# Efficient periodicity detection in irregular alarm event time series: application to building management system alarms*

Sylvain Marié, *Senior Member, IEEE*, Mathias Verdière, Pablo Knecht, Patrice Dake, Benjamin Norman

***Abstract*— Alarm rules are key to efficient monitoring systems (SCADA) in industrial facilities such as plants, datacenters, or tertiary buildings (BMS). Yet, misconfigured alarms drastically impact their operational usability. This paper first presents a large-scale exploratory data analysis of alarm volume distributions across more than a thousand real-world buildings, highlighting the need for efficient and scalable approaches. It then focuses on detecting periodic alarm patterns at scale in large databases, so that they may be filtered out or fixed. The proposed method combines novel, efficient resampling-free temporal distance metrics with k-means clustering. It is validated on an annotated real-world dataset of 321 alarm series from 22 buildings. The results show strong performance (91.5% F1-Score), outperforming classical methods, while providing valuable interpretability through the extraction of meaningful calendar-based (daily) periodicity patterns.**

*Keywords:* periodicity detection, alarm events timeseries, binary series, irregular sampling, pattern mining, temporal distance metrics, daily activity, alarm rationalization, building management.

## I. Introduction

Monitoring and control systems (SCADA) are massively used in industrial facilities such as plants, datacenters, or tertiary buildings - where they are also known as Building Management Systems (BMS). Alarms play a central role in usability of such systems for efficient monitoring; misconfigured alarm rules may result in catastrophic flooding, preventing operators from efficiently managing their process, and potentially causing damage due to missed critical events [1-3]. Standards such as ANSI/ISA-18.2 provide general guidelines and processes to improve the situation on a specific site, however the volume of alarm events can be such that facility managers and field engineers have difficulties finding a starting point.

Misconfigured alarms can exhibit various behaviors. For example, stale alarms remain in an "active" state for extended periods; chattering alarms repeatedly switch between reset and active states in an unstable manner. Periodic alarms trigger or reset according to a regular pattern, making them less informative for operators. For instance, an alarm ringing every 5h generates 33 events per week, contributing to the global flooding, although each individual event provides little useful information. Even when the overall volume of such events is low, field operators get the habit to ignore them – which can lead to underestimating a genuine incident. Detecting and filtering nuisance alarms, particularly periodic ones, is therefore a key step towards improving the visibility of truly informative alarms. Additionally, by identifying each kind of alarm nuisance, a system can also recommend more appropriate configurations – ultimately aiming for fully well-behaved alarms.

Binary irregular alarm series are a particular type of state transition event time series (STE-ts), a complex data object studied in the literature from multiple perspectives, ranging from sequence mining to time series analysis [4, 5]. A key challenge with such data is to avoid uniform resampling, a computationally intensive task inducing signal distortions [5].

Spectral analysis is the most widely used approach for periodicity detection. Typically, a periodogram or correlogram is computed using FFT or ACF, and peaks exceeding a predefined threshold are then identified as candidate periods. Improved period detection methods combine FFT and ACF [6, 7] or rely on FFA [8, 9]. However, these approaches generally involve resampling and require a trade-off between detecting high- and low- frequency patterns, often influenced by the sampling period. Another approach to periodicity detection consists of mining candidate patterns along with their support [10]. This strategy can be computationally expensive even with efficient algorithms such as FP-growth [11]. When focusing on daily or weekly periodicity for a given alarm, the time series can be segmented into daily or weekly intervals. In this case, the problem reduces to detecting high correlation (e.g. *Equal* correlation [12]) [13]. Dedicated time series similarity metrics provide alternative ways to measure such correlations [14]. Recent work on resampling-free temporal metrics such as Temporal Hamming, Temporal Jaccard, and their generalization, Selective Temporal Hamming, promises faster similarity computation with increased precision [5].

Some issues are common to most approaches: selecting a relevant period segment or time window, mitigating computational cost, handling irregular occurrence of periodic patterns, defining thresholds (e.g., minimum support or minimal spectral power), and distinguishing signal from noise. In buildings, many alarms are driven by human activity; thus, using predefined period segments such as weekly or daily does not substantially reduce generality. Moreover, detected periodicities must be reviewed by facility management teams for possible action. Therefore, our primary objective is to identify candidates for review. Identifying (mining) patterns serves as a secondary objective, aimed at improving explainability.

