A Case-Study of Wind Turbine Power Forecasting Using Machine Learning Techniques

Pierre Huyn*,
Panitarn Chongfuangprinya*,
Sunder Raju,**
Chandrasekar Venkatraman*

***Big Data Laboratory, R&D Division,**
Hitachi America, Ltd., USA
****Atria Power Corporation, India**

Abstract

Renewable energy resources such as wind generation are enjoying a growing proliferation driven by a mixture of favorable legislation, incentives, technology developments and cost reduction. These resources enable utilities companies to provide cheaper and cleaner services, and new levels of customer satisfaction. However, due to its intermittent nature and higher variability, increasing penetration of wind and solar power generation leads to potential impacts on planning and operations of power systems. Power generation forecasting for renewable energy sources has become an important requirement in the energy industry. This paper presents a case study of power generation forecasting for a wind turbine farm operated by Atria Power Corporation, a major Independent Power Producer (IPP) based in India. Like many others, this IPP is mandated to provide accurate day-ahead power generation forecasts in 15-minute intervals. In this case study, we propose an Internet of Things (IoT) platform for the energy domain, through which SCADA data is acquired from wind turbines in real time, cleansed, aggregated, compressed, and securely transmitted to the cloud for analysis. To produce forecasts we developed forecast

models using Machine Learning (ML) techniques that can learn and execute large numbers of models efficiently and provide accurate predictions. Efficient use of computing resources in forecasting proves to be important as increasing numbers of forecast analytics will be deployed to edge computing devices. We also address other challenges such as when to refresh the forecast models, how to integrate external meteorological data, tradeoffs between data acquisition sampling rates and forecast accuracy, and how to improve reliability of our IoT pipeline. The proposed solution can be used for any renewable energy system such as wind generation systems, photovoltaic (PV) systems, and hybrid systems that combine wind generation and PV.

1. Introduction

Power generation using renewable energy resources, wind turbines in particular, has become increasingly popular globally [1]. When an Independent Power Producer (IPP) supplies power generated from these resources to the grid, it becomes very important, even mandated in many cases, to provide power generation forecast. This is depicted in Figure 1.

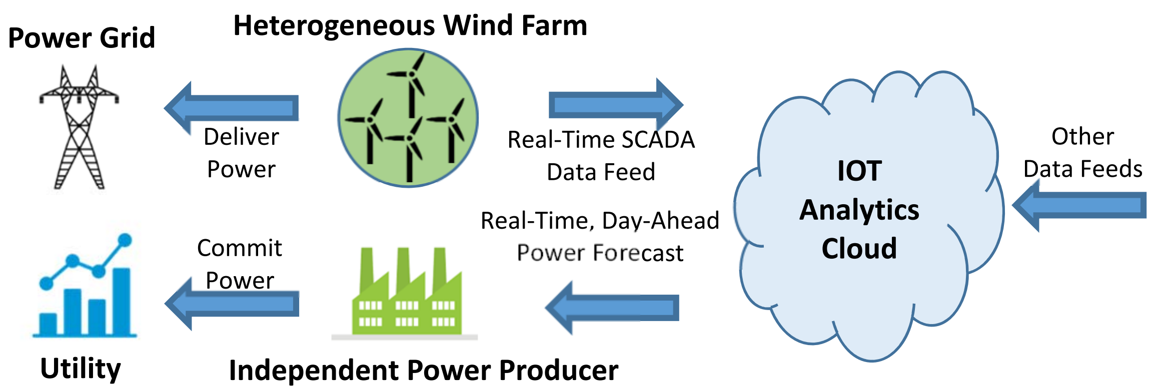

Figure 1. Supplying Renewable Energy to the Grid.

For these IPPs, forecast accuracy is their main business driver. Because meteorological processes which drive renewable energy generation, such as wind and cloud, can be highly unpredictable, accurate power forecast is a very challenging problem [2].

Since February 2016, we have been conducting a proof of concept (POC) with Atria Power Corporation, a major IPP in India, which provides us access to their wind turbine farms and turbine data generated in real time. They have completed energy projects in Mini Hydro Power, Wind, Solar, and Hybrid, with a goal to become a leading provider of technology-driven, clean, and renewable energy with grid and off-grid solutions. Their vision is to deliver disruptive solutions and enhance energy security for all of India. Besides forecasting output from their renewable energy projects, this IPP is also interested in using big data analytics for predictive and prescriptive maintenance, and reduction of spare parts inventory. Additionally they desire to improve output efficiencies of the wind turbines using ML techniques. This POC addresses one of their immediate business needs, to provide day-ahead wind power generation forecasts in 15-minute resolution every 90 minutes.

The data in this POC is retrieved from one of their wind farm site in South India, where 16 wind turbines, each of which at 1.6 MW capacity, are spread around 50 acres of land. Total capacity is 25.6 MW and wind turbine technology is based on a qualified induction generator.

This customer-driven co-creation research project has two main objectives:

- Gain access to wind turbine sensor data and drive development of a renewable energy Internet of Things (IoT) platform for power forecasting.
- Research renewable energy forecast analytics.

