A practical approach to storage and retrieval of high-frequency physiological signals

Andrew J Goodwin\textsuperscript{1,2}, Danny Eytan\textsuperscript{1,5}, Robert W Greer\textsuperscript{1}, Mjaye Mazwi\textsuperscript{1,4}, Anirudh Thommandram, Sebastian D Goodfellow\textsuperscript{1,8}, Azadeh Assadi\textsuperscript{1,7}, Anusha Jegatheeswaran\textsuperscript{3} and Peter C Laussen\textsuperscript{1,6}

This work was funded David and Stacey Cynamon Chair in Pediatric Critical Care at the The Hospital for Sick Children.

\textsuperscript{1}Department of Critical Care Medicine, The Hospital for Sick Children, Toronto, Ontario, Canada
\textsuperscript{2}School of Biomedical Engineering, University of Sydney, Sydney, New South Wales, Australia
\textsuperscript{3}Division of Cardiovascular Surgery, Department of Surgery, The Hospital for Sick Children, Toronto, Ontario, Canada
\textsuperscript{4}Department of Paediatrics, University of Toronto, Toronto, Ontario, Canada
\textsuperscript{5}Department of Medicine, Technion, Haifa, Israel
\textsuperscript{6}Department of Anaesthesia, University of Toronto, Toronto, Ontario, Canada
\textsuperscript{7}Institute of Biomaterial and Biomedical Engineering, University of Toronto, Toronto, Ontario, Canada
\textsuperscript{8}Department of Civil and Mineral Engineering, University of Toronto, Toronto, Ontario, Canada

Keywords: physiology, waveforms, timeseries, compression, indexing

Abstract

Objective: Storage of physiological waveform data for retrospective analysis presents significant challenges. Resultant data can be very large, and therefore becomes expensive to store and complicated to manage. Traditional database approaches are not appropriate for large scale storage of physiological waveforms. Our goal was to apply modern time series compression and indexing techniques to the problem of physiological waveform storage and retrieval. Approach: We deployed a vendor-agnostic data collection system and developed domain-specific compression approaches that allowed long term storage of physiological waveform data and other associated clinical and medical device data. The database (called AtriumDB) also facilitates rapid retrieval of retrospective data for high-performance computing and machine learning applications. Main results: A prototype system has been recording data in a 42-bed pediatric critical care unit at The Hospital for Sick Children in Toronto, Ontario since February 2016. As of December 2019, the database contains over 720,000 patient-hours of data collected from over 5300 patients, all with complete waveform capture. One year of full resolution physiological waveform storage from this 42-bed unit can be losslessly compressed and stored in less than 300 GB of disk space. Retrospective data can be delivered to analytical applications at a rate of up to 50 million time-value pairs per second. Significance: Stored data are not pre-processed or filtered. Having access to a large retrospective dataset with realistic artefacts lends itself to the process of anomaly discovery and understanding. Retrospective data can be replayed to simulate a realistic streaming data environment where analytical tools can be rapidly tested at scale.

1. Introduction

A large amount of heterogeneous data is generated as part of the process of patient care. Such multimodal data streams are integrated and utilized particularly by clinicians in the intensive care environment to understand patients’ clinical condition and response to treatment (Rush et al 2018, Charlton et al 2016, Maslove et al 2017). There has been an increase in research approaches that incorporate these kind of data into analysis (Johnson et al 2016, Orphanidou 2019, Yang et al 2019). One of these data types is physiological waveform data, which is generated by interaction of biological systems with technological devices. These information rich streams are sampled and displayed by medical devices and typically sampled at frequencies up to 500 Hz (Brossier et al 2018, Jacono et al 2010). These waveforms are produced, for example, by continuously measuring mechanical pressures within the arteries as an arterial blood pressure (ABP) waveform (McGhee and Bridges 2002). Other examples of waveform data collected in an intensive care
Physiological waveforms are continuously monitored in environments such as the intensive care unit where their density, continuity and granularity provide an optimal means of longitudinally monitoring patients at risk of sudden or unpredictable deterioration. These individual waveform signals contain information which can facilitate diagnosis. For example, a large number of quantitative physiological descriptors can be extracted from an ECG waveform including heart rate (HR) and heart rate variability (HRV) (Holder et al 2015, Acharya et al 2006, Stein 2013). Also, the waveform morphology itself can be used to diagnose an abnormal heart rhythm (Hannun et al 2019, Goodfellow et al 2018). The interactions between these signals can also shed light on coupled physiological subsystems that have the potential to modify each other’s behavior in a clinically relevant manner. An example of this coupling of physiological subsystems are cardiopulmonary interactions that measure interactions between the lungs and the cardiovascular system (Kiviniemi et al 2010, Penzel et al 2016, Novak et al 1993, Clark et al 2011).

Despite the potential utility for research and clinical purposes, storage of physiological waveform data for retrospective analysis presents significant challenges. Ironically, the complexity and granularity that makes this such an appealing data type for analysis also means that the resultant data can be very large, and therefore becomes expensive to store and complicated to manage (Burykin et al 2011). Additionally, access to data may be inhibited by industry vendors who may view data as proprietary information rather than understanding it to be the property of the patient (Wetzel 2018, Cosgriff et al 2019). Thus, collating archives of usable information in a clinical setting where comprehensive data collection is difficult presents a significant challenge for database administrators (Rodriguez et al 2018, Karhonen et al 2001).

One solution to the problem of storage is to keep the data for only a certain period and then purge. The difficulty is that these datasets are incomplete as they contain high resolution data only for recent admissions. Another approach for physiological signals is to store variables derived from waveforms, for example heart rate derived from ECG, or respiratory rate derived from the respiratory impedance (RI) waveform (Holder et al 2015). The rationale behind this approach is that derived data values tend to be relatively low frequency (0.0003 Hz–1 Hz) (Brossier et al 2018), resulting in significantly lower storage requirements than if the original waveforms had also been retained. However, these systems do not permanently retain the raw waveforms from which they were derived and therefore implicitly assume that all the features of the waveform are known at the time of extraction. Additionally, data stored at lower temporal resolutions may not adequately capture the variability of the underlying signal (Eytan et al 2019, Horvat and Ogoe 2019).

Derived variables may also be irreversibly affected by the algorithms that generate them, for example limitations of the inbuilt beat detection algorithms used in patient monitors may momentarily affect reported heart rate values (Berntson and Stowell 1998), or artefacts in an ABP waveform may momentarily affect the derived diastolic and systolic blood pressure readings (Li et al 2009).

Our research is predicated on the premise that there is additional information contained in the original waveform data beyond the standard suite of derived variables. Furthermore, analytical approaches will likely benefit from more robust approaches to the extraction of familiar variables, for example re-deriving heart rate from ECG waveforms (Manikandan and Soman 2012). However, in order to explore these new techniques, the waveform data must be retained at its original sample rate for future exploration.

Continuous high-frequency physiologic signals are characterized by being large in volume and high in variability. In the department of Critical Care Medicine at the Hospital for Sick Children (HSC), Toronto, Ontario, Canada, we use these signals for research with the goal to develop tools for real-time and online analysis available to clinicians at the bedside. To achieve this we first made the central philosophical decision that all of the physiologic data and waveforms are owned by the patient, not by the monitoring vendor or hospital. This meant that we capture and store these data permanently, and that the format and data access had to be such that it was readily discoverable (Lawrence 2017, Gil et al 2007). This is a unique approach to take, and in this report we describe the bespoke architecture that was developed to solve the problems of managing large volume time series physiologic data.

1.1. Previous work in this field

A variety of proprietary middle-ware platforms exist to collect and aggregate data from patient monitors and other devices, e.g. ViNES (Baxter, Deerfield, IL), Capsule SmartLinx (Capsule, Andover, MA), Bedmaster EX (Excel Medical Electronics, Jupiter, FL, USA), Bernoulli data collection system (Capsule Technologies Inc. Andover, MA) and iXellence (iXitos, Berlin).

