Interim Design Report

Lucas Alba, Simon Vellandurai, Jorge Rodriguez-Arraiz

The Catholic University of America

School of Engineering

ENGR 441 - 01

Contact Info:

Lucas Alba: (949)-331-667

Simon Vellandurai: (301)-944-4280

Jorge Rodriguez-Arraiz: (240)-713-9186

Introduction

The goal of this design project is to develop a renewable energy production forecasting system that uses real-time weather data and historical energy output data to predict the amount of energy that can be generated by solar and wind farms. By integrating these two datasets, the system would provide actionable insights for energy producers, enabling them to optimize their operations, forecast power output more accurately, and better manage energy supply based on environmental conditions.

The purpose of this project is to explore different data processing and machine learning techniques and their application to renewable energy production. We hope to design a system that will enhance decision-making for renewable energy operators and improve energy efficiency.

Information Gathering

When gathering information about our design problem, we researched four main topics. Such as benchmarking, patent searches, interviewing stakeholders, and literature reviews. Each of these topics offered a unique experience and view to help us develop an idea and solution to solve our project goal.

We began by benchmarking existing renewable energy forecasting tools and methods to understand the current landscape and identify gaps in available solutions. Through this, we found tools such as AWS's Forecasting API, Google DeepMind's work on energy prediction, and SolarEdge's energy prediction systems. However, we noticed that most of these solutions focus on highly specialized energy production models for either wind or solar but lack integrated

models that combine both data sources with flexible, real-time weather inputs, which led us to develop a more versatile, integrated solution.

The next step we took within our information gathering was patent searches; for this portion of the project, we conducted a patent search for renewable energy forecasting systems. However, We found no patents that covered a renewable energy forecasting model that considers weather and historical production data. Most patents in the field cover infrastructure or production component efficiency. From this, we believe that a project like this would be a great help to a broad range of people such as farmers, people who are interested in solar panels, environmentalists, or even help predict natural disasters.

While interviewing the stakeholders Two Six Technologies, it became clear that they were interested in our topic idea and project. While they have not told us what they expect from our project, they have offered different ideas for taking our project. Thus, they have pushed us to think about how this project can be applied to academia and the corporate world.

Finally, we did a literature review, where we reviewed a large number of academic papers and articles on renewable energy production forecasting and also different machine learning models that would work with real-time data handling. From this, we learned that Regression-based models and machine learning models (like Random Forests and Neural Networks) have been successfully applied in this context but typically focus on single energy sources. Thus, we plan to investigate which model would work best with our datasets when implementing them in our project. Additionally, when reviewing the articles, it became clear that our project is unique because the majority of the other papers only forecast using one dataset, which offers niche results because they are not predicting all renewable energy.

Concept Generation

Following brainstorming, we narrowed our focus to four potential ideas to satisfy our project's requirements. Each solution leverages different datasets and machine learning models for renewable energy production forecasting, Concept 1: Centralized Data Integration Model (renewable energy production forecasting system), Concept 2: Modular Machine Learning Model, Concept 3: Geospatial Optimization Tool and Concept 4: AI-Driven Decision Support System. Each of which has their pros and cons which will be discussed more in detail during the concept selection, however this section will offer an overview of each concept and a diagram representing the design.

The first concept we came up with is a Centralized Data Integration Model where this concept focuses on a centralized database that plans to integrate historical energy production data from solar and wind farms with real-time weather data collected via APIs. The system will collect/ pull the data from the API, where it will then be preprocessed such as cleaning the data and normalizing it. Once the data is normalized the model will be trained based on the data, this is where we will test a large array of different models to determine the best one, however from our research it has become clear that the random forest model should fit a situation like ours the best. The predicted data will then be stored in the database where it will then be displayed through a website, through graphs and charts, which can be seen Fig. 1 as it represents the whole pipeline and process for this concept.

