Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 220 (2023) 511–518

Procedia Computer Science

www.elsevier.com/locate/procedia

The 14th International Conference on Ambient Systems, Networks and Technologies (ANT)
March 15-17, 2023, Leuven, Belgium

PHAS: An End-to-End, Open-Source, and Portable Healthcare Analytics Stack

Mahyar Abbasian[a,*], Elahe Khatibi[a], Iman Azimi[a,b], Amir M. Rahmani[a,b,c]

[a]*Department of Computer Science, University of California, Irvine, CA, USA*
[b]*Institute for Future Health, University of California, Irvine, CA, USA*
[c]*School of Nursing, University of California, Irvine, CA, USA*

Abstract

In today's binary world, digital health is of paramount importance, primarily due to the prevalence of IoT devices, upsurging health costs, growing elderly population, and shortage of clinical providers, to name a few. In light of these, providing efficient and smarter full-stack healthcare data analytics to manage and process healthcare data is a crucial topic from both academic and professional perspectives. The key goals of this full-stack healthcare data analytics are to detect health issues and promote human-being health proactively. The existing healthcare data analytics stacks are generally classified into commercial or open-source solutions. In designing a healthcare data stack, it is critical to offer the researchers a collaborative, modular, easy-to-use, cost/time-effective, reproducible, uniform, and shared- knowledge framework. Such healthcare stacks need to pave the way for researchers to focus on developing data analytics algorithms, and the underlying infrastructure should comply with longitudinal characteristics of healthcare data. Nonetheless, the existing healthcare data analytics stacks need holistically incorporate all the parameters mentioned. In this paper, we propose a novel healthcare data analytics stack called Open-Source Portable Healthcare Analytics Stack (PHAS) to address this issue. PHAS considers the mentioned features by fusing the merits of open-source and commercial solutions at the right place in its architecture. PHAS proposes a new shared-knowledge and time-series-aware framework, which enables researchers to perform health data collection, integration, storage, visualization, and analysis. Moreover, we demonstrate the capabilities of the PHAS framework by implementing an anomaly detection algorithm for heart rate and blood pressure anomaly detection using the Medical Information Mart for Intensive Care III (MIMIC III) dataset. We provide open-source PHAS for the community to integrate into their solutions.

Keywords: Healthcare; Data Analytics Stack; Machine Learning; Framework;

* Mahyar Abbasian. Tel.: +1-949-992-5016 .
E-mail address: abbasiam@uci.edu

10.1016/j.procs.2023.03.065

1. Introduction

Digital health world witnesses unfettered growth due to the proliferation and accessibility of the Internet-of-Things (IoT) and wearable technologies, enabling providers to capture physiological, societal, and behavioral data ubiquitously. In addition, studies show the significance of preventive healthcare services due to the growth of at-risk patient populations in the near future. The need for these health services and the advances in big data collection necessitates artificial intelligence (AI)-based data analysis approaches with the ability to extract meaningful information from the data. In this regard, an efficient healthcare data analytics full-stack is required to accelerate data analytics development in digital health [20].

A healthcare data analytics stack is an integrated system that gathers, integrates, transforms, analyzes, and visualizes health data. The healthcare stack solution provides infrastructures and tools, enabling knowledge-sharing and collaborations to avoid reinventing the wheel. Existing studies in the literature have introduced and evaluated a variety of healthcare data analytics stacks. The state-of-the-art healthcare stacks are mainly categorized into two classes: i.e., commercial and open-source software solutions [12, 9]. The commercial healthcare stack provides a fully-implemented stack in three respects: infrastructure, platform, and application. However, they fail to offer a shared collaborative space. For example, IBM Watson provides a health analytics stack using pre-built components, but other researchers need support to contribute or exploit the setup in their solutions easily. Commercial health products are also complex to be learned or costly to be used in several applications. In contrast to the commercial stack, open-source healthcare products enable researchers to contribute to building their own healthcare solutions. Although open-source products partially mitigate the flaws of commercial products by offering a shared-knowledge space, they suffer from a lack of integration and high complexity.

