Sequential Rule Analysis of ICU Patient Vital Signals and Alarms

Michela Venturini[1,2] (✉), Len Feremans[3], Wouter De Corte[4], and Celine Vens[1,2]

[1] Department of Public Health and Primary Care, KU Leuven, Campus KULAK, Belgium
[2] ITEC - IMEC, Kortrijk, Belgium
[3] Department of Computer Science, University of Antwerp, Antwerp, Belgium
[4] Department of Anesthesiology and Intensive Care Medicine, AZ Groeninge Hospital, Kortrijk, Belgium

Abstract. In Intensive Care Units (ICUs), excessive medical alarms can cause alarm fatigue and desensitization, compromising patient safety. Alarm management is typically based on manual threshold adjustments, while advanced algorithmic solutions remain underused due to the complexity of patient conditions, dynamic environments, and missing contextual data. Our goal is to investigate the diagnostic utility of combining multiple signals and alarms to enhance relevance and minimize false alarms. A major challenge is integrating heterogeneous data sources, as vital signs are continuously sampled while alarms are event-driven. To bridge this gap, we encode the ICU data into a discretized symbolic representation, reducing dimensionality and improving pattern discovery. We propose a methodology for extracting sequential rules from multivariate datasets, structuring data into a sequence database using a sliding window transformation to capture temporal dependencies. To improve robustness, we introduce a rule ensemble approach, integrating patterns discovered across multiple representations. We applied our method to ICU data from 604 patients, incorporating continuous vital signs and alarm logs. Our findings reveal interpretable sequential rules, analyzed with clinical experts, including patterns highly relevant to intubation events. Our results highlight the potential of data-driven approaches to refine alarm management and improve patient monitoring in critical care settings.

Keywords: Rule Mining, Time Series Analysis, ICU, Alarm fatigue

1 Introduction

An Intensive Care Unit (ICU) is a specialized hospital department dedicated to providing intensive care and treatment to patients who are severely ill or critically injured. ICUs are furnished with advanced medical equipment that allows close monitoring of patients, provision of life support, and prompt intervention to stabilize their condition and prevent further deterioration. Such medical devices produce alarms designed to alert caregivers to any change in the condition of patients. This is particularly important in the ICU, as patients are often physiologically unstable and their condition can change very suddenly, requiring immediate action. However, with increasing patient parameters to be monitored, concerns have been raised about excessive and inappropriate alarm triggers. ICU alarms are a broad term that encompasses several types of alarm, classified into three groups: clinical, technical, and caused by intervention. Clinical alarms are the ones we are interested in and generally rely on a hard coded threshold for each physiological parameter separately, e.g., a "low saturation" alarm is triggered whenever the detected blood oxygen saturation (S_pO_2) falls below 88%. Unfortunately, these fixed alarm settings may not be appropriate for all ICU patients. For example, a patient with a respiratory condition may have a completely different saturation level considered normal compared to a young person admitted to the ICU for trauma. Technical alarms mostly refer to device malfunction or low battery. Finally, alarms caused by intervention are clinical alarms that are not triggered by actual changes in the patient state but rather by external factors (e.g., the patient is being transported or changed position in bed) [15,14]. It has been extensively shown that the proportion of actionable alarms is extremely low ranging from 1% to 26% in adult ICU settings [23], with an average of 771 alarms per bed per day [16], leading to alarm fatigue [5], desensitization to alarms [10], and ultimately impacting the safety of patients [2].

Current literature on managing clinical alarms can be broadly classified into two main strategies: customization of alarm settings and automated algorithms. The former strategy involves tailoring alarm settings based on patient-specific profiles, requiring nurses to manually adjust monitor settings according to the patient's current condition, and it is the most widely adopted in clinical practice. The latter approaches operate on signal acquisition, alarm validation, and alarm generation [13], in order to reduce false alarms. However, a limitation of both these methodologies is their specificity toward a single type of alarm. Others have studied the use of composite alarms, arguing that the correlation of multiple sensors provides a more accurate reflection of the state of a patient than single threshold alarms [12,3]. Despite these advancements, most alarm systems are designed to signal immediate concerns, overlooking longer-term alarm patterns. Moreover, strategies applied in the ICU setting did not show sufficient effectiveness in reducing the number of alarms [11].

Investigating sequential patterns of signals, alarms, and events may provide valuable insights for improving clinical alarms and reducing alarm fatigue. Identifying patterns that occur in close temporal proximity can help assess the usefulness of existing threshold alarms, uncover non-trivial patterns that anticipate meaningful changes in a patient's status, and ultimately avail the introduction of composite alarms that integrate multiple signals for a more accurate representation of the patient's condition. This approach could streamline alarm management, reduce the overall number of alarms, and enhance clinical efficiency. This work retrospectively explores and quantifies sequential rules in vital signs and alarms originating from ICU monitoring devices. Based on previous work, we hypothesize that alarm patterns exist and can be used to reduce the number of alarms [17]. We propose an approach that extracts non-redundant sequential rules from Mixed-Type Multivariate Time Series (MXT-MTS), consisting of multiple vital signs and alarm logs [6]. Our method allows us to identify and visualize such relationships for large volumes of signals and alarm data. We applied our method to previously collected data from the ICU of AZ Groeninge Secondary Care Hospital in Belgium. Our dataset comprises raw physiological signals and alarm logs generated from ICU monitors for hundreds of patients.

