

Detecting Anomalies in Irregular Data Using K-means Clustered Signal Dictionary

G. Talavera Reyes, Rajan M. Chandra, Ha Thu Le and Zekeriya Aliyazicioglu

Electrical and Computer Engineering Department, California State Polytechnic University Pomona, CA 91768, United States

Abstract: The critical nature of satellite network traffic provides a challenging environment to detect intrusions. The intrusion detection method presented aims to raise an alert whenever satellite network signals begin to exhibit anomalous patterns determined by Euclidian distance metric. In line with anomaly-based intrusion detection systems, the method presented relies heavily on building a model of “normal” through the creation of a signal dictionary using windowing and k-means clustering. The results of three signals from our case study are discussed to highlight the benefits and drawbacks of the method presented. Our preliminary results demonstrate that the clustering technique used has great potential for intrusion detection for non-periodic satellite network signals.

Key words: Intrusion detection, irregular data, K-means clustering, machine learning, signal dictionary.

1. Introduction

IDS (intrusion detection system) has been receiving much focus in the past decade. The possibility of an intrusion is now thought of as inevitable in many important sectors such as the Internet, wireless communication systems, power networks, industrial control systems, and so on. Therefore, it is imperative that robust and reliable methods for detecting intrusions are developed.

ID (intrusion detection) may be classified into two categories: statistical anomaly detection and rule-based detection [1]. Statistical anomaly detection seeks to build a model of normal or expected behavior. Rule-based detection, also referred to as signature

detection, defines a set of rules and patterns of attacks which are then used to determine if the system is being attacked.

Statistical anomaly based techniques have the advantage of detecting unforeseen intrusion patterns. However, this technique also has the drawback of classifying behavior not previously seen as malicious. Events that an IDS misclassifies as intrusions are called false positives. An effective IDS aims to minimize false positives. Rule-based techniques are limited to be able to detect intrusions based on known malicious patterns.

Anomaly detection can also be referred to as novelty detection in the literature. Several surveys have been published detailing the different methods developed for anomaly detection [2-5]. For instance, Garcia-Teodoro et al. [3] describe anomaly detection as “one-class classification” where the training data are used to build a model of “normal” data. This method is typically used when the quantity of available “abnormal” data are insufficient to construct explicit models. Furthermore, it classifies anomaly detection techniques into three main categories: statistical based, knowledge-based, and machine learning-based. These three techniques have their

Corresponding author:

G. Talavera Reyes, graduate student, research fields: machine learning and intrusion detection.

R. M. Chandra, Ph.D., professor, research fields: parallel processing and out of order execution, AI, machine learning, branch prediction, optimization techniques, and numerical analysis.

H. T. Le, Ph.D., assistant professor, research fields: solar power, smart grid, energy storage, power system stability and protection, demand response, energy conservation, distributed generation, optimization, and power quality.

Z. Aliyazicioglu, Ph.D., professor, research fields: radar signal processing, digital image processing, digital communication, unmanned aerial and unmanned ground vehicles.

advantages and disadvantages.

Statistical-based systems have the benefit of not requiring prior knowledge of normal activity and typically yield accurate notification of malicious activities. However, it is susceptible to be trained by attackers, and it is difficult to set parameters and metrics. Univariate and multivariate models as well as time series are typically statistical-based subtypes. Knowledge-based techniques tend to be robust, flexible and scalable. Unfortunately, it is difficult and time-consuming to maintain high-quality data. Finite state machines, description languages, expert systems and Bayesian networks fall under the knowledge-based technique category. Machine learning techniques draw particular interest as they are flexible and adaptable. Machine learning techniques tend to require a large number of resources and are highly dependent on assumptions about acceptable system behavior. Machine Learning techniques include Markov models, neural networks, fuzzy logic, genetic algorithms, clustering and outlier [2, 3].

A study in Ref. [6] presents various anomaly detection techniques specifically for time series data. Five detection techniques are covered in detail and compared: window based, proximity based, prediction based, Hidden Markov Models based, and segmentation based. Window based techniques divide the time series into fixed sized windows. When using a window based technique the size of the window is of utmost importance because it must be able to capture the anomaly. Proximity based techniques use distance or similarity measures to determine anomalies using mainly k-nn (k nearest neighbor) and clustering. The distance from the test time series to its k^{th} nearest neighbor in the training data set is used to determine whether there is an anomaly. In clustering, the training time series is clustered and the centroids are computed. This is followed by determining the distance between the test time series and its closest cluster centroid to determine anomalies.