This paper investigates algorithms to efficiently detect periodicity on industrial alarm time series, in particular BMS alarms, at scale, without resampling. The rest of the paper decomposes as follows. Section II presents a study of alarm events volume over a set of tertiary buildings and introduces

* S. Marié, P. Knecht, M. Verdière, P. Dake are with Schneider Electric, Artificial Intelligence Hub, IntenCity, 160, Avenue des Martyrs, 38000 Grenoble, FRANCE (e-mail: first.last@se.com).

B. Norman is with Schneider Electric, Artificial Intelligence Hub, 1216 Broaway Suite 320, New York, NY, 10001, United States.

the problems of repeat periodicity and daily periodicity detection. Section III details state of the art and its limitations, presents our approaches, and introduces performance evaluation metrics. Sections IV and V are devoted to experimental results and their analysis, both from a quantitative viewpoint and through qualitative discussions. We finally conclude in Section VI.

## II. Objective: Periodicity Detection On Binary Alarm Series

### A. Binary alarm series in Tertiary and Industrial Buildings

We consider binary alarms, that is, alarms with a state. This state is usually defined by a set of rules or "alarm conditions" such as a high threshold on a temperature variable, sometimes filtered with hysteresis or delay (Fig. 1).

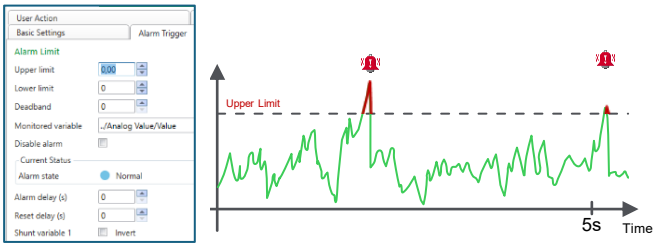


Figure 1. A BMS Alarm configuration screen (left) ; illustration of a simple upper limit temperature alarm (right)

*Active* is the state where the associated alarm conditions are met, while *Inactive* is the state where these conditions are not satisfied. A *Trigger* event corresponds to a transition from *Inactive* to *Active* state. A *Reset* event corresponds to a transition from *Active* to *Inactive*.

An alarm *series* is the sequence of timestamped transition events for the same alarm rule. If the initial state is known, the state of the alarm at any point in time can be reconstructed from these transition events. For this reason, BMS and industrial SCADA systems store *event logs*, as they provide a compact and lossless representation of data.

Building alarms in the field exhibit a wide range of dynamics, driven both by the nature of underlying monitored time series (e.g. from slowly changing water temperatures to rapidly changing AC power, motor speed or valve positions), and by the alarm rules themselves (a threshold set inappropriately is frequently crossed by the variable). Fig. 2 illustrates this diversity with two real-world examples.


Figure 2. Alarm dynamics illustration for a 1-month period in an educational building: low (top), high (bottom) activity air flow alarm

A recent survey conducted by our team on 1 413 real-world tertiary buildings spanning from offices to education, retail, hotels and hospitals confirms this diversity. The distribution in Fig. 3, although highly skewed toward low-activity series, still highlights an amazing range from a few events per month to 3.5 million events per month for a single alarm series.

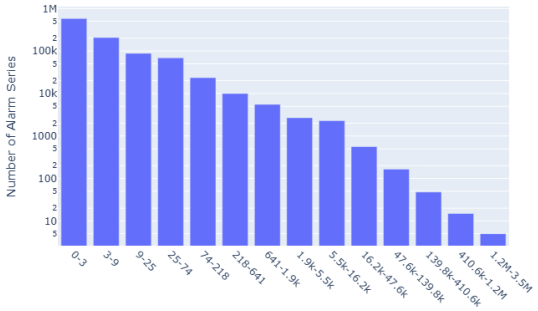


Figure 3. Distribution of the number of alarm and reset events (x-axis) for 1M alarm series from 1 413 real-world buildings (y-axis: nb of alarm series)

The overall volume of alarm events to process is also governed by the number of alarm series per building. We observed that up to 25 000 binary alarm series can be defined for a single building. We even found in a previous survey from 2024 a building with 32 733 alarm series (!). The average number is around 700 alarm series per building (Fig. 4).

