Elastic Data Binning for Transient Pattern Analysis in Time-Domain Astrophysics

Thanapol Phungtua-eng
Shizuoka University
Shizuoka, Japan
thanapol@yy-lab.info

Yoshitaka Yamamoto
Shizuoka University
Shizuoka, Japan
yyamamoto@inf.shizuoka.ac.jp

Shigeyuki Sako
University of Tokyo
Tokyo, Japan
sako@ioa.s.u-tokyo.ac.jp

ABSTRACT

Time-domain astrophysics analysis (TDAA) involves observational surveys of celestial phenomena that may contain irrelevant information because of several factors, one of which is the sensitivity of the optical telescopes. Data binning is a typical technique for removing inconsistencies and clarifying the main characteristic of the original data in astrophysics analysis. It splits the data sequence into smaller bins with a fixed size and subsequently sketches them into a new representation form. In this study, we introduce a novel approach, called elastic data binning (EBinning), to automatically adjust each bin size using two statistical metrics based on the Student's t-test for linear regression and Hoeffding inequality. We demonstrate the successful representation of various characteristics in the light curve data gathered from the Kiso Schmidt telescope using our approach and the applicability of our approach for transient pattern analysis using real world data.

CCS CONCEPTS

• **Applied computing** → *Astronomy*; **Astronomy**; • **Mathematics of computing** → Time series analysis;

KEYWORDS

Data binning, Data sketching, Hoeffding inequality, Student's t-test, Lightcurve

ACM Reference Format:
Thanapol Phungtua-eng, Yoshitaka Yamamoto, and Shigeyuki Sako. 2023. Elastic Data Binning for Transient Pattern Analysis in Time-Domain Astrophysics. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), March 27-March 31, 2023, Tallinn, Estonia.* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3555776.3577606

1 INTRODUCTION

Large-scale surveys of celestial objects and phenomena provide fundamental measurements in the search for astronomical discoveries. A highly sensitive optical telescope measures the light intensity of astronomical objects within a region over a certain time period in video format. The video data is subsequently converted into one-dimensional (1D) time series data by performing aperture photometry for each timestamp and astronomical object. The generated time series data are often known as *lightcurve* (LC) [24].

LCs are critical for modern research on *time-domain astrophysics analysis* (TDAA) that scopes temporary phenomena with short timescales from minute to second scales, because these phenomena can be identified as *transient* patterns appearing in LCs.

The characteristics of a transient pattern in our domain include the following three phases: a sudden and intensive occurrence from prior observations, an observational peak, and a return to normal conditions, such as gamma-ray bursts, flares, and outbursts [2]. In real-time analysis, it is often difficult for astronomers to aware the occurrence of a significant event because a wavelength's timespan ranges from a second or minutes. Hence, the analysis of *unaware transient* patterns contributes to astronomers' survey.

The central goal of transient pattern analysis is to seek or capture unexpected behavior by comparing it with prior evidence from some representative periods [1, 8, 12, 16]. However, LCs contain unwanted external factors, such as atmospheric turbulence and hardware measurement errors, which may lead to erroneous analytical conclusions [10].

To elucidate on the issue of external factors in LCs, we illustrate a typical LC from a specific celestial object in Figure 1 [1]. The y-axis measures the celestial object's brightness, and the x-axis provides the timestamps in modified Julian date units. The red line that exhibits minor changes is highlighted in the figure. It is difficult to judge whether these changes are reflected by a transient pattern or noise. Notably, the astronomers were judged to be real transients [2]. To restrict the influence of noise, the LCs are preprocessed.

Data binning is a typical technique that eliminates noise and subsequently extracts relevant characteristics of transient patterns [7]. Data binning makes a sketch of the time-series using bins, each of which stores statistical summary measurements, such as the mean, variance, and slope, for a specific corresponding segment. Notably, *bin size* denotes the number of samples to be summarized in one bin. The user must set this value according to the domain knowledge. Thus, data binning is explored in a parametric manner.

The bin size determines the distortion from the original data; if it is too large, we may lose its essential characteristics (that is, transient patterns). Conversely, if it is too small, we may face quality-degrading noise, which makes detecting transient patterns challenging. However, it is infeasible for users to set an appropriate bin size to accommodate various scenarios that manifest in large-scale surveys with LCs. Figure 2 illustrates the sample sketching, with the blue line representing the differences in bin size.

[1]The real-world dataset and videos from a survey in this paper were provided by M. Aizawa and K. Kashiyama. Interested readers can view the [2] for more details.

SAC '23, March 27-March 31, 2023, Tallinn, Estonia

ACM ISBN 978-1-4503-9517-5/23/03…$15.00
https://doi.org/10.1145/3555776.3577606

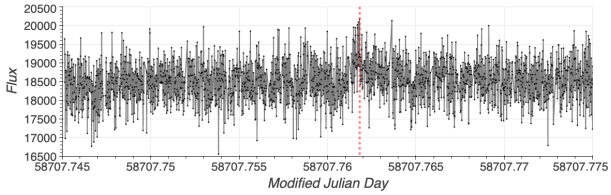
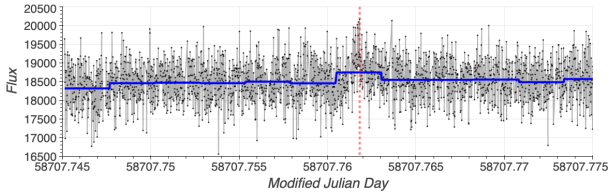
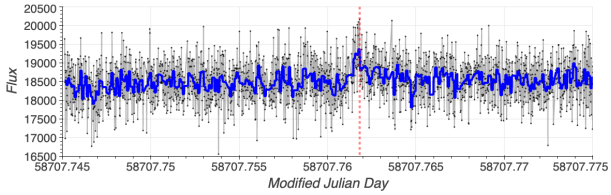


Figure 1: LC with real transient pattern occurring [2]



