Int. J. Nonlinear Anal. Appl.
Volume 12, Special Issue, Winter and Spring 2021, 959-970
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/IJNAA.2021.5543

The implementation of sax and random projection for motif discovery on the orbital elements and the resonance argument of asteroid

Lala Septem Riza[a,*], Muhammad Naufal Fazanadi[a], Judhistira Aria Utama[b], Taufiq Hidayat[c], Khyrina Airin Fariza Abu Samah[d]

[a]*Department of Computer Science Education, Faculty of Mathematics and Natural Science Education, Universitas Pendidikan Indonesia, Bandung, Indonesia.*
[b]*Department of Physics Education, Faculty of Mathematics and Natural Science Education, Universitas Pendidikan Indonesia, Bandung, Indonesia.*
[c]*Astronomy Research Division, Faculty of Mathematics and Natural Science, Institut Teknologi Bandung, Bandung, Indonesia.*
[d]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Cawangan Melaka Kampus Jasin, Melaka, Malaysia.*

(Communicated by Madjid Eshaghi Gordji)

Abstract

Motif discovery has emerged as one of the most useful techniques in processing time-series data. One of the implementations of motif discovery is in case study 1:1 mean motion resonance (MMR) in the astronomy field. This study aims to build a computational model and its implementation to process time-series data and predict 1:1 MMR from asteroid orbital elements in time-series form. This model proposes Symbolic Aggregate approximation (SAX) and Random Projection (RP) algorithms implemented in the Python programming language. Some experiments involving ten asteroids' orbital elements data have been carried out to validate the program. From the results obtained, we conclude that our computational model can predict the location of the motif and with which planet the motif is found for 1:1 resonance to occur.

Keywords: Motif discovery, Astrophysics, Random projection, Time series

*Corresponding author
Email addresses: lala.s.riza@upi.edu (Lala Septem Riza), mnf_naufal@student.upi.edu (Muhammad Naufal Fazanadi), j.aria.utama@upi.edu (Judhistira Aria Utama), taufiq@as.itb.ac.id (Taufiq Hidayat), khyrina783@uitm.edu.my (Khyrina Airin Fariza Abu Samah)

Received: June 2021 *Accepted:* September 2021

1. Introduction

In processing time-series data, the motif discovery method is beneficial in finding knowledge. These are some examples of time-series data motif discovery methods in recent times. Readers can find in the kinds of literature research conducted by Liu et al. [17] using the motif discovery algorithms with large amounts of data, the applications of matrix profiles by Pariwatthanasak & Ratanamahatana [23], and implementing time-series motif discovery for uncomplete data in the work of Zhu et al. [34]. Many researchers have used motif discovery time-series in diverse domains of science such as health [1], biology [2], economics [22], entertainment [5], seismology [35], telemedicine [8], robotic [20], [11], and nature phenomena [3], [27].

This motif discovery method can be divided into 2 methods: exact and approximate methods [17]. Mueen and Keogh [21] were able to find motifs appropriately but were often inefficient when dealing with large datasets. However, the approximate method by Jessica [14] has higher efficiency but lower accuracy. Furthermore, additional tasks are needed to determine the form of the motif when the pattern is not known a priori. Motif detection problems that were previously unknown and occurred several times in time-series data are interesting to discuss and solve [15], [16].

An example of a problem for motif discovery by not knowing the motif to be sought is looking for 1:1 mean motion resonance (MMR) [7, 31, 25] on the dataset of asteroid's orbital elements. The data is in the form of time-series involving six important orbit parameters in the Keplerian component, namely a (semimajor axis), e (eccentricity), i(inclination), Ω (longitude of ascending node), ω (argument of perihelion), and M (mean anomaly) [26]. The MMR is a dynamical mechanism such that a celestial body (for example, an asteroid) has its orbital period commensurable with the orbital period of a particular planet. We can determine the MMR location from the sun as the parent body for both previously mentioned objects using the 3rd Kepler Law.

One indicator for the occurrence of a 1:1 MMR between an asteroid and a particular planet is that the value of its semimajor axis is equal to the planet's value. This can be seen visually from the asteroid's semimajor axis plot as a function of time in the asteroid's orbit evolutionary history. One popular tool to check the MMR state between asteroid and planet is the SwiftVis application developed by Lewis and Levison and can be accessed at https://www.cs.trinity.edu/~mlewis/SwiftVis/. Therefore, this research focuses on building computational models and implementing motif discovery algorithms that can provide motif found in the time-series data of asteroid orbital elements. There are three main stages in this study. Those stages consist of pre-processing data stage to clear data, the main stage using the Symbolic Aggregate approximation (SAX) algorithm [14] and Random Projection (RP) algorithm [4], and the last, post-processing data to figure the correct information.