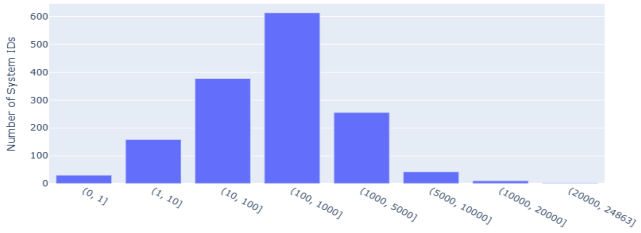


Figure 4. Distribution of the number of alarm series (x-axis) for 1 413 real-world buildings (y-axis: number of BMS systems)

As a result, the number of alarm events generated per building ranges from 1 to 8 million events per month, with an average of 77k events/month (Fig. 5). Note that unary (non-binary) alarm series (i.e. alarms that only generate alarm events but no reset events) were out of scope of this study – although they are also widely used in the field. Consequently, these figures likely underestimate the overall volume of alarm events in buildings.

### B. Periodicity detection

In this study we focus on two intuitive definitions of periodicity: "repeat periodicity" and "calendar periodicity". These definitions are commonly used in the literature, although other definitions exist [15].

*Repeat periodicity* focuses on time durations between two consecutive alarm events, ignoring the "reset" events. This concept therefore also applies to unary (non-binary) alarm series. A series exhibiting regular inter-alarm durations is considered periodic: for example, an alarm triggering every 5h. We extend this definition to multi-modal regularity: a series can have two or more reoccurring inter-alarm durations (e.g. 5h and 3min). On real-world buildings, we observe inter-activation durations ranging from nanoseconds to more than a month. Repeat periodicity patterns are therefore expected to be

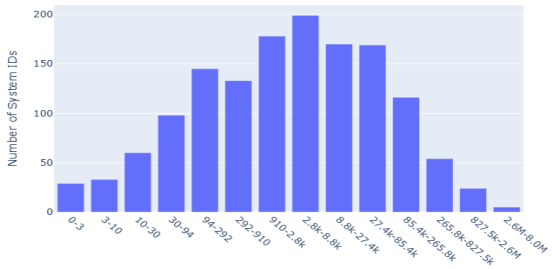


Figure 5. Distribution of 1 413 real-world buildings (y-axis: number of BMS systems) according to the number of alarm events (x-axis)

as diverse as this range. We name "repeat period" one of the regular inter-activity periods found if any. The series is considered repeat periodic if such a period is found to occur more than a given number of occurrences within the observation window. Fig. 6 presents a repeat periodic alarm over a month, with a single repeat period of 37.2h, occurring 20 times.


Figure 6. A repeat periodic alarm with 1 repeat period occuring 20 times

*Calendar periodicity* focuses on detecting repeating patterns on the binary alarm series, at a human calendar scale: daily, weekly, monthly… For this study we restrict our analysis to daily periodicity: an alarm series exhibiting "typical daily patterns" that would appear frequently is considered periodic. Here again, we extend this definition to multi-modal regularity, so several "typical days" could be present in a daily periodic series. Fig. 7 presents a daily periodic alarm over a week.

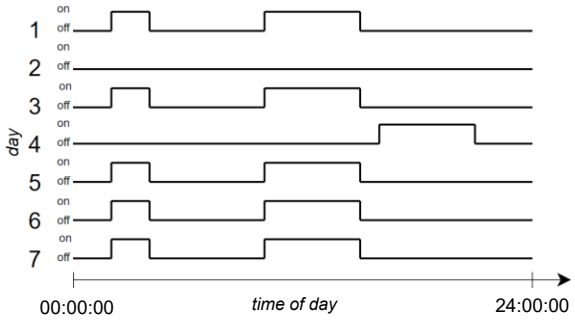


Figure 7. A daily periodic alarm with one typical day occuring 5 times

## III. State of the Art and our Approach

### A. State of the Art – limits

Repeat periodicity detection is usually addressed through autocorrelation and spectral analysis methods.

*Autocorrelation methods* consist in using a set of predefined lags to apply to the series, and to compute a similarity (usually cosine distance) for each lag. The autocorrelation function is noted ACF. Slotting and kernels [16] may be used to cope with irregular samples. Among known limitations, autocorrelation is very sensitive to noise and outliers, is highly impacted by the sampling period, and is not considered reliable to detect multiple periodicities [7, 17].

