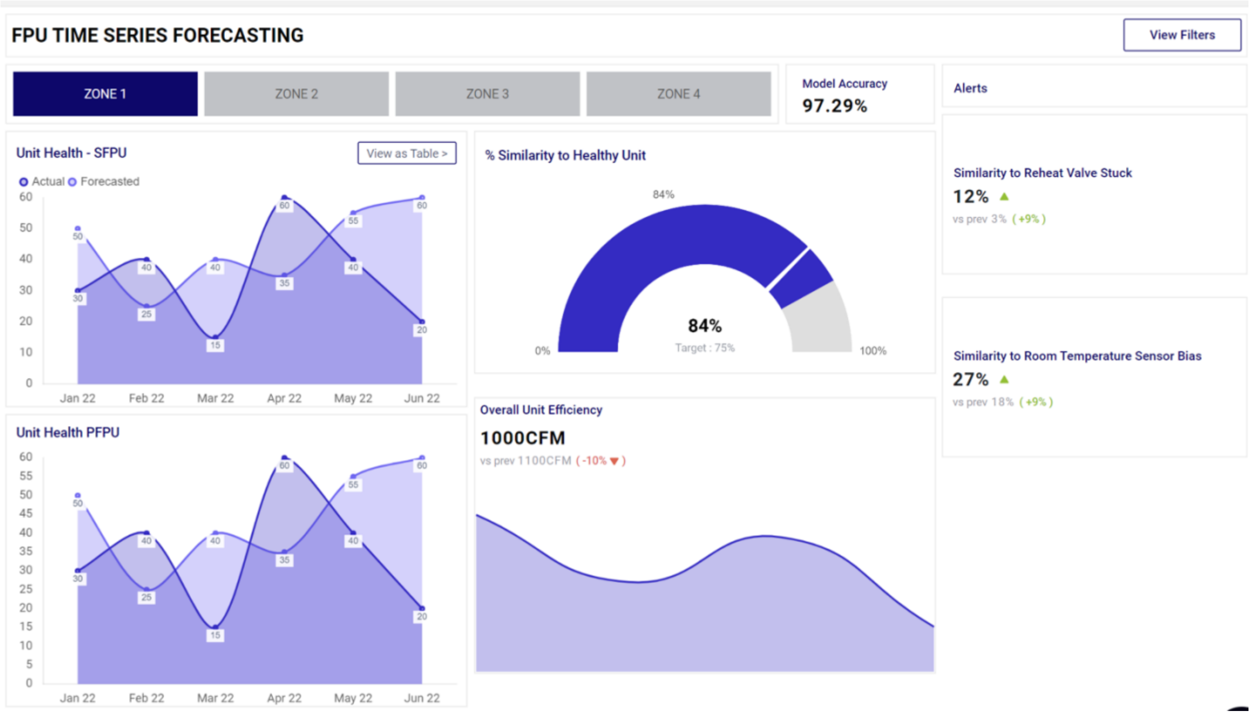
MLOps Platform Creation (3 weeks)

Internal Interface + Instrumentation (2 weeks)

External Beta/Pilot (1.5 months)

Here, we have aimed to provide a high-level business plan showing the requirements to carry on the project, the proposed timeline, and the overall business KPIs for each project.

Mock-Up for Final Dashboard:



FPU TIME SERIES FORECASTING

Model Accuracy 97.29%

Alerts

Similarity to Reheat Valve Stuck 12% ▲ vs prev 3% (+9%)