(a) Sample sketching by large bin size



(b) Sample sketching by small bin size

Figure 2: Influence by varying the bin size in Figure 1

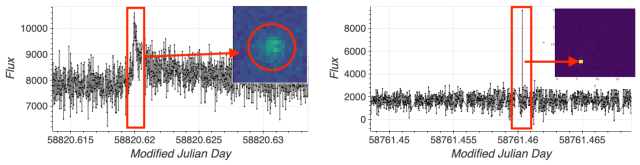
This study aims to address the trade-off problem of data binning between the sketching quality and quality-degrading noise, caused by the bin size. In this study, we propose *elastic* data binning (**EBinning**) that adjusts the bin size in a non-parametric manner. The observation is that this period with no signal can be zero-suppressed; only the characteristic periods are captured using bins. We achieve such an *auto-focus* function that adequately varies each bin's size. We demonstrate that the proposed method is helpful for TDAA, such as transient pattern detection using the real dataset.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work. Section 3 details the notation and terminologies used in the paper. Section 4 describes the proposed algorithm for EBinning. Section 5 presents information regarding the LCs analyzed in this study, and Section 6 presents our experimental results. Finally, Section 7 concludes the paper and suggests future research directions.

2 RELATED WORK

2.1 Outlier vs. transient pattern

The concepts of outliers and transient patterns are similar in that both describe observed data significantly deviating from normal behavior. The difference lies in the length of the observed data [3]. According to Blázquez-García et al. [3], an outlier is specified as an observed point x_t such that $|x_t - E[x_t]| > \tau$, where $E[x_t]$ denotes the expected value and τ denotes a threshold. Conversely, a transient pattern is defined as a sequence or representation of points X such that $d(X, E[X]) > \tau$, where $E[X]$ denotes the expectation



(a) Sample transient pattern **(b) Sample outlier**

Figure 3: Transient pattern vs. outlier [2]

of normality and d denotes the dissimilarity metric between two sequences. Figure 3 shows two examples of LCs. This study focuses on transient pattern analysis towards scientific discovery (e.g., the transient pattern in Figure 3a represents the flare phenomenon). Notably, outliers in LCs typically occur as measurement noises (e.g., the outlier in Figure 3b represents the noise from a video pixel).

2.2 Time series representation and data binning

There are numerous studies on representing the time-series data [5, 6, 13, 15, 21]. Their common issue arises from the trade-off relation between the resolution and quality to capture the key characteristics of original data.

Traditional data binning divides the original time-series into segments and thereafter summarizes them with bins. Every bin has the same size, which is fixed in advance, and stores the mean value among the corresponding segment. This technique is widely used for improving the signal-to-noise ratio of LCs [7]. It is alternatively known as *Piecewise Aggregate Approximation* (PAA) [11].

Symbolic Aggregate approXimation (SAX) [13] is well-known as a symbolic time-series representation based on PAA and widely applied in diverse domains. There is a recent work to add information about slope values for symbolic representation, known as *1D-SAX* [15]. However, these PAA-based techniques require certain key parameters, such as the bin and symbol sizes, which affect the resolution and quality of the data representation. It is necessary to determine their relevant values for sketching the original data, provided that the transient pattern characteristics are preserved.

There has been a proposed metric to determine them, known as *compression ratio* [4], which is defined by the description length (resolution) and the variance loss (quality) caused by sketching. The *Temporal Window In Network* (TWIN) [23] is a non-parametric framework based on the compression ratio to fix a relevant description length of the time-series data. In the context of data binning, the description length corresponds to the total number of bins. Thus, the number of bins can be determined using the metrics, while it remains unclear how the size of each bin can also be adjusted in accordance with the condition of the target time-series data.

2.3 Transient pattern analysis in TDAA

Rebbapragada et al. proposed an unsupervised anomaly detection method, known as PCAD, for LCs with three types of transient patterns [20]. This method is a modification of the k-means clustering. The LCs used in their study were filtered to eliminate noises, and their objective was to cluster the types of transient patterns in LCs. Conversely, we intend to sketch the LCs by data binning and thereafter detect transient patterns from the obtained sketch.

The use of data binning for Lunar laser ranging data was proposed by Shevlyakov and Kan in the *SK-method* (abbreviated from the authors' names) [22]. They seek suddenly changing points in the time-series based on the Chebyshev inequality. Notably, data binning in the SK-method belongs to the traditional scheme in which the bin sizes are equal and fixed in advance.

A transient pattern detection for LCs was proposed in a recent study by Phungtua-eng et al. [17, 18]. Their proposed method, known as PYS, introduces the bin-merge trick to incrementally merge two similar bins. The PYS addresses the issue of fixing the bin size while it uses a specific measurement based on the t-test score and treats only the mean value for capturing the statistic feature of each bin. In this paper, we introduce the bin-merge algorithm by generalizing the notion of bins and using the concentration equality, thereby addressing the limitation of the PYS.

Notably, the PYS was designed for transient pattern detection. It is not for sketching the time series. Our primary scope is to provide novel data binning that identifies the auto-focusing function for adjusting each bin size.

Adaptive-binning for TDAA with LCs was proposed by Lott et al [14], which was used to explore blazar events with LCs of Fermi-Large area telescopes. Adaptive-binning produced an appropriate bin size for each bin. Unfortunately, the main contribution of Adaptive-binning is based on the correlation between the source photon spectral index and the gamma-ray flux for adjusting the bin size. Our objective is to address unknown phenomena; therefore, we may not assume whether they correlate. Our objective is different from that of Adaptive-binning.

Matrix profile (MP) is a well-known technique applied in motif discovery, anomaly detection, and segmentation in time series data mining [9, 26]. MP provides a sufficient data structure that includes two primary components: a distance profile and a profile index for each subsequence. This algorithm has only one user-specified parameter, that is, the subsequence length. In this study, we demonstrate the application of MP to the task of transient pattern detection for LCs and subsequently compare the performance with our proposed method based on elastic data binning.