2. Methodology

Figure 2 shows the computational model in this work, whereas an example of a motif to be found shown in Figure 1. The computational model consists of several stages. The first stage is that the system will receive asteroids' orbital elements as time-series input data. Before the processing and calculating started, we needed to clean the data to be processed. At this stage, the pre-process time-series data input will be carried out. This pre-processing stage is cleaning the raw data to be processed by the following process. Next, the data will be converted into a data frame in Python, perform column cleaning, and remove the rest of the useless data.

After the data is ready to use, the next step is the Normalization data stage. The time-series data is normalized by using Z normalization so that all information is proportional to be processed. The reason for using this normalization is because time-series data tends to have data with a normal distribution. The formula of Z normalization is shown in equation (1). The Z is for normalized data,

the x_i symbol represents each data point, the x (with bar on top) symbol is the average value of time-series, and S for the standard deviation. The normalization process begins with minimalizing each data with the average of all data, then divided by the standard deviation of time-series. The output of this process is a normalized time-series.

$$Z_i = \frac{x_i - \bar{x}}{S} \qquad (1)$$

This stage and the next are the discovery motif stage in the time-series. The pattern in time-series can be a motif if two or more subsequence m have identical (or almost identic) lengths. The normalized time-series data will be converted into SAX to find the motifs, and the data produced in string form are processed by the motif discovery random projection algorithm. The SAX method allows time-series data to be converted into a string with the desired number of character or alphabets on strings. The alphabet size is an arbitrary integer a, where $a > 2$. This discretization procedure is unique because it uses intermediate representations between raw time-series and symbolic strings. This algorithm consists of two steps: (i) time-series transformation into PAA representation to reduce dimension, and (ii) PAA conversion into a symbolic representation of time series in the form of a string. In this work, we do not use the time-series transformation stage because all data is needed to be processed. The process of SAX algorithm is defined in the pseudocode in Figure 4 and the example of the time-series discretization is illustrated in Figure 3. After using the SAX algorithm and getting the output string data, the next step is to find the previously unknown motif using the Random Projection (RP) Algorithm. This algorithm is based on the problems of Planted Motif Search (PMS) in the biology computational. PMS aims to find all the motifs appear in every DNA sequence [19].

Random Project is one of the algorithms used to find motifs in DNA sequences [24]. In this algorithm, pieces of input data in the form of sub-sequences (l-mers) will be projected according to random positions determined based on the value of k (k-mers) [4]. Random projections represent that mutation can occur anywhere, so random projections are done randomly [28, 9]. After that, post-processing will be done to process the results of random projection into information needed by the user. This post-processing includes combining motifs found in rows (1 gap for each location of the motif), counting the gap of motif, and filtering at any location with 1:1 MMR. The output is the location of the motif, the occurance time for the motif, and the distance for the motif is found.

