Journal of Theoretical and Applied Information Technology
15th August 2022. Vol.100. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# IMPLEMENTING MULTI-THREADED AUTONOMOUS ANOMALY DETECTION (MAAD) IN HEALTH TRACKING DEVICES

**MUHAMMAD YUNUS BIN IQBAL BASHEER[1], AZLIZA MOHD ALI[2], NURZEATUL HAMIMAH ABDUL HAMID[3], MUHAMMAD AZIZI MOHD ARIFFIN[4], ROZIANAWATY OSMAN[5], SHARIFALILLAH NORDIN[6]**

Faculty of Computer & Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia

E-mail: [1]muhammadyunus185@gmail.com, [2]azliza@tmsk.uitm.edu.my, [3]nurzeatul@tmsk.uitm.edu.my, [4]mazizi@tmsk.uitm.edu.my, [5]roziana@tmsk.uitm.edu.my, [6]sharifa@tmsk.uitm.edu.my

**ABSTRACT**

Nowadays, people commonly wear health tracking devices or smartwatches as regular trackers for their health. There are many brands available in the market that are offered different functionalities. The device produces real-time health data, which should be monitored autonomously. The data from these devices is the number of calories burned, steps taken, heart rate, sleeping pattern and exercises. Based on daily activities, these devices produce a vast amount of data that could be used for health monitoring purposes. Analyzing data patterns can also provide insights into detecting health anomaly data. This paper presents a multithreaded autonomous anomaly detection algorithm (MAAD) for use in health tracking devices. The data analysis will be done in incoming streaming data, which continuously changes dynamically in the pipeline. Hence, we propose an autonomous approach using MAAD to detect anomalies from incoming smartwatch data. Firstly, we conduct data pre-processing using data gathered from several wearable devices. Once the data were ready, we sent the data via the chosen internet of things (IoT) pipeline. The data is then received by MAAD, which can handle two different processes or threads simultaneously. The first thread runs incoming data, and the other performs anomaly detection. The result shows that MAAD performs better in detecting health tracking data anomalies. This algorithm is also faster than AAD and streaming TEDA when presented with streaming data. In the future, the algorithm can be applied to any brand of smartwatch or IoT device that can supply data continuously.

**Keywords:** *Autonomous, Anomaly, Health Tracking Device, IoT, MAAD.*

## 1. INTRODUCTION

Wearable devices are increasingly affordable. They provide health tracking services with the innovative development of low-power design and edge computing technologies [1]. A wearable device such as a smartwatch is portable. It contains various sensors such as an accelerometer, a global positioning system (GPS), barometers, magnetometers, and gyroscopes that can track daily human activities [2], including steps taken, distance covered, and calories burned. This type of health tracking device, also known as a smartwatch, generates data every second without stopping. This data describes the users' movement and gets users to understand their health status. Furthermore, smartwatches have become versatile and are still developing [3]. Unfortunately, these devices can also produce anomalous data. This can make users lose confidence in their device.

This paper applies multithreaded autonomous anomaly detection (MAAD) to the smartwatch dataset, including smartwatch users' steps, distance, and calories. The research was conducted using the internet of things (IoT) architecture as a pipeline source to supply streaming data. This is because smartwatch data is read every second, and it cannot be stopped, such as streaming data [4]. Furthermore, streaming data changes constantly in the pipeline, and the data produced are thus unpredictable [5]. Meanwhile, as time goes on, streaming data will accumulate over time. As a result, memory consumption will increase as well. Therefore, memory needs to be used efficiently and data should be processed as soon as it becomes available [6]. Ultimately, processing streaming data from a

smartwatch is complex, and the algorithms used need to consider the challenging streaming data environment.

MAAD is an algorithm strongly suited to the use of using streaming data. It is an upgraded version of AAD [7], which works using offline data. MAAD was tested in [8], which demonstrated that MAAD worked well and consumed less time to process streaming data from the DHT22 sensor. Furthermore, MAAD does not require any prior assumptions or thresholds to define anomalous data. This is desirable, as the streaming data pattern is unknown. To overcome huge memory consumption, a recursive update is used, which will reuse the existence value to update the status of the data. In addition, MAAD uses an empirical data analysis (EDA) [9] mechanism that operates based on mutual data distribution. MAAD is also an autonomous algorithm that allows less human expert intervention [10]. Therefore, MAAD is assumed to be the most suitable algorithm for tackling the issues present in smartwatch streaming data. As a result, anomaly data read from the smartwatch can be monitored from time to time.