2 Sequential rule mining in MTS

2.1 MTS in medical time series

Figure 1 contextualizes the concept of MXT-MTS in the medical field and depicts a realistic scenario of a patient showing signs of respiratory failure. At time T_0, the patient is in a stable state. At time T_1, S_pO_2 drops, triggering an alarm for mild hypoxia (red band). In response, the body reacts by increasing cardiac output through an increase in heart frequency (HF) to compensate for reduced oxygen availability, leading to a tachycardia alarm. If no action is taken, as the cardiovascular system begins to struggle and it is impossible for the body to further increase the cardiac output, at time T_2, the mean arterial blood pressure (ABP_m) gradually decreases but remains within a range that does not yet trigger an alarm (blue band). As stated previously, sequential rules could be used to trigger early interventions before deterioration, and build clinically relevant composite alarms based on multiple signals, potentially less sensitive to false positives.

2.2 Sequential rule mining

Rule mining is a set of analytical techniques used in data mining to uncover interesting patterns, associations, or relationships among a set of items in large

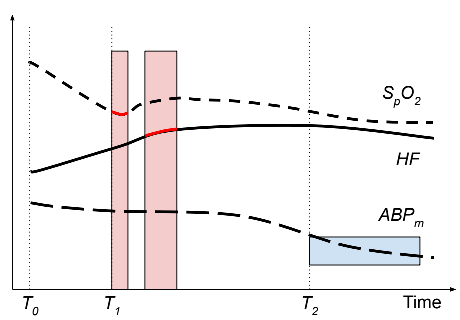

Fig. 1: Example of multivariate time series of a patient with alarms. S_pO_2 oxygen saturation, HF heart frequency, ABP_m arterial blood pressure.

databases [1]. It finds application in several areas, including medicine, where it was used, for example, to extract diagnosis patterns from medical health records [18]. These rules can help identify common co-occurrences of symptoms, laboratory results, or treatments that frequently appear together, aiding in clinical decision-making. Sequential Rule Mining focuses on finding relationships between sequences of events, where the order of occurrence matters. It aims to predict subsequent events based on preceding events [9]. The problem of mining sequential rules common to several sequences is defined as follows [8]. A sequence database T consists of a set of sequences $Q = \{q_1, q_2, ..., q_n\}$ built from a set of items $I = \{i_1, i_2, ..., i_m\}$, where i_j is an atomic element such as an alarm or an event. Each sequence q is an ordered list of itemsets $q = \langle I_1, I_2, ..., I_k\rangle$ such that each itemset I_j is a subset of I, i.e., $I_j \subseteq I$. A sequential rule $X \rightarrow Y$ is defined as a relationship between two itemsets $X, Y \subseteq I$ such that $X \cap Y = \emptyset$ and X, Y are not empty. The interpretation of a rule $X \rightarrow Y$ is that if the items of X occur in some itemsets of a sequence, the items in Y will occur in some itemsets afterwards in the same sequence. Notably, there is no ordering restriction between the items within X (or Y).

2.3 MTS applications

The above definition usually refers to sequences of events of the same type (or of a single time series). Despite its importance, few studies attempted to discover rules in multivariate time series data. Nguyen et al. [22] approached the problem by separately finding rules in each single time series, by using the Apriori algorithm, and then scanning them to find inter-patterns. Park et al. [24] created symbolic baskets from the MTS and then applied the Apriori algorithm to such encoding. Karaca et al. [19] introduced temporal abstractions and applied a modified version of the Prefix-Span algorithm to mine frequent patterns. A prevalent challenge in the applications mentioned above, as well as in mining patterns or rules in MTS generally, is the issue of rule explosion. This phenomenon, where conventional mining algorithms generate an excessive number of redundant rules, complicates the process due to the overwhelming volume of data to analyze. In this study, our focus shifts towards identifying non-redundant sequential rules rather than merely frequent ones. In addition, we explore a methodology that allows for the simultaneous analysis of multiple time series and alarm logs. The usefulness of such approach can easily be motivated by considering the example depicted in Figure 1 where the corresponding sequential rule might look like:

$$(S_pO_2 \downarrow, HF \uparrow) \Rightarrow (ABP_m \downarrow)$$

where $S_pO_2 \downarrow$ is the mild hypoxia alarm, $HF \uparrow$ is the tachycardia alarm, and $ABP_m \downarrow$ indicates that the patient's blood pressure is within a certain range, considered low. A single signal or alarm is insufficient to accurately assess the patient's status. A fluctuation in a single parameter, or an alarm triggered by an isolated change, could simply result from minor variations or patient movement rather than a clinically significant event. In contrast, analyzing multiple signals together provides a more comprehensive view, enabling a deeper understanding of the patient's condition.