Wind power generation forecast research generally comes with many challenges:

- Data access: most IPPs are unwilling to share their data due to business sensitivity; gain access to real-time data from an actual wind farm installation; transfer large amount of data under high round trip network latencies; turbines in the middle of nowhere and physical access to site is not convenient; unreliable networks, ISP quality of service not very high.
- How to use data science techniques to forecast generated power.
- Accurate power generation forecast is difficult when the wind is unpredictable and experiences large swings at the wind farm site.

The rest of this paper is organized as follows. In Section 2, we describe an IoT platform we developed in this POC that provides end-to-end support for delivery of power generation forecasts in real-time. Section 3 provides details on our technical approach to power forecast analytics. In Section 4, we share lessons learned and discuss future work. We conclude in Section 5.

2. Energy IoT Platform

Wind turbines nowadays are equipped with sensors that can generate large volumes of data, which need to be collected to:

- Serve both real-time applications such as equipment health monitoring and visualization, continuous forecast generation and delivery.
- Populate a historical data historian for analytics research and development purposes.

Traditional software architectures that are well established for cloud computing need to be extended and adapted to support our wind turbine applications, which present new challenges:

- Large volumes of SCADA sensor data are generated.
- Data needs to be delivered in a networking environment that is unreliable and has high latencies.
- Access to the turbine site is not convenient.
- Data is business-sensitive and needs to be secured in transit.

New software architectures need to be developed to support these requirements. Figure 2 shows our power forecasting application stack built on top of an IoT platform for renewable energy. A survey of broad IoT trends can be found in [3]. Figure 3 shows an end-to-end IoT architecture used in our POC prototype.

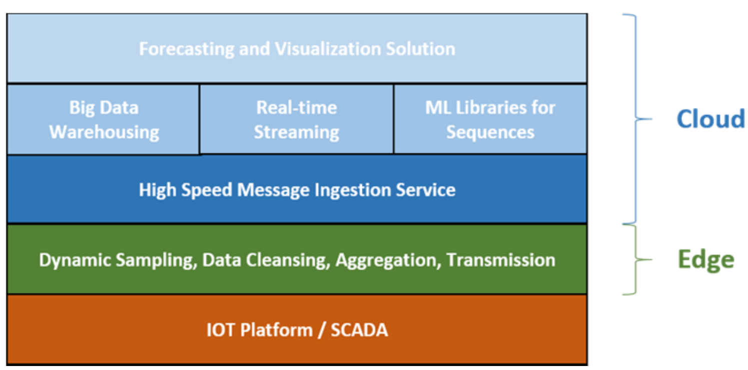

Figure 2. Power Forecasting Stack on top of an IoT platform.

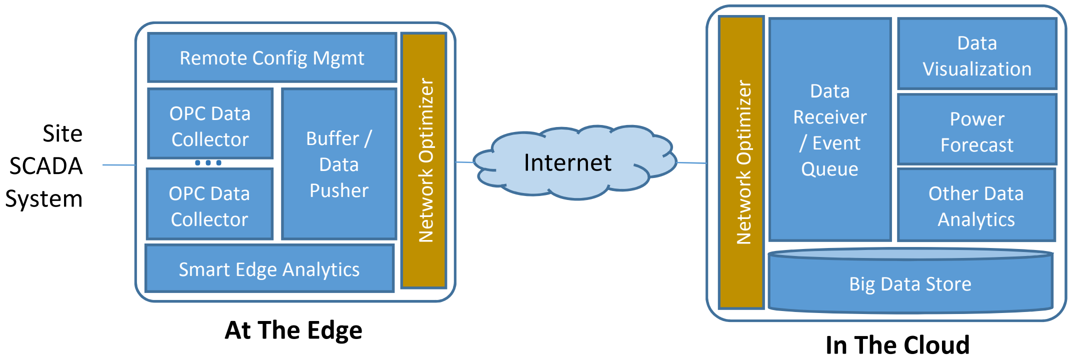

Figure 3. End-to-End IoT Architecture.

2.a Wind Turbine Sensor Data

Modern wind turbines generate a large number of sensor data that can be collected through a SCADA interface. The turbines in this POC can generate over 2,000 data points, some of which are shown in Table 1.

Table 1. Sample Wind Turbine Sensor Data

SCADA Sensor Data Collected	Measurement Unit
Generated Power	Kilowatts
Rotor Speed	Revolutions Per Minute
Wind Speed	Meters Per Second
Pitch Angle For Individual Blades	Degrees
Yaw	Degrees
Outside Air Temperature	Degrees Celsius
Turbine Operating State	Specific States
Generator Speed	Revolutions Per Minute
Nacelle Temperature	Degrees Celsius

Initially all sensor data are collected. Through exploratory data analysis, a small subset of highly relevant parameters can be determined. Collecting only a small number of parameters is a reasonable approach to minimize transmission bandwidth and data storage requirements. The module at the edge labeled “Remote Config Mgmt” in Figure 3 allows us to specify what sensor data to capture, close to the source, while letting us retain the flexibility to remotely reconfigure the settings when requirements change later.