This paper starts with an introduction, followed by data acquisition in section 2. MAAD implementation in smartwatch data collected from section 2 is described in section 3. Finally, sections 4 and 5 evaluate and conclude the overall research.

## 2. DATA ACQUISITION

Data acquisition was performed by collecting data from various smartwatch users [11]. The smartwatches used include several brands, including Huawei, Xiaomi, and Fitbit. As a result, there are a total of 1219 data points collected. These data contain 12 attributes. Table 1 shows five sample data selections from the data collected in this phase.

*Table 1: Sample of Collected Data.*

| *Date* | *Calories Burn* | *Steps Taken* | *Distance Travelled* |
|---|---|---|---|
| 1/10/2018 | 0.11 | 0.28 | 0.25 |
| 13/8/2018 | 0.01 | 0.03 | 0.03 |
| 18/2/2019 | 0.16 | 0.41 | 0.38 |
| 31/12/2017 | 0.02 | 0.03 | 0.03 |
| 19/10/2018 | 0.15 | 0.24 | 0.24 |

Based on table 1, the date represents when the user conducts the related activity. Calories burned are in kilocalories (kcal). Meanwhile, steps taken, and the distance travelled are in meters (m). Time active is in seconds. Resting heart rate consists of heartbeats rate when resting in one minute. Weight and height form BMI (kg/m2). There are four users with ages between 20 and 23 and a female user with age 40. All the data are reanalyzed so that only important or suitable data are used. Attributes such as date, time active, resting heart rate, BMI, weight, height, gender, name, and age are removed. There are calories burned, steps taken, and distance travelled are the chosen attributes in this research. The three attributes have a high correlation between the other collected nine attributes. It was found that the more the steps are taken, and the greater the distance travelled, the more the calories should be burned.

## 3. MULTITHREADED AUTONOMOUS ANOMALY DETECTION

MAAD is an algorithm that can process asynchronously. It is an algorithm that creates two threads that run simultaneously. Therefore, these two threads are independent. The first thread will handle incoming data which is from the smartwatch device. Meanwhile, the second thread will conduct anomaly detection based on the autonomous anomaly detection algorithm introduced in [7]. Eventually, MAAD incorporated an EDA mechanism that allows analysis based on mutual data distribution. Therefore, there are no thresholds used to assume the distribution of data. MAAD keep handling incoming data without the intervention of human experts. Hence, MAAD is an autonomous algorithm that analyses the smartwatch dataset on its own. MAAD also does not require any labelled data, making it an unsupervised algorithm.

### 3.1 MAAD Architecture

MAAD has a fairly robust architecture which includes AAD running at the background of the MAAD algorithm. MAAD will not wait for AAD to finish its process in the background. However, MAAD will continue grabbing incoming data ignoring the anomaly detection process. Before the data enter AAD in MAAD, the current data read by the smartwatch will be assumed as normal. This technique was also used in [12]. After AAD has finished its process, MAAD will insert current data into AAD to further detect any anomaly. The proposed asynchronous engine will eventually save time and prevent vital data from being missed. Imagine only one thread in the anomaly detection algorithm. Then, the first data are inserted into the anomaly detection algorithm. An anomaly detection algorithm will take time to process the data; during this process, there are essential data read by the smartwatch. But the smartwatch cannot send the data, since the algorithm is processing previous data.

Therefore, MAAD prevents such problems from happening

MAAD also promises to work efficiently. This is because AAD consists of four different stages. The first stage consists of Chebyshev inequality. This stage will evaluate all the data. Then, lower influence data will be declared as potentially anomalous. Meanwhile, other non-potential anomaly data will be declared as normal. The declared possible anomaly data then enter stage two until the final stage. Therefore, the Chebyshev inequality stage is the first stage that will reduce computational burden by the anomaly detection algorithm. Not only that, as time goes on, streaming data will begin to accumulate. Hypothetically, the more the data, the more time it takes for the algorithm to finish its process. This means the anomaly detection algorithm becomes slow from time to time. As a result, the algorithm becomes late to get ready for the following data. This is not the case with MAAD, which will continue grabbing incoming data to be sent to AAD. Figure 1 shows the architecture of the MAAD algorithm.