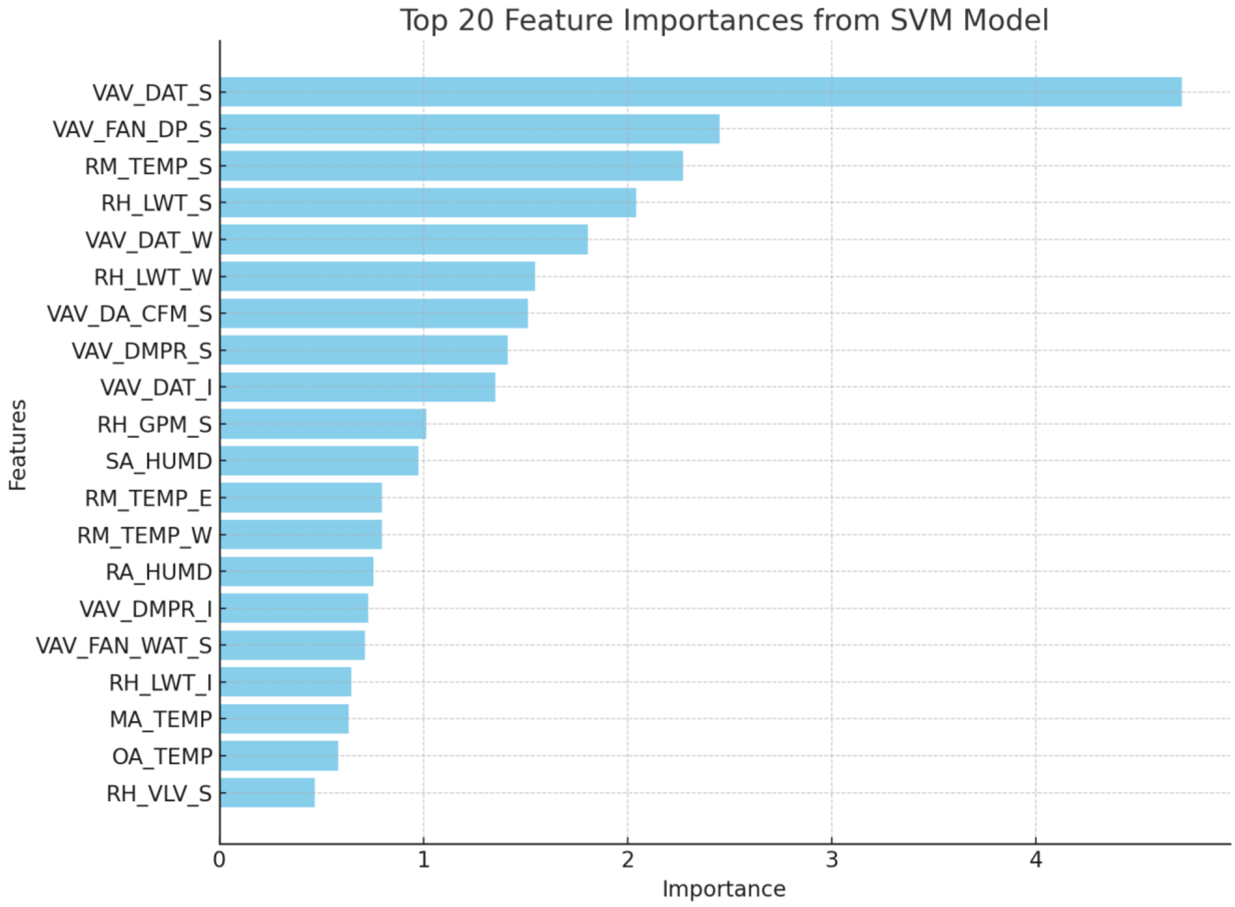
Similarity to Room Temperature Sensor Bias 27% ▲ vs prev 18% (+9%)