Figure 1: Time-Series Containing 3 Sub-sequences that are Almost Identical

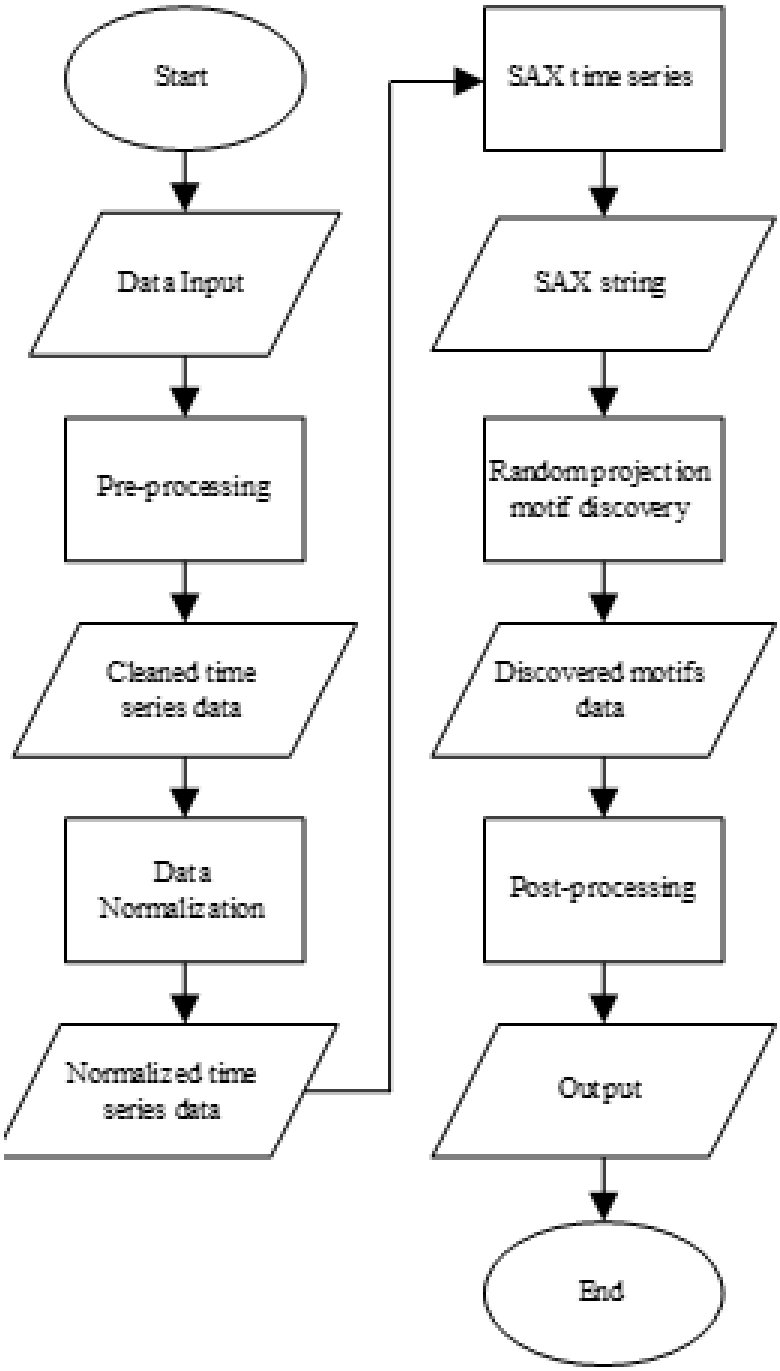

Figure 2: Research Computational Model

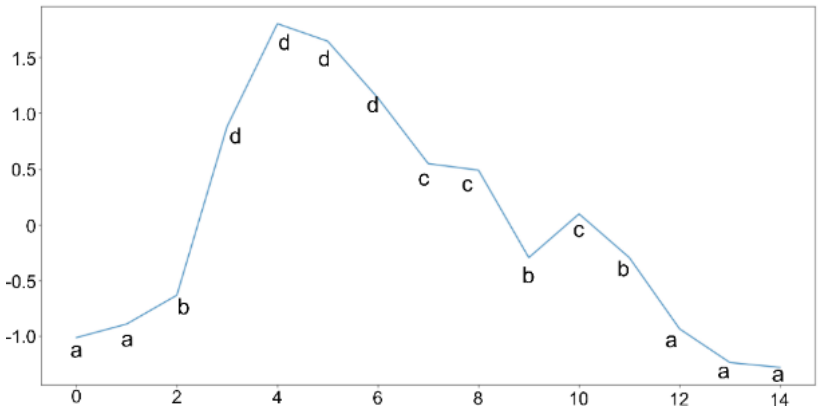

Figure 3: SAX Symbol with the Number of Symbol Divisions (a) = 4. Time Series has been Changed to the Word "aabdddccbcbaaa"

```
input: data time series (series),
           number of letter on SAX (a)
output: String representing symbols on time series datasets

process:
/* create dictionary */
breakpoints = {
  2: [-infinite, 0.00],
  3: [-infinite, -0.4307273, 0.4307273],
  4: [-infinite, -0.6744898, 0, 0.6744898]
}
/* create array consisting of letters */
alphabet=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p',
'q','r','s','t']

result = ""

for i = 0 : length(series) do
  num = series[i]
  if num >= 0 then
     j = a - 1
     while (j > 0) and (breakpoints[j] >= num)) do
        j = j – 1
     end while
     result.append(alphabet[j])
  else
     j = 1
     while (j < a) and (breakpoints[j] <= num) do
        j = j + 1
     end while
     result.append(alphabet[j])
  end if
end for

return result
```

Figure 4: Pseudocode SAX Algorithm

3. Experimental study

3.1. Problem statement

In celestial mechanics, periodic disturbances experienced by an object such as an asteroid approaching a planet can increase if the planet and asteroid share the same direction and motion. This periodic disturbance is a gravitational effect known as orbital resonance that determines the dynamical structure of the solar system. In the solar system, there are many orbits of celestial objects that have a commensurability of the frequencies of orbital revolution in the form of a simple integer ratio. Look at two arbitrary celestial objects that each have a mean motion of n_1 and n_2. The resonance conditions of the orbit (in this case, the mean motion resonance or MMR) are fulfilled when n_1/n_2 is close to a small ratio of integers, $(p+q)/p$, where $p \neq 0$ and $q \geq 0$ are integers. In a resonance state, the objects involved have periodic effects of gravity on one another, making the orbit stable or unstable. The location of the MMR condition can be calculated from the planet's and asteroid's orbital period (P, which is equal $2\pi/n$) information, that is directly related to the average distance (a) of the object from the sun, as written in equation (2).