An adaptive machine learning model for anomaly

detection is presented in Ref. [7]. The basis for anomaly detection is to establish what is normal and then compare new events to that pattern or model. To build the “normal” model accurately, the researchers use a machine learning model, a large enough sampling of data to get an accurate representation and the ability to define what forms a regular pattern in the training data.

The researchers in Ref. [7] analyzed the regular frequency of a normal beating heart, recorded using an electrocardiogram (EKG). An EKG shows the heart’s electrical activity via voltage pulses which are highly similar to one another. Substantial changes in the shape of the waveforms are indicative of physiological problems or equipment malfunction. Their goal was to build a model of what heartbeats should look and then use the model to detect irregular patterns in observed heartbeats. Part of the process of discovering the normal patterns necessitates expert insight into the system domain. For the EKG case, a doctor is essential to be able to determine whether a waveform observed is anomalous or normal.

Their approach for building a model of normal heartbeat behavior included windowing and clustering. Windowing involves extracting short sequences of the “normal” signal in such a way that the short sequences can be added back together to recreate (i.e. reconstruct) the input signal. EKG curves have repeated components separated in time. They exploited the similarity between the repeated components to build a model of “normal” heartbeat activity by aligning and clustering all of the short sequences being observed in a long recording.

Clustering all the signals resulted in a catalog of essential heartbeat waveforms (i.e. a dictionary of component waveforms) that are typical for normal heart behavior. They then applied the technique to an input test signal, compared it to the “normal” model and noted the level of error between the two. Low levels of errors between the re-constructed signal and the ideal suggest that the observed signal is close to

normal (i.e. not an anomaly) and vice versa.

They found that, as long as the input signal mimics the waveforms found in the signal dictionary closely, the reconstruction will be very good and the reconstruction error will be small. In essence, the dictionary is a model of what “normal” EKG signals can look like and the reconstruction error represents the degree to which the test signal approximates a healthy heartbeat. This technique produced surprisingly good results given its low level of complexity.

In this study, we aim to extend the aforementioned method to detect anomalies in complex, irregular, non-periodic signal data. The challenge of analyzing the signals of this type is, unlike the EKG signal, very difficult. The difficulty arises in identifying typical or similar patterns in the training signals to build the model of “normal” signal behavior. The following sections present an intrusion detection algorithm developed for irregular and non-periodic data, and a case study where the algorithm is applied.

2. K-means Clustering Algorithm

K-means clustering is based on the minimization of a performance index which is defined as the sum of the squared distances from all points in a cluster domain to the cluster center [8]. The steps listed below encompass the entire procedure:

Step 1: K initial cluster centers are chosen $c_1(1), c_2(1), \dots, c_k(1)$. The seeding process used in this experiment is explained in detail in section IIA.

Step 2: At the k^{th} iterative step the samples $\{x\}$ are distributed among the K cluster domains, using the relation,

$$x \in S_j(k) \text{ if } \|x - z_j(k)\| < \|x - z_i(k)\| \quad (1)$$

for all $i = 1, 2, \dots, K, i \neq j$, where $S_j(k)$ denotes the set of samples whose cluster center is $z_j(k)$ ties in Eq. (1) are resolved arbitrarily.

Step 3: From the results of Step 2, new cluster centers $z_j(k+1), j = 1, 2, \dots, K$, are computed such that the performance index, Eq. (2), is minimized. The

performance index is the sum of the squared distances from all points in $S_j(k)$ to the new cluster center.

$$J_i = \sum_{x \in S_j(k)} \|x - z_j(k)\|^2, j = 1, 2, \dots, K \quad (2)$$

The $z_j(k+1)$ which minimizes this performance index is simply the sample mean of $S_j(k)$. Therefore, the new cluster center is given by Eq. (3).

$$z_j(k+1) = \frac{1}{N_j} \sum_{x \in S_j(k)} x, j = 1, 2, \dots, K \quad (3)$$

where N_j is the number of samples in $S_j(k)$.