Numerous database systems and approaches have been developed specifically for storage of time series data (Bader et al 2017, Jensen et al 2017). Each of these systems is optimized for data with particular
characteristics, for example floating-point or integer values; regularly sampled time series or aperiodic data; and long term vs short-term storage (Pelkonen et al 2015, Lautenschlager et al 2017, Yang et al 2014, Andersen and Culler 2016, Blalock et al 2018, InfluxDB 2019). Some time series data formats have been developed specifically with physiological data in mind (Brinkmann et al 2009, Moody 2019), Cabeleira et al 2018, Lee and Jung 2018, Trigo et al 2011, Bond et al 2011, Chen et al 2017.

Other groups have leveraged a wide range of approaches to large scale storage of physiological waveform data, including ASCII files (Goldstein et al 2003, Rehm et al 2017, Liu et al 2012, Fanelli et al 2018), IBM DB2 relational database (Blount et al 2010), compressed binary data in floating-point format (Lee and Jung 2018, Liu et al 2018), compressed .wav files (Blum et al 2015), Microsoft SQL Server (Matam et al 2018), compressed Apache Avro (McPadden et al 2019, Vohra 2016), Hadoop (McPadden et al 2019, Sahoo et al 2014), MongoDB (Sutton et al 2019), compressed JSON files (Baljak et al 2018), and in proprietary formats such as TrendFace (iXitos, Austria) (Burykin et al 2011), Bernoulli (Cannesson et al 2019), and BedMaster Ex (Bonnici et al 2015, Hu et al 2017).

1.2. Our approach

In 2015, when initially faced with the challenge of collecting waveform data from the Critical Care Unit at the HSC, many of these technologies and systems were unavailable, immature, or not otherwise suitable for our particular situation (De Georgia et al 2015). Our goal was to apply modern time series compression approaches to the problem of physiological waveform storage and retrieval.

As part of the design process we established that an ideal physiological waveform data database would have the following characteristics.

(a) **Device Agnostic Data Collection**—Signals should be able to be collected from a variety of devices, platforms, and vendors such that data may be synchronized and integrated into a unified collection.

(b) **Compact Representation**—The ability should exist to compress waveform data to a small size that is cost-effective to store long term.

(c) **High Fidelity**—Information should be stored directly to disk exactly as supplied from the monitors/sensors, with no loss of information.

(d) **Long Term Storage**—Stored data should be retained indefinitely.

(e) **Indexing**—A structure should be imposed that permits creation of metadata that facilitates quality control and research cohort identification without the need to access the raw data directly.

(f) **Efficient Retrieval**—Data should be able to be retrieved rapidly for use in analysis and discovery. Programmatic data retrieval should be enabled and optimized for parallel and distributed computing applications.

We set out to design and deploy a physiological data collection system that exhibited all the above characteristics. We named the system AtriumDB™ as a reference to the atrium of the heart and the large open area at the Elizabeth Street entrance of the HSC.

2. Methods

2.1. Overview

Patients in a critical care setting are heavily monitored by a wide range of electronic sensors and devices (Rodriguez et al 2018, Zaleski 2015). These devices are primarily designed to display clinically relevant variables in real time to carers at the bedside. Secondary systems may be deployed to collect and store the data streams generated by such devices for later use. Such data collection systems consist of a device integration stage which collect data streams from bedside monitors and other devices and aggregate them into a unified data stream (Karhonen et al 2001). These aggregated data streams are often temporarily stored in an uncompressed text format for readability and simplicity; however such representations contain redundant information which can easily be compressed or removed. Efficient data storage requires data compression to reduce the size of the data and minimize data storage costs. While compressed data is compact, it may require specialized software to make it available for clinical and research purposes. This paper focuses primarily on the data compression, storage, and retrieval functionality provided by the AtriumDB system.

2.2. Device integration

We deployed ViNES Device Integration Software connected to 42 Philips monitors (Intellivue MP70, Philips, Amsterdam, Netherlands, software version J.10.50) in the critical care unit at the Hospital for Sick Children. Ancillary devices, such as ventilators, and regional saturation monitors (near-infrared saturations, or NIRS), were interfaced directly to the Philips monitors using the Intellibridge system. With this approach the
monitor acts as an aggregator of data and allows simultaneous and synchronized export of both monitor data and information from these peripheral devices. The ViNES system software uses the Philips Data Export Protocol (Philips 2015) to request data from the RJ45 serial port at the back of each monitor, transmitting it to the ViNES server via a Lantronix Serial to Ethernet adapter. The ViNES server unifies the signals, converts the data to JavaScript Object Notation (JSON) format, and stores it in a RabbitMQ (RabbitMQ 2019) cache for retrieval by downstream software. Non-waveform physiological data are retrieved from the Philips Central Server via HL7 feeds by the Trajectory, Tracking, and Trigger (T3™) Platform (Etiometry, Boston, MA). An overview of the system architecture is shown in figure 1.

2.3. Data collection
A service that polls the ViNES Rabbit Queue and retrieves the individual JSON messages was written in the PHP scripting language (PHP: Hypertext Preprocessor 2019). PHP was initially chosen because several of the team members were familiar with this programming language which enabled rapid prototyping. Each monitoring device would emit one messages containing multiple numeric data values for heart rate (HR), blood pressure (BP) etc every 1024 ms\(^9\), and one message for every 256 ms of each unique waveform signal. Each of the waveform messages (emitted once every 256 ms) contains between 32 and 128 values depending on the waveform sample frequency, and each numeric data message (emitted once every 1024 ms) contains multiple parameters such as heart rate, blood pressure, and respiratory rate. A typical throughput of the system in our 42-bed installation was around 600 JSON messages per second, with up to 1000 messages per second at peak data flow rates. An overview of the different kinds of signals, and the amount of each signal collected is presented in the appendix.

Typical collection rates in this system are around 100 million time-value pairs\(^10\) per hour from an average census of 35 occupied beds. Peak collection times may result in data collection from all 42 beds which generates around 130 million time-value pairs per hour. Examples of some of the signals collected are shown at different time resolutions in figure 2.

The number of time-value pairs collected each week since the system’s inception is shown in figure 3. Several notable stages of data collection can be seen in this figure. After the initial installation in February 2016, data was collected from a small group of four beds for a period of ten weeks to test the data collection system. After this trial additional beds were brought online throughout June to October 2016. By June 2017

---

\(^9\) For simplicity, numeric values that are collected once every 1024 ms will be referred to as 1 Hz throughout the rest of this paper.

\(^10\) The term time-value pair is used to describe a single value in a time series and its associated timestamp. Timestamps are explicitly stored in AtriumDB and use millisecond precision.
all 42 beds had been connected to the system. In December 2017 we introduced a ‘priority list’ that allowed us to specify the relative importance of each signal that we collected. This reduced the number of redundant signals collected, for example preventing collection of both 500 Hz ECG and 250 Hz ECG from the same lead. Introduction of the priority list reduced the average rate of data collection by around 20%, but improved the overall continuity and quality of the signals being collected. Two months of data (October and November 2016) were irreversibly lost during a hardware failure event. Additional backup measures were introduced after this event to prevent similar incidents happening in the future. Other variations in figure 3 are due to minor hardware outages or changes in patient census and acuity.

JSON messages are examined and filtered upon retrieval from the Rabbit queue and cached on a secure server as an uncompressed text file, with a single file containing only messages from a unique bed-signal combination for that hour. At peak load this results in around 700 unique text files being generated each hour. At the end of the hour an asynchronous process, also written in PHP, parses each file and compresses it into a custom file format (see section 2.5). After the conversion to compressed file format, historical JSON
files are moved to a different server for long term storage and backup. Temporary text files are deleted once this process has been completed.