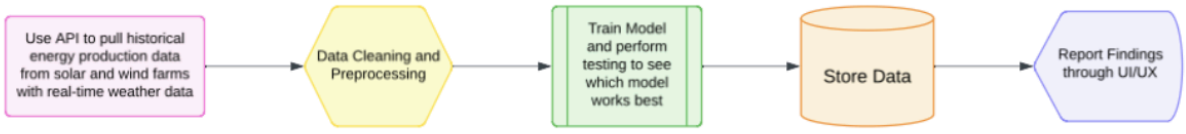

Figure 1. Centralized Data Integration Model

The second concept we worked on was the Modular Machine Learning Model, which involved creating a modular machine learning model for individual energy sources, such as historical energy production data for solar and wind and real-time weather data. Similar to the last concept, however, the models will be trained separately, so one model will predict historical energy production data and the other will use real-time weather data. Going through this pipeline, as seen in Fig 2, we will first grab the data from the two APIs, which then will undergo data cleaning and preprocessing, and then the models will be trained separately; after the model makes predictions, it will be stored in a database, and the findings will be reported through charts and graphs.

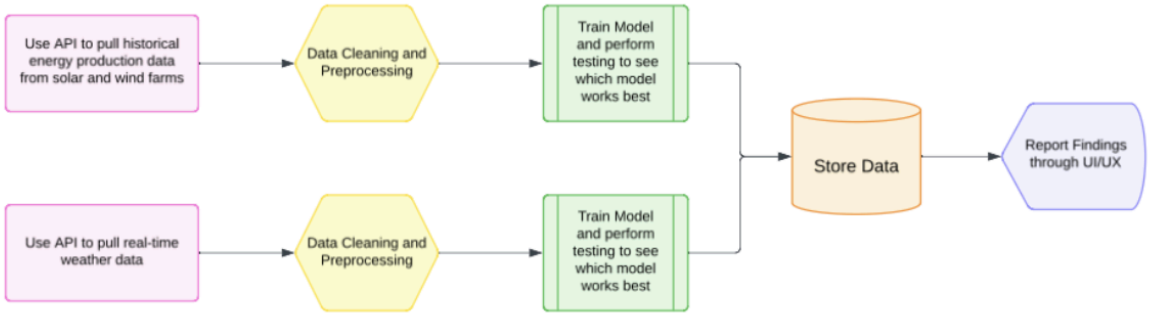

Figure 2. Modular Machine Learning Model

The third concept we generated was a geospatial optimization tool, which includes geospatial mapping of renewable energy sources. It helps predict energy generation by mapping wind speed and solar intensity to specific geographic locations, as seen in Fig 3. This concept will take data from one API, the current wind speed and solar data, and then clean that data and plot it on the geospatial map. It will then train a model on the data that was just pulled and create predictions on that data, which would then be plotted on the geospatial map. The resulting map will be displayed on an online website where users can see which areas will produce the most

energy in the coming years.

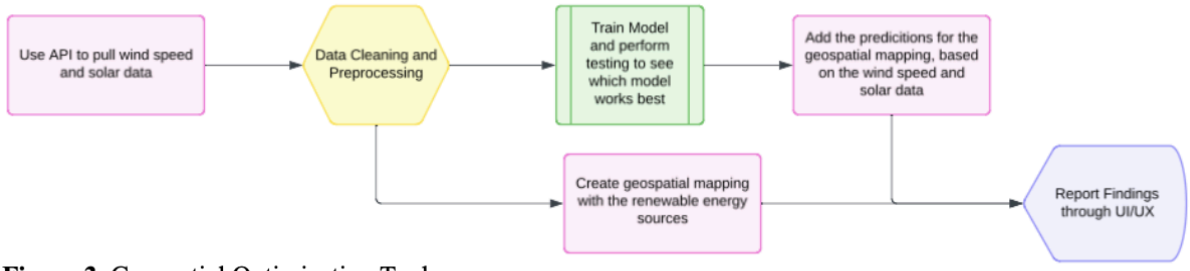

Figure 3. Geospatial Optimization Tool

The last concept we came up with was an AI-Driven Decision Support System. This idea leverages advanced AI techniques, such as neural networks known as deep learning. Which will offer predictions and actionable recommendations. It will pull current solar and wind data from an API, which is when the deep learning model will combine forecasting with prescriptive analytics and offer actionable recommendations to operators. Once this is completed, it will check the parameters for decision-making; if not all the parameters are met, it will go through this process again, gathering more data. If the parameters are met, it will proceed with making informed decisions and report the findings through UI/UX, which can be seen in Fig 4.