*Spectral methods* rely on the Fast Fourier Transform (FFT) to create classical/Schuster periodograms. The case of irregular (non-uniform) sampling is handled by variants relying on some fitting to accommodate for the gaps (NFFT, Lomb-Scargle Periodogram, widely used in the astrophysics domain)[15, 18]. For categorical and binary time series, a Walsch-Fourier Periodogram [19] is considered a better alternative, as Walsch functions accommodate for the sharp transitions that are naturally present in these series – as opposed to classical Fourier which models the signals as combination of sinusoids. Overall, proposed methods have an algorithmic complexity of at least $\mathcal{O}(n.log\ n)$ where $n$ is the number of samples after resampling – that is hardly compliant with the volumes of data observed in real-world buildings. A well-known significant drawback of FFT is that the sampling period $P$ impacts both the minimum detectable repeat period $2P$ (Nyquist frequency) and the frequency bin resolution: a tradeoff is needed.

Some methods [6, 7, 20] combine FFT and ACF to more efficiently detect multiple periods. A recent method [21] has been specifically proposed for irregularly sampled binary (-1, +1) event sequences. It however has high algorithmic complexity when raw timestamps are not aggregated to large time buckets and when all possible periods are tested.

Concerning daily periodicity detection, it can be considered a particular case of periodicity detection for period P=24h – in which case ACF and spectral methods apply and associated literature focuses on defining an appropriate threshold. An alternate set of work consider daily periodicity as a frequent pattern mining problem, where similar days are grouped in clusters [13] – an approach largely used in the Energy domain to search for frequent daily patterns [22, 23].

### B. Proposed Approach

*Repeat periodicity*: our approach to detect repeat periodicity is based on applying a standard clustering algorithm (e.g. k-means) on the set of all inter-activity periods of a series, that is, the time deltas between consecutive activation events. Clusters provide candidate repeat periods, that are validated or rejected in a post-processing step.


Figure 8. Repeat Periodicity detection

The final algorithm is a pipeline going from raw alarm binary event series to the periodicity detection, with the following aspects (Fig. 8).

- *Binary alarm event series*: they are extracted from the global Alarm Event Log as binary irregular time series where each timestamp is associated with a change of state. The series is sorted by increasing timestamp.
- *Timedelta computation*: we compute the time duration between all pairs of consecutive alarm activations.
- *Clustering*: we use k-means to cluster time deltas. Hyperparameters (e.g. number of clusters) are optimized through grid search with Silhouette score.
- *Validation*: we validate each cluster according to cluster statistics (nb occurrences, std deviation…)

When there is at least one valid cluster, the series is declared periodic. Note on complexity: k-means scales linearly with the number of data points, clusters, dimensions, and iterations. In our case dimension is 1 and number of valid clusters is low in practice (less than 10). The number of data points is the number of trigger events (discarding reset events), so it accounts for approximately half of the number of events.

*Daily periodicity:* our approach to detecting daily periodicity is a classic daily clustering methodology, combined with a resampling-free metric [5] for fast execution at scale, and to avoid distortion induced by resampling.


Figure 9. Daily Periodicity detection

The pipeline goes from raw alarm binary event series to periodicity detection, with the following aspects (Fig. 9):

- *Binary alarm event series*: they are extracted from the global Alarm Event Log and preprocessed as in the repeat periodicity pipeline.
- *Daily splitting*: split series to create one series per day.
- *Distance computation*: we compute temporal distance metrics [5] between pairs of time series. There are k*(k-1) pairs where k is the number of days in the observation period. Results are stored in the upper /lower triangles of a square symmetric distance matrix
- *Clustering*: we use Agglomerative Clustering with the custom distance metric to cluster days. Hyperparameters (e.g. number of clusters) are optimized through grid search with Silhouette score.
- *Validation*: we validate each cluster according to cluster statistics (nb occurrences, std deviation…)

When there is at least one valid cluster, the series is declared periodic. Note that for this method as well as for repeat periodicity, sensitivity to the clustering method was not explored as part of this study. We however expect other methods accepting custom metrics (Spectral, Agglomerative…) to lead to similar results, yet with their own specificities in hyperparameter optimization and complexity.

### B. Performance evaluation

We refer to state of the art classification metrics to assess the performance of our two detectors. *Accuracy* is the proportion of correctly labelled series. *Precision* measures the ability of our algorithm to detect the right series while *Recall* measures its ability to find *all* periodic series. *F1-Score* is their harmonic mean, that is better suited to unbalanced classes than *Accuracy* but is less interpretable. Each detection task (repeat periodicity, daily periodicity) is considered an independent 2-class classification task where the "is periodic" label provided by the detector is compared with the ground truth. An additional task (*overall periodicity detection*) is also evaluated with the same metrics by combining detections from both daily and repeat periodicity with a logical OR.