$$\frac{n_2}{n_1} = (\frac{a_1}{a_2})\frac{S}{2} \qquad (2)$$

Orbit stabilization can occur when both objects in the MMR condition move synchronously, making them never close in their respective orbits. An example of this is the dwarf planet Pluto and planet Neptune, although it is evident that Pluto's orbit intersects Neptune's orbit. Pluto and Neptune are in a 2:3 resonance condition, which means that 3x Neptune orbits the sun according to 2x that Pluto does. In this condition, when Pluto reaches its closest point with the sun (perihelion) and in one of two nodal point orbit in orbit, Neptune is always far from Pluto's position. This resonance condition also guarantees that Pluto and Neptune will not collide even though their orbits intersect.

Meanwhile, resonance that results in orbital instability is characterized by the celestial object's lack of existence or emptiness at a certain distance from the sun due to repeated strong gravitational disturbances in that place. For examples of this case can be found in the asteroid belt, the area between the orbits of Mars and Jupiter which stretches from 2.1 astronomical units to 3.3 astronomical units (1 astronomical unit is defined as the average distance of the Earth from the Sun, which is equal to $1.496 \times 10^8 km$), in the form of relatively empty areas of asteroids known as Kirkwood gaps as shown in Figure 5. The gaps result from MMR conditions between planet Jupiter and asteroids in the asteroid belt region, namely 3:1, 5:2, 7:3, and 2:1 MMR. Even if the presence of an asteroid is found in locations that experience those MMR, the existence of objects in that place is only temporary. These objects will experience an increase in orbit eccentricity (e) due to gravitational interactions with Jupiter, making these asteroids' orbits more elongated. Significant changes in eccentricity make the asteroids approach inner planets (especially Mars), which by gravitational pull or collisions, takes off the asteroids from MMR zones generating the observed gaps.

A condition of 1:1 MMR can be experienced by a small body such as an asteroid with a particular planet. The asteroid can only be in an MMR 1:1 condition if $|a - a_p| < (\mu/3)^{1/3} a_p$, where μ is the planet-sun mass ratio, and a and ap are the average distance of asteroid and planet from the sun, respectively. In case of the asteroid's eccentricity is very small, i.e. e $(\mu/3)^{1/3}$, then the object can have an orbit very similar to that of the planet. This kind of asteroid remains orbiting the Sun as a central object, but in the same orbit as the planet. Asteroids in this condition are not disturbed by the planet because they occupy a special location called the Lagrange points (L4 (in front of the planet) and L5 (behind the planet)). This special locations mark where the gravitational pull of two

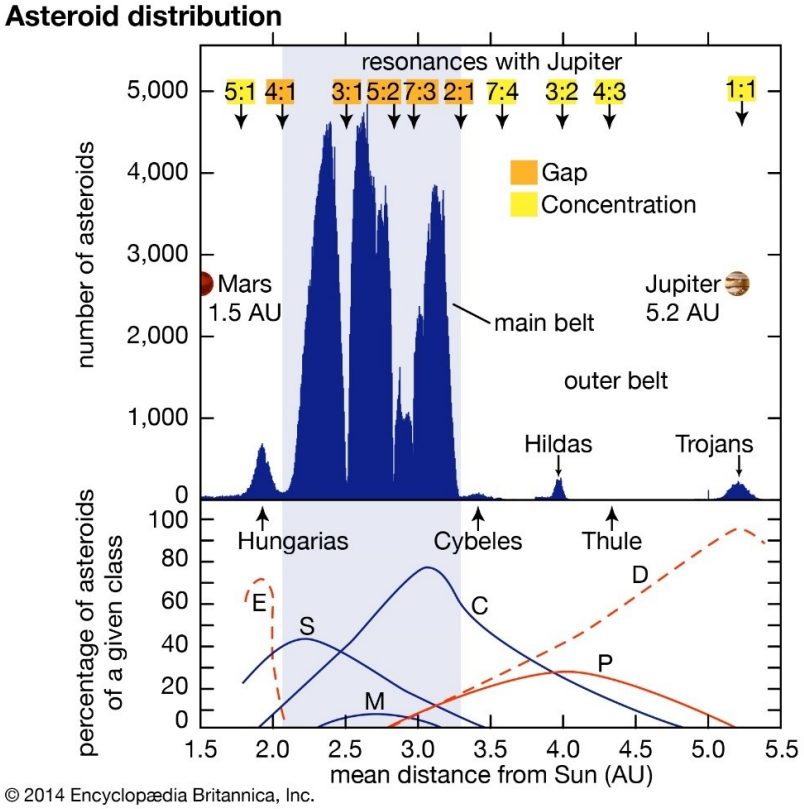

Figure 5: Kirkwood Gap in the Asteroid Belt that Marks the Location of the Orbit that is Unstable

large masses (the sun and planet) precisely equals the centripetal force required for a small object (asteroid) to move with them. Otherwise, if $e > (\mu/3)^{1/3}$, the asteroids will have different orbits from the planet. In this work, the conditions of 1:1 MMR are studied irrespective of the magnitude of the eccentricity of the asteroids.

