# HCMUS in MediaEval 2022 - Urban Air - Periodic frequent pattern discovery

Duc-Huy Tran[1,2,*,†], Vinh-Thuyen Nguyen-Truong[1,2,†], Xuan-Cuong Le[1,2,†], Minh-Triet Tran[1,2,†] and Hai-Dang Nguyen[1,†]

[1]Ho Chi Minh City University of Science, National University Vietnam in Ho Chi Minh City, Vietnam
[2]John von Neumann Institute, National University Vietnam in Ho Chi Minh City, Vietnam

Abstract

We propose a method to discover periodic traffic-pollution patterns in Dalat City, Vietnam. From the air pollution, weather, and CCTV stations data recorded in Dalat City, Vietnam, we convert them into an hourly uncertain temporal database and apply a periodic frequent pattern discovery algorithm to explore the most occurring patterns in the dataset, as well as finding some insights from the found patterns.

## 1. Introduction

In 2019, the WHO considers air pollution is the greatest environmental risk to health. Many activities of citizens in urban areas cause negative impacts on the environment. Because of that, many regulations have been made to guide, restrict and give punishments for citizen actions that worsen pollution. If we know the mutual relationship between urban life activities and air pollution, and the patterns that are most frequent and detrimental to the pollution. then governments can provide better and more efficient regulations, and citizens can act properly to lessen the damage caused to our environment.

In order to achieve that, we seek to create a multi-step process to find periodic frequent patterns in the UrbanAir task at the MediaEval 2022 workshop [1]. The task contains a prediction task on the future AQI levels using data collected in Datlat City, and finding insights by discovering periodic traffic-pollution patterns. Our contributions are:

- We propose a method to discover a traffic-pollution periodic frequent pattern in the given dataset. We attempted to change our implementation to achieve faster runtime while still producing many results.
- To our surprise, the most intriguing insight we gained from our experiments was that the air pollution remained very low, even with high traffic signals. Most of the time, the traffic signal are low.

## 2. Related Work

Research has recently been trying to discover the feasibility of various types of data to predict or understand the mutual relationship between human activities and air pollution. Mike et al

*MediaEval'22: Multimedia Evaluation Workshop, January 13–15, 2023, Bergen, Norway and Online*
*Corresponding author.
†These authors contributed equally.
huy.tran2021@ict.jvn.edu.vn (D. Tran); ntvthuyen@apcs.fitus.edu.vn (V. Nguyen-Truong); cuong.le2021@ict.jvn.edu.vn (X. Le); tmtriet@fit.hcmus.edu.vn (M. Tran); nhdang@selab.hcmus.edu.vn (H. Nguyen)

[2] introduces a large-scale dataset from 77 air monitoring and 580 weather stations in Taiwan. Additionally, they propose a machine learning approach to predict future PM2.5 values and evaluate the dataset. Minh-Son et al [3] investigate the feasibility of predicting AQI using the image captured by personal devices such as smartphones. With the weather, AQI and lifelog dataset, they conclude that there is a correlation between AQI and the environment's snapshots. Phuong-Binh et al [4] propose a method that utilizes lifelog data to associate the visual data - AQI rank relationship and to predict AQI ranks using visual data. Inspired by [5], we aim to find an interesting pattern to explain and find a connection between traffic and air quality.

# 3. Approach

## 3.1. Prepare and preprocess the data

After collecting CCTV data, we discover that the most continuous period is from 13 August 2022 to 31 October 2022. As a result, we download all CCTV images during this time period for training purposes (220463 images). We use the Yolo-V5 [6] (SOTA real-time instance segmentation), which is well trained on the COCO dataset [7] to detect and count five main traffic object types: *person*, *motorbike*, *car*, *truck*, and *bus* (more at Appendix A). The COCO dataset is comparable to our CCTV data because our five target objects are five of the main detected objects of it. Therefore, we decide to use the pre-trained model on the COCO dataset. Then we use the latitude and longitude to calculate the nearest cameras for each sensor (see appendix B). Lastly, following the organizers' document on how to convert AQI, and their statement on using 1-hour averages for AQI conversion, we group the values into each hour and calculate AQI based on each hour's average concentration values for air pollutants.