## IV. Experiments and Dataset description

### A. Dataset

For the purpose of this study, we extracted a reference anonymized dataset from real-world building alarm event logs. A group of data scientists and subject matter experts manually inspected the corresponding binary alarm series and for each series, agreed on a consensus boolean label for both kinds of periodicity patterns. The two labels of each series are independent. Alarm series with too many events (>500/days) were removed to ease the manual inspection process, as well as those for which annotation would be too subjective – resulting in a size of 237 annotated alarm series. A test set of 84 series was also created with a similar procedure [24].

### B. Experimental Setup

*Proposed approach*: we executed the repeat and daily periodicity detectors on the dataset. For daily periodicity we used temporal Hamming *TH* and Jaccard *TJ* metrics from [5] and introduced two new *STH*-derived metrics:

- *Inverse Temporal Jaccard* is defined as $STH_{(\{0\},\{1\})}$. As opposed to *TJ*, *ITJ* focuses on reset states.
- *Temporal Jaccard Average TJA* is defined as the average of *TJ* and *ITJ*: $(TJ + ITJ)/2$. As opposed to *TH* it also captures significant transient differences.

All functions were implemented in Python using `numpy`, `pandas`, `scikit-learn`.

*Baselines*: we executed the autocorrelation and Lomb-Scargle methods on the dataset. ACF is computed on the resampled time series with a "last" aggregation strategy using `statsmodel.acf` using the fft-based implementation (`numpy fft/pocketfft`). If a non-trivial peak (lag > `min_lag`) is detected on the correlogram above a given `threshold`, the series is declared repeat periodic [6]. If a peak exists above the threshold with period `k*24h` (with a `tolerance`), and other peaks appear at integer harmonic periods `n*k*24h` with `n=(2,3…)` a certain number of time (`min_nb_harmonic_peak` with a `tolerance`), the series is declared daily periodic. Normalized floating mean Lomb-Scargle algorithm [18] is executed on the raw time series with `lombscargle` function from `scipy` to produce a periodogram. If a peak exists with power above a `threshold` [25], the series is declared repeat periodic. If a peak exists above the threshold, with period 24h (with a `tolerance`), the series is declared daily periodic.

All algorithm parameters were tuned on the training dataset. Grid search ranges and optimal parameters are provided in Table I. All code was executed on a laptop PC with 32 Go RAM. We measured the average execution time over 50 runs using python `timeit`.

## V. Experimental Results

### A. Quantitative performance

Results for the experiment are presented in Table II. Overall, the proposed approach obtains the best F1 Scores for both Repeat and Daily periodicity detection, providing a significant improvement compared with the baselines. Yet the benefits highly depend on the task. For repeat periodicity, the huge performance gain (+15.8pts F1) due to a perfect precision

TABLE I. PARAMETERS

| Approach | Parameter name (R: repeat only, D: daily only) | Grid search ranges | Optimal Parameter | |
|---|---|---|---|---|
| | | | Repeat | Daily |
| Autocorr. | Resampling period | ["15s", "30s", "1min", "5min", "10min"] | 30s | 30s |
| | Threshold | [0.15, 0.16, 0.18, 0.20, 0.25, 0.30, 0.35] | 0.18 | 0.16 |
| | Maximum lag | [100,200,500,1000,1500,3000,3500,5000,10000,15000,20000] | 20 000 | 20 000 |
| | Minimum considered lag (R) | [3, 4, 5, 6] | 4 | - |
| | Base Freq tolerance (D) | [0.01, 0.02, 0.03] | - | 0.01 |
| | Min nb harmonic peaks | [2, 3] | 2 | 3 |
| | Harmonic tolerance (D) | [0.10, 0.20, 0.30] | - | 0.3 |
| Lomb-Scargle | Number of frequencies | 27 (repeat) or 50 (daily) numbers between 100 and 50000 | 15 000 | 3 000 |
| | Threshold | [0.01, 0.16, 0.18, 0.2 , 0.22, 0.24, 0.26, 0.3 , 0.4 , 0.5 , 0.6 , 0.7] | 0.7 | 0.2 |
| | Freq tolerance (D) | [0.01, 0.03, 0.05, 0.07, 0.09, 0.11, 0.13, 0.15] | - | 0.03 |
| Proposed Approach | Maximum number of clusters to search for (silhouette) | - | 9 | 9 |
| | Minimum cluster size as ratio of all samples | - | 40% | 9/31 (29.03%) |
| | Maximum cluster spread to validate a cluster | (R) -<br>(D) 30 values between 0 and 1 | 20 min | 0.37 |
| | Min. number of similar days to validate periodicity (D) | [4, 5, 6] | - | 4 |
| | Temporal distance metric (D) | *TJ, TH, ITJ, TJA* | - | *TJA* |
| | Linkage of agglom. clustering | Single, Complete, Average | - | Single |

(100%) has a major cost in terms of execution time, that may be explained by the number of time delta samples to be processed by the k-means algorithm. For daily periodicity the Autocorrelation method is surprisingly precise, so the gain is smaller (+3.5pts F1). The execution time of the proposed approach has a similar order of magnitude, which is very encouraging considering that some of its code is not compiled.