2.b Processing at the Edge

We use an Object Linking and Embedding for Process Control (OPC) client to acquire data, in real-time, from an OPC server which exposes data through a standard OPC interface. OPC is an industry standard for data communication. Using an industry standard for data capture will make it easy to adapt our software components to wind turbines from other vendors and even to other renewable energy resources. The modules labeled “OPC Data Collector” in Figure 3 request data from an OPC server using an open source OPC client library.

The internet connections to the cloud are often slow and unreliable. Since a potentially large number of data points captured at a potentially high sampling rate needs to be transmitted to the cloud, a buffering mechanism is needed to deal with these inefficient and unreliable networks. This is achieved by a module labeled “Buffer / Data Pusher” in Figure 3.

Wind turbine sensor data is business-sensitive. So, before data is shipped to the cloud, it must be encrypted in order to secure transmission. Also, because we expect to be able to transmit large volumes of data, data compression techniques that can leverage the specific nature of turbine sensor data can help greatly enhance network latency and use of network bandwidth. The module labeled “Network Optimizer” in Figure 3 on the Edge side implements these requirements.

Up to this point, the functionality at the edge is adequate to bring the wind turbine raw data to the cloud in real-time for further advanced processing. However, not all processing need to take place in the cloud and, as we deepen our understanding of the application and sharpen the requirements for our energy IoT platform, a number of tasks emerge as excellent candidate for migration to the edge. The advantages of performing these tasks at the edge rather than in the cloud include increased security, enhanced data quality, and shortened sense-act cycles. The module labeled “Smart Edge Process” in Figure 3 could perform data aggregation, forecast model execution, incremental model buildings, and automated model deployment.

2.c Processing in the Cloud

We now describe the processing that takes place in the cloud, that is, on a cluster of servers that have significantly more computing resources than the edge computing devices. These servers are typically located in a private network behind a firewall.

Here is a brief description of the elements of the IoT architecture shown in Figure 3:

- The modules labeled “Network Optimizer” and “Data Receiver” on the Cloud side are the counterpart of “Network Optimizer” and “Data Pusher” on the Edge side.
- All messages that are processed by the “Data Receiver” are published in real-time to a reliable event queue or message bus. While a number of open source message bus components are available, we chose Apache Kafka, a widely used streaming platform that is fault-tolerant, scalable, reliable, and supports very high data ingestion rate. An alternative good choice would have been RabbitMQ.
- Feeding off the message bus, the remaining modules on the server side follows the Lambda architecture (explained in detail in [4] and illustrated in Figure 4) where:

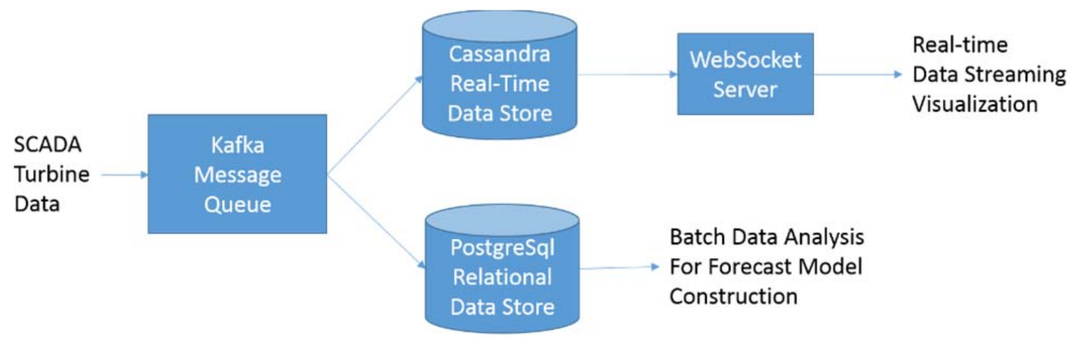

Figure 4. Instantiation of Lambda Architecture in Our Wind Power Forecasting Application.

  - A data stream is written to a real-time data store (Apache Cassandra is used in this POC) and powers a real-time visualization module, as illustrated in Figure 5.

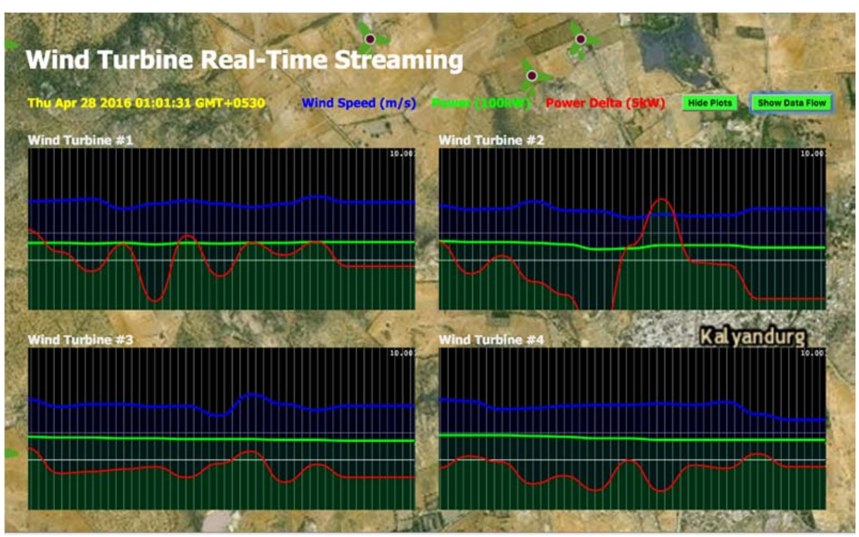

Figure 5. Visualizing Wind Turbine Data Streaming in Real Time.

