Journal of Computer Science

Original Research Paper

Anomaly Detection Algorithms for Streaming Data: Performance Comparison

Zirije Hasani

Faculty of Computer Science, University "Ukshin Hoti", Prizren, Kosovo

Article history
Received: 19-04-2020
Revised: 03-07-2020
Accepted: 17-07-2020

Email: zirije.hasani@uni-prizren.com

Abstract: Today's most of the data are streaming time-series data, where is very important anomaly detection over this data because gives significant information of possible critical situations. Detecting anomalies in big streaming data is yet difficult task because we have to process them in real time, even before they are stored and instantly alarm on potential threats. For real time streaming data is important the algorithm used for anomaly detection to be robust with low processing time, eventually at the cost of the accuracy. The aim of this paper is to measure the performance of such algorithms and them to compare with our previously proposed algorithm HW-GA with other existing methods as ARIMA, Moving Average and Holt Winters. The algorithms are implemented in R system and tested on the three Numenta datasets, with known anomalies and own e-dnevnik dataset with unknown anomalies. Evaluation is done by comparing achieved results (the algorithm execution time and CPU usage). As a result of this research we may say that our algorithm HW-GA outperforms others algorithm that we have compared by showing less CPU usage and execution time. Our continues interest is to monitor the streaming log data that are generating in the national educational network (e-dnevnik) that acquires a massive number of online queries and to detect anomalies in order to scale up performance, prevent network downs, alarm on possible attacks and similar.

Keywords: Time Series Data, Big Streaming Data, Anomaly Detection, Numenta, E-dnevnik

Introduction

The focus of researches nowadays is on anomaly detection in real time Big Data. The amount of data is bigger and bigger every day, the data are produced by different equipment like sensors, manufactory equipment, web applications, etc.

The speed of anomaly detection algorithm is very important when we have to deal with real time data.

Anomaly detection algorithm proposal for real time big data is not an easy task due to the fact that many researchers tend to show that their solution is better. In the previous research (Hasani *et al*., 2018) we have proposed an algorithm for detecting anomalies in large real-time data. There we have tested the accuracy of the algorithm, comparing it to several other algorithms that we singled out from previous research such as ARIMA (Kasunic *et al*., 2011), Moving Average and Holt Winters (Ekberg *et al*., 2011).

In this work the idea is to test the performance of the proposed algorithm HW-GA and compare it with other algorithms which are used for finding anomalies in large amounts of data. We are going to compare different algorithm such as HW-GA, ARIMA, MovingAverage, Holt Winter, etc.

Anomaly detection in real time Big Data is actual because the amount of data is increasing every day. There are three characteristics of large amount of data: Volume, veracity and variety of data. Hence the need for performance testing in order to meet the speed characteristic of large amounts of data. It is a vast field of research because it involves algorithms from different disciplines. Before selecting the correct algorithm for anomaly detection is important to specify firstly the data that will be analyzed in order to know how we make the algorithm selection.

The comparative methods will use this study in order to draw conclusions regarding comparative performance.

Experiments, statistical analysis and visualization were managed in R, a free software environment for statistical computing and graphics.

To test the algorithm there will be used benchmark and real time data. The NUMENTA benchmark (Lavin and Ahmad, 2015) database will be used and real time data from e-dnevnik application which is electronic education system in North Macedonia.

The paper has the following structure: In the second section is related work; in the third section is shown the benchmark datasets that are sed for experimental work, in the fourth section are shown the algorithms that are used for testing; the fifth section describe the comparison of algorithms performance; six section discuss the results and conclusion from this research.

Related Work

As our continuous research in this area (Hasani, 2017a) we have compared many algorithms as MAD, RunMAD, Boxplot, Twitter ADVec, DBSCAN, Moving Range Technique, Statistical Control Chart Techniques, ARIMA and Moving Average, to find which one is faster. During this study the most important aspect which we have considered in order to find anomaly detection algorithm suitable for future implementation in the online environment was the execution time (complexity), the CPU usage and the satisfactory quality of algorithm (measured through TP- True Positive, FP-False Positive, FN-False Negative, TN-True Negative anomalies found).

As a result of this research are selected the best algorithms ARIMA and Moving Average are compared with our proposed algorithm (Hasani *et al*., 2018) and Holt Winters where we have tested the correctness of our algorithm in our previous research (Hasani *et al*., 2018) and now in this research we are going to test the performance/speed and CPU usage of our algorithm.

