Smart Health 9-10 (2018) 275–286

Contents lists available at ScienceDirect

Smart Health

journal homepage: www.elsevier.com/locate/smhl

A scalable realtime analytics pipeline and storage architecture for physiological monitoring big data

Valentina Baljak*, Adis Ljubovic, Jonathan Michel*, Mason Montgomery, Richard Salaway

University of Virginia Health System Analytics and Reporting Data Science, United States

ARTICLE INFO

Available online 7 July 2018

Keywords:
Data pipeline
IoHT
Big data
Physiological monitoring data
Real - time analytics
Healthcare

ABSTRACT

Physiological monitors produce data essential to patient care. While the data is routinely observed and used at the bedside, it is rarely recorded permanently. Majority of data, especially high-resolution EKG waveforms, is discarded immediately. As the roles of big data and analytics in medicine are evolving, the streaming output of physiological monitors offers a potential source of highly relevant data for decision making and research. The inherently high volume and velocity of physiological data pose unique practical challenges for collection, storage, and analysis. A successful solution has to enable consistent and constant connectivity for streaming and to provide adequate storage and access capabilities. The solution also needs to meet security and privacy requirements for medical data. We propose a scalable, distributed architecture that leverages open-source stream processing software to connect raw monitoring output to a file storage system and an integration with existing data warehouse and data retrieval systems. Analytics pipeline has a potential to provide real-time feedback to the clinicians, including critical event detection or even prediction. The combination of real-time analysis and distributed storage of physiological big data, previously discarded, now opens up possibilities for future applications relevant to both clinicians and researchers alike. We have built and tested the first version of the continuously streaming data pipeline, from bedside physiological monitors to the storage. The results are promising, with raw or analyzed and enriched data already finding their place in supporting a variety of research use cases.

1. Introduction

Healthcare has seen a surge in use of digitally connected devices, bringing to the forefront new facets to daily routines in patient's care decision making and communication. Big Data, connected devices and *internet of healthcare things* (IoHT) are changing the way of absorbing, analyzing and interpreting available information, and this information now comes at the speed and resolution never before possible. While digitally measured and recorded data adds a level of accuracy and opens up new opportunities for analysis, research and decision making, it also poses unique challenges for collection, storage, analytics and feedback delivery.

* Corresponding authors.
E-mail addresses: vb8n@hscmail.mcc.virginia.edu (V. Baljak), al7pu@hscmail.mcc.virginia.edu (A. Ljubovic), jdm9y@hscmail.mcc.virginia.edu (J. Michel), mtm3ad@hscmail.mcc.virginia.edu (M. Montgomery), rns7a@hscmail.mcc.virginia.edu (R. Salaway).

http://dx.doi.org/10.1016/j.smhl.2018.07.013
2352-6483/© 2018 Elsevier Inc. All rights reserved.

Application of IoHT programs and technologies have proven to bring benefits in areas of remote patient monitoring, wellness and prevention, and operations (Accenture, 2017; Islam, Kwak, Kabir, Hossain, & Kwak, 2015). While the value of implementing and utilizing IoHT is undeniable, barriers to broader adoption still exist. In practice, technology in use at medical institutions usually lags behind Big Data trends. While developing systems such as these we have to consider legal and privacy concerns, inherent to all medical data since any technology based on network connectivity potentially opens up points of attack. Lastly, the question of where do we limit the use of technology, most prominently predictive analytics, in medical decision making remains an open one.

We are witnessing one of the momentous shifts in the way healthcare uses available technologies and the wealth of data sources. This shift is brought by the change in the perception as well as the availability of technologies. The healthcare industry is being transformed by the interconnectivity of digital devices and apps and by the evolving ways in which people are communicating and absorbing available information. Medical communities are more and more focusing on broader aspects of health care, such as population health, prevention, and wellness. IoHT is an emerging way of thinking as much as it is an implementation of technology to deliver healthcare solutions.

While technology provides access to patients and data on a scale never before possible, it is the shift in perception that health care encompasses more than reactive diagnosing and caring for sick that drives the need for adopting and integrating these technologies into daily routines. Capability to store data for the research, ability to analyze data in real-time and quickly deliver feedback, possibility to combine and absorb information from the variety of sources is beneficial not only to patients but the medical and research institutions. Evident across the board are *improved outcomes of treatment, disease management, reduced errors*, enhanced patient experience, management of drugs and decreased costs (Patel, 2016). Finally, health presents as a resource for the community, and medical professionals are now able to provide wellness and preventive measures for the benefit of the population in the overall health improvement while reducing healthcare costs.

Once a hype, relying on *big data* has become a norm of decision making in businesses and other organizations. The term refers to large data sets with heterogeneous formats, structured or unstructured, complex in nature and produced at a high rate. Coming from the varied sources, big data require powerful technologies and advanced algorithms. We can now observe and record data not only large in volume but also at a minuscule time and spatial scale. Phones, watches, fitness trackers, cars, all of these devices are adding to the more conventional devices used in medicine. Collecting and storing this level of information is one challenge, another being the ability to access, analyze and deliver feedback in a timely manner, where timely now means in real time.

Traditional data technologies and techniques lose some of the efficiency when faced with big data context. New technologies are emerging to fill the gap and enable efficient handling of data at this scale (Oussous et al., 2017). Amongst others, Apache Software Foundation is at the forefront of big data technologies with solutions ranging from data streaming to real-time predictive analytics and storage, such as Hadoop. Spark, Mahout, and Kafka. Most of these technologies are open source and provide a solution to storage analytics and continuous data streaming. These technologies can be combined to create efficient streaming and processing platforms. Here we present an architecture solution for physiological monitoring data collection, storage, and analytics. The core data pipeline is based on versatile and lightweight Kafka messaging platform. Our system provides integration with existing data warehouse (DW) and electronic health records (EHR) system, with an ultimate goal to offer real-time predictive analytics and feedback to medical staff in daily operations.

2. Data pipeline and storage architecture

The reality of an interconnected world affects healthcare providers in profound ways. More and more of our health is digitally monitored, in one form or another. Physiological monitoring at the bedside remains one of the critical sources of information for clinicians on a daily basis. The usual use of this data is intermittent observation; however, data is never stored or used beyond that, for further analysis or research. Recently, University of Virginia (UVA) has introduced new physiological monitors, and we have seen an increase in demand for capturing, storing and analyzing data from clinicians and researchers alike. To address this demand, the UVA Health Information and Technology team has developed an architecture for processing physiological bedside monitoring data. We aim to support data collection and archiving from the entire hospital, within the constraints that come with the nature of this data, such as the need to ensure privacy and security.

2.1. Physiological monitoring data

Physiological data provide crucial information for immediate healthcare action as well as for long-term health management. Studies show that specific changes in vital signs indicate, and even predict adverse outcomes, long after the event has occurred. Data streaming from bedside monitors have not been used for more than on-spot patient state observations even though they have a significant value as a resource for clinical and academic research. New bedside monitoring devices produce high volume, high-velocity data capturing patient's vital signs and heart rate waveforms. Clinicians and medical researchers are aware of a potential wealth of information at their fingertips that can ultimately help provide better care and faster response to adverse events.

