AUTOMATED DETECTION OF ANOMALIES IN HIGH-FREQUENCY WATER QUALITY SENSOR DATA USING MACHINE LEARNING

Xi Wang, Emil Sekerinski, McMaster University,
John B. Copp, Primodal Inc.*

Primodal Inc., Hamilton, Ontario, L8S 3A4
copp@primodal.com

INTRODUCTION

Wastewater treatment facilities are increasingly installing more and more high-frequency water quality sensors, as high-quality data is essential for plant operation and optimization. The sheer volume of data being collected and the necessity to avoid the collection of erroneous data, has created a need for automated tools to assess the quality of that data and signal for maintenance as the need arises. As these datasets have increased in size and complexity, it has become difficult to identify problems in a timely manner either manually or to use simple rules that might have been sufficient previously. A software solution is thus developed to provide a quick analysis of fault detection. The anomaly detection algorithm is developed based on deep learning technology, where the detection model is derived solely from the data and no prior knowledge required.

In November 2017 this project was launched to acquire real-world dataset from municipal and industrial wastewater plants. Our industrial partner, Primodal installed an RSM30 monitoring station with two high-frequency ammonia sensors at the primary effluent of the Dundas Wastewater Treatment Plant (WWTP) in Hamilton, Ontario. The generated data has been used to test the algorithms developed as part of this project. The two VARiON® Plus 700 IQ high-frequency sensors, conducts potentiometric measurement of ammonium concentration using ion-sensitive electrodes.

Primary effluent ammonia is influenced by daily, seasonal and weather issues and thus exhibits typical stochastic behavior observed at the treatment plant. This stochasticity is a non-trivial problem as any algorithm must distinguish real but normal events (observed as changes in concentration) from sensor anomalies (observed as changes in concentration). The term anomaly in this paper refers to any patterns or instances which differ from expected pattern. Figure 1 shows the daily concentration in normal dry days.

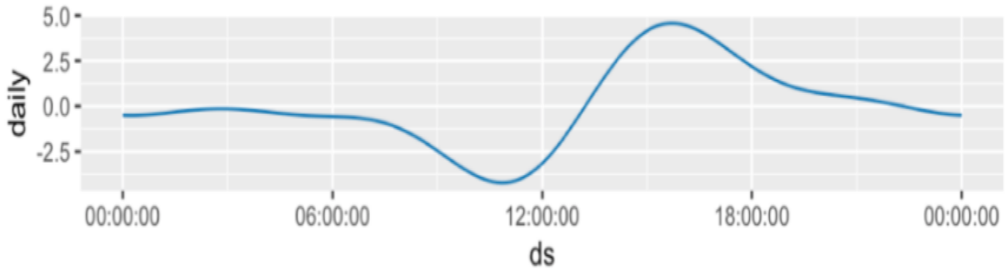

FIGURE 1: A typical ammonia concentration pattern in a typical dry day.

Traditional ad hoc anomaly detection approaches struggle to identify real faults from normal process variations due to their rigid logic. A static threshold is not enough to distinguish faulty data with normal data, as the range of normal concentration is usually changing over time. A threshold that works well for the winter may not work for summer anymore. Besides, a set of manually designed rules for one WWTP may not work another WWTP. The data-driven approach is preferable as all the "rules" are derived from dataset directly. A LSTM (Long Short-Term Memory) network is chosen as the data-driven approach due to its success in the recent time series problems in difference applications (Graves et al., 2013). It has the ability to learn long-range patterns and store the "rules" as a predictable model. Prediction of future values is calculated with its preceding values and the predictable model. The anomaly score can be calculated based on the difference between the predicted values and actual values. The anomalies can be detected and ranked based on the score.

Moreover, some of the detected anomalies are the results of rain events. These precipitation-caused anomalies should be eliminated from the results when possible, as the "real" anomalies caused by sensor failures are of more interest. It is obviously possible to eliminate the rain events when precipitation or flow data are available. However, these two datasets are usually not accessible by the simple water quality sensors. An attempt was made to eliminate the "real" concentration anomalies with the temperature datasets collected by sensors.

METHODOLOGY

Rule-based Approaches

Anomaly detection can be done manually by someone with sufficient domain knowledge. However, anomalies indicate that the system may not operate properly. The problem needs to be alarmed as soon as possible to avoid collecting faulty data. It is infeasible for a human engineer to monitor data 24/7. It is thus advantageous to automate this process by employing algorithms.

Simple thresholds such as an upper or bottom threshold can be applied, where a violation of a threshold triggers an alarm. This simple static rule works well for situations such as emergence when the concentration reaches extreme values. However, the normal range of ammonia concentration is not static. For the concentration in this project, in winter time such as January, an upper threshold can be set at 20 as no reading can be above this value. Using the same threshold will trigger a lot of false alarms in the summertime such as June, as normal concentration can be easily above 25. A more proper threshold should be around 28 at this season.