The benchmark Numenta Anomaly Benchmark (NAB) (Lavin and Ahmad, 2015) is proposed, this benchmark is used in our research. Numenta Anomaly Benchmark (NAB), this benchmark provides a controlled and repeatable environment of open-source tools to test and measure anomaly detection algorithms on streaming data. The perfect detector would detect all anomalies as soon as possible, trigger no false alarms, work with real-world time-series data across a variety of domains and automatically adapt to changing statistics.

Zhang *et al.* (2019) propose an online and unsupervised anomaly detection algorithm for streaming data using an array of sliding windows and the Probability Density-based Descriptors (PDDs) (based on these windows). The experimental results and performances are presented based on the Numenta anomaly benchmark.

Boldt *et al.* (2020) investigate to what extent sequence-based Markov models can be used for anomaly detection by means of the end-users' control sequences in the video streams, i.e., event sequences such as play, pause, resume and stop. This anomaly detection approach is further investigated over three different temporal resolutions in the data, more specifically: 1 h, 1 day and 3 days. The proposed anomaly detection approach supports anomaly detection in ongoing streaming sessions as it recalculates the probability for a specific session to be anomalous for each new streaming control event that is received.

Falcão *et al.* (2019) they evaluate experimentally a pool of twelve unsupervised anomaly detection algorithms on five attacks datasets. Results allow elaborating on a wide range of arguments, from the behavior of the individual algorithm to the suitability of the datasets to anomaly detection. They identify the families of algorithms that are more effective for intrusion detection and the families that are more robust to the choice of configuration parameters.

Zhu *et al.* (2018) propose a real-time anomaly detection framework with low computational complexity and high efficiency. They propose Histogram of Magnitude Optical Flow (HMOF) which capture the motion of video patches. They show that HMOF is more sensitive to motion magnitude and more efficient to distinguish anomaly information. Experimentally they show that the framework outperforms state-of-the-art methods and can reliably detect anomalies in real-time.

Yuanyan *et al.* (2018) this paper introduces the extreme value theory and proposes a data streams anomaly detection algorithm based on self-set threshold with extreme value theory (ESOD). They say that the proposed algorithm an update the threshold in real time in order to adopt it for real time streams. In their results that say that their algorithm has good usability and high efficiency.

Jankov *et al.* (2017) presents the implementation of a real-time anomaly detection system over data streams. They implement their algorithm in Java and C++ and in this paper, they provide technical details about the data processing pipelines. They detect anomalies in real time streaming data produced by manufactory equipment.

Also, Wang *et al.* (2019) in their work show the anomaly detection with K-Means clustering algorithm in both batch and real time environment. Their method is dedicated to diagnose potential problems for offshore rotating machinery. Their experiments are compared with the conventional signal analysis method.

Starting from this research related to the works done before, we have identified a research gap related to the analysis of speed of anomaly detection algorithms HW-GA.

Benchmark Datasets

The experiments are done by real data and benchmarks. The real data from e-dnevnik and NAB benchmark (Lavin and Ahmad, 2015). The aim of the experiments is to test the performance of HW-GA algorithm and to camper it with other algorithms.

Table 1: Part of benchmark data used for experiments

HotGym		CPU utilization		Nyctaxi		Rtime e-dnevnik	
timestamp	kw_wnwrgy_consumption	timestamp	metric_value	timestamp	value	timestamp	value
7/2/2010 0:00	21.2	4/10/2014 0:04	93.1456	7/1/2014 0:00	10844	6/13/2016 0:00	6413
7/2/2010 1:00	16.4	4/10/2014 0:24	94.5935	7/1/2014 0:30	8127	6/13/2016 0:00	345
7/2/2010 2:00	4.7	4/10/2014 0:44	93.5210	7/1/2014 1:00	6210	6/13/2016 0:00	354