Quality of healthcare metrics relies on scores such as National Early Warning Score (NEWS) (McGinley & Pearse) and Acute Physiology and Chronic Health Evaluation (APACHE) (Knaus, Zimmerman, Wagner, Draper, & Lawrence, 1981), which in turn use metrics based on vital signs, e.g., body temperature, respiratory rate, heart rate, blood pressure and oxygen. Real-time access, with a capability to calculate and analyze this data in the context of other available information, could significantly improve the accuracy of these scores and metrics used to make healthcare decisions, even to point out patients in critical need of immediate action. With the goal to support data-driven, real-time decision making for clinicians on the hospital floor and for the research, both clinical and academic, we have built an architecture for capturing, archiving, retrieving and analyzing complex high - resolution time series data.

2.2. Architecture overview

Physiological data qualifies as big data based on high velocity, high volume, multi-dimensionality and source diversity. Streaming, storing and integrating this type of data with existing infrastructure imposes significant challenges. Our system had to satisfy inherent big data requirements and legal as well as privacy requirements for medical data protection. Furthermore, the system had to integrate with existing data warehouse and EHR seamlessly. Additional constraint we had to address was the available hardware and software infrastructure. Due to security reasons, we have developed a solution based on existing and open-source platforms, hosted completely in-house.

Required functionality includes collecting metrics from sensors on the patient, e.g. vital signs, calculating early warning scores and indicators of health such as hypertension or arrhythmia, triggering alarms for decompensation (bradycardia, apnea, hemorrhage) and ultimately providing basis for prediction of disease and injuries caused by treatments, such as acute kidney injury, sepsis or atrial fibrillation. Predictive analytics should be available on both, streaming and stored data. To support this functionality, system, at the minimum, needs to provide:

- Constant and consistent real - time physiological data streaming
- Real - time reliable data capture
- Integration with existing data warehouse
- Integration with patient records (EHR - Epic)
- Exploratory analysis
- Predictive modeling
- Feedback for early warning systems
- Research algorithms deployment
- Research data portal with options to browse and download data and create datasets

UVA Medical Center uses GE bedside monitors continually streaming two distinct types of data. Both of these types are data of high volume and velocity; however, they have different structure and resolution, and thus require different processing, storage, and access utilities.

EKG Waveforms represent the output readings from EKG (Electrocardiogram, or ECG), and are used to calculate and monitor a variety of heart function indicators, such as RR interval, or heart rate. We receive readings in proprietary GE format, and transform it into 2-byte arrays, from up to 12 EKG leads at 240 Hz or 120 Hz resolution. The system processes and stores approximately 500.000 messages per minute.

Vitals or vital signs, such as blood pressure, body temperature, heart rate and respiratory rate, are indicators of body's life-sustaining functions. Up to 80 different vitals and settings stream from bedside monitors as single numerical values at 0.5 Hz rate. Currently, approximately 100.000 messages are processed and stored in the database.

Based on current operations, we anticipate the storage capacity, hospital-wide, at around 23TB per year at full capacity or 20TB per year at standard expected capacity.

Fig. 1 shows data pipeline and storage architecture, including current operational solution, as well as the support for future expansion of data sources and analytics pipeline. Already in operation, data access point is at high-speed digital interface (HSDI) device, namely, GE' s Carescape Gateway (GE Healthcare). Data format at the source is GE proprietary binary format, with a header describing the type and structure of data, and sensor output values. The data is then processed, decoded and repackaged for publishing to Kafka pipeline within Java-based connector application. The transformation for publishing includes decoding data from the proprietary format and encoding a Kafka key-value message. Each key contains identifying information about the source (bed, unit, patient), time and type of a recorded measure. The value comprises 2-byte arrays that represent either a unique measurement value for vitals or a time series chunk for EKG waveforms. The streaming pipeline itself contains different Kafka topics (channels) for different types of data. Next, data is read from these topics, decoded and stored. Vital signs are stored directly into the database, while waveforms are temporarily stored in an intermediary file format, initially pipe delimited text. However, JSON format has proven to be much more flexible for processing; it is faster for parsing, well supported and provides cleaner organization of file storage. An HDF5 output format is supported as well if required; however, it has proven to be a much slower output solution. Finally, files are indexed and verified against existing databases and stored in the final file storage, compressed JSON files. The benefit of this storage solution is faster access to data based on key search conditions, and it lends itself well to web-based tools for display and download. The intention is to provide a simple, adaptable and portable file format, easily parsed and transformed when

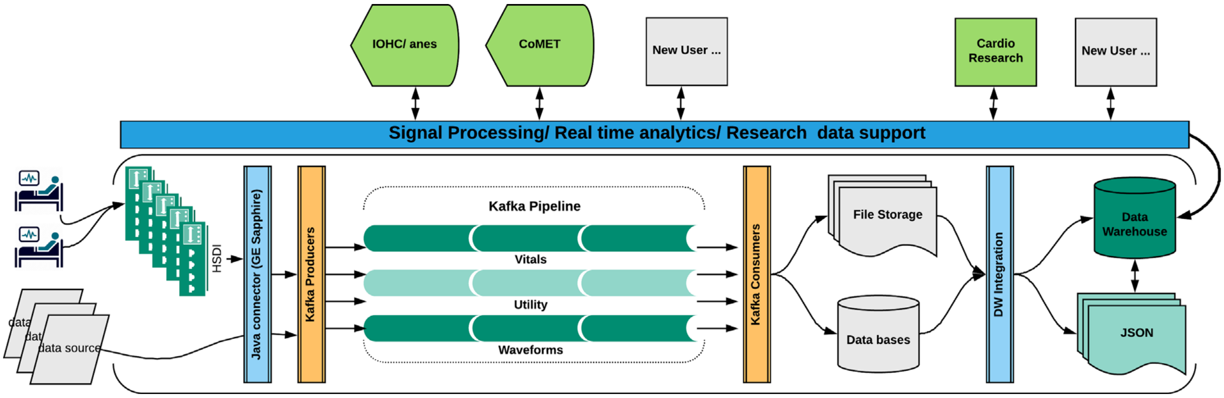


Fig. 1. Data pipeline and storage architecture.

needed. Additional data sources are EHR and data warehouse. We anticipate further expansion to include remote healthcare devices and other viable sources, such as personal trackers or different types of medical devices. Finally, the top layer consists of modules for signal processing, predictive analytics and access points for a variety of use cases, most prominent of which are data downloads for supporting medical research.

