Paper SAS645-2017

Real-time Analytics at the Edge: Identifying Abnormal Equipment Behavior and Filtering Data near the Edge for Internet of Things Applications

Ryan Gillespie and Saurabh Gupta, SAS Institute Inc.

ABSTRACT

This paper describes the use of a machine learning technique for anomaly detection and the SAS® Event Stream Processing engine to analyze streaming sensor data and determine when performance of a turbofan engine deviates from normal operating conditions. Turbofan engines are the most popular type of propulsion engines used by modern airliners due to their high thrust and good fuel efficiency (National Aeronautics and Space Administration 2015). For this paper, we intend to show how sensor readings from the engines can be used to detect asset degradation and help with preventative maintenance applications.

INTRODUCTION

The data set used is the 2008 Prognostics and Health Management (PHM08) Challenge Data Set on turbofan engine degradation (Saxena and Goebel 2008). We use a single-class classification machine learning technique, called Support Vector Data Description (SVDD), to detect anomalies within the data. The technique shows how each engine degrades over its life cycle. This information can then be used in practice to provide alerts or trigger maintenance for the particular asset on an as-needs basis. Once the model was trained, we deployed the score code on to a thin client device running SAS® Event Stream Processing to validate scoring the SVDD model on new observations and simulate how the SVDD model might perform in Internet of Things (IoT) edge applications.

IOT PROCESSING AT THE EDGE

IoT processing at the edge, or edge computing, pushes the analytics from a central server to devices close to where the data is generated. As such, edge computing moves the decision making capability of analytics from centralized nodes and brings it closer to the source of the data. This can be important for several reasons. It can help to reduce latency for applications where speed is critical. And it can also reduce data transmission and storage costs through the use of intelligent data filtering at the edge device. If you have a better sense of what data is valuable and what data is not valuable at the edge of your network, you can be more selective of what data you choose to transmit and store. With intelligent data filtering, the use of real-time event stream processing engines can reside at the edge of the network to not just score new data, but to pre-process and filter the data before analysis or scoring.

In our use case, we are evaluating sensors from a fleet of turbofan engines to determine engine degradation and future failure. To do this, we constructed a scoring model to be able to do real-time detection of anomalies indicating degradation. In practice, it is easy to anticipate an organization trying to monitor a large fleet of turbofan engines or other capitally intensive pieces of equipment. This equipment might not have access to a centralized cluster of computers to be able to be monitored in real time. Or, the latency associated with transmitting the data might be greater than the practical benefit of monitoring the asset. This is an area where edge computing can help to resolve issues where the scale of data transmitted is a concern.

The closer we can put the model to the sensors, or the source of the data, the less time it takes to make a decision due to the decrease in data movement. The risk of the decision process being interrupted by an unreliable or slow network connection is also decreased if a light weight scoring engine resides at the edge of the network.

Edge computing can also help reduce the costs associated with the analytics infrastructure. Training your failure model happens less frequently than scoring the data, so you can reduce costs by using metered computing to train your model and then deploying the decision or score code to many smaller nodes at the edge. Most analytics use cases do not use all available data. This is because it is redundant, incomplete, or noisy data. By leveraging edge computing, you can filter or pre-process the data with an event streaming engine closer to the source. Then you can use only the relevant data in the proper format to both train your model and to make your predictions or alerts.

THE POTENTIAL MARKET FOR EDGE COMPUTING

According to Cisco, there will be 50 billion connected devices by 2020 (Cisco 2014). Intel has an even more bullish prediction at 200 billion connected devices by 2020 (Intel 2017). No matter who is right, the opportunity for edge computing appears large. IDC has estimated that investment on IoT, which includes devices, connectivity solutions, and services will reach $1.7 trillion (IDC 2017). Telefonica estimates that 90% of cars will be connected by 2020 (Telefonica 2014). And GE estimates that the “Industrial Internet”, which is the Internet of connected industrial machinery such as turbofan engines, will add about $10 to $15 trillion to the global GDP in the next 20 years (General Electric and Accenture 2014). This amount of investment and value at stake will likely require analytics solutions that can be deployed at a centralized location or on edge devices for a variety of prediction and real-time monitoring applications such as anomaly detection.