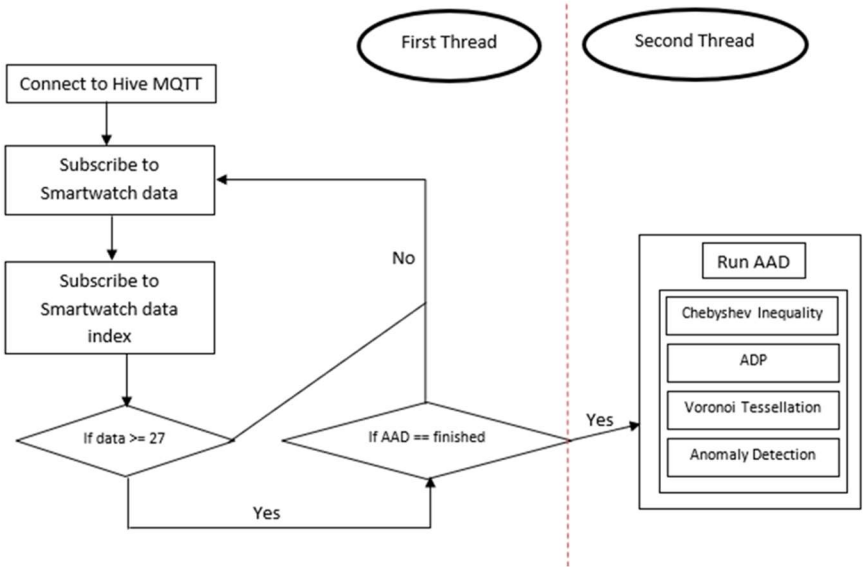


*Figure 1: MAAD Architecture*

Based on figure 1, the first thread of the MAAD algorithm will connect to hive MQTT, which is an internet broker that will act as a pipeline. Then, the first thread will subscribe to smartwatch data and index to allow data capturing from smartwatch devices. Then, if the data are more than or equal to 27, it will check whether AAD had finished its previous process or not. If AAD is not running or has finished its process, MAAD will send data from the first thread to AAD, which then activates second thread. After that, MAAD will continue subscribing next data from the smartwatch, since the algorithm cannot stop, while data keep coming in. If data is less than 27 or AAD is running, MAAD will also subscribe following data from a smartwatch. Note that first and second threads run simultaneously, and the processes are not dependent on each other. There is also pipeline architecture, so that the smartwatch device can read user health status and send data to MAAD systematically. This is shown in figure 2.

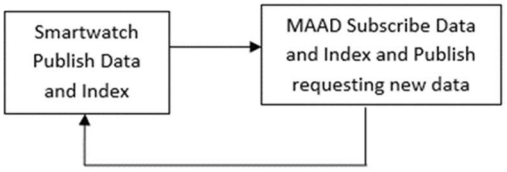


*Figure 2: Pipeline Architecture*

The pipeline architecture in figure 2 shows about what happen inside the proposed system. Firstly, smartwatch will send first data which may contain steps taken, distance and calories burned to hive MQTT. Then, MAAD will subscribe the data from hive MQTT and further process, it as shown in figure 1. After that, MAAD will publish to hive MQTT to request the next data. As a result, the architecture in figure 2 will allow system subscription and publication of data systematically. The algorithm

will also work fine, and data can be analyzed sequentially based on time when they arrive.

### 3.2 User Interface

Based on the proposed MAAD algorithm, the algorithm was build based on IoT mechanism. Nowadays, IoT are widely used, and there is no doubt if IoT can be implemented in smartwatch devices. It is also important to visualize the results. Therefore, the system interface proposed in [8] is used. Figure 3 shows this system interface.