The actual hardware architecture is shown in Fig. 2. This architecture is currently in the operational testing phase. We have reached $>85\%$ of active data collection time, with $\approx 15\%$ of time reserved for testing, improvements and unexpected outages. We expect the system to be fully functional by the end of the year, and it complies to all security and configuration requirements in alignment with the rest of information technologies standards enforced at UVA Health System. Kafka cluster and file storage will be hosted on new servers, with necessary redundancy in place. Vital signs database and file indexing tables are currently running in a test environment and will be fully integrated into existing data warehouse. Due to limitations in hardware infrastructure availability, initially, processing will be hosted on the same servers as Kafka service. This solution comes with functional vulnerabilities, e.g., if one Kafka node fails, it might bring down all processes hosted on the same server. Standard prevention measures are set in place; however, for the upcoming upgrades, we intend to develop more robust solutions. Data collection and storage pipeline is operationally independent of the actual bedside monitoring. Failures and stoppages in the pipeline do not affect immediate patient care and safety in any way. By its intent and nature, this architecture is a one-way data streaming platform, and it ensures that failures do not propagate backward.

2.3. Data collection and streaming

Core functional requirement for this system is continuous and reliable large-scale data streaming from bedside monitors to the permanent storage. Our solution is based on Kafka, a distributed streaming and message handling platform (Apache Software Foundation). Kafka was developed at Linkedin for managing their large-scale time-oriented data sets. Linkedin contributed the Kafka message handling and queuing software as an Apache open source project.

Starting as a log processing platform (Kreps, Narkhede, & Rao, 2011), Kafka has become a go-to technology for any system that needs to collect and deliver high volumes of data at low latency and with minimal loss. The project aims to provide a unified, high-throughput, low-latency platform for real-time handling of data feeds, as Wikipedia page states. Characteristics that make Kafka so adept at handling high volume and velocity data include fault-tolerant records streaming and processing as records occur. More importantly, the key concept that sets Kafka apart is a scalable publish/ subscribe message queue architected as a distributed transaction log. In practice, Kafka is a scalable, distributed system that functions as a publish-subscribe service. One of the fundamental concepts is a Kafka topic. A topic is by its nature a dedicated channel where data can be published and read, with in-built latency period during which any subscriber can go back and collect the data. Two core API are Producer and Consumer. These API encapsulate the core functionality, publishing and consuming data, respectively. Scalability means that Kafka cluster can support an arbitrary number of consumers and producers, and it can be extended both through software and added hardware. Not bound by any hard-coded data format, Kafka is language and data-type agnostic, thus providing a potent tool for handling diversified data.

UVA's implementation of physiological data streaming platform utilizes Kafka to manage streams of high - resolution, time series sensor data. For example, EKG transducer is collecting data from a patient, together with related metadata. This data is received at the source, transformed into Kafka-messages and published to an appropriate topic. The transformation creates a message in the form of key-value. Actual data types of either one of them are not relevant and are a matter of choice and need for the specific system. In our case, we need to handle at least two different types of data, with different scale and format. Kafka topics enable us to separate these data and process them accordingly. EKG waveforms are received in quarter seconds bursts, with arrays of 30 - 60 values for each unique message key which is formed by patient's ID, location, EKG lead, timestamp and block sequence number. Vitals, on the other hand, are recorded at a 2-second frequency, with one value per vitals type. A unique key is formed correspondingly, with name and type of vital sign included.

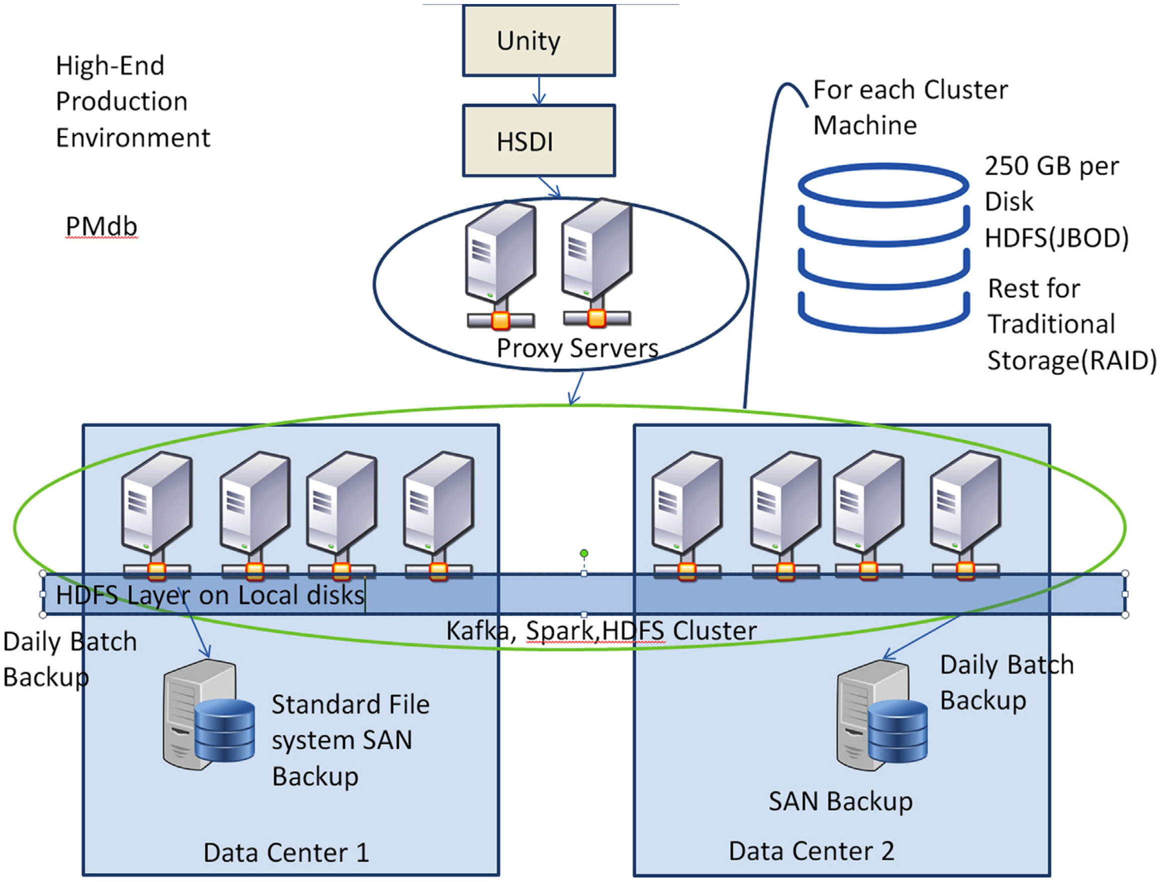


Fig. 2. Overview of the planned physical architecture.

One example of data flow looks like this: the bedside monitor sends ADT (admission, discharge, transfer) data record when a patient is placed in a bed, and the bedside monitoring system starts, a signal is then sent to the network that indicates a new patient session is starting. The UVA IoHT automatically acquires the new patient information, logs it, and begins to collect the vitals and EKG sensor streaming data.

EKG waveforms consist of continuous streams of high-resolution data, and they pose the highest demand on the speed of processing and storage capacity. At the 240 Hz rate, with at least 7 sensory leads and 512 beds to monitor, the primary challenge is to absorb this data and record it into intermediary files as quickly as possible. Even at our current scale of a single consumer, we are able to do this in real time. Records are processed as they occurred and stored in a temporary file format, with one file representing one patient per bed per day. A separate process is used to verify the information against the data warehouse, and to index files. Finally, once the information is reconciled with the existing data in the warehouse, and verified, files are converted into final storage file format.