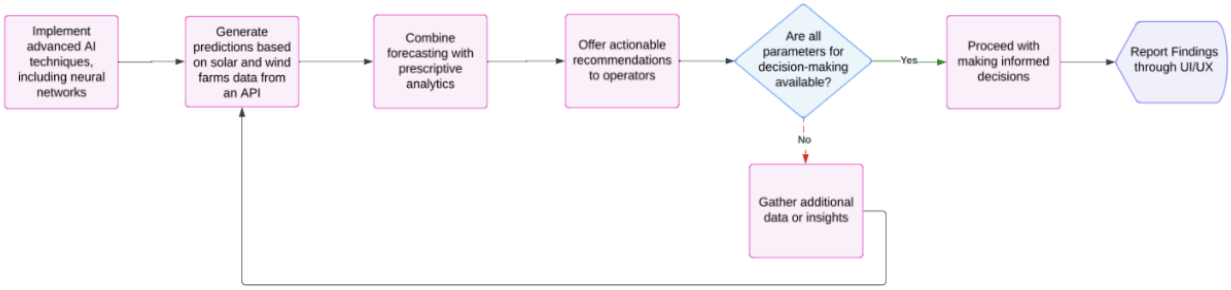

Figure 4. AI-Driven Decision Support System

Concept Selection

Once we developed our concepts and ideas for different ways we would like to take our project. We had to decide which idea would best fit our topic. To do this we decided to create a concept selection matrix. Which is represented in Fig. 5, we are comparing the topics that we previously created to concept 1, the Centralized Data Integration Model. Each of the concepts will be compared to concept 1 on the following topics: Scalability, Prediction accuracy, Reliability, Computational efficiency, User friendliness, Cost and Ease of Implementation. Looking at Fig. 5 we can break down the grading for each of the concepts based on the previously stated criteria. For clarity, each concept is labeled as follows:

- **Concept 1:** Centralized Data Integration Model
- **Concept 2:** Modular Machine Learning Model
- **Concept 3:** Geospatial Optimization Tool
- **Concept 4:** AI-Driven Decision Support System

Criteria		Concepts		
	Centralized Data Integration Model	Modular Machine Learning Model	Geospatial Optimization Tool	AI-Driven Decision Support System
Scalability		S	-	-
Prediction accuracy		-	-	+
Reliability		S	S	S
Computational efficiency		S	+	-
User friendliness		S	+	-
Cost		S	S	+
Ease of Implementation		-	+	-
S = same as the datum		+ = better than the datum		- = worse than the datum

Figure 5. Concept Selection Matrix

Let's compare Concept 1 to Concept 2. First the scalability, which we determined to be the same, because these concepts are similar, changing the API's can result in new projects which allows the projects to be flexible for future use and changes. The prediction accuracy is less within the Concept 2 because the model is trained separately on the two API's resulting in less accurate responses because a model trained on both has more accurate data to use. Reliability is the same because they offer the same results and outputs. Computer efficiency is the same because the model is only grabbing data from two datasets, and it will train two models off the same amount of data. User friendliness is also the same because they both will be displayed to a UI/UX web page for easy user access. Ease of implementation is easier to implement within concept 1 because creating one model rather than two is easier because there is less testing that has to be done.

Next let's compare concept 3 and concept 1, starting with scalability it is not as easy to scale concept 3 because the output will be set to a geospatial map. The prediction accuracy will not be as good as concept 1 because it is only using one dataset to make the predictions. Once again the reliability is the same between both of these concepts. Computer efficiency is better regarding concept 3 because it is only storing and handling data from one API rather than from two APIs. User friendliness should be more straightforward within concept 3 because it will display a map of the USA with certain locations highlighted that are projected to produce more energy. The cost for both concepts should also be the same. Ease of implementation should be easier because we would only be making predictions on one dataset and only using one machine learning model.

Lastly, let's compare concept 4 and concept 1, the scalability within concept 4 is limited because when working with a deep learning model, the whole model needs to be retrained on the data. Moreover, a deep learning model requires a lot of data to be trained on, so it will be challenging to change the topic for the model. The prediction accuracy for concept 4 would be higher because it would be able to make predictions because deep learning excels at making predictions with non-linear data, while machine learning models tend to look more for patterns within manually engineered features. The reliability is also the same within both of these concepts. Computational efficiency would be lower within concept 4 because the deep learning model involves many parameters and requires extensive matrix operations which result in heavy computations. The user friendliness would also be a bit lower as the predictions and results would not be displayed as straightforward compared to concept 1. The cost of concept 4 would also be a lot more because deep neural networks usually require high-performance hardware like GPUs, not to mention the vast amount of data needed to train the model. The ease of implementation would also be a lot harder because deep neural networks usually require specialized expertise to create, making a project like this would be challenging.

