Anomaly Detection in Machine Data:

Understanding the Challenge

January 18, 2018

Team Members:

Garrett Cheung

Michael Galarnyk

Jared Goldsmith

Jillian Jarrett

Orysya Stus

Advisors:

Yoav Freund (UCSD)

Chad Holcomb (Solar Turbines)

Challenge

Our challenge is to develop an anomaly detection system for Solar Turbines' machine data. In an ideal world, we would build a complete pipeline: our model would be scaled and integrated into Solar Turbines' data ingestion processes. Our model would detect and differentiate different types of anomalies and would predict machine failures in real time. Predictions would allow for employees and customers to take preventative actions and reduce, if not eliminate, major failures. In terms of quantifying the cost of early inspection/repair versus catastrophic failure, Solar Turbines uses an order of magnitude difference as a baseline. A $100,000 spent on early repairs could prevent a $1 million catastrophic failure. Our anomaly detection system would minimize unplanned downtime (and associated costs) and hence be invaluable to Solar Turbines and their customers.

Opportunities (Set of Questions)

- **Is our 'grand challenge' achievable? What can we reasonably expect to do given our constraints? (see roadblocks below)**
  Given our limitations, it is unlikely to we will be able to create a full pipeline. Instead, our aim is to create a baseline model as a proof of concept.
- **Where do we get the data?**
  Solar Turbines. We can also bring in any third party data that might be relevant to our end goal.
- **What kind of limitations do we face using proprietary data?**
  - **What types data can we use?**
    We are subject to the limitations/constraints of what data Solar is willing to make public and share with us.
  - **How do we process data to get it off the Solar network?**
    Data needs to be anonymized and normalized, making it difficult to interpret.
- **How do we process the data to achieve our goals?**
  To be determined. Will require assistance from our advisors, and a good amount of experimentation.
- **Can we interpret the data?**
  Interpreting the data is going to be difficult due to our lack of domain knowledge and the anonymization/normalization of our dataset. However, we have enlisted the help of several Solar Turbines engineers to facilitate the process.
- **How do we generate labels?**
  The data set we currently have (machine data) does not have accompanying labels, making differentiating normal from anomalous behavior extremely difficult. We are in the process of requesting failure information from Solar Turbines. If this pans out, we can use it to label our data. Otherwise, we will have to rely techniques on such as clustering.

Data sources

Machine data. This data comes from the turbine packages themselves. Solar Turbines' packages are units comprised of compressors, combustors, turbines and application specific

components. The machine data is primarily sensor readings. More specifically, they are primarily temperature and pressure readings, but also include vibration/displacement data, speeds, programmable logic controller values, etc. Solar currently stores 1 hour data (used to generate alerts) and 10 minute data. 1 second data also exists, but only for intervals surrounding shutdowns. A single package has between 200-600 tags (columns), depending on model, control system, application type, etc.

While there are ~1,900 connected packages worldwide, our domain expert, Chad, recommended narrowing our focus on two engine models across a 2 year period (12/2015-12/2017). Model 1 is comprised of 33 packages and Model 2 is comprised of 39 packages. For these packages, we're taking a subset of features that exist across the entire fleets (only tags that exist for all 33 Model 1 packages or all 39 Model 2 packages). Of those, we are reducing our dataset to include package basics and features used to generate alerts. For Model 1, that reduces our dataset to 146 features, and 77 features for Model 2. Specifically, we are looking at 1hr and 10 minute resolution data. To reduce variability within our dataset, we are considering only data where the package is running close to its capacity (on load conditions) and running on gas fuel, per Chad's suggestion. Also, because the column names have to be anonymized, we're in the process of generating a data dictionary to give the team a better feel for what each column refers to.

Alerts data. Alerts are generated when a package parameter (corresponding to a column of the machine data) goes above a custom set limit. Once an alert is generated, a fleet manager views and responds to the alert. Ideally, they are supposed to write if/what action was taken but there is huge variability/inconsistency in this regard. We originally thought this data source might be useful for labeling our machine data, but inconsistencies in the data and other impracticalities have stopped us from exploring this potential data source. Most importantly, Solar Turbines is unwilling to let us anonymize and publish this data for our project.

Events data. We have two types of events: Shutdowns and Alarms. These are both generated by the programmable logic controller roughly every 30 milliseconds. However, they are not directly mappable to the machine data and difficult to interpret (i.e. shutdowns can happen due to a number of reasons, not just engine failure). Additionally, this data was not authorized by Solar Turbines when they agreed to let use their data for our project.