3 NOTATION AND TERMINOLOGIES

3.1 Preliminaries

The LC is defined as $X_{1,t} = [x_1, x_2, \ldots, x_t]$, where t is the prior timestamp and each x_i ($1 \leq i \leq t$) is a measurement of light intensity on the target astronomical object at timestamp i. The subsequence of LC from the timestamp u to the timestamp v is described as $X_{u,v} = [x_u, \ldots, x_v]$ ($1 \leq u \leq v \leq t$).

The LC is alternatively described as $[X_{u_1,v_1}, \ldots, X_{u_p,v_p}, \ldots, X_{u_w,v_w}]$, where w is the number of subsequences and each u_p and v_p ($1 \leq p \leq w$) denote the starting and ending timestamps of the pth subsequence. Accordingly, it holds that $u_1 = 1$, $v_w = t$ and $u_{p+1} = v_p + 1$. Each subsequence is summarized as its statistical features (e.g., mean, slope) value and then stored into a tuple, called a *bin*.

Definition 1. *bin*: Given a subsequence $X_{u,v}$, the *bin* w.r.t. $X_{u,v}$ is defined as the tuple (n, f, st), where n is the length of $X_{u,v}$ (i.e., $n = u - v + 1$), f is composed of the statistical features on $X_{u,v}$, and st is composed of the auxiliary variables used for updating f. f (and st) is different in every data binning algorithm. Their detailed definition are described later. n is called the *bin size*. We often denote by bin_p the bin w.r.t. X_{u_p,v_p}.

Definition 2. *window*: The sequence of bins obtained from $X_{1,t}$ is maintained in the *window* (W), i.e., $W = [bin_1, bin_2, \ldots, bin_w]$. w denotes the window size, where $1 \leq w \leq t$. Generally, the value of w is bounded by the memory resource limitation.

3.2 Properties of target transient patterns

We focus the research interest on seeking for *unknown* transient patterns. We must find them without any assumption when they will occur and what kind of features (peaks, decays, and shapes) they will have.

Let $X_{u,v}$ be a transient pattern to be the target. Then, the task of a data binning algorithm is to compactly and concisely capture $X_{u,v}$. If we deal with unknown transient patterns, there is no information about the values u and v of $X_{u,v}$. This condition makes it difficult to fix the bin size n in advance (as shown in Figure 2). To capture $X_{u,v}$ using small bins (ideally one bin), it is necessary to adjust each bin size in accordance with the similarity of neighboring bins. This motivates the bin-merge trick to auto-adjust bins presented in our proposed algorithm. Good sketching provides information that can be easily and rapidly used to analyze unknown transient patterns.

4 ELASTIC DATA BINNING (EBINNING)

The EBinning automatically specifies the appropriate bin size (n) for each bin in a window (W). It considers every two neighboring bins in W and measures the ability to combine the two bins, which is referred to the *mergeability* score (k).

For example, if k is a low value, it indicates that two neighboring bins have similar features, and we can merge them into a larger bin. Conversely, if k is a high value, it indicates that two neighboring bins exhibit different features, and the corresponding period may be associated with changes in behavior. In this case, two neighboring bins cannot be merged because they may distort characteristic changes. EBinning dynamically adjusts the bin size according to the various characteristics of original data.

4.1 Baseline of EBinning

EBinning is used to measure the mergeability between two bins and to determine if the two bins should be merged. We present the baseline of the proposed method in Algorithm 1.

The time series LC, initial bin size n, and window size w are given as input. Note that n is set to a small value so that every initial bins belongs to a single distribution. In addition, we defined w using TWIN [23]. TWIN searches w with a good balance between compression ratio and variance.

First, we create an empty buffer for initialization, window (W), and *ScoreProfile*. Then, x_i is read and stored in the buffer, where the buffer size does not exceed the initial bin size. After the buffer is full, we summarize the sub-sequence of the buffer into a *bin*. This process is called initialization (Line 5).

After initialization, this bin is appended into W. We then compute the mergeability score (k) with its neighboring bin, that is, $bin_{|W|-1}$ (Line 6). Once W is full, we search the *ScoreProfile* for the index p whose bin has the minimum k value (Line 8) and merge bin_p and bin_{p-1} into a new bin (Line 9).

When two neighboring bins are merged, we update only two k values of neighboring bins for the merged bin (Line 11).

Algorithm 1: Baseline EBinning

Input: $X_{1,t}$, n (initialize bin size), w (window size)
Output: $W = \{bin_1, bin_2, \ldots, bin_w\}$

```
1  W ← ∅ , ScoreProfile← ∅, Buffer ← ∅;
2  for i ← 1 to t do
3      Buffer ← Append x_i ;
4      if (Buffer is full) then
5          W ← Append Initialization(Buffer) ;
6          ScoreProfile ← Append
             ComputeScore(bin_|W|, bin_|W|−1) ;
7          if (W is full) then
8              p ← FindIndexOfMinK(ScoreProfile) ;
9              bin_p−1 ← Merge(bin_p, bin_p−1) ;
10             Pop(bin_p);
11             Update(ScoreProfile);
```

4.2 Mean-EBinning (M-EBinning)

The mergeability score is determined in accordance with the statistical test to distinguish two bins. It is based on the concentration equality. In a normal situation without any occurrence of events (i.e., noises or transients), each x_t is randomly and independently generated from a specific distribution. Hence, if two bins are generated from the same distribution, we can guarantee a bounded range of their means. We call this approach *Mean-EBinning (M-EBining)* .