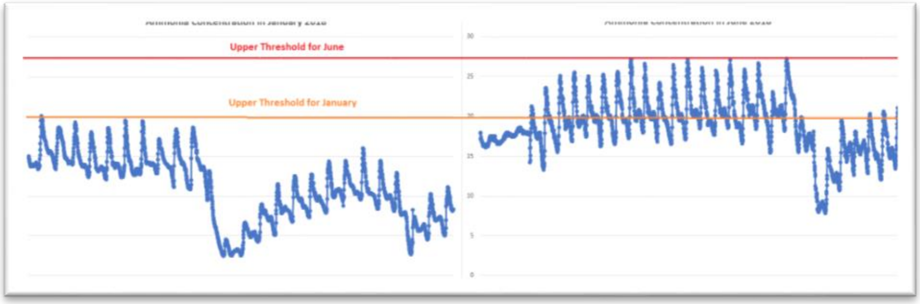

FIGURE 2: Threshold Comparison between January and June.

A more feasible approach is to detect anomaly using its statistical features. One possible approach is to detect anomaly based on the possibility of distribution. As most data are within the normal range in a well-operated system, anomalies are usually rare events (Vallis et al., 2014). For example, for a normal distributed system, most of the data aggregates around its mean as shown below. Those unusual data on the two sides are very likely to be anomalies. A small percentage such as 5 percent can be set as the threshold, as it can exclude most the normal data.

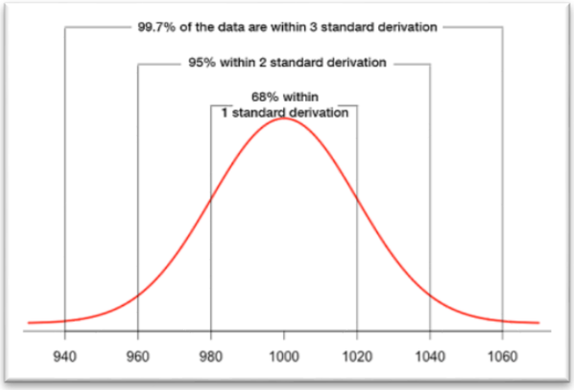

FIGURE 3: Normal Distribution.

The Extreme Studentized Deviate (ESD) test (Rosner, 1983) is an algorithm which detects anomalies assuming data is normally distributed. However, the concentration dataset exhibits a multimodal distribution due to its daily pattern so that general ESD cannot be applied directly. To address this problem researchers from Twitter built the Seasonal Hybrid ESD (S-H-ESD) on the top of the general ESD test (Vallis et al., 2014). A procedure was taken to decompose the seasonal components such as daily pattern using the median and MAD (Median Absolute Deviation). MAD is defined as the median of the absolute deviations from the sample median. The resulting residual component has a unimodal distribution that is suitable for anomaly detection with general ESD.

The second possible statistical approach is to make use of the derivative, as an anomaly is often related to the rapid or unusual rate of change in a time series. The third option is to make use of the daily pattern, as the expected value of a data sample can be estimated from its preceding value based on the expected pattern (LinkedIn, 2018). By moving lagging windows across the full dataset, a series of the subset can be generated to compute the expected values for each sample. Anomalies are identified from deviations from the expected value. The three approaches discussed above, i.e., S-H-ESD, derivative, and moving average are all based on different statistical features. There are more potential rule-based approaches available for similar tasks. Some more complex statistical techniques may even achieve satisfying results after being carefully designed by experts. However, they suffer from similar limits, i.e., the rules are "custom-made" for specific datasets. The customization requires extensive effort and these handcrafted rules tend to be difficult to adapt to a new system.

Data-driven Approach

Another way to handle this problem is to take the data-driven approach, where the "rules" are generated from data directly instead of design by an expert. In this paper, deep learning is chosen to be the data-driven approach because it has shown robust capabilities in a variety of tasks in recent years (Graves et al., 2013). Deep learning can learn high-level representations of datasets automatically with little or no need for manual feature engineering and domain expertise. The basic structure of deep learning algorithms is called an artificial neural network (ANNs). The artificial neural networks are inspired by the structure and function of the biological neural networks in the brain. The term deep refers to architectures consisting of multiple hidden layers. In this paper, all the neural networks will be referred to as artificial neural networks.

In general, a neural network consists of an input layer, multiple fully connected hidden layers, and an output layer. The input layer is the first layer which receives the input data. The input layer only forwards the input data without any processing. The output layer is responsible for processing the output. The layers in-between are referred to as hidden layers. Each layer contains multiple computational units, which are also called nodes or neurons. The nodes within hidden layers are responsible for memory and the inner state. A neuron receives input vectors from neurons in the previous layer along the directed edges. Each edge has a corresponding weight associated with it. The input data will be then be multiplied by the weight and subsequently added to the bias. Afterward, the sum serves as input for the activation function or transfer function. The output of a single neuron is usually a non-linear function of the weighted sum of its inputs and bias, which is achieved by the activation function. The final output will then be passed to neurons in the next layer. The complex features of the dataset are learned during this process, which is similar to a pipeline where each layer does part of the task.

