Technological Feasibility

3/20/2025

WillowWatt

Capstone - WillowWatt

Team Members:
Ayla Tudor
Elliott Kinsley
Luke Bowen

Sponsor - OP Ravi, Director of Energy Transformation, Willow Inc.
Mentor - Jeevana Swaroop Kalapala

Table of Contents:

1 Introduction:

Buildings consume 76% of all electricity in the United States and contribute 40% of total CO_2 emissions. With the growing urgency of climate change and rising energy costs, the built world remains one of the most significant yet under-optimized areas in the global effort towards environmental sustainability. Addressing this problem is crucial not only to reduce environmental impact, but also to improve energy usage and its corresponding economic presence. Currently, most buildings rely on energy management and operational systems such as static schedules, manual adjustments, and reactive maintenance. This approach is outdated and unsustainable with the ever-present need to reduce energy consumption and costs. Existing systems typically focus on a single objective, such as adjusting lighting schedules or reducing HVAC consumption, without considering the broader optimization opportunities that exist by utilizing all of a building's assets.

Willow, a company focused on sustainable building management, is pioneering efforts to tackle this challenge. Willow is actively monitoring energy usage across multiple buildings on Northern Arizona University's (NAU) campus, collecting and analyzing data to uncover inefficiencies and potential optimizations. However, despite this advanced data collection, there remains an opportunity to elevate the system's capabilities by transforming the collected data into actionable optimization efforts. We aim to create a solution that dynamically optimizes energy usage across hundreds of facilities and assets by leveraging Willow's cutting-edge digital twin technology and real-time data.

Our solution, WillowWatt, will complement and extend Willow's existing efforts. We are developing an AI/ML-powered optimization platform designed to intelligently manage building operations, reduce energy consumption, and enhance sustainability outcomes without sacrificing comfort or performance. WillowWatt will integrate Willow's data from various building assets, creating a dynamic optimization model that continuously adapts to environmental, operational, and economic factors. With our solution, buildings can forecast energy usage based on historical data, intelligently shift energy loads, and reduce reliance on high-cost, high-emission power during peak times. Our goal is to help NAU cut energy costs, lower emissions, and meet its carbon-neutral goal by 2030, all while making campus buildings run smarter. We will be working with the Director of Energy, OP Ravi, to achieve this goal. OP Ravi has been very helpful and communicative in helping the team to visualize and plan our project. He is very knowledgeable when it comes to energy efficiency and how Willow operates.

1.1 Segue:

Now that we have introduced the main issue that Willow deals with, and how our group WillowWatt will tackle that challenge, we are ready to dive into a deeper analysis of both the specific challenges as well as our potential solutions. Next we will discuss the main challenges our project will face, and an analysis of how we plan to address the main challenge of the project. In the beginning stages of our project, there are still many moving parts and variables that have not been set in stone yet, so we are only focused on outlining the main issues and solutions at a low level, as opposed to a very detailed view. In this technological feasibility document we will begin by outlining the main technological challenges we anticipate facing in this project; in the next section, we will perform a technological analysis of the main issue we anticipate facing with our project. Within this technological analysis we will take a deeper look at the issue itself, followed by characteristics that we envision the potential solution having. Following this, we will explain each potential solution that we have identified at this stage in the project, followed by an analysis of each potential solution. This section will then end with our chosen solution moving forward, as well as proving the feasibility of the solution. Following the technological analysis we will be explaining the planned integration of our technology, as well as a conclusion for the document.

2 Technological Challenges:

- We will need an accurate and easy to use Forecasting tool that will allow us to forecast the data we use from willow.
- Furthermore we will need a fast and reliable algorithm for these tools to forecast the data.
- We will need a way to integrate our AI model directly into Willow's platform for future utilization of our model.
    - This may require usage of the ONNX AI model.
    - Requires permission and resources from Willow in order to integrate our model in an agreeable way.
    - ONNX is also used for data visualization, which we will explore the feasibility below.
- We will need a reliable connection between our AI model and Willow's database to ensure real-time updates and energy usage monitoring.
- We will need to implement some sort of Machine Learning algorithm within our AI model to be able to forecast future energy usage to identify potential areas for improvement.
- We will need a way to provide an intuitive interface to display our energy usage forecasting from our AI model.
- To accurately predict how optimizations will affect energy usage, we will need permission and access from NAU to test real-time adjustments, like turning off lights or adjusting HVAC settings. This will help us measure the impact of changes and improve our predictions.
- Even when we get access, making real-time adjustments to building assets may require additional approvals and administrative oversight, which could slow down development and testing.

- Not all assets in a building are monitored, which means we don't have a complete picture of energy usage. This could affect the accuracy of our forecasts and make some optimization recommendations less reliable.

3 Technology Analysis:

3.1 Issue Introduction: Missing Data

Our current main issue our team is facing is getting ahold of the real time data that willow has to offer. We need this data to begin performing forecasting and optimization. We are hoping that this data is in a CSV format for easy integration into a model and that the data is already standardized and normalized so we don't get any missing or wrong data. We are currently working on getting this data (more below).

3.1.1 Missing Data: Desired Characteristics

The characteristics that we would like to get from our missing data is a nicely formatted data structure that we can easily pull information from and feed it into our forecasting model. We plan on using a split compare learning sequence that will split our data randomly to have some data train the model and other data to compare to while training the model. This is a very standard approach in the machine learning field.