2.4. Data compression

An ideal storage format must be both compact and highly accessible. A wide variety of different storage solutions were considered that would achieve these goals. For planning purposes, we estimated the rate of data ingest to be approximately $1.0 \times 10^{12}$ time value pairs per year. Traditional database approaches are not suitable for storing such large volumes of time series information (Stonebraker and Cetintemel 2005).

Recently there has been a push towards NoSQL type architectures for storage of time series data (Dunning and Friedman 2014). Such databases exploit inherent redundancies in sequences of values in the time series to provide more compact representations of the data. In these systems, intervals of contiguous data are compressed together into a single chunk or block. This facilitates a wider range of compression options, and allows indices to be more coarsely constructed, reducing overhead and leading to lower latency queries.

Medical devices sample and deliver physiological waveforms in a way that makes delta encoding particularly effective (Chang and Lin 2010). Physiological waveforms are often smoothed and pre-processed in the sensor or monitor such that iterative application of delta encoding results in a more compressible byte stream. Delta encoding records an initial value, and then only stores the difference between subsequent values (Shafer et al 2013, Engelson et al 2000, Singh et al 2015). This approach results in lossless compression when applied to arrays of integers (Lemire et al 2019). In our system, each signal type was profiled to identify the optimal sequence of pre-processing steps, for example selecting the number of iterations of delta encoding to apply.

We utilize a custom designed file format that segments the signals into a series of intervals of user defined length. Each individual block of data may be pre-processed in a variety of ways to decrease its entropy and therefore increase its compressibility (Kriechbaum et al 2019). The pre-processed arrays are then further compacted using an open source compression algorithm.

This compression approach is similar to that employed by Blalock et al (2018). Each sequence of time-value pairs is first separated into two distinct arrays of equal length. One array contains the values while another array of equal length stores the time associated with each value in the first array. Times are stored explicitly so that gaps in the data can be faithfully represented. This differs from other approaches that implicitly store time by assuming a constant sample rate. The number of bytes required to store the delta encoded array is determined based on the range of values in the resultant array, and the most compact representation is used for each separate block of data. Binary representations of the delta encoded arrays are subsequently compressed using the BZip2 algorithm (Seward 2000). This additional step further reduces the size of each block’s payload by applying run length encoding (RLE) and otherwise removing any remaining redundancy in the data representation. Alternative open source compression algorithms may be employed at this stage in place of BZip (for example, GZip, ZStandard, LZ4, Snappy etc) (Ziv et al 1977, LZ4 2019, zlib 2019, ZStandard 2019, Snappy 2019) which may result in slightly larger files and faster encode/decode operations. Compressed data files can be re-encoded using a different block size or using different compression options at a later time to meet the size and speed requirements of a particular situation.
Each time-value pair stored in AtriumDB consists of a millisecond precision timestamp coupled with an associated integer value. Signals that arrive as floating-point values are converted to an integer by multiplying by a power of 10, with the multiplying factor chosen such that the resulting integer is representative of the underlying precision of the data (Isenburg 2013). Ideally the raw integer values collected by the A/D converters on the various sensors are available directly or can be reverse engineered. These integer values may be subsequently re-scaled to a floating-point representation with relevant units, such as mmHg for pressure waveforms or mV for ECG. The scaling factors required for these conversions are stored as tables in a MariaDB database (MariaDB 2019).

Use of a ‘compression ratio’ to measure data compression is problematic because the signals exist in a number of formats before being compressed. It is therefore not clear which of these formats should be used for comparison, so the effectiveness of our compression is reported in *bits per value*\(^{11}\). This approach provides an objective way to assess the amount of compression achieved when averaged across a large amount of data from a given signal type.

The database currently comprises 194 different signal types, including 48 waveforms and 146 unique 1 Hz signals. The signals exhibit a wide range of compressibility as the compressed size is roughly proportional to the entropy of the signal (Balakrishnan and Touba 2007). Compression ratios achieved for some commonly collected waveforms are shown in table A1 and for other commonly collected physiological parameters in table A2 in the appendix. Labels used in these tables may be vendor-specific, and further work is required to unify nomenclature (Kasparick et al 2015, Kasparick et al 2019, Singh et al 2019).

### 2.5. TSC file format

We designed a file structure that allowed multiple compressed segments of a signal to be stored in a single file. These segments, called blocks, can vary in size but typically contain between 1000 and 100,000 time-value pairs. The block size parameter is set globally for each signal type when the system is initialized. In our prototype system a block size of 10 minute duration was used for all signal types. In order to access any value (or any sequence of values) within a block the entire block must be decompressed. Use of smaller blocks facilitates more granular access but may result in an increase in handling and container overhead (Kriechbaum et al 2019). Longer queries may span multiple blocks, allowing parallelized decompression. Multiple channels of data are supported in the file format, but not used in our current system. The files have a .tsc extension (where the tsc file extension stands for time series compression) The structure of the TSC file format is shown in figure 4.

Each block header contains all information required to decompress that individual block. In this way individual blocks can be passed across a network and decompressed on a remote system. This approach to data transfer reduces the load on the central server by offloading the computational load associated with de-compression to a client application or a distributed compute node.

Information stored in the block header includes the number of elements in the stored arrays, the size of the compressed payload in bytes, the initial values required for reversing the delta encoding, the number of iterations of delta encoding applied, the number of bytes required to store each value in the delta encoded array, and the type of open source algorithm employed in the secondary compression stage. Summary statistics such as the number of values, maximum, minimum, mean and standard deviation of the values, are also stored in the block headers. The main file header lists the number of blocks, the start and end times of each block, and the location (in bytes) of the beginning of each block of data in the file. A single block may contain up to 256 channels, and a single TSC file may contain up to 65,536 blocks.

Although this data format was designed for high-frequency physiological data (i.e. waveforms), it can also be used for lower frequency data (\(\leq 1\) Hz). Storage of data in this format results in one compressed file per device-signal-hour. Each of the compressed files is stored on a secure server, with a folder structure that is organized by date. Details regarding each compressed file are recorded as a row in a MariaDB relational database. Metadata about the file, such as file size, number of values, statistical properties of the values in the file, and the start and end times of the data stored in that file, are also recorded in the database.

*DeviceName* is used as the primary key in the database instead of *patientID*. This approach ensures that retrospective adjustments can be made to the admission/discharge information without altering the folder structure or file names. Patients are mapped to a particular bed space as a function of time via a separate table in the MariaDB database.

---

11 Throughout this paper, we use the term *bits per value* to refer to the average number of bits required to store each time-value pair across the database (or a given subsection thereof). One bit is 1/8\(^6\) of a byte.
2.6. High fidelity data

By design, AtriumDB aggregates and stores all information collected in the form in which it was provided by the medical devices, including its flaws and inconsistencies. Information is stored in the same data types that the monitors and other medical devices produce, and lossless compression is employed. In this sense ‘high fidelity’ means that the information is stored in its raw form, with the goal of minimizing loss of information up to the point of storage on disk. Stored data are immutable, so the database represents an indelible record of what was collected. The utility of this approach is outlined in section 4.1.

Over time algorithms may be developed to detect and characterize known data quality issues. These algorithms are implemented in-line at the time of retrieval and may be modified or improved over time. These signal quality indices (SQIs) are discussed in section 4.2.

2.7. Data intervals

AtriumDB uses the concept of data intervals to delineate contiguous areas of interest in the time series data. Each interval is a period from \( t_1 \) to \( t_2 \) for a given bedspace-signal combination. Intervals are useful abstract data types as they often form the foundation for subsequent data requests.