Detection of asteroids in a 1:1 MMR state with one of the terrestrial planets (Mercury to Mars) through numerical computing is interesting. These asteroids have an orbital period almost the same as the planet orbits the sun (also known as co-orbital asteroids). Employing the numerical method of the N-body problem in celestial mechanics, we can also predict the steady-state population of these objects.

3.2. Data collection

The data used in our research experiment shown in Table 1. Due to space limitations we only show the first 10 rows of ten asteroids from 1000 near-earth asteroids as our sample. The data was generated by numerical computation using full model of the solar system (the sun, 8 planets, and the moon as massive bodies) with the use of Swift-RMVS (Regularized Mixed-Variable Symplectic) integrator [12]. In this data each particle (or asteroid) is sampled every 1000 years. The number of columns in this data is 8, i.e. ParticleID, $t, a, e, i, \Omega, \omega$, and M (representing asteroid-id, time, semimajor axis, eccentricity, inclination, longitude of ascending node, argument of perihelion, and mean anomaly). The information used in our study is only the first 3 columns, namely ParticleID, t, and a, because we focus on finding the location of resonance and time duration for that event.

Table 1: Examples of data used in experiments

ParticleID	t (1000 yr)	a (au)	e	i (radian)	Ω(radian)	ω(radian)	M (radian)
1	0	1.061678	0.051489	0.022113	4.421005	5.529237	2.02073
2	0	1.942493	0.555555	0.079524	4.411401	4.030324	3.01016
3	0	0.830664	0.388456	0.088272	2.900785	5.581375	4.84435
4	0	0.988711	0.013889	0.075412	4.614986	1.527315	3.08849
5	0	2.085440	0.606626	0.032093	4.734172	1.805200	0.44300
6	0	0.996808	0.012064	0.144113	0.434161	3.888507	3.87216
7	0	1.210198	0.177278	0.134907	2.571753	0.872216	5.25321
8	0	1.007883	0.139036	0.038552	3.671272	5.048894	5.85180
9	0	1.306163	0.246871	0.075872	5.275945	5.695753	0.27101
10	0	0.992638	0.083851	0.013485	1.455619	2.622877	4.49503

3.3. Experimental scenario

In this experiment, we performed a scenario on the first 10 particles in a dataset with time series lengths that were different for each particle. In this scenario, we use 5 input parameters to process data, i.e. sax_cut, l_find, mismatch, n_try, threshold, and 1:1 threshold. For the values and explanations for each parameter we use the following values: (a) sax_cuts = 20. The sax_cut parameter functions as a parameter enter the number of alphabets used in the SAX algorithm. The sax_cut parameter is filled with number 20 because we take the most number of alphabets that can be specified in SAX; (b) l_find = 10. The l_find parameter serves to determine the minimum length of the motif sought in the random projection algorithm. This parameter contains a value of 10 because we call a resonance in the asteroid orbital data if the particle (i.e. the asteroid) has a minimum length of 10,000 years or 10 units of time in the time series; (c) mismatch = 1. The mismatch parameter to determine the minimum amount of mismatch that occurs in the motive in the random projection algorithm; (d) n_try = 1. The parameter n_try is to determine the number of experiments or iterations that will be performed to run the random projection algorithm; (e) threshold = 2. The threshold parameter to determine the minimum number of similar motives found so that the collection of motifs is called resonance by the user. We give a value of 2 because it takes the minimum amount on the bucket to be called a motive, and (f) threshold_planet = 0.05. The last parameter is tolerance value for asteroid's semimajor axis to planet's semimajor axis as the maximum and minimum limits of resonance 1:1 to be recognized to occur.

4. Results and discussion

Based on the scenario designed in the previous section, Table 2 shows the experiment results with the scenario. There are 4 columns, namely ParticleID, Particle Length, Time, and Results. The ParticleID column is the processed asteroid. The Particle Length column is the particle's length of time series. Time column is the running time by the system from input data to output in seconds. The Results column is the result of the system output.