- Another data stream is written to a durable store (Apache Hadoop HDFS) as a permanent storage and system of record for all historical data.
- Another data stream is written to a relational database (PostgreSql is used in this POC, but an MPP relational database would be ideal). Batch processing is applied to transform the data to a form suitable for exploratory (adhoc) analysis and for conducting advanced analytics such as building ML models.
- Another data stream is written to an operational data store (the same PostgreSql database is used here and is adequate for our purposes in this POC) so that recent data history can be made available in an aggregated form (e.g. data aggregates and time series moving averages) that is used as input to a power forecast model that executes in real-time or to other streaming analytics.
- As of this writing, an open source project Apache Kudu is emerging as an elegant solution that bridges the gap between supporting complex analytic queries on big data and serving fast-ingesting data quickly. As such, Kudu [5] has the potential to significantly simplify the IoT server side architecture.

3. Forecast Analytics

The core problem of interest here is called day-ahead power forecasting. We first give a precise definition of the problem, and then present our solution approach in the remaining subsections.

3.a Day-Ahead Power Forecasting

In our power forecasting problem, we make the following assumptions:

- Wind turbine SCADA sensor data is available and can be collected from the turbines in real-time.
- There is only a limited amount of historical data we can use to train our forecast model.
- There is limited availability of external meteorological data.
- As input to our forecast model, multiple time series SCADA data are available.

Figure 6 depicts the core forecast problem: at time T, we use recently observed SCADA data up to time T (as multiple time series) to predict power generation at time T + h. The forecast horizon h ranges from a few hours to a few days.

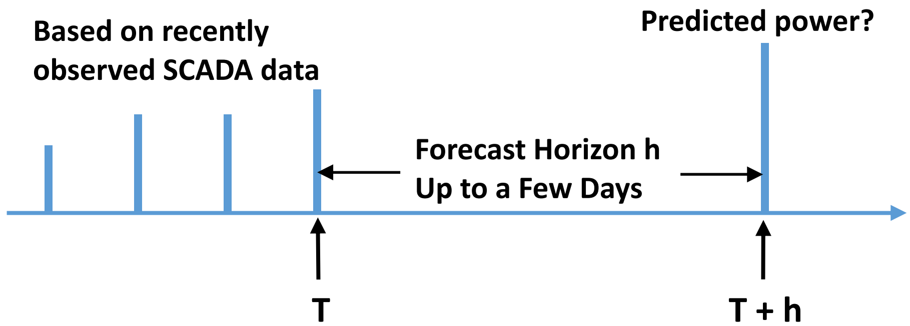

Figure 6. Day-Ahead Power Forecasting.

How do we define forecast accuracy? The quantity we are trying to minimize is the mean absolute percentage error (MAPE), defined as the mean of:

| <*Predicted Power*> – <*Actual Power*> | / <*Rated Turbine Capacity*>

where <*Rated Turbine Capacity*> is a constant that reflects the maximum power a turbine is configured to generate. MAPE is a forecast performance metrics commonly used in the wind energy industry [6] and most importantly, it is also a business requirement in our POC.

3.b Direct Power Forecast

Unlike traditional approaches which aim to forecast wind speeds which is then fed to a turbine-manufacturer-provided power curves (see [2], [6] for a brief overview of commonly used approaches to wind power forecasting), we model power directly. By doing so, we minimize power forecast inaccuracies that result in wind speed forecast errors amplified multiple times, since power behaves as a cubic function of wind speeds.

Power curves as provided by turbine manufacturers are theoretical/nominal transfer models that do not account for the variabilities in operational characteristics between different turbine installations such as aging factors, turbine-specific hysteresis effects, location, and altitude of installation.

Wind farm owners often operate turbines from a variety of manufacturers with different models. It is highly desirable not to depend on the availability of power curves and be able to provide a forecast solution in a manufacturer-neutral way.

Our approach used in this POC not only aims to forecast power directly, but customizes these forecasts to specific turbines. In doing so, we automatically account for turbine idiosyncrasies while remain independent of turbine manufacturers and models, individual turbines operational characteristics, turbine operational states and hysteresis effects, their ages, and their installation locations and altitudes. In order to support such a direct model of power generation, we only require the ability to collect power and wind speed data from SCADA sensors that are installed in the turbine.

3.c Forecasting Framed as a Regression Problem

Traditional approaches to forecasting in time series use ARIMA models [7]. While we are not familiar with the variety of extensions to the basic ARIMA models, the latter have serious limitations: 1) they are fundamentally linear methods and as such are not intended to capture non-linear dependencies in the time series; 2) they are designed to fit univariate time series and ARIMA methods for multivariate time series have not been thoroughly worked out; 3) fitting ARIMA models requires un-broken time series, something that does not hold in IoT data collection environment.