3 Our approach

We propose a novel approach for identifying sequential rules within MXT-MTS derived from ICU data and alarm logs. Our approach is based on a representation that combines occurrences of discrete alarms with continuous vital signals discretized in time and value. A discretized representation of time series is crucial as it allows for dimensionality reduction and more efficient data manipulation. Additionally, we need to consolidate different vital signs that are sampled at different fixed rates and alarm logs that are event-driven and recorded only when predefined thresholds are exceeded. The steps of our approach are depicted in Figure 2 and are explained in more detail in the next subsections. In the initial phase, data preprocessing converts the raw multivariate time series data into a symbolic representation in the form of letters, a format suitable for rule discovery, using Symbolic Aggregate approXimation (SAX) [21]. Subsequently, the encodings for each physiological variable and alarm are merged into a unified tabular representation. The tabular representation is then transformed into a sequence database using a sliding window transformation. Next, we use a state-of-the-art method to efficiently extract the top-k non-redundant sequential rules [9]. Our approach depends on several parameters, and we analyze their sensitivity. We propose an ensemble of rules discovered in multiple representations in Section 4.

3.1 Time series representation

SAX is a data-adaptive technique that transforms a time series into a symbolic sequence. It is used to effectively represent time series, as it allows dimensionality reduction of data while maintaining their original characteristics. The SAX procedure comprises three steps, depicted in Figure 2. The first step involves applying Piecewise Aggregate Approximation (PAA) [20] to the original time series. PAA reduces the dimensionality of the time series by dividing it into segments of a predefined length W and representing such segments with their average value. Given the time series $\mathbf{S} = [s_0, s_1, ..., s_N]$, the resulting approximation is $\bar{\mathbf{S}} = [\bar{s}_0, \bar{s}_1, ..., \bar{s}_{N/W}]$, and the i_{th} element $\bar{s}_i$ of $\bar{\mathbf{S}}$ is obtained as follows:

$$\bar{s}_i = \frac{1}{W} \sum_{j=i\times W}^{(i+1)\times W} s_j. \tag{1}$$

W, which in Figure 2 is 2 minutes, significantly affects the level of data compression and fidelity. The second step is the discretization of the time series, which consists of mapping the continuous values obtained from the PAA to an alphabet $\mathcal{A}$ with a given set of discrete symbols. Each average value from the PAA step is then assigned to a symbol based on its value. This mapping is

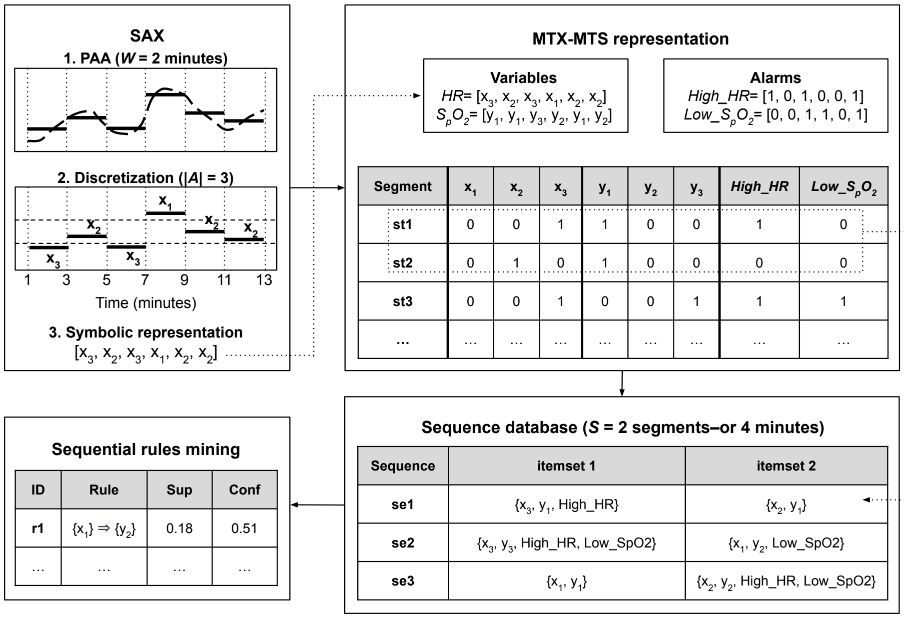

Fig. 2: Overview of our approach. The first step is a modified version of SAX to obtain a symbolic representation of each physiological time series. The second step consists of creating a joined tabular representation of physiological variables and alarms. The third steps consists of translating this representation into a sequence database, the proper input for rule mining (fourth step). The final result is a set of sequential rules. We vary the parameters for preprocessing, i.e. the segment length (W), alphabet size ($|\mathcal{A}|$), and sequence length (S).

performed using breakpoints that divide the distribution of the data into intervals, corresponding to each symbol in the alphabet. The canonical version of SAX partitions data into bins based on the assumption of a normal distribution. However, when data are not normally distributed, this can result in some value ranges being assigned more symbols than others, distorting the data structure. To avoid this, we used quantiles, ensuring that each interval contains the same number of observations. In this way, all parts of the signal contribute equally to the symbolic representation, preventing high-density regions from dominating the encoding and low-density regions from being overlooked. The final step in the SAX process involves converting the discretized values into a symbolic sequence. Each segment's symbol from the discretization step is concatenated to form a symbolic representation of the original time series. The resulting approximation is $\hat{\mathbf{S}} = [\hat{s}_0, \hat{s}_1, ..., \hat{s}_{N/W}]$. The symbolic representation significantly reduces the size and complexity of the data. Within the framework of encoding MTS, the SAX technique is applied individually to each variable. However, the parameters selected for the transformation are kept consistent across all variables. It is worth mentioning that we apply normalization across all patients rather than per patient to ensure that SAX binning captures meaningful patterns at the population level. This allows the sequential pattern mining algorithm to detect trends that are globally relevant rather than patient-specific variations.