In the Results column, there is an output format from the results of the system processing. If the system cannot find a 1:1 resonance, the printed column results in "no resonance". If the system finds a motif of resonance 1:1 with a particular planet, then the Results column will be printed in the form: "a = x, 1:1 planet's name, a - b". The first part is the distance from the sun where the resonance 1:1 occurs, the second part is the name of the planet in resonance with the asteroid, and

the last part is the time for resonance to occur and hence the duration for resonance in the unit of 1000 years.

Based on the experiments carried out, the system can recognize motif in input data and filter 1:1 resonance with the certain planet. The average execution time for ten asteroids is 23.605 seconds. Execution time is the time taken from starting to read input data up to produce output. This execution time is influenced by the data length of the asteroid being processed; the longer time-series data, the longer time is needed.

Table 2: Experimental results

ParticleID	Particle Length	Time	Results
1	181	19.735	There is no resonance 1:1
2	505	22.616	There is no resonance 1:1
3	624	23.883	a = 0.91, 1:1 Moon, 63 - 73
			a = 0.98, 1:1 Earth, 69 - 125
			a = 1.52, 1:1 Mars, 212 - 226
			a = 1.55, 1:1 Mars, 219 - 228
			a = 1.55, 1:1 Mars, 231 - 240
			a = 1.56, 1:1 Mars, 220 - 229
			a = 1.56, 1:1 Mars, 221 - 230
			a = 1.56, 1:1 Mars, 233 - 242
4	305	21.393	There is no resonance 1:1
5	378	21.191	There is no resonance 1:1
6	360	22.228	a = 1.51, 1:1 Mars, 71 - 82
			a = 1.54, 1:1 Mars, 76 - 86
7	1140	26.231	There is no resonance 1:1
8	499	21.693	There is no resonance 1:1
9	908	24.759	a = 1.52, 1:1 Mars, 35 - 45
			a = 1.55, 1:1 Mars, 38 - 58
10	2307	32.319	There is no resonance 1:1

After we experimented, we made a comparison between the system we built and the SwiftVis application. SwiftVis is a data analysis and visualization package written in Java, capable of mean motion resonance searching. To get the output from the SwiftVis application, users must input data, add selection functions, choose the filter and general plot functions to obtain any particles in 1:1 resonance with a particular planet. Table 3 illustrates the comparison of the outputs from the experiments we did and the output from the SwiftVis application with the same threshold_planet parameter, which is 0.05. In the column SwiftVis application output, if there are particles in a resonance state with a particular planet, the column delivers the information concerning the result. In the first part, there is a description of which planet the particle in 1:1 resonance. In the second part, the particle's time range (in 1000 years unit) experienced a 1: 1 resonance.

Based on Table 3, the system we built has an average accuracy of 88.3% of the ten particles. The minimum length of resonance requirement is based on input l_find in the scenario, i.e., 10. Accuracy is obtained from the similarity on the duration for 1:1 resonance to occur between our model and SwiftVis. To calculate the accuracy percentage, all the resonance time ranges of the experimental results that have intersection are combined first and then compared to match the results calculated by SwiftVis. For example, the particle with id 6 experienced two times 1:1 resonances with Mars in the time range 71 - 82 and 76 - 86. The two-time ranges have intersections from 76 to 82, then they

will be combined into a new time span of 71-86. Hence the duration is compared to SwiftVis output which has the same time ranges. Then the accuracy obtained for this particle is 100%.

There is tolerance in calculating accuracy. If the program produces an output whose results are not present in SwiftVis output, the results will be ignored and will not affect the accuracy percentage. For example, the particle with id 3 results in a 1:1 resonance with the moon, but at the SwiftVis output, the particle does not experience 1:1 resonance with the moon. Tolerance is also applied if the time span generated by the program exceeds the one generated by SwiftVis. At the experimental output, the particle with id 3 has a 1:1 resonance with the earth in the time range 69 – 125. According to SwiftVis, this particle has a range of 94 - 122. Then the experimental output is considered suitable because it includes the results of SwiftVis output even though the time span exceeds the time range of SwiftVis.