Although interval may be specified by a user, the most common method of interval generation is via an automated process within the database API, particularly when accessing large portions of the database. For example, a user or computer program may ask for a list of time intervals where ‘continuous’ 500 Hz ECG Lead II waveform is available for a particular patient. Because the stored data contains many small gaps, this request may return dozens, or indeed thousands of intervals, each indicating a segment of time where data is guaranteed to be available. In this sense the intervals are a property of the data, as opposed to a user defined period. These data structures are explicitly available to users of the database by design. An interval may be used as a foundation for a subsequent data request, via an automated or manual process. The size of the gaps in the data may range from very small (a few hundred milliseconds) to very large (minutes or hours), so the definition of ‘continuous’ may be variable and context dependant. When generating lists of intervals AtriumDB allows the user to specify a ‘gap tolerance’ parameter that defines which gaps are considered to be significant (see section 2.8).
2.8. Data continuity
Gaps in physiological signals may be due to a variety of reasons including the intrinsic absence of data, networking issues, removal of sensors for cleaning, or as the result of a particular sampling approach. A suite of standard SQIs are applied during the data ingest process, these include identification of flatline signals and clipping waveforms (see section 4.2). Other custom quality indices are being actively developed and applied retrospectively. These SQIs may be used as constraints when selecting data that result in removal of portions of the signal, introducing additional gaps.

Some subsequent use cases may not be affected by small amounts of missing data, for example a gap of 500 ms duration, but might be adversely affected by longer gaps. With this in mind, portions of data with small gaps may still be considered to be 'continuous' in some contexts.

AtriumDB parametrizes continuity by defining a concept called 'gap tolerance'. The AtriumDB API can return a list of continuous intervals where the required continuity (as specified using the gap tolerance parameter) is guaranteed. For example, if we request all intervals of 'continuous' data with a gap tolerance of two seconds, then the database will return a list having time periods from $t_1$ to $t_2$ where each of those intervals is guaranteed not to contain any periods longer than two seconds that do not contain any data. A smaller gap tolerance results in a larger number of intervals, and vice versa.

2.9. Composite signals
Physiological models or analytical techniques may require multiple signals as inputs. These models are unable to operate unless all necessary signals are available in the input data. For a given application we can request a list of time intervals where all required input signals are available, and where they meet the quality criteria required for that analytical technique. We refer to these groups of signals as a 'composite signal'. A composite signal comprises two or more signals from a given bedspace, each of which may have a different specified gap tolerance. Queries to the API can define the list of required signals, and their required gap tolerance, and optionally also constraints as defined by one or more SQIs (see section 4.2).

A hypothetical example of a composite signal is shown in figure 5. It shows the intervals of continuous time for blood pressure, heart rate, and an ECG waveform, with a gap tolerance of 10 s, 60 s, and 60 s respectively. The composite intervals in this example are defined by the intersection of the input components' continuous time intervals. These intervals can optionally be further subdivided by providing a parameter that describes the maximum number of time-value pairs that an interval may contain, or by specifying the maximum interval duration. Intervals of this kind, containing multiple signals, may be used as a substrate for training machine learning models.

2.10. Data retrieval
Once we identify intervals of time where data of sufficient quality and continuity is available, a secondary request may be issued to the API requesting the raw data from those times. This two-step process eliminates the risk of requesting data from a time where no information was available or was of insufficient quality. In the current implementation data requests can be made using a Representational State Transfer (REST) based API, and data are returned via HTTP in JSON format.

A secondary method of accessing the database utilizes a Software Development Kit (SDK) written in the Julia programming language (Bezanson et al 2017). The Julia SDK interfaces easily with other programming languages such as C, C++, R, and the Python programming language (Lobianco 2019). This interface is

| Inputs | Gap Tolerance | Data Continuity | Availability |
|--------|---------------|-----------------|-------------|
| HR     | 10 s          |                 | ~70 %       |
| BP     | 60 s          |                 | ~60 %       |
| ECG    | 60 s          |                 | ~60 %       |
| Composite | Mixed     |                 | ~50 %       |

*Figure 5. Composite data continuity.*
2.11. Data labels

Many machine learning applications rely on data ‘labels’ that define a particular event or outcome such as true states of clinical outcomes or the true disease phenotypes (Xiao et al. 2018). For example, mortality is often used as a label, dividing a cohort into two groups, patients who survived, and those who did not. Supervised machine learning techniques rely on labelled data for model training, creating a need for collection of, or automatic generation of labelled events that can be used to provide context to physiological signals (Komorowski 2019). Labeling of clinical data is often expensive, labor intensive, and consensus is difficult to obtain (Johnson et al. 2016).

Labels for training a supervised machine learning model can be generated from a wide variety of sources. They can be collected in real time at the bedside, for example as part of an ongoing project aimed at recording cardiac arrests that occur within the unit (Tonekaboni et al. 2018). They can come from the EHR and include such labels as diagnosis, age, length of stay, readmission, and surgery procedure. In some cases, a small retrospective training dataset is manually labeled through visualization of the physiological data. This approach was used for example when building a training dataset for automated labelling line access events in the arterial blood pressure waveforms (Nagaraj et al. 2019).

2.12. Ethical framework

We have taken the approach that data housed within AtriumDB is the property of the patient and is fully discoverable. Data is stored losslessly in its raw form and is not purged, meaning that no bias is introduced during the data collection process. These data are available for clinical use so consent is not obtained. Physiological data are merged with an admit, discharge and transfer (ADT) data feed from the electronic health record (EHR) to correctly associate data with a patient. AtriumDB is housed within the clinical data center at the Hospital for Sick Children behind the hospital firewall and is protected in a similar manner to other clinical systems. The database only accessible by authorized individuals using appropriate security credentials.

Secondary research utilization of AtriumDB is regulated by a governance structure similar to that of a biobank. This governance structure covers the use of anonymized or anonymous patient information for research purposes. Following de-identification, the data is time shifted in a manner that is patient or study specific in order to prevent inadvertent re-identification of patients based upon temporal associations. Keys permitting re-identification of patient data by re-association of a subject identifier with a medical record number, and re-alignment in time, are retained, allowing re-identification of the data should the need arise.
This governance mechanism facilitates clinical research, as well as data-driven exploration and hypothesis generation.

3. Results

Physiological waveform collection via ViNES/AtriumDB began in the critical care unit in February 2016, ramping up to having all 42 beds connected by April 2017. Data collection has continued since that time, and as of December 2019 the database contained over 720,000 patient-hours of data, collected from over 5300 patients. These data are represented as $2.8 \times 10^{12}$ time-value-pairs stored in approximately 14.3 million compressed files. The dataset comprises almost 200 unique signal types, including waveform data for the complete duration of each patient’s stay.

3.1. Data compressibility

As of December 2019, the database is 820 GB in size and growing by around 285 GB per year. At this rate of collection, we can continue to store the complete dataset at full resolution. With a small disk footprint it becomes affordable to store the data on high-performance disk drives. We utilize an array of solid-state drives (SSDs) to store the compressed data files which further improves the input/output (I/O) performance of the database.

The average compression rate for all signals combined was calculated by dividing the sum of the sizes of all TSC files by the total number of time-value-pairs stored in the database. Using this approach resulted in an average of 2.57 bits per time-value pair in the installation at the HSC. Similarly, we calculated compression rates for each signal type for each by averaging the across the entire database for that particular signal. ECG Lead II sampled at 500 Hz for example is compressed to 2.34 bits per value, or around 526KB per hour. Compressibility of selected signals is shown in tables A1 and A2 in the appendix.

3.2. Retrieval performance

The HTTP based API was load tested and shown to be able to deliver approximately 800,000 time-value pairs per second to a single endpoint application. This is fast enough to provide data to a web based visualization, or for exporting small amounts of data for explorative purposes. This is not indicative of the maximum throughput of the system however as the server is able to process many requests simultaneously. Database performance was stress tested by setting up multiple applications to simultaneously request data via the HTTP based API. During this test ten separate computers were programmed to simultaneously and repeatedly request a random hour of 500 Hz ECG data from the system. The maximum throughput of the system under this parallel load was approximately 5 million time-value pairs per second via the HTTP based API. This scenario better simulates the server conditions in a multi-user scenario, or a distributed computing environment.

