International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

IoT Anomaly Detection using Multivariate

Soundararajan Ezekiel, Abdullah Ali Alshehri, Larry Pearlstein, Xin-Wen Wu, Adam Lutz

Abstract*: Devices associated with Internet of Things are typically constrained in their resources and do not have the computational power necessary to analyze their input and detect anomalies that occur. Smart devices or and environmental sensors that measure temperature, air quality, or seismic activity are all built for specific purposes with minimal resources and often do not have enough security in place to protect against infiltration or detect abnormal behavior. Additionally, because these devices and sensors are typically always connected and transmit constant data in near real-time, the high dimensionality of the raw readings are extremely computationally intensive to analyze. A possible solution to reduce the dimensionality of the data while also extracting the most significant features is to use multivariate analysis techniques such as Principal Component Analysis. PCA is a method of multivariate analysis meant to reduce the size of matrices while not only keeping the most significant variables but also learning the interactions between them. In this paper, we propose exploring anomaly detection in IoT using multivariate analysis techniques to reduce the dimensionality of sensor input to reduce the computational complexity of analysis and learning the most significant variables. While the normal conditions of sensor data are often readily available, the size of the data makes it difficult to precisely determine instances of targeted anomalies. In this study, PCA is used to analyze the available features of the data and from them can determine the sensors under normal conditions. Once the normal conditions are determined, outliers which constitute anomalies can be determined through techniques such as Mahalanobis distance to determine the variance of each observation from the normal distribution. Our work can also be expanded to use other methods of dimensionality reduction and feature extraction such as t-Distributed Stochastic Neighbor Embedding.

Keywords* : Internet of Things, Anomaly Detection, Multivariate Analysis, Principal Component Analysis, Dimensionality Reduction.

I. INTRODUCTION

In recent years, devices and sensors associated with Internet of Things (IoT) have seen extensive use in areas such as smart devices, home automation, healthcare devices such as heart and insulin monitors, and environmental sensors that measure temperature, air quality, or seismic activity often do not have the resources to efficiently implement security measures to deter intrusions or other abnormalities that might occur within sensor networks and devices. IoT devices are often built with specific tasks in mind and are minimal in terms of computational power. Due to this, when abnormalities occur within devices, they often begin functioning erroneously without the ability to take corrective actions independently. Within the past decade, as recently as 2016 onward, there have been numerous large-scale attacks that specifically targeted resource constrained, low security devices, including malware that infected unsecure IoT devices to serve in botnets that have been used in multiple destructive DDoS attacks [5, 6, 7]. Devices that had been infected primarily consisted of smart home devices such as doorbell cameras, smart fridges, etc. that typically ran lightweight distributions of Linux that still had default credentials. Although infected devices became part of botnets that were used for destructive attacks, other than some increased network usage and occasional performance drops, they continued to perform the functions they were designed for and did not outwardly exhibit signs of infection. Because of the computational limitation and other security flaws of these devices, developing anomaly detection techniques that can identify minute changes in the sensor readings of devices is necessary for the safety of resource constrained devices. In this study, we investigate the use of multivariate analysis techniques such as principal component analysis (PCA) to analyze the outputs and other readings of IoT devices and detect when anomalies occur. Principal component analysis is a widely used method of feature extraction and dimensionality reduction of large multivariate datasets [8, 9]. It is a technique that is used to reduce the dimensionality of large-scale matrices while still retraining the most significant information about the dataset. When PCA is applied to a larger matrix, its correlated variables are reduced into uncorrelated variables known as principal components, which retain the most significant features of the dataset while also learning the interactions between the variables. Through reducing the dimensionality of large datasets, it prevents the overfitting of models by eliminating insignificant features as well as reducing the computational complexity of analyzing the data. The principal components are ordered in such a way that the most significant principal component is first and is the component with the largest variance that encompasses most of the data. Each of the principal components are linearly independent as they are all orthogonal from each other.

Revised Manuscript Received on February 28, 2020.
***** Correspondence Author
Soundararajan Ezekiel*, Computer Science, Indiana University of Pennsylvania, Indiana, Pennsylvania, USA. Email: ezekiel@iup.edu.
Abdullah Ali Alshehri, Electrical Engineering, King Abdulaziz University, Rabigh, Saudi Arabia. Email: ashehri@kau.edu.sa.
Larry Pearlstein, Electrical Engineering, The College of New Jersey, Ewing, NJ, USA. Email: pearlstl@tcnj.edu
Xin-Wen Wu, Electrical Engineering, The College of New Jersey, Ewing, NJ, USA. Email: pearlstl@tcnj.edu
Adam Lutz, Electrical Engineering, The College of New Jersey, Ewing, NJ, USA. Email: pearlstl@tcnj.edu

PCA is commonly used in the fields of image processing to fuse multimodal and multispectral images [10], signal processing for feature extraction and classification [11, 12], as well as multivariate analysis to reveal the interactions between variables of high-dimensional and cluster analysis [13, 14] to group similarities.