3.1.2 Missing Data: Alternatives

An alternative route to the missing data problem, in the event that we cannot get access to this information (which is unlikely) is to use another website such as Kragle to pull building energy consumption data and use our data forecasting models on this instead. This will show Willow how our model can be integrated into their platform without needing the data which is protected. Moreover, there are a variety of alternative paths that we can use for the data forecasting tools, but we have done some research on using XGBoost if we find that the SciKit learn modeling doesn't work for us. However, based on the proving feasibility we have determined that SciKit learn is the optimal route when it comes to forecasting. Also we have done some comparative analysis on using XGRegression as the replacement to SciKits Random Forest Regression. XGRegression has shown that it can semi accurately predict information on large data sets, however we plan on using RFR as our main forecasting algorithm choice.

3.1.3 Missing Data: Analysis

Starting with the missing data analysis, we have reached out to both Willow and NAU to acquire the real time data that Willow uses. We have gotten no response back from either, but we are continuously following up with both to hopefully get access so we can start getting a grasp on the feasibility of the data usage.

3.2 Issue Introduction: Choosing a Forecasting Modeling Tool

One issue our team is encountering is choosing which forecasting modeling tool to use. There are many different options available to us, but we have chosen two to pick from. That is using one of the most standard modeling tools available, SciKit Learn. SciKit Learn is a good tool to use because it is easy to use and it has a variety of subtools to use that will help us visualize the data given to us. Our second option for a modeling tool is using XGBoost, which is a very efficient and accurate tool that works best on large datasets due to its ability to “boost”.

3.2.1 Choosing a Forecasting Modeling Tool: Desired Characteristics

One of the most important aspects that we wish to have from our data forecasting tools is to have fast and highly accurate models, which we will use to forecast weekly data. We need this model to perform machine learning algorithms (see below) fast on large datasets like the one Willow is going to give us.

3.2.2 Choosing a Forecasting Modeling Tool: Alternatives

One potential alternative for this issue is using one of the most popular Python Graphical User Interface (GUI) libraries, MatplotLib. Most of us already have prior experience with Matplotlib through university assignments/projects. Matplotlib is essentially the standard GUI package for Python, as it comes already installed with the standard Linux, Windows, and MacOS versions of Python. The Matplotlib package has been around for decades at this point, being developed in the late 80s. Matplotlib is commonly used in things such as desktop applications, as well as within applications for tasks such as data analysis, which would closely align with the needs for this issue.

3.2.3 Choosing a Forecasting Modeling Tool: Analysis

Next, our team has done in depth research on which data forecasting tool we can use. We started off by pulling data from Kagle.com which is a website that provides large data sets for people who want to use machine learning and AI. We were able to find building energy consumption data that spans from 2004-2018! After we were able to find quality data we used the pandas library in python to extract the data into columns of useful information like timing and energy consumption in MegaWatts. Now we are able to visualize what the data looks like (data visualization analysis in proving feasibility) and we can begin making a forecasting model using Scikit learn and Random Forest Regression. After we made the forecasting model on the split data, we were able to display the information into a graph and package the model into the

ONNX format. This was highly beneficial to our project analysis because it showed that we have a viable route to forecasting the data and packaging it into the specified format.

3.3 Issue Introduction: Choosing a Forecasting Algorithm

Another technology choice we have for this project is deciding which forecasting algorithm to use. There are countless different approaches we can take, but after a little research our team was able to choose between two different algorithms. The first algorithm is called Random Forest Regression. Random Forest Regression (RFR) uses a tree-like structure to make decisions on an optimal "branch" or "leaf" of the tree, then optimizing the forecast through regression and loss learning. We will explore why this is a good choice below. Another route that we have explored is using XGRegression, which uses similar math as RFR, however it uses a "boost" feature, which can speed up the forecasting process and deliver an accurate prediction.

3.3.1 Choosing a Forecasting Algorithm: Desired Characteristics

Our forecasting algorithm is going to be measured on two main features: the accuracy and the speed of the algorithm. We will need our forecasting algorithm to predict the forecast as accurately as possible, and do so in a timely manner.

3.3.2 Choosing a Forecasting Algorithm: Alternatives

As of the writing of this document, our team is leaning towards using Random Forest Regression for our forecasting algorithm. With that being said, the potential alternative we have is XGRegression. If our RFR algorithm does not forecast quickly enough, then we know we can switch to using the XGRegression algorithm for additional speed within our model.

3.3.3 Choosing a Forecasting Algorithm: Analysis

Now that we have a better understanding of our main route, we were able to test the feasibility of our alternative route, which is XGBoost and XGRegression. Similar to how we did the main route, we pulled the same data from Kagle.com and we were able to make a model using XGBoost and forecast the data using XGRegression. This was also hugely beneficial to our analysis since now we are able to compare the data visualization, the modeling tools and the forecasting algorithms to choose the best route.

3.4 Issue Introduction: Visual analytics

Once we are able to receive the data from either Willow or NAU directly, we want a way to visualize the data so that we can observe certain trends and identify useless data easily. Furthermore, people who aren't familiar with data analysis can easily look at the graphs and make assumptions. We currently have two different routes we can take, which include using Matplotlib's graphing tools to easily construct graphs and tables. We can also use ONNX for data visualization using Willows developer tools to allow us to have easy integration with existing willow platforms.