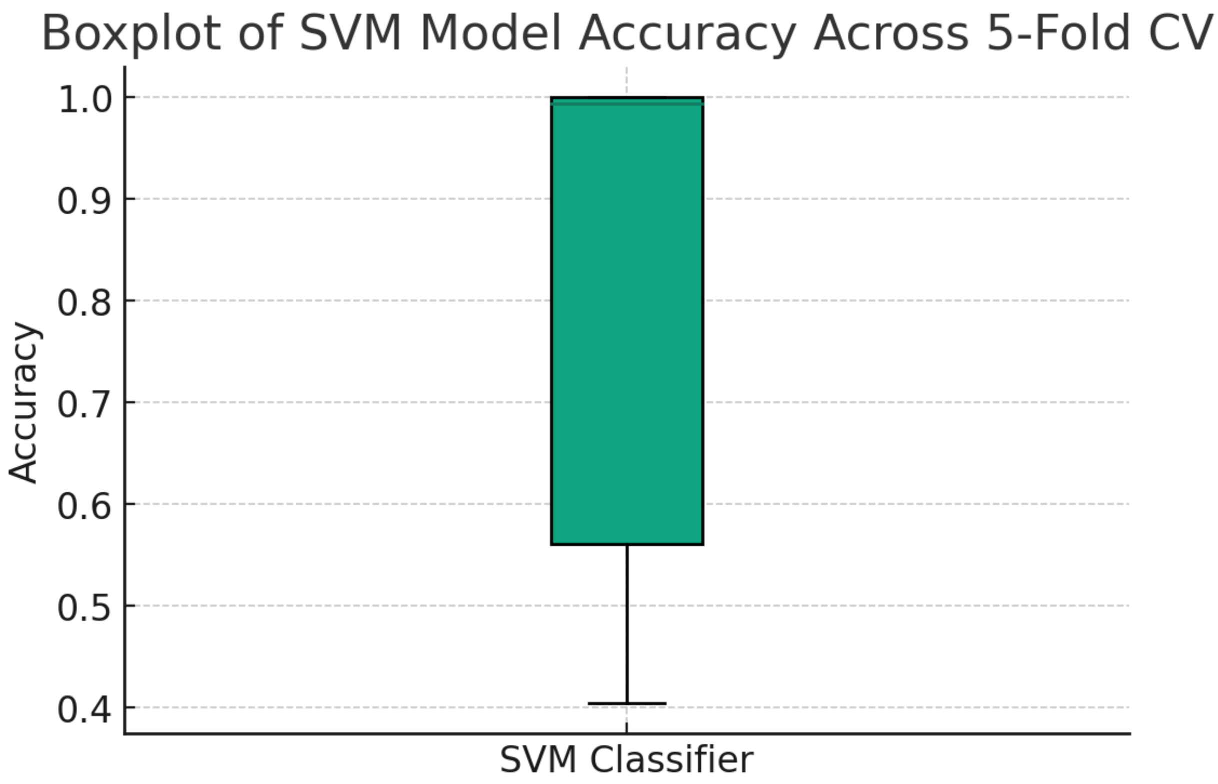
Overall Unit Efficiency 1000CFM vs prev 1100CFM (-10% ▼)

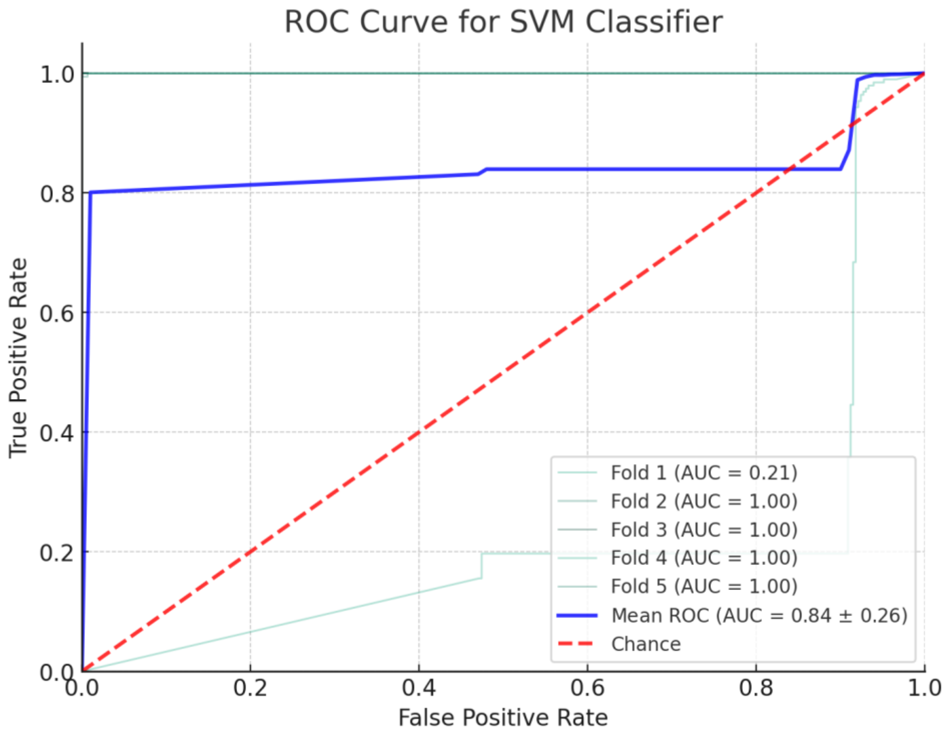
This example dashboard considers a paradigm in which we want to measure the current unit health in relation to a model’s interpretation of a healthy unit.