Results for the global classification task reflect the above conclusions: the proposed approach provides performance gains (+4.1pts F1) at the cost of the large repeat periodicity execution time. Autocorrelation confirms its position as a solid alternative. Although Lomb-Scargle is theoretically promising as a resampling-free method, its weak recall scores make it a less preferable choice for building applications, where finding most issues is crucial.

### B. *Discussion, qualitative performance*

In this section we discuss some of the periodicity detection results from this experiment. Fig. 10 illustrates multiple *daily periodicity* patterns found on alarm series 'P206'. Cluster 0 shows how the method accommodates small discrepancies (day 27). Clusters 2 and 5 illustrate the precision of the method: while clusters are almost similar, cluster 5 pattern has one more "bump" and is shorter. This capability to extract relevant patterns is a clear benefit as opposed to autocorrelation method.

Fig. 11 illustrates a repeat periodic series with two typical periods. Our method detected two clusters with respective periods 1min15s and 12h46min8s. We performed a simple autocorrelation on the resampled 5min series with lags of 5min and max lag 1day. The correlogram plot shows a peak at 153 lags (12h45min) but it is a bit "fuzzy" as there is also a lot of energy at 154 and surrounding. The peak at zero lag "hides" the small peak at 1 lag, corresponding to 5min period. This highlights the influence of the choice of sampling period and lag size. Our methodology can detect any period, small or large, precisely and independently of such parameters tuning.

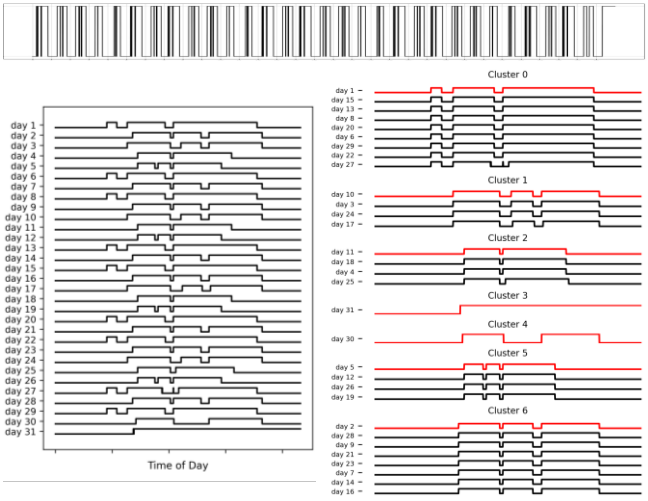


Figure 10. Example daily periodicity results: series (top), split by days (left), patterns found (right)

## VI. CONCLUSION

This work explored solutions to address the major problem of alarm flooding in Building Management Systems and generally in industrial SCADA systems – where scalable analysis of millions of events is critical. We first introduced a novel study on alarm volume statistics for 1 413 real-world buildings, illustrating the challenge induced by the scale of real alarm events. We then presented a method to efficiently detect two kind of periodicity patterns in binary alarm series at scale, thanks to a combination of novel resampling-free distance metrics and standard clustering algorithms. By avoiding uniform resampling, the method addresses key challenges of fast execution and high precision of found patterns. Experiments on an annotated real building dataset confirm the capability of the method to accurately detect both repeat and daily periodicity, and to extract interpretable relevant patterns. In particular, the new *TJA* metric proves valuable to capture small pulse-like differences in daily patterns. However, the execution time for repeat periodicity raises doubts concerning its scalability. Future work on this task may therefore investi-