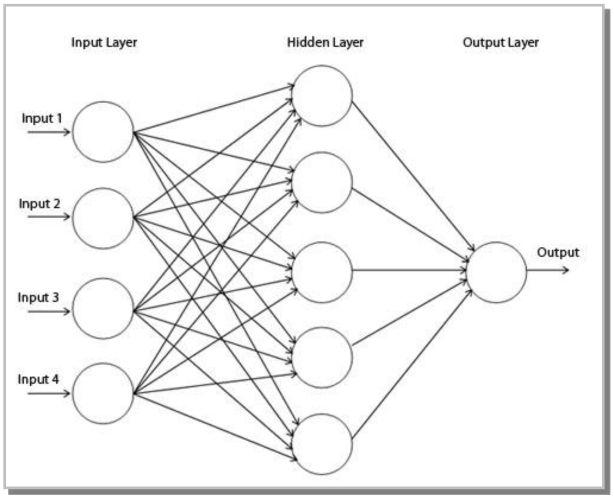

FIGURE 4: Artificial Neural Network

Training Neural Networks

The difference between the predicted output and the expected output is referred to as the loss function or cost function. The training purpose of a neural network is to find the optimal parameters which minimizes the loss function. When the predicted output is close enough to the actual output, i.e. the loss function is minimized, the training process can stop as the information has been "learned". One commonly used loss function is the mean squared error (MSE), which measures the averaged squared distance between the predicted values and actual values. The gradient is the measure of the change in the loss function corresponding to changes in the parameters. The algorithm used to calculate the gradients is called backpropagation, which calculates the gradients concerning the parameters. The "back" part of the name comes from the fact that the calculation of the gradient propagates backward through the neural networks. The gradient of the final layer of weights is calculated first while the gradient of the first layer of weights is calculated last. Partial computations of the gradient from one layer are reused in the computation of the gradient for the next layer based on the chain rule of derivatives. This backward flow of the error information allows for efficient computation of the gradient at each layer.

The calculated derivatives are used by an optimization algorithm, gradient descent, to adjust the weights up or down, depending on the direction that minimizes the loss function. The optimization is an iterative process, where the training data needs to be passed multiple times before it reaches the optimal result. One pass over all the training datasets is referred to as an epoch. After every epoch, the parameters, i.e., weights and biases get closer to their optimum values which minimize the loss function. A proper number of epochs can be set such that the parameters move from under fitting to optimal.

Recurrent Neural Networks with Long Short-Term Memory (LSTM)

There are different types of neural networks, which expertise in different tasks. Recurrent neural networks (RNNs) as one type of ANNs, are designed for temporal tasks such as speech recognition. The idea behind RNNs is to make use of sequential information. RNNs are called recurrent because they perform the same operation on every sample of a sequential input, with the output being dependent on the previous steps. The main feature of an RNN is the hidden state, which is calculated based on the previously hidden state and the input at the current step. The first hidden state is typically initialized to all zeroes. RNNs are formed with a chain of repeating modules. At any time *t*, the RNN unit receives the input from the current time step and the hidden state from the previous time step. The output is then calculated, and the hidden state is also updated. The current output depends on all the previous inputs. Thus, memory is maintained during the training.

In theory, RNNs can learn dependencies between steps that are arbitrarily far apart. However, in practice, an RNN is only able to remember short-term memory sequences due to the so-called vanishing/exploding gradient problem (Pascabu et al., 2013) caused by the repeated use of the recurrent weight matrix. Vanishing gradient refers to the problem when the gradients flowing through the network become very small as the chain rule is applied many times in the backpropagation. In the worst case, this may completely stop the neural network from further training. The exploding gradient problem is the opposite situation, where the weights become so large that the model becomes unstable and unable to learn from the training data.

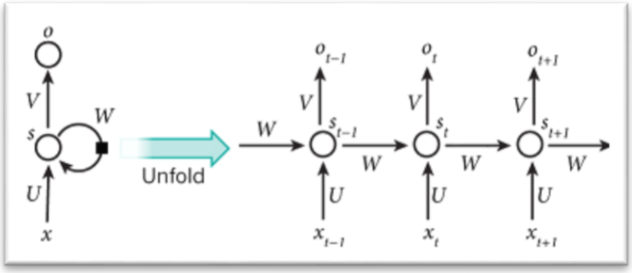

FIGURE 5: Recurrent Neural Network (RNN) (Lecun et al., 2015).

LSTMs were introduced to solve the problems in traditional RNNs (Hochreiter and Schmidhuber, 1997). Gating mechanisms are introduced in the LSTM units to enable the model to decide whether to accumulate certain information or not. There are three gates, input gate, forget gate and output gate. The gradient can now flow through these gates, and only the useful information can flow in and store in the state.

