VCU College of Engineering

# MULT 25-610 Smart Buildings-Sustainability and Efficiency

## Final Design Report

Prepared for

Kenneth Cossaboon and Byron Burns

Siemens


By

Esha Sharma, Grant Forest-Collins, Jamie Gerardo Juarez, Daniel Gubay


Under the supervision of

Daniel Cranston and Joao Soares

05/05/2025

# Executive Summary

The primary objective of the MULT 25-610 capstone project is to develop a robust predictive maintenance (PdM) model leveraging operational data from laboratory HVAC systems within the VCU College of Engineering. Sponsored by Siemens, a leader in building automation systems, the project is aligned with Siemens' Internet of Things (IoT) network, which facilitates smart building operations across the VCU campus.

Key project goals include collecting historical laboratory HVAC data, conducting extensive data mining and profiling, constructing a machine-learning model, and performing rigorous testing and validation to ensure model reliability. The PdM model will be optimized to predict potential failures, and deployed with an intuitive front-end.

High-Level Overview of Timeline:

1. **Data Collection and Preprocessing**: Completed in November 2024
2. **Model Training and Cross-Validation**: Completed in February 2025
3. **Model Iterative Fine-tuning and System Validation**: Completed in March 2025
4. **Final Optimization and Prototype Delivery**: Completed in April 2025
5. **Final Report and Presentation Preparation**: Completed in April 2025

The final predictive maintenance solution adopts an unsupervised, two-layer LSTM autoencoder with 64 hidden states per layer and a single-step reconstruction sequence step. All rows containing missing values are discarded to maintain data quality, and the feature set is trimmed to the most informative continuous and binary FCU data, enriched by six cyclic encodings for hour, day, and season plus an engineered room-temperature-deviation. Training data are drawn from a month-stratified 50 % subsample, preserving daily and seasonal patterns while halving runtime. A five-fold cross-validation loop establishes the anomaly threshold as the mean plus one standard deviation of validation reconstruction loss and embeds this value to be used during testing.

Three principal lessons emerged. First, severe class imbalance (minimal fault-labeled data) rendered fully and semi-supervised models unreliable; the unsupervised approach avoided these pitfalls. Second, uniform sampling with large time gaps introduced temporal blind spots that hid both faults and subtle normal behaviour, whereas fine-grain intra-month sampling retained the important patterns the LSTM needed to learn. Third, pairing stratified sampling with statistical thresholding produced the largest performance gain without a supervised learning approach.

The resulting model performs well in homogeneous settings, predicting the majority of genuine normals in normal behavior datasets and detecting the majority of faults in fault-heavy datasets. Performance declines in heterogeneous mixes—for example, 95 : 5 normal-to-fault—because one-step reconstruction blurs the boundary between minor deviations and true anomalies. Addressing this limitation will require longer sequence windows and additional computational resources, but the current design already satisfies the primary objective of high-recall fault detection with minimal operational overhead.

## Table of Contents

## Section A. Problem Statement

The primary objective of this capstone project is to leverage operational data from laboratory HVAC systems in the VCU College of Engineering to implement a predictive maintenance (PdM) model for facility maintenance and management. The sponsor of this capstone project, Siemens, is a technology company focused on building automation systems, software, healthcare, industrial automation, and energy. In this project's scope, Siemens provides the Internet of Things (IoT) network that implements building automation systems around the VCU campus. The project stakeholders are Siemens and the VCU building operations team.

The current approach to building operations is a reactive maintenance model, which involves allowing systems or, more minutely, subsystems to malfunction and only intervening to repair them after the issue arises. This break-fix cycle is expensive and disruptive due to spontaneous out-of-budget repair expenses needed for sudden machine failures and subsequent unplanned machine downtime [1]. By implementing a PdM model, we aim to improve the reliability of laboratory HVAC systems, reducing the risk of sudden machine failures and the associated costs and disruptions.

The application of PdM modeling in building operations also supports efforts towards sustainability. By proactively identifying emerging failures in HVAC systems, energy consumption, waste, and a building's carbon footprint are reduced. PdM fosters a safe environment for building occupants. Addressing laboratory HVAC system failures proactively decreases the likelihood of hazardous situations that affect occupants' safety. Proper ventilation, heating, and cooling in a laboratory allows for tight air quality control, protecting occupants' health and preventing outdoor air contamination [2].

In the PdM project life cycle, data pre-processing and data mining are used to construct descriptive or predictive (machine learning) models. These models discover knowledge from systems data to predict anomalies and the state of health of equipment [3]. Knowledge-based or data-driven approaches to PdM vary based on business objectives and resource availability. Knowledge-based approaches rely on the knowledge of experts in the domain where predictive maintenance is applied. Thus, rule-based, case-based, and fuzzy algorithms are utilized in knowledge-based PdM models and can suffer from low accuracy when applied to complex data. Data-driven PdM models use machine learning and data analytics on historical system datasets to predict system health and detect anomalies in real-time data [4].


**Figure 1. Workflow of predictive maintenance [3]**

Remaining Useful Life (RUL) PdM models estimate the lifespan of a machine in building subsystems based on historical, run-to-failure data periods. By analyzing the operational performance of machine components near the end of their lifespan, based on historical work orders, RUL models provide knowledge of when building operators can anticipate equipment failures and implement necessary maintenance. Kang et al. used the principal component analysis feature selection, grid search parameter optimization, and a multi-layer perceptron (MLP) machine learning algorithm to implement an RUL model. The MLP model outperformed implementations of random forest and support vector regression algorithms for the machine learning component of the RUL models [5].

Normal Behavior Models (NBM) establish a “normal” operation baseline for building sub-systems. By mining historical, healthy system data to construct a machine learning-based NBM model, anomalies in real-time data that indicate potential malfunctions or failures can be flagged by the model and reviewed by building operators. Chesterman et al. used linear interpolation for missing values imputation, sensor signal filtering, and an elastic net regression (ENR) machine learning algorithm to implement an NBM model. Chesterman et al. found the ENR model outperformed a more complex machine learning model, support vector regression. The ENR underperformed when compared to lighter machine learning models, multi-layer perceptron and gradient boosting machines. However, the study found that elastic net regression suffered less model degradation from extreme changes in data compared to the lighter models [6]. By detecting deviations from normal behavior early, building operators can implement necessary maintenance before the problem worsens.

## Section B. Engineering Design Requirements

The Engineering Design Requirements for this project were derived by carefully analyzing Siemens' specific needs, industry benchmarks, and existing smart building systems. The design objectives focus on creating an efficient, predictive maintenance system that integrates seamlessly with Siemens' infrastructure while optimizing system performance and reducing downtime. Each requirement, such as cost, safety, usability, and scalability, was researched and aligned with relevant codes and standards to ensure a robust and compliant solution. These requirements will be revisited periodically to ensure that the design stays aligned with the project's goals.

## B.1 Project Goals (Client Needs)

The purpose of this project is to develop a comprehensive predictive maintenance system that leverages Siemens' data and machine learning technologies to enhance the management of smart building systems. The client, Siemens, requires an efficient solution that can harness large volumes of operational data from building systems to predict equipment failures, optimize energy use, and improve overall system reliability. The goals of this project focus on extracting actionable insights from this data and delivering them in a user-friendly manner that enhances decision-making and performance monitoring.

The overall goals of the project include:

1. **Access and Process Siemens Data**: Collect, manage, and analyze operational data from Siemens' smart building systems.
2. **Understand and Recognize Patterns in Data**: Use advanced data analysis techniques to detect patterns and trends in the data that correlate with mechanical faults and system inefficiencies.
3. **Develop a Machine Learning Model**: Implement a machine learning model that accurately predicts mechanical failures in key systems like HVAC, pumps, and compressors based on the patterns identified in the data.
4. **Testing and Validation of the Model**: Conduct thorough testing and validation to ensure the model improves on the initial benchmarks.
5. **Optimize System Performance**: Provide insights and recommendations to optimize the operational efficiency of building systems, focusing on reducing energy consumption and extending equipment lifespans.
6. **Data Visualization**: Create a visualization dashboard to display system performance, failure predictions, and energy optimization insights, enabling building managers to make informed, proactive decisions.

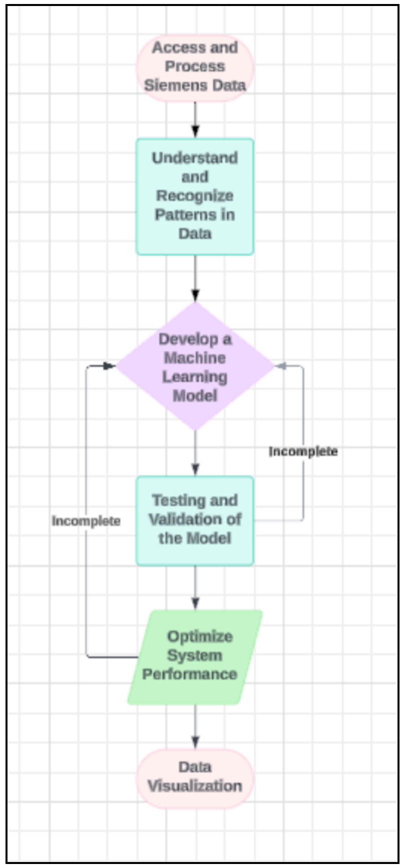



**Figure 2. Project Goals Flowchart**

In summary, this project aims to provide Siemens with a robust predictive maintenance system that utilizes machine learning to analyze operational data from smart building systems. The primary goal is to predict mechanical failures, optimize energy efficiency, and enhance system reliability. By leveraging advanced data analysis techniques, a machine learning model will be developed, tested, and validated to accurately forecast equipment issues. The project will also include a data visualization dashboard, enabling building managers to make informed decisions and ensure the smooth operation of critical systems.

## B.2 Design Objectives

Below are the design objectives following the S.M.A.R.T guidelines.

**Specific**

- The design will predict potential failures in HVAC units using historical data related to the units' operational parameters, maintenance history, and environmental factors.

**Measurable**

- The design will achieve a failure prediction accuracy that meets or improves upon the accuracy of a baseline model or metric.

**Achievable**

- The design will be achievable within the available data collection and computational resources, leveraging current machine learning techniques like classification algorithms and time-series analysis.

**Realistic**

- The design will realistically provide predictions within a timeframe that allows for preventive action, generating anomaly alerts two weeks before a potential malfunction or failure is likely to occur.

**Time-bound**

- The design will be implemented, tested, and deployed within a 6-month timeline, with iterative model improvements over 2-week sprints.

## B.3 Design Specifications and Constraints

The goal of this project is to develop a **predictive maintenance system** that leverages operational data from Siemens' building control systems to anticipate mechanical failures in critical systems such as HVAC, pumps, and compressors. This project focuses on identifying potential system breakdowns before they occur, allowing for timely interventions and reducing unplanned downtime.