Step 4: If $z_j(k+1) = z_j(k)$ for $j = 1, 2, \dots, K$, the algorithm has converged or if the maximum of iterations has been reached, the procedure is terminated. Otherwise, steps two through four are repeated.

2.1 Initializing Cluster Centers

The steps below show how the initial centroids are chosen. They are based on an in-depth analysis in Ref. [9] where a modified algorithm, K-means++, is explained. The modification improves both speed and accuracy of k-means by augmenting the initial randomized seeding.

Step 1: The seeding starts with the random selection of an observation from the data set, this becomes the first centroid, c_1 .

Step 2: The distance from each observation to c_1 is computed. The distance between c_j and the observation m is denoted as $d(x_m, c_j)$.

Step 3: The next centroid, c_2 is selected at random from the data set with probability given by Eq. (4).

$$\frac{d^2(x_m, c_1)}{\sum_{j=1}^n d^2(x_j, c_1)} \quad (4)$$

Step 4: The j^{th} centroid is chosen by computing the distances from each observation to each centroid, then assigning each observation to its closest centroid. This is followed by selecting centroid j at random from the data set with probability given by Eq. (5).

$$\frac{d^2(x_m, c_p)}{\sum_{\{h, x_h \in C_p\}} d^2(x_h, c_p)} \quad (5)$$

where C_p is the set of all observations closest to

centroid c_p . x_m belongs to C_p .

Step 5: Step 4 is repeated until k centroids have been chosen.

3. Intrusion Detection Algorithm for Irregular, Non-periodic Signal Data

The algorithm developed to detect intrusions in irregular data is a combination of different techniques in the literature. The concept of a signal dictionary as a model of “normal” is adapted from Ref. [7] with the modification to use the Euclidian distance measured between an off-nominal test signal (i.e. the signal with possible anomalies) and the reconstructed signal to classify intrusions. Fig. 1 shows an overview of the intrusion detection algorithm.

Step 1: Preprocessing of data

The training (i.e. normal) data are divided into windows of length t_1 seconds. The window size t_1 is determined by pre-processing the training data using a trial-and-error technique.

Step 2: Building a signal dictionary

K-means clustering is applied to the set of training windows, using k cluster centers. Once the clustering

algorithm is complete, this becomes the signal dictionary with the cluster centers acting as keys.

Step 3: Reconstructing test signal

The test signal is divided into windows of length t_1 seconds. The test signal is reconstructed by first calculating the Euclidian distance between the test window and all the cluster centers. The cluster center with the smallest Euclidian distance is selected, c_{sd} . All the windows assigned to cluster center, c_{sd} , are compared to the test window. The window which minimizes the Euclidian distance to the test window is selected as the reconstruction window. The reconstructed signal is composed of the reconstruction windows.

Step 4: Distance determination

The distances between the test signal windows and the reconstructed signal windows are denoted $D(W_{test}, W_{Rec})$. The threshold is set to the mean of all the distances, $Mean(D(W_{test}, W_{Rec}))$.

Step 5: Flagging anomalies

An anomaly alert is raised if the distance calculated exceeds the mean n times within t_2 minutes.

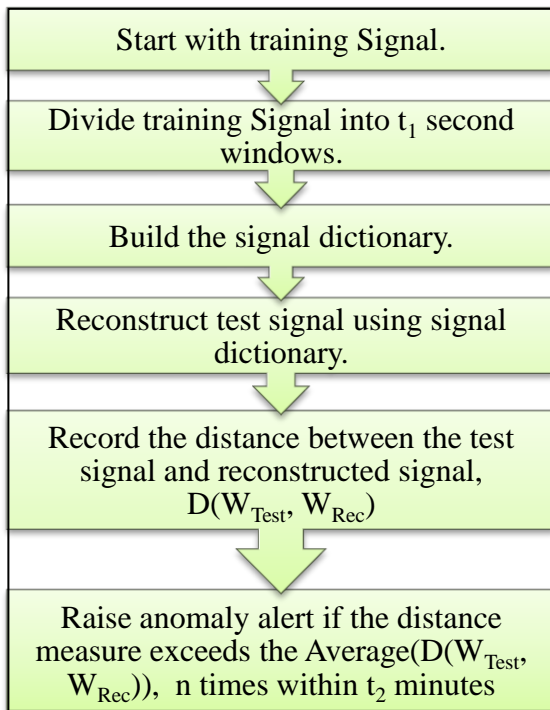


Fig. 1 Intrusion detection algorithm overview.