As an alternative approach, our multivariate time series can be sliced into a series of fixed length windows. These fixed-length vectors are used as inputs to our regression model, where the single continuous outcome is the power we want to forecast. This is illustrated in Figure 7, where we use windows of lengths *d* (for the power time series) and *d'* (for the wind speed time series). Obviously the choice of *d* and *d'* can be a function of the forecast horizon *h*.

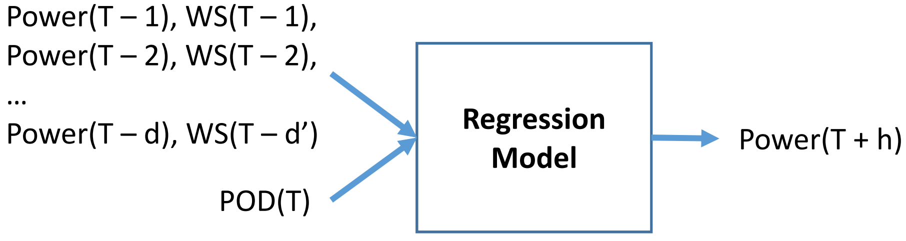

Figure 7. Power Forecasting Framed as a ML Regression Problem.

To leverage day-to-day seasonality patterns, we break up a day into 96 15-minute periods, and we use *Period-of-Day*, denoted *POD(T)* in Figure 7, an integer between zero (0) and 95 to index the 15-minute period where the time to forecast T falls into, as an additional predictor. Period-of-day has a significant impact on forecast accuracy for long forecast time horizons such as half a day away from now.

3.d Using SVM Regression Models

The day-ahead power forecasting problem for a wind farm can involve a potentially large number of models to build, execute and refresh frequently. As an example scenario where we have to provide forecast for every 15-minute periods for the next 24 hours, we will use 96 models for each turbine in the wind farm and thousands of models need be learned. Various ML techniques can be used for regression modeling:

- Traditional Artificial Neural Networks.
- Contemporary deep learning techniques such as Multi-Layer Perceptrons and Recurrent Neural Networks.
- Support Vector Machines (SVM) rooted in Statistical Learning theory ([8], [9]), developed and made popular in the 90s, and having enjoyed many practical successes in various domains.

For this particular application we chose SVM because learning SVM models can be super-efficient, compared with the other competing ML techniques which all require considerable training data and computational resources. While we currently train these models in an IoT cloud server with plentiful resources, these resource issues will become critical in the foreseeable future when we deploy forecasting analytics on edge devices. These competing modeling techniques not only take long time to converge, but more importantly, they suffer from producing solutions that are often locally optimal because of the non-convexity of their objective function. Fitting an SVM model involves solving a quadratic programming problem that can be solved efficiently. SVM has regularization built-in that can handle high-dimensional model inputs and as such, it supports the use of large number of predictors with minimal chances of overfitting the model.

To directly predict power generated during some future interval, we use power and wind speed measurements that have been observed in the recent past. In other words, as inputs to our prediction models, we used both power and wind speeds time series that are collected over a short time window, for example, spanning the last 30 minutes.

Typically SCADA sensors data is collected at a much higher rate than the required forecast resolutions. For example, while data is sampled at a rate of six (6) per minute, we need to forecast power averaged over a 15-minute period only. So the first preprocessing step is to aggregate the raw data to the right granularity level. Note that the time intervals over which data is to be aggregated can be anchored at any point in time. In other words, while the intervals for which we need to provide average power forecast are consecutive (and thus do not overlap), those used to aggregate data as input to the forecast model do not have to be disjoint and may in fact overlap.

For example, suppose power forecast is required for each 15-minute period. Data that is used as input to the forecast model will be aggregated at the same granularity level, that is, over 15-minute periods. If we decide to use four (4) such aggregates as input (this decision results from tuning our forecast model, as described later in this report). The corresponding four (4) periods may be disjoint, in which case they would span a window of one hour. Or they may overlap with anchors 5 minutes apart, spanning a window of 30 minutes. There is also another important reason these 15-minute periods do not have to be anchored at the top of the quarter on the clock, even though the forecast periods are so: by allowing anchors at every 5 minutes instead of every 15 minutes, we can easily triple the size of our training dataset.

4. Lessons Learned and Future Work

The following are customer's business requirements:

- Power forecast for a wind farm with 16 turbines spread over a 24 km area, with over 25 Megawatts rated capacity.
- The wind farm is located in a region that experiences unpredictable and turbulent wind patterns, especially during the monsoon months of June-September.
- Accurate power forecast is important because wind farm sells energy to a power grid and is mandated to provide day-ahead power forecast. If actual generation does not meet committed power delivery, the company will incur a penalty.
- Forecast parameters: frequency every 90 minutes; resolution where power averaged over 15-minute periods; forecast horizons ranging from T + 90 minutes to T + 90 minutes + 24 hours; target accuracy at 10% of rated capacity

This forecast problem is depicted in Figure 8.