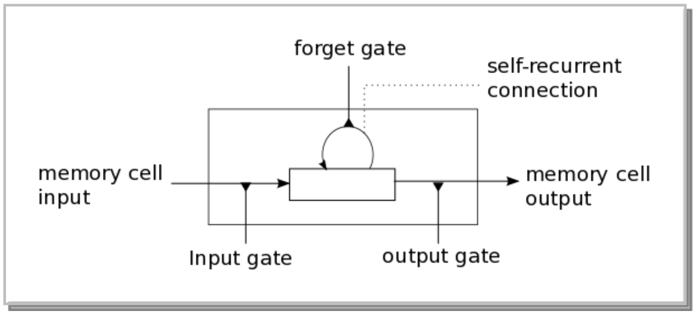

FIGURE 6: Long Short-Term Memory (LSTM) Memory Cell

LSTM (long short-term memory) as a variant of RNN have become the model of choice due to its ability to handle long-range dependences in a time sequence. It is thus chosen as the type of neural network used for this project. In this paper, the training and testing are done with a Python library TensorFlow developed by Google. Three weeks' concentration is selected as a training set. During these three weeks there was no precipitation, so the training set has a good representation of the normal daily pattern. The daily pattern of normal dry days will be learned and stored in the LSTM networks as a predictable model. The predicted or expected ammonia concentration at any time is estimated based on its preceding values and the predictable model. The prediction error is calculated as the offset between the predicted and actual values. The calculation for the anomaly score is done with The Numenta Platform for Intelligent Computing (NuPIC) (Lavin et al., 2015). Like the probability distribution discussed above, the anomaly score is ranked based on the likelihood of the prediction error. One or more thresholds can be set for the anomaly score.

Datasets

To evaluate the performance of algorithms, the datasets and the anomalies they contain need to be analyzed first. The ammonia dataset is collected by sensors which take reading every minute.

In terms of the duration of anomalies, the ammonia anomalies can be point anomalies such as Anomaly-1 in Figure 6 or anomalies sections such as Anomaly-2 and Anomaly-3. Points anomalies are the single instance deviated from the typical pattern. Different reasons can cause point anomalies, and they are generally random events. A point anomaly is momentary and usually does not affect the operation too much. The detection in this project focuses on continuous sections of anomalies rather than point anomalies. The performance of algorithms is evaluated based on their ability to detect anomalous sections instead of point anomalies. In the following, anomalous sections will be referred to as anomalies.

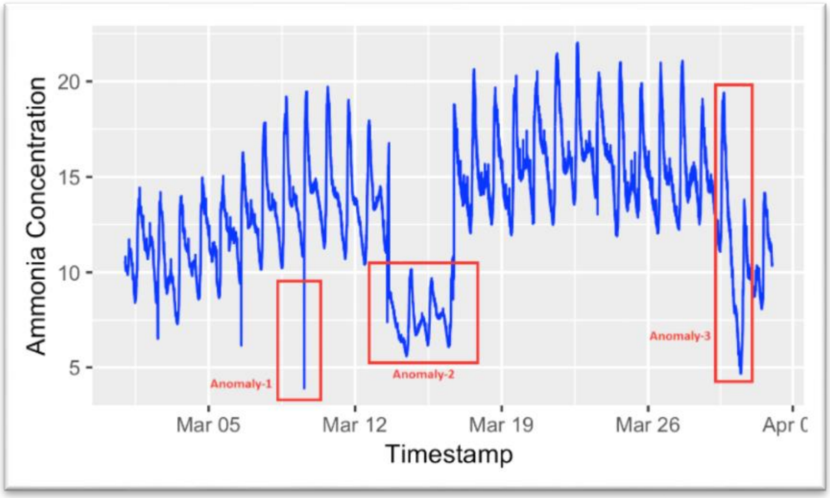

FIGURE 7: Ammonia Concentration in March 2018.

The ammonia concentration depends on both the ammonia load and wastewater load. The daily ammonia load comes from the industry and residence and has its daily pattern. The ammonia load is relatively stable regardless of the weather condition. However, the wastewater load is heavily affected by the weather, especially precipitation. The ammonia concentration thus depends on the precipitation. The concentration may become abnormal compared to the dry days when there is precipitation. For example, for Anomaly-3 of Figure 7 that occurred at the end of March, the abnormal concentration dropped gradually and came back to the normal range in the next day. We are not so concerned about these anomalies as they are still normal events. The anomalies caused by other factors such as sensor faults are of more interest. Anomaly-2 was observed from March 13th to 16th when a wrong calibration was made, creating readings of only half of the actual values. In situations like this, an alarm should be sent when the non-precipitation anomalies are detected. Thus, after all anomalies are detected, "real" anomalies need to be eliminated based on the precipitation information.

The most straightforward way to distinguish Anomaly-2 from Anomaly-3 is to consult the precipitation data. Any concentration anomalies which occur in a rainy period is more likely to be caused by precipitation. The precipitation dataset is downloaded from the Environment Canada website. However, precipitation data can only be used to explain the anomaly but not detect an anomaly with the algorithm for several reasons. Firstly, the precipitation is available on a daily basis, which is not enough to feed into the neural networks. Secondly, there is a distance between the weather station and the WWTP. The increased wastewater amount may not be accurately reflected by the precipitation.

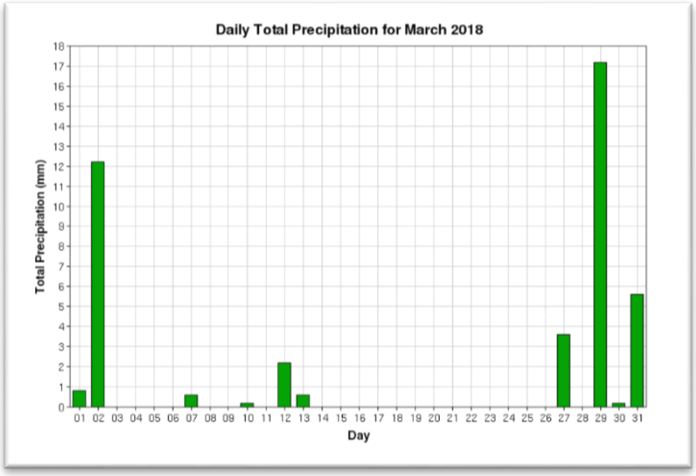

FIGURE 8: Precipitation in March 2018.