Appendix

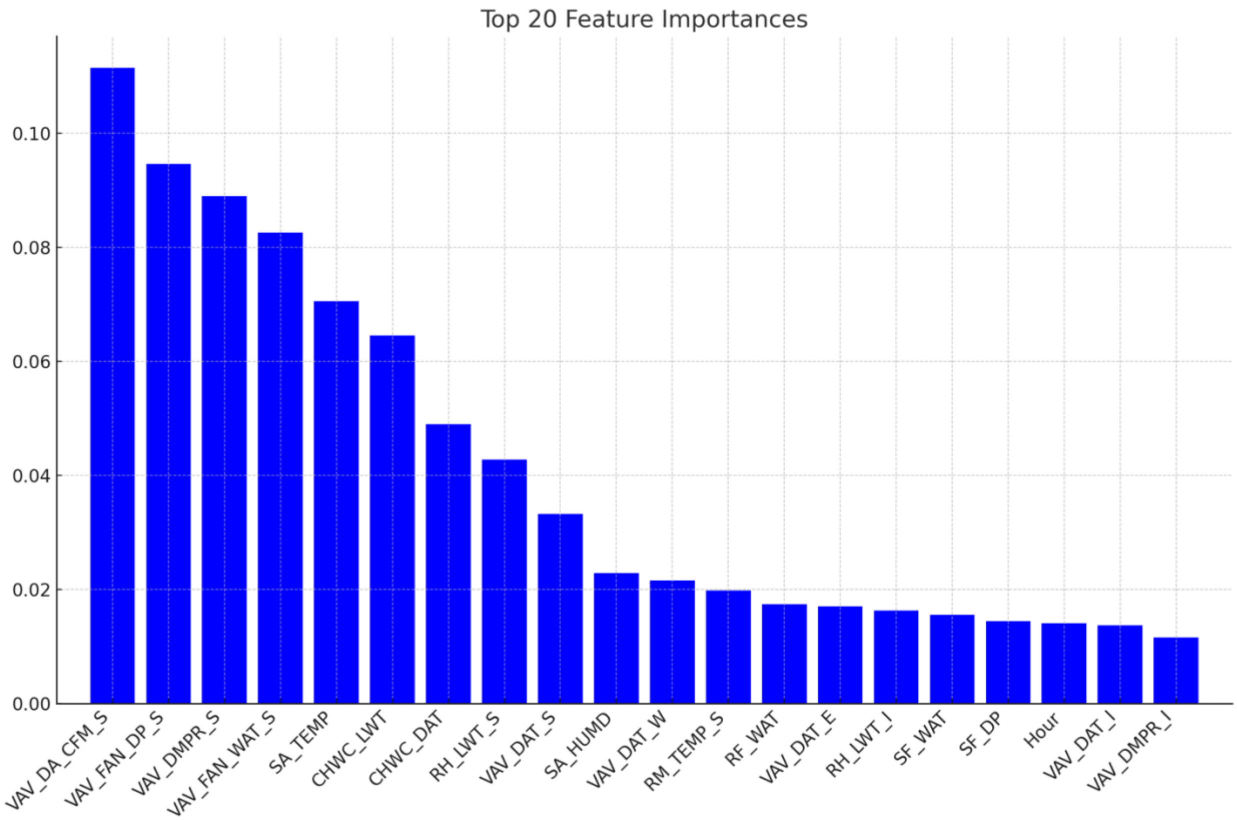
SVM Dataset I: Prediction with two categories:

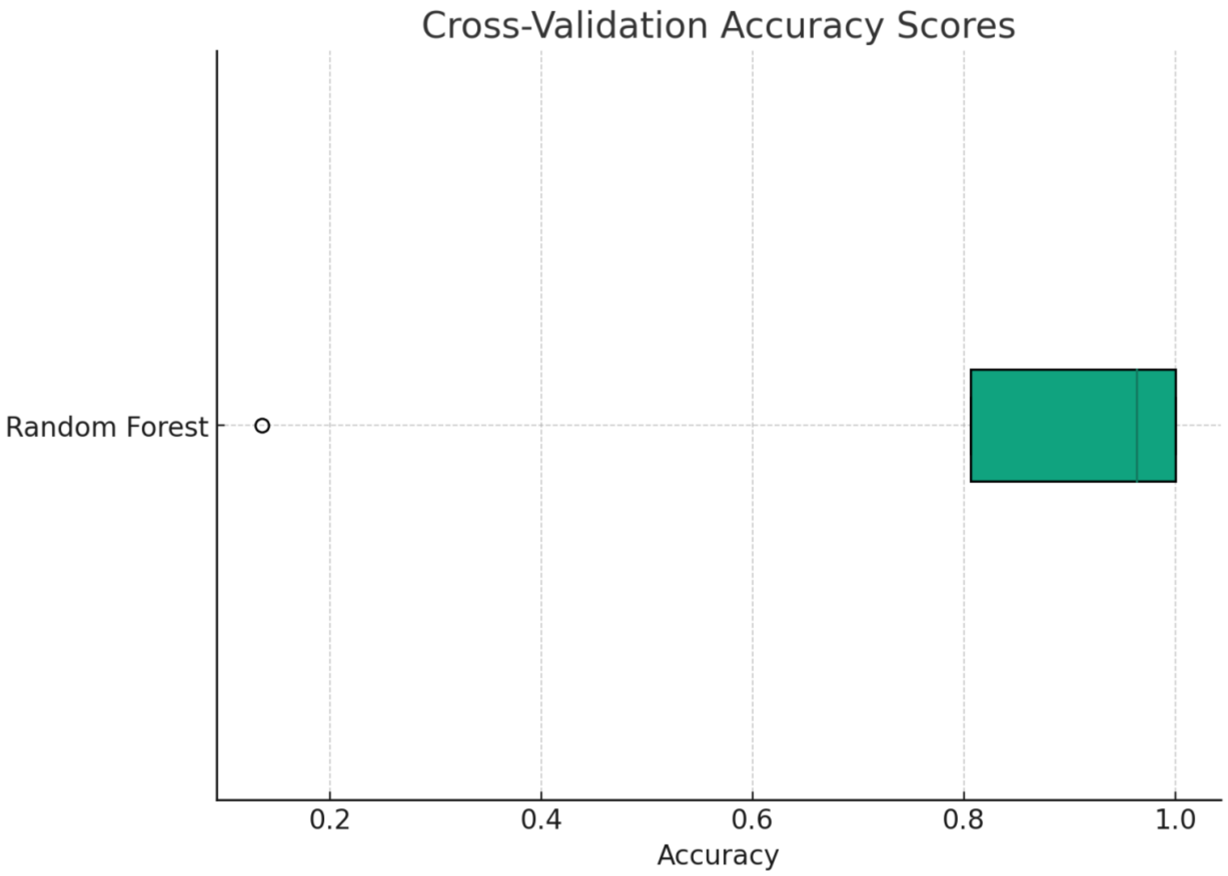


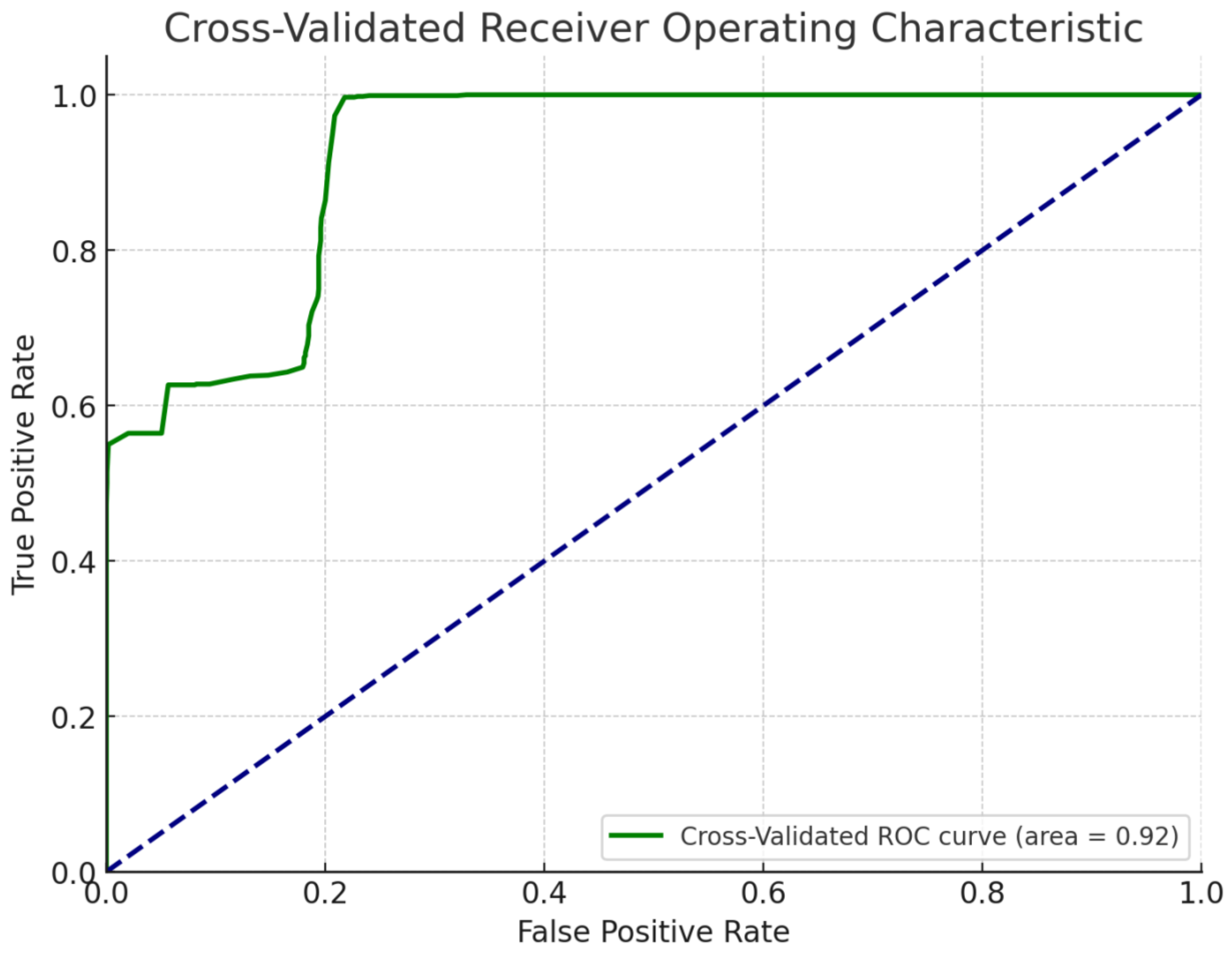


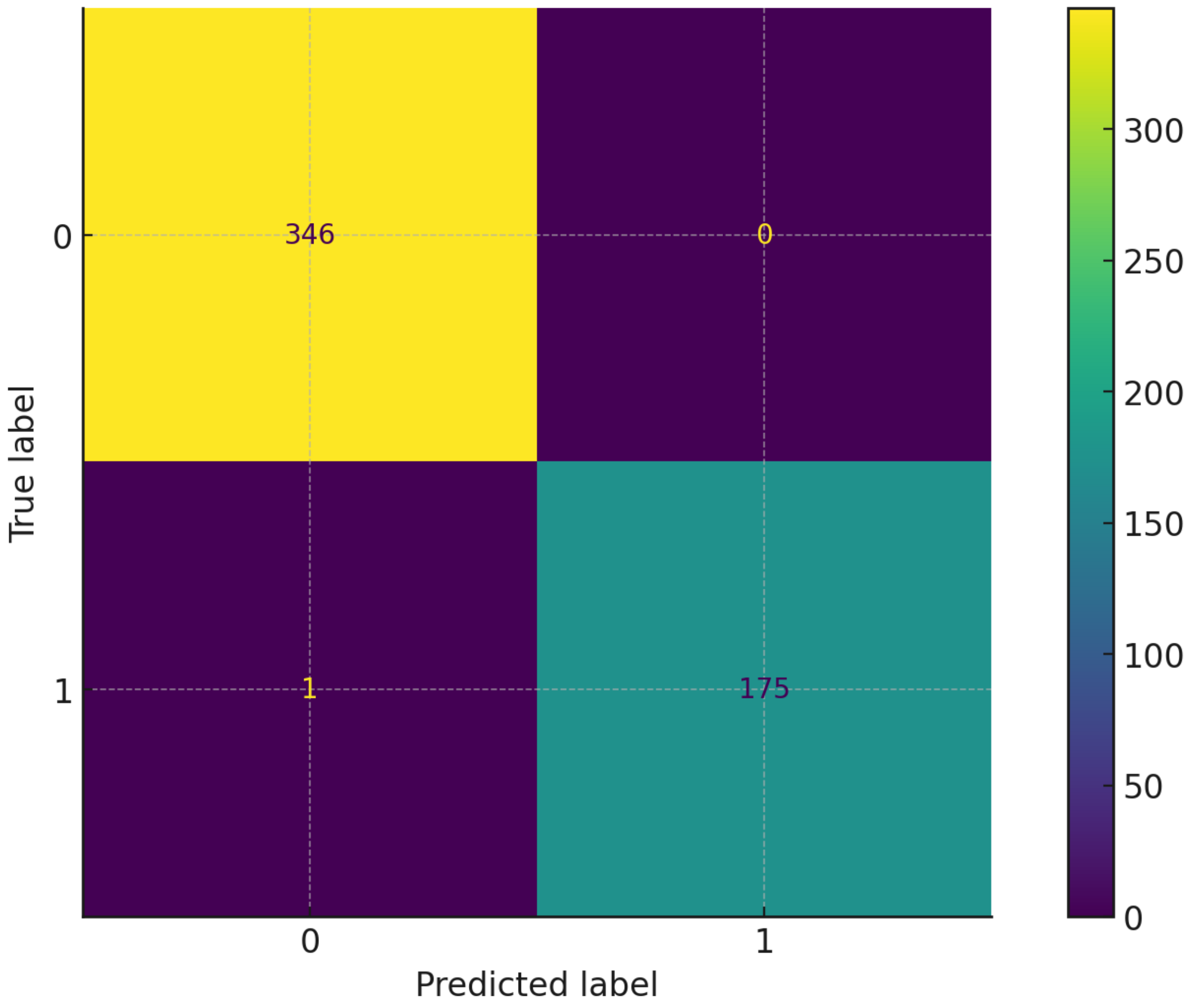


Random Forest Dataset I: Prediction with Two Categories


Top 20 Feature Importances
0.10
0.08
0.06
0.04
0.02
0.00
VAV_DA_CFM_S
VAV_FAN_DP_S
VAV_DMPR_S
VAV_FAN_WAT_S
SA_TEMP
CHWC_LWT
CHWC_DAT
RH_LWT_S
VAV_DAT_S
SA_HUMD
VAV_DAT_W
RM_TEMP_S
RF_WAT
VAV_DAT_E
RH_LWT_I
SF_WAT
SF_DP
Hour
VAV_DAT_I
VAV_DMPR_I