The application of multivariate analysis of PCA can be extended for the purpose of anomaly detection, by applying PCA to datasets containing the normal sensor readings of an IoT network, the interactions between the parameters of the network can be modeled. PCA is applied to both the training and test sets to train models of both the normal interactions of the network as well as the anomalous data. After models are trained for both sets, the Mahalanobis distance can then be calculated for every point in both the training model, representing normal conditions and the test model representing an anomaly within the network. The Mahalanobis distance (MD) [15, 16] is a commonly used distance metric that is used to measure the distance of a point P, that represents each observation of a dataset, from a distribution D, where D is calculated using the covariance matrix and mean distribution of the parameters of the training set [17, 18, 19, 20, 21]. The mean distribution of the Mahalanobis distance is then used to determine a threshold for anomalies, with abnormal observations having MDs that widely differ from the average MD. The MD can then be calculated for each point in the test data, with any observation exceeded the calculated threshold being identified as a possible anomaly [22, 23, 24, 25]. The remainder of this paper is organized as follows. Section 2 shows the technical background and Section 3, describes our anomaly detection algorithm. Section 4 shows preliminary results of the algorithm. Lastly, we conclude the paper with Section 5 which discusses the conclusion of the paper.

II. TECHNICAL BACGROUN

A. Principle Component Analysis

Principal Component Analysis (PCA) is a multivariate analysis technique commonly used for feature extraction and dimensionality reduction of large matrices [26, 27, 28]. PCA reduces the correlated variables in larger matrices into their principal components (PCs), which are linearly independent variables that contain the most significant features of the data. For a matrix $\boldsymbol{X}$ that has n rows, that for our purposes represents every sensor reading observation, with m columns that represent the different parameters of the dataset, PCA is defined as an orthogonal linear transformation that transforms $\boldsymbol{X}$ into an m-dimensional set of vectors of weights w, defined as:

$$w_{(k)} = (w_1, w_2, \ldots, w_m) \tag{1}$$

The vectors of weights contain the PC scores t mapped from each row of $\boldsymbol{X}$ where t is defined as:

$$t_{(i)} = (t_1, t_2, \ldots, t_l) \tag{2}$$

where $t_{k(i)} = x_{(i)} \cdot w_{(k)}$ for $i = 1, \ldots, n \quad k = 1, \ldots, m$. Each of the k^{th} PCs are ordered such that the first PC has the largest variance that captures most of the data. The first PC, $w_{(1)}$ maximizes the variance by satisfying the following:

$$w_{(1)} = \arg\max_{\|w\|=1} \{\|Xw\|^2\} = \arg\max_{\|w\|=1} \{w^T X^T X w\} \tag{3}$$

Each subsequent k^{th} PC can then be found by subtracting the first $k-1$ PCs from $\boldsymbol{X}$ such that:

$$\hat{X}_k = X - \sum_{i=i}^{k-1} X w_i w_i^T \tag{4}$$

and then calculating the weight vector that maximizes the variance:

$$w_{(k)} = \arg\max_{\|w\|=1} \{\|\hat{X}_k w\|^2\} = \arg\max_{\|w\|=1} \{\frac{w^T \hat{X}_k^T \hat{X}_k w}{w^T w}\} \tag{5}$$

Because PCA is an orthogonal linear transform, each pair of PCs is also orthogonal as they are derived from the eigenvectors of the covariance of the data, which are always symmetric [29].

B. Mahalanobis Distance

The Mahalanobis distance is a multivariate distance metric used to determine the distance between points and the distribution. The Mahalanobis distribution can be used to determine outliers in a set of data. The Mahalanobis distance measures the distance a point is from the distribution of all points [30]. To compute the Mahalanobis distribution the formula is:

$$D^2 = (x-m)^T C^{-1} (x-m) \tag{6}$$

where $\boldsymbol{D^2}$ is the square of the Mahalanobis distance, x is the vector of observation, m is the mean values of the independent variables, and $\boldsymbol{C^{-1}}$ is the inverse covariance matrix of the independent variables. When taking the:

$$(x-m)C^{-1} \tag{7}$$

Part of the equation we are taking the distance from the vectors to the mean, then multiplying them by the inverse of the covariance. If the correlation between the points is high, then we will have a smaller distance between the points. Although if the covariance is smaller than there will be a larger distance.

C. Data Preprocessing

Data must first be preprocessed before being used as input for either multivariate analysis or machine learning in order to correctly format it. The data is first loaded into a Pandas DataFrame, converting the timestamp to DateTime format rather than be a separate parameter entire is used as the index for the table. The parameters are then normalized between 0 and 1, ensuring that they're all on the same scale and that a

single parameter doesn't skew the data in any way. The timestamps are then extracted to a separate frame to be cyclically encoded.

In this, each timestamp is separated into hour, day, etc. and converted to be in terms of sine and cosine, essentially divided into a unit circle of appropriate segments. In doing this, the cyclic recurring nature of time can be considered, rather than every timestamp being a unique entry, and allows the model to find correlations between various specified time cycles and the parameters of the network. In this manner, midnight would be the same distance from 12:10 AM as it would be the 11:50 PM, rather than being at the opposite ends where one is at the beginning of the cycle and the other at the end. The encoded data is then appended onto the original DataFrame containing the normalized data to finish preprocessing the data.