The next option is to use the flow provided by WWTP, which contains readings every 15 minutes. The flow dataset also has a daily pattern, while an anomaly can be observed when there is precipitation (March 2nd and March 28th). However, flow data is not available, so it is not a perfect substitute for precipitation data.

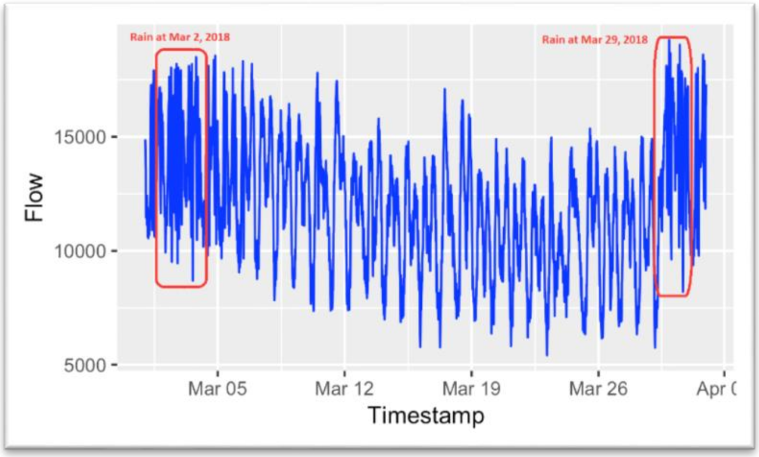

FIGURE 9: Flow data in March 2018

The temperature data is usually available as it is attached to the ammonia sensor, which also takes reading every minute. It can also reflect the precipitation indirectly. For example, in the winter, the temperature of the freezing rain is much lower than that of the wastewater. When there is rainfall, there will also be an anomaly in the temperature.

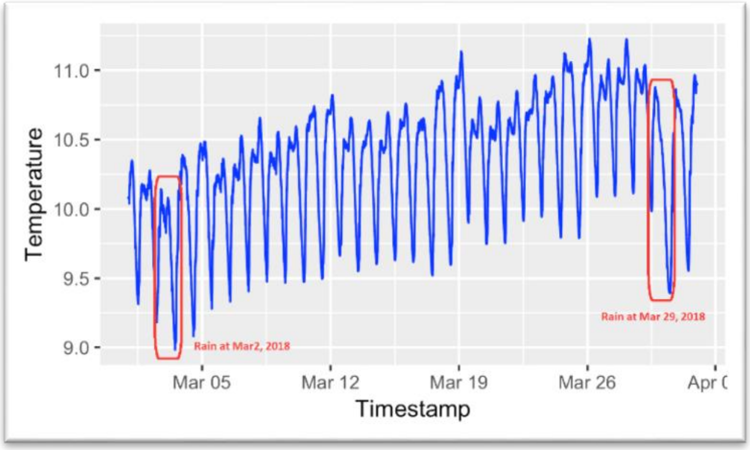

FIGURE 10: Temperature data in March 2018.

As a summary, there are four datasets – concentration, precipitation, flow and temperature in total. Concentration, temperature and weather data contain reading from January 2018 to June 2018. Flow data contains readings from January 2018 to March 2018. Weather data and flow data are collected and published by industry experts with a high standard quality control. We assume there are no faulty data caused by mechanical failures or operation errors.

We went through the ammonia concentration manually and identified that there are 11 anomalies. Nine of 11 anomalies can be explained and evidenced by the other datasets. The remaining two anomalies are the “real” anomalies. For the “real” anomaly in March, the wrong calibration caused the concentration reading is half of the expected values, while the temperature and flow data were not affected. The second “real” anomaly occurred at the end of May when the WWTP was pumped empty. The sensors were exposed to the air during that time, so both concentration and temperature showed extreme values. The flow data was not available for this time.

RESULTS AND DISCUSSION

The rule-based algorithms are used as the baseline. The S-H-ESD algorithm is implemented by the R library *AnomalyDetection* developed by Twitter. By specifying the length of the repeating pattern, the detected anomalies are indicated by green cycles. The beginning part of Anomaly-2 with rapid change is detected and circled by the green dots, but the major part is not detected. The precipitation caused anomaly cannot be detected.