4.2.1 Initialization. bin_p of M-EBinning is given in the form of $\langle n_p, f, st\rangle$ where f is μ (mean) and st is empty. Let bin_p and μ_p be the bin and mean of X_{u_p,v_p}, respectively. Then, the expected value of X_{u_p,v_p} can deviate from μ_p within the boundary (ε). Let δ be an error probability that the expected value does not fall into the range $[\mu_p - \varepsilon, \mu_p + \varepsilon]$. By using the Hoeffding's inequality, ε can be described with respect to δ as follows:

$$\varepsilon_p = \sqrt{\frac{c^2}{2n_p} log \frac{2}{\delta}}, \tag{1}$$

where c is the difference between the min and max values of X_{u_p,v_p}, and n_p is the bin size of bin_p. Because c and δ are regarded as constants, ϵ_p is determined by n_p. Accordingly, we can capture the expected value of bin_p by using μ_p and n_p.

4.2.2 Score computation. Let bin_p and bin_{p-1} be two neighboring bins. Assume that they belong to the same distribution whose expected value is e. Under this assumption, the Hoeffding inequality states that e belongs to both $[\mu_p - \epsilon_p, \mu_p + \epsilon_p]$ and $[\mu_{p-1} - \epsilon_{p-1}, \mu_{p-1} + \epsilon_{p-1}]$ hold (See Figure 4).

Based on this observation, we use the following k value to judge whether the bin_p and bin_{p-1} are mergeable or not:

$$k = \frac{|\mu_p - \mu_{p-1}|}{min(\varepsilon_p, \varepsilon_{p-1})} \tag{2}$$

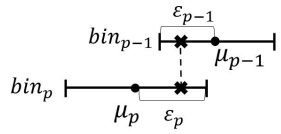


Figure 4: Two boundaries in the same distribution

4.2.3 Merging function. We merge bin_{p-1} and bin_p into $bin_{new} = \langle n_{new}, \mu_{new}\rangle$ using the following equation:

$$n_{new} = n_p + n_{p-1} \tag{3a}$$

$$\mu_{new} = (n_p\mu_p + n_{p-1}\mu_{p-1})/(n_p + n_{p-1}) \tag{3b}$$

4.3 Linear-EBinning (L-EBinning)

The mean value is sometimes inappropriate for representing a gradual change in time series. Therefore, we consider using linear regression to represent these trends in the original sequence. The second approach represents each bin based on linear representation. Hereafter, we call it *Linear-EBinning (L-EBinning)*.

4.3.1 Initialization. In the initialization step, we summarize the buffer storing $X_{u,v}$ into a bin of form $\langle n, f, st\rangle$. Now, f is defined as the tuple (α, β), where α and β are the intercept and slope of linear regression, respectively. In turn, st is defined as the tuple (μ, σ^2, $\sum_{t=u}^{v} t$, $\sum_{t=u}^{v} t^2$, $\sum_{t=u}^{v} x_t t$). α and β are computed as follows: let $\bar{t}$ be $(u + v)/2$ (i.e., the medium timestamp of $X_{u,v}$).

$$\text{(intercept) } \alpha = \mu - \beta\bar{t} \tag{4a}$$

$$\text{(slope) } \beta = \sum_{t=u}^{v}(t - \bar{t})(x_t - \mu)/\sum_{t=u}^{v}(t - \bar{t})^2 \tag{4b}$$

4.3.2 Score computation. The mergeability score k in L-EBinning is based on the Student's t-test for the equality of slopes of the two regression lines. The mergeability score between two neighboring bins (bin_p and bin_{p-1}) is defined as:

$$k = \frac{|\beta_p - \beta_{p-1}|}{\sqrt{(\sigma_p^2/n_p) + (\sigma_{p-1}^2/n_{p-1})}}. \tag{5}$$

In strict, Eq.(5) is an appropriate test when the variance of bin_p and bin_{p-1} belongs to the different distribution for each other. Therefore, this equation is prone to introduce some statistical bias [25]. To avoid this bias, we switch to the alternative k value in Eq.(6) when the two variances belong to the same distribution.

$$k = \frac{|\beta_p - \beta_{p-1}|}{\sqrt{[(\sigma_p^2 + \sigma_{p-1}^2)/(n_p + n_{p-1} - 2)]\,[(1/n_p) + (1/n_{p-1})]}} \tag{6}$$

For switching Eqs. (5) and (6), we apply the F-test to judge if the variances of the two bins belong to the same distribution. The F-test score is defined according to Eq.(7):

$$\text{F-test} = \frac{\sigma_{large}^2}{\sigma_{small}^2} \tag{7}$$

where σ^2_{large} (*resp.* σ^2_{small}) is the larger (*resp.* smaller) variance of either bin_{p-1} or bin_p. Algorithms 2 presents the computation of k in L-EBinning.

Algorithm 2: *ComputeScore* function of Linear-EBinning

Input: bin_p, bin_{p-1}, θ
Output: k
$F \leftarrow$ calculate the F-test by Eq. (7);
if $F \leq F_{\theta/2;n_p-1;n_{p-1}-1}$ **then**
  $k \leftarrow$ calculate score by Eq. (6) ;
else
  $k \leftarrow$ calculate score by Eq. (5) ;
return k

Note that $F_{\theta/2;n_p-1;n_{p-1}-1}$ is called a critical value of the F-distribution with respect to $n_{p-1}-1$ and n_p-1 degrees of freedom and a significance level of θ. In this paper, we set a 5-percent significance level for F-test because the 5-percent significance level is regarded as a convention for F-test [25].

4.3.3 Merging function. Assume bin_{p-1} and bin_p are merged into bin_{new}. We then compute the values of bin_{new} as follows:

$$n_{new} = n_p + n_{p-1} \tag{8a}$$

$$\beta_{new} = \sum_{t=u_{p-1}}^{v_p} (t-\bar{t})(x_t - \mu_{new}) / \sum_{t=u_{p-1}}^{v_p} (t-\bar{t})^2 \tag{8b}$$

$$\alpha_{new} = \mu_{new} - \bar{t}\beta_{new} \tag{8c}$$

where $\bar{t}$ is the median timestamp (i.e., $(u_{p-1}+v_p)/2$), and μ_{new} is shown at Eq. 3b.