4. Case Study

In this case study, the developed algorithm is evaluated using two large data sets provided by NGC (Northrop Grumman Corporation). One data set contains nominal data (i.e. correct data with NO anomalies) and the other set contains off-nominal data (i.e. data potentially containing anomalies). The challenge is to determine the location of anomalies, if any, in the off-nominal data set with minimal domain knowledge.

The data sets are simulated satellite decoded messages/numeric data between the BC (bus controller) and RT (remote terminals) sampled from the network at 8 Hz. The nominal data contain 2.67×10^4 seconds of data, around 7.4 hours. The off-nominal data contain approximately 4.2×10^4 seconds worth of data, about 11.7 hours. Both data sets contain time-series data for 54 signals.

The developed algorithm was applied per signal. The nominal signal was divided into 8-second windows. The K-means clustering algorithm was then applied to the segmented signals using $k = 1,500$. The corresponding off-nominal signal was then used as the test signal. The test signal was reconstructed by finding the windows in the dictionary closest to the test windows. The distances between test windows and selected dictionary window were computed. The Euclidean mean of all the window distances was then used as a metric to detect anomalies.

An anomaly alert was raised when 1,700 windows exceeded the mean in a 12 minute period. These parameters were selected through trial-and-error analysis of BC to RT10 signal results. In an effort to test whether the counter and time interval parameters could be globally set for the whole system the aforementioned parameters were applied to all the signals. The obtained results for different signals between the 7 RT and the bus controller BC are presented in the following sections.

4.1 Results for RT2 to BC Signal

The signal from remote terminal 2 to the bus

controller resulted in raising the most alerts. Fig. 2 shows 3 graphs: the graph of the nominal signal, the graph of the off nominal signal, and lastly when anomaly alerts were triggered. Fig. 3 shows the distribution of the distances between the reconstructed windows and the off nominal windows as well as the mean of the distances. These particular results demonstrated the need to select parameters based on the distribution of distance results. The recurrence of high distance values periodically could mean that it is normal signal behavior and should be filtered, but with zero domain knowledge it cannot be conclusively determined.

4.2 Results for BC to RT10

Fig. 4 shows the results for the signal from the bus controller to the 10th remote terminal. The graph only shows one alert at the beginning of the simulation and a second towards the end. These are very promising results since the alerts almost perfectly align with the small disturbances that can be observed in the off-nominal signal. Fig. 5 shows the distribution of the distances as well as the mean. Inspecting the high concentrations occurring in both the beginning and end

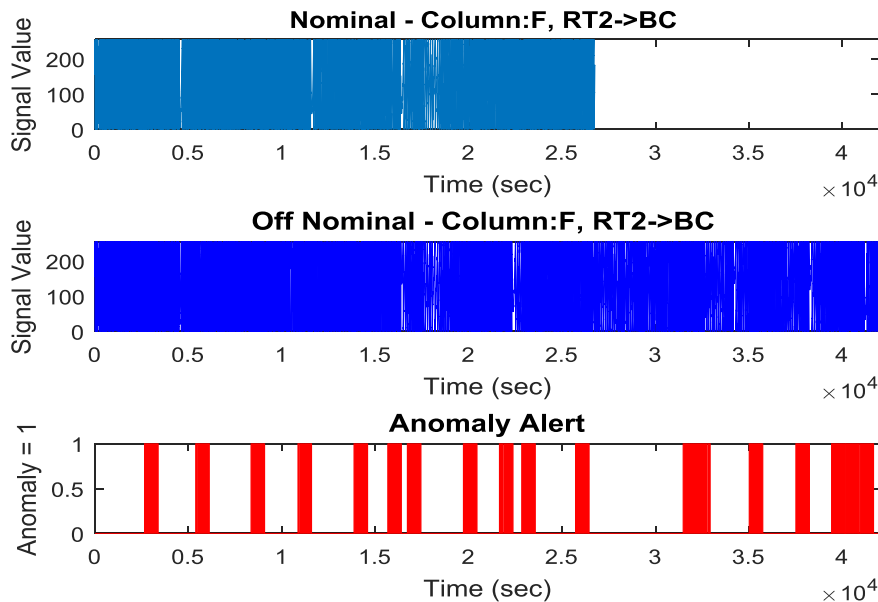


Fig. 2 Anomaly alert results for RT2 to BC.