## 3.2. Discover periodic frequent patterns

After we have the table from 3.3, for each row, we apply the fuzzy negation method mentioned in [5] to generate the items and make the transactions, with the setting shows in Table 1:

**Table 1**
Initial fuzzy negation mapping for columns

| Other columns | | | AQI columns | | | | | |
|---|---|---|---|---|---|---|---|---|
| LOW | MED | HIGH | lv1 | lv2 | lv3 | lv4 | lv5 | lv6 |
| 12.5% | 25% | 50% | 5 | 12 | 45 | 100 | 200 | 300 |

Taking ideas from [5], we use a separate fuzzy mapping for AQI columns. This is due to our processing the AQI-related items differently - instead of calculating the percentage of [min, max] range first, we directly use its values to convert with the same mapping for all AQI columns. See Appendix C for more details on why we choose to do so.

Each transaction will also be linked with a timestamp. A default time is chosen as the beginning of time. We simply subtract all datetime values with the beginning datetime (converted to hour) to get the timestamp.

After this step, we'll have an uncertain temporal database. We reimplement the algorithm from [8] to find periodic frequent patterns in our uncertain temporal database.

## 3.3. Discover periodic frequent patterns (updated)

In this attempt, we decided to *use all cameras* for all sensors (each sensor will be linked with images from all cameras, that are closest in sensor data's time).

We made some changes in our implementation. Aside from fixing some errors, we change the way we calculate the prefixed item cap into using the minimal probability of previous items in each transaction: $PI\_cap(i_k, t_{tid}) = P(i_k, t_{tid}) * min(P(i_1, t_{tid}), P(i_2, t_{tid}), ...P(i_{k-1}, t_{tid}))$.

We also changed our fuzzy negation mapping into as follow:

**Table 2**
Updated fuzzy negation mapping for columns

| Other columns | | | AQI columns | | | | |
|---|---|---|---|---|---|---|---|
| LOW | MODERATE | HIGH | lv1 | lv2 | lv3 | lv4 | lv5 |
| 10% | 20% | 40% | 5 | 12 | 50 | 100 | 200 |

Furthermore, we decided to diminish all items with a probability below 0.5. This means no two items from the same feature (for example, *Temperature_LOW* and *Temperature_MED*) can appear in one transaction. By doing this, all transactions will have statistically independent items, assuming that the features are statistically independent.

We also split a subset of data with only traffic-related features and AQI, to explore the interesting patterns between traffic and air pollution. We customized our search to use each and only one AQI item in one mining. Please refer to our source code [9] for more information.

There are also a few temporal gaps in our database, which can be at most 120 hours (5 days). It can be because of some sensors and cameras not working for a moment, leading to no data recorded during those hours. This gap affects the periodicity of those patterns because of their high values (since the periodicity is calculated by the maximum difference between occurring timestamps). So we delete a part of our database to lower the effect of these gaps on our results.

## 4. Results and Analysis

The results reported here are only a notable part of our results.

### 4.1. Results

Because of the changes in our implementations, we provide our updated results for the full data mining (using all features) for clarity. We do not think, however, that the new result will change our insight. So we also provide new results from mining patterns with each AQI-related item the only AQI-related item present in the database, to see which most frequent factors contribute to each AQI level.

### 4.2. Analysis

Our patterns shows that the traffic is generally low, and the same can be said for the pollutant AQI values. Even with the highest levels of traffic (HIGH label), the AQI values mostly stay low - so air quality in Da Lat is generally good. However, with high traffic signal, especially a high number of motorbike, the AQI can still rise and thus affect the environment.

In the customized AQI item mining, we find it interesting to see that the patterns with highest expected support are mostly low traffic signals, while the higher traffic signals come from most periodic (lowest periodicity) patterns.