Table 3: Results comparison

Particle ID	System experimental results	SwiftVis application output	Accuracy
1	no 1:1 resonance	no 1:1 resonance	100%
2	no 1:1 resonance	no 1:1 resonance	100%
3	a = 0.91, 1:1 moon, 63 - 73 a = 0.98, 1:1 earth, 69 - 125 a = 1.52, 1:1 mars, 212 - 226 a = 1.55, 1:1 mars, 219 - 228 a = 1.55, 1:1 mars, 231 - 240 a = 1.56, 1:1 mars, 220 - 229 a = 1.56, 1:1 mars, 232 - 241 a = 1.56, 1:1 mars, 221 - 230 a = 1.56, 1:1 mars, 233 - 242	1:1 earth, 94 - 122 1:1 mars, 204 - 238	88%
4	no 1:1 resonance	no 1:1 resonance	100%
5	no 1:1 resonance	no 1:1 resonance	100%
6	a = 1.51, 1:1 mars, 71 - 82 a = 1.54, 1:1 mars, 71 - 86	1:1 mars, 76 - 86	100%
7	no 1:1 resonance	1:1 mars, 211 - 234	0%
8	no 1:1 resonance	no 1:1 resonance	100%
9	a = 1.52, 1:1 mars, 35 - 45 a = 1.55, 1:1 mars, 38 - 58	1:1 mars, 34 - 57	95%
10	no 1:1 resonance	no 1:1 resonance	100%

5. Conclusion

The main contribution of this research is (i) to provide computational models for the discovery of time series motifs in the case of mean motion resonance. This model contains several stages, such as pre-processing data models, Z normalization, SAX, random projection, and post-processing data; (ii) to conduct several experiments in the case of mean motion resonance. Based on the obtained results, we can state that the proposed model can be used to obtain motif of location, how many times motif occurs, and with which planet a 1:1 mean motion resonance exists.

In the future, we plan to improve the accuracy and speed of the system being built. Big data platforms [30, 18, 6] and Apache Spark [32, 33] also machine learning algorithms [10, 13, 29] can be applied to increase the speed of large-scale data processing.

Acknowledgement

Authors would like to acknowledge the Ministry of Research and Technology/National Research and Innovation Agency for funding this work through a research grant of 281/UN40.LP/PT.01.03/2021.

References

[1] H. Abe and T. Yamaguchi, *Implementing an integrated time-series data mining environment-a case study of medical kdd on chronic hepatitis*, 1st Int. Conf. complex Med. Eng., (2005) 1–4, 2005.
[2] I. Androulakis, J. Wu, J. Vitolo, and C. Roth, *Selecting maximally informative genes to enable temporal expression profiling analysis*, in Proc. of Foundations of Systems Biology in Engineering, (2005) 23–26.
[3] C. P. Asmoro, L. S. Riza, N. D. Ardi, and Y. R. Tayubi, *Analysis of meteorological parameters wind speed, temperature, and pressure profiles during Tropical Cyclone Cempaka Dahlia 2017 using time series analysis*, In Journal of Physics: Conference Series , 1280 (2) (2019) 022076.
[4] J. Buhler and M. Tompa, *Finding motifs using random projections*, J. Comput. Biol., 9 (2) (2002) 225–242.
[5] B. Celly and V. Zordan, *Animated people textures*, in 17th International Conference on Computer Animation and Social Agents (CASA), (2004) 1–8.
[6] R. K. Chawda and G. Thakur, *Big data and advanced analytics tools*, in 2016 Symposium on Colossal Data Analysis and Networking (CDAN 2016), (2016) 1-8.
[7] E. Forgács-Dajka, Z. Sándor, and B. Érdi, *A fast method to identify mean motion resonances*, Mon. Not. R. Astron. Soc., 2018, 477 (3) (2018) 3383–3389, doi: 10.1093/mnras/sty641.
[8] T. Guyet, C. Garbay, and M. Dojat, *Knowledge construction from time series data using a collaborative exploration system*, J. Biomed. Inform., 40 (2) (2007) 672–687, doi: 10.1016/j.jbi.2007.09.006.
[9] N. C. Jones, P. A. Pevzner, and P. Pevzner, An *introduction to bioinformatics algorithms*,MIT press, 2004.
[10] M. I. Jordan and T. M. Mitchell, *Machine learning: trends, perspectives, and prospects*, Science, 349 (6245) (2015) 255-260.
[11] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," Int. J. Rob. Res., 34(2015) 705–724, doi: 10.1177/0278364914549607.
[12] H.F. Levison and M.J. Duncan, *The long-term dynamical behavior of short-period comets*, Icarus, 1994, 108 (1) (1994) 18–36,doi: 10.1006/icar.1994.1039.
[13] T. Li, H. Shen, Q. Yuan, X. Zhang, and L. Zhang, *Estimating ground-level pm2.5 by fusing satellite and station observations: a geo-intelligent deep learning approach*, Geophys. Res. Lett., 44 (23) (2017) 11-985.
[14] J. Lin, E. Keogh, L. Wei, and S. Lonardi, *Experiencing sax: a novel symbolic representation of time series*, Data Min. Knowl. Discov., 15 (2) (2007)107-144.
[15] J. Lin, E. Keogh, S. Lonardi, and P. Patel, *Finding motifs in time series*, Proc. 2nd Work. Temporal Data Min., (2002) 53–68.
[16] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, *A symbolic representation of time series, with implications for streaming algorithms*, in Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, DMKD, (2003) 2-11.
[17] B. Liu, J. Li, C. Chen, W. Tan, Q. Chen, and M. Zhou, *Efficient motif discovery for large-scale time series in healthcare*, IEEE Trans. Ind. Informatics, (2015) 583–590, doi: 10.1109/TII.2015.2411226.
[18] S. Londhe and S. Mahajan, *Effective and efficient way of reduce dependency on dataset with the help of mapreduce on big data*, Int. J. Students' Res. Technol. Manag., 15 (1) (2015) 401-405.
[19] H. M. Martinez, *An efficient method for finding repeats in molecular sequences*, Nucleic Acids Res., 11 (13) (1983) 4629-4634.
[20] A. Mueen and E. Keogh, *Online discovery and maintenance of time series motifs*, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2010) 1089–1098, doi: 10.1145/1835804.1835941.
[21] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. Westover, *Exact discovery of time series motifs*, in Society for Industrial and Applied Mathematics - 9th SIAM International Conference on Data Mining 2009, Proceedings in Applied Mathematics, (2009) 473–484, doi: 10.1137/1.9781611972795.41.
[22] S. Mullainathan and J. Spiess, *Machine learning: an applied econometric approach*, J. Econ. Perspect., 31 (2) (2017) 87–106.
[23] K. Pariwatthanasak and C. A. Ratanamahatana, *Time series motif discovery using approximated matrix profile*, in Advances in Intelligent Systems and Computing, (2019) 707–716, doi: 10.1007/978-981-13-1165-9_64.