**Prediction Accuracy**

- **Specification**: The predictive model must achieve an anomaly prediction accuracy that improves upon the anomaly prediction of a baseline model/metric.
- **Objective**: Ensure the system reliably predicts anomalies in equipment operations, providing actionable insights for preventive maintenance through predictive data mining.
- **Measurable Criteria**: The predictive model will be validated through usage of appropriate testing datasets to confirm that the model can accurately make anomaly predictions.

**Data Collection and Processing**

- **Specification**: The system must collect and process real-time data from **at least 1** critical building system (HVAC, pumps, compressors) with a data refresh rate of **1 week**.

- **Objective**: Provide continuous monitoring of system performance to detect and predict mechanical failures.
- **Measurable Criteria**: The system's data refresh rate and real-time monitoring capabilities will be tested to ensure compliance with the specified time frame.

**Integration with Siemens Building Control Systems**

- **Specification**: The predictive maintenance system must integrate seamlessly with Siemens' existing building control systems, utilizing operational data from at least one sensor installed in the Engineering Research Building without requiring additional hardware infrastructure.
- **Objective**: Leverage existing data streams from Siemens infrastructure to implement the predictive maintenance model.
- **Measurable Criteria**: Integration tests will confirm that the system can access and process data directly from Siemens' control systems without disruptions.

**Budgetary Constraint**

- **Specification**: The total project cost, including software and any necessary infrastructure upgrades, must not exceed $1,000, unless more funding is received.
- **Objective**: Ensure the project remains cost-effective while meeting all functional requirements.
- **Measurable Criteria**: All expenditures will be tracked, and the final project costs must remain within the $1,000 limit.

**Testing and Validation**

- **Specification**: The system must undergo **extensive testing and validation** using historical operational data to confirm its predictive accuracy.
- **Objective**: Ensure the system's predictive model is reliable before live deployment.
- **Measurable Criteria**: The system will be tested against historical data to verify its prediction accuracy, and it must achieve this before moving to real-time deployment.

**Scalability**

- **Specification**: The system must be designed to scale across multiple subsystems within a single building without requiring significant changes in architecture.
- **Objective**: Ensure the predictive maintenance model can expand to cover more systems in the future.
- **Measurable Criteria**: The system architecture will be reviewed to ensure scalability and tested on additional systems for performance consistency.

**Real-Time Alerts**

- **Specification**: The system must generate **preventative alerts on a weekly basis** for building operators when performance metrics deviate beyond acceptable thresholds.
- **Objective**: Provide timely notifications for preventive action before failure occurs.
- **Measurable Criteria**: The system's alert system will be tested to ensure that alerts are generated immediately when thresholds are exceeded.

## B.4 Codes and Standards

For the development of a predictive maintenance system using machine learning, the following codes and standards are critical to ensure data security, system interoperability, and electrical safety:

1. **IEEE 802.3 (Ethernet Standards) – Networking Protocols**: Defines the standards for wired Ethernet networks, ensuring that data transmission between building control systems and the predictive maintenance model is reliable and fast.
    - **Relevance**: Ensures seamless communication and integration between sensors, building control systems, and the machine learning platform.
2. **ISO/IEC 27001 (Information Security Management) – Data Security**: Specifies requirements for establishing, implementing, maintaining, and improving an information security management system.
    - **Relevance**: Critical for ensuring that data gathered and processed by the predictive maintenance system is secure and protected from unauthorized access or breaches.
3. **NIST SP 800-53 (Security and Privacy Controls for Information Systems) – Data Integrity**: Provides a catalog of security and privacy controls for federal information systems, including cloud-based data storage.
    - **Relevance**: Ensures that data processed by the machine learning algorithms is stored and handled securely, especially if cloud-based resources are used.
4. **Project Haystack (Data Tagging Standard for Building Automation) – Data Tagging and Organization**
    - **Relevance**: Project Haystack offers an open standard for tagging and organizing data generated by building automation systems like HVAC. This ensures that data collected from Siemens' systems can be structured in a way that enhances pattern recognition and machine learning, making it easier to identify trends and anomalies.
5. **ASHRAE 135 (BACnet – A Data Communication Protocol for Building Automation and Control Networks) – System Interoperability**
    - **Relevance**: This standard is critical for ensuring that different control systems, such as HVAC, lighting, and metering, can communicate seamlessly using the BACnet protocol. Integrating your predictive maintenance system with Siemens' building control infrastructure will rely on this standard to ensure data is shared effectively between subsystems.

6. **ANSI/ASSP Z9.5-2022 (Laboratory Ventilation) – Ventilation Safety for Laboratories**
   - **Relevance**: This standard establishes the minimum requirements for laboratory ventilation systems, focusing on controlling air contaminants and ensuring safe airflow within lab spaces. If your predictive maintenance system will include HVAC systems for laboratory spaces, this standard is crucial to ensure compliance with proper ventilation and air quality management.
7. **AIHA Laboratory Ventilation Standard (Z9.5) – Airflow and Safety in Labs**
   - **Relevance**: This standard outlines best practices for airflow and contaminant control in laboratory environments. If your predictive maintenance system includes HVAC systems in labs, compliance with this standard ensures proper ventilation, air quality, and safety. It specifically addresses fume hoods and exposure control devices, which are key to maintaining safe lab environments.

## Section C. Scope of Work

### Key Objectives

- Design and implement a machine learning model to predict potential HVAC unit failures.
- Set a baseline for our preliminary ML model
- Ensure model precision and build upon baseline numbers.
- Provide failure/maintenance predictions of two weeks prior to the occurrence.
- Deploy the model within 6 months, two-week sprint cycles. As progress is made on the project, the plan for the upcoming month will be updated on a 2-week sprint basis.

### Boundaries

- Time**:** The project will be completed within a 6-month period.
- Budget**:** The project must stay within the allotted budget (exact amount defined to be $1000).
- Resources**:** The team will have access to data from Siemens, computational resources for training the model, and guidance from the faculty advisor.

### Development Methodology

The **Agile** methodology will be followed, with 2-week sprints to allow for iterative development, frequent feedback from stakeholders, and adjustments to the model as needed. Regular communication with the project sponsor and faculty advisor will ensure the project stays on track.

## C.1 Deliverables

### Deliverables

- Preliminary design report
- A functional ML model capable of predicting HVAC failures with an improved accuracy based on our initial baselines.The trained predictive maintenance models will be given to Siemens in the form of Python code.
- Documentation detailing the training and testing datasets, preprocessing steps, data splitting methodologies, model architecture, and testing procedures.
- Fall poster and presentation
- A simple website that displays our graphs/model.
- Final report including a summary of the project, results, and future recommendations.
- A drive holding all the data used (post-processed testing and training datasets, database implementation of data and related schemas)
- Capstone EXPO poster and presentation
- This should all culminate in a minimum viable product.

- **Out of Scope:**
  - HVAC system hardware modifications.
  - Long-term maintenance of the model post-deployment.
  - Ongoing data collection and retraining after the initial deployment phase (stop after minimum viable product)
  - Limitation of data domain (data is collected from only one lab)

**Risks/Obstacles**

- **Engineering the model**: It is the team's first time constructing a machine learning model; thus, obstacles will arise as the project progresses. The team may take more time in some areas during development so the defined sprint schedules will be iterated upon as needed.
- **Data availability issues**: If sufficient historical data is not available, the timeline may shift to accommodate additional data collection.
- **Model performance**: If the model does not meet performance criteria, the team will explore alternative algorithms and techniques within the timeline.
- **Scope creep**: Any new requests or changes to the project scope will need formal approval from both the faculty advisor and project sponsor.

## C.2 Milestones

This project follows a structured approach where key milestones represent major phases necessary to ensure the successful delivery of the predictive maintenance model for HVAC systems. These milestones include critical tasks such as data collection, computational model development, and testing procedures. The project timeline is broken into 2-3 week increments, with each milestone focusing on specific activities like data preprocessing, model training, optimization, and validation. As the project progresses, adjustments may be made based on findings from earlier phases, consistent with an Agile development approach. The timeline takes into account the time needed for data processing, development of the machine learning model, testing, and preparing final reports and presentations. The completion dates of each milestone are carefully planned to keep the project on track, ensuring the final deliverables are completed within the allocated time frame. Regular reviews with the faculty advisor and project sponsor will ensure that any necessary changes to the project scope or timeline are documented and agreed upon.

1. **Data Collection and Preprocessing**: Complete by January 2024
2. **Model Training and Cross-Validation**: Complete by First Half of February 2025
3. **Model Iterative Fine-tuning and System Validation**: Complete by Second Half of February 2025
4. **Final Optimization and Prototype Delivery**: Complete by March 2025
5. **Final Report and Presentation Preparation**: Complete by March 2025

**Timeline & Milestones**

**Disclaimer: There will be 2 two-week sprints for each month, which will be specified based on ongoing substantive progress and/or potential setbacks. Currently, only October and November have a defined sprint schedule.**

- **October-January: Data Collection, Preprocessing, Profiling**
    - Gather data on HVAC system operational parameters, including airflow rates, pressure differentials, heat transfer efficiencies, and mechanical wear in components such as fans, compressors, and heat exchangers.
    - Analyze mechanical failure modes and degradation patterns over time (e.g., wear and tear on bearings, belt slippage, or coil fouling).
    - Gather historical data on HVAC operational parameters, maintenance records, and environmental factors
    - Clean and preprocess data, imputation of missing or incorrect values, handle imbalanced classes, feature selection, feature engineering
        - Sprint Schedule for October
            - 14th-31th:
                - Evaluate and finalize the source lab and historical time-frame for data collection
                    - Collect historical data from Siemens team
                - Choose database technologies for data storage and develop data schema for configuration of database
                - Initialize data preprocessing and profiling
                    - Examine data formats, feature naming conventions, data types and intervals, potential missing values

            - Perform data distribution analysis on numerical and symbolic features.
        - Inspect mechanical systems in the lab, to verify their operational condition and suitability for the data collection timeframe
    - Sprint Schedule for November
        - 1st-15th:
            - Continue data preprocessing and profiling
                - Implement correlation and significance tests and data preparation techniques to remove noise from dataset
                - Investigate if the objects and features in the raw dataset are valid, useful, high-enough quality for anomaly detection modeling
            - Compare historical data and recent weekly batch data for data consistency
            - Validate mechanical data features like pressure drops, airflow, and motor RPM. Identify and address inconsistencies that could affect predictive modeling for mechanical failure.
        - 16th-30th:
            - Continue data preprocessing and profiling
                - Evaluate metrics to flag and exclude anomalies from testing dataset for modeling purposes
                - Evaluate and address potential class imbalances through addition of new computed objects, class oversampling/undersampling
                - Select preliminary data mining and machine learning algorithms for model training