Our solution is built around Kafka pipeline. As a starting point, we have written a connector Java application that absorbs data from HSDI source in the native binary format, transforms data into appropriate Kafka message formats, consisting of a key and values, where a key is the identifier of a record, e.g., the type of vital sign. Transformed data is then published by a Kafka producer application to an appropriate topic. In our system, two main topics are dedicated to vital signs and EKG waveforms. Dedicated consumer applications read data from assigned topics, process them and store them in previously described manner. In the current version of the system, there are usually at least two consumers active, one for waveform data and one for vitals. As needed, we run additional consumers, e.g., for storing waveforms in a different file format. The key to reliable, fast and lossless data streaming, collection, and storage lies in described characteristics of Kafka. Our current solution is still well below Kafka's throughput power. Potential lags in consuming and processing data from the topic need to be addressed within consumer and processing applications. For example, writing data to the files has proven to be a time-consuming process. To avoid the bottleneck, we have separated processing and compression of files from consuming and data output. While we do have one additional process to get all the data into the file storage, we have avoided output latency, with reading capacity being four times faster than real time. While the system is in the normal operation mode, this ensures real-time output to as many as 500 files at the same time. In the case of lag, this speed would allow the system to catch up to the current data in 1/4 of the lag time.

The upcoming extended version of the system will aim to utilize parallelization capabilities of Kafka, such as topic partitions and consumer clusters to a full extent. A resulting significant improvement in data throughput would provide a platform with power to incorporate other potential data sources and to increase processing capabilities.

2.4. Data storage

Vital signs (vitals) data, like heart rate and blood pressures, has been a mainstay in health care for decades. Historically, health care providers monitor inpatients by periodically checking on their patient's state. Additionally, Anesthesiologists heavily rely on vitals monitoring to provide feedback on the patient's state and adjust accordingly to assure vitals stay within normal range. While these use cases have proven to be effective, they fall short of current technological capabilities seen in other industries. With the real-time infrastructure we have developed for streaming and storing vitals we will be able to:

1. Perform 24/7 real-time monitoring vs. the standard periodic check-ups by health providers, thus improving detection of potential deterioration in patients.
2. Monitor data over time, with insight into change trends instead of relying only on the current patient's state.
3. Develop algorithms that predict adverse outcomes.
4. Enable 24/7 real-time alerting system set by customizable triggers.

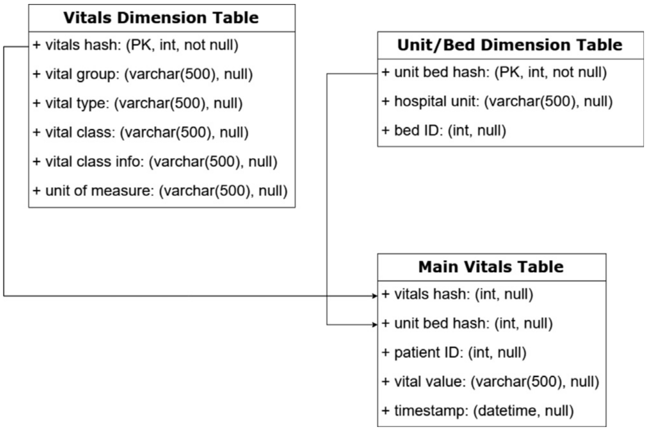
In contrast to vitals, waveforms are prohibitively large to be stored directly in the database. Instead, we store them in the files and index the location of files in the database, so the main database design concerns are with storing vitals, minimizing the size while providing the speed of access.

2.4.1. Vitals – Design Goals

The real-time vital pipeline we built was designed to meet two main goals. First, it needs to be easily accessible by internal software developers, data scientists, and business analysts. Second, it needs to have latency no more than 1 second. Accessibility to software developers was easy, as we would give them access only to the Kafka API to develop applications using vitals data. To make it accessible for data scientist and business analysts, we decided to work within the existing environment and integrate the vitals pipeline to our existing SQL Server database. Following the internal logic enables sophisticated users to include vitals data in analytics and prediction datasets and reports in the same way as any other data.

2.4.2. Vitals – Design Details

Each vital record comes with the following data points: *Vital Group, Vital Type, Vital Class, Vital Class Additional Info, Unit of Measurement, Vital Value, Patient Id, Hospital Unit, Bed ID, and timestamp*. Approximately, there are at least 100+ million rows per day, so it is of the essence to minimize the size of each row. The main optimization we did was to make dummy dimensions of related columns. The related columns would be hashed, and placed in a dimension table with its hash value. In the main vitals table, the related columns would be replaced with only the hashed value of those columns. For example, the columns (Vital Group, Vital Type, Vital Class, Vital Class Additional Info, Unit of Measurement) were replaced with the hash value of those columns. This transformation results in a considerable improvement in storage space, as these vitals columns took up 120 B per row as compared to the hash value which takes up only 4 bytes per row. That's a massive 30X improvement in storage. The same technique was to reduce (Hospital Unit, Bed ID) into the hash value of those columns. As a result, our final vital table schema looks like shown in Fig. 3.



Vitals Dimension Table
+ vitals hash: (PK, int, not null)
+ vital group: (varchar(500), null)
+ vital type: (varchar(500), null)
+ vital class: (varchar(500), null)
+ vital class info: (varchar(500), null)
+ unit of measure: (varchar(500), null)

Unit/Bed Dimension Table
+ unit bed hash: (PK, int, not null)
+ hospital unit: (varchar(500), null)
+ bed ID: (int, null)

Main Vitals Table
+ vitals hash: (int, null)
+ unit bed hash: (int, null)
+ patient ID: (int, null)
+ vital value: (varchar(500), null)
+ timestamp: (datetime, null)

Fig. 3. Example of database structure for storing vital signs.

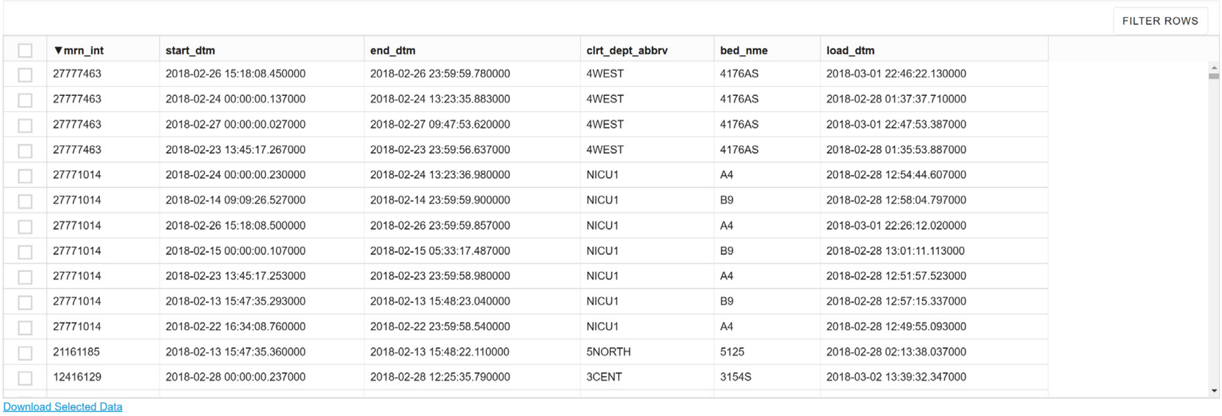
Another optimization that we made was making (Patient ID) an integer instead of varchar, assigning the minimum varchar length to the (Vital Value) column, and using page compression. With all these optimizations, we managed to reduce the row size to 18 bytes per row. The write speed into the database is less than a second. However, the query speed was slower than required due to a large number of rows (100 M+ per day). To resolve this issue, each day of data was partitioned, and the table was implemented to use “dirty” reads. This feature improved query speed to less than a second and met our design requirement.