3.4.1 Visual Analytics: Desired Characteristics

Once we have the data we will need a way to visualize the data into graphs and tables. We need this data visualization to depict highly accurate information, with no deviation from other data visualization platforms. We want a platform that we can easily put our model into and have analytical information displayed to the user in an easy to understand format.

3.4.2 Visual Analytics: Alternatives

Another possible alternative for this issue is using the ONNX AI model and integrating it directly into the Willow platform, to be visualized on their existing graphs. ONNX was recommended to our team by our client in order to simplify the process of integrating our AI model into the Willow app, as Willow already has an existing method to embed an AI model of this type into their app. ONNX is an open source AI platform that supports interoperability among different AI tools. This platform would likely simplify the integration process of our model embedding into the Willow app, but may be more difficult to implement as our group is not as familiar with this platform.

3.4.3 Visual Analytics: Analysis

Overall the visual analytics for this project will likely be a much simpler issue than some of the other issues we will be dealing with. Having a way to easily port our model directly into the Willow platform through ONNX cuts down on a great deal of work that we would have to do otherwise.

3.5 Issue Introduction: Optimization

Lastly, we have an issue with how we would like to optimize the energy consumption of buildings at NAU. There are a few methods that we will be exploring which include real time analysis, asset alteration, and simulated environments to understand how we can reduce forecasted energy consumption.

3.5.1 Optimization: Desired Characteristics

We want an effective and quantifiable way of reducing energy consumption on our forecasted data. We want to ensure that our optimization route is easy to replicate and understand. Furthermore, we want to be able to have a clear reduction goal put in place before we start to optimize the buildings.

3.5.2 Optimization: Alternatives

If we cannot get access to a control environment, such as an unused building at NAU, we might just find asset energy consumption values and just implement them directly into our model, rather than changing real time building data to see if there is optimization in reducing peak load.

3.5.3 Optimization: Analysis

After discussing with our client OP Ravi, we have determined that the issue with optimization is on the outer shell of our scope for this project. If we can get through the main issues with the forecasting and visualization of data for Willow, then we will tackle this next problem of optimization. As of right now, we think we will be able to knock out all other issues relatively quickly, hopefully giving ourselves lots of time to figure out the optimization for our project.

3.6 Chosen Approach:

After fully analysing our main and alternative route, our team has concluded that using Scikit learn and Random Forest Regression is the most accurate, dependable and repeatable method for our project. We have shown our client the models we have made for both and he also agrees that this is the best approach based on the analysis below. Furthermore, we were able to deduce that the ONNX packaging system for willow is a viable and accurate route for both packaging and data visualization aspects for our project. Lastly, we also decided that using the standard numpy and pandas packaging for calculations and data pulling is the easiest route for the project, we don't need alternative routes for these since we already know they do the job as good as any other tool.

3.7 Storing and Managing Data:

Our data management system is going to be slightly different from a standard software project. Since we will likely be working within the Willow platform, we are going to be working with data that they have already collected and cleaned. This means that for the most part, our only interactions with the data will just be pulling from one of Willow's databases, instead of gathering and storing our own. This data will be in the form of CSV files, that we will likely read in through our backend using our chosen tool (this will most likely be the Python Pandas library). Because of this setup, we will likely not be managing any data at all, rather just accessing it. Since Willow manages all their data privately, we will be accessing it once we are granted the clearance to do so. Currently as of writing this document we are still in the process of getting NDA documents approved through Willow and NAU faculty. Once that is approved, it is simply a matter of each of us logging into the Willow platform with our login credentials and viewing any data we need. Once on the Willow platform, there are options to download CSV files for the data they have collected and cleaned up. There are also many options as to what sections of data you can download, such as which building/device you are viewing, the time/date range of data that you would like to view, and what time interval you would like to see data collected at (30 minutes, 1 hour, 2 hours, etc..). Since the data is already clean and available with such customizations, us accessing the data will likely just be running simple scripts to download/access different datasets, or just manually downloading particular datasets to upload to our program.

3.8 Proving Feasibility:

To figure out whether our forecasting tools and models would work the way we intended, we explored two main routes: one using Scikit-learn with Random Forest Regression (RFR), and a second route using XGBoost with XGBRegression. For both, we trained forecasting models using a large dataset of energy usage from Kaggle, and then compared the accuracy and performance of each one. We also tested how well each model worked when converted to ONNX format, since we'll need to use that to integrate with the Willow platform later on.