- **January-February: Model Training, Cross-Validation Testing, Iterative Fine-Tuning**
    - Include mechanical-specific metrics in the machine learning models, such as motor operating efficiency, vibration levels, and temperature gradients.
    - Ensure the model takes into account mechanical wear and tear patterns, predicting failures of parts such as belts, bearings, and heat exchangers
    - Develop initial ML models (such as logistic regression, decision trees, random forests, clustering, neural networks)
    - Train the models on the preprocessed dataset and evaluate model performance on test dataset through cross-validation and comparisons to baseline
- **March: Model Optimization, Model Deployment**
    - Fine-tune the model by adjusting hyperparameters and combining multiple learning algorithms if necessary
    - Deploy the model and training/testing datasets as a deliverable in a cloud workspace

## C.3 Resources

The completion of this project will require various essential resources. Siemens will provide weekly data sets and a large historical data set, both of which will be pre-processed, mined, and analyzed to develop a training dataset for model development in Python. Libraries such as NumPy, Pandas, scikit-learn, Tensorflow, and Pytorch will be utilized for modeling. Additionally, the team is gathering construction documents to analyze the mechanical components currently installed, ensuring the model aligns with the real-world system. Meetings with Siemens will further aid in the integration of data and the decision-making process regarding the cloud data solution, which will be finalized based on cost considerations. Cloud service providers such as AWS and Azure will be utilized for large-scale data processing and model training. GitHub will manage version control and team collaboration. These resources ensure efficient progress toward the project's deliverables.

# Section D. Concept Generation

## Design Concept 1: Deep Learning with a Long Short-Term Memory Neural Network

LSTM neural networks are appropriate for time-series data as they learn long-term dependencies, allowing information to persist across many time-steps. HVAC systems' behavior is influenced by short-term fluctuations and long-term seasonal trends. So, the implementation of LSTm can be effective in predictive maintenance.

1. **Smoothing for Outlier Removal: Seasonal Trend Decomposition (STM)**
    a. Algorithm:
        i. LOESS STL
        ii. X11
    b. Reasoning:
        i. Decomposition into trend, seasonal, and residual components enforces seasonality in data
2. **Noise Reduction: Moving Hourly Averages**
    a. Algorithm:
        i. Moving average (simple or weighted)
        ii. Exponential moving average
    b. Reasoning:
        i. Reduces effects of short-term fluctuations in data, thereby reducing noise
3. **ML Model: LSTM**
    a. Architecture:
        i. Bidirectional: forward and backward patterns in data
        ii. Deep: two or more hidden layers within LSTM
    b. Activation Functions:
        i. ReLU

c. Loss Function:
    i. Binary Cross-Entropy

4. **Model Validation**
    a. Predictive Quality Metrics:
        i. Accuracy
        ii. MCC (Matthew's Correlation Coefficient)
        iii. Sensitivity
        iv. Specificity
        v. Cohen's Kappa

Positives of this design concept include the temporal dependencies of LSTM, flexibility in parameterization of the neural network, and high predictive performance of LSTM. Cons of this design concept include the computation cost and risk of overfitting due to imbalanced classes (normal behavior vs. faults). The black-box nature of neural networks means it will be hard to explain the classification choices of the model to stakeholders.

## Design Concept 2: Traditional Machine Learning with Feature Engineering

Simpler, more traditional machine learning models like Random Forest and Gradient Boosted Trees are still effective, and easier to explain to stakeholders. Feature engineering offers the ability to select the most informative features, enhancing model performance.

1. **Outlier Removal: Statistical Methods**
    a. Algorithm:
        i. Z-score
        ii. Interquartile Range (useful for skewed classes)
    b. Reasoning:
        i. Removing outliers improves generalization and validity of model
2. **Noise Reduction: Moving Average/Filtering**
    a. Algorithm:
        i. Moving average (simple or weighted)
        ii. Savitsky-Golay filter
        iii. Fourier Transform
    b. Reasoning:
        i. High-frequency noise reduction
3. **Feature Engineering: Principal Component Analysis (PCA)**
    a. Reasoning:
        i. Extraction of most informative features reduces risk of overfitting and improves model performance
4. **ML Model**

    a. Algorithm:
        i. Random Forest: Forest of decision trees
        ii. Gradient Boosting (XGBoost, LightGBM): Sequential ensemble of boosted trees
        iii. Support Vector Regression: Implements the power of Support Vector Machine with speciality for time-series data
5. **Model Validation**
    a. Predictive Quality Metrics:
        i. Accuracy
        ii. MCC (Matthew's Correlation Coefficient)
        iii. Sensitivity
        iv. Specificity
        v. Cohen's Kappa

Positives of this design concept include simplicity and computational speed, with the exception of SVR which sacrifices speed for accuracy. Other pros include easier interpretability of decision making and robust noise handling. Cons of this design concept include limited learning of temporal trends in Random Forest and GBM compared to SVR. Additionally, if too many methods to reduce dimensionality and noise are used, then model performance may suffer. It is hard to say how many methods are too many. Repeated iteration of this design concept is needed for optimization.

## Design Concept 3: Hybrid Model with Time-Series and Classification Layers

This design concept combines the strengths of LSTM in terms of time-series data handling and traditional ML classification algorithms. The overall idea is to predict expected values and then classify deviations as anomalies.

1. **Outlier Removal: STM and Statistical Methods**
    a. Algorithm:
        i. LOESS or X11 STM
        ii. Z-score
        iii. Interquartile Range (useful for skewed classes)
    b. Reasoning:
        i. Combining these two methods removes trend-related outliers and raw outliers
2. **Noise Reduction: Hybrid Smoothing/Filtering**
    a. Algorithm:
        i. Moving average (simple or weighted) + Savitzky-Golay Filtering
    b. Reasoning:

        i. Reduces noise but also keeps important time-series trends

3. **ML Model: Hybrid Approach**
    a. Part 1: Time-series forecasting
        i. Algorithm:
            1. LSTM or similar algos like Prophet
        ii. Reasoning:
            1. Forecast expected behavior of HVAC system component
    b. Part 2: Anomaly Detection
        i. Algorithm:
            1. Isolation Forest, Random Forest, SVR, or One-Class SVM
        ii. Reasoning:
            1. Highly effective classification algorithms will help detect potential anomalies from forecasted data
4. **Model Validation**
    a. Predictive Quality Metrics:
        i. Accuracy
        ii. MCC (Matthew's Correlation Coefficient)
        iii. Sensitivity
        iv. Specificity
        v. AUC-ROC

Positives of this design concept include the hybrid approach which is robust in handling the two objectives of forecasting and anomaly detection. Cons of this design concept include the increased computational complexity, the black-box nature of the model choice, increased design complexity, and high sensitivity to hyper parameterization. The computational intensity of this design might impact real-time anomaly detection. There is also a risk of learning the potential errors in the forecasting model and applying them to the anomaly detection. The fact that this is a hybrid approach creates this problem.

## D.4 Summary

This section evaluates three predictive maintenance approaches. The first leverages Long Short-Term Memory (LSTM) neural networks to handle time-series dependencies, using techniques like Seasonal Trend Decomposition and moving averages for data smoothing. The second focuses on traditional machine learning models, such as Random Forest and Gradient Boosting, paired with feature engineering methods like Principal Component Analysis for simplicity and interpretability. The third hybrid approach combines LSTM for time-series forecasting with classification algorithms like Isolation Forest for anomaly detection. Each concept is validated using metrics such as Matthews Correlation Coefficient, sensitivity, and specificity, balancing predictive accuracy, computational efficiency, and implementation complexity.

## Section E. Concept Evaluation and Selection

## E.1 Selection Criteria:

1. **Model Validity**
    a. Metric:
        i. MCC (Matthew's Correlation Coefficient)
    b. Rationale:
        i. MCC calculates correlation between predicted and actual outcomes, taking into account all four types of values in a confusion matrix (true positives, true negatives, false positives, false negatives).
        ii. MCC is an effective predictive quality index for datasets with skewed class distributions.
2. **Scalability**
    a. Metric:
        i. Training time (seconds or minutes)
    b. Rationale:
        i. Scalability measures model performance when it receives big data as input. Shorter training times are preferred. This is important because the amount of data being handled on a daily basis is only increasing as technology progresses.
3. **Financial Cost**
    a. Metric:
        i. The total dollar amount required to integrate the model into Siemens systems, host the model on a cloud platform, and stream real-time data into the model.
    b. Rationale:
        i. This criteria is important to determine whether large-scale model deployment can be accounted for in the budget.
4. **Real-Time Detection Capability**
    a. Metric:
        i. Latency (ms)
    b. Rationale:
        i. The measurement of time delay during real-time anomaly detection is important, especially when prompt maintenance needs to be implemented.
5. **Computational Complexity**

a. Metric:
    i. Big O notation
b. Rationale:
    i. The overall computation cost of the model, including data preparation, model training, and model testing is important to monitor. Ultimately, this cost hinges on the choice of the ML algorithm, which will remain permanent after model deployment. It is important to strike a balance between Big O computational complexity and model correctness/validity. Having less of one usually means having more of the other.

In this data science-centered project, it is not possible to completely evaluate the design concepts before model development. As of December 2024, implementation of all three design concepts has not been completed. The first half of the project was spent on understanding the problem domain, data collection, data preparation, and model architecture design.

**Table 1. Selection Criteria Decision Matrix**

| | **Design Concept 1** | **Design Concept 2** | **Design Concept 3** |
|---|---|---|---|
| **Model Validity (MCC)** | | | |
| **Scalability (milliseconds)** | | | |
| **Financial Cost ($)** | | | |
| **Real-Time Detection Capability (milliseconds)** | | | |
| **Computational Complexity (Big O)** | | | |
| | | | |
| **Total Score** | | | |

## E.2 Summary

This section outlines the criteria for evaluating the predictive maintenance design concepts. Key metrics include model validity (using MCC for assessing predictive quality in imbalanced datasets), scalability (measured by training time), financial cost (total deployment expenses), real-time detection capability (latency), and computational complexity (Big O notation). These criteria balance accuracy, efficiency, and feasibility. Although the decision matrix remains incomplete, progress in data preparation and architecture design will guide the evaluation as models are developed and tested.

# Section F. Design Methodology

## F.1 Computational Methods (Modeling, Time-series Analysis, Training Neural Networks, Boundary Conditions and Assumptions)

- For our computational methods, we used Python libraries like scikit-learn for pre-processing and PyTorch to construct our neural network.

**Time-Series Analysis**:

- To apply time-series forecasting techniques, we plan on using Long Short-Term Memory (LSTM) neural networks, to predict future performance and detect deviations.