3.2 Alarm log representation

The encoding of the alarm logs follows a similar method, without the need for SAX representation. Since alarms occur irregularly and can be triggered multiple times in short intervals, the most reasonable way to reduce the dimensionality of alarm logs is to divide them into segments of a predefined length W and represent each segment using a binary indicator that denotes whether at least one alarm of a given type occurred within that segment. Consider the original alarm logs for a specific alarm type, represented by $\mathbf{Z} = [z_0, z_1, ..., z_N]$, where z_t represents the presence (1) or absence (0) of this alarm type at time t. For each time segment of length W, we denote the presence/absence of alarms, regardless of the number of identical alarm occurrences, obtaining the approximation $\hat{\mathbf{Z}} = [\hat{z}_0, \hat{z}_1, ..., \hat{z}_{N/W}]$.

3.3 MXT-MTS representation

Given our objective to analyze sequential rules in both alarms and patterns collectively, we need a unified encoding that represents all signals and alarms. Upon deriving the SAX representation for each variable individually, we transform the MTS into a unified representation. For each variable, a dummy version of the SAX representation is generated as a matrix, with rows representing time segments and columns indicating symbols. This approach facilitates the encoding of the presence (1) or absence (0) of symbols in each time segment, thereby enabling straightforward concatenation of multiple variables and alarms into a symbolized MXT-MTS (sMXT-MTS). The resulting representation consists of $\frac{N}{W}$ rows and $|M_v| \cdot |\mathcal{A}| + |M_a|$ columns, where M_v is the set of variables, $\mathcal{A}$ is the alphabet, and M_a is the set of alarm types. In Figure 2, these sets are defined as $M_v = \{HF, S_pO_2\}$, $\mathcal{A} = \{x_1, x_2, x_3, y_1, y_2, y_3\}$, and $M_a = \{High\_HF, Low\_S_pO_2\}$. The notation $|\cdot|$ symbolizes the cardinality of a set. This representation is an intermediate step to summarize the data in intervals and reduce dimensionality. The following section illustrates the steps to obtain the final data format.

3.4 Encoding MTS and alarm logs into a sequence database

Sequential rule mining algorithms require a sequence database for input. As such, we need to convert the sMXT-MTS to a proper representation. The sequence database is organized temporally, with each row corresponding to a sequence of length S and containing itemsets representing subsequent segments of length W. Each itemset contains symbols of the sMXT-MTS present in the time frame of the corresponding segment. Within each itemset, symbols are treated as concurrent, lacking temporal precedence, yet the sequence of itemsets is crucial, as their order is instrumental in identifying sequential rules. Figure 2 shows the resulting sequence database. In the example, $S = 2$ segments (or 4 minutes), so we obtain sequences of two itemsets (representing 2 segments), where the events of itemset 1 happen before those of itemset 2, and each itemset can contain information about multiple signals (HF, S_pO_2) and alarms ($High\_HF$, $Low\_S_pO_2$). Intuitively, if the original data contain signals and events from multiple individuals, the final sequence database contains multiple sequences for each individual, where the number of sequences per individual depends on the number of recorded time points and S.

3.5 Sequential rule mining

We use a sequential rule mining technique that focuses on mining the top-K most frequent occurring non-redundant rules. This reduces the risk of rule explosion and enables control of the number of generated sequential rules. First, we define two representative measures to evaluate the candidate rules.

Definition 1. *The support of a rule $X \rightarrow Y$ is the relative frequency of the co-occurrence of X and Y, and is calculated by dividing the number of transactions containing both X and Y by the total number of transactions $|T|$.*

$$sup(X \rightarrow Y) = \frac{|\{q = [I_1, I_2, \ldots, I_n] \in T \mid X \subseteq I_{k_1} \wedge Y \subseteq I_{k_2} \wedge k_1 < k_2\}|}{|T|} \quad (2)$$

Definition 2. *The confidence of a rule measures the likelihood of the occurrence of the consequent event Y in all transactions that contain the antecedent event X.*

$$conf(X \rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)}. \quad (3)$$

To discover sequential rules, we used TNS [9], an algorithm for discovering the top-K non-redundant sequential rules appearing in a sequence database. The problems of top-K sequential rule mining and redundancy are defined as follows.

Definition 3. *A rule mining algorithm discovers a set $\mathcal{L}$ containing K rules in transaction database $\mathcal{D}$ such that:*

$$\begin{aligned} &\forall r \in \mathcal{L} : conf(r) \geq minconf \\ &\forall r \in \mathcal{L} : \nexists s \in \mathcal{L} : conf(s) \geq minconf \wedge sup(s) > sup(r). \end{aligned} \quad (4)$$

This is in contrast to most rule mining algorithms that require that the user set a minimum support threshold parameter that is hard to set, i.e., usually users set it by trial and error. In addition, we avoid many redundant rules, with potentially thousands of variations of rules having the same support and confidence. The following definition eliminates redundancy in results by keeping similar rules that have a smaller antecedent and a larger consequent.