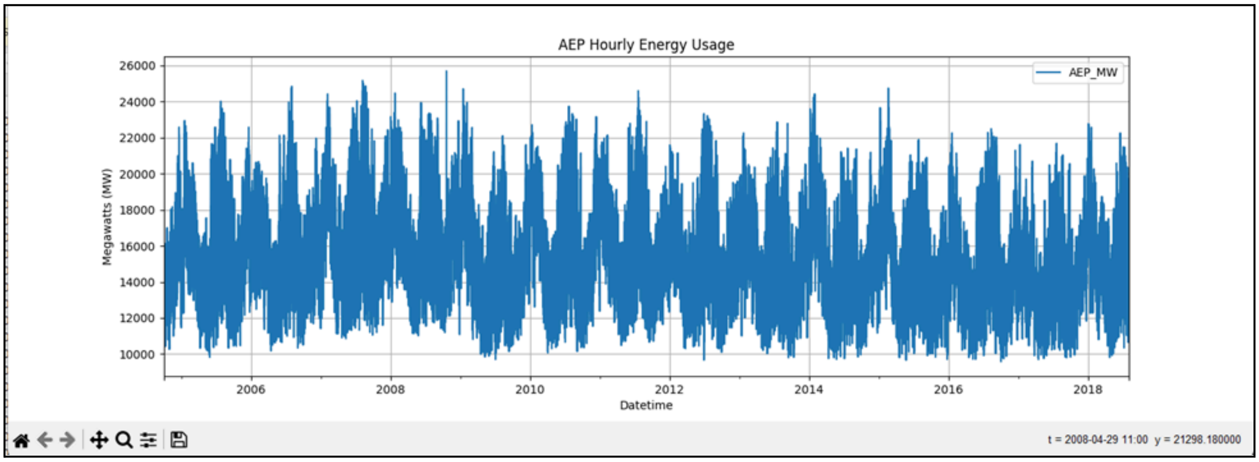
3.8.1 Analysis of Scikit learn:

Our goal here was to build a forecasting model that could predict energy usage over time. We started by installing the libraries we needed, like pandas for loading and cleaning the data, numpy for math operations, Scikit-learn for building the model, and matplotlib to help us visualize the results. We also used ONNX so we could eventually package the model for Willow. We pulled the dataset from Kaggle, which included building energy usage data from 2004–2018. Using pandas, we cleaned it up and focused on the most relevant columns—like timestamps and energy consumption in megawatts.

```
Dataset loaded successfully!
              Datetime   AEP_MW
0  2004-12-31 01:00:00  13478.0
1  2004-12-31 02:00:00  12865.0
2  2004-12-31 03:00:00  12577.0
3  2004-12-31 04:00:00  12517.0
4  2004-12-31 05:00:00  12670.0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121273 entries, 0 to 121272
Data columns (total 2 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   Datetime  121273 non-null  object
 1   AEP_MW    121273 non-null  float64
dtypes: float64(1), object(1)
memory usage: 1.9+ MB
```

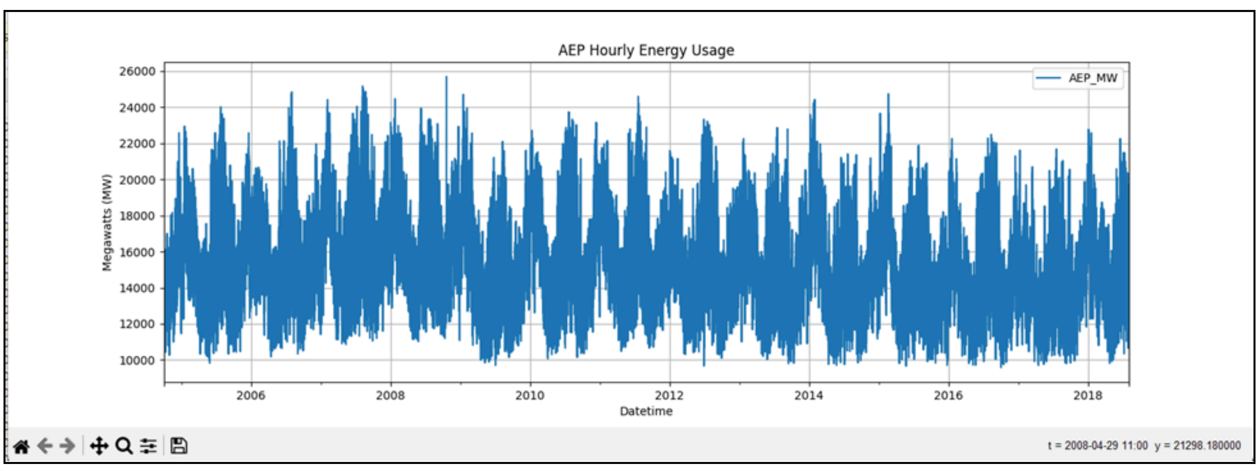
Successful data loading confirmation from pandas

Once we had the data, we split it and created plots of the raw values to get an idea of what the trends looked like before training the model.

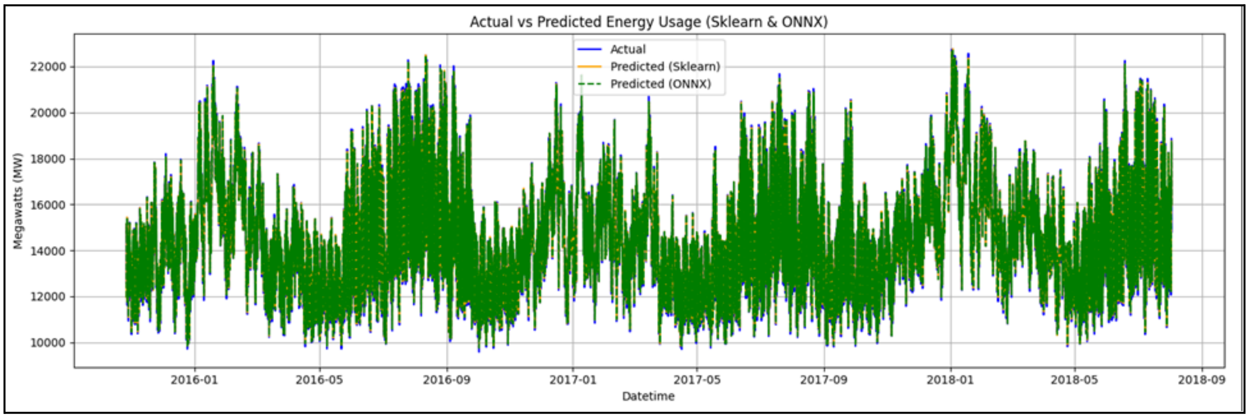


Initial data plotting (Datetime vs. Megawatts)