2.5. Data retrieval, visualization and Delivery

Data discovery and retrieval are facilitated through the implementation of a web-based application. This visual interface allows users to find desired data and discover new data through filtering the database by relevant features as well as previewing data analysis performed on data of interest. Once identified, desired data may be downloaded to the user's local machine via the interface's download buttons. The Kafka process described above outputs daily batches of files that follow an informative naming convention and contain the recorded values of streaming monitoring data. The creation of these files initiates a new process designed to compress the original file to approximately 25% of the original storage size and index the final file storage location with metadata about the data collection session (e.g., date, patient ID number, medical unit, patient condition). The metadata provides additional features on which the user may filter to find desired datasets.

The image in Fig. 4 shows the table provided in the web interface where users may filter files on relevant features. Each column provides an additional means of sorting or filtering data in aggregate. Selection boxes on the left allow users to select specific datasets for downloading, previewing the underlying data or previewing on-demand data analysis. Preliminary data analysis is presented for any dataset selected. The application performs a set of analysis algorithms on each selected dataset and displays the output of this preliminary data analysis along with a visual representation of the underlying data itself. The streaming monitoring data contains values recorded from 10 or more monitor nodes, or leads, on the electrocardiogram (EKG) monitor. Depending on the specific use case, analysis algorithms may be performed on any or all of the monitor lead signals. However, for the purposes of the data analysis preview provided by the web application, data analysis is only performed on the signal recorded from the monitor's lead denoted as lead II, which is commonly used when a strong signal-to-noise ratio is desired (Campbell et al., 1985; Garson, 1993; Moss, Schwartz, Crampton, EH, & Carleen, 1985). One calculation commonly performed on data recorded by an EKG monitor is a determination of the R-R intervals. These intervals are an instantaneous and consecutive measure of the time observed between peaks characteristic of a specific phase of the heartbeat cycle (de Chazal, O'Dwyer & Reilly, 2004 ; Sloan et al., 2007). Taken in aggregate, analysis of these R-R interval observations provide insight into the underlying phenomena. Common analysis techniques include observation of the distribution of R-R intervals and spectral analysis of the R-R series by fast Fourier transform (FFT). For each user-selected dataset of EKG data, the application computes the R-R intervals, summarizes the distribution, and performs FFT signal analysis all on-demand, in real time. The visual output from the R-R interval analysis is shown, along with a visual representation of the associated lead II signal, as part of the data analysis preview for each selected dataset. The image in Fig. 2 depicts a sample data analysis preview generated from the selection of a single dataset. Additional analysis outputs can be overlaid on the charts for additional dataset selections.

Fig. 5 shows results of a signal analysis. The signal from EKG monitor leads and the corresponding R-R intervals constitute continuous signal series. Their values are instantaneous with respect to time and independent of any temporal binning. However, analysis such as FFT signal analysis and R-R interval distributions require aggregate binning of a recorded signal. While the raw dataset may provide continuous observations over a full 24 hour period, users may be interested in an analysis performed on the temporal binning of data corresponding to a specific time (e.g., the 5 minutes of recorded data preceding a specific observed event). The advantage of the on-demand data analysis preview is that the user is able to



□	▼mrn_int	start_dtm	end_dtm	clrt_dept_abbrv	bed_nme	load_dtm
□	27777463	2018-02-26 15:18:08.450000	2018-02-26 23:59:59.780000	4WEST	4176AS	2018-03-01 22:46:22.130000
□	27777463	2018-02-24 00:00:00.137000	2018-02-24 13:23:35.883000	4WEST	4176AS	2018-02-28 01:37:37.710000
□	27777463	2018-02-27 00:00:00.027000	2018-02-27 09:47:53.620000	4WEST	4176AS	2018-03-01 22:47:53.387000
□	27777463	2018-02-23 13:45:17.267000	2018-02-23 23:59:56.637000	4WEST	4176AS	2018-02-28 01:35:53.887000
□	27771014	2018-02-24 00:00:00.230000	2018-02-24 13:23:36.980000	NICU1	A4	2018-02-28 12:54:44.607000
□	27771014	2018-02-14 09:09:26.527000	2018-02-14 23:59:59.900000	NICU1	B9	2018-02-28 12:58:04.797000
□	27771014	2018-02-26 15:18:08.500000	2018-02-26 23:59:59.857000	NICU1	A4	2018-03-01 22:26:12.020000
□	27771014	2018-02-15 00:00:00.107000	2018-02-15 05:33:17.487000	NICU1	B9	2018-02-28 13:01:11.113000
□	27771014	2018-02-23 13:45:17.253000	2018-02-23 23:59:58.980000	NICU1	A4	2018-02-28 12:51:57.523000
□	27771014	2018-02-13 15:47:35.293000	2018-02-13 15:48:23.040000	NICU1	B9	2018-02-28 12:57:15.337000
□	27771014	2018-02-22 16:34:08.760000	2018-02-22 23:59:58.540000	NICU1	A4	2018-02-28 12:49:55.093000
□	21161185	2018-02-13 15:47:35.360000	2018-02-13 15:48:22.110000	5NORTH	5125	2018-02-28 02:13:38.037000
□	12416129	2018-02-28 00:00:00.237000	2018-02-28 12:25:35.790000	3CENT	3154S	2018-03-02 13:39:32.347000

Fig. 4. Web application - files download tool.