Definition 4. *A rule $r_a : X \rightarrow Y$ is redundant with respect to another rule $r_b : X_1 \rightarrow Y_1$ if and only if $conf(r_a) = conf(r_b) \wedge sup(r_a) = sup(r_b) \wedge X_1 \subseteq X \wedge Y \subseteq Y_1$.*

The TNS algorithm is based on a depth-first search procedure. TNS is an approximate algorithm that generates non-redundant rules, which might not always be the exact top-K non-redundant rules. TNS depends on a parameter δ, which can be used to improve the chance that the result is exact (the higher the delta value, the higher the chance that the result will be exact), i.e., it allows to make a trade-off between accuracy and runtime performance. In our setting δ was left to default, in all experiments approximate rule mining took less than 1 hour.

4 Case study: multivariate time series and alarm logs form ICU

4.1 Data collection

We collected data from Philips hospital monitors installed in the Intensive Care Unit (ICU) of AZ Groeninge Secondary Care Hospital in Kortrijk, Belgium, over a seven month period in 2021, for more than 1000 patients. The hospital server stored all monitor signals and alarm logs from bedside patient monitors with a sampling rate of 0.2 Hz. These logs contained the timestamp, ID, alarm category, and nature of the alarm. As for the monitoring signals, we focused on Heart Frequency (HF), Respiratory Frequency (RF), systolic, diastolic, and mean Arterial Blood Pressure (ABP_s, ABP_d, ABP_m), end-tidal carbon dioxide ($etCO_2$), and oxygen saturation (S_pO_2). The alarm categories belong to physiological or technical monitor alarms. Physiological alarms included mainly threshold alarms such as $High\_HR$ or $Low\_S_pO_2$. The physiological thresholds for generating these alarms could be a standard set of values or modified by the medical team based on the specific clinical profile of the patients, but this information was not available to us. The medical data used in this study are confidential and cannot be shared due to privacy regulations.

4.2 Data preprocessing

We sampled signals and alarm logs with a frequency of 2 measurements per minute (0.033 Hz). Furthermore, we excluded patients with less than 6 hours of recording, patients with less than 5 variables recorded among HF, RF, ABP_s, ABP_d, ABP_m, $etCO_2$ and SpO_2. Moreover, since we were specifically interested in alarms, we excluded patients without alarms recorded, and included only time frames within five minutes before and after the alarm. Finally, we excluded all alarms that occurred in the same time frame as technical alarms to exclude potential biases in the discovered rules. Since sequential rule mining techniques naturally handle missing values, and all signals had a percentage of missing values lower than 6% (except $etCO2$), we did not impute the data set. We decided to binarize the variable $etCO2$ in present/absent as it is an indicator of intubation.

The final cohort comprised 604 patients, 7 signals, and 31 alarm types, totaling more than 3700 hours of recorded data, 16 million data points, and 3 million alarms.

4.3 Discovering sequential rules

We applied our method to the preprocessed dataset. We run the experiments in Python 3.10 on a laptop with a 2.60 GHz 6 cores CPU with 32 GB of memory. The operating system used in this machine is Ubuntu 20.04.6 LTS. We used the TNS Java implementation from the SPMF library [7][5], available through a Python wrapper[6]. The code is available upon request to the corresponding author.

Hyper-parameters Table 1 lists the parameters for preprocessing and sequential rule mining. We used all parameter combinations to perform a sensitivity analysis and extract different rule variants. The size of the alphabet $|\mathcal{A}|$ determines the granularity of the representation of time series. The length of the sequence S affects the capacity to capture long-term dependencies within the data. Longer sequences work as upper bound for identifying patterns, allowing us to identify both patterns close to each other and patterns that span in a longer time period. The choice of segment length W is important to capture the trend of the signal correctly. W is also relevant to the representation of alarm logs, as often alarms of the same type occur in clusters, close to each other, and choosing a small W would capture redundant patterns. For sequential rule mining we use default values for min_conf and K.

Sensitivity analysis parameters There is an inherent trade-off between timeliness and accuracy when selecting S. Although increasing S increases confidence, it can reduce the clinical utility of the rules discovered. For example, an algorithm might detect events that repeat in the same order over a 24-hour sequence,

5 http://www.philippe-fournier-viger/spmf/
6 https://pypi.org/project/spmf/

Table 1: Parameter ranges for pre-processing. For sequential rule mining we use default values.

Parameter	Description	Value(s)
$\|\mathcal{A}\|$	Alphabet size for SAX	$[3, 5, 7]$
W	Segment length for SAX and alarm grouping (minutes)	$[1, 3, 5]$
S	Length sliding window (minutes)	$[20, 40, 60]$
min_conf	Minimum confidence threshold	0.5
K	# of sequential rules	1000

but their apparent co-occurrence might be due to their high prevalence in the dataset, rather than a true temporal relationship. As expected, the sensitivity analysis revealed that the confidence was proportional to S and inversely proportional to W. Furthermore, the effect of $|\mathcal{A}|$ is not negligible. With increasing alphabet size, the number of observed patterns containing alarms increases and confidence decreases.