D. Auto Encoder

Autoencoders create a compressed representation of their input in a lower dimensional space by encoding it. This allows for both the reduction of high dimensional datasets as well as ensuring that the most significant features of the input data are extracted while filtering out extraneous noise from the data. Autoencoders are composed of an encoder and a decoder with a hidden layer between them as to ensure that the autoencoder does not simply copy its input to its output. The encoder is responsible for reducing the input data to a compressed representation within a latent space. The decoder then attempts to reconstruct its input by decoding the encoded representation and is essentially the inverse of encoder. Since autoencoders are trained to recreate their input, each are unique to the data on which they were trained, meaning that attempting to use an observation from a completely different dataset would cause a high amount of reconstruction loss. The basic procedure for the training of the autoencoder model is shown in Algorithm 1.

Algorithm 1: Auto Encoder Training

```
Algorithm 1
procedure TRAINAUTOENCODER(dataset, hiddenSize)
    [trainData, testData] ← PREPROCESSDATA(dataset, '0.6')
    while (i <maxEpochs) do
        autoenc ← TRAINENCODER(trainData,batchSize,maxEpochs, valiSplit = '0.05')
        encodedData ← ENCODER(trainData)
        decodedData ← DECODER(encodedData)
        reconstructionLoss ← RMSE(decodedData, trainData)
    lossThreshold ← 3 * STD(reconstructionLoss)
    anomalyDetection ← PREDICT(autoenc,testData)
```

E. Thresholding

Thresholding in terms of anomaly detection is the process of separating background noise and normal events from statistically significant anomalies that occur. In the case of multivariate analysis, the distribution of the Mahalanobis distance is used to determine a threshold, similarly using an autoencoder the distribution of the reconstruction loss is used. In either case, the more abnormal any single observation's parameters are from the rest of the data, the further away from the mean of the distribution its respective metric will be. Anomalous events that cause the parameters of the data to change will cause the higher reconstruction loss in an autoencoder trained on the normal operating conditions of a network or be further from the mean distribution of the non-anomalous data in the case of multivariate analysis. By measuring the distribution of either the Mahalanobis distance or the reconstruction loss, statistical anomalies can be defined as points in the data that are several standard deviations away from the mean of the distribution, enough to ensure that the detection does not falsely identify normal background data as anomalies.

III. METHODOLOGY

Now we are going to discuss our PCA Mahalanobis Anomaly detection algorithm and explain each step of the algorithm. The general methodology used is outlined in Fig. 1. Datasets containing anomalies as well as various open-source IoT datasets were used in this study. The dataset containing sensor data is first preprocessed to use time as the index of the data rather than a distinct reading. The datasets are then split into train and test sets, with the training set representing the normal conditions of the devices and the test set containing possible abnormal behavior. Each set is then preprocessed to normalize the data as well as encoding each of the timestamps in terms of hour of day, day of week, etc. which allows the cyclic nature of time to be considered. Principal Component Analysis is then applied to the normalized datasets, with a varying number of principal components being extracted to determine the optimal amount. The covariance matrix and mean distribution of the training set are then calculated and applied to calculate the Mahalanobis distances for each observation in the train and test sets. The distribution of the Mahalanobis distance of the training data is then used to determine a threshold for anomalies, about three standard deviations away from the mean of the distribution. This threshold value is then applied to each observation in the train and test sets, with any observation with a Mahalanobis distance higher than the calculated threshold being identified as a possible anomaly.

his journal uses double-blind review process, which means that both the reviewer (s) and author (s) identities concealed from the reviewers, and vice versa, throughout the review process. All submitted manuscripts are reviewed by three reviewer one from India and rest two from overseas. There should be proper comments of the reviewers for the purpose of acceptance/ rejection. There should be minimum 01 to 02 week time window for it.

Fig. 1.General Methodology

Algorithm 2: Anomaly Detection

Algorithm 2

1: **procedure** ANOMALYDETECTION(*dataset*)
2: `[trainData, testData]` ← PREPROCESSDATA(*dataset*)
3: `[PCATrain, PCATest]` ← PCA(*trainData,testData, nComponents = '3'*)
4: `[cov, invCov]` ← COVARIANCE(*PCATrain*)
5: *meanDist* ← MEAN(*PCATrain*)
6: *MDTrain* ← MAHALANOBISDIST(*PCATrain, invCov, meanDist*)
7: *MDTest* ← MAHALANOBISDIST(*PCATest, invCov, meanDist*)
8: *Threshold* ← MEAN(*MDTrain*) + (3 ∗ *std*(*MDTrain*))
9: *anomalyDetection* ← PREDICT(*Threshold,MDTest*)

IV. RESULTS

In order to see how well our algorithm preformed we used two datasets which was explained in section 2.4 and ran our algorithm on them. The results of using PCA to reduce the dimensionality of large datasets and Mahalanobis distance as a threshold was compared to a machine learning application of anomaly detection, namely autoencoders. Like using PCA and Mahalanobis distance, autoencoders are used to determine anomalies using the reconstruction loss as the threshold. Comparable to Mahalanobis distance, if an observation has a reconstruction loss higher than the determined threshold it's identified as a possible anomaly. Datasets tested include the smart meter readings from the room in a smart home as well as the smart meter readings of an academic university building. The same train and test sets were used with both techniques, also seeing the effects of encoding time versus not doing so. The train and test sets for the room in a smart home is shown in Fig. 2 & 3, with the training set representing the normal conditions of the room and the test set containing a possible anomalous point, in which a parameter sharply drops.