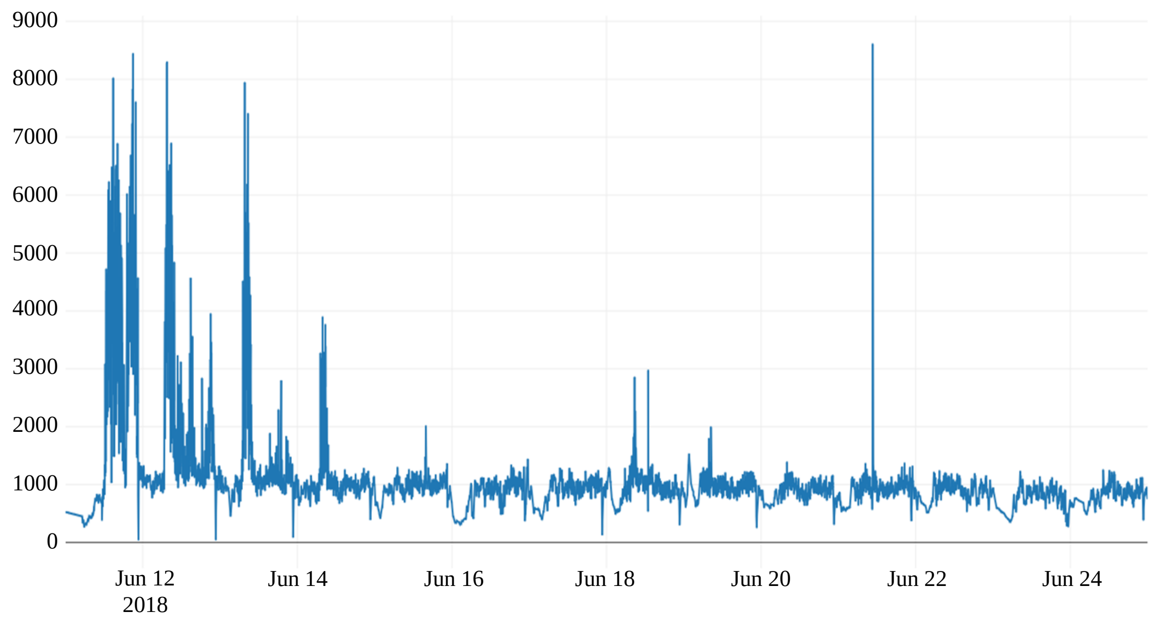

Fig. 1: e-dnevnik two-week data

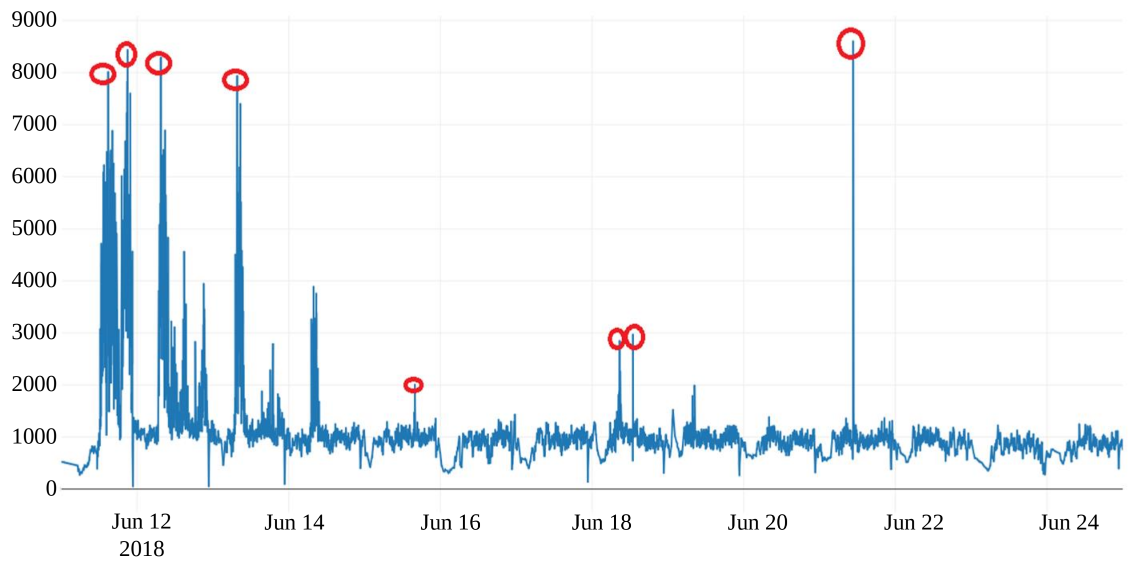

Fig. 2: Anomalies in e-dnevnik data

NAB contains datasets with a real world, labeled data files across multiple domains and the associated anomaly detectors applicable for the streaming data. We use three NAB datasets, HotGym (the energy consumption in one gym center in Australia), CPU utilization and NycTaxi (the number of rides for NYC taxi), as also our e-dnevnik. Parts of the datasets are in the Table 1.