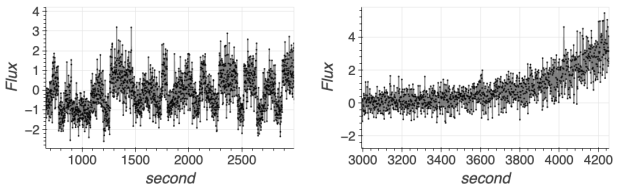
5 LCS DATASET FROM THE OPTICAL TELESCOPE

Our target dataset considered here comprised LCs, and it was obtained from an optical wide-field video observation system composed of a mosaic complementary metal-oxide semiconductor camera on the Kiso Schmidt telescope for sky survey [2]. We categorized several essential characteristics into four scenarios that occurred during the survey as follows:

- Stable behavior: This period corresponds to near-constant stability and may provide noisy data.
- Unstable behavior: This occurs suddenly or abruptly within a period. This behavior may be caused by cloud turbulence or the brightness of a nearby star, as shown in Figure 5a
- Outlier point: Outliers are uncommon and may be a result of noise, as shown in Figure 3b
- Gradual change: A period that exhibits gradual changes that may become the new normal behavior. Although this is a unique pattern in astronomy, it is not our objective, as shown in Figure 5b

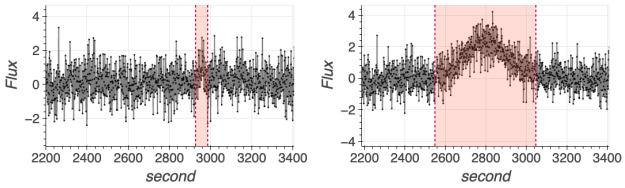
Notably, we applied the z-normalization to each LC for evaluation. EBinning was thereafter explored to the normalized LCs.

[2]The dataset provided was obtained from the Tomo-e Gozen project of the Kiso Schmidt telescope. For more detail, visit https://tomoe.mtk.ioa.s.u-tokyo.ac.jp/



(a) Unstable behavior **(b) Gradual change**

Figure 5: Sample scenarios in our target dataset.



(a) Smallest triangle-wave **(b) Largest triangle-wave**

Figure 6: Smallest and largest transient patterns

In our experiment, we injected two types of artificial transient patterns: square- and triangle-waves to evaluate the performance of our proposed method. For each LC, we injected one transient pattern. Thereafter, we knew the starting and end points of each transient pattern, which are used for computing the accuracy.

We defined square- and triangle-wave transient patterns with various durations and heights. For each combination of transient pattern, we have 432 LCs files. In total, there are 7,616 LCs files each of which has one square- or triangle-wave transient pattern.

The duration and height of the smallest transient pattern are 60 instances and 1σ, respectively. Conversely, the largest transient pattern exhibits 500 instances for the duration and 3σ for strength height. Notably, σ is the standard deviation of the original data for each file. The smallest and largest transient patterns are presented in Figs. 6a and 6b, respectively.

NOTE: Interested readers can view the report [19] to see more samples from our target dataset.

6 EXPERIMENTS

The performance of the proposed method was experimentally evaluated. M- and L-EBinning were implemented using Python 3.9.2. The evaluations of this study consider three different objectives: i) transient patterns capturing quality, ii) representation quality and iii) transient pattern detection. We used five methods for evaluation as follows: PYS [17, 18], PAA [11][3], SAX [13][3], 1D-SAX [15][3] , and Matrix Profile (MP)[4] with Fast Low-cost Unipotent Semantic Segmentation (FLUSS)[5] [9, 26].

[3]PAA, SAX, and 1D-SAX were implemented using the Tslearn library. For more details, visit https://tslearn.readthedocs.io/en/stable/index.html.
[4]MP was implemented using the MatrixProfile library. For more details, visit https://matrixprofile.docs.matrixprofile.org
[5]FLUSS was implemented using the Stumpy library. For more details, visit https://stumpy.readthedocs.io/en/latest/index.html.

Table 1: Comparison IOU of LCs representation

Methods	w	Square-wave transient pattern				Triangle-wave transient pattern			
		Heigth = 1σ		Heigth = 3σ		Heigth = 1σ		Heigth = 3σ	
		Duration 60	Duration 500	Duration 60	Duration 500	Duration 60	Duration 500	Duration 60	Duration 500
M-EBinning	20	0.66	0.71	0.91	0.78	0.39	0.46	0.52	0.30
L-EBinning	20	0.22	0.50	0.32	0.56	0.19	0.47	0.32	0.54
PYS	20	0.39	0.76	0.38	0.76	0.20	0.43	0.17	0.42
PAA	20	0.26	0.43	0.26	0.43	0.26	0.43	0.26	0.43
SAX N = 4	20	0.13	0.68	0.23	0.71	0.11	0.55	0.15	0.51
SAX N = 20	20	0.24	0.54	0.26	0.64	0.21	0.47	0.25	0.46
1D-SAX 4x10	20	0.24	0.51	0.26	0.51	0.21	0.44	0.24	0.43
1D-SAX 5x5	20	0.22	0.56	0.26	0.59	0.16	0.46	0.24	0.43
1D-SAX 10x4	20	0.24	0.52	0.26	0.54	0.22	0.43	0.25	0.43
MP L = 60	20	0.13	0.29	0.15	0.30	0.14	0.29	0.14	0.29
MP L = 200	20	0.18	0.35	0.18	0.37	0.19	0.33	0.18	0.38
PAA	71	0.61	0.12	0.63	0.12	0.64	0.12	0.63	0.12
SAX N = 4	71	0.54	0.62	0.62	0.93	0.45	0.43	0.61	0.67
SAX N = 20	71	0.56	0.51	0.63	0.93	0.51	0.41	0.62	0.61
1D-SAX 4x10	71	0.61	0.24	0.63	0.31	0.63	0.22	0.62	0.23

6.1 Parameter setting

6.1.1 Window size (w). In our evaluation, we first find an appropriate w (from 5 to 50) for each LC and method by TWIN and thereafter compute the average among all the obtained w values. Consequently, the average is 20, and we use it for fair evaluation.