SUPPORT VECTOR DATA DESCRIPTION (SVDD)

One potential solution to the issue of anomaly detection is a method called Support Vector Data Description. It’s a machine learning technique that can be used to do single-class classification. The model creates a minimum radius hypersphere around the training data used to build the model. The hypersphere is made flexible through the use of Kernel functions (Chaudhuri et al. 2016). As such, SVDD is able to provide a flexible data description on a wide variety of data sets. The methodology also does not require any assumptions regarding normality of the data, which can be a limitation with other anomaly detection techniques associated with multivariate statistical process control.

If the data used to build the model represents normal conditions, then observations that lie outside of the hypersphere can represent possible anomalies. These might be anomalies that have previously occurred or new anomalies that would not have been found in historical data. Since the model is trained with data that is considered normal, the model can score any observation that is not considered normal whether it has seen an example like it before or not.

Being able to detect new anomalies can be key to Internet of Things applications or detecting new threats related to cyber-security or fraud. Given that the model also only requires one class of data for construction, it can be beneficial for applications where there is a severe imbalance of observations between normal and non-normal operating conditions.

The implementation of SVDD used for this paper can be found in SAS® Visual Data Mining and Machine Learning.

APPLICATION OF SVDD

To illustrate how SVDD can be applied to a predictive maintenance scenario, we used the algorithm on the 2008 Prognostics and Health Management (PHM08) Challenge Data Set on turbofan engine degradation (Saxena and Goebel 2008). The data set consists of examples of simulated turbofan engine degradation that were used for a data challenge competition at the 1st international conference on Prognostics and Health Management.

DESCRIPTION OF DATA SET

The data set consists of multivariate time series information for a fleet of engines with the engine operating normally at the start of the series and degrading at a point until failure is reached.

There are 26 variables within the data set. They correspond to the engine ID, the cycle number of the time series, three operational settings, and 23 sensor measurements. Within the training set, there are 218 different turbofan engines simulated to a point of failure, with the number of cycles to failure ranging between 128 and 357 cycles with a mean failure point of 211 cycles. In total, there are 45,918 observations within the training data set.

APPLYING SUPPORT VECTOR DATA DESCRIPTION TO THE PROBLEM

The Support Vector Data Description algorithm was applied to the problem to help determine when the time series is beginning to deviate from normal operating conditions. The output measurement of the algorithm provides a scored metric that can be used to assess the degradation of the engine and help put in place preventative measures before the failure point.

To train the model, we sampled data from a small set of engines within the beginning of the time series that we assumed to be operating under normal conditions. As previously noted, the SVDD algorithm is constructed using the normal operating conditions for the equipment or system. It can also handle various states of normal operating conditions. For example, a haul truck within a mine might have very different sensor data readings when it is traveling on a flat road with no payload and when it is traveling up a hill with ore. However, both readings represent normal operating conditions for the piece of equipment.

With this in mind, we randomly sampled 30 of the 218 engines from the data set to be used to build the SVDD model. Of the 30 engines that were sampled, the first 25% of each engine's measurements were then used to train the model. As such, it was estimated that the data within this region was related to normal operating conditions. This resulted in a training set used for the model consisting of 1,512 observations out of the total 45,918 observations.

It should be noted that examination of the three operational setting variables indicated that there were six different operational setting combinations within the data set. Given that the algorithm is flexible enough to accommodate varying operating conditions, no additional indicator flags or pre-processing work was performed on the data to model the different operating conditions.

The model was trained using the svddTrain action from the svdd action set within SAS Visual Data Mining and Machine Learning. The ASTORE scoring code generated by the action was then saved to be used to score new observations using SAS Event Stream Processing on a gateway device.

SCORING NEW OBSERVATIONS TO DETECT DEGRADATION

As the type of example is related to a potential Internet of Things (IoT) use case, it was decided to implement the scoring code within a SAS Event Stream Processing engine running on a gateway device. This was done in order to validate implementing predictive maintenance scoring algorithms near the edge for IoT use cases.