4.4 Rule interpretation

We extracted 1000 rules for each combination of parameter values. To evaluate the extracted rules, we selected the 50 rules with highest support and directly inspected them. We report a selection of both trivial and interesting rules and discuss their medical relevance in Table 2. First, we discuss our ensemble approach, since the reported rules are discovered using different representations of the data, i.e. using a combination of values for preprocessing parameters as shown in Table 1.

Ensemble of sequential rules We use multiple parameter values for preprocessing, thereby reporting rules spanning both short and long periods and considering different granularities of physiological variables. That is, we mined sequential rules using all combinations of $|\mathcal{A}|$, W and S, and aggregated the resulting rules. It is important to note that multiple rules or similar variations were found for different combinations of the parameters we tested. Intuitively, the support and confidence changed slightly as the parameters affected the itemset construction. We discuss the effect of each parameter independently:

- By varying S we discover sequential rules in a varying time horizon. Existing sequential rule mining algorithms enable a temporal constraint that is enforced on the maximal duration of a rule [4]. In contrast, by varying S we enable domain-experts to inspect both short-term and long-term rules that are of interest. Moreover, we can filter redundant rules automatically, i.e., rules discovered in a short window are also discovered using a longer window, while the opposite is not true. In our case we discover rules where the gap between the condition and consequent event is at most 60 minutes.
- The choice of binning strategy for continuous signals is determined by $|\mathcal{A}|$ and has a direct impact on the extracted rules in two key ways. First, it influenced the granularity and relevance of the rules: using a coarse binning (e.g., 3 bins for heart frequency) captures broad trends but may miss finer physiological variations, whereas a finer binning (e.g., 7 bins) allows for more specific conditions but increases rule set complexity. Second, binning affected support and confidence: rules derived from higher granularity bins tend to be more specific and potentially more clinically meaningful, but they also become rarer in the dataset, leading to lower support and confidence. These rules may apply only to specific subgroups of patients or reflect less frequent but highly relevant clinical conditions.

– W determines the smoothing of continuous vital signs and has a similar effect on granularity to $|\mathcal{A}|$ but in the time domain. With lower values of W we detect peaks and sudden changes, while with higher values we focus on increasing and decreasing trends. Using W_{min}, i.e. 1 minute, we take the average of 2 values, while using W_{max} we take the average of 10 raw time series measurements.

Table 2: Selection of trivial and rules of interest for the intubation alarm. We run our method with different combinations of $|\mathcal{A}|$, W, and S. Event codes are HF (heart frequency), RF (respiratory frequency), and systolic, diastolic and mean arterial blood pressure (ABP_s, ABP_d, ABP_m). Conf. confidence, Sup. support.

ID	Rule	Conf.	Sup.
(a)	$\{HF\_low\} \rightarrow \{HF < 60\}$	0.62	0.11
(b)	$\{ABP_s < 100\} \rightarrow \{ABP_s\_low\}$	0.61	0.30
(c)	$\{S_pO_2 < 91\} \rightarrow \{S_pO_2\_low\}$	0.57	0.30
(d)	$\{ABP_m > 93, ABP_s > 156\} \rightarrow \{ABP_s\_high\}$	0.50	0.13
(e)	$\{HF < 60, 28 < RF < 31\} \rightarrow \{intubation\}$	0.55	0.02
(f)	$\{74 < HF < 86, RF > 30\} \rightarrow \{intubation\}$	0.53	0.03
(g)	$\{ABP_d < 47, RF > 30\} \rightarrow \{intubation\}$	0.52	0.05
(h)	$\{ABP_d < 45, ABP_m < 64, RF > 31\} \rightarrow \{intubation\}$	0.51	0.02
(i)	$\{intubation, ABP_d < 45,\ 91 < S_pO_2 < 93\} \rightarrow \{ABP_m < 64\}$	0.51	0.02

We make a distinction between medically trivial and interesting rules, specific to intubation, and provide the following interpretation explaining their prevalence and medical significance.

Trivial rules Rules (b), (c), and (d) are trivial, as they reflect the standard response to vital signs that cross a predefined threshold and are commonly embedded in the ICU monitor, to alert physicians to possible clinical deterioration. The low confidence can be explained by two main reasons: the possibility for physicians to manually change the threshold to make it more suitable for specific patient profiles and the possibility to turn off an alarm if it is not needed. Rule (a) depicts a scenario where the alarm goes off first and subsequently the vital sign is still low. Rules such as this one were found relatively often, and although they might appear counterintuitive, multiple reasons could contribute to their presence with a relatively high confidence. First, the vital sign can still vary after the alarm goes off, and even if doctors take action, it might take time before the value returns to normal ranges. Moreover, the threshold can be set differently after the alarm goes off or the alarm can be switched off.