The datasets contain a timestamp and single value based on the log. The first two have real and next two integer values. Known anomalies are detected by human inspection and confirmed by HTM algorithm in all three NAB datasets, while in the e-dnevnik dataset anomalies are not known.

In Fig. 1 is presented a sequence of data analyzed from e-dnevnik. They are data with seasonality, is the working period from 07:00 to 19:00 and not the working period 19: 00-07: 00. If there is an increase in demand during the non-working period, it is calculated as an anomaly. Figure 1 shows the data for the two-week period.

In Fig. 2 are shown with red circles what means one anomaly in our real data and the algorithms tend to pick them in the result.

Algorithms used for Testing

In statistics and econometrics and in particular in time series analysis, an Autoregressive Integrated Moving Average (ARIMA) (Kasunic *et al*., 2011) model is a generalization of an Autoregressive Moving Average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting) Moving average. In time series analysis, the Moving-Average (MA) model is a common approach for modeling univariate time series. Together with the autoregressive (AR) model, the moving-average model is a special case and key component of the more general ARMA and ARIMA models of time series, which have a more complicated stochastic structure.

HW-GA algorithm: The Adaptive Algorithm for Anomaly Detection. In Fig. 3 the positive feedback optimization method for continuous adaptation of the anomaly detection parameters is shown. The method is composed of four different stages.

First is the annotation of the anomalies in the training dataset. The anomaly annotation is defined as a time interval where an anomaly is located. The annotation is done by a human or an oracle.

The second stage is the computation of anomaly detection parameters for our algorithm using GAs, i.e., computation of HW or TDHW parameters, together with δ, *k* and *n*. GAs have been successfully applied to solve optimization problems, both for continuous (whether differentiable or not) and discrete functions. This enables us to find near-optimal values of the anomaly detection parameters very successfully.

The third stage is the actual anomaly detection engine based on the computed optimal parameters from the second stage. This stage outputs the detected anomalies with our proposed algorithm.

The fourth stage is the human acknowledgment of the output data and classify the output data into TP (true positive), False Positive (FP) and False Negative (FN). The result of the verification/acknowledgment stage is then used again in the second stage for further optimization of the anomaly detection parameters.

Performance Comparison of Anomaly Detection Algorithms

In real time analytic is very important the speed because it have to deal with real time data. Our proposed algorithm HW GA (Hasani *et al*., 2018) is tested for correctness but not for performance. In this paper we test the performance of our algorithm and compare it with other selected algorithms from previous research.