Data access rates using the Julia library are influenced by a wide range of factors including the number of CPU cores available and the performance characteristics of the data storage medium. Under a typical query load data access rates of 50 million time-value pairs per second can be achieved using a desktop computer with eight cores reading from a local cache of TSC files housed on a SSD.

4. Discussion and future directions

Most ICUs offer 24/7 monitoring of vital signs but lack the capability for data collection, integration, and analysis of physiological data (Mann et al 2020). Recent advances in computing have enabled the long-term storage of high fidelity physiological signals (Dunning and Friedman 2014). The availability of large, high-frequency physiological datasets has opened up new possibilities for deep learning and other data driven approaches (Hong et al 2019, Naylor 2018, Wang et al 2019).

The application of artificial intelligence and machine learning in healthcare has grown rapidly in recent years. These models are data driven, and model training times are often dictated by data access speed. A major limitation in utilizing high-frequency physiologic waveforms for research and clinical purposes is the lack of a scalable architecture that can store these data permanently while also providing data retrieval rates that are fast enough to allow practical exploration and/or utilization of the entire dataset.

AtriumDB addresses these limitations by providing cost effective data storage coupled with high-performance data access. The availability of a large, readily accessible, high throughput data storage engine for physiological signals leads to faster model training times, and enables larger cohorts to be used during the model training phase.

It takes time to accumulate a critical mass of data to apply to research question, and now that we have a corpus of information established, numerous projects are underway assessing the utility of the data for
research purposes. For example see Eytan et al (2016), Tonekaboni et al (2018), Eytan et al (2019), Nagaraj et al (2019), Goodfellow et al (2018), Eytan et al (2019).

4.1. Data utility
Because the data streams are collected and stored exactly as they were produced in the critical care unit, they provide an ideal substrate for development and testing of ‘real time’ data algorithms, as retrospective data can be replayed to simulate a streaming data environment. In this way any model developed with the intention of deployment in a streaming data environment can first be tested retrospectively to see how it deals with the wide variety of signal quality and data availability issues that are present in a real ICU.

Our experience has been that having access to a large dataset with realistic artefacts lends itself to the process of anomaly discovery and understanding. Anomaly detection and removal algorithms can be iteratively developed and tested on a broad range of data conditions (see section 4.2). The approaches developed during this process can be deployed in-line as data is exported from the database or applied prospectively as a pre-processing step to streaming data being used as input for real time algorithms.

The initial prototype of AtriumDB was deployed without a user interface. The goal was to expose a highly functional API that would facilitate rapid development of multiple visualization tools, data exploration tools, and user interfaces. For example Eytan et al (2019). Several proof of concept applications have been developed, and a range of user interfaces are being developed that will provide clinicians and researchers the opportunity to request data directly or visualize signals via a web-based portal.

4.2. Signal quality indices
Physiological signals exhibit a wide variety of artefacts and noise, meaning that data may be unusable for some analytical and modelling purposes (Nizami et al 2013, Takla et al 2006). There are many approaches to assessment of physiological signal quality, for example see Clifford et al (2012), Elgendi (2016). We designed the AtriumDB system to be able to store SQIs at scale, and to have them readily available when selecting data. For example, our group has been developing approaches for real time detection of artefacts from streaming data. One example of this is automated identification of line access artefacts in arterial blood pressure waveforms (Nagaraj et al 2019). By adopting this approach, we can optionally utilize one or more SQIs as constraints or indexes while requesting data or scoping research projects.

4.3. Automatic generation of retrospective cohorts
If the data input requirements for a given algorithm are known, they can be specified as data availability, quality, and continuity constraints. This information enables AtriumDB to provide a report on how many patients match those constraints and provide a distribution of how many hours of data are available for each of those patients. The generalizability of a model can be explored through this process, allowing a researcher to examine the trade-off between a highly specific model that may have onerous input requirements, and a model that is designed to run on whatever data streams are most commonly available.

4.4. Creating derivative databases
Subsections of the database relating to a specific study or project can be exported and stored in a secondary database using the TSC file format. Parameters such as block size and compression approach, can be optimized for each specific use case. Physiological signals in the derivative database may be optionally cleaned, pre-processed, and de-identified, producing an optimal data distribution format for research groups who are focusing on a small portion of the database (Ferrao et al 2016).

4.5. Limitations
AtriumDB is optimized for high-performance retrospective access. Data becomes available to users approximately one hour after collection, so a separate system must be used for ‘real time’ applications. Applications that require access to real-time, streaming data can pull messages directly from a separate and dedicated RabbitMQ feed. Data for currently admitted patients are stored in an InfluxDB database (InfluxDB 2019) which allows very recent data to be queried if required.

Deploying this system at a different hospital would require development of numerous bespoke components, for example to integrate laboratory and admission/discharge information from a different source than used at the HSC. System initialisation also requires installation of new wiring systems and servers to facilitate data collection and integration.

4.6. Future directions
The speed at which data can be exported can be a rate limiting step in a high-performance computing environment, some aspects of the system may be rewritten in a compiled language such as C/C++ for faster performance.
Several projects are underway that will allow researchers and clinicians to pro-actively label events at the bedside, providing further context about patient care and providing labels that can be used for supervised machine learning.

Now that we have established a proven and stable architecture, a longer term goal is to allow clinicians at the bedside to access the data real time, and to integrate the data seamlessly into predictive models that can be incorporated into online clinical protocols that enhance decision making for critically ill patients in the intensive care unit.

5. Conclusions

AtriumDB is a time series storage engine that was designed from first principles and specifically optimized for physiological waveforms. The highly compact data representation facilitates continuous collection and storage of physiological waveforms and other information collected in an ICU environment. The web based API provides versatile data access, while the SDK facilitates rapid programmatic retrieval of data for use in high-performance computing applications. The system has been used to compile a large dataset that is representative of the streaming data environment in an ICU, and as such can be used to meaningfully test and tune ‘real time’ algorithms on retrospective data before deployment.

Acknowledgment

The authors acknowledge support from the David and Stacey Cynamon Chair in Pediatric Critical Care at The Hospital for Sick Children.
## Reference tables

**Table A1. Availability and compressibility of commonly collected waveforms as of December 2019.**

| ID | Signal Name           | Symbol | Signal Units | Sample Frequency (Hz) | Number of Patients | Number of Files | Data Size (MB) | Number of Values (Millions) | Compression Ratio (Bits/Value) |
|----|-----------------------|--------|--------------|-----------------------|--------------------|----------------|----------------|-----------------------------|---------------------------------|
| 1  | Airway flow           | AWF    | l/min        | 125                   | 2318               | 222 998        | 27 055         | 96 851                      | 2.34                            |
| 2  | Airway Pressure       | AWP    | cm H₂O       | 125                   | 2312               | 204 185        | 24 772         | 87 765                      | 2.37                            |
| 3  | Airway Volume         | AWV    | ml           | 125                   | 2165               | 194 301        | 12 413         | 84 273                      | 1.24                            |
| 4  | Arterial Blood Pressure | ABP | mmHg         | 125                   | 3933               | 297 349        | 49 854         | 127 103                     | 3.29                            |
| 5  | End-Tidal CO₂         | CO₂    | mmHg         | 62.5                  | 2901               | 300 774        | 14 914         | 65 473                      | 1.91                            |
| 6  | Central Venous Pressure | CVP  | mmHg         | 125                   | 2324               | 184 853        | 22 585         | 80 266                      | 2.36                            |
| 7  | Diaphragm Electrical Activity | Edi | uV          | 125                   | 404                | 30 372         | 33             | 13 153                      | 0.02                            |
| 8  | Respiratory Impedence | Resp  | Ohms         | 62.5                  | 5294               | 743 130        | 75 208         | 161 411                     | 3.91                            |
| 9  | Intracranial Pressure | ICP   | mmHg         | 125                   | 233                | 12 456         | 902            | 53 031                      | 1.43                            |
| 10 | Lead I - ECG          | ECG-I  | mV           | 500                   | 3629               | 123 437        | 34 027         | 144 360                     | 1.98                            |
| 11 | Lead II - ECG         | ECG-II | mV           | 250                   | 718                | 32 654         | 9733           | 27 445                      | 2.97                            |
| 12 | Lead II - ECG         | ECG-II | mV           | 500                   | 5262               | 627 479        | 298 872        | 1073 554                    | 2.34                            |
| 13 | Lead III - ECG        | ECG-III| mV          | 250                   | 2216               | 250 303        | 86 170         | 212 126                     | 3.41                            |
| 14 | Lead III - ECG        | ECG-III| mV          | 500                   | 4538               | 113 793        | 22 470         | 96 135                      | 1.96                            |
| 15 | Lead III - ECG        | ECG-III| mV          | 250                   | 650                | 20 670         | 6492           | 17 044                      | 3.20                            |
| 16 | Pulse Oximetry        | Pleth  | N/A          | 125                   | 5264               | 744 487        | 119 745        | 323 907                     | 3.10                            |
| 17 | Right Atrial Pressure | RAP   | mmHg         | 125                   | 269                | 28 659         | 3663           | 12 511                      | 2.46                            |
| 18 | Left Atrial Pressure  | LAP    | mmHg         | 125                   | 84                 | 4234           | 571            | 1829                       | 2.62                            |
Table A2. Availability and compressibility of commonly collected physiological variables as of December 2019.