Based on the comparisons that were made, we decided to pursue concept 1, for the following reasons. Between concept 4 and concept 1 it was clear that concept 4 was not a practical project due to the time constraints we have and the funding. Even though concept 4 would offer accurate predictions. Moreover, while concept 3 offers better computational efficiency, user friendliness and ease of implementation however, as our project is focused on real-time decision making the most important factor for us is prediction accuracy, thus concept 3 is lacking in because it is only handling one dataset. Finally between concept 1 and 2, which are

very similar except that concept 1 surpasses concept 2 in ease of implementation and prediction accuracy. Which led us to our final decision of pursuing concept 1.

Embodiment Design

As previously stated our decided product is focusing on a centralized database (renewable energy production forecasting system) that will integrate energy production data from solar and wind farms with real-time weather data collected via APIs which can be seen through Fig. 1. It's important to reiterate this notion because of the machine learning aspect of our project and its benefits in the scope of a school project and as an asset in companies such as Two Six Technologies. To further build on this, we need to establish the product/system architecture to make sure it aligns with sponsors and clients alike.

To preface, this year's senior design class took a new approach becoming "Interdisciplinary Senior Design" which implied engineering students at The Catholic University of America would be joining forces in groups to bring their specialized areas of expertise into one big group collaboration. With this in mind our project is even more unique than it would've been in previous years since it came at the last minute and the group was specifically designed to achieve this goal. In addition to this rare situation, all the members of this group are computer science majors, removing the interdisciplinary aspect of this. We only bring this to you because our project doesn't have any dimensions or materials. The most important part of our project is the machine learning model and how it processed the historical data along with the real time unstructured data.

We have not entirely decided on what model to use, as this will be confirmed when we are handling the real data. But as of now based on our research we can assume that the best model will be the random forest model. Thus, we can explain how machine learning such as the rain forest model would work. The model is first initialized where the developer will define the parameters of the model such as the number of trees or the max depth of the model. Next the model will build the different decision trees, where bootstrap sampling generates multiple samples of the training data. The trees are then constructed where subsets of the features are randomly selected at each fold, creating folding criteria such as gini impurity or entropy. Then the trees are grown until they stop meeting the criteria. After the decision trees are created the model will aggregate all of them and combine the predictions from all of the decision trees to create the random forest. Thus, the model is ready to make predictions, each decision tree will give a prediction and the most popular prediction is what is outputted from the random forest, to better visualize the process Fig. 6 illustrates the pipeline.