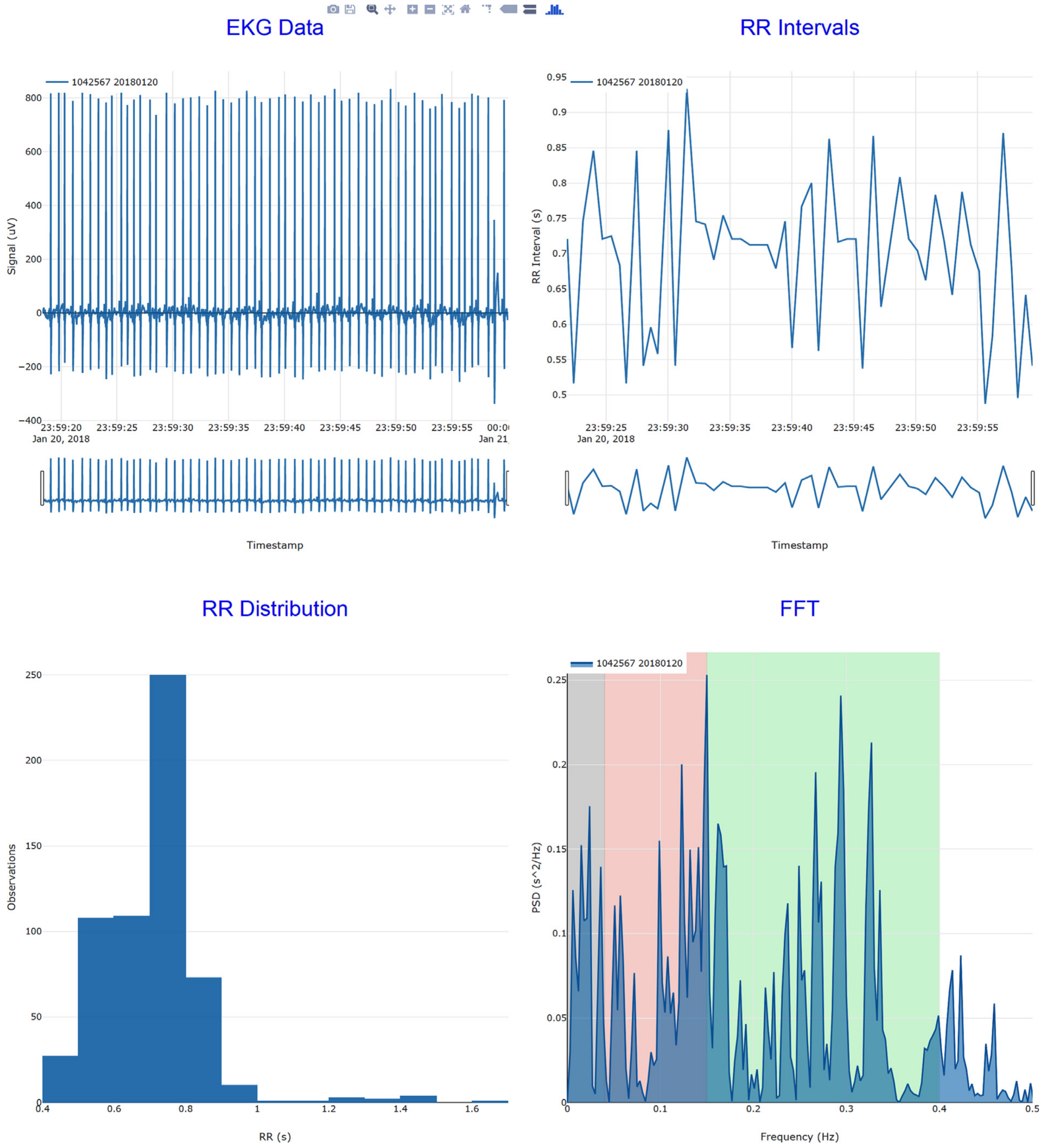


Fig. 5. Web application - EKG analytics.

explore selected data before saving to a local machine. Temporal range selectors on the EKG and R-R interval charts in Fig. 2 allow the user to select a specific subset of the continuous signal. This temporal selection automatically updates the R-R interval and FFT signal analysis previews so that the output reflects the analysis performed only on the data contained in the selected temporal range. In this way, users can search datasets for events or characteristics of interest with an interactive, real-time, analysis workbench. When events of interest or characteristic data are identified, the user has the option of downloading only the subset of data selected. In this way, the user can avoid management of big data files when only a targeted subset of data is the focus. The web-based application provides the user with an interactive, responsive interface for efficient filtering, exploration, and retrieval of data. The summation of these capabilities quickly connects the user to the most relevant data, while excluding the distraction of extraneous data.

2.6. Analytics and signal processing modules

An EKG is a measure of electrical activity in the heart over time. The heartbeat and associated signals can be broken down into a series of actions and waves. First, the atria contract to fill the ventricles, the P wave. Then the ventricles contract to send blood to the body and lungs, the QRS complex. The QRS complex contains the most distinctive feature of the EKG, the R peak, which can be used to measure heart rate. Finally, the ventricles repolarize and relax before the cycle repeats itself, the T wave. Each wave carries information which can indicate pathologies, such as the absence of the P wave indicating atrial fibrillation or similar conditions of the atria (EKG practicaltraining,trai).

We use the Python packages numpy, scipy.signal, and peakutils (Negri, 0000) to process the EKG data. The basis for this algorithm is modified approach to analyzing and classifying EKG (Clay, 2017). We first smooth the signal baseline using numpy's flatten function to reduce baseline drift. With the baseline smoothed, the signal can be read by peakutils to find the R peaks. The time between R peaks can be converted into an instantaneous heart rate. The P and T waves are far more challenging to detect algorithmically because of their lower amplitude and greater variability in pathology. With this initial EKG signal processing module in place, we further plan to implement machine learning techniques to label the T and P waves as accurately as current methods allow, which is between 92 and 96 percent accuracy (Jambukia, Dabhi, & Prajapati, 2015). In addition to analyzing historical data for research purpose, we intend to explore the possibility to apply signal processing to EKG waveforms in real-time for possible clinical use.

3. Applications for IoHT

Two main areas of application for the IoHT system is in operations, *quality and safety in patient care, and research in improving patient care*. These two areas have helped focus the design of the system, with some commonalities between them, as well as the differences, and have driven the service levels that we have developed to meet.

Real-time automated monitoring of patients in acute care settings is becoming more prevalent. Scores such as NEWS and APACHE II, utilize measures based on vital signs, e.g., body temperature, respiratory rate, heart rate, blood pressure and peripheral capillary oxygen saturation (SpO2). These scores are increasingly used by hospital physicians to monitor patients. Real-time monitoring systems such as the UVA developed COMET (Blackburn et al., 2018) system take in real-time data to compute an aggregated score and show the relative risk and progression of the patient. These systems are being developed to manage and prevent the decompensation during a hospital stay.