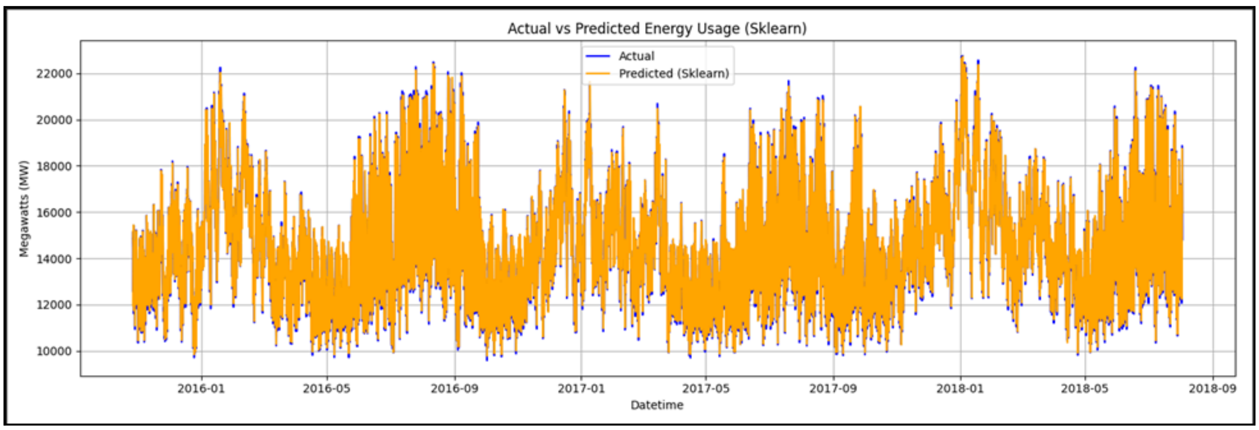
From there, we trained a forecasting model using Scikit-learn's Random Forest Regression and used matplotlib to visualize the output. We also converted the model into ONNX format to test its compatibility and portability. When we compared the predictions from both the original model and the ONNX version, the results were nearly identical. This told us that converting the model didn't affect the accuracy in any noticeable way. However, we did notice some small inaccuracies in the peaks and dips of the forecast.