**Training Neural Networks**:

- Train neural networks using labeled historical data. Features will include temporal patterns like seasonal fluctuations or repetitive faults.

**Boundary Conditions and Assumptions**:

- Historical data must be comprehensive and accurately labeled.
- Assumption: Historical conditions represent typical operational scenarios.

**Data Exploration and Visualization:**

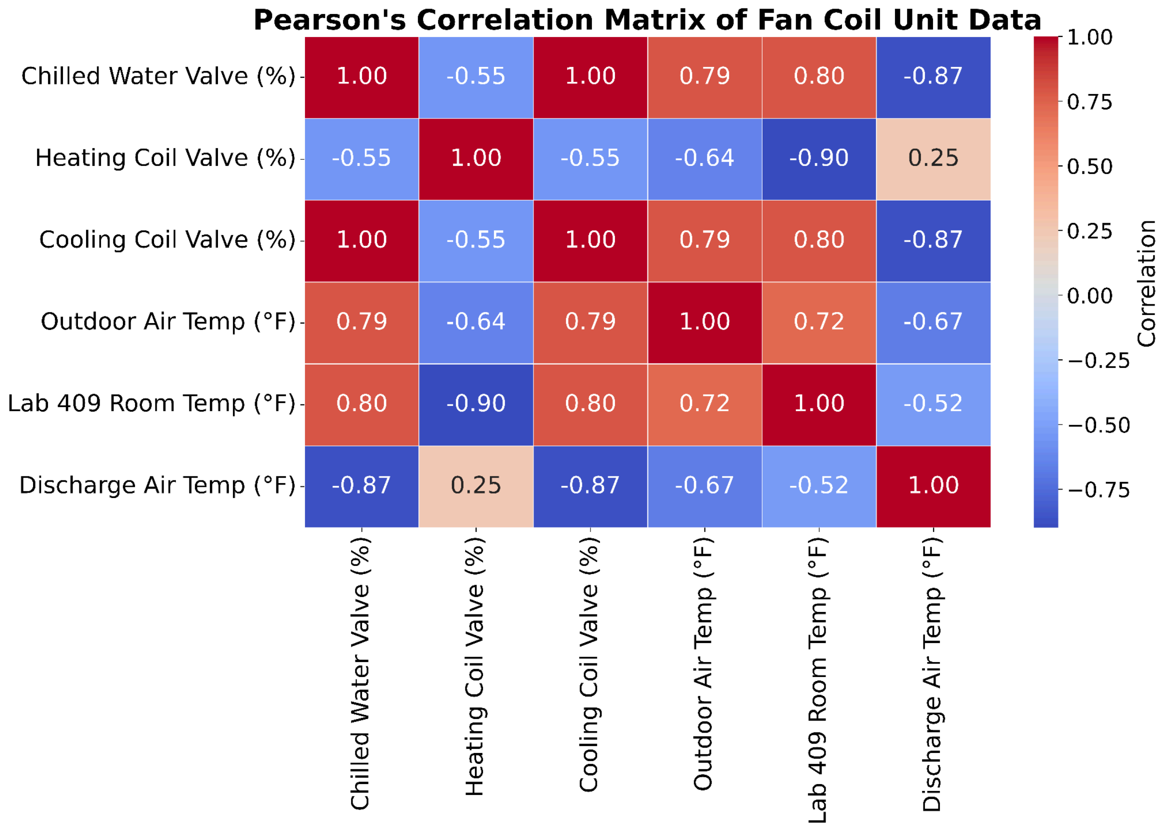



**Figure 3. Correlation Matrix**

The correlation matrix (figure above) helps identify strong relationships between FCU variables from COE West Hall Lab 409 HVAC data. This establishes normal system behavior—i.e., the foundation for anomaly detection and equipment lifespan prediction.

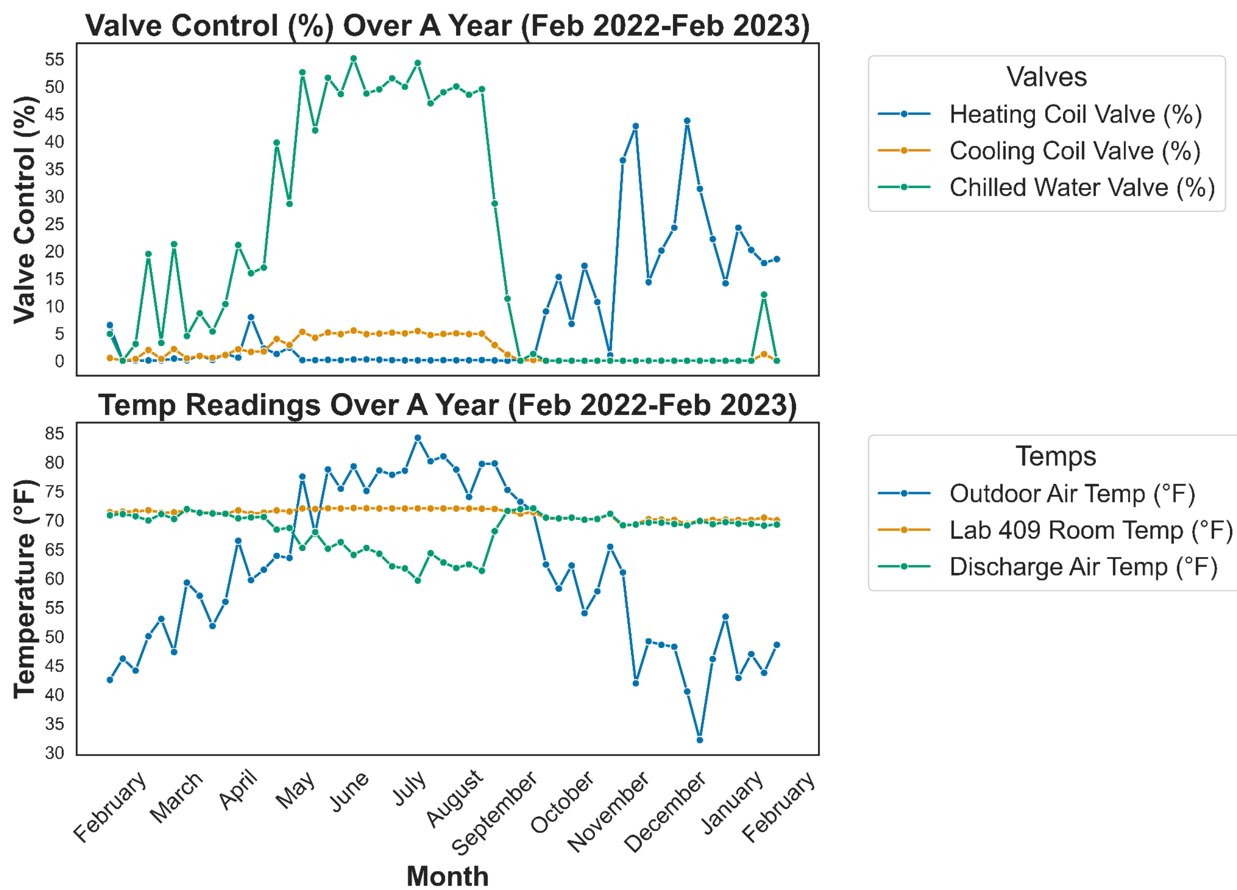



**Figure 4. Visualization of HVAC Data (COE West Hall Lab 409)**

As shown in the graph above, specifically in the top graph, knowledge of an area's seasons helps us predict which valve will be used and how much. We can then set a baseline of operation, which allows us to flag potential anomalies as they occur.

Heating Coil Valve (%) and Outdoor Air Temp (°F) have an inverse relationship. This is normal system behavior. Gathering observations like this is critical to building a correlation matrix. These types of correlations will then be used to help build the neural network.

## F.2 Experimental Methods (Validation Dataset, Fault Simulation)

Focus shifts to validating predictions against known outcomes in the historical dataset:

- **Validation Dataset**:
  - Collection of COE West Hall Lab 409 historical HVAC data, ranging from February 2022 to February 2023
  - Reserve a portion of historical data as a test set to measure model accuracy.
- **Fault Simulation**:
  - Overlay synthetic faults on historical trends to test the robustness of anomaly detection.

## F.3 Architecture/High-level Design

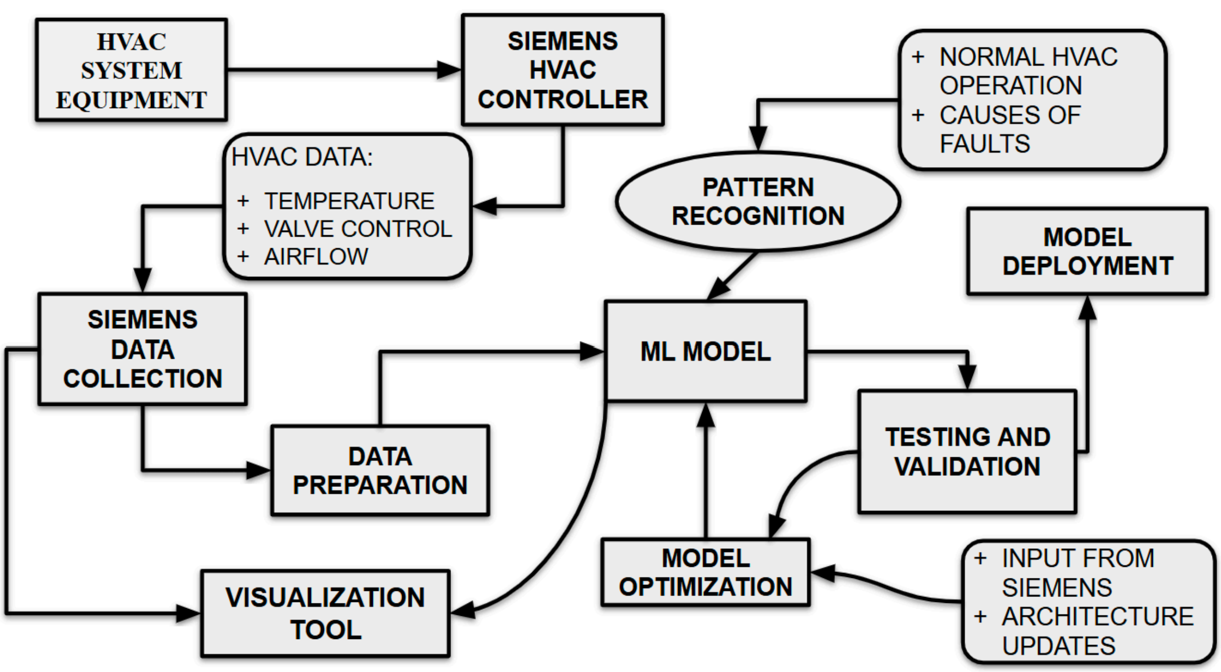



**Figure 5:** Predictive Maintenance Architecture

**System Architecture:**

- The PdM system architecture consists of three key components:
  1. Data Collection Layer: Sensors integrated with Siemens control systems gather operational data.