Interesting rules In addition, we inspected patterns that include *intubation* for mechanical ventilation, which was chosen as an event because it is one of the highest-risk procedures in the ICU and was straightforward to extrapolate from the data. Intubation is an intervention performed when a patient cannot maintain adequate gas exchange and it can be planned or unforeseen. In the latter case, it can lead to negative consequences for the patient and ultimately to increased mortality if not performed in a timely manner. Intubation is usually required in cases of acute respiratory failure, shock, or neurological deterioration. The scenarios that can lead to intubation can be, for example, severe hypoxia (low S_pO_2) in patients with worsening oxygenation despite oxygen therapy; hypercapnic respiratory failure (elevated CO_2 levels and high RF) in conditions such as COPD exacerbation; cardiogenic or septic shock, usually accompanied by low ABP in addition to elevated HF as a compensatory response to hypoxia, hypotension, or metabolic acidosis. Rules (e) to (i) were selected and assessed to determine whether our method could extract clinically plausible patterns and thus hold potential utility for future applications. Rule (e) associates intubation with a combination of bradycardia and mild tachypnea, which could indicate early respiratory failure. Rule (g) indicates hypotension in combination with tachypnea, which could be related to shock or respiratory distress. Both rules (e) and (g) could be improved in clinical relevance and usefulness if included information regarding blood gases, which were not available in the dataset, or oxygen saturation (S_pO_2). Rule (f) describes patients with normal HF and moderate tachypnea and is too broad to be useful to predict intubation. Rule (h) indicates more severe hypotension and tachypnea, which could be representative of septic and cardiogenic shock with respiratory distress and is associated with a higher likelihood of intubation. Finally, rule (i) suggests that intubated patients have a low ABP_m when they are still mildly hypoxic, but it is an observational rule rather than a predictive rule. Surprisingly, patterns related to the event of intubation lack oxygeneation in favor of HF, RF and ABP. Moreover, due to limitations related to available data, we missed important markers such as $PaCO_2$, pH, and information about neurological condition such as the Glasgow coma scale.

Sequential rule clusters Finally, we examined how frequently sequential rules (without and with alarms) appeared within individual patients and across multiple patients. To achieve this, we first extracted the top-50 rules, obtained with constant $|\mathcal{A}| = 7$, while exploring all the possible combinations of parameter values for W and S. To simplify the calculation of rule occurrences per patient and across patients, we focused on a single parameter configuration ($W = 5$, $S = 60$), ensuring consistency in how the patients were encoded into a sequential database. For each rule, we counted the number of patients where the rule appeared at least once, and, for these patients only, also the average number of times the rule appeared in each patient. Figure 3 shows the distribution of the rules in terms of the number of patients and the average number of occurrences per patient. The figure allows to distinguish "general" and "cluster-specific" rules, where the former apply to the majority of the population but are less specific, while the latter apply to specific sub-populations. The presence of "cluster-specific" rules suggests that patients might be categorized into subgroups with different temporal patterns, potentially relevant for clinical decision support. In our setting, rules exclusively containing signals are more general, while rules containing alarms tend to appear in fewer patients but with a higher number of occurrences per patient, and might be of interest to personalize alarm settings to specific patient profiles.

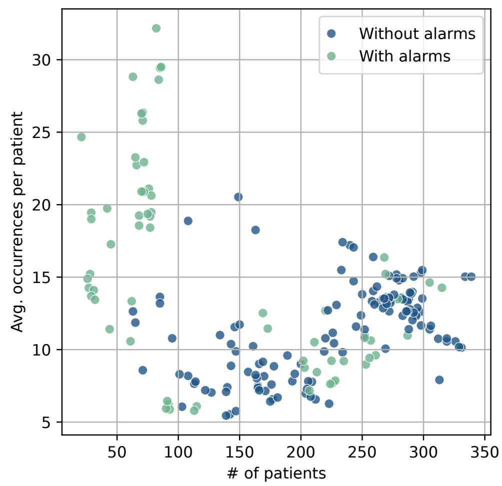

Fig. 3: Each dot represents a rule. The x-axis represents the number of patients for which the sequential rule holds, and the y-axis represents the average rule occurrences within each of these patients.

5 Limitations

Despite the potential usefulness of our framework and the relevance of the application, we must acknowledge multiple limitations of this work. First, ICU monitors generate high-frequency data that are susceptible to sensor artifacts and noise. Although the data were smoothed during preprocessing, this may not have been sufficient to prevent outliers from affecting the extracted rules. Moreover, although one of our core motivations was to address the problem of false alarms, our dataset could not be linked to clinically verified alarm annotations. Thus, we were unable to quantify false alarms directly. In addition, the framework was applied to a retrospective single-center cohort of patients from a Belgian hospital, limiting its generalizability to other ICU settings with different monitoring systems, patient populations, and clinical protocols. Due to privacy constraints, we are currently unable to publicly release the dataset. While we filtered rules using support and confidence thresholds, their interpretation still relied on expert review. Furthermore, although our approach identifies frequent and clinically plausible sequential rules, it does not establish causal relationships between physiological signals and alarms or events. Finally, the nature of the alarms based on thresholds that are predefined but easily changed by clinicians to fit the patient profile, could introduce biases into the discovered alarms. Future work should include the use of multiple cohorts of patients from different hospitals. Efforts should also focus on linking signals and alarms with clinically valid annotations to enable verification of false alarms and to quantify the usefulness of the proposed rules. In addition, future studies should investigate the use of additional interestingness metrics and domain-specific constraints to automatically prioritize informative rules.