In this paper, we introduce an Open-Source Portable Healthcare Analytics Stack (PHAS) to provide a fully collaborative, reproducible, easy-to-use, temporal-aware, and cost-effective framework. We first delve into the state-of-the-art healthcare stacks in the literature. Then, we present five key components required for healthcare stacks analytics, including PHAS. The components allow researchers to perform health data collection, integration, storage, visualization, and analysis. Moreover, we demonstrate our proposed framework using a case study on anomaly detection for heart rate and blood pressure. We provide open-source PHAS for the community to be integrated into their solutions[1].

The remainder of this paper is outlined as follows. In Section 2, we describe related work. Section 3 outlines the health stack architecture. Section 4 presents the proposed PHAS framework. Section 5 discusses the experimental results, and Section 6 concludes the paper.

2. Related Work

In healthcare data analytics, researchers are often willing to focus on data-analytic algorithms development rather than software engineering aspects. In addition, they generally intend to share their research outcomes as an application. Furthermore, research teams often need cost-effective toolchains with minimal repetitive work, easy-to-use (less required knowledge to use the tools), fast local set-up with minimum adjustment at the end-user side, and collaborative environment [13, 10, 12].

We focus on classifying the existing health analytics stacks with respect to researchers' needs. We classify healthcare analytics stacks into two major groups. The two main groups are i) large-scale commercial healthcare analytics stack and ii) open-source healthcare analytics.

2.1. Large-scale Commercial Healthcare Analytics Stacks

This group offers large-scale commercial software platforms for big data analysis, which could be deployed in diverse fields, such as education, health, e-banking, and communication. It encompasses four sub-groups: i.e., On-premises, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software/Application as a Service. The focus of healthcare products is mostly on either PaaS or SaaS. PaaS is a cloud computing model in which cloud providers offer a platform to the developers whereby the developers can build their applications. SaaS is a cloud-based method of delivering applications over the Internet as a service [12, 15, 18].

Products falling under the PaaS sub-group entail merits, including a fully implemented end-to-end stack in three respects: infrastructure, platform, and application. Furthermore, for enterprise solutions in large-scale projects, the entire end-to-end pipeline can be set up swiftly. Amazon Web Service (AWS) [1], Microsoft Azure [2], and IBM Watson [6] are in the PaaS sub-group. The second sub-group, namely SaaS, demonstrates the same advantages. SaaS

[1] https://github.com/HealthSciTech/PHAS

can be built on top of the PaaS (e.g., AWS marketplace) or implemented separately on an infrastructure provided by PaaS providers. BioT [3] is a secure and intelligent healthcare software solution that uses real-time geolocation data from hospital devices. It can offer predictive healthcare recommendations. Flywheel [4] is a multi-modal research data and collaboration software solution for the healthcare sector, specializing in images.

Although the commercialized category offers advantages, both its sub-groups suffer from some pitfalls. The pricing plans are often not affordable for research teams. SaaS products show less flexibility to be adjusted and optimized for each client's requirements/conditions or environment. In other words, such SaaS products fail to comply with each project's goals or concerns, and researchers are restricted to merely a set of already available algorithms, tools, and platforms in such pipelines [9].

Moreover, commercial products lack to offer a shared knowledge space. For this group, it is harder to create a collaborative environment from a developmental perspective. To use such products/tools, researchers require extensive software engineering and cloud computing knowledge [12].

2.2. Open-source Healthcare Analytics Stacks

To address some of the issues of the first category —large-scale commercial health stack— research teams strive to build their healthcare analytics stacks. These products offer open-source modularized components in diverse levels: core/architecture and application levels. We categorize such products into three distinct classes: i.e., platform, protocol/guidelines, and framework.

1. Platform: Platform is the first sub-group, which puts forth an already developed architecture/core to address specific goals.