2. Machine Learning Layer: Preprocessed data is used to train models for predictive maintenance.
3. Visualization Layer: Insights and alerts are displayed on a dashboard for facility managers.

**High-Level Workflow:**

1. Data collection and preprocessing
2. Feature engineering and model training
3. Model evaluation and validation
4. Deployment of predictive maintenance dashboards

**Assumptions:**

- HVAC system parameters remain consistent over the testing period.

## F.4 Mechanical Validation Procedure

**Normal HVAC Operations**

- Heating Coil Logic:
    - During colder outdoor temperatures:
        - Validate that the heating coil valve opens proportionally to outdoor temperature decreases to maintain indoor setpoints.
        - Compare model predictions with operational data trends under similar conditions.
- Cooling Coil Logic:
    - During higher outdoor temperatures:
        - Validate increased cooling coil valve activity and airflow to maintain setpoints.
        - Ensure airflow rates align with expected heat transfer requirements.

**Fume Hood Operation Validation**

- Input Parameters:
    - Fume hood sash positions, exhaust airflow rates, and duct static pressure.
- Expected Behavior:
    - Exhaust airflow should adjust dynamically based on sash position and laboratory ventilation demands.
    - Validate that fume hood systems maintain proper negative pressure in lab spaces.
- Validation Steps:

    - Cross-check fume hood airflow data with ASHRAE 135 and AIHA Z9.5 guidelines.
    - Ensure exhaust flow meets minimum safety requirements (e.g., cubic feet per minute).

**Normal and Fault States**

- Pattern Recognition Outputs:
    - Detect operational anomalies, such as:
        - Stuck valves (e.g., heating coil valve remains closed despite low temperatures).
        - Blocked filters (indicated by increased pressure drops).
        - Compressor inefficiencies (unexpected deviations in temperature and pressure ratios).
- Validation Steps:
    - Compare model-flagged anomalies against recorded fault logs from Siemens HVAC controllers.
    - Ensure flagged faults align logically with the operational data and known mechanical failure patterns.

## F.5 Validation Procedure

**Design Validation Plan:**

- Validation ensures that the PdM system meets Siemens' operational needs and objectives.
- The final design will be presented to Siemens in **March**, with a demonstration of the prototype, experimental results, and simulation outputs.

**Stakeholder Feedback:**

- Feedback from building managers and VCU's operations team will be solicited for system usability and effectiveness.

**Key Metrics for Validation:**

1. **Predictive Quality Metrics**
    - **Accuracy:** Measures how often the model correctly predicts the outcome
    - **Matthews Correlation Coefficient (MCC):** Evaluates prediction quality across true positives, true negatives, false positives, and false negatives, particularly for imbalanced datasets.
    - **Sensitivity:** Measures the model's ability to detect failures accurately, ensuring no critical issues are overlooked.

- **Specificity:** Assesses the model's accuracy in identifying normal operations, reducing false alarms.
- **Cohen's Kappa Coefficient:** Validates agreement between predicted and actual classifications beyond random chance, ensuring robustness.

These metrics collectively ensure the model's predictions are reliable and meaningful for operational use.

2. **Timelines**:

   Timeliness ensures that actionable insights are generated at least two weeks before potential failures.

   - Advanced time-series analysis, including Long Short-Term Memory (LSTM) networks, is employed to identify trends and anomalies in operational data.
   - The system's ability to provide advance warnings will be validated by comparing predicted failure timelines against actual occurrences.
   - Meeting this metric is critical for minimizing disruptions and enabling proactive maintenance.

3. **System Integration**:

   Seamless integration with Siemens' existing infrastructure is a core requirement.

   - The PdM system will be tested for compatibility with Siemens' building control systems to ensure smooth and uninterrupted data processing.
   - Adherence to standards such as BACnet for interoperability and ISO/IEC 27001 for data security ensures robust and secure operation.
   - Validation will confirm that the system enhances Siemens' infrastructure without requiring significant modifications, maintaining operational stability.

## F6. Design Methodology Summary

This section outlines the methodologies for developing the predictive maintenance (PdM) system, focusing on computational, experimental, and validation approaches. Using Python libraries like scikit-learn and PyTorch, LSTM models are employed for time-series forecasting and anomaly detection, leveraging historical data for accuracy. Experimental methods include reserved datasets for validation and synthetic fault simulations to test robustness. The PdM architecture integrates data collection, machine learning, and visualization layers to provide actionable insights. Mechanical validation ensures HVAC components, including heating/cooling coils and fume hoods, comply with ASHRAE and AIHA standards. Validation metrics such as MCC, sensitivity, and specificity ensure the model's reliability, with insights generated at least two weeks before failures. The system integrates seamlessly with Siemens' infrastructure, ensuring scalability, security, and operational stability.

## Section G. Results and Design Details

The final predictive maintenance model is an LSTM Autoencoder that learns the normal behavior of time-series HVAC data. The LSTM encoder fits each data sequence it sees into compact memory; the decoder tries to rebuild the original data sequence. In ideal conditions, there are minimal errors when the model reconstructs normal behavior data. When the model tries to reconstruct data associated with a fault, it makes more errors. This jump in error serves as a signal that a fault has been detected. Therefore, we do not have to use a supervised (labeled) approach when training the model.

## G.1 Modeling Results

| **Iteration** | **Architectural/Data-Processing Change** | **Principal Metric (F-score) on Mixed Test Set*** | **Observation** |
|---|---|---|---|
| 0 (Baseline) | Simple autoencoder, no temporal features | F1 = 0.00 | Model predicted test points as all normal |
| 1 | LSTM autoencoder (64-hidden state encoder/decoder) | F1 ≈ 0.05 | Began to capture some fault patterns, but still biased to normal |
| 2 | Added cyclic hour/day/season encodings & appropriate feature scaling; removed faulty feature | F1 ≈ 0.42 | Recall (how often model correctly identifies fault instances from all the actual fault samples in the test set) improved; false-alarm rate increased as well |
| 3 (Final) | 5-fold cross validation to obtain validation losses, statistical threshold = $\mu + \sigma$ of val. loss, month-stratified 50% sampling (≈8000 data object input for training), 250-epoch full retrain | F1 = 0.78 | Decent balance of recall/precision on mixed data |

**Figure 6. Model Development Pipeline**

*Iterations were evaluated again the same “mixed” test set (75% faults) timestamp-wise, but feature vector differed per iteration

The final model consists of a double-layer LSTM encoder and decoder (64 hidden states each) followed by a fully connected reconstruction head. To minimize overfitting, the sequence length (lookback) is fixed at 1 (point-wise reconstruction). Larger sequences (sequence length>=6) were not able to be computed due to high computational complexity. Sequence lengths greater than 1 and less than 6 did not provide significant improvements in model performance. So, sequence length of 1 is kept to balance performance and computation time. Additionally, to reduce the chance of inputting noise into the model, weekly subsampling on a monthly basis plus 5-fold validation with dropout are used. The anomaly threshold is selected fold-wise as $\mu + \sigma$ of reconstruction loss. The overall best threshold is chosen for testing.

| **Step** | **Action Taken** | **Reason** |
|---|---|---|
| Missing Values | Missing values in the dataset were removed from the dataset by removing all rows with 1 or more missing values No interpolation or imputation was performed. | This preserved data quality by avoiding usage of synthetic data. The proportion of missing data was relatively small. |
| Feature Removal | Reheat Valve Command was dropped after audit of data showed that it duplicated Chilled Water Valve’s values even though they serve opposite functions. | This eliminated the case where two independent variables are incorrectly highly correlated, confusing the classifier. |
| Late-Stage Feature Engineering | Cyclic encodings of hour, the day of the week, and season and a Room Temp Deviation feature were engineered. | This gave the model temporal context for cyclic HVAC operations and created a signal feature for the setpoint. |
| Final Feature Vector (during training) | Seven continuous numerical, three discrete binary, and six discrete cyclic features in training. Otherwise, 13 features in the dataset csv files. | Overall, this gave more useful information to the model. |

**Figure 7. Data Preprocessing and Feature Engineering Pipeline**

| Explored Variation | Description | Outcome/Reason for Abandonment |
|---|---|---|
| Semi-Supervised LSTM Autoencoder with an SVM post-filter | Normal behavior training data used to train the autoencoder, and smaller set of labeled faults used to fit SVM | This led to poor performance metrics, showing strong overfitting to the few labelled faults. The model forgot what normal behavior looked like. |
| Supervised LSTM Classifier (more traditional machine learning algo) | 2-layer 32 hidden state LSTM sigmoid head, used Binary Cross Entropy loss, and fault-labeled data was an extreme minority class (lack of plentiful, high quality fault data) | High accuracy that hid the model's inability to distinguish between normal vs. faulty testing data in many types of test sets (mixed sets, 95% N : 5% F sets, 100% F sets). Model learned class imbalance too well. |
| Down-sampling Every Other Day | Kept one 24 hour window, skipped the next, so on and so forth | The faults were almost entirely lost because they were sparse to begin with. The model biased to all normal. |
| Skip every other 15 minute sample | Thinned the dataset by 50% | Model regressed severely, as it was completely unable to identify faults. This helped runtime but removed important fault information because testing data now had 30 minute intervals instead of 15 minute intervals. |
| Month-stratified 50% sampling (final choice) | Randomly choose 50% of data within each month, while also preserving daily/weekly cycles | This led to the best performance metrics, and also found the best anomaly thresholds for validation and testing. This also reduced training time slightly. |

**Figure 8. Model Architectures Exploration**

Important Lessons Learned

1. Severe class imbalances make fully and semi-supervised models weak at detecting test faults. Most likely, the small amount of fault-labeled training and testing data that we collected was not high quality enough to represent the diversity of real-life faults.

2. Granular, more random representation of the data beat uniform, but more sparse representation. Large time-skips and intervals between data objects created temporary blind spots where faults occurred, or where normal behavior was best represented. Intra-month random sampling with 15 minute intervals, and with HVAC time cycle patterns kept intact resulted in best model performance out of all our experiments.
3. The unsupervised LSTM Autoencoder when paired with our chosen sampling method and k-fold threshold tuning gave the largest gain in F-score, with the least labelling effort. We found that there was simply not enough fault data, and we could not develop a trustworthy fault-labeling system. HVAC systems are complex, and we did not possess enough expert knowledge to create a high-quality supervised model. Our efforts at supervised learning also returned poor model performance.