[24] P. A. Pevzner and S. H. Sze, *Combinatorial approaches to finding subtle signals in DNA sequences*, Proc. Int. Conf. Intell. Syst. Mol. Biol., 8 (2002) 269-278.

[25] E. Pilat-Lohinger, Á. Süli, P. Robutel, and F. Freistetter, *The influence of giant planets near a mean motion resonance on earth-like planets in the habitable zone of sun-like stars*, Astrophys. J., 681 (2008) 1639–1645, 2008, doi: 10.1086/587501.

[26] L. S. Riza, J. A. Utama, S. M. Putra, F. M. Simatupang, and E. P. Nugroho, "Parallel exponential smoothing using the bootstrap method in r for forecasting asteroid's orbital elements," Pertanika J. Sci. Technol., 26 (1) (2018) 441–462.

[27] L. S. Riza, Y. Wihardi, E. A. Nurdin, N. D. Ardi, C. P. Asmoro, A. F. C. Wijaya and A. B. D. Nandiyanto, *Analysis on atmospheric pressure, temperature, and wind speed profiles during total solar eclipse 9 March 2016 using time series clustering*, In Journal of Physics: Conference Series, 771 (1) (2016) 012009.

[28] L. S. Riza, T. F. Dhiba, W. Setiawan, T. Hidayat, and M. Fahsi, *Parallel random projection using R high performance computing for planted motif search*, Telkomnika, 17 (3) (2019) 1352-1359.

[29] L. S. Riza, F. D. Pratama, E. Piantari, and M. Fashi, *Genomic repeats detection using Boyer-Moore algorithm on Apache Spark Streaming*, Telkomnika,18 (2) (2020) 783-791.

[30] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, *The Hadoop distributed file system*, in 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST2010), (2010) 1-10.

[31] P. Wiegert, M. Connors, and C. Veillet, *A retrograde co-orbital asteroid of Jupiter*, Nature, 543 (2017) 687–689, doi: 10.1038/nature22029.

[32] M. Zaharia, R.S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, *Apache spark: A unified engine for big data processing*, Commun. ACM, 59 (11) (2016) 56–65, doi: 10.1145/2934664.

[33] S. M. Zobaed and M. A. Salehi, *Big Data in the Cloud*, in Encyclopedia of Big Data Technologies, 47 (2019) 98-115.

[34] Y. Zhu, A. Mueen, and E. Keogh, *Matrix Profile IX: Admissible Time Series Motif Discovery with Missing Data*, IEEE Trans. Knowl. Data Eng., (2019) 2616–2626, doi: 10.1109/TKDE.2019.2950623.

[35] Y. Zhu, Z. Zimmerman, N.S. Senobari, C.C.M. Yeh, G. Funning, A. Mueen, P. Brisk, and E. Keogh, *Matrix profile II: exploiting a novel algorithm and GPUs to break the one hundred million barrier for time series motifs and joins*, in Proceedings - IEEE International Conference on Data Mining (ICDM), (2017) xxx–xxx, doi: 10.1109/ICDM.2016.126.