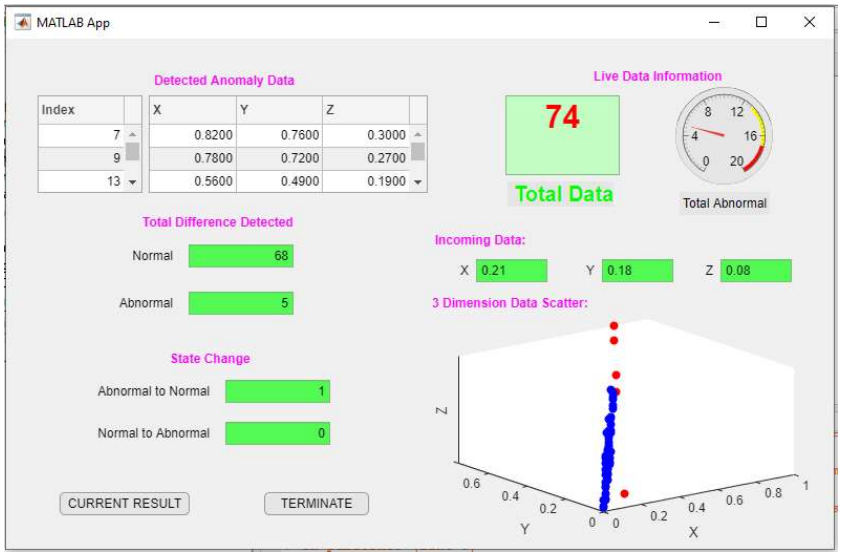


*Figure 3: User Interface*

Based on figure 3, the user interface displays all information about real time events from a smartwatch. There are detected anomaly data as well as total number of data which had entered MAAD algorithm. Incoming data section consists of X, Y and Z which consist of steps, distance and calories burned respectively. Then, the live data are plotted in a 3D scatter graph. There is also a total difference detected section, which will display a total number of anomalies and normal data detected throughout time. Then there is the state change section, which will tell about change of data from abnormal to normal and normal to abnormal throughout time. This is because data status will continuously change along the time, and this is why streaming data is said to exist dynamically. The current result button will visualize current result in table. Finally, the terminate button will terminate overall system and visualize table for the recent results.

### 3.3 Discussion

Overall, MAAD is an enhancement of AAD which works offline. Furthermore, AAD was changed to work asynchronously, forming MAAD. MAAD consists of two threads that can communicate with IoT device. The smartwatch will send data to hive MQTT which then captured by MAAD algorithm throughout the time. As time goes on, AAD will take time to process anomaly data. This is because data will continue accumulating over time.

Meanwhile, MAAD cannot wait for AAD to finish its process because there are data ready to be received from the smartwatch device. Hence, MAAD consists of two threads which run simultaneously. Each thread in MAAD is independence and MAAD will not wait for AAD to finish its process. To visualize real time data, the system interface from [8] is used. The system interface enables interaction from end users, assisting them in making decisions.

## 4. EVALUATION

There are two evaluation tests conducted which are performance test which involve whether it can detect anomaly data in smartwatch dataset and time test which compare three different algorithms in terms of speed when injected with streaming data. Besides, time test is very important because it will test which algorithm can react rapidly and continue to receive next data since streaming data cannot be pause and will always enter the data cloud [4]. Meanwhile, performance test involves whether it can detect majority of data which are far from normal pattern. Hypothetically, when data are similar or the

same from time to time, this is normal. However, if the data formed have a rare location, then they need to be analyzed, since there is a probability that the data are anomalous.

### 4.1 Performance Test

In this test, data were analyzed using three-dimensional (3D) scatter plot. As mentioned before, there are three attributes considered. The attributes include steps and distance in meters (m) and calories in kilocalories (kcal). When injected with 1219 data points, the MAAD algorithm creates a 3D scatter plot, as shown in figure 4.

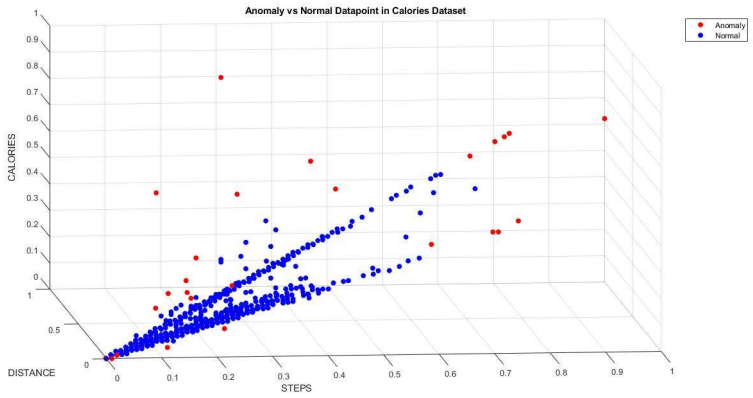


*Figure 4: Three-Dimensional Scatter Plot*

As can be seen in figure 4, most of the data that are far from the normal pattern which exist closely apart are colored red marking the data is anomaly. Therefore, action needs to be taken, since the device has produced faulty data. Based on figure 4, there are 26 anomalies detected by MAAD. Most of these data can be easily understand as true anomalies.