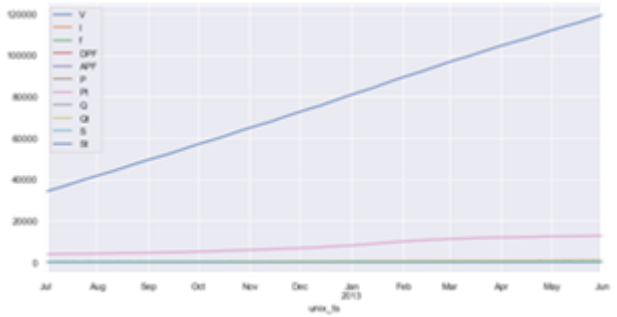

Fig. 2.Train Set

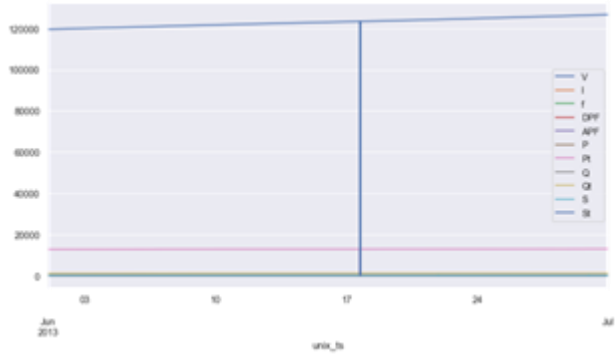

Fig. 3.Test Set

After being preprocessed, each set was tested using both multivariate statistical analysis and machine learning techniques to compare the respective techniques. PCA was applied to the train and test sets and the first two principal components containing the two highest variances were extracted. The Mahalanobis distance was then calculated for each observation in the train and test set using the covariance and mean distribution of the training data, the square of the distribution of the distance shown in Fig. 4 is then used to calculate a threshold value for anomalies.

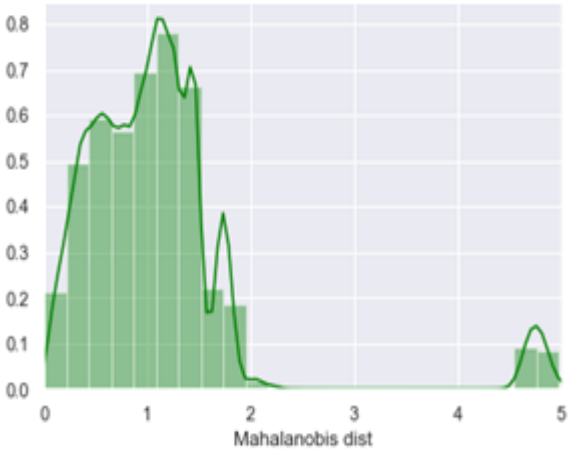

Fig. 4. Mahalanobis Distance

The mean of the distribution was then used to determine a threshold, with any observation having a Mahalanobis distance three standard deviations or more away from the mean being statistically significant and a possible anomaly, in this case the threshold value was then applied to each observation in the train and test data with any observation with a Mahalanobis distance exceeding the threshold being identified as a possible anomaly, shown in Fig. 5.

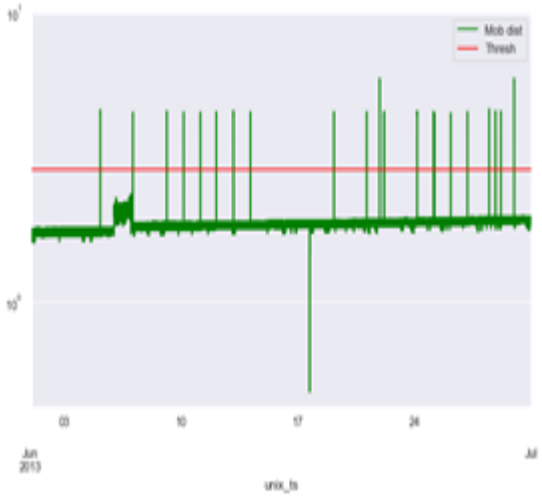

Fig. 5.MD Test Results

In the case of using multivariate analysis, the observation in the test data where the significant drop occurs had the lowest Mahalanobis distance, several standard deviations lower than the mean rather than higher. The timestamps were then encoded for each observation to determine how contextually adding time would affect the Mahalanobis distance. Each observation of the train and test set was augmented to add hour of day, day of week, and day of month in terms of sine and cosine, the test set being shown in Fig. 6.

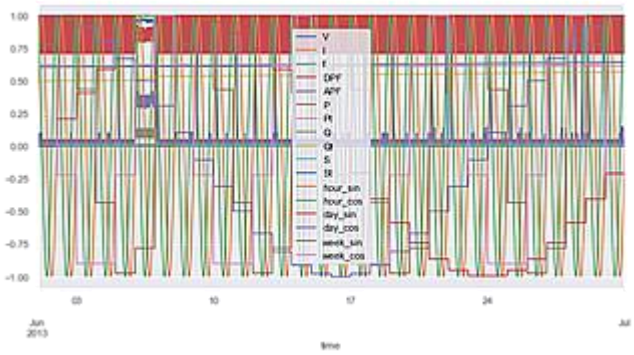

Fig. 6.Test Set

The same technique was then applied to the cyclically encoded dataset, taking the first two principal components and using them to determine the

Mahalanobis distance for each observation, with the square of the distribution shown in Fig. 7.