In the case of aircraft engines or other assets that can generate potentially large amounts of data, it is beneficial to be able to bring the analytics to where the data resides. This can help in two ways. In the first, it can help reduce latency in instances where an extremely quick decision is required (or help to make a decision if there is no network connection to a source that can score the new observation). The second way is that it can also help to filter data at the edge. If a new observation is deemed to be outside of the normal operating range, it can be sent back to a central storage system for data collection, analysis, and future model building. Similarly, for all observations within the normal operating range, it is possible to select a sample of them to be sent back for collection and future use. As such, the volume of normal operating condition observations transmitted and stored can be greatly reduced.

A Dell Wyse 3290 was set up with Wind River Linux and SAS Event Stream Processing (ESP). An ESP model was built to take the incoming observations, score them using the ASTORE code generated by the VDMML program and return a scored distance metric for each observation. This metric could then be used to monitor degradation and create a flag that could trigger an alert if above a specified threshold.

The remaining observations associated with the 188 engines that were not used for model training were then loaded onto the gateway device and streamed into the processing engine to be scored. In an application, data would be fed to the gateway device, scored and/or sampled, acted on or monitored if necessary, and then sent to a central location for storage or further processing. For the purposes of the validation, the scoring results were output to a comma-delimited file for analysis of how the model scored the new observations.

SAMPLE RESULTS

The scoring results from the hold-out data set illustrate the degradation in the engines captured by using the SVDD model. Four random samples were taken from the 188 scored engines with their SVDD scored distance plotted versus the number of cycles. This is shown in Figure 1, Sample SVDD Scoring Results. As seen in the figure, each engine shows a relatively stable normal operating state for the first portion of its useful life, followed by a sloped upward trend in the distance metric leading up to the failure point. This upward trend in the data indicates that the observations are moving further and further from the centroid of the normal hypersphere created by the SVDD model. As such, the engine operating conditions are moving increasingly further from normal operating behavior.

With increasing distance indicating potential degradation, an alert can be set to be triggered if the scored distance begins to rise above a pre-determined threshold or if the moving average of the scored distance deviates a certain percentage from the initial operating conditions of the asset. This can be tailored to the specific application that the model is used to monitor.