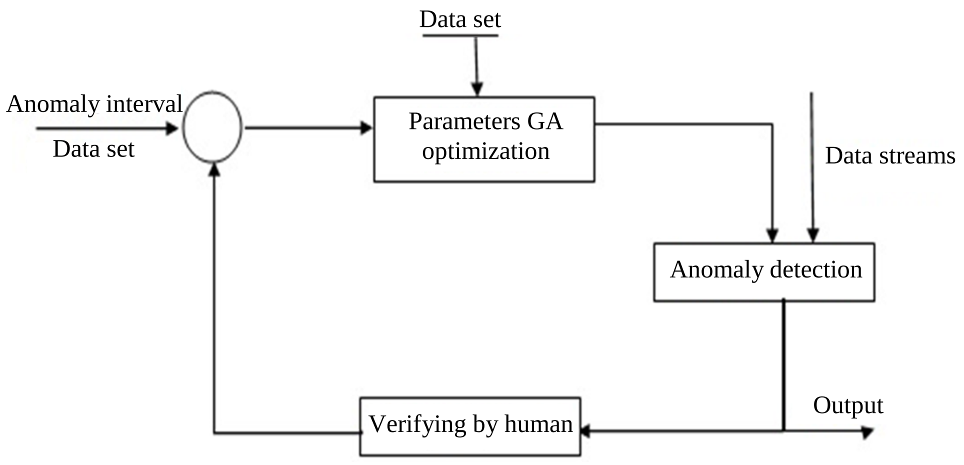

Fig. 3: Model for HW-GA method for anomaly detection

Table 2: Experimental results from CPU usage and execution time

	E-dnevnik>40000		NYcTaxi-1441		HotGYM-169		CPU usage-3653	
Algorithms	Execution time (seconds)	CPU Usage	Execution time (seconds)	CPU Usage	Execution time (seconds)	CPU Usage	Execution time (seconds)	CPU Usage
HW GA	3.36	27%	1.20	9.8%	1.69	9.9%	0.32	12%
HW calc. MASE	12.17	43%	1.28	11.3%	1.47	14.6%	5.66	18%
HW def. MASE(k)	5.77	39%	1.44	10.4	1.21	10.4%	5.04	17.9%
HW def. MASE(k,n)	5.98	45%	1.22	10.6%	1.37	10.5	5.51	18%
ARIMA	3.52	32%	1.21	3%	0.73	2.3%	4.07	17.6%
MA	23.38	65%	1.27	7.6%	0.53	3.2%	14.38	22%

Our proposed algorithm (HW GA) (Hasani *et al*., 2018) with GA optimized parameters (α, β, γ, δ, k, n) and with improved $MASE_{(\alpha,\beta,\gamma,\delta,k,n)}$ is compared with ARIMA, MA (implemented in our previous work (Hasani, 2017a), HW where smoothing parameters are calculated by formula and default MASE (HW calc.MASE), HW by default smoothing parameters (optimized in R) and default MASE (HW def.MASE), HW by default smoothing parameters and improved $MASE_{k,n}$ (HW def.MASE(k,n)).

Algorithms evaluation focus on execution response time and CPU usage. These two parameters are measured in the running time of the algorithms. The algorithms are implemented in R language and we have added in the code in the star of the algorithm the timer and in the end of the algorithm it shows the time taken to finish the algorithm, on the other side the CPU usage is monitored in real time when the algorithm is running how much CPU it use.

These parameters are important for us because, in the future, the algorithm HW-GA have to work in the real-time environment. Other criteria should be robustness, flexibility, scalability and simplicity to implement in our online infrastructure (Hasani *et al*., 2015; Hasani, 2017b).

Table 2 show the execution time and CPU usage for six algorithms which are compared between them HW-GA with others. The experiments are done in real data and benchmark data as a described above. From the result we can see that in real data e-dnevnik data the execution time is faster in HW-GA 3.36 sec compared to other which is larger also the CPU usage is smaller in real data. The Execution time depends also from the amount of data to be tested this affects also the CPU usage but when it is compared to HW-GA it shows better results.

In real data the last algorithm MA show the larger CPU usage 65% but is not the same situation which benchmark data, but this happen because the amount of data in e-dnevnik real data is larger than 40000 where in others is much smaller (NYcTaxi-1441 records, HotGYM-169 records, CPU usage-3653 records).

Results and Discussion

The results from experiments are shown in Table 2 from where we can see that CPU usage depends from the number of records in one dataset. In general, two smaller datasets of NYcTaxi and HotGYM the execution time is smaller compared to CPU usage which have more than three thousand records. This is for benchmark dataset but the real dataset from e-dnevnik have larger amount of data more than 40 thousand and here we can see that CPU usage is larger compared to others. The other thing that we can see is that ARIMA and MA in general in all datasets have smaller CPU usage this because they are not complicated algorithms compared to HW-GA but related to correctness they are not good.

If we compare HW-GA with Holt Winters and their modifications we made the CPU usage is smaller in all datasets.

The result we get from experiments for execution time we can say that when the dataset is larger our algorithm HW-GA show better results compared to smaller datasets. For example, the execution time in e-dnevnik dataset for HW-GA is 3.36 sec all others have more than 5 sec execution time. The worse result show MA with 23.38 sec execution time on e-dnevnik dataset.

On two smaller datasets NYcTaxi and HotGYM our algorithm good execution time 1.2 and 1.69 sec on these two datasets. Also, the others algorithm is these two small benchmark datasets outperform well.

On the other side the third benchmark dataset CPU-Usage which have more than 3 thousand records our algorithm has performed much better than others with execution time 0.32 sec. When it is compared with others the difference is very large more than 5 sec. The worst case is with MA with 14.38 sec execution time.

Based on these facts that we describe here we can say that our algorithm outperforms others algorithm in both measurements parameters (execution time and CPU usage) and better results are shown with large amount of data which is very important for Big Data.

