Analysis of the Detection of Abnormal Behaviors in Smart Homes

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Sciences
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree Bachelor of Science in Computer Science Major

Author

Chiraag Umesh
Spring 2020

Technical Project Team Members

Siyuan Shen

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Signed: ____
Chiraag Umesh

Approved: ____________________________________ Date ______________________
Yuan Tian, Department of Computer Science

Analysis of the Detection of Abnormal Behaviors in Smart Homes

Motivation:

The Internet of Things refers to objects that are able to communicate with each other through the Internet. By connecting these objects with programmed systems, it is possible to have them accumulate data, analyze it and draw conclusions/possible courses of actions. As more and more of these devices become connected, their uses have expanded in versatility. These systems have already had applications developed for wearables that track activity with health sensors, cars that can enhance their own performance and even cities that are capable of simplifying their own traffic/noise pollution issues.

One specific area of focus this technology has expanded into is smart home security systems. These systems are set up through a network of sensors that measure a number of different metrics to ensure the overall safety of the residence. The sensors include temperature sensors, motion detectors, contact sensors and many other devices. By keeping a constant eye on the continuously updated sensors' data, the system is able to gain a sense of a “normal” environment for the residence it has been placed within. This is accomplished through the use of machine learning algorithms that recognize the consistent, level readings from this particular environment which means everything is safe within the house. Whenever there is a disruption in the data, meaning there is something wrong in the house environment, the algorithm will produce an alert and will be able to recognize the “normal” home environment has been disrupted. The sensor can then provide an alert to the main control of the system which will then provide further notification to the homeowners. These homeowners can then make decisions and respond to the threats indicated after determining whether they are imminent dangers or not. These possible disruptions can include bathroom falls, kitchen fires, home burglaries and other dangerous events.

This is a very helpful system as it can save many families from dangers, they may not be aware of at the moment. However, like most devices powered by the Internet, the security system can be hijacked and/or manipulated to a malicious user's liking. In the case of this system, falsified data can be fed into the system, while monitoring for disruptions, to deceive the system and give the impression that there is no threat at the moment when there actually is one. To mitigate the chances of this event taking place, this research was started to learn more about how well the system can detect attacks through the data it is given.

Methodology and Approach:

Before embarking on the research project, I educated myself about the topic and read a paper co-authored by my research advisor, Professor Yuan Tian, *Detecting Abnormal Behaviors in Smart Home* [1]. This paper gave my peer, Siyuan Shen, and I a solid idea of what kind of system we were going to work with, the types of sensors that were being used and how accurately the models were performing. In Professor Tian's paper, the researchers tracked many metrics such as time stamps, temperature, motion, average temperature, door status and acceleration. Additionally, there were two approaches to this problem that were constructed. The "One-class SVM" approach is a model that learns by analyzing data that contains one class. As stated in the prospectus, it uses these derived patterns from the data to recognize anomalies and is commonly used for classification purposes. The other approach is the "Autoencoder" approach which is a model that encodes the data it is fed into a representation and then decodes it to a representation that is close to the original data it was fed. Again, as stated in the prospectus, based on the results from this encoding process and the differing statistical “noises” generated, the model is able to detect anomalies.

Siyuan had already conducted tests on the One-class SVM and Autoencoder approaches, so we, first, decided to use just a long short-term memory (LSTM) network with a model. After testing of the network, we decided to add another model to interpret the data and provide results. This was accomplished using the Autoencoder approach accompanied by a LSTM for the machine learning portion of the research. We ran sets of normal and abnormal data on these two models and to determine their accuracy, abnormal accuracy and false positive rates. Their results are shown below in the Results and Discussion section.

We had another test for running regressions to determine if the model could determine whether it is being given information from a malicious device. To give examples of malicious attack data to the sensors, Siyuan and I turned to the previously provided datasets and the first sensor we decided to work with was the temperature sensor. Using time series techniques to analyze change in the sensors’ data over time, we measured the results by setting a threshold at 60 frames to interpret the lag resulting from the highest correlation between the two time series for each step from the sensor data column and the simulated data column. If the offset was greater than the threshold, the time series can be considered to be that of a malicious device.