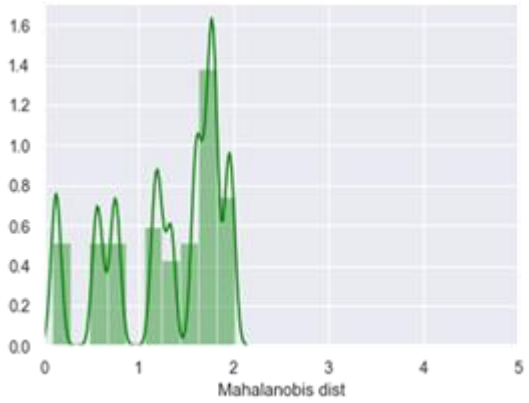

Fig. 7. Square of Distribution

As can be seen, the square of the distribution differs significantly from the original data and no longer follows a general $\chi 2$ distribution. Despite this, cyclically encoding time appears to have an effect, with the oscillating nature being reflected in the Mahalanobis distance. Despite no longer detecting the false positives, the Mahalanobis distance when the anomaly occurs in the test data still has the lowest value, shown in Fig. 8.

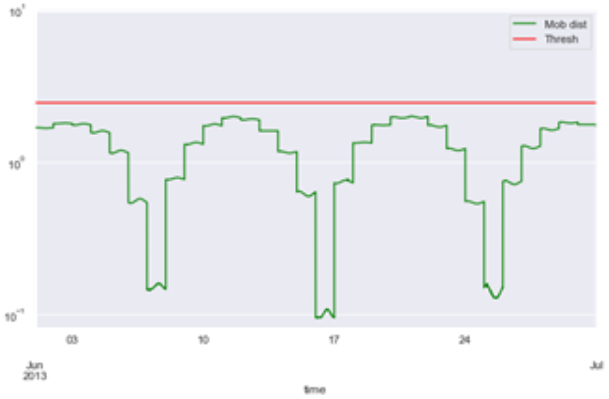

Fig. 8.MD Test Results Cycle

The same dataset, cyclically encoded, was then used to train an autoencoder, again creating a model that represents the normal operating conditions of the smart house. After being fully trained, like using the distribution of the Mahalanobis distance to select an appropriate threshold, the distribution of the reconstruction loss of the autoencoder was used to determine a threshold, shown in Fig. 9.

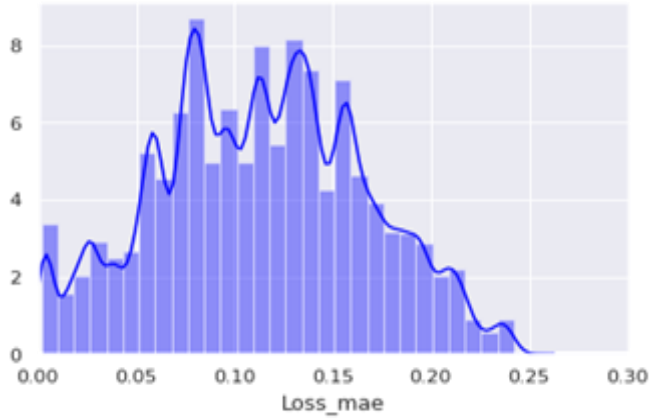

Fig. 9.Distribution of Reconstruction Loss

As with using multivariate analysis, the threshold value was selected to be such that it is statistically significant away from the mean of the distribution, several standard deviations away. The fully trained autoencoder network is then used to reconstruct the values of the test data using the training data as the model, with any observation in the test data that contains abnormalities causing higher reconstruction loss to occur in the prediction, indicating possible anomalies as shown in Fig. 10.

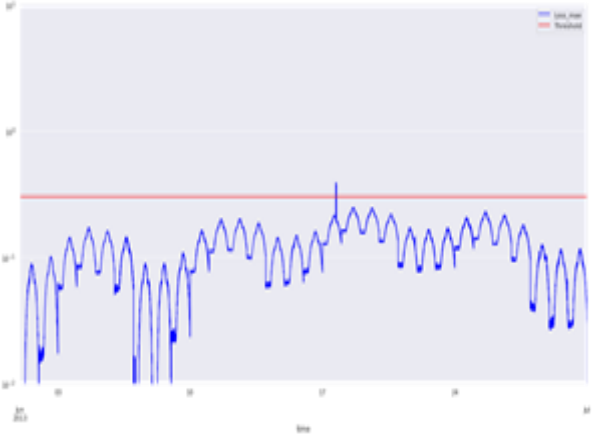

Fig. 10.Autoencoder Test Results

As with using multivariate analysis, encoding time causes reconstruction to show an oscillating pattern, however the results of the autoencoder's prediction did find that the significant parameter drop-off that occurred in the test data did have a reconstruction loss that crossed the threshold, indicating a possible anomaly. Additionally, it can also be seen that the reconstruction loss before and after the anomaly occurs increases, indicating possible signs of anomalous behavior before the event occurs as well as for a period after the occurrence. In order to test for false positives and observe the results of anomaly detection when no confirmed anomalies are present, the academic university building does not contain any confirmed anomalous points. Rather, it has one parameter in a state of constant linear increase, such that the parameter in the test data is higher than any values in the training data, however, is still a steady, consistent linear increase, shown in Fig. 11.