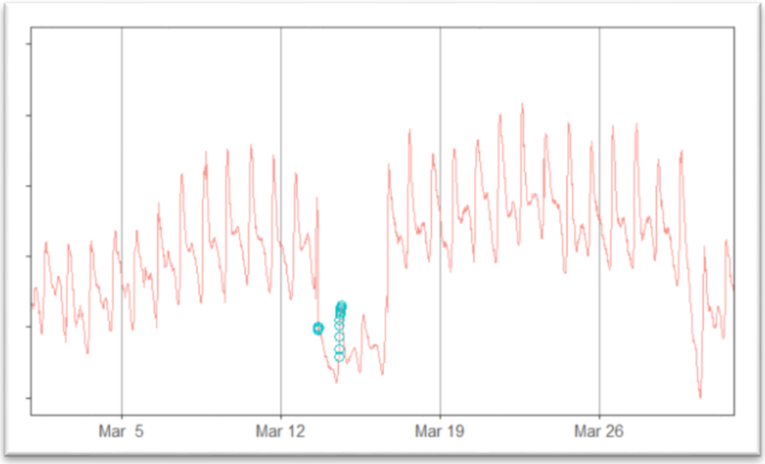

FIGURE 11: The detection result of the S-H-ESD algorithm

The derivative and moving average algorithms are both implemented by Luminol by LinkedIn. The anomaly score is calculated by combining the results from two algorithms. Figure 12 displays the anomaly score of concentration at March 2018. Similar to the S-H-ESD algorithm, only the beginning of the anomaly has a distinguishable anomaly score.

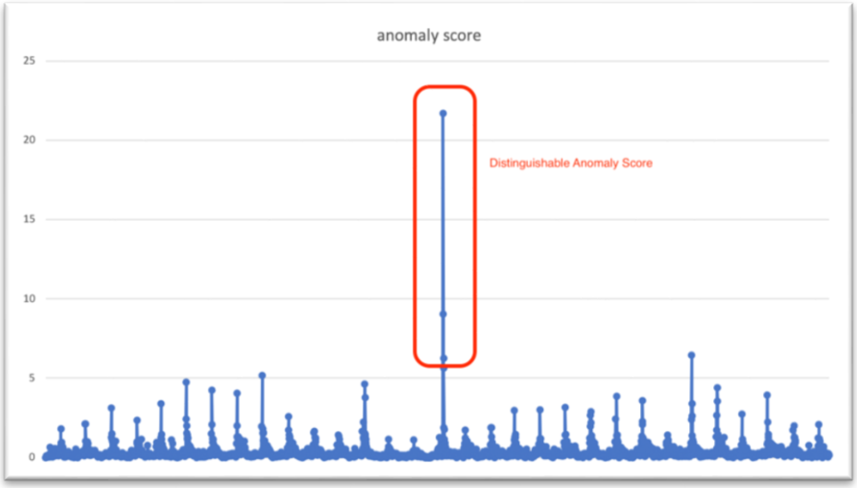

FIGURE 12: Detection Result of Derivative and Moving Average Algorithms.

The two rule-based algorithms can both detect an anomaly with rapid change, i.e., the beginning part of the anomaly. However, for this anomalous section lasting for three days, it means most of the faulty readings were still labeled as normal. In practice, this is easy to be confused with a point anomaly, as only a few points with a high score are detected.

The prediction errors are thus the offset between predicted values and actual values. The anomaly score or anomaly likelihood is calculated based on the probability of prediction errors, with 0.0 as the lowest likelihood and 1.0 as the highest likelihood. The threshold can be set according to the user. In this paper, the threshold is set such that any score higher than 0.5 is guaranteed to be an anomaly (highlighted with dark red), and any score between 0.4 and 0.5 are possible to be an anomaly (highlighted with light red). The two "real" anomalies are both detected.

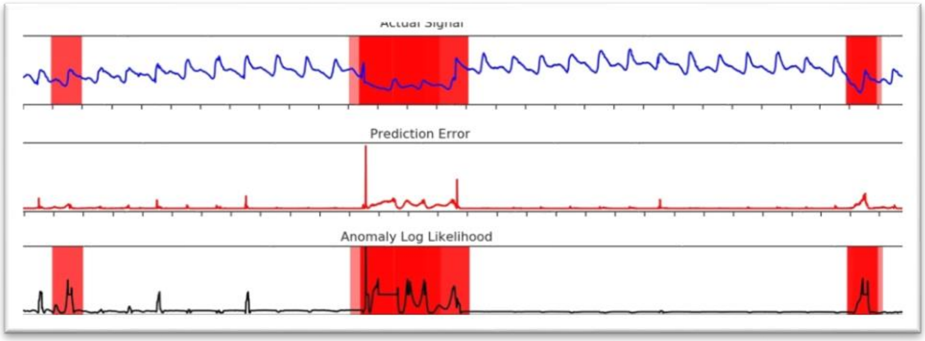

FIGURE 13: Detection Result for March 2018.

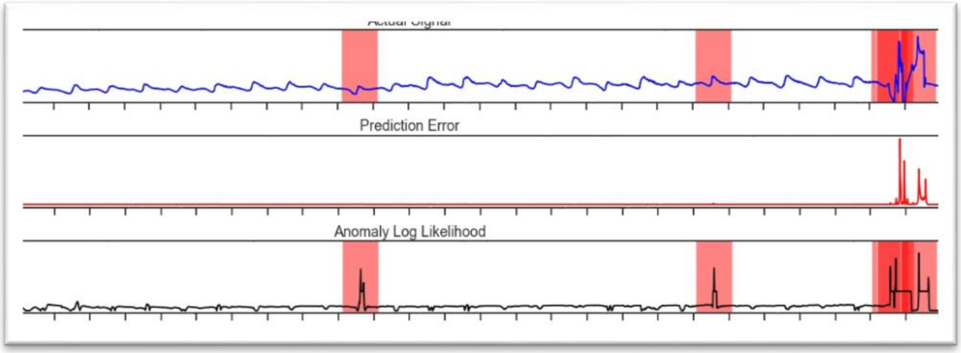

FIGURE 14: Detection Result for May 2018.