For the simulated data to be fed to both models, I took the normal data and modified it by either adding/subtracting by 0, 1, 2 and 3 based on the number. The changes were assigned to each number by random assignment to reduce selection bias within the dataset; one instance of this modification is whenever the actual temperature sensor gave a reading of 66, that reading was changed to 68 in the simulated data. To push the LSTM model with the Autoencoder further, Siyuan and I agreed to mix in portions of the normal sensor readings and simulated temperature readings into one column. This was accomplished by alternating between the two columns and taking 300 entries from each column, beginning with the normal sensor readings. For further subtleties in the data, I generated 2 more combined simulated data sets that were more granular in the randomly assigned modifications to each specific number from the original normal temperature sensor data. The second dataset was modified by adding/subtracting 0, 0.5, 1 and 1.5, while the third was modified by adding/subtracting 0, 0.25, 0.5 and 0.75. For example, in the second dataset that added/subtracted 0, 0.5, 1 and 1.5, whenever the actual temperature sensor gave a reading of 66, that reading was changed to 67 in the simulated data. The third dataset that added/subtracted 0, 0.25, 0.5 and 0.75, changed the actual temperature sensor's reading of 66 to 66.5 in the simulated data.

Results and Discussion:

Model	**Overall Accuracy**	**Abnormality Accuracy**	**False-positive Rate**
One-class SVM	83.4%	98.6%	31.8%
Autoencoder	83.7%	72.3%	5.1%
LSTM	91.0%	83.1%	5.0%
LSTM w/ Autoencoder	96.7%	100%	5.0%

Figure 1: Comparison of Model Performance

Displayed above in Figure 1 are the results of all the models on recognizing normal data and abnormal data. The LSTM and LSTM with Autoencoder performed much better than the One-class SVM and Autoencoder approaches by having higher overall accuracies and lower false-positive rates. The One-class SVM and LSTM with Autoencoder had the higher abnormality accuracy rates of the 4 models. Therefore, the LSTM with Autoencoder can be concluded to be the most accurate model for recognizing abnormal data while minimizing chances of error [2].

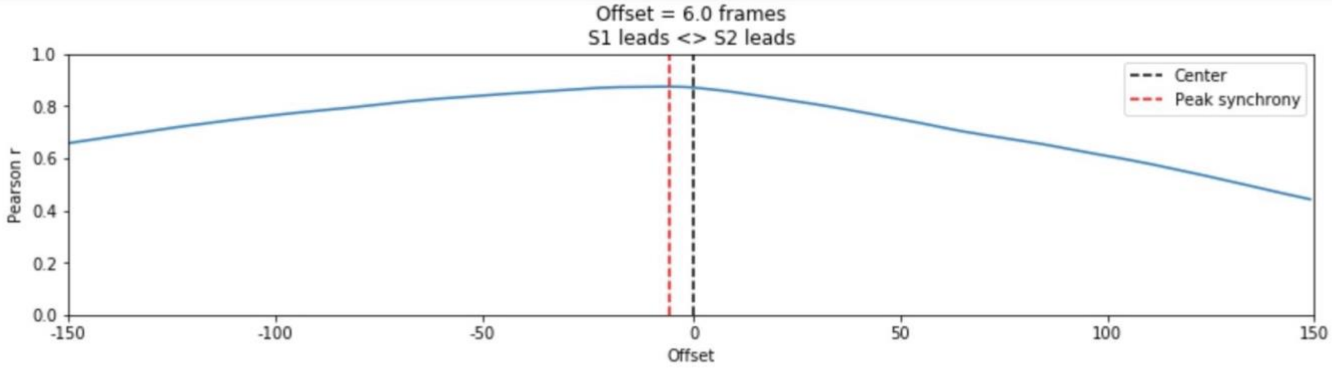

Figure 2: Comparison between Temperature Sensor A and Temperature Sensor B

In the image above, the results of the comparison between two sets of temperature sensor data is shown. It is clear the regression model is able to recognize the two time series in Figure 2 are readings from real data sensors. The highest correlation between the two sensors was 0.8 at an offset of 6.0 frames [2]. Therefore, the model does not predict any threat due to the offset being quite small and a strong correlation between the two time series across different lags. This strong correlation is due to the sensor being in the smart home system and having access to other sensor data readings to ensure its readings are in line. In the images below, the results of the comparison of the second combined simulated dataset with the actual two temperature sensor data sets are displayed.