*Table 2: Five Samples of Anomalies Data*

| *Steps* | *Distance* | *Calories* |
|---|---|---|
| 0.82 | 0.76 | 0.3 |
| 0.78 | 0.72 | 0.27 |
| 0.11 | 0 | 0.04 |
| 0.65 | 0.63 | 0.25 |
| 0.23 | 0.23 | 1 |

Table 2 shows sample of data anomalies. This shows that steps taken, distance and calories can be easily corelated. Calories burned are based on steps made and distance travelled. More distance and steps mean more calories burned. For the first row in table 2, there are 0.82 steps and 0.76 m travelled by the user. But the number of calories burned is quite illogical, the user has approximately the same steps and distance, but with higher calories burned. Then, for second row from table 2, the user made 0.78 steps with 0.72 distance travelled. However, calories burned were very much lower than the first row in table 2. Next, for the third row, 0.04 kcal calories burned was quite wrong, since there are normal data from a user who have 0.09 distance travelled with 0.13 steps made which burned the same number of calories. For the fourth row, 0.63 distance travelled with 0.65 steps taken will not burn 0.25 kcal of calories. This is because there are normal data with 0.63 steps and distance, but which burned 0.37 kcal, which is more than the anomaly calory. Finally, the fifth-row records high number of calories burned at 1 kcal with very low number of steps and distance.

As a result, MAAD is proved to detect most of the true anomalies exist in the dataset. Furthermore, the collected dataset has also been tested with streaming TEDA which introduced in [13]. However, streaming TEDA does not detect any anomalies. For the information, streaming TEDA need assumptions or thresholds set before the algorithm run. The threshold was set to three for the first run. Then, for the second run, the threshold is set to 6. But there is not any impact on the threshold change and the algorithm still cannot detect any anomalies. Therefore, MAAD can be said to be the most successful algorithm using streaming data, especially from smartwatch devices.

### 4.2 Time Test

Time tests were conducted using three different algorithms. These algorithms were the proposed MAAD algorithm, autonomous anomaly detection (AAD) [7], and streaming TEDA [13]. Note that

although AAD is not built for streaming data but can be manipulated to receive IoT data continuously. For the information, the more the data presented to anomaly detection algorithms, the more time it takes to process accumulated data. Therefore, over time, the algorithm will become slower since it handles accumulated number of data. In this test, data miss by each algorithm is not evaluated. Each algorithm will receive the same amount of data to test time taken to process the respective data. Figure 5 shows result for streaming TEDA algorithms.

In figure 5, as the data comes in, the graph begins to rise up. Therefore, streaming TEDA cannot handle accumulated streaming data. Furthermore, it uses much time when process streaming data. By the time it is processing previous data, new data may be produced, and the data will not be able to be sent to the streaming TEDA. As a result, streaming TEDA may lose some important data. Then, the collected data were also tested in the AAD algorithm. Figure 6 shows the results for the AAD algorithm.