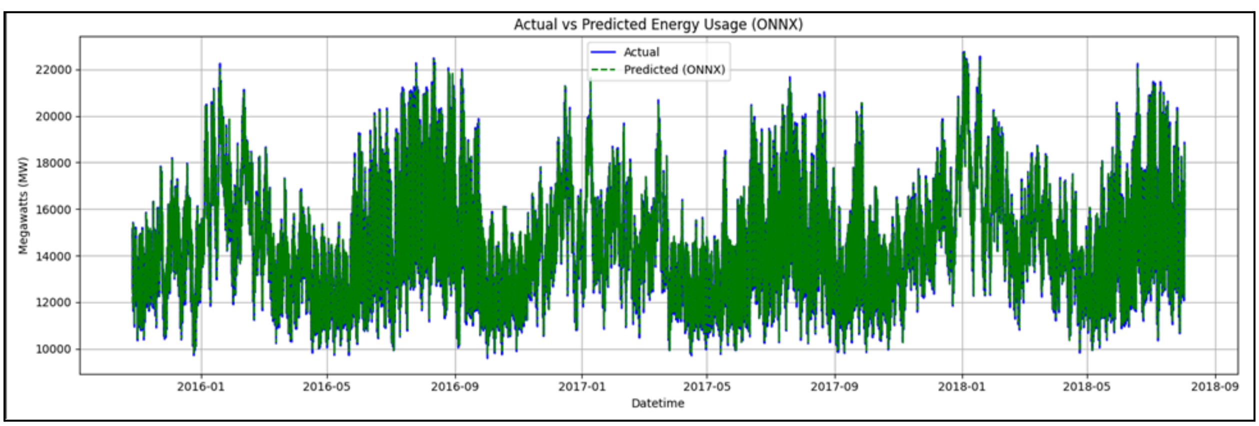
Actual Data



Actual vs both Sklearn and ONNX combined



Actual vs just Sklearn

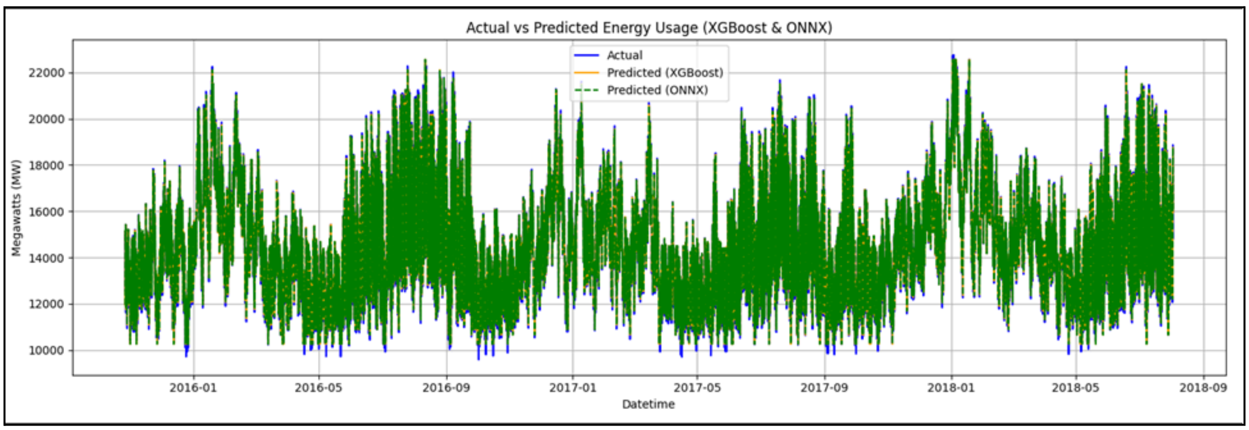


Actual vs just ONNX

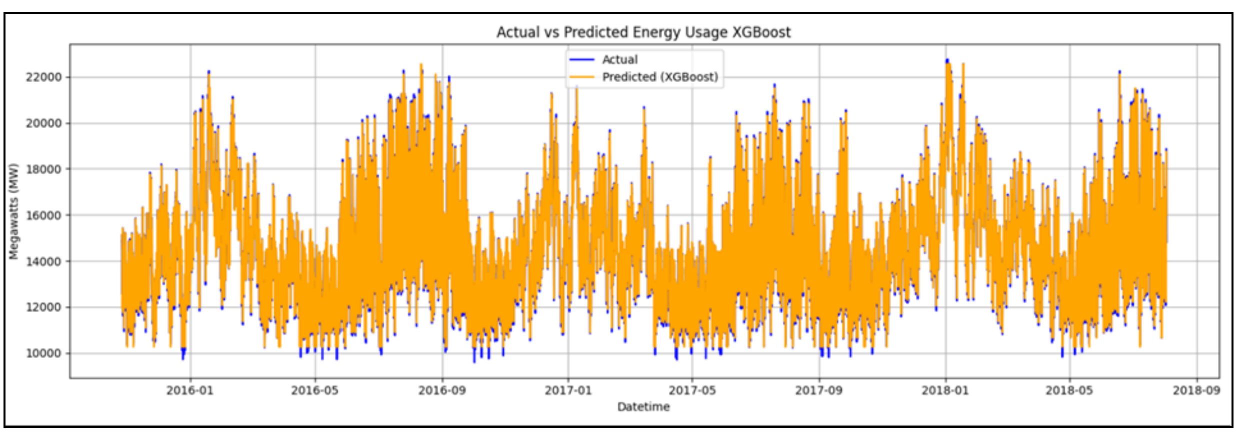
These visuals confirmed that the ONNX-converted model closely mirrors the original, and there was no major deviation in the forecast trend between the formats.

3.8.2 Alternative route: XGBoost and XGBRegression

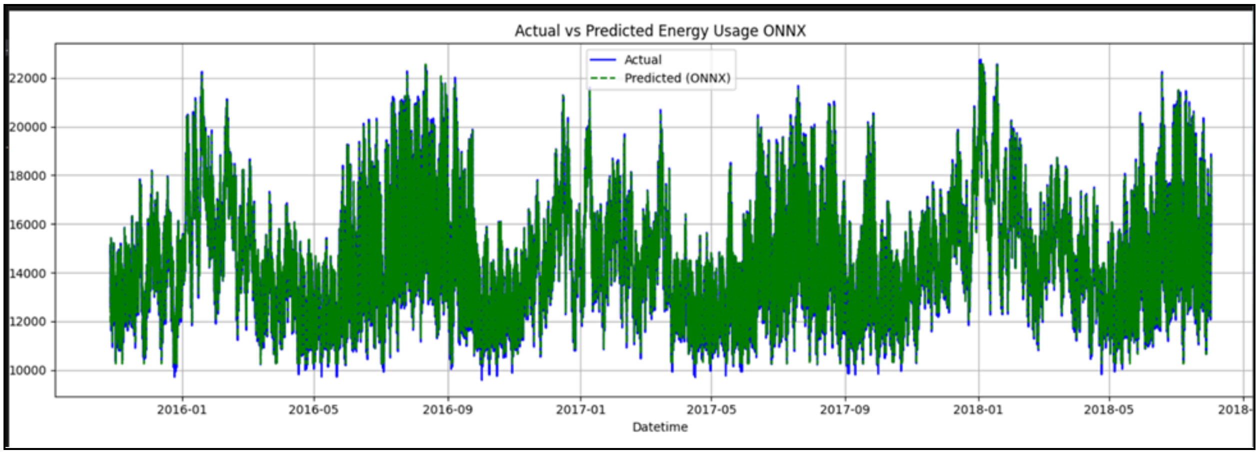
After testing Scikit-learn, we wanted to see how XGBoost would perform with the same dataset. We followed a similar process: cleaned and prepped the data, trained the forecasting model using XGBRegression, visualized the output, and converted the model to ONNX. The results were again very similar to what we got from Scikit-learn. The ONNX version worked just as well here too, and the forecasts lined up pretty well with the actual data. Like before, we saw some overshooting and undershooting in a few places, but nothing that made the model unusable. Overall, this gave us confidence that both tools are viable, and that ONNX is a reliable format for either route.