The comparison of different detection results are summarized in Table 1. The manual inspection shows that there are 11 anomalies, where 9 of them are caused by precipitation. The two "real" anomalies that are shown in Figure 13 and Figure 14, were both are successfully detected, with the whole anomalous sections being highlighted as the red zone.

The same data-driven approach is applied to the flow data and temperature datasets to eliminate the "real" concentration anomalies. In this paper, "real" anomalies are the two anomalies in Figure 13 and Figure 14.

TABLE 1: Comparison of three detection algorithms.

	S-H-ESD	Derivate & Moving Average	RNN with LSTM
True Positive	2 *	2 *	10
False Positive	0	0	1
False Negative	0	0	1

*: Only the beginning portion of anomalies are detected

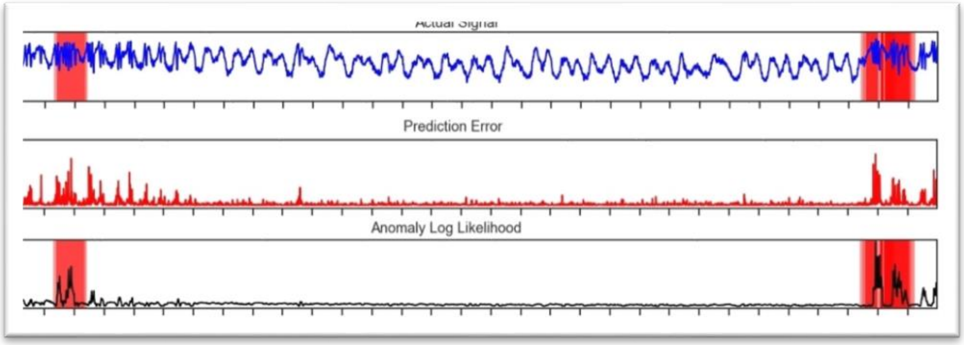

FIGURE 15: Detection Result of Flow for March 2018.

The flow data between January 2018 and March 2018 contains five anomalies, which were caused by five major precipitation events. Comparing the two detection results between concentration and flow, the period when both results contain anomalies are likely to be affected by rainfall. By eliminating concentration anomalies during these periods, the "real" anomalies can be eliminated. However, eliminating "real" anomalies with the flow data is usually not an applicable option in similar projects. Flow data is acquired by WWTP, and it is infeasible to measure it with simple sensors.

TABLE 2: Detection result for flow data.

	RNN with LSTM
True Positive	5
False Positive	1
False Negative	0

The temperature data is used as a substitute for flow data as the temperature sensor is attached to the concentration sensor. There are ten anomalies in the temperature dataset. The detection for temperature shows a less accurate result compared to that of the concentration and flow, as shown by the false positive and false negative results.

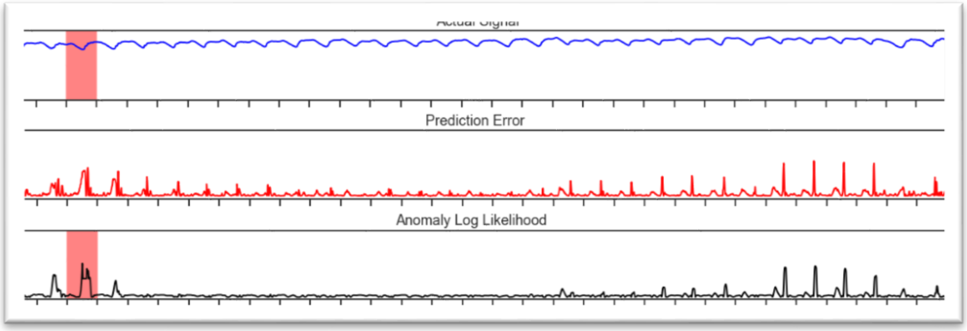

FIGURE 16: Detection Result of Temperature for March 2018.

TABLE 3: Detection result for temperature data.

	RNN with LSTM
True Positive	6
False Positive	2
False Negative	4

After further analyzing the results, it is found to be difficult to eliminate the "real" concentration anomalies purely based on temperature anomalies. In the summertime such as June, when the rain temperature is close to the wastewater temperature, the precipitation does not affect the temperature too much thus no temperature anomaly can be detected. The detected anomaly (normal rain event) in Figure 17 can be misinterpreted as "real" anomaly as it only appeared in concertation but not temperature.