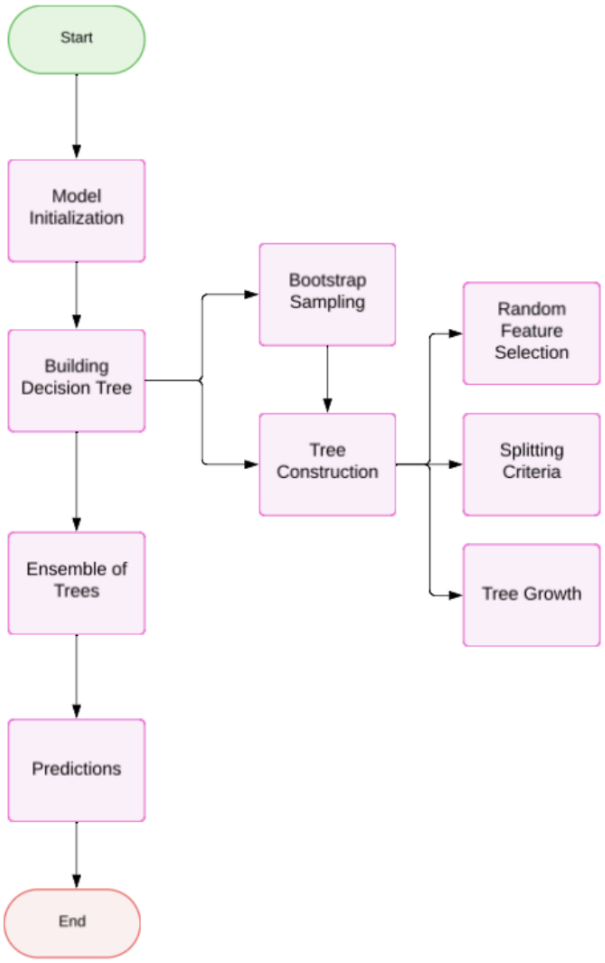

Figure 6. Random Forest Model

FMEA

While we've been hard at work improving our product, we were assigned a task which provided us the opportunity to work on an assignment which helped us identify a large variety of risks and potential failure areas that would prevent our project from reaching its potential. We did this through what is abbreviated to FMEA. FMEA stands for Failure Mode and Effects Analysis and is a systematic process that helps identify potential failures in a product. This was crucial to the development of our project as it allowed us to realize potential failures we previously wouldn't have considered feasible or considered at all. Our original analysis allowed us to identify potential issues in Data Ingestion from APIs, Data Preprocessing and Cleaning, Model Training with Random Forest, Real-Prediction and Forecasting, and Data Storage and Management.

Data Ingestion from APIs was identified as a significant issue because it can greatly impact our system by providing missing/outdated data which could damage our prediction accuracy. This issue can stem from a large number of places. Examples include network issues, server outages, changes in API versions, or modifications to the data format provided by the API. We have begun to tackle this issue in our product through a variety of enhancements that will ensure our prediction accuracy is stable and efficient. The first implementation was a robust error-handling mechanism which includes a retry method focused on failed API calls, allowing the system to reconnect a specified number of times before an alert. In addition to this we've integrated a fallback data source, allowing our system to run despite the main APIs being down. These data checks will allow us to handle unexpected format changes before the data is

processed. These changes have greatly enhanced our product's ability to handle this potential issue.

As for the primary asset of our project, Real-Time Prediction and Forecasting, there is a failure of considerate seriousness that needed to be addressed. There is a possibility that there is a delay with the actual predictions. This failure can be caused by high data volumes or changes within the environment, which, in the end, can cause model drift. This failure can affect a broad range of things, but most notably, the system's accuracy, reliability, and outdated forecasts. These all can negatively affect the decisions based on these predictions. The causes include an outdated model that is not adapted to the new seasonal shifts and increased data processing loads, which, in the end, can slow down response time. As a result of this we modified the design to support adaptive model retraining while also monitoring the environment. This has allowed our model to periodically be retrained on the latest data to stay relevant.

Data preprocessing and Cleaning was identified as our most dire potential failure because our design requires us to have load balancing and optimized data processing pipelines to handle large volumes of data without impacting the prediction speed. As a result of this we've implemented real-time performance monitoring. This monitoring has been a significant enhancement because it can alert us in case the model's accuracy or response times decrease. These changes have ensured that the likelihood and the severity of the impact on the Integrated Solar and Wind Forecasting System is less significant.

Data Storage and Management is crucial because the function focuses on potential failure modes such as data loss and corruption, which could lead to losing very important historical data, interfering with our models' retraining and analysis. This would impact our system's ability to

produce accurate forecast predictions due to the absence of reliable historical data. This is also detrimental because it impacts our system's learning over time. Some causes of the failures may include a lack of regular backups, software bugs leading to database corruption, and inadequate storage solutions that cannot handle our data volume. To counter this issue we implemented a robust data management strategy with regular and automated backup procedures. This has ensured the data is safely stored in multiple locations to avoid complete data loss. From here, we have worked on the integration of regular data checks to verify our stored data and counter any corruption issues. Employing a scalable storage solution that can handle growing data volumes. Through these implementations, we have enhanced the reliability of the data storage and management system.

DFX

A significant part for our group's project has to do with our project's design planning. This is especially important considering we're dedicating an entire semester to it. The importance of this stage can be exemplified with our topic. The potential variety of our audience leaves room for design issues that must be addressed. Fortunately, thanks to our *Design for X Paper* assignment we gained access to a great variety of design tasks that we may not have previously considered. We wrote about three design topics: design for reliability, assembly, and ergonomics.