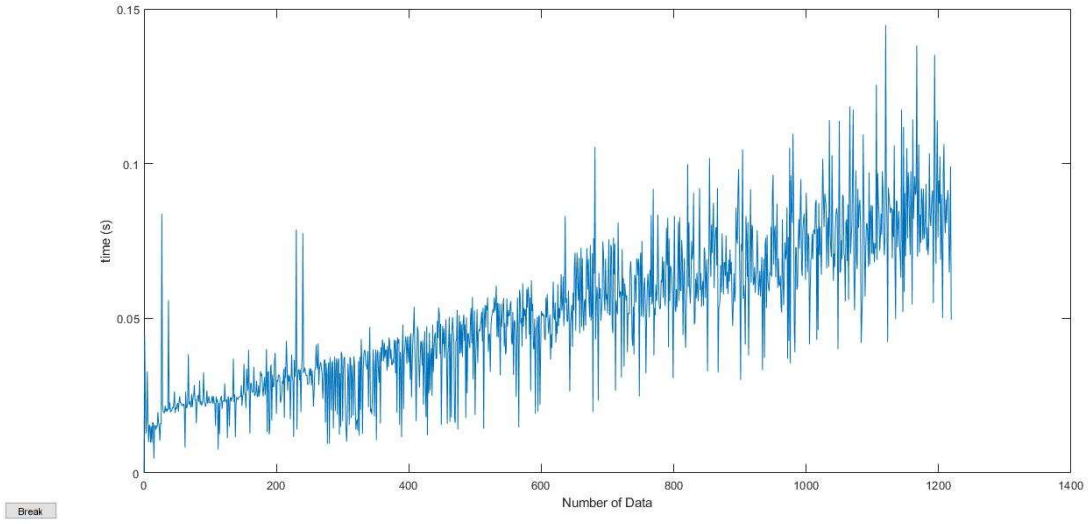


*Figure 5: Time Graph for Streaming TEDA*

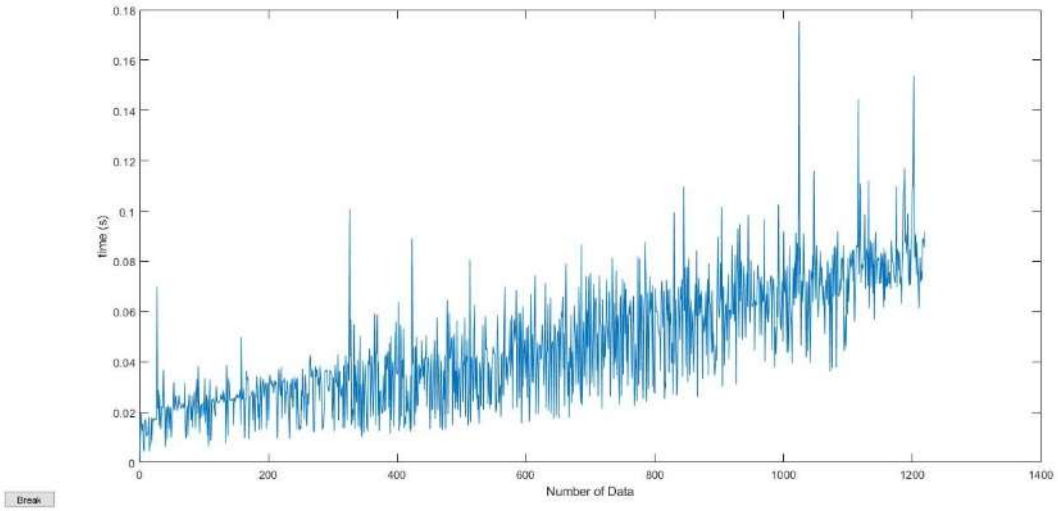


*Figure 6: Time Graph for Streaming AAD*

Based on figure 6, AAD also cannot handle accumulated streaming data. This is shown as the graph rising as time goes on. But AAD is better than streaming TEDA, since the gradient of AAD graph is lower than streaming TEDA. However, AAD will also miss some important data, as in streaming TEDA. Finally, figure 7 shows a graph for MAAD when applied to the collected smartwatch dataset.

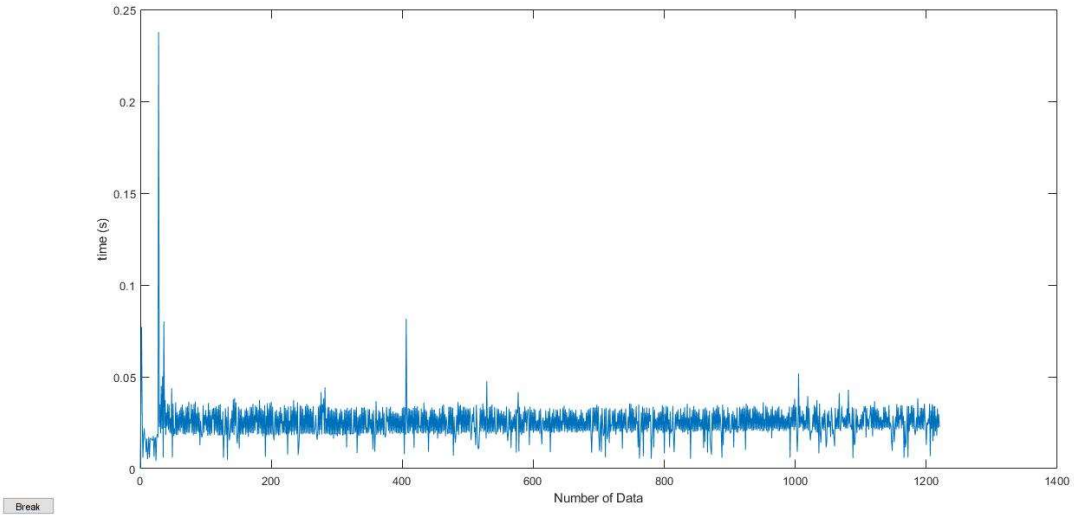


*Figure 7: Time Graph for Streaming MAAD*

The results shown in figure 7 are very impressive. MAAD is not affected by accumulated streaming data. Hence, MAAD can handle streaming data from a smartwatch. It can alert the user that there is an anomaly, and the smartwatch and the MAAD algorithm will miss no data. Table 3 below shows that MAAD is faster in processing streaming data than streaming TEDA and AAD. These results are between the 1000th to 1010th streaming smartwatch data.