We also found that changing the way we calculate the prefixed item cap make the second algorithm run faster than the previous implementation, but with fewer patterns found. Due to stronger bounds to the actual probabilities, more recursive calls are pruned, thus results in less time and (potentially) fewer patterns.

**Table 3**
Example results from our updated approach

| |
|---|
| **Initial (submitted) results**<br>*Full data (minSup = 4000, maxPer = 120)* |
| Altitude_HIGH, Truck_LOW, AQI_NO2_lv1, Humidity_HIGH, AQI_SO2_lv1: [5630.8, 120.0]<br>AQI_CO_lv2, Bus_LOW, Altitude_HIGH, Truck_LOW, AQI_NO2_lv1, Humidity_HIGH, AQI_SO2_lv1: [4752.2, 120.0] |
| **Results from updated implementation**<br>*Full data (minSup = 200, maxPer = 1000)* |
| Motorbike_LOW, AQI_O3_lv1, Person_LOW, AQI_CO_lv2, Truck_LOW, Altitude_HIGH, UV_LOW, Bus_LOW, AQI_NO2_lv1, Temperature_HIGH, AQI_SO2_lv1, Humidity_HIGH:[698.87, 63.0] |
| *Traffic and AQI features only (minSup = 3, maxPer = 1000)* |
| Motorbike_LOW, AQI_PM2.5_lv3, Car_MODERATE, AQI_O3_lv1, Person_LOW, Truck_LOW, AQI_PM10_lv2, AQI_CO_lv2, Bus_LOW, AQI_NO2_lv1, AQI_SO2_lv1:[143.128, 92.0]<br>Motorbike_HIGH, AQI_PM10_lv2, Car_MODERATE, Person_MODERATE, AQI_O3_lv1, Truck_LOW, AQI_PM2.5_lv3, Bus_LOW, AQI_CO_lv2, AQI_SO2_lv1, AQI_NO2_lv1:[42.44, 81.0] |
| *AQI specific mining patterns* |
| Person_HIGH, Car_MODERATE, Truck_LOW, Motorbike_HIGH, Bus_LOW, AQI_CO_lv5:[17.36, 200.0]<br>Motorbike_HIGH, Car_MODERATE, Truck_LOW, Person_MODERATE, Bus_LOW, AQI_PM2.5_lv5:[2.53, 688.0]<br>Motorbike_HIGH, Car_MODERATE, Person_MODERATE, Truck_LOW, Bus_LOW, AQI_O3_lv5:[22.95, 163.0] |
| *High traffic specific signal patterns* |
| AQI_PM10_lv2, AQI_O3_lv1, AQI_PM2.5_lv3, AQI_CO_lv2, AQI_SO2_lv1, Motorbike_HIGH, AQI_NO2_lv1:[353.91, 55.0] |

## 5. Discussion and Outlook

Thanks to its usually low traffic, combined with its environments and other factors, we can conclude that Da Lat's air quality is overall good. High traffic signal, however, can still worsen air pollution.

## 6. Future work

The algorithm is slow when used for a large database with more than 20 frequent items and many timestamps. Hence, it's recommended to optimize this algorithm, especially proposing stronger upper and lower bound to reduce the patterns needed to recursively explore. We provide our source code [9] for further experiments.

## 7. Acknowledgement

Duc-Huy Tran, Vinh-Thuyen Nguyen-Truong, Xuan-Cuong Le were funded by Vingroup JSC and supported by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Institute of Big Data, codes VINIF.2021.ThS.JVN.09, VINIF.2021.ThS.JVN.10, VINIF.2021.ThS.JVN.08.

## References


[1] M.-S. Dao, T.-H. Dang, T.-L. Nguyen-Tai, T.-B. Nguyen, D.-T. Dang-Nguyen, Overview of MediaEval 2022 Urban Air: Urban Life and Air Pollution (2022) 5.

[2] M. Lee, L. Lin, C.-Y. Chen, Y. Tsao, T.-H. Yao, M.-H. Fei, S.-H. Fang, Forecasting air quality in taiwan by using machine learning, Scientific Reports 10 (2022). URL: UR-https://doi.org/10.1038/s41598-020-61151-7. arXiv:10.1038/s41598-020-61151-7.