Design for Reliability

A major component of our design relates to its reliability not just in terms of its prediction accuracy, but also in the real time decision making area where we would like our

product to provide answers at a moments notice in relation to potentially newly processed data as a result of its machine learning algorithms. This process is reliant on several components which are subject to wear, failure, or degradation over time. To ensure that our system remains reliable on a day-to-day basis, Lucas identified three main components that seem most prone to fail. From here he diligently studied, analyzed, understood the factors that contributed to its vulnerability and suggested design modifications to mitigate these risks which were promptly implemented. The three components we determined were API Connections, Real-Time Processing with Load Handling, and issues with the Machine Learning Model components.

As for API Connections we identified that this section was most likely to be the first to face failures as a result of the APIs being an external service. As a result of this, there is the chance for outages or slower production. In addition to this, they face constant development and change, which causes software that references them to be outdated. This would result in outdated API connections which is a significant issue because this is our main source of data. The solution to this task came from our research regarding different methods to counteract API's development changes, thus developing a data abstraction layer, which is essentially a method that standardizes and pre-processes the incoming data and can generalize any change that might occur to the API.

Regarding the Real-time processing/load handling, we believe that this would be the next component to fail. We came to this conclusion while researching latency or crashes during high data loads (a quite common occurrence) which could cause failure due to large volumes of data being processed simultaneously for predictions that can overwhelm system resources, particularly during peak tunes of extreme weather events. As a result of the different types of data and different amounts, I believe this will be one of the first components to fail because the

system may not be prepared for the amount of incoming data. To resolve this predicament we decided to use load-balancing algorithms to spread the processing tasks across multiple servers during high-demand periods, in addition to introducing a caching for commonly accessed data to reduce computation time. Through the combination of these tools we've discovered a solution to greatly reduce the possibility of this failure.

The final component of this trio is the Model Drift. This debacle has to do with the reduced prediction accuracy over time caused by the environment's constant evolution. An example of this can be seen in the scenario where historical data is used for training our model, with this in mind this historical data may no longer reflect current patterns which could cause predictions to become less reliable. This instance is once again dealing with the issue of constantly changing data which could result in failure. We've pursued an answer to this through our research which pointed us in the direction of two potential solutions. The first solution would see us implementing a continuous learning pipeline that retrains the model periodically using the latest data. The alternative solution would see us using a drift detection algorithm to identify when the model's accuracy is starting to decline and trigger retraining automatically. As it currently stands we have pursued the first solution.

Design for Assembly

Major aspects of our design directly reflect the findings from the Design for Assembly (DFA) study, specifically the proposed system's modularity to optimize testing and integration. For example, the use of standard APIs for weather data and popular machine learning libraries in our design ensures performance and compatibility across systems, and minimizes the need for custom code, aligning with the DFA principle of standardizing components. By implementing a

modular architecture that separates data preprocessing, model training, and the user interface, interdependencies between components are reduced and the processes of refinement and maintenance are simplified. Additionally, we address DFA recommendations to minimize adjustments and enhance assembly efficiency by automating the data handling processes, such as cleaning and preprocessing, and by incorporating containerization for deploying our model we also adhere to DFA principles for ease of scalability. These design choices collectively improve system reliability, reduce complexity, and ensure a streamlined workflow for deployment and scaling.

Design for Ergonomics

For the design for ergonomics section the instructions helped guide our design plan. For this section we designed a human factors evaluation on an aspect of our design. We further executed this design plan by gaining information which allowed us to identify specific physical or cognitive human factors under consideration and managed to relate them to our product. From these instructions we also gained valuable real life experience through the opportunity which allowed us to practice evaluating human factors and to analyze the results.

We decided to separate our topic into categories with subsections to highlight our priorities and how we tackled these issues. Regarding the human factors considerations we broke it down into two sections, the primary load and the experience/usability before discussing our evaluation, results, difficulties, and necessary changes to our design.

The primary focus of this design section is to expand our knowledge of our product's users and their potential strengths or weaknesses. An important aspect to discuss is who exactly would be using our product. When we first decided to tackle this aspect of our real time decision

making through unstructured data, we believed this would be most beneficial for farmers and people using renewable energy sources such as solar panels to see if they should invest their resources based on real time and historical data. However, on poster day a civil engineering professor who has his own company approached us and discussed our project. His excitement was apparent as a result of how his company could benefit from this through its ability to predict fires. We only bring this information to your attention because this instance along with our research has given us a greater understanding of the variety of our audience which greatly influences how we'll portray our analytical results through symbols and other assets.