Based on table 3, MAAD is very fast, with huge differences between AAD and streaming TEDA. This means MAAD is far better than AAD and streaming TEDA, especially in handling streaming data from a smartwatch. Therefore, MAAD has been proven to work in streaming smartwatch data and solving the main problem, which is that the accumulation of data may make any anomaly detection algorithm work slower as time goes on.

*Table 3: Time Taken for Each Algorithm*

| *Number of Data* | *MAAD(s)* | *AADS(s)* | *Streaming TEDA (s)* |
|---|---|---|---|
| 1000 | 0.0224413 | 0.0919004 | 0.0905143 |
| 1001 | 0.0337205 | 0.0687727 | 0.0821994 |
| 1002 | 0.0221278 | 0.0671864 | 0.0649554 |
| 1003 | 0.0092147 | 0.0767836 | 0.0774021 |
| 1004 | 0.0191979 | 0.632841 | 0.075811 |
| 1005 | 0.0517872 | 0.437898 | 0.0415316 |
| 1006 | 0.0221714 | 0.07803 | 0.0584569 |
| 1007 | 0.0341811 | 0.0437146 | 0.0689901 |
| 1008 | 0.0226402 | 0.0529116 | 0.0697194 |
| 1009 | 0.0286223 | 0.0860684 | 0.0772597 |
| 1010 | 0.0240973 | 0.066473 | 0.0690283 |

## 5. RESEARCH FINDINGS AND DISCUSSION

In this paper, MAAD can be used to explore anomaly data in incoming smartwatch data. Smartwatch data is streaming data. MAAD is proven to handle streaming data faster than streaming TEDA and AAD. MAAD can absorb high-velocity impact from streaming smartwatch data. It can pass the data to anomaly detection algorithms while focusing on other incoming data. As a result, MAAD will not miss any data from time to time, preventing data leakage.

Furthermore, MAAD was also tested for whether it can detect most true anomalies or not. The result shows that MAAD detects most of the true anomalies. This can be seen when MAAD declared data that are far away from the normal pattern as an anomaly. The collected health tracking devices' data were also tested using streaming TEDA. However, streaming TEDA does not detect anomalies whether the thresholds are set to 3 or 6.

Besides, thresholds and assumptions must be avoided as streaming data changes dynamically in the pipeline [5]. MAAD does not use any

assumptions and is entirely an autonomous algorithm. Furthermore, MAAD also has a user-friendly user interface that lets the user view MAAD results and act based on them. Finally, MAAD works asynchronously and is a very impressive algorithm that can drive future researchers to build new anomaly detection algorithms based on it

Although MAAD can work in streaming smartwatch data, without a doubt, there are also some weaknesses in the algorithm. MAAD can only work when a smartwatch has sent more than 27 data. This is because AAD was previously built for offline and not for streaming environments.

Moreover, MAAD was proved to work faster than previously invented AAD. Streaming data cannot be accumulated over time since storage is limited. In MAAD, incoming data is collected and stored. This is because reevaluation is needed every time new data arrive in MAAD. Besides, anomaly data or normal data that had been declared previously may change its form to either normal or anomaly. The recursive update is used in the proposed AAD algorithm to overcome storage issues. It reuses and updates the required variables such as mean, data density and average scalar product. Therefore, MAAD still can work at its best with streaming smartwatch data.

## 6. CONCLUSION

In conclusion, this paper presents an approach to detecting anomaly data from smartwatch devices using the MAAD algorithm. MAAD successfully analyzed incoming streaming data, which continuously changes along the pipeline. Data processing was done successfully from collected smartwatch devices. There are three chosen attributes in the data: steps, distance, and calories since there are correlated. This data is then sent to the IoT pipeline, which MAAD later grabs. MAAD runs two threads simultaneously, preventing any data from being a miss. In the evaluation phase, MAAD was proved to detect most anomalies which are far from the normal pattern. In the time test, MAAD performed faster than AAD and streaming TEDA. This can be proved in the presented graph during the evaluation phase. MAAD shows no impact in an increasing amount of data. The difference between AAD and streaming TEDA increases its processing speed along the time. Finally, this algorithm is believed to detect anomalies in any device that can supply data continuously.