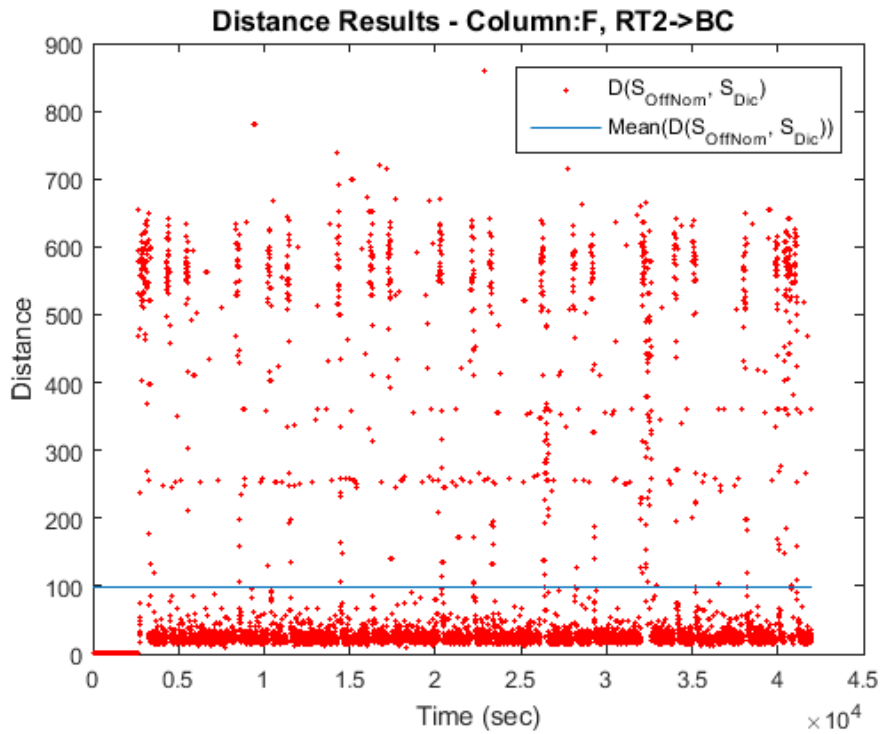


Fig. 3 Distances between reconstructed windows and off nominal windows for RT2 to BC.

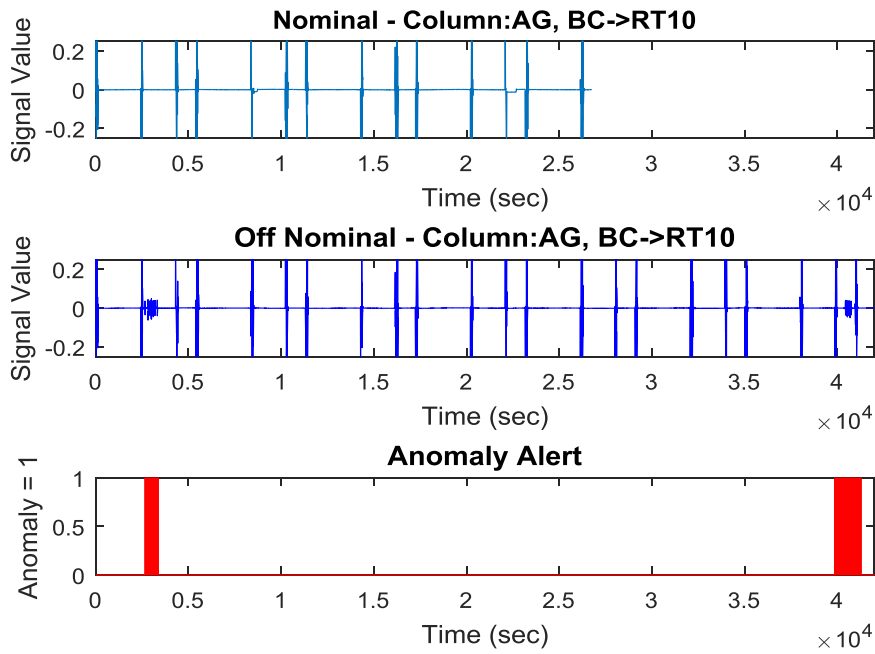


Fig. 4 Anomaly results for BC to RT10.