Conclusion

The researchers are focused now mainly in anomaly detection in real time Big Data because the amount of data is growing every day.

This paper covers this problem and our previously proposed algorithm HW-GA for real time anomaly detection is compared with other existing methods by measuring the performance. As a conclusion from our experiment, we may say that HW-GA is efficient concerning execution time and CPU usage. Our algorithm has smaller CPU usage and less execution time compared to other algorithm. The results are shown in Table 2.

As a result of this research we can say that our algorithm outperforms others algorithm in both measurements parameters (execution time and CPU usage) and better results are shown with large amount of data which is very important for Big Data.

In or Continuous Work

In our continuous work, we are building an infrastructure for anomaly detection in the big log files in real-time that contains computational, storage, scalability and real-time challenge. To make proper choice of infrastructure we have done extensive investigation reported in (Hasani *et al*., 2015; Hasani, 2017b). We have found infrastructure appropriate, because it is possible to modify it when needed by adding various other components or scale up or down by adding (duplicate, triplicate) some of its existing components. Ongoing experiments are motivated by need to use such an infrastructure for anomaly detection in big log data generated by load balancers of servers in Faculty of Computer Sciences and Engineering (FINKI).

The next phase of our research is to modify the proposed algorithm in order to add more parameters into the optimization procedure with Genetic Algorithms in order to see if it will give better performance. Also, we plane to implement this algorithm in our proposed infrastructure (Hasani, 2017b) and to test it in real time environment.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Boldt, M., Borg, A., Ickin, S., & Gustafsson, J. (2020). Anomaly detection of event sequences using multiple temporal resolutions and Markov chains. Knowledge and Information Systems, 62(2), 669-686.

Ekberg, J., Ylinen, J., & Loula, P. (2011, December). Network behaviour anomaly detection using Holt-Winters algorithm. In 2011 International Conference for Internet Technology and Secured Transactions (pp. 627-631). IEEE.

Falcão, F., Zoppi, T., Silva, C. B. V., Santos, A., Fonseca, B., Ceccarelli, A., & Bondavalli, A. (2019, April). Quantitative comparison of unsupervised anomaly detection algorithms for intrusion detection. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (pp. 318-327).

Hasani, Z. (2017a, June). Robust anomaly detection algorithms for real-time big data: Comparison of algorithms. In 2017 6th Mediterranean Conference on Embedded Computing (MECO) (pp. 1-6). IEEE.

Hasani, Z. (2017b, April). Implementation of infrastructure for streaming outlier detection in big data. In World Conference on Information Systems and Technologies (pp. 503-511). Springer, Cham.

Hasani, Z., Jakimovski, B., Kon-Popovska, M., & Velinov, G. (2015). Real time analytic of SQL queries based on log analytic. ICT Innovations, 78-87.

Hasani, Z., Jakimovski, B., Velinov, G., & Kon-Popovska, M. (2018, November). An Adaptive Anomaly Detection Algorithm for Periodic Data Streams. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 385-397). Springer, Cham.

Jankov, D., Sikdar, S., Mukherjee, R., Teymourian, K., & Jermaine, C. (2017, June). Real-time high performance anomaly detection over data streams: Grand challenge. In Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems (pp. 292-297).

Kasunic, M., McCurley, J., Goldenson, D., & Zubrow, D. (2011). An investigation of techniques for detecting data anomalies in earned value management data. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.

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

Moving Average, https://cran.r-project.org/web/packages/smooth/vignettes/sma.html

Wang, Z., Zhou, Y., & Li, G. (2019, November). Anomaly detection for machinery by using Big Data Real-Time processing and clustering technique. In Proceedings of the 2019 3rd International Conference on Big Data Research (pp. 30-36).

Yuanyan, L., Xuehui, D., & Yi, S. (2018, November). Data streams anomaly detection algorithm based on self-set threshold. In Proceedings of the 4th International Conference on Communication and Information Processing (pp. 18-26).

Zhang, L., Zhao, J., & Li, W. (2019). Online and Unsupervised Anomaly Detection for Streaming Data Using an Array of Sliding Windows and PDDs. IEEE Transactions on Cybernetics.

Zhu, H., Liu, B., Lu, Y., Li, W., & Yu, N. (2018, December). Real-time Anomaly Detection with HMOF Feature. In Proceedings of the 2018 the 2nd International Conference on Video and Image Processing (pp. 49-54).