TABLE II. RESULTS

| Pattern | Approach | Test (Train) Metrics | | | | |
|---|---|---|---|---|---|---|
| | | *Accuracy* | *Precision* | *Recall* | *F1 score* | *Avg. (+/- std) execution time in s* |
| Repeat periodicity | Autocorrelation | 79.3% (76.1%) | 58% (75%) | **84.3%** (66.6%) | 68.7% (70.5%) | **4.89 (0.32)** |
| | Lomb-Scargle | 75.0% (81.4%) | 77.8% (62.2%) | 58.3% (79.7%) | 66.7% (69.9%) | 48.43 (4.06) |
| | This approach | **84.5%** (98.7%) | **100%** (98.4%) | 63.9% (96.8%) | **84.5%** (97.6%) | 109.16 (11.63) |
| Daily periodicity | Autocorrelation | 85.2% (75%) | **90.4%** (87.8%) | 83.2% (69.2%) | 86.6% (77.4%) | **4.68 (0.19)** |
| | Lomb-Scargle | 69.0% (73.0%) | 88.2% (92.0%) | 57.7% (58.4%) | 69.8% (71.4%) | 6.93 (0.57) |
| | This approach | **88.1%** (85.8%) | 86.2% (90.1%) | **96.2%** (82.4%) | **90.1%** (86.4%) | 7.61 (0.52) |
| Periodicity (repeat or daily) | Autocorrelation (x2) | 85.2% (83.3%) | 89.7% (92.1%) | 85.3% (82.4%) | 87.4% (87%) | **9.50 (0.45)** |
| | Lomb-Scargle (x2) | 69.0% (78.9%) | **91.9%** (93.5%) | 59.6% (69.9%) | 72.3% (80.0%) | 55.36 (4.50) |
| | This approach | **88.1%** (92.5%) | 88.5% (92.8%) | **94.7%** (94.2%) | **91.5%** (93.5%) | 116.77 (11.78) |

-gate faster alternatives including hybridation with ACF, FFT/Walsch or FFA [6-9, 19-20] – aiming at keeping the accuracy benefits while increasing the speed. An additional scalability experiment could also be performed for both detectors, similar to [5], as the manually annotated dataset was biased towards small series due to the annotation process. Finally, research perspectives include extension of calendar periodicity to pure events (non-binary) and application of temporal metrics to efficient alarms grouping.

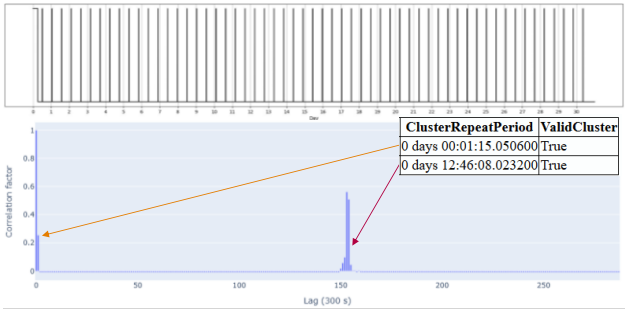


Figure 11. Example repeat periodicity results: series (top), found clusters (center right), autocorrelation peaks (bottom)

## REFERENCES


[1] S. Kordic, P. Lam, et al., "Associative data mining for alarm groupings in chemical process", in *Int. Conf. Intelligent Systems and Knowledge Engineering (ISKE)*, Chengdu, China, 2007.

[2] M.L. Bransby, J. Jenkinson, "The management of alarm systems: a review of current practice in the procurement, design and management of alarm systems in the chemical and power industries", *HSE Research Report*, CRR 166, 1998. humanfactors101.com

[3] W. Hu et al., "An Application of Advanced Alarm Management Tools to an Oil Sand Extraction Plant", *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 641-646, 2015. doi: 10.1016/j.ifacol.2015.09.040

[4] M. Lucke, M. Chioua, et al, "Advances in alarm data analysis with a practical application to online alarm flood classification", *J. Process Control*, vol. 79, pp. 56-71, 2019. doi: 10.1016/j.jprocont.2019.04.010

[5] S. Marié, P. Knecht, "A Selective Temporal Hamming distance to find patterns in state transition event timeseries, at scale", in *1st Int. Conf. on Big Data Analytics and Applications (BDAA' 2025)*, Nov 2025, Innsbruck, Austria. hal-05383600

[6] M. Vlachos, P. Yu, and V. Castelli, "On periodicity detection and structural periodic similarity", in *Proc. 2005 SIAM international conference on data mining*, Newport Beach, USA, 2005, pp. 449–460.