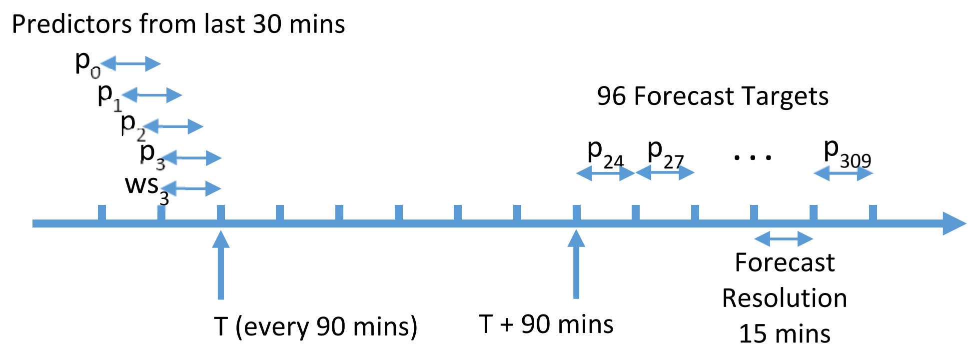

Figure 8. Forecast Problem in POC

A prototype has been operational since July 2016. This prototype is currently deployed on two Amazon Web Services servers. At our customer's request, a report containing 96 power forecast values is generated every 90 minutes and sent to the customer. Every two (2) weeks, the 96 forecast models are refreshed based on data collected from the previous four (4) weeks. Every week, a forecast performance report that compares actual power with predicted power is generated and sent to the customer. Our customer continues to validate the forecasting results we have been providing.

4.a Prototype Implementation

The cloud side of our IoT platform consists of two (2) pipelines:

- A streaming pipeline that produces the required forecast values in real-time (every 90 minutes), as shown in Figure 9.

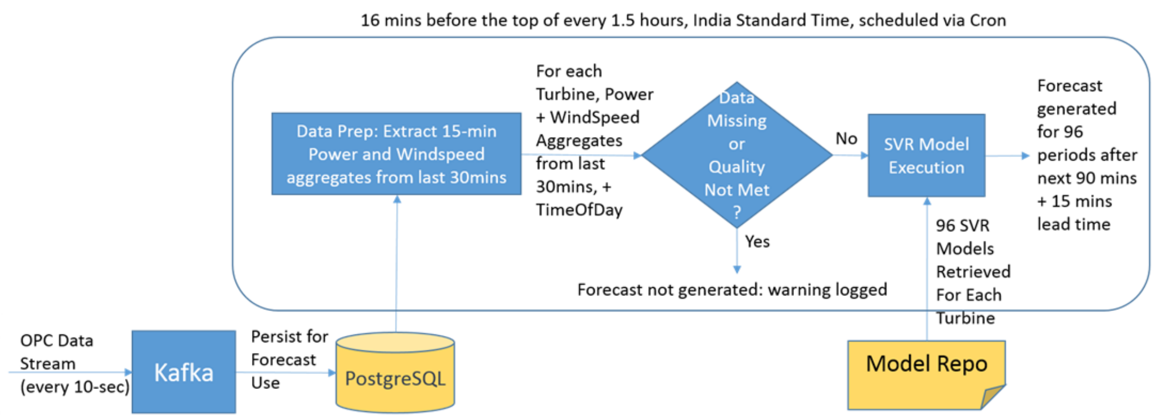

Figure 9. Real-Time Forecast Pipeline.

- A separate batch pipeline that train SVM forecast models using data collected over the previous four (4) weeks and aggregated at the 15-minute resolution, as shown in Figure 10.

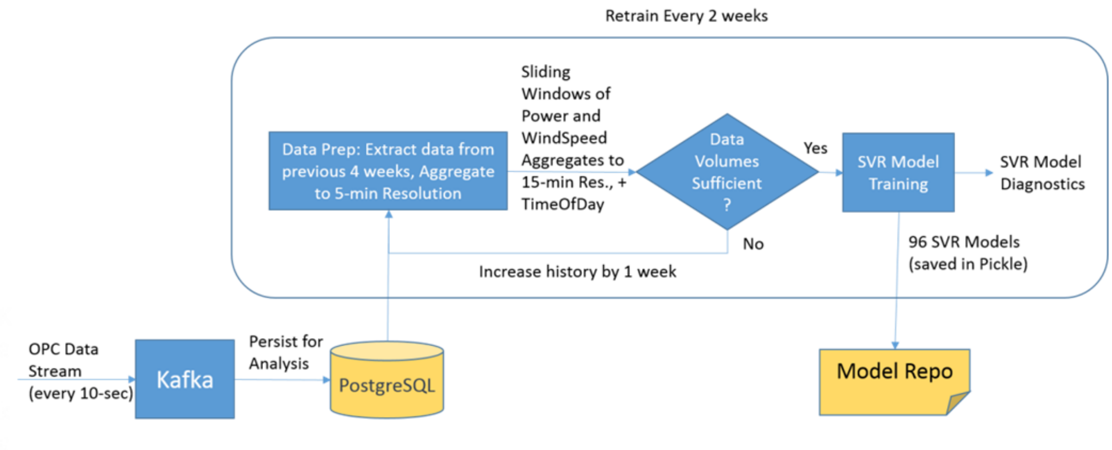

Figure 10. Batch Forecast Model Building Pipeline.