## G.2 Experimental Results

| **Test Sets*** | **Fault %** | **Accuracy** | **F-Score** | **Precision** | **Recall/Sensitivity** | **Specificity** |
|---|---|---|---|---|---|---|
| Mixed (25 N:75 F) **fcu_testing_25_75_mixed_data.csv** | ≈75% | 0.651 | 0.780 | 0.776 | 0.784 | 0.165 |
| All-Normal: **fcu_testing_normal_data.csv** | 0% | 0.960 | 0.000 | - | 0.000 | 0.960 |
| Mostly-Fault: **fcu_testing_mostly_fault_data.csv** | 98% | 0.970 | 0.985 | 0.980 | 0.990 | 0.000 |
| Mixed (95 N:5 F) **fcu_testing_95_5_mixed_data.csv** | 5% | 0.8452 | 0.000 | 0.000 | 0.000 | 0.8987 |

**Figure 9. Performance Metrics of Varying Test Sets**

*Test sets can be found on the project's Github repository

| | Predicted Normal | Predicted Fault |
|---|---|---|
| Actual Normal | 13 | 66 |
| Actual Fault | 63 | 228 |

**Figure 10. Confusion Matrix of Mixed (25 N:75 F) Test Set**

| | Predicted Normal | Predicted Fault |
|---|---|---|
| Actual Normal | 167 | 7 |
| Actual Fault | 0 | 0 |

**Figure 11. Confusion Matrix of All-Normal Test Set**

| | Predicted Normal | Predicted Fault |
|---|---|---|
| Actual Normal | 0 | 6 |
| Actual Fault | 3 | 288 |

**Figure 12. Confusion Matrix of Mostly Fault Test Set**

| | Predicted Normal | Predicted Fault |
|---|---|---|
| Actual Normal | 71 | 8 |
| Actual Fault | 5 | 0 |

**Figure 13. Confusion Matrix of Mixed (95 N:5 F) Test Set**

Key Findings

1. The model performs well in homogenous scenarios. The model correctly predicts normal behavior when it sees normal-only testing datasets. The model correctly predicts failure states when it sees fault-heavy testing datasets.
2. Model performance worsens significantly in heterogeneous scenarios where the proportion of normal to fault data is 95-5% or 25-75%. This means that the reconstruction loss when the model sees faulty testing data overlaps onto the learned normal behavior distribution. Unfortunately, this is a known limitation with 1-sequence step autoencoders. A larger sequence step, as well as more compute power, may be needed to see improved model performance in these scenarios.

## G.4 Final Design Details/Specifications

| **Item** | **Specification** |
|---|---|
| Model file | fcu_model_LSTM.py (code on Github) |
| Features Inputted to Model | 7 continuous + 3 binary + 6 cyclic encodings = 16 dimensions |
| Training Data | 2022-02 → 2023-02; 50% month-wise stratified sample |
| Sequence Length | 1 timestep (15 minutes) |
| Hidden States | 64 states (encoder & decoder) |
| Training Regimen | 5-fold cross validation (125 epochs/fold) → full retrain for 250 epochs |
| Anomaly Threshold | $\mu + \sigma$ of validation loss |
| Deliverables | Source code, saved scalar object, performance metrics CSV files, training dataset, and various testing datasets |

**Figure 14. Model Specification Table**

**The process of model development and resulting model described in this section demonstrated a significant improvement in the F-score metric over the baseline model. These results satisfy the project goal of a predictive maintenance model using HVAC data.**

## G.5 Working Prototype

### Working Prototype

The current working prototype consists of a full-stack implementation that integrates a machine learning model (backend) with a user-friendly dashboard interface (frontend). This allows users to train, test, and visualize the performance of an HVAC fault detection model.

### Backend (Machine Learning Model)

The backend is a Python-based Flask API that serves as the bridge between the ML model and the frontend. The core of the backend is an LSTM Autoencoder trained to detect anomalies in HVAC system data. Key features include:

- **Model Training**: The user can initiate model training from the frontend. The backend processes historical HVAC data, applies normalization, and trains the LSTM model using K-Fold cross-validation. The model is then saved for reuse.

- **Model Testing**: The API supports testing against multiple datasets (fault, normal, or mixed) and calculates key performance metrics such as accuracy, precision, recall, and F1-score.

- **Result Formatting**: The backend returns all test metrics as formatted strings and stores confusion matrix and metric results as CSV files.

### Frontend (Interactive Dashboard)

The frontend is built using React and Tailwind CSS. It allows users to interact directly with the model without needing to run code. Features include:

- **Train and Test Buttons**: Users can trigger model training or testing using clearly labeled buttons.

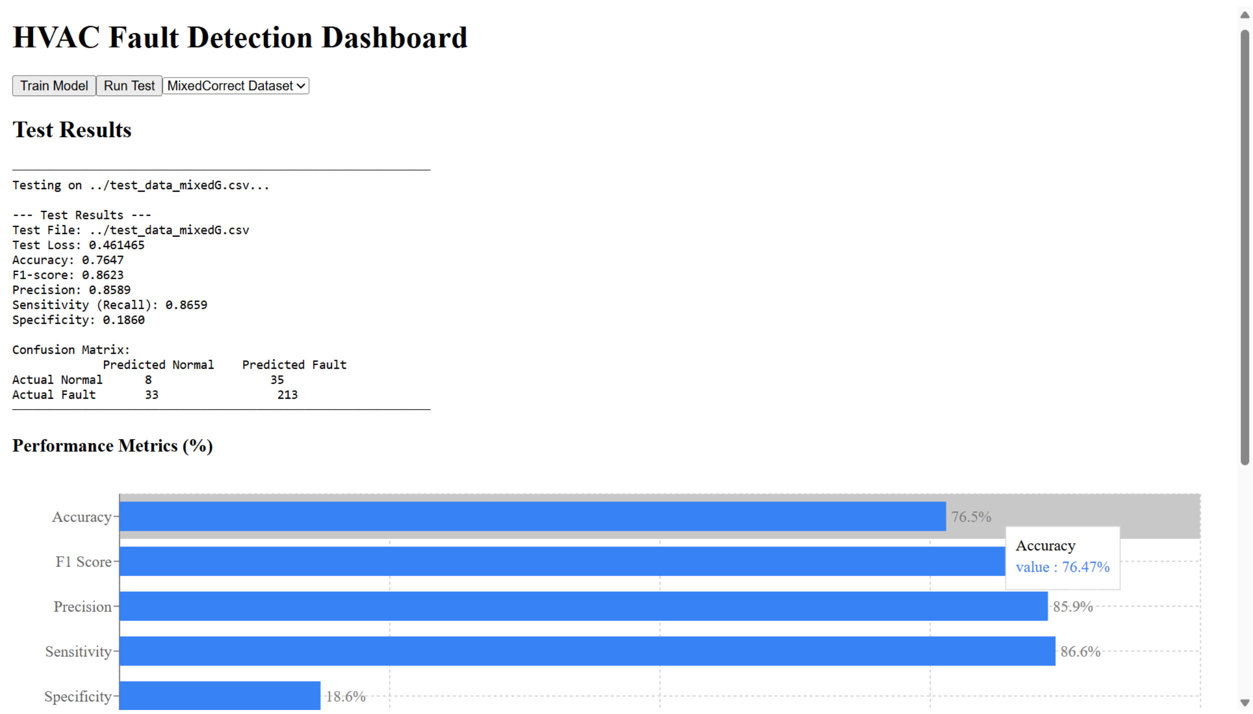
- **Progress Feedback**: During long-running operations, a dynamic progress bar gives real-time feedback on the status of the task.

- **Dataset Selection**: A dropdown menu allows users to choose between different datasets (e.g., Fault, Normal, Mixed, Both) to run tests on.

- **Performance Visualization**:

- ○ **Metric Bar Chart**: A percentage-based bar chart shows key model performance metrics.
- ○ **Stacked Confusion Matrix**: A visually intuitive stacked bar chart displays confusion matrix values as percentages, showing correct and incorrect predictions.

- **Result Console**: Below the charts, raw test results are displayed in a styled code block, giving users full transparency.




### Integration

The frontend sends user input to the Flask backend using Axios POST requests. The backend processes the request, performs model training or testing, and returns a structured response. The frontend parses the response and renders the results both textually and graphically.

This prototype demonstrates the full lifecycle of a machine learning model deployment: training, testing, inference, and user interaction. It is modular, extendable, and designed with user accessibility and interpretability in mind.

## Section H. Societal Impacts of Design

The development of the predictive maintenance (PdM) system for Siemens' smart building infrastructure extends beyond technical innovation, addressing various societal, regulatory, economic, and ethical impacts. This section evaluates the potential implications of the PdM system on public health, safety, and welfare, as well as its societal, political, economic, environmental, global, and ethical impacts.

## H.1 Public Health, Safety, and Welfare

The PdM system is designed to improve the reliability and efficiency of building operations, which directly enhances public health, safety, and welfare by:

- **Proactive Identification of Failures**: By predicting potential mechanical failures in HVAC systems, the design reduces the risk of hazardous conditions caused by poor ventilation, temperature instability, or laboratory system downtime.
- **Improved Indoor Air Quality**: Ensuring HVAC systems operate at optimal levels enhances air quality, reducing the risks of airborne contaminants in sensitive lab environments.
- **Energy Efficiency**: Optimizing energy use minimizes risks associated with overheating or overloading electrical systems.
- **Safety Standards Compliance**: The design adheres to ASHRAE 135 and AIHA Z9.5 standards, ensuring proper ventilation and the safe operation of laboratory environments.

## H.2 Societal Impacts

The PdM system offers several societal benefits, including:

- **Reduced Maintenance Disruptions**: Proactive maintenance minimizes disruptions in academic and research environments, allowing uninterrupted operations.
- **Educational Enhancement**: The project provides valuable real-world learning experiences for engineering students, fostering innovation and skill development.
- **Technology Adoption**: Encouraging the integration of advanced IoT and AI technologies into everyday building management.

## H.3 Political/Regulatory Impacts

The PdM system aligns with local, state, and federal regulatory frameworks, influencing building management practices and policies:

- **Regulatory Compliance**: Adherence to standards like NIST SP 800-53 for data security and ASHRAE 135 for interoperability ensures compliance with key regulatory bodies.
- **Policy Influence**: Demonstrating the efficacy of PdM systems may drive policy changes encouraging the adoption of predictive maintenance in public infrastructure and government buildings.

## H.4. Economic Impacts

The PdM system has significant economic implications for both Siemens and building operators:

- **Cost Reduction**: Reducing unplanned downtime and expensive repairs results in lower operating costs.
- **Market Competitiveness**: Siemens can leverage the PdM system as a value-added feature to maintain a competitive edge in the smart building market.
- **Scalability Benefits**: The design's scalability allows for cost-effective deployment across multiple subsystems, maximizing return on investment.