Both plots of actual vs predicted XGBoost and ONNX data representation



Just actual VS XGBoost

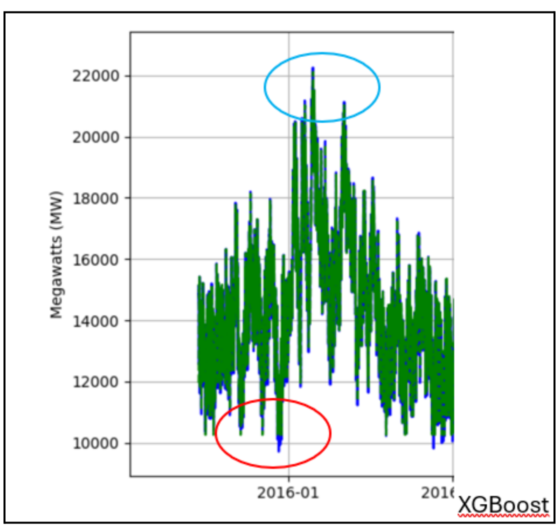


Just actual VS ONNX

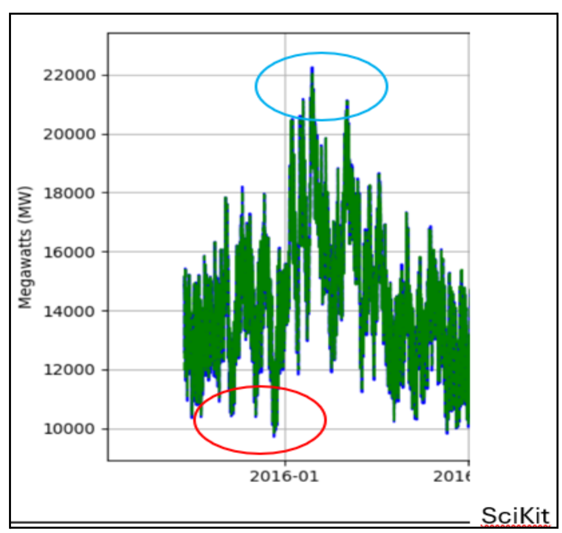
Again, we saw consistent outputs across formats, with minor forecasting errors in some of the high and low values.

3.8.3 Investigation into which route is better: (Scikit Vs XGBoost)

To figure out which option was better for our specific project, we directly compared the two models side-by-side using their ONNX outputs. We looked at how accurate they were, how easy they were to train and convert, and how well they handled peak usage times. Both models worked, but we found that Scikit-learn's Random Forest Regression did a better job predicting peaks and valleys in the data. It also had a simpler setup and smoother conversion to ONNX, which saves us time in the long run. XGBoost did offer faster performance on large datasets and had some tuning options to reduce overfitting, but Scikit-learn felt like the more straightforward and stable choice for what we're building right now.