The user's ability to use our product is a very serious concern. We discussed that we have a general audience but depending on our user's relative situations, our product could be rendered useless as a result of some of our users using devices that simply aren't strong enough to use this asset. This will be an area of heavy focus during the front end development section of this project.

We recently concluded that we have a statistically probable audience (individuals who are dependent on the weather such as farmers, solar panel owners and more) but also acknowledged the potential for an audience outside of this field. This will require extra UX design to ensure that our audience consisting of a large variety of backgrounds would all be able to interpret our data and symbols. To figure out the recognition of the potential audiences, Jorge gathered a group of people at school, specifically Melany Martinez who lives on a farm and could find great use in this project. As part of the process, Jorge had the group engage in a variety of tasks. The group members took turns monitoring live forecast data to establish their understanding of

symbols/graphs and opinions on what they would do moving forward and interpreting hybrid predictions of solar and wind outputs similar to the ones our project will be dissecting.

The feedback was very diverse and informative allowing us to get better insight on how to design for people from different backgrounds and how symbolism could help everyone involved along with how our project can be more direct about its interpretation of all the statistics.

Our group gained a better understanding for focused audiences and less likely audiences which is a very big discovery that will greatly impact the front end development aspect of our project and enhance the user experience providing a great experience from the technical standpoint and the user experience. In addition to this, the group's ability to interpret historical trends was very insightful to our design and even informative to the people participating in the "exam." A lot of them reported learning a lot from simply looking at historical data despite some of them needing a breakdown of some of the graphs.

While the study provided us with some insightful information we did acknowledge that the sample group could be running on the smaller end which could potentially cause inaccurate data so this is definitely something we will continuously redo throughout the development process.

As a result of our study we've come to a great amount of changes that need to be implemented. This study provided us with a ton of input especially regarding the frontend aspect. The first necessary implementation has to do with simplifying the visual components. We will implement simplified symbolism so that audiences of all experience can interpret our provided

data and conclusion. These sections will have interactive annotations allowing people to get further understanding into more complex matters to better process and understand the data. We have also come to the conclusion that the current concept of toggling between historical and real time data to see the difference and historical alignment seems to have room for the user to get tired. We are currently working on a manner to seamlessly have them together in unison or to make the toggle feature a lot easier.

Test planning

To ensure the reliability and accuracy of our forecasting system, testing will focus on the model's predictive performance and its robustness under varying conditions. The testing process will begin with a thorough evaluation of the model using historical data, split into training, validation, and testing sets. We will use standard evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 Score to quantify the model's accuracy. This step will also help identify any underfitting or overfitting issues during initial deployment. To simulate real-world conditions, cross-validation techniques (e.g., k-fold validation) will be used to ensure the model generalizes well across different datasets.

Stress testing will be conducted to evaluate the model's performance under extreme weather scenarios, such as unusually high wind speeds or prolonged cloud cover, by incorporating artificial data that represent such conditions. Doing so will help determine the model's ability to handle outliers in the data or edge cases without a significant drop in predictive accuracy. For assessing real-time functionality, we will compare the model's predictions against live weather data collected over several weeks, analyzing the alignment between predicted and observed energy outputs. Latency tests will ensure the system processes real-time data inputs

without delays that could impact usability. All testing results will be documented to refine the model's hyperparameters, optimize performance, and improve integration with the system's overall architecture. This iterative process ensures that the model meets both technical and user-centric requirements before deployment.

Finally, we will hold a feedback session to hear our sponsor's comments on the system's performance. This session will help identify shortcomings or areas for refinement, and any additional features that need to be incorporated as we move to deployment.