In addition, we evaluate the influence of w on PAA, SAX and 1D-SAX. Note that these methods fix the bin size in advance. The case that $w = 71$ corresponds to that we set the bin size (n) as 60 (i.e., $t = w \times n$, where t is the LC length). In this case, the bin size has the same transient pattern as the duration of 60.

6.1.2 Symbol size for SAX and 1D-SAX. SAX and 1D-SAX request the symbol size to quantize the mean and mean-slope values for each bin, respectively. We quantize the mean values into four symbols for SAX ($N = 4$). In contrast, we quantize the mean values into four and the slope values into 10 (4×10) for 1D-SAX.

We add a merging function to SAX and 1D-SAX for fair comparison in such a way that for each bin, if its symbol is the same as that of the neighbor, then we merge two bins into one. For example, when SAX has ten bins, it would be represented by $[b, a, a, a, b, c, b, a, d, c]$. Three bins are the same symbols (red symbols). We then merge these three bins into one bin. The result is as follows $[b, a, b, c, b, a, d, c]$.

6.1.3 Subsequence length (L) for MP. For MP and FLUSS, we set two parameters of the subsequence length (L) and window size (w), respectively. We use the same window size $w = 20$ as explained in Section 6.1.1. For each subsequence of the LC, the MP can compute the distance with its most distinct subsequence over the LC. This discord distant is applicable to detect a transient pattern in our domain. Indeed, if a subsequence was transient, then its discord distance should become higher. We use two values of L; $L = 60$ and $L = 200$ correspond to the transient patterns with durations of 60 and 200, respectively.

NOTE : We summarized only the essential results because of space limitation. For more details, interested readers can obtain the report [19] to see all results.

6.2 Transient patterns capturing quality

The experimental goal is to justify how the transient pattern is appropriately captured by small bins. This evaluation is performed using the notion of *intersection over union* (IOU).

First, we focus on those bins that have non-empty intersections with the transient pattern injected in each LC. For example, let $X_{100,159}$ be the injected transient pattern with a duration of 60. Suppose that there are three bins associated with the interval $[100, 159]$. This implies that the transient pattern requires the three bins for its representation. We may consider this extreme bin that summarizes 500 instances containing $X_{100,159}$. The bin size is too large to compactly capture the transient pattern. Then, we consider the IOU metric to measure the overlap between the transient pattern and bins. Given a transient pattern $X_{u,v}$ and a bin b_p w.r.t. Xu_p, v_p, the IOU of b_p w.r.t. $X_{u,v}$ is defined as $|X_{u,v} \cap X_{u_p,v_p}|/|X_{u,v} \cup X_{u_p,v_p}|$. In case we find multiple bins associated with the transient pattern, we select only one bin that has the highest IOU.

Table 1 summarizes the results of IOUs for each method and transient pattern. Notably, the IOU in in Table 1 closest to 1 indicates that there exists a bin that appropriately covers the transient pattern.

As summarized in Table 1, M-EBinning has the highest IOU with $w = 20$. M-EBinning may not capture the large triangle-wave transient pattern (height of 3σ and duration of 500) with a high IOU. We found that L-EBinning is the highest IOU for large triangle-wave transient pattern capturing.

However, the IOU of L-EBinning is lower than that of M-EBinning for short-duration triangle-wave transient pattern capturing (duration of 60). This is because some short-duration triangle-wave transient patterns make it difficult to distinguish the slopes (see Figure 6a).

PYS had a low IOU for a short-duration transient pattern capturing, compared with M-EBinning. Note that the mergeability score of PYS is based on the T-testing for equal means. Our proposed score is based on the Hoeffding inequality that relaxes the Gaussian assumption for statistical T-testing of the PYS score. The experimental result demonstrates that M-EBinning is more suitable for short-duration transient pattern capturing.

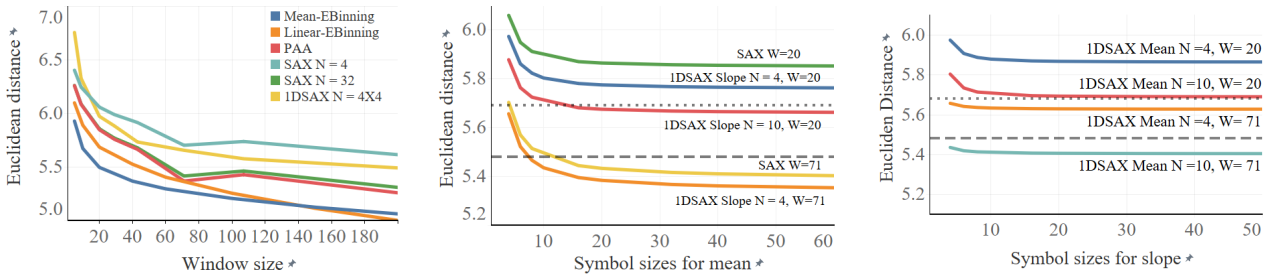
We explore two values of w as 20 and 71 for the three methods PAA, SAX, and 1D-SAX that fix the bin size. When the window size is 71 ($w = 71$), the bin size is approximately 60. This bin size corresponds to the transient pattern with a duration of 60. Accordingly, we observed that the IOUs of those methods had significant values when the duration of the transient pattern was 60. Hence, if we determined the transient pattern duration, it was possible to apply PAA, SAX, or 1D-SAX by adequately setting the bin size.

We also found that SAX with $w = 71$ had a high IOU with the long-duration transient pattern (duration of 500). Note that we added the merging function to SAX; thereafter, it could adjust the bin size similar to EBinning. If we did not apply the merging function, the result of SAX had the same trend as that of PAA.