[3] M.-S. Dao, K. Zettsu, U. K. Rage, Image-2-aqi: Aware of the surrounding air qualification by a few images, in: Advances and Trends in Artificial Intelligence. From Theory to Practice: 34th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2021, Kuala Lumpur, Malaysia, July 26–29, 2021, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2021, p. 335–346. URL: https://doi.org/10.1007/978-3-030-79463-7_28. doi:10.1007/978-3-030-79463-7_28.

[4] P.-B. Vo, T.-D. Phan, M.-S. Dao, K. Zettsu, Association model between visual feature and aqi rank using lifelog data, 2019 IEEE International Conference on Big Data (Big Data) (2019) 4197–4200.

[5] T.-V. La, M.-S. Dao, K. Tejima, R. U. Kiran, K. Zettsu, Improving the Awareness of Sustainable Smart Cities by Analyzing Lifelog Images and IoT Air Pollution Data, in: 2021 IEEE International Conference on Big Data (Big Data), IEEE, Orlando, FL, USA, 2021, pp. 3589–3594. URL: https://ieeexplore.ieee.org/document/9671403/. doi:10.1109/BigData52589.2021.9671403.

[6] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammana, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Y. , changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, P. Rai, ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, 2020. URL: https://doi.org/10.5281/zenodo.4154370. doi:10.5281/zenodo.4154370.

[7] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: common objects in context, CoRR abs/1405.0312 (2014). URL: http://arxiv.org/abs/1405.0312. arXiv:1405.0312.

[8] R. Uday Kiran, P. Likhitha, M.-S. Dao, K. Zettsu, J. Zhang, Discovering Periodic-Frequent Patterns in Uncertain Temporal Databases, in: T. Mantoro, M. Lee, M. A. Ayu, K. W. Wong, A. N. Hidayanto (Eds.), Neural Information Processing, volume 1516, Springer International Publishing, Cham, 2021, pp. 710–718. URL: https://link.springer.com/10.1007/978-3-030-92307-5_83. doi:10.1007/978-3-030-92307-5_83, series Title: Communications in Computer and Information Science.

[9] D.-H. Tran, Source code for urban air pf pattern discovery, 2022. URL: https://github.com/TranDucHuyVnuHcmUs/MediaEval2022-UrbanAir-HCMUS-public.

[10] Geopy, Geopy, 2022. URL: https://geopy.readthedocs.io/en/stable/.

# A. Traffic object statistics

We execute the pre-trained YOLO V5 on 20463 images from 16 CCTV stations. Afterward, we accumulate all the results by CamereID of *person*, *motorbike*, *car*, *truck*, and *bus*. We choose these five types of objects because CCTV mostly covers them (the number of other types is extremely small compared to these 5 main objects).

**Table 4**
The average table of traffic factors of CCTV dataset from 13 August 2022 to 31 October 2022. .

| CameraCode | Person | Motorbike | Car | Truck | Bus |
|---|---|---|---|---|---|
| Camera01 | 9.146 | 11.378 | 0.715 | 0.147 | 0.025 |
| Camera02 | 2.481 | 3.684 | 5.341 | 0.791 | 0.237 |
| Camera03 | 3.497 | 4.201 | 1.406 | 0.035 | 0.003 |
| Camera04 | 3.574 | 8.319 | 2.247 | 0.115 | 0.043 |
| Camera05 | 2.901 | 3.94 | 6.09 | 0.73 | 0.333 |
| Camera06 | 5.618 | 5.229 | 3.225 | 0.18 | 0.128 |
| Camera07 | 0.933 | 2.61 | 0.254 | 0.016 | 0.001 |
| Camera08 | 1.361 | 6.232 | 2.177 | 0.054 | 0.002 |
| Camera09 | 1.425 | 4.344 | 0.943 | 0.07 | 0.08 |
| Camera10 | 6.19 | 8.786 | 2.84 | 0.159 | 0.13 |
| Camera11 | 1.745 | 1.497 | 2.72 | 0.126 | 0.086 |
| Camera12 | 3.547 | 6.038 | 2.264 | 0.066 | 0.032 |
| Camera13 | 1.247 | 1.991 | 0.475 | 0.059 | 0.004 |
| Camera15 | 3.462 | 5.016 | 4.218 | 0.274 | 0.056 |
| Camera16 | 0.008 | 0.0 | 0.002 | 0.0 | 0.0 |