## 7. ACKNOWLEDGEMENT

The authors would like to express gratitude to the Ministry of Higher Education for the FRGS-Racer Research Grant (RACER/1/2019/ICT02/UITM//4) and Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA for all support given.

## REFERENCES:

[1] T. Basaklar, Y. Tuncel, S. An, and U. Ogras, "Wearable Devices and Low-Power Design for Smart Health Applications: Challenges and Opportunities," *Proc. Int. Symp. Low Power Electron. Des.*, vol. 2021-July, p. 6654, 2021, doi: 10.1109/ISLPED52811.2021.9502491.

[2] S. Mekruksavanich and A. Jitpattanakul, "Smartwatch-based Human Activity Recognition Using Hybrid LSTM Network," *Proc. IEEE Sensors*, vol. 2020-Octob, pp. 2020–2023, 2020, doi: 10.1109/SENSORS47125.2020.9278630.

[3] M. Fan, C. Wen, and J. Tang, "Design of a Smartwatch for IoT Application under 5G Environment," *Proc. 2020 IEEE 4th Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2020*, no. Itnec, pp. 306–309, 2020, doi: 10.1109/ITNEC48623.2020.9085198.

[4] T. Bomatpalli and G. J. Vemulkar, "Blending IoT and Big Data Analytics," *Int. J. Eng. Sci. Res. Technol.*, vol. 5, no. 4, pp. 192–196, 2016, doi: 10.5281/zenodo.48868.

[5] L. Rettig, M. Khayati, P. Cudre-Mauroux, and M. Piorkowski, "Online anomaly detection over Big Data streams," *Proc. - 2015 IEEE Int. Conf. Big Data, IEEE Big Data 2015*, pp. 1113–1122, 2015, doi: 10.1109/BigData.2015.7363865.

[6] V. M. Tellis and D. J. D'Souza, "Detecting Anomalies in Data Stream Using Efficient Techniques: A Review," *2018 Int. Conf. Control. Power, Commun. Comput. Technol. ICCPCCT 2018*, pp. 296–298, 2018, doi: 10.1109/ICCPCCT.2018.8574310.

[7] X. Gu and P. Angelov, "Autonomous anomaly detection," *IEEE Conf. Evol. Adapt. Intell. Syst.*, vol. 2017-May, pp. 1–8, 2017, doi: 10.1109/EAIS.2017.7954831.

[8] M. Y. Iqbal Basheer, A. M. Ali, N. H. Abdul Hamid, M. A. Mohd Ariffin, R. Osman, and S. Nordin, "Detecting Anomaly in IoT Devices using Multi-Threaded Autonomous Anomaly Detection," *2021 4th Int. Symp. Agents, Multi-*

*Agent Syst. Robot.*, pp. 111–118, 2021, doi: 10.1109/isamsr53229.2021.9567894.

[9] P. Angelov, X. Gu, D. Kangin, and J. Principe, "Empirical Data Analysis," *2016 IEEE Int. Conf. Syst. Man, Cybern. SMC 2016*, 2017.

[10] B. S. J. Costa, P. P. Angelov, and L. A. Guedes, "Real-time fault detection using recursive density estimation," *J. Control. Autom. Electr. Syst.*, vol. 25, no. 4, pp. 428–437, 2014, doi: 10.1007/s40313-014-0128-4.

[11] N. H. Abdul Halim, "Lifestyle Data Analytic using Wearable Device," *Univ. Teknol. Mara, Malaysia*, no. July, p. 109, 2019.

[12] R. Nawaratne, D. Alahakoon, D. De Silva, and X. Yu, "Spatiotemporal anomaly detection using deep learning for real-time video surveillance," *IEEE Trans. Ind. Informatics*, vol. 16, no. 1, pp. 393–402, 2020, doi: 10.1109/TII.2019.2938527.

[13] L. M. D. da Silva *et al.*, "Hardware architecture proposal for TEDA algorithm to data streaming anomaly detection," *arXiv*, pp. 1–27, 2020.