6 Conclusions

In this work, we introduce a novel framework for mining patterns in multivariate ICU signals and alarms, providing a data-driven approach to understanding alarm triggers and physiological trends. By applying the framework to a case study from a Belgian hospital, we demonstrate its potential to extract clinically meaningful patterns that may help reduce alarm fatigue, optimize the early warning system, and support the decision-making process in the ICU. The extracted patterns are consistent with clinical expectations, but also highlight potential limitations of current ICU alarm systems, such as false positive alarms and threshold-based dependencies. This work serves as a basis for further refinement of ICU alarm systems by integrating multiple signals and creating alarms that provide a meaningful indication of the overall state of the patient.

Acknowledgments M.V. and L.F. are funded through the Research Fund Flanders (project G0A2120N and grant 12B0V24N, respectively). The authors also acknowledge the Flemish Government (AI Research Program).

Declaration of competing interest The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Agrawal, R.: Fast algorithms for mining association rules. VLDB (1994)
2. Alsuyayfi, S., Alanazi, A.: Impact of clinical alarms on patient safety from nurses' perspective. Informatics in Medicine Unlocked **32**, 101047 (2022)
3. Bitan, Y., O'Connor, M.F.: Correlating data from different sensors to increase the positive predictive value of alarms: an empiric assessment. F1000Research **1** (2012)
4. Cule, B., Feremans, L., Goethals, B.: Efficiently mining cohesion-based patterns and rules in event sequences. Data Mining and Knowledge Discovery **33**(4), 1125–1182 (2019)

5. Cvach, M.: Monitor alarm fatigue: an integrative review. Biomedical instrumentation & technology **46**(4), 268–277 (2012)
6. Feremans, L., Vercruyssen, V., Cule, B., Meert, W., Goethals, B.: Pattern-based anomaly detection in mixed-type time series. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I. pp. 240–256. Springer (2020)
7. Fournier-Viger, P., Lin, J.C.W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., Lam, H.T.: The spmf open-source data mining library version 2. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part III 16. pp. 36–40. Springer (2016)
8. Fournier-Viger, P., Nkambou, R., Tseng, V.S.M.: Rulegrowth: mining sequential rules common to several sequences by pattern-growth. In: Proceedings of the 2011 ACM symposium on applied computing. pp. 956–961 (2011)
9. Fournier-Viger, P., Tseng, V.S.: Tns: mining top-k non-redundant sequential rules. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. pp. 164–166 (2013)
10. Frankenthal, H., Ben Shlomo, I., Kurzweil Segev, Y., Bubil, I., K, A., Orkin, D., Kobo Greenhut, A.: Perceived reliability of medical device alarms—a major determinant of medical errors driven by frozen medical thinking. International Journal for Quality in Health Care **34**(1), mzac009 (2022)
11. Gul, G., Intepeler, S.S., Bektas, M.: The effect of interventions made in intensive care units to reduce alarms: A systematic review and meta-analysis study. Intensive and Critical Care Nursing **75**, 103375 (2023)
12. Hever, G., Cohen, L., O'Connor, M.F., Matot, I., Lerner, B., Bitan, Y.: Machine learning applied to multi-sensor information to reduce false alarm rate in the icu. Journal of clinical monitoring and computing **34**, 339–352 (2020)
13. Huo, J., Wung, S., Roveda, J., Li, A.: Reducing false alarms in intensive care units: A scoping review. Exploratory Research and Hypothesis in Medicine **8**(1), 57–64 (2023)
14. Imhoff, M., Kuhls, S.: Alarm algorithms in critical care monitoring. Anesthesia & Analgesia **102**(5), 1525–1537 (2006)
15. Imhoff, M., Kuhls, S., Gather, U., Fried, R.: Smart alarms from medical devices in the or and icu. Best Practice & Research Clinical Anaesthesiology **23**(1), 39–50 (2009)
16. Jones, K.: Alarm fatigue a top patient safety hazard (2014)
17. Joshi, R., van Pul, C., Atallah, L., Feijs, L., Van Huffel, S., Andriessen, P.: Pattern discovery in critical alarms originating from neonates under intensive care. Physiol Meas **37**(4), 564–579 (Mar 2016)
18. Kang, S.M., Wagacha, P.W.: Extracting diagnosis patterns in electronic medical records using association rule mining. International Journal of Computer Applications **108**(15) (2014)
19. Karaca, M., Alvarado, M.M., Gahrooei, M.R., Bihorac, A., Pardalos, P.M.: Frequent pattern mining from multivariate time series data. Expert Systems with Applications **194**, 116435 (2022)
20. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and information Systems **3**, 263–286 (2001)
21. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. Data Mining and knowledge discovery **15**, 107–144 (2007)

22. Nguyen, D.T., Khuat, B.D.L.: Discovery of temporal association rules in multivariate time series. International Journal of Applied Engineering Research **14**(1), 79–84 (2019)
23. Paine, C.W., Goel, V.V., Ely, E., Stave, C.D., Stemler, S., Zander, M., Bonafide, C.P.: Systematic review of physiologic monitor alarm characteristics and pragmatic interventions to reduce alarm frequency. Journal of hospital medicine **11**(2), 136–144 (2016)
24. Park, H., Jung, J.Y.: Sax-arm: Deviant event pattern discovery from multivariate time series using symbolic aggregate approximation and association rule mining. Expert Systems with Applications **141**, 112950 (2020)