XGBoost Accuracy



SciKit Accuracy

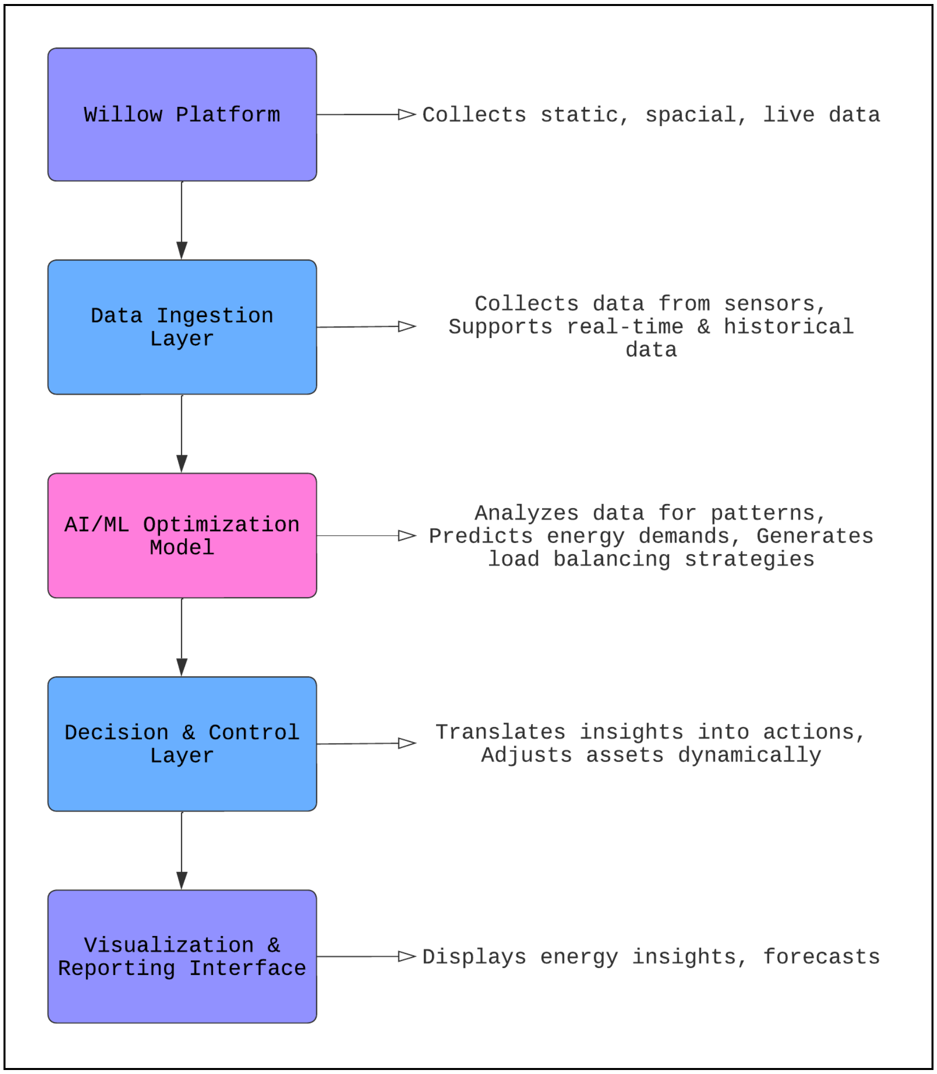
It appears that XGBRegression isn't as accurate as using Skearns random forest regression. As highlighted via the red and blue ovals, we can see that XGBRegression isn't forecasting some of the peaks and troughs that RFR is able to predict. However, both models don't seem to be perfect. Here's a quick summary of how the two compare:

Feature	**Scikit-Learn Random Forest**	**XGBoost**
Speed	Fast for small datasets	Optimized, faster for larger sets
Accuracy	Good	Often better due to boosting
Overfitting	Can overfit	Less prone with tuning
ONNX Conversion Simplicity	Easiest	Slightly more setup
Feature Importance	Yes	Yes
GPU Support	CPU only	GPU available
Model Size	Larger	Usually smaller

In the end, we chose Scikit-learn with Random Forest Regression as our main forecasting route. It's reliable, accurate enough for our needs, and easy to work with—plus, our client agreed that it's a solid choice based on what we've seen so far.

4 Technology Integration:

Successfully optimizing building energy efficiency requires that our team's micro-solutions to the individual challenges we've highlighted come together into a cohesive system where each part collaborates and adapts dynamically. Having a platform like WillowWatt that integrates solutions such as energy usage forecasts and AI-driven energy load balancing will ensure not only reduced energy consumption but also significantly improve the sustainability of multiple assets and buildings across NAU's campus. To achieve this, we propose a system that utilizes Willow's live energy data that is already being collected on building assets like HVAC systems and lighting. Our AI/ML optimization model will integrate directly with this data and transform it into actionable strategies for load shifting and asset scheduling.



System Diagram

As shown in the system diagram provided above, the data ingestion layer will collect both real-time and historical data from Willow's platform. From there, this data will be processed and analyzed by our model. The AI/ML optimization model will continuously monitor the live data to predict upcoming energy demands and identify opportunities for optimization. The system will then translate these insights into actionable strategies such as pre-cooling a building before peak hours, allowing for building's energy efficiency to be optimized without the need for manual adjustments. Finally, the WillowWatt system will integrate with Willow's platform to visualize and report these components in an intuitive way. This will help decision makers to understand and track the impacts of WillowWatt's optimization. By integrating each of these components into a cohesive architecture, our system ensures that Willow's data insights will be fully realized as actionable optimization and sustainability solutions.

5 Conclusion:

Buildings continue to be one of the most significant contributors to energy consumption and carbon emissions. Understanding and optimizing building energy usage across campus while maintaining operational efficiency and comfort is critical in the effort to reach NAU's goal of carbon neutrality by 2030. WillowWatt is a crucial step toward a smarter and more sustainable campus infrastructure. In this document, we have explored key energy challenges, analyzed potential solutions, and highlighted our approach to integrating live data collection with AI-driven insights. By combining Willow's data with our optimization model, WillowWatt will be able to maximize solutions in a way that helps NAU minimize energy costs, increase energy performance, and prioritize sustainability. Moving forward, our team is focused on obtaining necessary data, developing an initial prototype, and making sure that our planned system architecture works as intended. Our team has decided to approach this project in different phases. The product of phase one of the project will be the visualization/prediction model, and the second phase will add optimization suggestions and predictions. Once both phases are complete, the WillowWatt final product will be an AI/ML model that forecasts the energy usage of buildings and their assets, uses the forecasts to suggest actions that can be taken to optimize the building's energy, and predicts the effects that the suggested optimizations will result in. This will likely be done by using tools such as SciKit Learn and Random Forest Regression to aid in creating the model. We can then use the tool ONNX to package our model and easily port it to Willow's platform. Through continued research, iterative development, and close collaboration with Willow and NAU stakeholders, our team is committed to transforming our vision into a working, scalable solution that meaningfully impacts energy usage and sustainability across campus.