Failure data. We are in the process of requesting data from field service repairs and engine overhauls. Such information has potential to be useful for generating labels and differentiating normal from anomalous operating behavior. The format and composition of such data has yet to be determined.

Third party data sources. While it's possible that third party data sources would be useful for our end goal, we have yet to identify any as such.

Approach

This is a tentative game plan to understanding and finding meaning in our dataset. It is subject to change and any direction or feedback is greatly appreciated.

Data Acquisition and Preprocessing: Prepare data to be moved off the Solar network

- Pull data:
    - 1hr and 10 minute resolution machine data from Solar databases for 72 packages.
    - Timeframe: December 2015 – December 2017
- Preprocess data
    - Filter by fuel type (only looking at data points where dual fuel engines are running on gas)
    - Filter by load conditions (data points where the package is running)
    - Filter columns of interest (those consistent across the model, those used to generate alerts, package basics)
    - Construct composite tags (tags created from other tags and found to be meaningful by domain experts. They are generally linear combinations of other tags. We only need to create these for 10 minute data as they are already generated and stored for the 1 hour data. Construction for the 10 minute data follows the same formulas used for the 1 hour data.)
- Anonymize data
    - Alias column names
    - Normalize Data
        - Columns used to generate alerts were normalized by their global limits
        - Other columns were normalized by factors suggested by Chad, i.e. engine hours were normalized by a package's expected lifetime
        - Engine start counts and timestamps were not normalized
- Move data off the network

Data Exploration: Explore the data, look for anomalies

- Data forensics
- Anomaly identification techniques:
    - Statistical analyses
    - Dimensionality reduction
    - Clustering

Data Comprehension and Analysis: Make sense of our findings

- Will require assistance from our domain experts
- Identify anomaly subtypes, including but not limited to
    - Removed sensor
    - Sensor malfunction
    - Unit conversion error
    - Engine overhaul
    - Turbine component malfunction
- Rank/assess severity of anomalies
- Create labels

Machine learning: Build a model

- Construct a model to classify anomalies using labels we created
- Validate model

Visualizations: Visually encode our work

- Visualize our data
- Visualize our model's predictions

Reporting: Document our journey

- Detail our methodology
- Discuss our findings
- Allow for future groups to expand on our work

Roadblocks

- Resources
    - Limited manpower: 5 people
    - Timeframe: 5 months
    - Budget: $2000
- Data availability
    - Policy/privacy related
        - Requires permission from Solar Turbines
        - Can't use any Solar data not explicitly authorized by Solar's Digital Team
        - Limited by the amount and type of data Solar is willing to provide
    - Many types of machine failures don't translate to the tags we have (or any sensors on the package at all)
    - Labels don't exist
- Technical
    - Lack of expertise/domain knowledge: limited understanding of the data
    - Data had to be anonymized before it could be moved on the Solar network, making it difficult to read and interpret
    - Most team members' first time working with time series data
    - Human machine interface??
- Cultural
    - Team members have different schedules and work styles
    - Three of the team members don't work at Solar Turbines

Team Roles and Responsibilities

- Garrett Cheung
    - Solar-side coder (anonymization/data preprocessing and publishing)
- Michael Galarnyk (Bookkeeper)
    - Keep track of expenses (AWS fees, etc)
- Jared Goldsmith (Record keeper)
    - Responsible for maintaining our github repository

- Jillian Jarrett (External Team Coordinator)
    - Liaison between Solar Turbines and UCSD/SDSC
    - Responsible for reporting progress and correspondence with Ilkay
- Orysya Stus (Internal Team Coordinator)
    - Responsible for managing project priorities, timely code delivery, and overall progress

Project Coordination and Communication Plan

Project coordination and communication plans are still being developed. That said, Jillian will be our team coordinator to make sure we're on schedule and individual's unique skill sets are being utilized. The team will meet with our domain expert, Chad Holcomb, at Solar Turbines on a monthly basis to discuss progress and address any issues that require a domain expert. Our meeting schedule with our advisor, Yoav Freund, has not been established, but we are anticipating something along the lines of monthly meetings. In addition to meetings with our advisors, the team members will meet on a biweekly basis to share our progress, strategize and support one another.