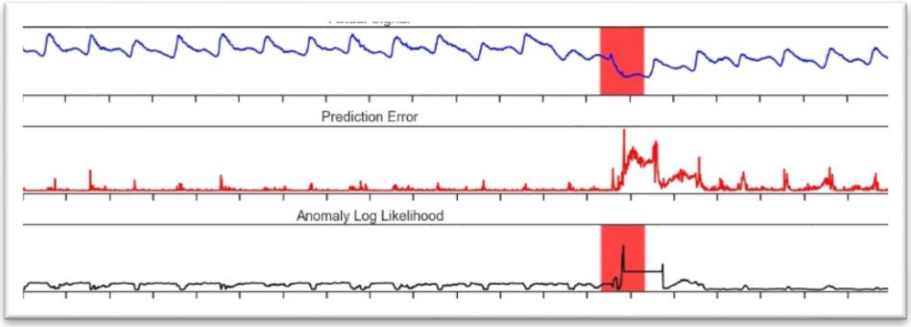

FIGURE 17: Detection Result of Concentration for June 2018.

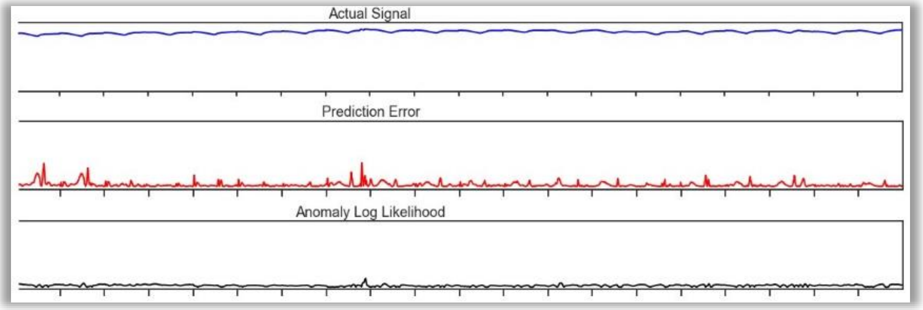

FIGURE 18: Detection Result of Temperature for June 2018.

The opposite situation may also happen, where “real” anomaly may be misinterpreted as a normal rain event. In Figure 14 and Figure 19, in the last two days of May when the sensors were exposed to the air, both concentration and temperature showed anomalies. In such a case, both anomalies are caused by precipitation, and they are both “real” anomalies. Eliminating concentration anomalies based on temperature will lead to the wrong result.

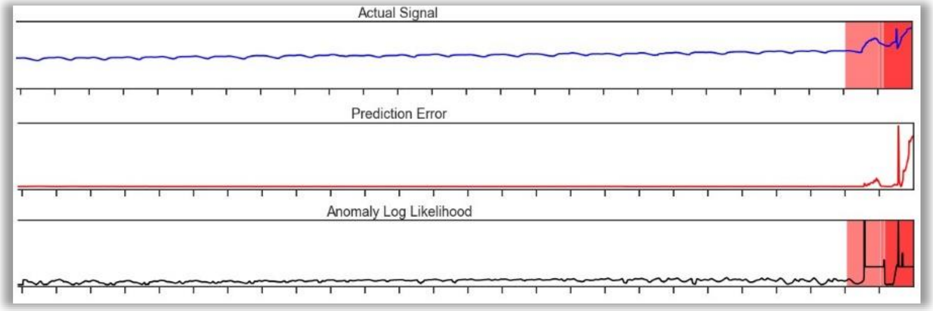

FIGURE 19: Detection Result for May 2018.

CONCLUSIONS

High-quality, high-frequency data is essential for successful process modeling and process control. In this paper, a data-driven approach which makes use of deep learning techniques was taken to solve a real-world ammonia concentration dataset. Two rule-based algorithms serve as the benchmark, where both algorithms detect anomalies based on the statistical features. The LSTM approach considers periodicity to distinguish the normal with the abnormal behaviors, with the predicted anomalous data flagged and qualitatively ranked based on the severity and likelihood that the data are faulty (i.e., good, maybe faulty, probably faulty, definitely faulty).The results show that the LSTM based algorithm outperform the rule-based algorithm, where ten out of 11 anomalies can be detected with only one false positive. Both “real” anomalies were successfully detected. Further elimination of the “real” anomalies was then attempted with the flow and temperature datasets. The results show that temperature is not a perfect substitute for flow data. In practice, some water quality datasets may be needed to fully eliminate the impact of precipitation. The algorithms have been successfully applied to well-maintained sensor signals and are now being tested with poorly maintained sensors to judge their suitability in a real-world application.

ACKNOWLEDGEMENT

We would like to show our gratitude to Global Water Futures and Southern Ontario Water Consortium for their support for this research paper.

BIBLIOGRAPHY

Rosner, B. (1983). Percentage points for a generalized ESD many-outlier procedure. *Technometrics*, *25*(2), 165-172.

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, *2*(1-3), 37-52.

Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In Acoustics, speech and signal processing (icassp), 2013 ieee international conference on (pp. 6645-6649). IEEE.

Vallis, O., Hochenbaum, J., & Kejariwal, A. (2014, June). A Novel Technique for Long-Term Anomaly Detection in the Cloud. In HotCloud.

Linkedin. (2018, January 09). Linkedin/luminol. Retrieved from https://github.com/linkedin/luminol

Pascanu, R., Mikolov, T., & Bengio, Y. (2012). Understanding the exploding gradient problem. CoRR, abs/1211.5063.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436.

Lavin, A., & Ahmad, S. (2015, December). Evaluating Real-Time Anomaly Detection Algorithms--The Numenta Anomaly Benchmark. In Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on (pp. 38-44). IEEE.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.