## H.5 Environmental Impacts

The PdM system aligns with sustainability goals by minimizing environmental impacts:

- **Energy Conservation**: Optimizing HVAC system performance reduces energy waste, decreasing the carbon footprint of buildings.
- **Material Efficiency**: Prolonging equipment lifespans reduces the need for frequent replacements, decreasing material waste.
- **Sustainability Standards**: Aligning with Project Haystack standards ensures the responsible use of IoT data for energy-efficient operations.

## H.6 Global Impacts

As part of Siemens' global smart building initiative, the PdM system has the potential to:

- **Standardize Maintenance Practices**: Facilitate global adoption of predictive maintenance systems by showcasing their efficiency and reliability.
- **Enhance Global Collaboration**: Establish Siemens as a leader in sustainable building technologies, fostering partnerships across industries and countries.

## H.7. Ethical Considerations

The PdM system raises several ethical questions, which were addressed during the design process:

- **Data Privacy**: Adherence to ISO/IEC 27001 ensures the ethical use of operational data, protecting stakeholders' privacy.
- **Accessibility**: The design is intended to be scalable and adaptable, ensuring equitable access to efficient maintenance technologies across various economic contexts.
- **Responsibility**: Ethical considerations were prioritized to ensure the design benefits both end-users and the environment, aligning with global sustainability and safety goals.

## H.8 Summary

The development of the predictive maintenance (PdM) system as part of this VCU-Siemens collaboration demonstrates significant societal, environmental, and economic impacts while addressing key regulatory and ethical considerations. By improving HVAC reliability, the system enhances public health, safety, and indoor air quality, ensuring safe and efficient operations in critical spaces like laboratories. It minimizes maintenance disruptions, fosters energy efficiency, and reduces material waste, aligning with global sustainability standards. The system's scalability and compliance with standards like ISO/IEC 27001 and ASHRAE 135 further reinforce its relevance in modern building management.

# Section I. Cost Analysis

As of this stage in the MULT 25-610 Smart Buildings-Sustainability and Efficiency project, no direct expenditures have been incurred. All programming and data analysis tasks have been conducted on team members' personal laptops using freely available software libraries such as NumPy, Pandas, TensorFlow, and PyTorch. These tools have enabled data preprocessing, model training, and simulations without requiring additional software purchases. This approach ensures the project remains cost-effective and within the $1,000 budget allocated for potential future expenses.

Looking ahead, anticipated costs may arise as the project progresses. These include cloud computing services such as AWS or Azure for large-scale data processing and machine learning model deployment, with estimated rates ranging from $0.90 to $1.50 per hour for GPU instances. Additionally, if hardware upgrades or external storage solutions are needed for extensive data sets, their costs will be evaluated and incorporated into the budget. The final project expenditures will also consider expenses related to the development and hosting of the visualization dashboard and any resources required for integration with Siemens' infrastructure. All costs will be carefully monitored and documented to ensure transparency and adherence to the project's financial constraints.

## Section J. Conclusions and Recommendations

The MULT 25-610 capstone project set out to address a key operational challenge faced by Siemens: the reliance on reactive maintenance for HVAC systems at VCU. Our team aimed to develop a predictive maintenance (PdM) model that could anticipate system failures before they occurred, reducing downtime, lowering costs, and supporting campus sustainability efforts. This project required a multidisciplinary approach, combining mechanical engineering with machine learning and data science.

We followed a structured engineering design process, beginning with stakeholder research and data exploration. One of the first challenges we encountered was the quality and consistency of the HVAC operational data. The fan coil unit (FCU) in West Hall Lab 409 produced valuable data, but much of it required cleaning and standardization. We implemented a preprocessing pipeline that included outlier removal, hourly data resampling, normalization using MinMaxScaler, and time-based feature engineering to account for patterns based on season and hour. These steps helped us prepare the data for use in a time-series model capable of identifying system anomalies.

After preparing the dataset, we developed a Long Short-Term Memory (LSTM) Autoencoder trained on normal operational behavior. This model learned to reconstruct expected system performance, and any significant deviation in reconstruction loss was treated as a potential fault. We used five-fold cross-validation to help evaluate model performance and found that the model performed well when data was clearly categorized as normal or faulty. However, we also discovered that it struggled to detect faults that were subtle or intermittent, especially when buried in otherwise normal sequences. This highlighted the need for future work on improving model sensitivity and fine-tuning anomaly thresholds. The final design consists of a fully functional predictive maintenance model built in Python using PyTorch. It features two LSTM layers with ReLU activation, a dense output layer, and a training process optimized through cross-validation and loss tracking. The model inputs include temperature readings, valve positions, and time indicators such as day and hour.

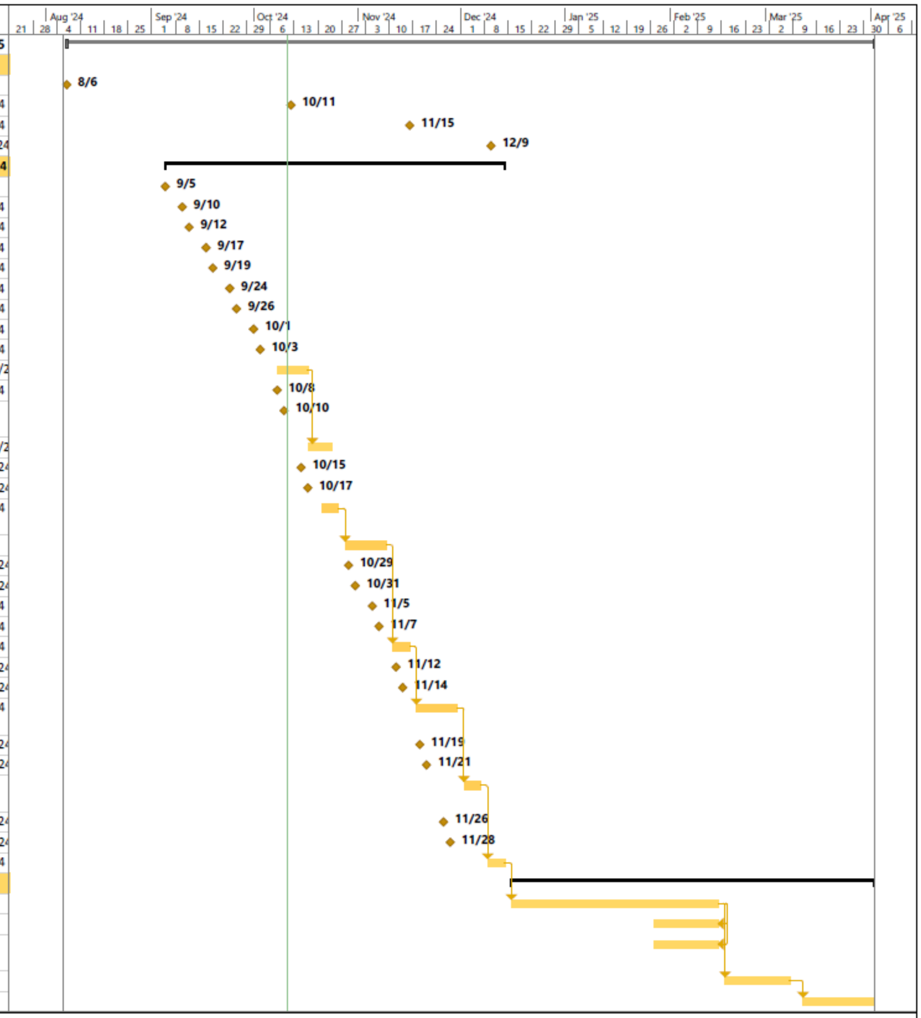
For future development, we recommend exploring adaptive thresholds that adjust based on seasonal changes in HVAC behavior. Expanding the feature set to include environmental factors like humidity and occupancy levels could further improve prediction accuracy. Additionally, incorporating a dashboard interface using platforms like Streamlit or Flask would provide real-time visualization for end users. There is also potential to explore more hybrid architectures that combine the forecasting ability of LSTM with other machine learning algorithms like GRUs.

This project lays a strong foundation for future senior design teams. We have completed the core tasks of data collection, preprocessing, model development, validation, and documentation. All code, data, and configuration files are stored in a version-controlled GitHub repository, which will be shared with Siemens and faculty advisors. For those continuing the project, we recommend starting by validating the current model on other HVAC units and extending its functionality across additional systems. Additionally, false positive rates remain high in datasets where faults are sparse; sequence-level voting, rolling-window smoothing, or a lightweight post-classifier could reduce these rates without requiring supervised learning. Adaptive thresholds that drift with seasonal load, richer contextual sensors (supply air humidity), or an attention-based encoder could potentially increase recall and specificity. Finally, moving the model to a cloud platform can allow for an increase of the sequence length, potentially improving model performance.

In conclusion, this PdM model has proven its value as a proactive tool in smart building management. With further refinement and deployment, it has the potential to scale across VCU's campus and beyond, supporting Siemens' vision of efficient, sustainable infrastructure.