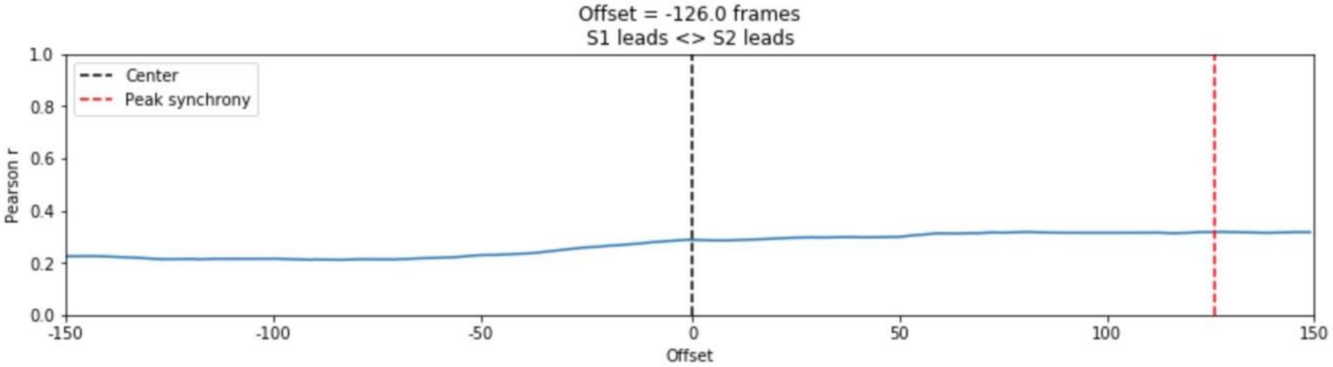

Figure 3: Comparison between Temperature Sensor A and Second Combined Simulated Dataset

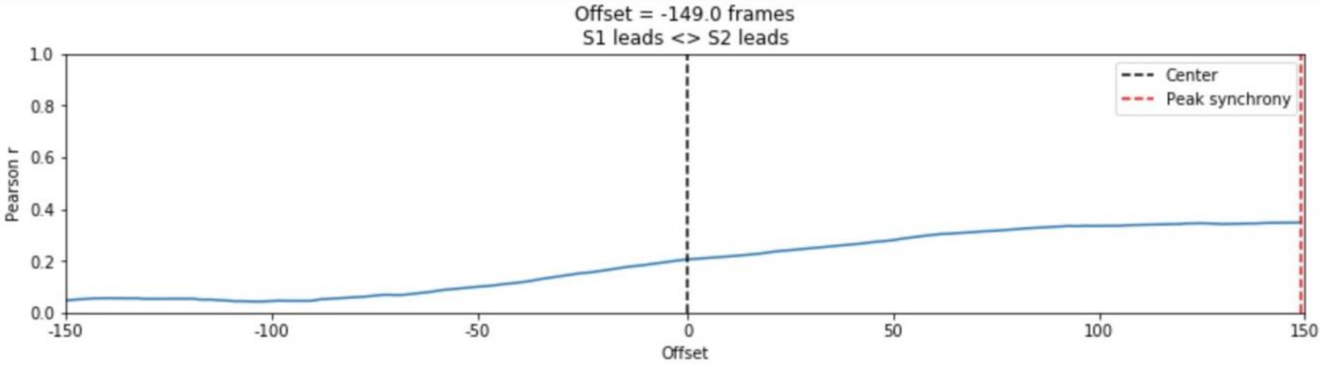

Figure 4: Comparison between Temperature Sensor B and Second Combined Simulated Dataset

As seen in the graphs, the combined simulated datasets are easily recognized to have come from a malicious device. The offsets are much larger here, 126.0 frames in Figure 3 and 149.0 frames in Figure 4, compared to the offset in Figure 2, 6.0 frames. This is due to the malicious device not being in the smart home system and not having its readings in-sync with the readings of the other sensors. This disconnect also leads to the weak correlation across all the lags, concluding that the simulated data must have been from a source that is not the sensor. In turn, proving this regression model is capable of identifying when it is given malicious data mixed with normal sensor data.