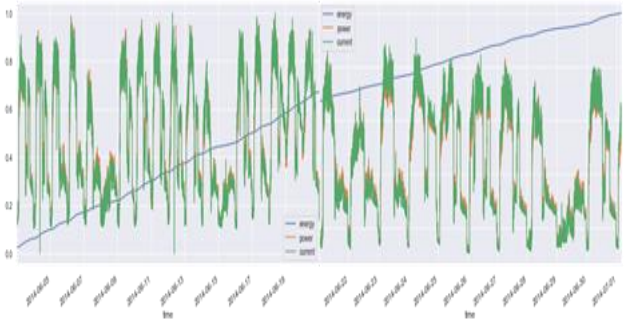

Fig. 11. Linear Increase

The same steps were then performed to calculate the Mahalanobis distances of the train and test sets, shown in Fig. 12 representing the square of the distribution of the Mahalanobis distance of the training data.

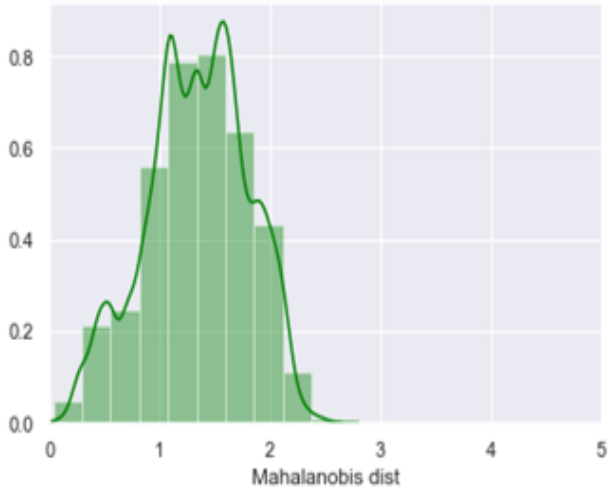

Fig. 12. Square Distribution.

The threshold was again set to three standard deviations away from the mean of the distribution to ensure that any flagged value is statistically significant, with the results of the anomaly detection shown in Fig. 13.

Fig. 13. MD Test Results

Complimentary to the room in the smart home, false positives appear in the anomaly detection. In this case, the Mahalanobis distance appears to be constantly rising due to the parameter in the data that also constantly rises. Due to this, the timestamps of each observation was again encoded in a cyclic manner in order to add time as a context, shown in Fig. 14.

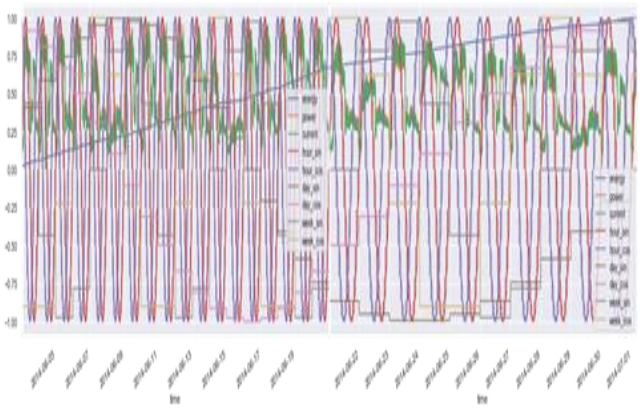

Fig. 14. Encoded Train (Left)/Test (Right) Set

The results of the anomaly detection after cyclically encoding time appears to again have the beneficial results of removing the false positives, with the oscillating nature again being reflected, shown in Fig. 15.

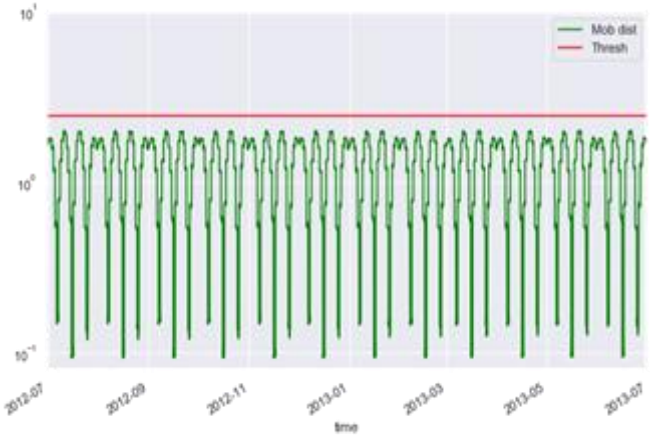

Fig. 15. MD Test Result

As with the smart home dataset, this same train and test sets were then used with an autoencoder. The training set was used to fit the autoencoder model over a set number of epochs, with the distribution of the reconstruction loss shown in Fig. 16.