| Signal Name                  | Symbol | Signal Units | Derived From Waveform | Number of Patients | Number of Files | Data Size (MB) | Number of Values (Millions) | Compression Ratio (Bits/Value) |
|-----------------------------|--------|--------------|-----------------------|--------------------|----------------|----------------|-------------------------------|-------------------------------|
| ABP (Diastolic)             | ABPd   | mmHg         |                       | 4                  | 2222           | 181 051        | 367                          | 605                           | 5.08                          |
| ABP (Mean)                  | ABPm   | mmHg         |                       | 4                  | 2235           | 181 337        | 374                          | 617                           | 5.08                          |
| ABP (Systolic)              | ABPs   | mmHg         |                       | 4                  | 2222           | 181 051        | 384                          | 605                           | 5.33                          |
| Airway Respiration Rate     | awRR   | breaths/min  |                       |                    | 2773           | 208 266        | 416                          | 699                           | 4.99                          |
| Barometric Pressure          | PB     | mbar         |                       |                    | 1770           | 155 371        | 276                          | 529                           | 4.39                          |
| CVP (Mean)                  | CVPm   | mmHg         |                       | 6                  | 2209           | 191 835        | 379                          | 645                           | 4.93                          |
| ECG Heart Rate              | HR     | bpm          |                       | 12                 | 5285           | 741 117        | 1608                         | 2510                          | 5.37                          |
| End-Tidal CO₂               | etCO₂  | mmHg         |                       |                    | 2818           | 295 249        | 495                          | 992                           | 4.19                          |
| Esophageal Temperature       | T_{esoph} | °C          |                       |                    | 903            | 80 632         | 157                          | 274                           | 4.81                          |
| Fractional Inspired Oxygen  | FIO₂   | %            |                       |                    | 1775           | 156 803        | 345                          | 534                           | 5.43                          |
| Impedance Respiration Rate  | RR     | breaths/min  |                       | 8                  | 5285           | 735 804        | 1721                         | 2493                          | 5.79                          |
| Inspired Minimum CO₂        | imCO₂  | mmHg         |                       |                    | 2814           | 294 102        | 416                          | 988                           | 3.54                          |
| Non-invasive BP (Systolic)  | NBPs   | mmHg         |                       |                    | 5266           | 441 198        | 487                          | 1101                          | 3.71                          |
| Non-invasive BP (Diastolic) | NBPd   | mmHg         |                       |                    | 5266           | 440 758        | 489                          | 1095                          | 3.75                          |
| Non-invasive BP (Mean)       | NBPm   | mmHg         |                       |                    | 5266           | 441 230        | 490                          | 1098                          | 3.75                          |
| Patient Work of Breathing   | WOBp   |              |                       |                    | 1368           | 110 496        | 227                          | 184                           | 10.36                         |
| Peripheral Oxygen Saturation| SpO₂   | %            |                       | 16                 | 5270           | 746 619        | 1368                         | 2523                          | 4.55                          |
| Premature Ventricular Beats | PVC    | bpm          |                       |                    | 5264           | 727 064        | 1062                         | 2469                          | 3.61                          |
| Pulse Rate                  | Pulse  | bpm          |                       | 12,16              | 3470           | 292 924        | 698                          | 978                           | 5.99                          |
| Pulse Rate from NBP         | Pulse  | bpm          |                       |                    | 5053           | 401 737        | 443                          | 1006                          | 3.69                          |
| Pulse Rate from Plethysmo-  | Pulse  | bpm          |                       | 16                 | 5291           | 739 485        | 1696                         | 2495                          | 5.70                          |
| gram                        |        |              |                       |                    |                |                |                              |                               |                               |
| Rectal Temperature          | T_{rect} | °C          |                       |                    | 250            | 7855           | 14                           | 26                            | 4.60                          |
| Relative Perfusion          | Perf   |              |                       |                    | 5270           | 746 264        | 2088                         | 2522                          | 6.94                          |
| Rise Time                   | RiseTi | %            |                       | 2                  | 1672           | 111 302        | 155                          | 373                           | 3.49                          |
| Ventilator Work of Breathing| WOBv   |              |                       |                    | 1387           | 125 956        | 416                          | 429                           | 8.14                          |
ORCID iD