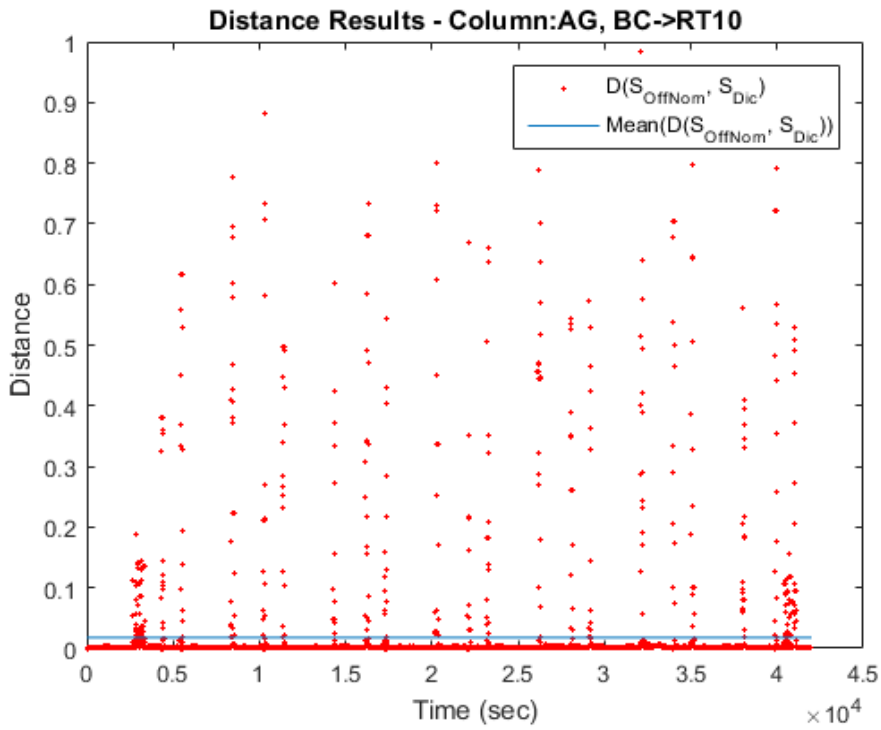


Fig. 5 Distances between reconstructed windows and off nominal windows for BC to RT10.

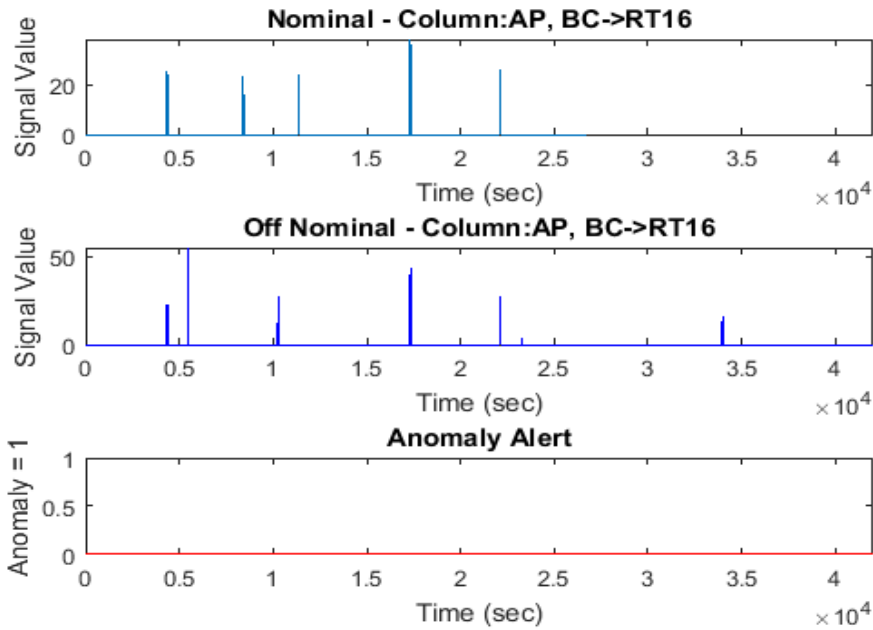


Fig. 6 Anomaly results for BC to RT16.