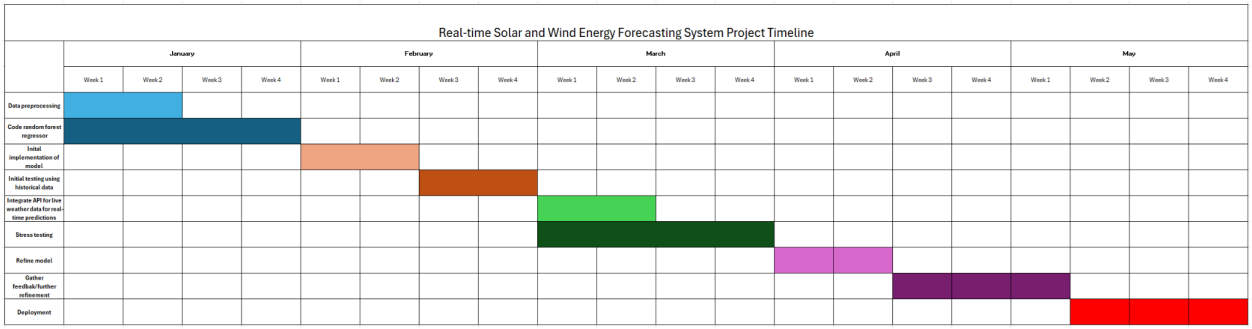

Figure 7: Gantt Chart for Jan 2025-May 2025

Cost estimate

This project is a small scale software development job, so there are no expensive materials or components needed for design and assembly. The largest sources of cost will come from data acquisition and leasing cloud computing services, namely API access to real-time weather and historical energy data, and a subscription to either AWS or Google Cloud. Services like OpenWeatherMap and NOAA offer access to their data through monthly premiums, ranging from $50-$100 a month depending on the amount of data needed. Historical energy data will likely need to be purchased as a set from third-party providers with the price depending on the characteristics of the different datasets. Through exploration of different providers online, we

have estimated this cost to be around $200. Our estimated usage of a cloud computing service for storing and processing our data is 50-100 compute hours monthly during integration and training, which can total anywhere from $250-$500. Machine learning frameworks required for building, training, testing, and evaluating the model are open-source (free), and the integrated development environment we plan to use for designing the user interface has sufficient built-in functionality at no cost. As we are unsure at the moment whether our system is going to be deployed and available for use, the cost of hosting the system online is not an expense we are anticipating to have. If this happens, we expect to pay anywhere from $100-$500 monthly for hosting the system on some platform online from which renewable energy operators can login and access the forecasting model.

Appendix

Product:	Real-time Decision making with unstructured data.								
Team:	Team 09								
			1 = Low - 5 = Severe		1 = Low - 5 = Severe		1 = Low - 5 = Severe		
Function	Potential Failure Mode	Potential Effects of Failure	Severity	Potential Causes	Occurence (O)	Process Controls	Dectection (D)	Risk Priority Number	Mitigation
Data Ingestion from APIs	API downtime or unavailability / Data inconsistency or unexpected format changes	Missing or outdated data leads to inaccurate predictions.	3	Network issues or server outages, API version changes	3	Monitoring API status, detect format changes	4	36	Retry mechanism, alternative data sources
Data Preprocessing and Cleaning	Outliers or inconsistent data ranges / data type mismatches / missing values in the dataset	Incomplete data, reducing model accuracy	4	API data issues, poor data quality, incorrect parsing, data quality issues	4	Data completeness checks before feeding into the model, type validation checks, outlier detections	3	48	Use imputation for missing values, ensure data consistency
Model Training with Random Forest	Overfitting or underfitting the model / inadequate model performance on new data	Poor predictive performance, model accuracy declines over time	2	Imbalanced data or model drift	3	Cross-validation, use of regularization techniques, regular model evaluation and retraining	3	18	Adjust model periodically, regular evaluation
Real-Predicton and Forecasting	Delayed or incorrect predictions	Loss of reliability in forecast accuracy	3	Outdated model due to environmental changes, high volume of data	4	Implement load balancing, regular performance monitoring	3	36	Model updates to reflect changing conditions
Data Storage and Managment	Data loss or Corruption in storage	Loss of valuable data for model retraining and analysis	3	Lack of backups, data corruption	2	Regular backups, data integrity checks	4	24	Redundant storage systems, scheduled data integrity checks

References

[1]J. A. Stephenson and K. M. Wallace, "Design for Reliability for Mechanisms," *Springer eBooks*, pp. 245–267, Jan. 1996, doi: https://doi.org/10.1007/978-94-011-3985-4_13.