Sebastian D Goodfellow  https://orcid.org/0000-0001-8522-9880

References

Acharya U R, Joseph K P, Kannathal N, Lim C M and Suri J S 2006 Heart rate variability: a review Med. Biol. Eng. Comput. 44 1031–51
Allen J 2007 Photoplethysmography and its application in clinical physiological measurement Physiol. Meas. 28 R1
Andersen M P and Culler D E 2016 BTrDB: optimizing storage system design for timeseries processing 14th (USENIX Conf. on File and Storage Technologies (FAST 16) pp 39–52
Bader A, Kopp O and Falkenthal M 2017 Survey and comparison of open source time series databases Datenbanksysteme für Business, Technologie und Web (RTW 2017)-Workshopband
Balakrishnan K J and Touba N A 2007 Relationship between entropy and test data compression IEEE IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 26 386–95
Baljak V, Ljubovic A, Michel J, Montgomery M and Salaway R 2018 A scalable realtime analytics pipeline and storage architecture for physiological monitoring big data Smart Health 9 275–86
Berntson G G and Stowell J R 1998 ECG artifacts and heart period variability: don’t miss a beat! Psychophysiology 35 127–32
Bezanson J, Edelman A, Karpinski S and Shah V B 2017 Julia: A fresh approach to numerical computing SIAM Rev. 59 65–98
Blackson D, Madden S and Guttaj J 2018 Sprintr: Time Series Compression for the Internet of Things Proc. ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies vol 2 p 93
Blount M, Ebling M R, Mikael Eklund J, James A G, McGregor C, Percival N, Smith K P and Sow D 2010 Real-time analysis for intensive care IEEE Eng. Med. Biol. Mag. 29 110
Blum J M, Joo H, Lee H and Saeed M 2015 Design and implementation of a hospital wide waveform capture system J. Clin. Monit. Comput. 29 359–62
Bond R R, Finlay D D, Nugent C M and Moore G 2011 A review of ECG storage formats Int. J. Med. Inform. 80 681–97
Bonnici T, Charlton P, Pierre D, Tarassenko L, Watkinson P J and Beale R 2015 ‘Experiences implementing a system for widespread recording of patient physiology’
Brinkkoson B H, Bower M R, Stengel K A, Worrell G A and Stead M 2009 Large-scale electrophysiology: acquisition, compression, encryption and storage of big data J. Neurosci. Methods 180 185–92
Brossier D, El Taani R, Sauthier M, Roumeliotis N, Emeriaud G and Jouvè P 2018 Creating a high-frequency electronic database in the PICU: the perpetual patient Pediatr. Crit. Care Med. 19 e189–e198
Burykin A, Peck T and Buchman T G 2011 Using ‘off-the-shelf’ tools for terabyte-scale waveform recording in intensive care: computer system design, database description and lessons learned Comput. Methods Programs Biomed. 103 151–60
Cabeleira M, Ercole A and Smielewski P 2018 HDF5-based data format for archiving complex neuro-monitoring data in traumatic brain injury patients Intracranial Pressure and Neuromonitoring XVI (Berlin: Springer) pp 121–5
Cannesson M, Ikeda T, Rinchert J, Lee C, Subramaniam K, Baldi P, Dubravskis A and Pinkay M R 2019 Machine learning of physiological waveforms and electronic health record data to predict, diagnose and treat haemodynamic instability in surgical patients: protocol for a retrospective study BMJ Open 9 e031988
Chang G and Lin Y 2010 An efficient lossless ECG compression method using delta coding and optimal selective Huffman coding 6th World Congress of Biomechanics (WCBO 2010) (August 1–6, 2010 Singapore) (Berlin: Springer) pp 1327–30
Charlton P H, Villarroel M and Salguiero F 2016 Waveform analysis to estimate respiratory rate Secondary Analysis of Electronic Health Records (New York: Springer) pp 377–90
Chen Y, Wang Z-y, Yuan G and Huang L 2017 An overview of online based platforms for sharing and analyzing electrophysiology data from big data perspective Wiley Interdisciplinary Reviews: Data Min. Knowl. Discovery 7 e1206
Clark M T et al 2011 Breath-by-breath analysis of cardiorespiratory interaction for quantifying developmental maturity in premature infants J. Appl. Physiol. 112 859–67
Clifford G D, Behar J, Li Q and Rezek I 2012 Signal quality indices and data fusion for determining clinical acceptability of electrocardiograms Physiol. Meas. 33 1419
Cosgriff C V, Celi L A and Stone D J 2019 Critical care, critical data Biomed. Eng. Comput. 10 1179597219856564
De Georgia M A, Kafashif J, Jacono F J and Lopari K A 2015 Information technology in critical care: review of monitoring and data acquisition systems for patient care and research Sci. World J. 2015 727694
Dunning T and Friedman E 2014 Time Series Databases New Ways to Store and Access Data, Volume 1st Edit (Sebastopol, CA: O’Reilly Media)
Elgendi M 2016 Optimal signal quality index for photoplethysmogram signals Bioengineering 3 21
Engelson V, Fritzon D and Fritzon P 2000 Lossless compression of high-volume numerical data from simulations Proc. Data Conf. Eyton D, Goodwin A, Laussen P and Gueruquierian A-M 2016 Insights from multi-dimensional physiological signals to predict and prevent cardiac arrests Pediatr. Crit. Care Med. 17 81–2
Eyton D, Jegatheeswaran A, Mazwi M L, Assadi A, Goodwin A J, Greer R W, Goodfellow S D and Laussen P C 2019 Temporal variability in the sampling of vital sign data limits the accuracy of patient state estimation Pediatr. Crit. Care Med. 20 e33–e341
Eyton D, Mazwi M L, Goodwin A J, Goodfellow S D, Assadi A, Greer R W, Jegatheeswaran A and Laussen P C 2019 Revisiting oxygen dissociation curves and bedside measured arterial saturation in critically ill children Intensive Care Med. 45 1832–4
Fanelli A, Jalsanokar R, Filipiedis A, Holappa J and Heldt T 2018 A waveform archiving system for the GE solar 8000i bedside monitor Intracranial Pressure and Neuromonitoring XVI (New York: Springer) pp 173–7
Ferrao J C, Oliveira M D, Janela F and Martins H M 2016 Preprocessing structured clinical data for predictive modeling and decision support Appl. Clin. Inf. 7 1135–53
Gil Y et al 2007 Examining the challenges of scientific workflows Computer 40 24–32
Goldstein B, McNamara J, McDonald B A, Ellenby M, Lai S, Sun Z, Krieger D and Scabass R J 2003 Physiologic data acquisition system and database for the study of disease dynamics in the intensive care unit Crit. Care Med. 31 433–41
Goodfellow S D, Goodwin A J, Greer R, Laussen P C, Mazwi M and Eyton D 2018 Atrial fibrillation classification using step-by-step machine learning Biomedical Physics & Engineering Express 4 045005
Hannun AY, Rajpurkar P, Haghpanahi M, Tison G H, Bourn C, Turakhia M P and Ng A Y 2019 Cardiologist-level arrhythmia detection with convolutional neural networks Nature Medicine 25 65
Holder A L and Clermont G 2015 Using what you get: dynamic physiologic signatures of critical illness Crit. Care Clinics 31 133–64
Hong S, Zhou Y, Shang J, Xiao C and Sun J 2019 Opportunities and challenges in deep learning methods on electrocardiogram data: A systematic review arXiv preprint arXiv:2001.01550
Horvat C M and Oge H 2019 Variability in vital sign documentation as a barrier to modeling patient state: Why patient records may need more complexity? Pediat. Crit. Care Med. 20 690–1
Hu P F et al 2017 Reliable collection of real-time patient physiologic data from less reliable networks: a ‘monitor of monitors’ system (MOMS) J. Med. Syst. 41 3
InfluxDB https://www.influxdata.com/ Accessed: 2 July 2019
Isenberg M 2013 Lazip: lossless compression of ladar data Photogramm. Eng. Remote Sens. 79 209–17
Jacono F J, De Georgia M A, Wilson C G, Dick T E and Loparo K A 2010 Data acquisition and complex systems analysis in critical care: developing the intensive care unit of the future J. Healthc. Eng. 1 337–55
Jensen S K, Pedersen T B and Thomsen C 2017 Time series management systems: A survey IEEE Trans. Knowl. Data Eng. 29 2581–600
Johnson A E W, Ghassemi M M, Nemati S, Niehaus K E, Clifton D A and Clifford G D 2016 Machine learning and decision support in critical care Proc. IEEE Inst. Electr. Electron. Eng. 104 444
Karhonen I, Van Gils M and Gade J 2001 The challenges in creating critical-care databases IEEE Eng. Med. Biol. Mag. 20 58–62
Kasparick M, Schlichting S, Golatowski F and Timmermann D 2017 New IEEE 11073 standards for interoperable, networked point-of-care medical devices 37th Ann. Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC) pp 1721–4
Kasparick M, Andersen B, Franke S, Rockstroh M, Golatowski F, Timmermann D, Ingenfer I and Neumuth T 2019 Enabling artificial intelligence in high acuity medical environments Minim. Invasive Ther. Allied Technol. 28 120–6
Kiviniemi A M et al 2010 Low-frequency oscillations in R–R interval and blood pressure across the continuum of cardiovascular risk Auton. Neurosci. 158 92–9
Komorowski M 2019 Artificial intelligence in intensive care: are we there yet? Intensive Care Med. 45 1298–300
Kriechbaumer T, Jorde D and Jacobsen H-A 2019 Waveform signal entropy and compression of whole-building energy datasets Proc. Tenth ACM Int. Conf. on Future Energy Systems (New York: ACM) pp 56–67
Lautenschlager I, Filipiçep M, Kumlehn A and Akersberger J 2017 Chronix: long term storage and retrieval technology for anomaly detection in operational data 15th USENIX Conf. on File and Storage Technologies (FAST 17) pp 229–42
Lawrence N D 2017 Data readiness levels arXiv preprint arXiv:1705.02245
Lee H-C and Jung C-W 2018 Vital Recorder—a free research tool for automatic recording of high-resolution time-synchronised physiological data from multiple anaesthesia devices Sci. Rep. 8 1527
Lemire D and Boytow L 2015 Decoding billions of integers per second through vectorization Softw. Pract. Exp. 45 1–29
Li Q, Mark R G and Clifford G D 2009 Artificial arterial blood pressure artifact models and an evaluation of a robust blood pressure and heart rate estimator Biomed. Eng. Online 8 13
Liu D, Görges M and Jenkins S A 2012 A review of the University of Queensland vital signs dataset: development of an accessible repository of anesthesia patient monitoring for research Anesth. Analg. 114 384–9
Liu Q, Ma L, Fan S-Z, Abbod M F, Lu C-W, Lin T-Y, Jen K-K, Wu S-J and Sheib J-S 2018 Design and evaluation of a real-time physiological signals acquisition system implemented in multi-operating rooms for anesthesia J. Med. Syst. 42 148
Lobianco A 2019 Interfacing Julia with other languages Julia Quick Syntax Reference (Berkeley, CA: Apress) ch 7, pp 91–112
LZ4 https://github.com/lz4/lz4 Accessed: 2 July 2019
Manikandan M S and Soman K 2012 A novel method for detecting r-peaks in electrocardiogram (ECG) signal Biomed. Signal Process. Control 7 118–28
Mann J K, Kaffashi F, Vandendriessche B, Jacono F J and Loparo K 2020 Data collection and analysis in the ICU Neurocritical Care Informatics (Berlin: Springer) pp 111–34
MariaDB https://mariaodb.org/ Accessed: 2 July 2019
Maslow D M, Lamontagne F, Marshall J C and Heyland D K 2017 A path to precision in the icu Crit. Care 21 79
Matam B R and Duncan H 2018 Technical challenges related to implementation of a formula one real time data acquisition and analysis system in a paediatric intensive care unit J. Clin. Monit. Comput. 32 559–69
McGhee B H and Bridges E J 2002 Monitoring arterial blood pressure: what you may not know Crit. Care Nurse 22 60–79
McPadden J et al 2019 Health care and precision medicine research: Analysis of a scalable data science platform J. Med. Inform. Res. 21 e3043
Moody G B 2019 WFDB Applications Guide 10th edn (Cambridge, MA: Harvard-MIT Division of Health Sciences and Technology)
Nagaraj S, Goodwin A, Goodfellow S D, Greer B, Eytan D, Goldenberg A and Mazwi M 2019 A novel approach to machine learning-based automated vascular catheter access detection in a pediatric critical care setting Joint Paediatric Critical Care Int. Conf. Meeting, London, United Kingdom, November, 2019
Naylor C D 2018 On the prospects for a (deep) learning health care system Jama 320 1099–100
Nichols D and Hanrahan S 2007 Pressure control ventilation Crit. Care Clinics 23 183–99
Nizami S, Green J R and McGregor C 2013 Implementation of artifact detection in critical care: A methodological review IEEE Rev. Biomed. Eng. 6 127–42
Novak V, Novak P, de Champlain J, Le Blanc A, Martin R and Nadeau R 1993 Influence of respiration on heart rate and blood pressure fluctuations J. Appl. Physiol. 74 617–26
Orphanidou C 2019 A review of big data applications of physiological signal data Biosprf. Rev. 11 83–7
Pelkonen T, Franklin S, Teller J, Cavallaro P, Huang Q, Meza J and Veeraraghavan K 2015 Gorilla: A fast, scalable, in-memory time series database Proc. VLDB Endowment 8 1816–27
Penzel T et al 2016 Modulations of heart rate, ECG and cardio-respiratory coupling observed in polysomnography Frontiers Physiol. 7 460
Philips 2015 Data Export Interface Programming Guide—IntelliVue Patient Monitor and Avalon Fetal Monitor—X2, MP Series, MX Series, FM Series Technical Report (www.medical.philips.com) PHP: Hypertext Preprocessor https://php.net/ Accessed: 2 July 2019
RabbitMQ https://www.rabbitmq.com/ Accessed: 2 July 2019
Rehm G B, Kuhn B T, Delplanque J-P, Guo E C, Lieng M K, Nguyen J, Anderson N R and Adams J Y 2017 Development of a research-oriented system for collecting mechanical ventilator waveform data J. Am. Med. Inform. Assoc. 25 295–9
Rodríguez A, Smielewski P, Rosenthal E and Moberg D 2018 Medical device connectivity challenges outline the technical requirements and standards for promoting big data research and personalized medicine in neurocritical care Mil. Med. 183 99–104
Rush B, Celi L A and Stone D J 2018 Applying machine learning to continuously monitored physiological data J. Clin. Monit. Comput. 1–7