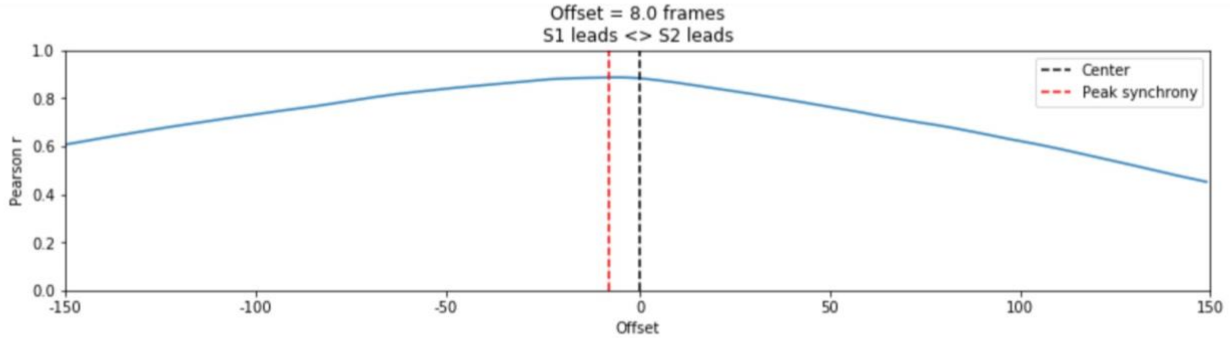

Figure 5: Comparison between Temperature Sensor A and Second Non-Combined Simulated Dataset

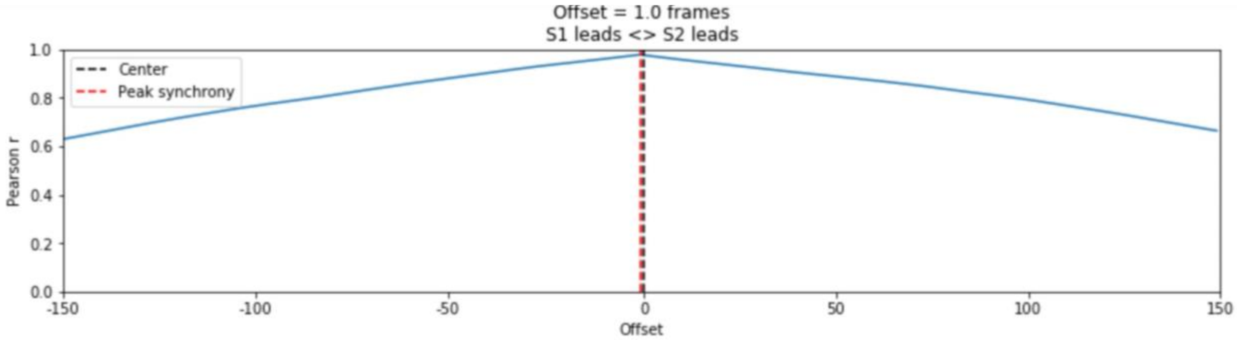

Figure 6: Comparison between Temperature Sensor B and Second Non-Combined Simulated Dataset

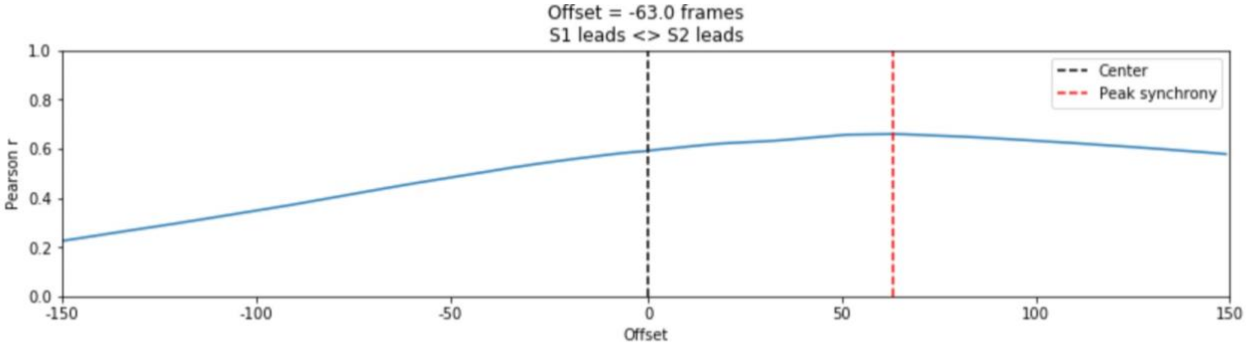

Figure 7: Comparison between Temperature Sensor C and Second Non-Combined Simulated Dataset

However, a different result from the regression model is shown in Figure 5, 6 and 7. In these exhibits, the results of the regression model are shown for multiple temperature sensors and the second simulated dataset that did not have a mixture of the sensor data and simulated data. Although not very visible in Figure 7's graph, the simulated data greatly confuses the regression model in Figure 5 and 6. The simulated data tricks the model into believing the data is coming from a non-malicious sensor as shown by Figure 5 and 6's graphs. In those graphs, the offsets are very small with Figure 5's being 8.0 frames and Figure 6's being 1.0 frames. Additionally, in Figure 7, the data is detected to be from a malicious sensor, but just barely, by 63.0 frames.