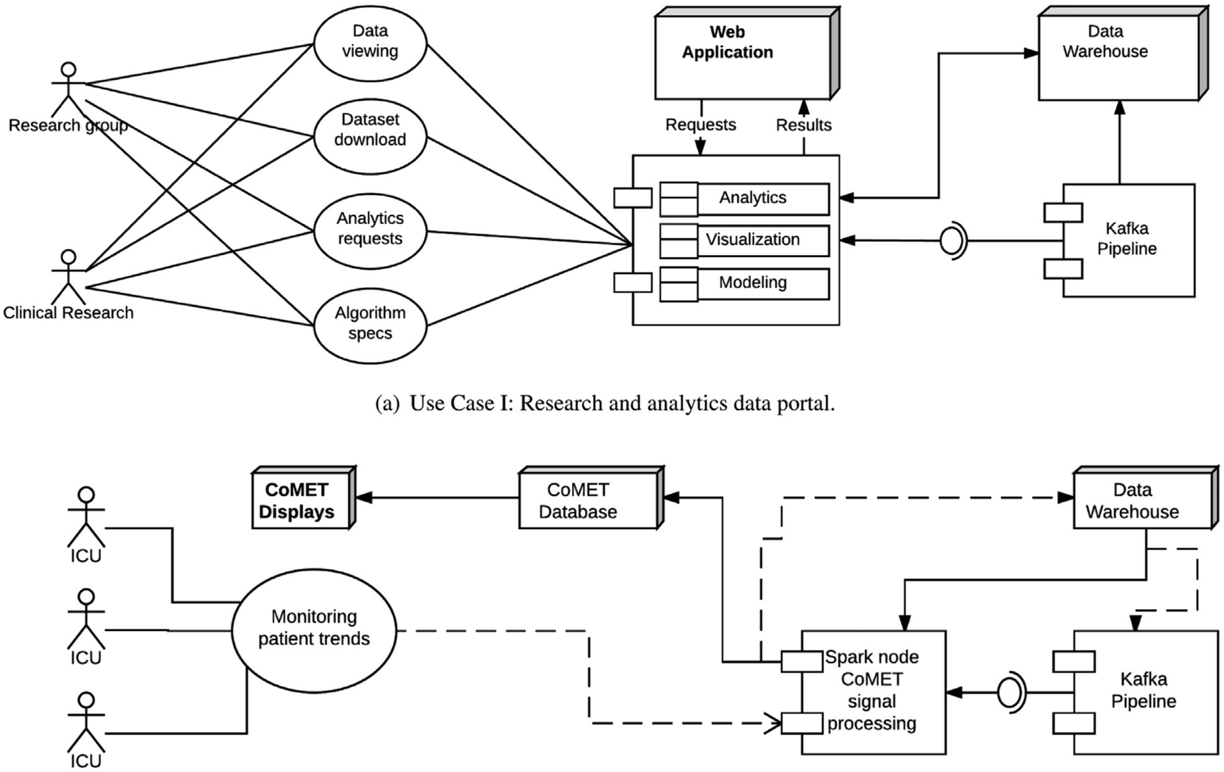
Another of the high-value uses of the streaming data comes from a problem during surgery. The length of time a patient remains hypotensive while under anesthesia is positively correlated with an increase in risk for acute kidney injury (AKI) (Sun, Wijeysundera, Tait, & Beattie, 2015). The intervention designed is to have a module monitoring the real-time patient's vitals during the surgery, accumulating the time under a specified blood pressure value. This module will transmit back a message to the anesthesiologist indicating the amount of time where the blood pressure is below a determined value.

An existing system being tested within the surgery and trauma ICU is a system called Continuous Monitoring of Event (CoMET). CoMET is a product being developed and tested by the AMP3D Company in conjunction with the University of Virginia Medical Center Cardiologist, Randall Moorman. In the past, the CoMET team relied on single monitor applications built by 3rd parties to stream the data to their system. The CoMET system integrates the streams and displays the patient information on the monitor. The IoHT platform gives them the ability to pick up any bed within the hospital and apply their CoMET monitoring software to the vitals and EKG. This will eliminate the need for individually licensed software to receive the data, increase the speed of access and refresh rates to near-real-time and include a possibility for further analytics.

Ventilator Adverse Events (VAE) are hospital-acquired infections that increase mortality and length of stay. In collaboration with the pulmonary specialist at UVA medical center, new methods for identifying VAE risk are being developed, including new methods for more rigorously identifying adverse events at an earlier stage. The ventilator system feeds data through the physiological monitor system and is part of the streaming data available in the IoHT feed. Both the configuration settings and the dynamic patient needs settings are available for analysis. This data can be used to identify the progression of the patient and accurately identify or even predict these events.

Fig. 6 shows examples of high-level interaction diagrams for some of the use cases identified in practice. In Fig. 6(a) we show an anticipated most common user interaction with the system. A researcher or a clinician accesses data portal through the web application, searching for either a specified patient data or setting parameters for creating a dataset. Depending on the request, our system collects, connects, analyzes and displays requested data. Results, as well as raw data, are then available for the download in several data formats (JSON, HDF5, etc.). We anticipate that web application will mostly be used for browsing and downloading historical data; however, there might be a need for more interactive real-time analytics. In Fig. 6(b), we can see an example of a real-time use of data pipeline. CoMET monitoring system would be able to either access data stream in the Kafka pipeline directly or through analytics modules.

Each of use cases we have identified so far have their own unique set of requirements and challenges, e.g., IOH requires prompt feedback to a one, specific display, while CoMET will allow for higher latency but demand access to multiple patients' data. On the other hand, most of the academic research use cases will demand better analytics and access to broader integrations with other data sources into detailed data sets for further analysis as well as the access to raw data in downloadable format. In practice, what this means for our system, is that we can roughly split the access demand to



(a) Use Case I: Research and analytics data portal.

(b) Use Case II: Real-time feed to the CoMET decompensation monitoring system and display.

Fig. 6. High level interaction diagrams for some of the identified use cases.

historical, stored data and access to the real-time data stream. Our platform strives to provide both and the use of Kafka as a core data pipeline enables us to do precisely that.

As IoHT is gaining ground, not only as a technology but as a way of approaching health care, our platform will serve as a basis for further expansion and integration of other data sources such as personal devices, bringing access to medical information to an unprecedented level. Medical professionals will have a tool that enables the depth of insight into patient's record's, resulting from analyzing correlations between data, and even including prediction of immediate outcomes.

4. Results and future work

As of the time of writing, the first version of data pipeline is in operation at UVA Health System. With minor interruptions for development and maintenance, we have been continuously streaming data from, on a daily average, 400 beds since the beginning of December 2017, bringing us to 6 months of continuous operation. Data collection is in real-time, data is immediately stored in intermediary files (waveforms) or database (vitals). Files are processed on a daily basis, indexed and reconciled with the data warehouse. Also, our data is already in active use for research, e.g., neurology research into a possible connection of cardio events to epileptic seizures, currently mostly on demand and historical data. Already, by replacing third-party data acquisition solutions, we are saving time and money for the UVA Health System, up to estimated 2 million per year.

While the collection and storage system has its vulnerabilities and there is a need for maintenance and outage downtime, patient care and safety are never affected by this. The architecture supports one-way streaming, with no back-propagation of failures. This means that on-site bedside monitoring is never affected by any failure, bottleneck or outage that might happen in our system.

At current capacity, our system meets the demand for collecting, processing and storing data streaming from bedside monitors. The platform itself maintains the flexibility and capacity to include other data sources and to provide more sophisticated predictive analytics capability.

The next version and expansion of the platform are already in preparation. A new server is ready to be added, and an expanded Kafka cluster will be deployed, with an anticipated minimum of three processing nodes. File storage is being expanded as well, and it will include a number of SAND storage drives capable of supporting the growth of approximately

20TB per year. Web application as a data portal will be a general access point for researchers and users. While main functionality is in place, we are currently working on full database access, download formats, and potential analytics modules. Visualization of data will undergo some changes to comply with existing standards but also include more interactive capabilities.