Fig. 16. Distribution of Reconstruction Loss

The distribution of the reconstruction loss is then again used to determine a threshold for anomalies, with the autoencoder then being used to predict the values of the test data using the trained model, with the reconstruction loss for the train and test sets shown in Fig. 17.

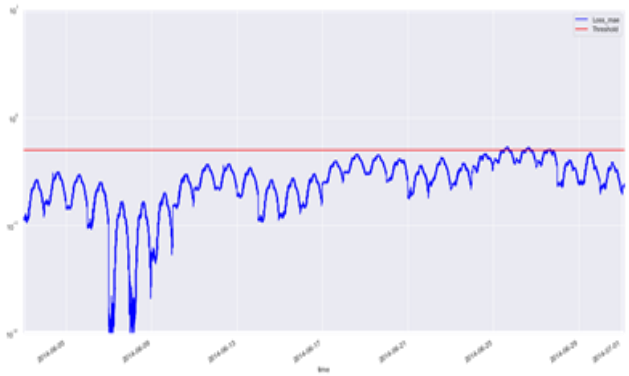

Fig. 17. Reconstruction Loss

In this case, the autoencoder appears to be slightly more sensitive to changes in the parameters over time, as although applying multivariate analysis provides a uniform pattern over the entire dataset, the reconstruction loss of the autoencoder briefly surpasses the threshold before decreasing again. Interestingly, the values of the parameters around this area do appear to differ slightly from previous and subsequent days around the same time, meaning it is possible that a unique or abnormal event occur did occur during this week, however this is unable to be confirmed as the nature of the dataset does not explicitly list the locations of any anomalies.

V. CONCLUSIONS AND FUTURE WORK

In this paper we proposed an anomaly detection algorithm using multivariate analysis. Our preliminary results showed that the algorithm was able to identify anomalies in the dataset, which shows that there is a potential for this to be benchmark techniques for IoT anomalies. However, there is still a significant amount of work that needs to be done for this to perform at a higher level. Using both multivariate analysis and machine learning techniques show promise to be able to detect when abnormal events occur within sensor networks. By fitting a model that represents the normal operating conditions of a network, a ground truth reference can be created while reducing the dimensionality of large-scale datasets. In the case of multivariate analysis, taking the principal components with the highest variance allows for the most significant features of the data to be captured while significantly reducing the dimensionality, whereas training an autoencoder model creates a lower dimensional representation that captures the most significant features of the data. In terms of being able to detect when statistically significant abnormal events occur, our initial results found that autoencoders were slightly more effective, as the metric used was the reconstruction error of the autoencoder rather than the distance away from a mean distribution. When testing using a dataset that had an abnormal event, the autoencoder was able to accurately identify the anomalous observations in the test data, whereas in the case of multivariate analysis, although the anomalous observations were statistically significant away from the mean distribution of the Mahalanobis distance, was several standard deviations lower than the mean. Future work to further fine-tune the anomaly detection will involve training a generative machine learning model on datasets which have been augmented to include controlled anomalies. In doing this, an arbitrary amount of synthetic anomalous datasets can be generated, based off the original data. This would provide a more robust method of generating anomalous datasets that meet the specifications of the original data, that would normally only be able to be gathered by statistically augmenting anomalies into the data or by attempting to collect anomalous behavior from devices directly, which can be infeasible. By controlling the conditions of the anomalies, the exact effect of each parameter on the network would be able to be quantified in a way that would normally be difficult to capture.

ACKNOWLEDGMENTS

This work was funded by Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (829-104-D1435). The authors, therefore, acknowledge with thanks DSR technical and financial support. The authors would like to thank IUP students; Derrick Swint, Noah Knepp, Rohith Gattu, Shane Peterson, John Conlen III, and Tim Valentine for the integration of the MATLAB code, work on the project, and helping in the manuscript editing and formatting.

REFERENCES

1. Zhang, Z. K., et.al., S. IoT security: ongoing challenges and research opportunities. In 2014 IEEE 7th international conference on service-oriented computing and applications (2014) (pp. 230-234).
2. Xu, T., Wendt, J. B., Potkonjak, M. Security of IoT systems: Design challenges and opportunities. In Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design (2014) (pp. 417-423).
3. Mukherjee, A. Physical-layer security in the Internet of Things: Sensing and communication confidentiality under resource constraints. Proceedings of the IEEE, 103(10), (2015) 1747-1761.
4. Barnaghi, P., Wang, W., Henson, C., Taylor, K. Semantics for the Internet of Things: early progress and back to the future. International Journal on Semantic Web and Information Systems (IJSWIS), 8(1), (2012) 1-21.
5. Kolias, C., Kambourakis, G., Stavrou, A., Voas, J. DDoS in the IoT: Mirai and other botnets. Computer, 50(7), (2017) 80-84.
6. Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Kumar, D. (2017). Understanding the Mirai botnet. In 26th USENIX Security Symposium (USENIX Security 17) (pp. 1093-1110).
7. Bertino, E., Islam, N. Botnets and internet of things security. Computer, (2), (2017) 76-79.
8. Jolliffe, I. Principal component analysis (2011) (pp. 1094-1096). Springer Berlin Heidelberg.
9. Wold, S., Esbensen, K., Geladi, P. Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), (1987) 37-52.
10. Zhang, Y. Understanding image fusion. Photogramm. Eng. Remote Sens, 70(6), (2004) 657-661.
11. Subasi, A., Gursoy, M. I. EEG signal classification using PCA, ICA, LDA and support vector machines. Expert systems with applications, 37(12), (2010) 8659-8666.
12. Juan, L., Gwon, L. A comparison of sift, pca-sift and surf. International Journal of Signal Processing, Image Processing and Pattern Recognition, 8(3), (2007) 169-176.
13. Friston, K., et.al.,. Nonlinear PCA: characterizing interactions between modes of brain activity. Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences, 355(1393), (2003) 135-146.
14. Filzmoser, P., Maronna, R., Werner, M. (2008). Outlier identification in high dimensions. Computational Statistics & Data Analysis, 52(3), (2008) 1694-1711.