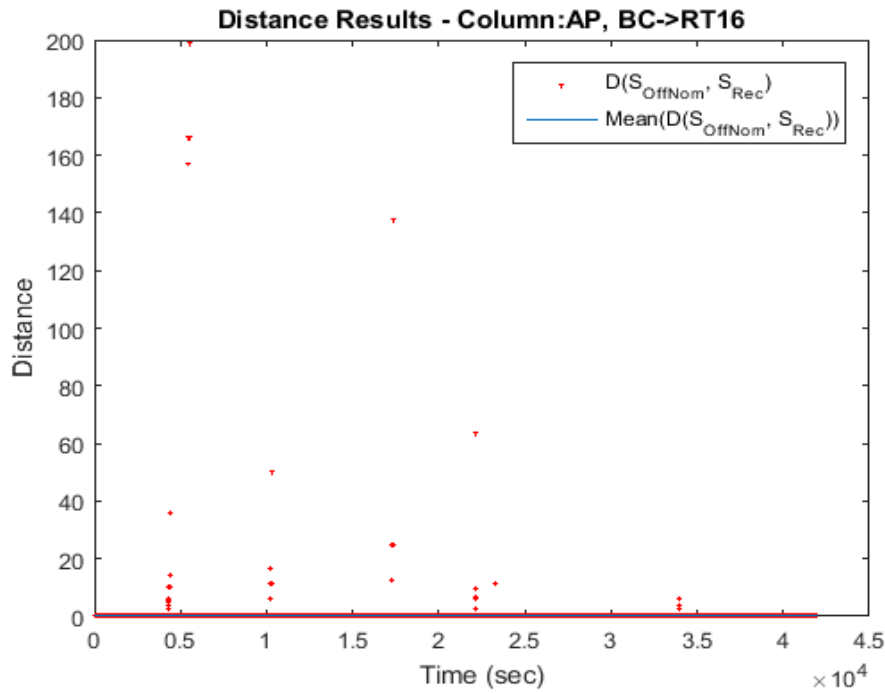


Fig. 7 Distances between reconstructed windows and off nominal windows for BC to RT16.

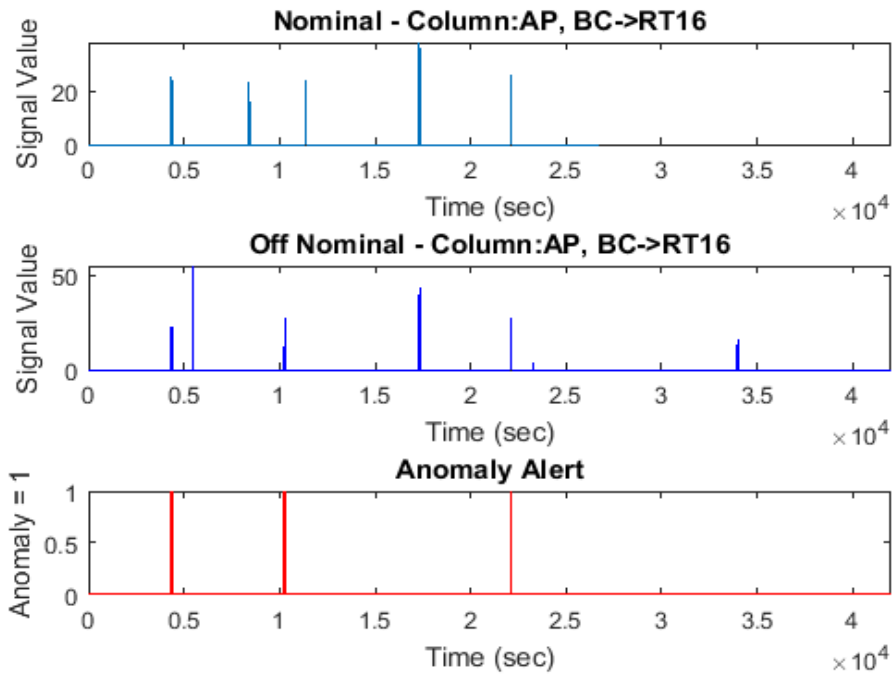


Fig. 8 Anomaly results for BC to RT16 with adjusted parameters.