Automation of analytics modules is the next step, and we are investigating Apache Spark, Kafka Streams and KSQL and several other solutions for development and deployment. The idea here is not to limit analytics to several known algorithms and models, but to enable development and deployment of new and advanced algorithms that would support a variety of research. Our guideline, supported by technologies that we have already adopted, is to build a flexible and scalable system. This scalability includes both hardware and software. We are optimistic that our platform will become a go-to source of high-definition data and quality information.

5. Conclusion

Physiological monitoring data provides vital information for daily patient care. With the emergence of big data technologies and increased awareness that healthcare extends beyond medical institutions, we have come to understand the need to collect and store this data. In it' s high-resolution raw form, physiological data, especially EKG waveforms, is invaluable to diverse medical research, such as heart conditions or even neurological conditions. A multitude of vital signs, on the other hand, establishes a basis for early deterioration detection, including a potential to develop clinical alerts. Combined, and integrated with other data sources, existing databases, and EHR, physiological data creates a basis for potential advanced predictive analytics, both, in real-time as well as research on historical data.

Our team has built the data streaming and analytics pipeline and storage architecture capable of capturing high volume, high velocity, heterogeneous data from bedside monitors. We leverage open source big data technologies, with Apache Kafka messaging platform at its core, to collect, enrich store and integrate large-scale physiological data. Kafka has provided the solution for streaming data due to its adaptiveness and independence of data types. It is a potent messaging platform capable of losslessly handling large amounts of data.

The next unique challenge, storing raw EKG waveforms, has been solved through the use of file storage, where files are indexed and integrated with the existing data warehouse. Vital signs are being streamed and stored directly in the database. Our system also provides a capability to transform and analyze data, most notable example being RR interval calculations. The first version of the platform is in use at UVA Health System. The initial results are not only promising, but the data is already in use for various ongoing medical research. We aim to expand this platform to automate real-time analytics modules and feedback delivery system and to provide an extensive research data portal and advanced algorithms deployment module. We believe that this platform will provide a robust basis for incorporating any additional data sources that require Big Data treatment. Security and privacy requirements for sensitive medical data introduced practical challenges to the choice of technologies, hardware infrastructure, and networking solutions. While it has added certain limitations, it has also provided an opportunity to test and prove that advanced big data technologies can successfully integrate with existing technologies and infrastructures and that such an architecture can provide a way for medical institutions to utilize the full potential of data at their disposal.

Acknowledgment

For their extensive contributions to the architecture and software, we want to thank Ian German Mesner, Chief Software Architect at ChartIQ, Ryan Bobko, System Architect at Ostrich Emulators, LLC and Kim Scott, Co-founder and VP of Data Science at Astraea, Inc. Their work has been indispensable at critical stages of this project. For their genuine support and enthusiasm for using our data, we would like to extend our gratitude to Dr. J. Randall Moorman, MD, Dr. Douglas Lake, MD, Dr. Biken Naik, MD, Dr. Andrew Schomer, MD, Dr. John Odackal, MD and Dr. Taison Bell, MD, all of the University of Virginia. Their vision for possibilities this scale of data brings to the research, and the clinical application continues to inspire and drive our efforts. Finally, this project would not be possible without the support, ideas and substantial efforts of Daniel O'Malley, Director of Data Operations and Glenn Wasson, Administrator of Analytics, both at ART UVA Health System.

Conflict of interest statement

None.

References

Accenture, (2017). Internet of health things survey, White Paper (May 2017). https://www.accenture.com/us-en/insight-accenture-2017-internet-health-things-survey.

Apache Software Foundation, Apache Kafka, a distributed streaming platform: Introduction. ⟨https://kafka.apache.org/intro⟩.

Blackburn, H. N., Clark, M. T., Moorman, J. R., Lake, D. E., & Calland, J. F. (2018). Identifying the low risk patient in surgical intensive and intermediate care units using continuous monitoring. *Surgery*

Campbell, R. W. F., Gardiner, P., Amos, P., Chadwick, D., & Jordan, R. (1985). Measurement of the qt interval. *European Heart Journal*, *6*, 81–83.

Clay, E. (2017). Analyzing the electrocardiogram (ecg) and classifying what's healthy and what's not., Speect at PyData London 2017 (May). ⟨https://pydata.org/london2017/schedule/presentation/43/⟩.

de Chazal, P., O'Dwyer, M., & Reilly, R. B. (2004). Automatic classification of heartbeats using ecg morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering*, *51*(7), 1196–1206.

EKG practical training. ⟨http://www.practicalclinicalskills.com/ekg⟩.

Garson, A. (1993). How to measure the qt interval - what is normal? *The American Journal of Cardiology*, *72*, 14B–16B.

GE Healthcare, Carescape gateway. URL ⟨http://www3.gehealthcare.com/en/products/categories/patient_monitoring/networking/carescape_gateway⟩.

Islam, S. M. R., Kwak, D., Kabir, M. H., Hossain, M., & Kwak, K. S. (2015). The internet of things for health care: A comprehensive survey. *IEEE Access*, *3*, 678–708.

Jambukia, S.H., Dabhi, V., Prajapati, H. (2015). Classification of ecg signals using machine learning techniques: A survey (03 2015).

Knaus, W., Zimmerman, J. E., Wagner, D. P., Draper, E., & Lawrence, D. E. (1981). Apache-acute physiology and chronic health evaluation: A Physiologically Based Classification System. *Critical Care Medicine*, *9*, 591–597.

Kreps, J., Narkhede, N., Rao, J. (2011). Kafka: a distributed messaging system for log processing.

McGinley, A., & Pearse, R. M. A national early warning score for acutely ill patients, BMJ 345.

Moss, A., Schwartz, P. J., Crampton, R. S., EH, L., & Carleen, E. (1985). The long Qt syndrome: A prospective international study. *Circulation*, *71*, 17–21.

Negri, L., Peakutils: Python package for EKG peak calculations. ⟨https://bitbucket.org/lucashnegri/peakutils⟩.

Oussous, A., Benjelloun, F.-Z., Lahcen, A. A., & Belfkih, S. (2017). Big data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*

Patel, K. (2016). 6 benefits of iot for hospitals and healthcare (Jul. 2016). http://readwrite.com/2016/07/18/top-6-benefits-internet-things-iot-hospitals-healthcare-facilities-ht1/.

Sloan, R., McCreath, H., Tracey, K., Sidney, S., Liu, K., & Seeman, T. (2007). Rr interval variability is inversely related to inflammatory markers: The cardia study. *Molecular Medicine*, *13*(3–4), 178–184, http://dx.doi.org/10.2119/2006-00112.Sloan.

Sun, L. Y., Wijeysundera, D. N., Tait, G. A., & Beattie, W. S. (2015). Association of intraoperative hypotension with acute kidney injury after elective non-cardiac surgery. *Anesthesiology*, *123*(3), 515–523.