According to the result of YOLO-V5 on our CCTV data (Table 4), Camera16 doesn't find too many traffic objects (to our own knowledge, Camera16 mostly points to natural scenes from above). On the other hand, Camera 01 takes the most crowded pictures. We also have a broken camera - Camera14.

Thís approach are prone to error: Idle objects, especially parking vehicles, though not emitting chemicals, can still be included.

# B. Merging sensor data with camera data

There are two attributes that need to consider before merging: time and space.

For the space, we calculate the geographical distances between all pairs of camera-sensor, and are calculated using *GeoPy* [10]. In the first attempt, we merged the traffic data from every sensor into its nearest camera, as shown in the table below:

For the time, we only merge data rows of the camera-sensor with their time difference not larger than a defined threshold. In our initial setting, this threshold is $329.015(7) = (maxDistance/minDrivingSpeed) * 3600$, with $minDrivingSpeed = 32(km/h)$, and $maxDistance$ is the maximal value of each nearest camera-sensor pair. The intuition is that vehicles are mostly moving, and since the sensors recorded their values at a different time compared to the cameras, we assume that these vehicles would have moved a distance by the time each sensor's respective camera took another picture.

In our second attempt, we merged all cameras with all sensor data, and use 300 (second) as tolerance constant to merge the data by time (which means camera rows can only match with sensor rows if their time difference is no more than 300 seconds).

**Table 5**
The nearest cameras for each sensor, and their respective distance

| SensorCode | CameraCode | Distance |
|---|---|---|
| Sensor01 | Camera11 | 0.4150114345835359 |
| Sensor02 | Camera15 | 0.6724809262023985 |
| Sensor03 | Camera09 | 0.5866375637048777 |
| Sensor04 | Camera06 | 0.4050335834399718 |
| Sensor05 | Camera16 | 0.820374327598685 |
| Sensor06 | Camera12 | 1.0222252540677688 |
| Sensor07 | Camera10 | 1.5597868694504484 |
| Sensor08 | Camera12 | 2.9245846846435417 |
| Sensor09 | Camera13 | 0.8956313587788634 |
| Sensor10 | Camera06 | 1.7211426697457426 |

## C. Fuzzy negation

Two tables below explain our initial mappings (see table 1) used for our submission. For values in one column X (not related to AQI) that are in 0% - 12.5% of its [min, max] range, these transactions will be given an item as X_LOW(1). For values in 18.75% (average of 12.5% and 25%) of its range, those transactions will have 2 items X_LOW(0.5) and X_MED(0.5). Note that both items will exist in those transactions.

**Table 6**
Initial fuzzy negation mapping for other columns (explained)

| Percentage of value | Label |
|---|---|
| 0% - 12.5% | LOW(1) |
| 12.5% - 25% | LOW(1-0)/MED(0-1) |
| 25% - 50% | MED(1-0)/HIGH(0-1) |
| 50% - 100% | HIGH(1) |

We process the AQI-related items differently - instead of calculating the percentage of [min, max] range first, we directly use its values to convert. Because the meaning and purpose of AQI is a unified metric for measuring the concentration of (and health affect caused by) air pollutants, and each air pollutant has a different [min, max] range, we don't convert the values to percentages to not lose the unified meaning.