Sahoo S S et al 2014 Heart beats in the cloud: distributed analysis of electrophysiological ‘big data’ using cloud computing for epilepsy clinical research J. Am. Med. Inform. Assoc. 21 263–71

Seward J 2000 The bzzip2 and libbz2 official homepage (http://sourceware.cygnus.com/bzip2/)

Shafer I, Sambasivan R R, Rowe A and Ganger G R 2013 Specialized storage for big numeric time series Presented as Part of the 5th $\$USENIX$\S$ Workshop on Hot Topics in Storage and File Systems

Singh B, Kaur A and Singh J 2015 A review of ECG data compression techniques Int. J. Comput. Appl. 116 975–887

Singh H et al 2019 Development of data dictionary for neonatal intensive care unit: advancement towards a better critical care unit JAMIA Open 00064

Snappy https://github.com/google/snappy Accessed: 2 July 2019

Stein P K 2013 Challenges of heart rate variability research in the icu Crit. Care Med. 41 666–7

Stonebraker M and Cetintemel U 2005 One size fits all: an idea whose time has come and gone IEEE 21st Int. Conf. on Data Engineering (ICDE’05) pp 2–11

Sutton J R, Mahajan R, Akbilgic O and Kamaleswaran R 2019 Physonline: An open source machine learning pipeline for real-time analysis of streaming physiological waveform IEEE J. Biomed. Health Inform. 23 59–65

Takla G, Petro J H, Doyle D J, Horibe M and Gopakumaran B 2006 The problem of artifacts in patient monitor data during surgery: a clinical and methodological review Anesth. Analg. 103 1196–204

Tonekaboni S, Mazwi M, Laussen P, Fytan D, Greer R, Goodfellow S D, Goodwin A, Brudno M and Goldenberg A 2018 Prediction of cardiac arrest from physiological signals in the pediatric ICU Machine Learning for Conf. pp 534–50

Trigo J D, Alesanco A, Martínez I and García J 2011 A review on digital ecg formats and the relationships between them IEEE Trans. Inf. Technol. Biomed. 16 432–44

Vohra D 2016 Apache avro Practical Hadoop Ecosystem (Berlin: Springer) pp 503–23

Wang F, Casalino L P and Khullar D 2019 Deep learning in medicine—promise, progress and challenges JAMA Intern. Med. 179 293–4

Wetzel R C 2018 First get the data, then do the science! Pediat. Crit. Care Med. 19 382–3

Xiao C, Choi E and Sun J 2018 Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review J. Am. Med. Inform. Assoc. 25 1419–28

Yang F, Tschetter E, Léauté X, Ray N, Merlino G and Ganguli D 2014 Druid: A real-time analytical data store Proc. 2014 ACM SIGMOD Int. Conf. on Management of Data ACM pp 157–68

Yang S, Stansbury L G, Rock P, Scalea T and Hu P F 2019 Linking big data and prediction strategies: tools, pitfalls and lessons learned Crit. Care Med. 47 840–8

Zaleski J 2016 Connected Medical Devices: Integrating Patient Care Data in Healthcare Systems. Taylor & Francis Group, Boca Raton, FL

Ziv J and Lempel A 1977 A universal algorithm for sequential data compression IEEE Trans. Inf. Theory 23 337–43

ZStandard https://facebook.github.io/zstd/ Accessed: 2 July 2019

Zlib www.zlib.net/ Accessed: 2 July 2019