4.b Forecast Performance

On the average, our models achieve a MAPE of less than 8.5%. Figure 11 shows in greater detail how these MAPE values are distributed. To compute these performance figures, we consider only the first six (6) of 96 models. Accuracy of these models is very important because these six (6) forecasts are final, unlike the remaining 90 forecasts which can be revised in the following forecast cycles. Note that while these 90 forecasts can be viewed as tentative, day-ahead forecasting is a business requirement from our POC customer.

Observe that the forecast power curve is trying to mirror the actual power curve but with some delay. This delay is due to the smallest forecast horizon of 90 minutes. In order words, at time T, we calculate the forecasts for periods later than T + 90 minutes, based on what is observed before T. These forecasts are a function of these observations and are not affected by what happens between T and T+ 90 minutes.

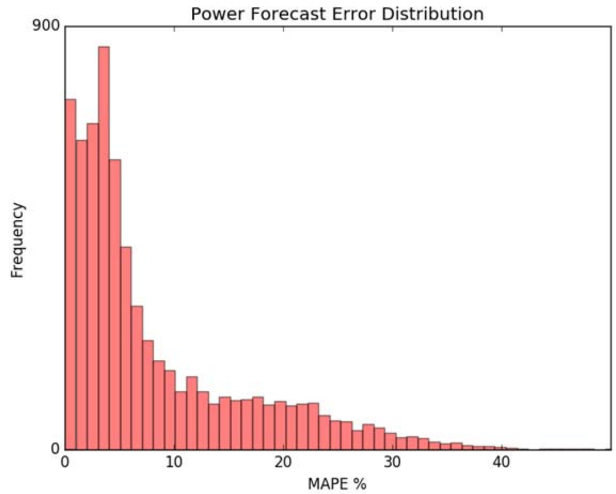

Figure 11. Power Forecast MAPE Distribution.

Figure 12 compares actual power vs. power forecast during three (3) full days in November 2016.

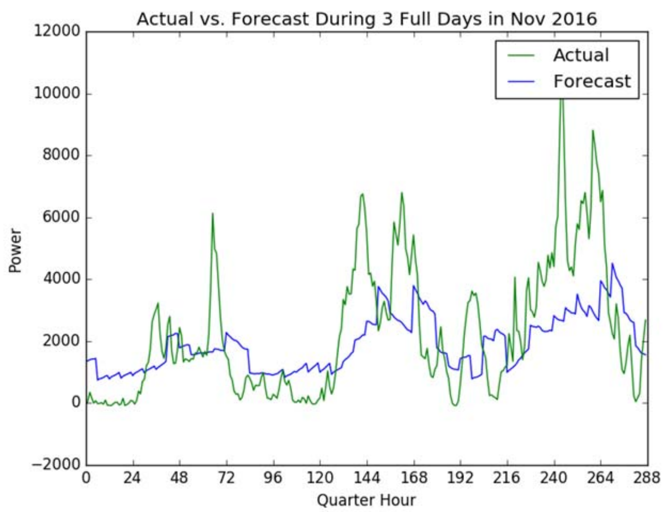

Figure 12. Actual vs. Power Forecast During 3 Days in November 2016.

Another observation that can be made is that sometimes, such as toward the end of the plot, the power forecast curve is tracking the actual power in the right direction but not close enough value-wise. One possible explanation is that if the training set does not contain certain patterns, these patterns (of turbulent time series) are not captured by the model and as a result, the model will not be capable of producing these patterns in the forecast.

4.c Unreliable Data Collection Can Impact Forecast Accuracy

During the POC, we learned that collection of wind turbine data, at least when using the current IoT platform architecture, is not always reliable. This unreliability manifests itself in different ways. The OPC server that we use to stream data out can often be non-responsive. If it is not monitored and restarted automatically, then manual recovery will be slow and incurs additional data loss. Data streaming can also be broken because of defects in the software components used in the pipeline. Finally, network delays can cause real-time messages to arrive too late to be considered, and this effectively contributes to decrease the actual SCADA data sampling rate. Then estimates based on averaging will become very noisy.

Figure 13 shows the number of "good" 15-minute periods during a 3-month period in 2016. A period is deemed good if we receive sufficient data samples during the period to get an accurate estimate of the average value for that period.

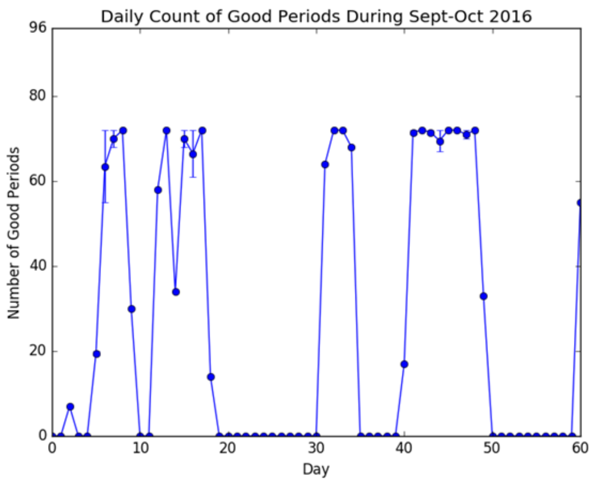

Figure 13. Daily Count of Good Periods During September – October 2016.