**Table 7**
Initial fuzzy negation mapping for AQI columns (explained)

| AQI range | Label |
|---|---|
| 0 - 5 | lv1(1) |
| 5 - 12 | lv1(1-0)/lv2(0-1) |
| 12 - 45 | lv2(1-0)/lv3(0-1) |
| 45 - 100 | lv3(1-0)/lv4(0-1) |
| 100 - 200 | lv4(1-0)/lv5(0-1) |
| 200 - 300 | lv5(1-0)/lv6(0-1) |
| >= 300 | lv6(1) |

It's worth noticing that our mapping is not the same as the AQI conversion table provided for us: lv1, lv2, and lv3 are in the 'GOOD' category. Our reasoning to divide the first 3 levels into such small gaps is because the average AQI values of each air pollutant are below 50 ('GOOD'

category). Especially for NO2, SO2 and PM10 air pollutant, their mean values are lower than 14, and for SO2, its max values is 14.285714.

**Table 8**
AQI columns' statistics of our grouped table (used for making uncertain temporal databases), consisting of 7098 data rows.

| Attribute | AQI_O3 | AQI_NO2 | AQI_SO2 | AQI_CO | AQI_PM2.5 | AQI_PM10 |
|---|---|---|---|---|---|---|
| mean | 39.15 | 1.9 | 1.44 | 38.61 | 44.99 | 13.01 |
| std | 96.13 | 2.21 | 0.21 | 88.85 | 30.12 | 11.63 |
| min | 0.0 | 0.0 | 0.0 | 13.64 | 0.0 | 0.0 |
| 25% | 0.0 | 0.0 | 1.43 | 13.64 | 20.83 | 4.63 |
| 50% | 0.0 | 0.94 | 1.43 | 13.64 | 41.25 | 9.26 |
| 75% | 0.0 | 3.77 | 1.43 | 13.64 | 63.41 | 17.59 |
| max | 501.0 | 15.09 | 14.29 | 501.0 | 342.82 | 188.12 |

## D. Performance of the algorithm

Our experiments with both *max* and *min* functions ('mode') to calculate the prefixed item cap show that the second algorithm run faster than the previous implementation, and with fewer potential patterns (patterns that may meet the requirements, but not ensured). Due to stronger bounds to the actual probabilities, more recursive calls are pruned, thus results in less time and fewer potential patterns to be considered, and will be included or filtered out in the final step.

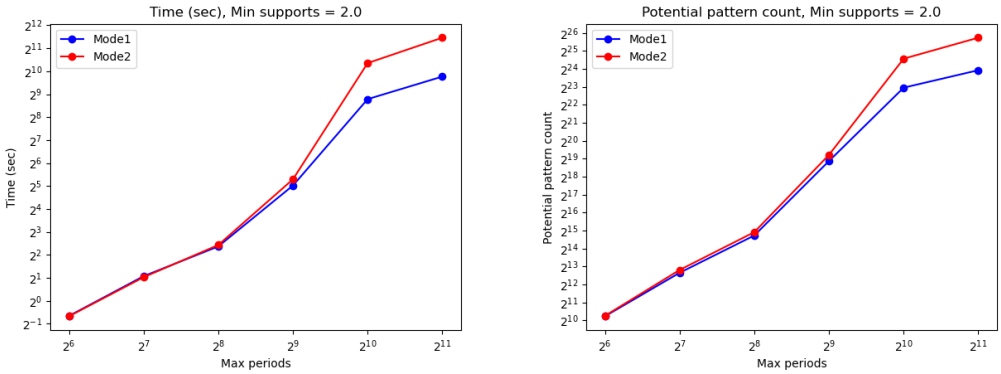


**Figure 1:** Running time and the number of potential pattern from our implementation with different max period and min support = 2. Mode 1 use the *min* function, mode 2 use the *max* function.

Our results also show that two modes can produce different longest patterns, both potential and final.

High difference (more than 1) of the longest length between potential patterns and final patterns can be a problem, as too many potential patterns are not satisfying the requirements, thus increasing runtime while not gaining better results. Therefore, it's our future works to reduce this difference. An improvement that can be made is to introduce closer bounds to prune more recursive calls.