[7] T. Puech, M. Boussard, A. D'Amato, G. Millerand, "A Fully Automated Periodicity Detection in Time Series", *Advanced Analytics and Learning on Temporal Data. AALTD 2019*, LNCS, vol 11986. Springer, Cham. doi: 10.1007/978-3-030-39098-3_4

[8] Y. Yang et al., "Long-Periodicity Detection from Massive Alarms in Power System", in *2024 International Conference on Networking, Sensing and Control (ICNSC)*, Hangzhou, China, 2024, pp. 1-6. doi: 10.1109/ICNSC62968.2024.10760156

[9] V. Morello, E.D. Barr, et al., "Optimal periodicity searching: revisiting the fast folding algorithm for large-scale pulsar surveys", *Monthly Notices of the Royal Astronomical Society*, vol. 497, no. 4, Oct. 2020, pp. 4654–4671, doi: 10.1093/mnras/staa2291

[10] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD*, Washington D.C., USA, 1993, 207–216. doi: 10.1145/170036.170072

[11] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation", in *Proc. 2000 ACM SIGMOD Int. Conf. on Management of Data*, New York, USA, 2000, pp. 1–12.

[12] J.F. Allen, G. Ferguson, "Actions and Events in Interval Temporal Logic", *Spatial and Temporal Reasoning*, 1997, pp 205–245, ISBN 978-0-585-28322-7, Springer. doi: 10.1007/978-0-585-28322-7_7

[13] M. Sepesy Maučec, G. Donaj, "Discovering Daily Activity Patterns from Sensor Data Sequences and Activity Sequences", *Sensors,* vol. *21*, no. 20, 6920, 2021. doi: 10.3390/s21206920

[14] C.-T. Do, et al, "Multi-modal and multi-scale temporal metric learning for a robust time series nearest neighbors classification*", Information Sciences*, pp. 418–419, 2017, doi: 10.1016/j.ins.2017.08.020

[15] A.F. Smeaton, F. Hu, "Periodicity Intensity Reveals Insights into Time Series Data: Three Use Cases", *Algorithms*, vol. 16, no. 2, 119, 2023. doi: 10.3390/a16020119

[16] K. Rehfeld, N. Marwan, et al., "Comparison of correlation analysis techniques for irregularly sampled time series", *Nonlinear Processes in Geophysics*, vol. 18, pp. 389–404, 2011, doi: 10.5194/npg-18-389-2011

[17] Yang, Z., Hou, L. and Zhao, X, "Robust Autocorrelation for Period Detection in Time Series", in *Proc. 17th Int. Conf. on Agents and Artificial Intelligence - Volume 2: ICAART*; pp. 36-44. doi: 10.5220/0013092500003890

[18] J.T. Vanderplas, "Understanding the Lomb–Scargle Periodogram", *The Astrophysical Journal Supplement Series*, 2018, vol. 236, no. 1, pp. 16 doi: 10.3847/1538-4365/aab766

[19] D. S. Stoffer,"Walsh-Fourier Analysis and Its Statistical Applications." *Journal of the American Statistical Association*, vol. 86, no. 414, 1991, pp. 461–79. doi: 10.2307/2290595.

[20] S. Parthasarathy, S. Mehta, and S. Srinivasan, "Robust periodicity detection algorithms", in *Proc. 15th ACM international conference on Information and knowledge management (CIKM '06)*, New York, USA, 2006, pp. 874–875. doi: 10.1145/1183614.1183774

[21] Q. Yuan, J. Shang et. al., "Detecting Multiple Periods and Periodic Patterns in Event Time Sequences", in *Proc. ACM 2017 Conference on Information and Knowledge Management (CIKM '17)*, New York, USA, 2017, pp. 617–62, doi: 10.1145/3132847.3133027

[22] G. Chicco, R. Napoli and F. Piglione, "Comparisons among clustering techniques for electricity customer classification", *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 933-940, May 2006.

[23] H. Najmeddine, F. Suard, et al, "Mesures de similarité pour l'aide à l'analyse des données énergétiques de bâtiments", in *RFIA 2012 (Reconnaissance des Formes et Intelligence Artificielle)*, Lyon, France, Jan 2012, pp.978-2-9539515-2-3. hal-00656546

[24] M. Verdière, P. Knecht, and S. Marié, "BMS alarm timeseries with periodicity labels (1.0.0)", *Zenodo*. doi: 10.5281/zenodo.19855612

[25] T. Ruf, "The Lomb–Scargle Periodogram in biological rhythm research: analysis of incomplete and unequally spaced time-series", *Biological Rhythm Research*, vol. 30, no. 2, pp. 178–201, 1999. doi: 10.1076/brhm.30.2.178.1422