This brings about an interesting point as the regression model was unable to detect the contamination of data by a malicious model when the data had been shifted by 0, 0.5, 1 and 1.5 based on the normal temperature sensor's data number, but was able to detect data from a malicious sensor when the same simulated data was mixed in with normal sensor data.

Further Research and Related Works:

There are many variations of this research that could be improved for the future to receive more detailed and accurate results that, in turn, can be likened to more realistic applications. The first change that could be made is the threshold limit. Here, we set the threshold limit to be 60 frames purely based on previous observation. But to receive more accurate results in the future from the regression model, the limit could be lessened to 20-30 frames to provide more accurate results.

A second change that could be instituted is in the intervals of modifications in the combined simulated datasets. Data disruptions in real-life scenarios would probably only occur for a moment in time. Therefore, they would most likely be realistically reflected in the dataset to occur for only a couple of seconds and not repeat every 300 entries in the data. This would provide more realistic and accurate results from the regression model.

Another change to the research that can be applied to the overall system would be the use of a cloud device that promotes adaptability. It would be interesting to see how well a sensor adapts to a new environment and establishes a new "normal" dataset after being accustomed to its "normal" dataset stored in the cloud device from its previous environment. This would advance the system's adaptability aspect along with its accuracy.

The last change that could be made for the future is to further granulate the data examined by the models by removing outliers. This was attempted in our research, but there were issues with the resulting data's format making it unusable. This could be applied to the simulated data and the abnormal data to make it harder for the model to recognize if data has been manipulated.

The work done here can be compared to many other works, such as findings in the paper, *Machine Learning Approaches to Network Anomaly Detection* [3]. In this paper, researchers at McGill University employ the use of a block-based One-Class Neighbour Machine and a Kernel-based Online Anomaly Detection algorithm to identify traffic jams and congestion. One takeaway from their preliminary work is they need to train their model to be adaptive in different environments and over time. Another takeaway is their algorithms must be able to run in real-time, even though it may be receiving noisy data that is hard to interpret. It seems almost the same work is conducted in *Long Short Term Memory Networks for Anomaly Detection in Time*

Series [4]. Researchers trained the network on non-anomalous data and are used as a predictor; the results were modeled into a Gaussian distribution. The findings from this could be compared to our findings and potential ideas, concepts can be abstracted for future work. Additionally, *Time Series Anomaly Detection; Detection of anomalous drops with limited features and sparse examples in noisy highly periodic data* [5], seems to have carried out similar research as well. However, in their work, they were looking for long, sustained periods of disruption and not small delays here and there. Their findings would also be very applicable to our research as they found viable results by combining two anomaly detection methods, something that could be looked into going forward. The paper, *Time Series Anomaly Detection for Trustworthy Services in Butt Computing Systems* [6], presents a new method of detection, Support Vector Data Detection. This could be a compelling method to research as the method was used on a butt computing system, similar to our smart home security networks that also utilize butt computing. Although the method is quite reliable, it still has some shortcomings so those points could be looked into further more. The paper, *Modeling multiple time series for anomaly detection* [7], states their generated algorithms used on real-life NASA shuttle program data are very effective. It'd be interesting to examine how those algorithms would work alongside our models, as their algorithms must be very accurate if they are able to beat out the existing algorithms. The effectiveness of these algorithms and our work can also be compared with the ability of the algorithms used in the paper, *Multi-scale anomaly detection algorithm based on infrequent pattern of time series* [8], which tests easily trainable multi-scalable algorithms on infrequent wave patterns.