15. Shi, J., Luo, Z. Nonlinear dimensionality reduction of gene expression data for visualization and clustering analysis of cancer tissue samples. Computers in biology and medicine, 40(8), (2010) 723-732.
16. De Maesschalck, R., Jouan-Rimbaud, D., Massart, D. L., The Mahalanobis distance. Chemometrics and intelligent laboratory systems, 50(1), 1-18.
17. Xiang, S., Nie, F., Zhang, C. Learning a Mahalanobis distance metric for data clustering and classification. Pattern recognition, 41(12), (2008) 3600-3612.
18. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A., Extracting and composing robust features with denoising autoencoders, In Proceedings of the 25th international conference on Machine learning (pp. 1096-1103).
19. Du, Q., Fowler, J. E. Hyperspectral image compression using JPEG2000 and principal component analysis IEEE Geoscience and Remote sensing letters, 4(2), (2007) 201-205.
20. Shyu, M. L., Chen, S. C., Sarinnapakorn, K., Chang, L. A novel anomaly detection scheme based on principal component classifier. Miami University Coral Gables, Florida, Department of Electrical and Computer Engineering.
21. Campbell, N. A. Robust procedures in multivariate analysis I: Robust covariance estimation. Journal of the Royal Statistical Society: Series C (Applied Statistics), 29(3), (1980) 231-237.
22. Liu, C., Wechsler, H. Comparative assessment of independent component analysis (ICA) for face recognition. In International conference on audio and video based biometric person authentication, (1999).
23. Jackson, D. A., Chen, Y. Robust principal component analysis and outlier detection with ecological data. Environmetrics: The official journal of the International Environmetrics Society, 15(2), (2004) 129-139.
24. Worden, K., Manson, G., Fieller, N. R. Damage detection using outlier analysis. Journal of Sound and Vibration, 229(3), (2000) 647-667.
25. Wang, W., Battiti, R. Identifying intrusions in computer networks with principal component analysis. In First International Conference on Availability, Reliability and Security (ARES'06) (2008) (pp. 8-pp).
26. Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., Srivastava, J. A comparative study of anomaly detection schemes in network intrusion detection. In Proceedings of the 2003 SIAM International Conference on Data Mining (pp. 25-36).
27. Kramer, M. A. Nonlinear principal component analysis using autoassociative neural networks. AIChE journal, 37(2), (1991) 233-243.
28. Kambhatla, N., Leen, T. K. Dimension reduction by local principal component analysis. Neural computation, 9(7), (1997) 1493-1516.
29. Lu, Y., Cohen, I., Zhou, X. S., Tian, Q. Feature selection using principal feature analysis. In Proceedings of the 15th ACM international conference on Multimedia (2007) (pp. 301-304).
30. Weng, J., Zhang, Y., Hwang, W. S. Candid covariance-free incremental principal component analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(8), (2003) 1034-1040.

AUTHORS PROFILE

Soundararajan Ezekiel received his M.A and PhD degree from the Department of Mathematics, University of Pittsburgh, Pittsburgh, PA USA. He also received his MSc degree in Mathematics at Loyola college, Post Graduate Diploma in Operations Research at Anna University, and his M. Phil at Madras Christian College, India.

He is currently professor in computer science, Indiana University of Pennsylvania, PA.
Professor Ezekiel is the recipient of three-time SFFP fellow and seven-time VFRP fellow. His research includes Image Processing, Signal Processing, Wavelet Analysis, Artificial Intelligence, Machine Vision, Deep Learning, and Cyber Security.

Abdullah A. Alshehri received his B.S. in 1993 in Electrical Engineering from University of Detroit, Detroit, MI. USA. He received his M.S. and Ph.D. in Electrical Engineering from University of Pittsburgh, PA in 1999 and 2004 respectively. From 2005 to 2010 he worked as assistant professor in the College of Telecom and Electronics CTE and Jeddah College of Technology JCT, Saudi Arabia. In December 2010, he joined the Electrical Engineering Department at King Abdulaziz University-Rabigh KAU, Saudi Arabia and is now an Associate Professor. His areas of interests are in advanced signal and image processing such as time-frequency, wavelet transform, neural networks, and statistical signal processing. Dr. Alshehri is a member of IEEE since 1992 and a member of the Saudi Engineers Council SEC since 2005.