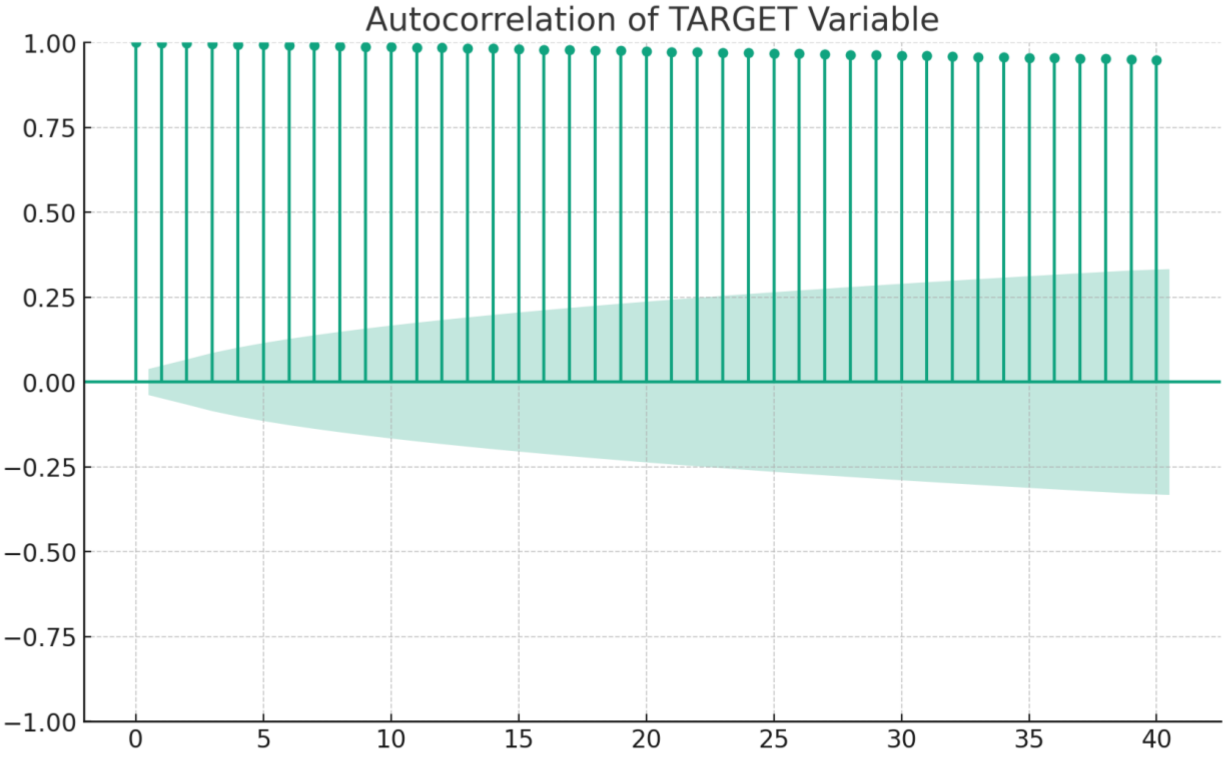

Cross-Validation Accuracy Scores
Random Forest
0.2
0.4
0.6
0.8
1.0
Accuracy


Cross-Validated Receiver Operating Characteristic
True Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
Cross-Validated ROC curve (area = 0.92)



LSTM

AutoCorrelation:



Ultimately the LSTM model was abandoned as the lack of robust data per day per category-- our implementation over indexed and overfit to the testing data.