MP had the lowest IOU because the subsequences of LCs are generally similar to each other. MP compares the similarity between the observational subsequence and every subsequence of LCs. Therefore, MP cannot obtain the most distinct subsequence capturing the transient pattern into one bin in this situation.

6.3 Representation quality

We evaluated the representation quality in terms of approximation error using the Euclidean distance between the original LC and the binning result for each method.



(a) Various window size **(b) Various symbols for means** **(c) Various symbols for slopes**

Figure 7: Comparison of Euclidean distance by various parameters

Figure 7 shows the comparison results of the Euclidean distance for various parameters. First, we compared the influence of window size on the PAA, SAX, and 1D-SAX methods (Figure 7a). The Euclidean distance depends on the window size, and it can be controlled. However, a large window may not provide satisfactory results for transient pattern capturing, as demonstrated in the IOU results of Section 6.2. M-EBinning had the lowest Euclidean distance compared to other methods with the same window size.

Figures 7b and 7c show the influence by the number of the symbols for the mean and slope, respectively. Note that we set two thresholds; the dashed and dotted lines refer to the Euclidean distance of M- and L-EBinning, respectively, with w of 20. The results exhibit the same trend as shown in Figure 7a. Hence, we achieve higher IOUs and lower Euclidean distances by adjusting the bin size when the window sizes are equal.

6.4 Transient pattern detecting

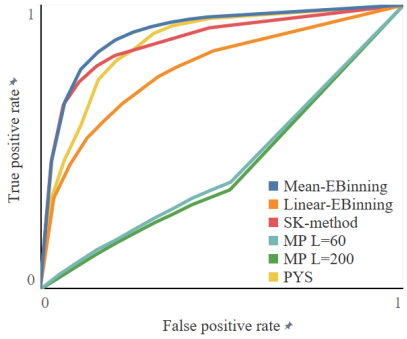


Figure 8: Comparison of ROC curve by various methods

We focused on the true positive and false positive rates for detecting the injected transient pattern for each method. The detection procedure comprises the following three steps:

(1) We summarize each LC by using each binning method with the window size $w = 20$;
(2) We list the top-k bin boundaries of the LC, according to the metrics of each method;
(3) If there exists a bin boundary when the injected transient pattern occurs, we admit that the transient pattern detection succeeds in the LC for each method.

Figure 8 shows the results of this experiment using the following five methods: M-EBinning, L-EBinnning, SK, MP, and PYS. There are some remarks. In the second step, we used the mergeability scores (Eqs. (2) and (5)) for ordering the bins of M- and L-EBinning, respectively. In turn, we used the existing SK and PYS methods by implementing them ourselves. The ordering metrics of SK (*resp.* PYS) is based on the Chebyshev inequality (*resp.* t-test score). In terms of MP, we selected the top-k subsequences with a certain length (L), according to their most distinct distances.

In total, we obtained satisfactory overall results of M-EBinning for detecting both the square- and triangle-wave transient patterns. In contrast, MP had a low positive rate, because the segmentation quality was not suitable in this situation, as described in Section 6.2. This MP result was consistent with that of the previous research experiment using MP with a seismograph recording time series [27]. Indeed, MP may have a high accuracy rate along with data binning techniques. This integration would be fruitful, although it is beyond the scope of the study.

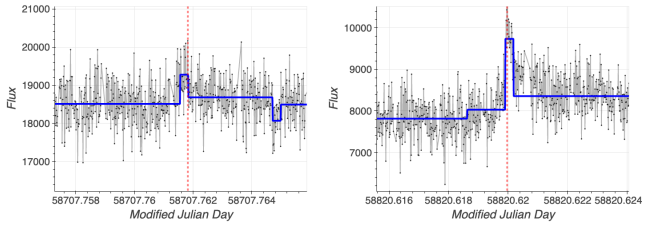
6.5 Special case studies in LCs with real natural phenomena

For a case study, we focus on LCs that contain real transient patterns of natural phenomena, not injected artificial transient patterns [6]. We discuss the essential result using M-EBinning and some limitations that need to be improved for further research.

6.5.1 Real transient pattern. M-EBinning exhibits normal behavior and transient patterns in LCs that have various noises from unwanted external factors. The obtained bins are regarded as a data summary of LCs that captures the real transient pattern. Figure 9 shows the obtained bins capturing the real transient pattern.

6.5.2 Two stars in same video. We found some LCs where M-EBinning generated false positives through expert human-analysis. Figure 10 shows this false positive case. It resembles a transient pattern in Figure 10, and we checked the original video. Thereafter, the video contained two stars; the first star occurred at the center of the video, and the second star was in front of the first star. Accordingly, the second star affected the LC. Although we have never noticed it before our application, this phenomenon could be managed by changing the radius for aggregating the light intensity. Although

[6]LCs dataset of this subsection were provided by M. Aizawa and K. Kashiyama. Interested readers can view the [2] for more details.



(a) (b)

Figure 9: Sketching with real transient patterns [2]

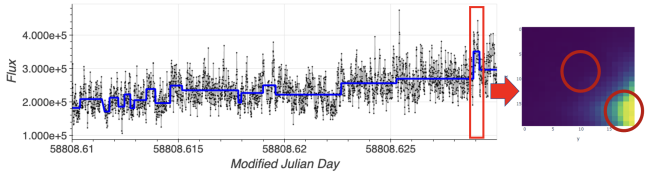


Figure 10: Example of a false positive case study [2]

the data binning technique can generate these false positives by unexpected external factors, it is useful to make users aware of those occurrences.

7 CONCLUSIONS

This paper introduced elastic data binning methods, known as EBinning, to summarize the original time series data using bins, provided that the characteristics of LCs similar to transients are preserved. The main conclusions are as follows:

Our proposed method can adjust the bin size by using the merge operation, which is based on a statistical hypothesis, wherein neighboring bins with similar features are merged into a large bin. The large bins corresponded to stable situations. Conversely, sequences containing small bins over a short period exhibited rapid changes.