# Appendix 1: Project Timeline

| ID | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 0 | **Senior Design Project** | **170 days** | **Tue 8/6/24** | **Tue 4/1/25** |
| 1 | **Milestones** | | | |
| 2 | Team Contract | 0 days | Tue 8/6/24 | Tue 8/6/24 |
| 3 | Project Proposal | 0 days | Fri 10/11/24 | Fri 10/11/24 |
| 4 | Fall Poster Due | 0 days | Fri 11/15/24 | Fri 11/15/24 |
| 5 | Preliminary Design Report Due | 0 days | Mon 12/9/24 | Mon 12/9/24 |
| 6 | **Fall Semester** | **72 days** | **Thu 9/5/24** | **Fri 12/13/24** |
| 7 | Weekly Faculty Meeting | 0 days | Thu 9/5/24 | Thu 9/5/24 |
| 8 | Weekly Student Meeting | 0 days | Tue 9/10/24 | Tue 9/10/24 |
| 9 | Siemens Navigator Training | 0 days | Thu 9/12/24 | Thu 9/12/24 |
| 10 | Weekly Student Meeting | 0 days | Tue 9/17/24 | Tue 9/17/24 |
| 11 | Weekly Faculty Meeting | 0 days | Thu 9/19/24 | Thu 9/19/24 |
| 12 | Weekly Student Meeting | 0 days | Tue 9/24/24 | Tue 9/24/24 |
| 13 | Weekly Faculty Meeting | 0 days | Thu 9/26/24 | Thu 9/26/24 |
| 14 | Weekly Student Meeting | 0 days | Tue 10/1/24 | Tue 10/1/24 |
| 15 | Meeting with VCU Maintenance | 0 days | Thu 10/3/24 | Thu 10/3/24 |
| 16 | Analyze Construction Drawings | 7 days | Tue 10/8/24 | Wed 10/16/2 |
| 17 | Weekly Student Meeting | 0 days | Tue 10/8/24 | Tue 10/8/24 |
| 18 | Meeting with Siemens Predictive Maintenance | 0 days | Thu 10/10/24 | Thu 10/10/24 |
| 19 | Analyze HVAC Manuals | 5 days | Thu 10/17/24 | Wed 10/23/2 |
| 20 | Weekly Student Meeting | 0 days | Tue 10/15/24 | Tue 10/15/24 |
| 21 | Weekly Faculty Meeting | 0 days | Thu 10/17/24 | Thu 10/17/24 |
| 22 | Initialize Data Preprocessing and Profiling | 5 days | Mon 10/21/24 | Fri 10/25/24 |
| 23 | Continue Data Preprocessing | 10 days | Mon 10/28/2 | Fri 11/8/24 |
| 24 | Weekly Student Meeting | 0 days | Tue 10/29/24 | Tue 10/29/24 |
| 25 | Weekly Faculty Meeting | 0 days | Thu 10/31/24 | Thu 10/31/24 |
| 26 | Weekly Student Meeting | 0 days | Tue 11/5/24 | Tue 11/5/24 |
| 27 | Weekly Faculty Meeting | 0 days | Thu 11/7/24 | Thu 11/7/24 |
| 28 | Validate Mehcanical Data | 5 days | Mon 11/11/2 | Fri 11/15/24 |
| 29 | Weekly Student Meeting | 0 days | Tue 11/12/24 | Tue 11/12/24 |
| 30 | Weekly Faculty Meeting | 0 days | Thu 11/14/24 | Thu 11/14/24 |
| 31 | Continue Data Preprocessing and Profiling | 10 days | Mon 11/18/24 | Fri 11/29/24 |
| 32 | Weekly Student Meeting | 0 days | Tue 11/19/24 | Tue 11/19/24 |
| 33 | Weekly Faculty Meeting | 0 days | Thu 11/21/24 | Thu 11/21/24 |
| 34 | Identify and Adress Mechanical Inconsistenceis | 5 days | Mon 12/2/24 | Fri 12/6/24 |
| 35 | Weekly Student Meeting | 0 days | Tue 11/26/24 | Tue 11/26/24 |
| 36 | Weekly Faculty Meeting | 0 days | Thu 11/28/24 | Thu 11/28/24 |
| 37 | Prepare Initial Prototype | 5 days | Mon 12/9/24 | Fri 12/13/24 |
| 38 | **Spring Semester** | **77 days** | **Mon 12/16/2** | **Tue 4/1/25** |
| 39 | Develop Initial ML Model | 45 days | Mon 12/16/2 | Fri 2/14/25 |
| 40 | Train the Model | 15 days | Mon 1/27/25 | Fri 2/14/25 |
| 41 | Ensure Model Takes into account Mechanical Wear & | 15 days | Mon 1/27/25 | Fri 2/14/25 |
| 42 | Fine Tune Model | 15 days | Mon 2/17/25 | Fri 3/7/25 |
| 43 | Deploy the Model | 15 days | Wed 3/12/25 | Tue 4/1/25 |




Page 1

## Appendix 2: Team Contract (i.e. Team Organization)

Step 1: Get to Know One Another. Gather Basic Information.

| ***Team Member Name*** | ***Strengths each member bring to the group*** | ***Other Info*** | ***Contact Info*** |
|---|---|---|---|
| Grant Forest-Collins | Analytical thinking, problem-solving, strong leadership skills, and project management experience | Experienced in mechanical engineering and project management, skilled in using software tools like AutoCAD and MATLAB, familiar with Lean Construction and OSHA safety standards. Involved in various extracurricular activities, including leadership roles in Omega Psi Phi Fraternity and The National Society of Black Engineers | forestcollgc@vcu.edu |
| Esha Sharma | Logical reasoning/problem-solving skills, research skills, strong skills/knowledge of software architecture and tools, effective at communication | Experienced in fundamental computer science, especially machine learning (pytorch, tensorflow), data science and NLP (sentiment analysis), fluent in Java, C, Python, SQL | esharma@vcu.edu |
| Daniel Gubay | Analytical thinking, problem-solving, Strong knowledge of high-level software concepts, ability to learn new coding languages | Strong in automation using tools like Ansible and Terraform. Experienced in fundamental computer science concepts. | gubaydd@vcu.edu |
| Jaime Gerardo Juarez | Creative design abilities, effective communication skills, hands-on experience with | Experienced in AutoCAD and managing construction projects. Active in SHPE, helping with leadership, networking, and community activities. | gerardojuajs@vcu.edu |

| | | | |
|---|---|---|---|
| | SolidWorks, and strong teamwork | | |

| ***Other Stakeholders*** | ***Notes*** | ***Contact Info*** |
|---|---|---|
| Daniel Cranston | CS Faculty Advisor | dcranston@vcu.edu |
| Joao Soares | MNE Faculty Advisor | jsoares@vcu.edu |
| Siemens, Kenneth Cossaboon | Company Advisors | kenneth.cossaboon@siemens.com |

Step 2: Team Culture. Clarify the Group's Purpose and Culture Goals.

| ***Culture Goals*** | ***Actions*** | ***Warning Signs*** |
|---|---|---|
| Foster open communication and respect | Encourage team members to share their ideas and feedback freely in meetings and discussions.<br><br>Use Discord/texting and in-person meetings to communicate | Team members becoming quiet or hesitant to share their thoughts<br><br>Lack of responses to emails, messages<br><br>Lack of attendance of meetings |
| Promote accountability and reliability | Set clear expectations and deadlines for tasks; follow up regularly on progress. | Missed deadlines and lack of updates on assigned tasks. |
| Encourage collaboration and teamwork | Plan regular check-ins and collaborative sessions to work together on tasks and projects | Team members working in isolation without seeking input from others. |

| | | |
|---|---|---|
| Recognize and appreciate contributions | Acknowledge individual efforts and successes in group meetings and via group communication. | Feelings of being undervalued or unnoticed within the team. |

Step 3: Time Commitments, Meeting Structure, and Communication

| ***Meeting Participants*** | ***Frequency Dates and Times / Locations*** | ***Meeting Goals Responsible Party*** |
|---|---|---|
| Students Only - Esha Sharma, Grant Forest-Collins, Daniel Gubay, Jaime Gerardo Juarez | As Needed, On Discord Voice Channel | Update group on day-to-day challenges and accomplishments (Esha will record these for the weekly progress reports and meetings with advisor) |
| Students Only | Every Tuesday at 5 pm in Engineering West Hall/ERB | Actively work on project (Esha will document these meetings by committing to Github with detailed commit messages, etc, then post on Discord and update Capstone Report) |
| Students + Faculty advisor - Esha Sharma,<br>Grant Forest-Collins, Daniel Gubay, Jaime Gerardo Juarez, Daniel Cranston, Joao Soares | Every Thursday at 10 am-11am in conference room ERB4310 | Update faculty advisor and get answers to our questions<br>(Esha will scribe) |
| Kenneth Cossaboon, Byron Burns, Students, Faculty Advisor | Monthly meetings on Thursday at 10 am in conference room ERB4310 | Update project sponsor and make sure the team is on the right track (Esha will scribe; Daniel will present prototype so far) |

Step 4: Determine Individual Roles and Responsibilities

| ***Team Member*** | ***Role(s)*** | ***Responsibilities*** |
|---|---|---|
| Esha Sharma | Group communication /Scribe | ✔ Keep a detailed record of meeting notes and share with group |

| | | |
|---|---|---|
| | | ✔ Record notes for weekly progress reports and meetings with advisor<br>✔ Write detailed commit messages for Github, etc, then post on Discord and update Capstone Report |
| Daniel Gubay | Test Engineer | ✔ oversees experimental design test plans, procedures, and data analysis<br>✔ establishes test protocols and schedules<br>✔ leads presentation of experimental findings and resulting recommendations. |
| Grant Forest-Collins | PM | Manages all tasks; develops overall schedule for project; writes agendas and runs meetings; reviews and monitors individual action items; creates an environment where team members are respected, take risks and feel safe expressing their ideas. Required: On Edusourced, under the Team tab, make sure that this student is assigned the Project Manager role. This is required so that Capstone program staff can easily identify a single contact person, especially for items like Purchasing and Receiving project supplies. |
| Jaime Gerardo Juarez | Financial Manager | ✔ Handle all team purchase requests<br>✔ Monitor and track teams budget<br>✔ Ensure spending stays within allowed amount<br>✔ Assist in decision making for new acquisition<br>✔ Support team as needed |

Step 5: Agree to the above team contract

*Team Member: Esha Sharma* *Signature: Esha Sharma*

*Team Member: Daniel Gubay* *Signature: Daniel Gubay*

*Team Member: Grant Forest-Collins* *Signature: Grant Forest-Collins*

*Team Member: Jaime Gerardo Juarez* *Signature: Jaime S Gerardo Juarez*

## References

[1] Gulvin, L. (2024, August). *Predictive vs preventive maintenance*. ibm.com. https://www.ibm.com/think/topics/predictive-vs-preventive-maintenance

[2] Puccio, K. (2020, May 26). *HVAC for Laboratories: Ventilation, Lab Types & Equipment - Therma*. Therma. https://www.therma.com/hvac-for-laboratories/

[3] Achouch, M.; Dimitrova, M.; Ziane, K.; Sattarpanah Karganroudi, S.; Dhouib, R.; Ibrahim, H.; Adda, M. On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. Appl. Sci. 2022, 12, 8081. https://doi.org/10.3390/app12168081

[4] Niima Es-sakali.; Moha Cherkaoui.; Mohamed Oualid Mghazli.; Naimi, Z. Review of predictive maintenance algorithms applied to HVAC systems. Energy Reports 2022, 8, 1003–1012. https://doi.org/10.1016/j.egyr.2022.07.130

[5] Kang, Z.; Catal, C.; Tekinerdogan, B. Remaining Useful Life (RUL) Prediction of Equipment in Production Lines Using Artificial Neural Networks. Sensors 2021, 21, 932. https://doi.org/10.3390/s21030932

[6] Chesterman, X.; Verstraeten, T.; Pieter-Jan Daems.; Nowé, A.; Helsen, J. Overview of normal behavior modeling approaches for SCADA-based wind turbine condition monitoring demonstrated on data from operational wind farms. Wind Energy Science 2023, 8(6), 893–924. https://doi.org/10.5194/wes-8-893-2023