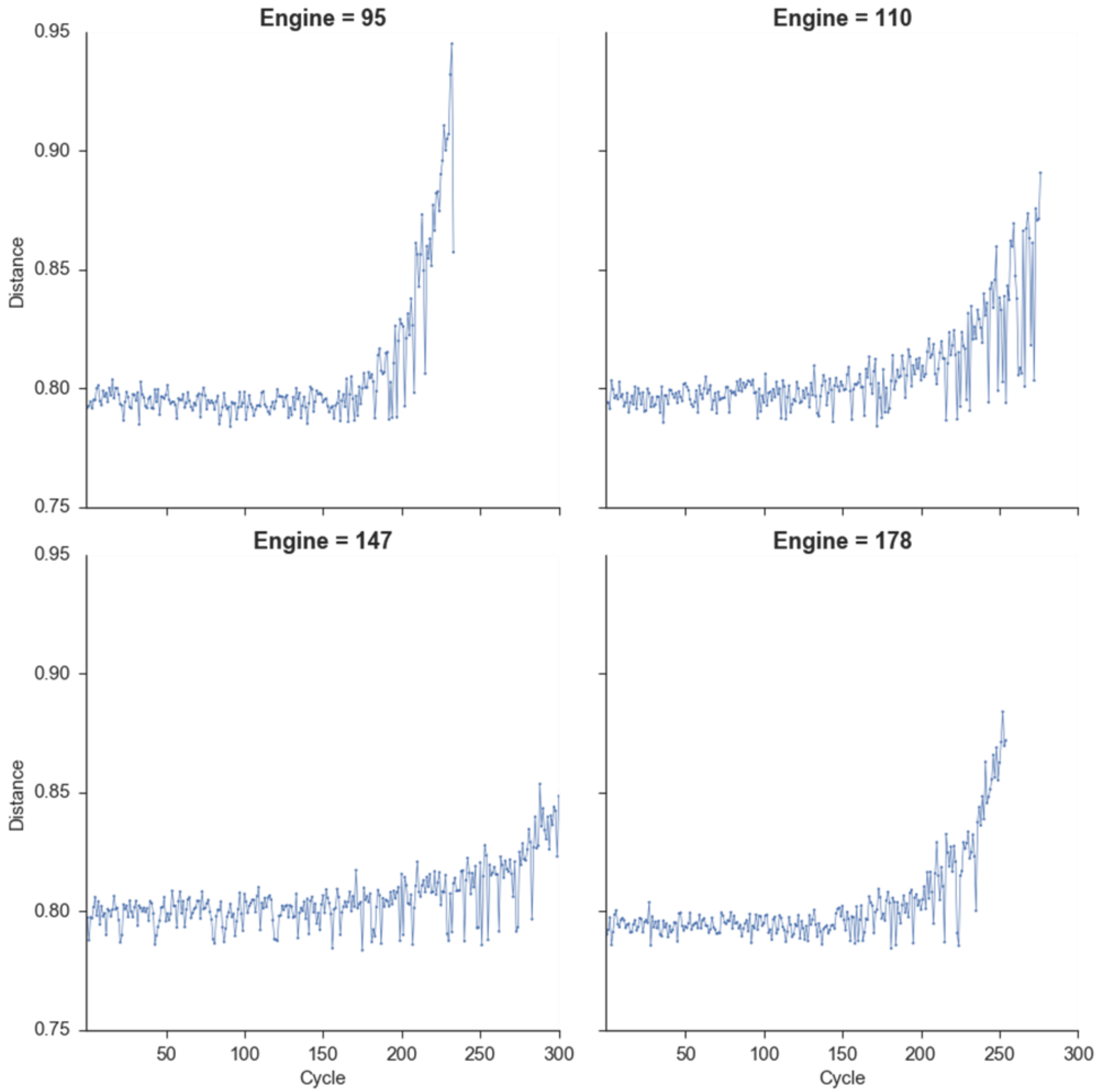

Figure 1. Sample SVDD Scoring Results

CONCLUSION

Anomaly detection can be a useful tool to detect asset degradation and help with preventative maintenance efforts. In this paper, we discuss how we applied a single-class classification technique called Support Vector Data Description to monitor how turbofan engines degrade from normal operating conditions. Given the potential use of real-time anomaly detection for Internet of Thing applications, we also tested scoring the model on a gateway type device to mimic application in the field. The results of the model on new data show visual trends indicating the degradation with the turbofan engines used in the example.

REFERENCES

National Aeronautics and Space Administration. “Turbofan Engine.” Retrieved on February, 28, 2016 from https://www.grc.nasa.gov/www/k-12/airplane/aturbf.html

A. Saxena and K. Goebel. 2008. "PHM08 Challenge Data Set." NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA. Retrieved February 24, 2017.

Chaudhuri, Arin, Deovrat Kakde, Maria Jahja, Wei Xiao, Seunghyun Kong, Hansi Jiang, and Sergiy Peredriy. 2016. “Sampling Method for Fast Training of Support Vector Data Description.” eprint arXiv:1606.05382, 2016.

Cisco. “The Internet of Things.” Retrieved February 25th, 2017, from http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/iot-aag.pdf

GE and Accenture. “Industrial Internet Insights Report for 2015.” Retrieved February 25th, 2017, from https://www.ge.com/digital/sites/default/files/industrial-internet-insights-report.pdf

IDC. “Connecting the IoT: The Road to Success.” Retrieved February 25th, 2017, from http://www.idc.com/infographics/IoT

Intel. “A Guide to the Internet of Things Infographic.” Retrieved February 25th, 2017, from http://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html

Telefonica. “Connected Car Industry Report 2014.” Retrieve February 25th, 2017, from https://iot.telefonica.com/multimedia-resources/connected-car-industry-report-2014-english

ACKNOWLEDGMENTS

Thanks to Seunghyun Kong, Dev Kakde, Allen Langlois, and Yiqing Huang whose code contributions and help made this paper possible. And also, thanks to Robert Moreira for suggestions and input on the ideas in the paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Ryan Gillespie
SAS Institute Inc.
Ryan.Gillespie@sas.com

Saurabh Gupta
SAS Institute Inc.
Saurabh.Gupta@sas.com