The bins obtained by M-EBinning are essential when analyzing and exploring unknown transient patterns. Herein, we focused on data sketching from LCs for astronomical surveys. However, the slope feature obtained by L-EBinning was not so crucial compared with the mean feature by M-EBinning on this dataset. In future work, we intend to discover the conditions under which the approach with the linear regression can be used for available datasets in other fields.

In conclusion, the results of EBinning are highly appropriate for the representation of real transient patterns and will be helpful to astronomers when studying unknown phenomena in real-world situations with external factors. In future, we will develop further improvements to our proposed method such that it may be most effectively applied to a real environment.

ACKNOWLEDGMENTS

We are grateful to M. Aizawa and K. Kashiyama for their useful suggestions and providing the LCs dataset from survey for M dwarfs flares using Tomo-e Gozen on Kiso Schmidt telescope.

REFERENCES

[1] C. C. Aggarwal. 2017. *An Introduction to Outlier Analysis.* Springer International Publishing, Cham.
[2] M. Aizawa et al. 2022. Fast optical flares from M dwarfs detected by a one-second-cadence survey with Tomo-e Gozen. *PASJ* 74, 5 (Aug. 2022), 1069–1094.
[3] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano. 2021. A Review on Outlier/Anomaly Detection in Time Series Data. *ACM Comput. Surv.* 54, 3, Article 56 (Apr 2021), 33 pages.
[4] G. Chiarot and C.Silvestri. 2021. Time series compression: a survey. *CoRR* abs/2101.08784 (2021). arXiv:2101.08784 https://arxiv.org/abs/2101.08784
[5] G. Cormode. 2022. Current Trends in Data Summaries. *SIGMOD Rec.* 50, 4 (Jan 2022), 6–15.
[6] P. Esling and C. Agon. 2012. Time-Series Data Mining. *ACM Comput. Surv.* 45, 1, Article 12 (Dec 2012), 34 pages.
[7] Astropy Collaboration et al. 2018. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *The Astronomical Journal* 156, 3 (Aug 2018), 123.
[8] J. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, and A. Bouchachia. 2014. A Survey on Concept Drift Adaptation. *ACM Comput. Surv.* 46, 4, Article 44 (March 2014), 37 pages.
[9] S. Gharghabi et al. 2017. Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels. In *2017 IEEE International Conference on Data Mining (ICDM).* 117–126.
[10] G. Helou and C. A. Beichman. 1990. The confusion limits to the sensitivity of submillimeter telescopes. In *Liege International Astrophysical Colloquia (Liege International Astrophysical Colloquia)*, B. Kaldeich (Ed.), Vol. 29. 117–123.
[11] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. 2001. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowledge and Information Systems* 3, 3 (01 Aug 2001), 263–286.
[12] T. Kim and C.H. Park. 2020. Anomaly pattern detection for streaming data. *Expert Systems with Applications* 149 (2020), 113252.
[13] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. 2003. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms *(DMKD '03).* Association for Computing Machinery, New York, NY, USA, 2–11.
[14] Lott, B., Escande, L., Larsson, S., and Ballet, J. 2012. An adaptive-binning method for generating constant-uncertainty/constant-significance light curves with Fermi-LAT data. *A&A* 544 (2012), A6.
[15] S. Malinowski, T. Guyet, R. Quiniou, and R. Tavenard. 2013. 1d-SAX: A Novel Symbolic Representation for Time Series. In *Advances in Intelligent Data Analysis XII*, A. Tucker, F. Höppner, A. Siebes, and S. Swift (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 273–284.
[16] J. R. Martínez-Galarza et al. 2021. A method for finding anomalous astronomical light curves and their analogues. *Monthly Notices of the Royal Astronomical Society* 508, 4 (Sep 2021), 5734–5756.
[17] T. Phungtua-Eng, Y. Yamamoto, and S. Sako. 2021. Detection for Transient Patterns with Unpredictable Duration using Chebyshev Inequality and Dynamic Binning. In *2021 Ninth International Symposium on Computing and Networking Workshops (CANDARW).* 454–458.
[18] T. Phungtua-Eng, Y. Yamamoto, and S. Sako. 2021. Dynamic Binning for the Unknown Transient Patterns Analysis in Astronomical Time Series. In *2021 IEEE International Conference on Big Data (Big Data).* 5988–5990.
[19] T. Phungtua-Eng, Y. Yamamoto, and S. Sako. 2022. Supplementary material. https://sites.google.com/view/elasticdatabinning
[20] U. Rebbapragada, P. Protopapas, C. E. Brodley, and C. Alcock. 2009. Finding anomalous periodic time series. *Machine Learning* 74, 3 (01 Mar 2009), 281–313.
[21] Khalid S. 2006. 1 - Introduction. In *Introduction to Data Compression (Third Edition)* (third edition ed.), Khalid S. (Ed.). Morgan Kaufmann, Burlington, 1–11.
[22] G. Shevlyakov and M. Kan. 2020. Stream Data Preprocessing: Outlier Detection Based on the Chebyshev Inequality with Applications. In *2020 26th Conference of Open Innovations Association (FRUCT).* 402–407.
[23] R. Sulo, T. Berger-Wolf, and R. Grossman. 2010. Meaningful Selection of Temporal Resolution for Dynamic Networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs (MLG '10).* Association for Computing Machinery, New York, NY, USA, 127–136.
[24] B. D. Warner. 2016. *A Practical Guide to Lightcurve Photometry and Analysis* (2nd ed. ed.). Springer Cham, Cham, Switzerland.
[25] B. L. Welch. 1938. The Significance of the Difference Between Two Means when the Population Variances are Unequal. *Biometrika* 29, 3/4 (1938), 350–362.
[26] C. M. Yeh et al. 2016. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM).* 1317–1322.
[27] Y. Zhu et al. 2016. Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins. In *2016 IEEE 16th International Conference on Data Mining (ICDM).* 739–748.