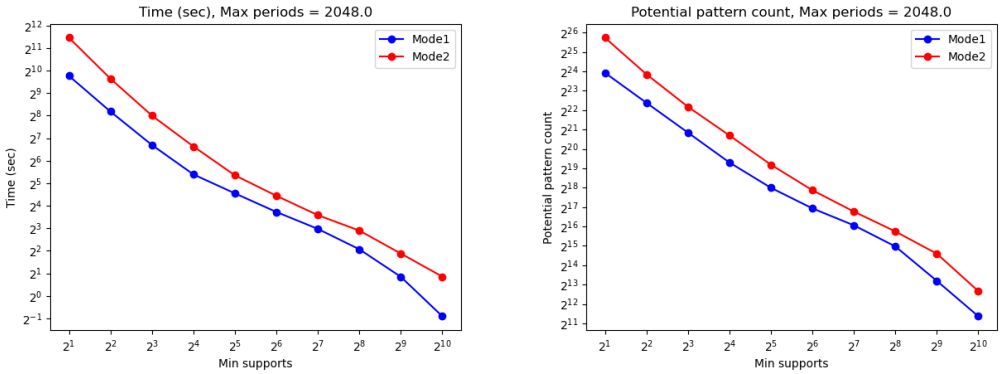


**Figure 2:** Running time and the number of potential pattern from our implementation with different min supports and max period = 2048. Mode 1 use the *min* function, mode 2 use the *max* function.

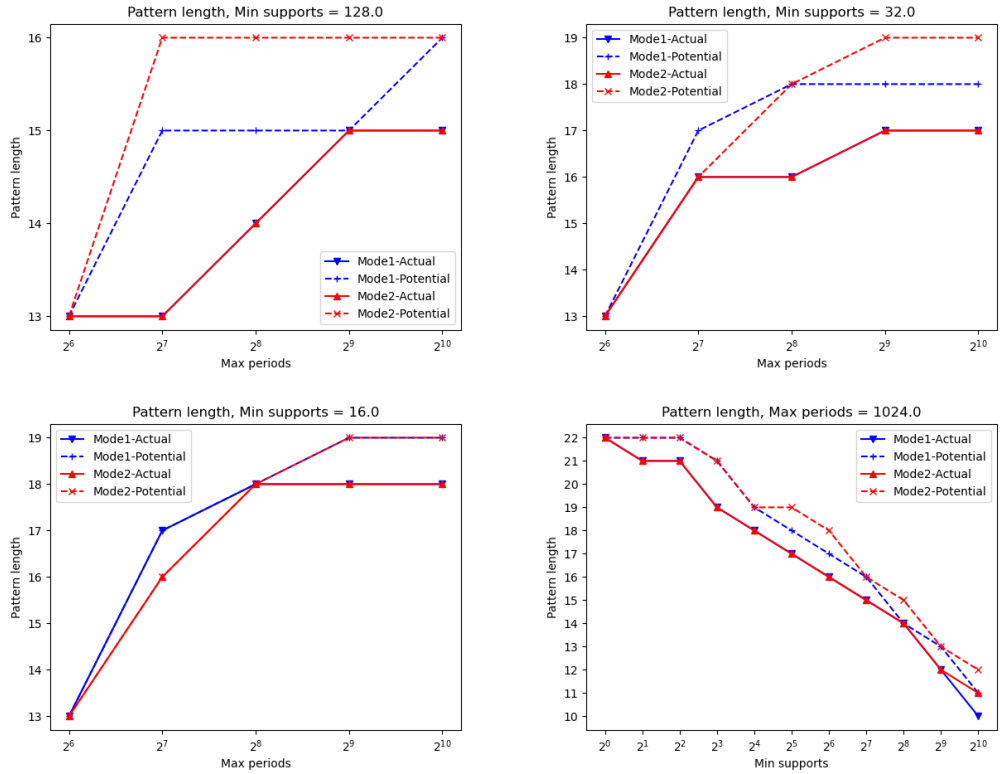


**Figure 3:** Longest lengths of potential and final pattern. Longer patterns mean better results.