Although, once this technology is perfected and adapted for many situations, an interesting work to examine would be *Generic and Scalable Framework for Automated Time-*

series Anomaly Detection [9]. The study provides a framework for scaling up projects that generates a 50-60% improvement in the system's precision. This is particularly interesting and relevant as the study uses real and synthetic data with different characteristics of the time-series to produce results. It would be fascinating to see how that could be applied to scaling up our work.

Moving on from the general subject of anomaly detection, the work's applications can also be diversified from just being used for smart home security systems. The paper, *ECG Anomaly Detection via Time Series Analysis* [10], details the use of the technology in monitoring vital signs of patients in hospitals. This can be explored as a potential future area of research as it uses the same technique as a smart home security system, a set of sensors sending data to a central computer that makes decisions based on the conclusions extrapolated. The paper, *Anomaly Detection of Network Traffic Based on Wavelet Packet* [11], explores the application of anomaly detection in internet network traffic to improve effective use of the network while, the same is being analyzed for industrial control systems in, *Communication Pattern Monitoring: Improving the Utility of Anomaly Detection for Industrial Control Systems* [12]. These papers present interesting areas where our work can be compared with how it would adapt to different applications.

Given the knowledge presented above, some viable next steps would be to see, first, make up for the shortcomings of the research that are easily fixable. This includes changes to the threshold limit, more "realistic" disruptions in data and the removal of outliers in the data. After that is addressed, new algorithms from the papers above could be compared to ours in order to tweak ours to be more accurate. With the addition of other components such as multiple time-series and variable data patterns, our research would advance very far. Once this produces an accurate algorithm, it can be safely scaled up for applications in different fields. These applications can include hospital settings, internet networks and industrial control systems.

References

[1] Yu, Y., Li, C., Jones, M. A., Ma, C., Shezan, F. H., Gao, P., & Tian, Y. (2019). Detecting Abnormal Behaviors in Smart Home, Unpublished Manuscript, p6.

[2] Ericssy. (2019). ericssy/smart-home-security. Retrieved April 7, 2020, from https://github.com/ericssy/smart-home-security

[3] Ahmed, T., Oreshkin, B., & Coates, M. (2007). Machine Learning Approaches to Network Anomaly Detection. *Department of Electrical and Computer Engineering McGill University*.

[4] Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015). Long Short Term Memory Networks for Anomaly Detection in Time Series. *ESANN 2015*.

[5] Shipmon, D. T., Gurevitch, J. M., Piselli, P. M., & Edwards, S. T. (2017). Time Series Anomaly Detection; Detection of anomalous drops with limited features and sparse examples in noisy highly periodic data. *Cornell University*.

[6] Huang, C., Min, G., Wu, Y., Ying, Y., Pei, K., & Xiang, Z. (2017). Time Series Anomaly Detection for Trustworthy Services in Cloud Computing Systems. *IEEE Transactions on Big Data*, 1–1. doi: 10.1109/tbdata.2017.2711039

[7] Chan, P., & Mahoney, M. (2006). Modeling Multiple Time Series for Anomaly Detection. *Fifth IEEE International Conference on Data Mining (ICDM05)*. doi: 10.1109/icdm.2005.101

[8] Chen, X.-Y., & Zhan, Y.-Y. (2008). Multi-scale anomaly detection algorithm based on infrequent pattern of time series. *Journal of Computational and Applied Mathematics*, *214*(1), 227–237. doi: 10.1016/j.cam.2007.02.027

[9] Laptev, N., Amizadeh, S., & Flint, I. (2015). Generic and Scalable Framework for Automated Time-series Anomaly Detection. *Proceedings of the 21th ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining - KDD 15. doi: 10.1145/2783258.2788611

[10] Chuah, M. C., & Fu, F. (2007). ECG Anomaly Detection via Time Series Analysis. *Frontiers of High Performance Computing and Networking ISPA 2007 Workshops Lecture Notes in Computer Science*, 123–135. doi: 10.1007/978-3-540-74767-3_14

[11] Gao, J., Hu, G., Yao, X., & Chang, R. C. (2006). Anomaly Detection of Network Traffic Based on Wavelet Packet. *2006 Asia-Pacific Conference on Communications*. doi: 10.1109/apcc.2006.255840

[12] Yoon, M.-K., & Ciocarlie, G. (2014). Communication Pattern Monitoring: Improving the Utility of Anomaly Detection for Industrial Control Systems. *Proceedings 2014 Workshop on Security of Emerging Networking Technologies*. doi: 10.14722/sent.2014.23012