Low sampling rates not only impact data quality of the training set and model accuracy, but also the quality of estimated predictor values that are used as input to a model to produce forecasts. At the time of writing, we have not done the analyses to quantify how data sampling rate can impact MAPE. Using the historical data, we should be able to build forecast models under different sampling rates and assess their performance. More analyses remain to be done.

4.d Using External Meteorological Data

The main challenge here is to find a data source of meteorological forecast data that is sufficiently accurate at high resolution in both spatial and temporal dimensions. Since wind patterns can vary widely with altitude, good resolution is needed not only in longitude and latitude but also at altitudes where turbines are installed. Another challenge is that even if high resolution wind data forecast is available, we still need to adjust it based on the specific terrain characteristics that could locally distort the wind patterns.

Using an external meteorological forecast data source can be very useful to complement our forecasting approach. Currently forecast resolution in Asia is not as good as in the USA and UK. In India for example, weather forecast is provided at a resolution of three (3) hours, which is rather coarse for our forecasting problem. Another issue is that these forecasts do not seem to refresh often enough and thus the forecasts we receive can be stale. More seriously forecasts are provided at a spatial resolution of 40 km and at a single altitude of 10 meters.

If a good external source of forecast for wind speed and wind direction is available, it could be made more accurate by correlating it with turbine-specific SCADA data we are already collecting, such as wind speed and turbine yaw position, to obtain a more accurate forecast for effective wind speeds as received by the turbines. Historical wind forecasts, enhanced as described, can then be used to build power conversion models, which can then be applied to the enhanced real-time weather forecast data stream to product power forecast in real-time.

4.e Using Stand-By Models

Our models can be refreshed and deployed on a fixed periodic schedule, such as every week or two. Alternatively, we can use a more dynamic and adaptive model switching strategy. Even more generally, we consider using multiple competing models for forecasting. These standby models may be built using different training data or even using different selections of predictors. We continuously monitor and compare forecast accuracy of these models using actual data. Once a model consistently outperforms the others, especially the currently deployed model, it is then activated and becomes the deployed model.

4.f Deploying Analytics at the Edge

There are several advantages to deploying advanced analytics to edge devices rather than the cloud:

- Mitigate IoT security issues
- Improve reliability and therefore forecast accuracy
- Faster forecast turnaround times
- Vastly simplify the overall end-to-end architecture.

The challenge is edge devices do not have the same computing resources as the servers used in the cloud. While it remains to be seen how much smart analytics can be effectively pushed to the edge, we propose using Docker containers [10] to host these smart analytics, which makes them easier to deploy on a leaner stack, independent of the actual edge environment. We propose the following capabilities at the edge:

- Automatic sampling of SCADA data
- Data ingestion service
- Real-time data aggregation
- Forecast model execution
- Automated forecast model construction, storage, and smart deployment.

5. Conclusion

This paper presents a case study of power generation forecasting for a wind turbine farm operated by Atria Power Corporation, a major IPP based in India. Our solution includes 1) an end-to-end energy IoT platform for power forecast analytics which provides capabilities ranging from SCADA sensor data capture and transformation, real-time data streaming, model construction using historical time series data, and delivery of forecast results in real-time, and 2) forecast analytics that leverage machine learning techniques based on SVM to generate forecasts. A prototype has been operational since July 2016, achieving an average MAPE accuracy of around 8.5%. We plan to continue to refine our wind power forecasting solution core, namely to improve forecast accuracy as well as data collection reliability. We believe our approach to forecasting wind power generation can be easily extended to other types of renewable energy technologies such as solar thermal systems, PV systems, and hybrid systems that combine wind generation and PV.

6. Acknowledgment

The authors would like to thank Atria Power Corporation and Atria Institute of Technology for their support and collaboration in this project.

7. References

[1] M. Zieher, M. Lange, and U. Focken, "Variable Renewable Energy Forecasting – Integration into Electricity Grids and Markets – A Best Practice Guide," 2015.

[2] Daniel Kirk-Davidoff, "Wind Power Forecasting -- Independent system operators, regional transmission organizations, and utilities all have a critical need for accurate forecasts of electrical generation by wind," *Wind Systems*, pp. pp. 26--30, April 2012.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, pp. 1645--1660, 2013.

[4] Nathan Marz and James Warren, *Big Data: Principles and best practices of scalable realtime data systems*.: Manning Publications, 2015.

[5] Inc. Cloudera. (2015, Sept) Kudu: Storage for Fast Analytics on Fast Data. [Online]. https://kudu.apache.org/kudu.pdf

[6] K. Parks, Y. Wan, G. Wiener, and Y. Liu, "Wind Energy Forecasting: A Collaboration of the National Center for Atmospheric Research and Xcel Energy," Golden, Colorado, 2011.

[7] Rob Hyndman and George Athanasopoulos. (2012) Forecasting: Principles and Practice. [Online]. https://www.otexts.org/fpp

[8] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*.: Cambridge University Press, 2000.

[9] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2011.

[10] Docker, Inc. (2016) Introduction to Container Security: Understanding the isolation properties of Docker.