of the simulation resulted in the selection of 1,700 windows and 12 minute interval as parameters for raising an anomaly alert.

4.3 Results BC to RT16

The results of each of the BC to RT16 signals were

very similar to those of Fig. 6. This particular signal illustrates a major drawback to utilizing the same count and time-interval parameters to determine anomalous behavior for all signals. Off-nominal BC to RT16 signal has a total of 1,712 points greater than zero sparsely scattered throughout the 11.7 hours of

data, see Fig. 7. The parameters used to determine anomalies for the signals previously analyzed were set to count at least 1,700 instances that exceed the mean in 12 minutes. It is no surprise then to find that these parameters fail to detect any anomalies for all of BC to RT16 signals.

However, the number of windows and the time parameters can be tailored to better detect anomalous behavior in BC to RT16 signals. The window count was lowered to 300 instead of the 1,700 applied previously. The time parameter was also decreased to one minute instead of 12 minutes. These new parameters are a better fit for this signal with sparse data. Fig. 8 shows the effect of the new parameters on a BC to RT16 signal. The new parameters resulted in greater detection sensitivity.

5. Conclusion

The goal of this research was to develop an adaptive algorithm that could in theory be fine-tuned with expert domain knowledge. It proved to be quite a challenge to develop an algorithm that would detect anomalies for a system where there is virtually zero domain knowledge.

Much of the success of anomaly-based intrusion detection systems is highly dependent on developing a robust model of normal data. Theoretically, the higher the amount of normal data available to train the anomaly-based IDS, the higher the potential to yield better detection rates and less false positives. Even though the case study presented had less normal data than the test signal, the results demonstrate that the method applied can highlight areas in the test signal that do not match what is stored in the signal dictionary. The results also served to highlight the importance of domain knowledge. A better understanding of the expected signal patterns would allow for better parameter selection and in turn improved results. The method presented has great

potential to yield better results with greater domain knowledge and more normal data.

The next step in this research is to find a method to remove the window counts and time intervals. Exploring Kohonen's self organizing map to highlight anomalies in the data could be a possible alternative. Using a self organizing map has the potential to eliminate the need for the parameters to count suspicious events in a given period of time as a trigger to raise an anomaly alert.

Acknowledgement

The authors wish to thank Northrop Grumman Corporation for providing a gift to support this research.

References

- [1] Stallings, W. 2011. *Cryptography and Network Security Principles and Practice*. New York: Pearson.
- [2] Chadola, V., Banerjee A., and Kuma, V. 2009. "Anomaly Detection: A Survey." *ACM Computer Surveys (CSUR)* 41 (3): 1-58.
- [3] Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., and Vazquez, E. 2009. "Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges." *Computers and Security* 28 (1-2): 18-28.
- [4] Patcha, A., and Park, J. 2007. "An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends." *Computer Networks* 51 (12): 3448-70.
- [5] Pimentel, M., Clifton, D., Clifton, L., and Tarassenko, L. 2014. "A Review of Novelty Detection." *Signal Processing* 99: 215-49.
- [6] Cheboli, D. 2010. "Anomaly Detection of Time Series." Master Thesis, University of Minnesota, Minneapolis.
- [7] Dunning, T., and Friedman, E. 2014. *Practical Machine Learning: A New Look at Anomaly Detection*. Sebastopol: O'Reilly Media, Inc.
- [8] Gonzalez, R., and Tou, J. 1974. *Pattern Recognition Principles*. Reading: Addison-Wesley Publishing Co.
- [9] Arthur, D., and Vassilvitskii, S. 2007. "K-means++: The Advantages of Careful Seeding." In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Philadelphia.