The Platform group enables integration, reduces repetitive work, and partially establishes a knowledge-sharing space compared to commercialized products. The contributors in the platform group are limited to the core's features and architecture. If users need specific features, they should either request the new features from the developers or contribute to altering the core/architecture concerning their goals. Therefore, they can add their customized features. In this group, the core development team is in charge of the decision- making processes about design updates or adding new features. Concerning contribution, complexity is moderate in this subgroup, and researchers need to learn the skills and knowledge required to modify the system.

Personicle [16], RADAR-Base [19] and MD2K [8] are classified under this sub-group. Personicle, a chronicle of life events, is a multi-modal evolving personal situation recognition system. It collects and analyzes the stream of personal data to offer personalized, actionable insights. RADAR-Base, which is a modular application, attempts to provide an off-the-shelf platform in the mHealth area for remote data collection to promote patient health. MD2K uses cutting-edge algorithms, platforms, and software to pave the way for biomedical researchers to boost human health under big data tools. MD2K is a part of the National Institutes of Health (NIH) Big Data to Knowledge (BD2K) initiative.

In this group, the core development team is responsible for the decision-making process about design updates or adding new features.

2. Protocol/general guidelines: This sub-group calls for a broad contribution from the scientific/engineering community in both architecture/core and application development levels.

The Protocol group is a collaborative, modular, and shared knowledge environment, thus reducing repetitive tasks. The Protocol group provides the opportunity to replicate the previous work in an easy and well-defined fashion. Knowledge sharing is more convenient in this group than in the platform group. The Protocol group allows the contributors to pursue their own rules; therefore, this approach may arrive at non-uniform components ultimately. Hence, they may suffer from a lack of coherent integration due to heterogeneity and broad collaboration of contributors worldwide at different levels.

In addition, already existing powerful tools cannot easily be integrated into the guideline nowadays. Therefore, there will be some levels of re-inventing the wheel for some of the features provided. For example, Kibana, Grafana, and d3.js are powerful visualization tools and libraries that cannot be easily integrated into such pipelines or require extra developers' work. Due mainly to high-level contributions, the protocol group has more complexity, as the researchers need to gain extensive knowledge of diverse parts of the system to be able to exploit them. Moreover, deploying this paradigm in a local setup is difficult for researchers. Digital Biomarker Discovery Pipeline (DBDP) [11], which is an open-source pipeline for end-to-end digital biomarker development, belongs to the Protocol group.

3. Framework: Framework is the third sub-group, which is a hybrid solution that establishes an end-to-end healthcare stack. This group reduces time and complexity of deployment. The main goals of the framework group are to maximize knowledge sharing, decrease repetitive tasks, enable local deployment, provide partial integration, and reduce complexity in terms of learning, usage, and contribution from researchers. To this end, the framework group

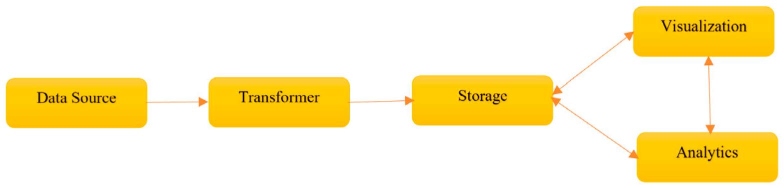

Fig. 1. Architecture of Healthcare Data Analytics Stacks

offers fundamental architectures. ELK Stack [22] is a collection of three open-source products — i.e., Elasticsearch, Logstash, and Kibana — and is classified into the Framework group. ELK offers a powerful platform that collects and processes data from miscellaneous data sources and provides a set of visualization tools as well. ELK does not consider the temporal (time-series) aspects of healthcare data, because Elasticsearch is a text-based analysis tool. Therefore, it is not devised for intensive longitudinal data analysis. Moreover, researchers face the burden of learning both Elasticsearch and Logstash query language to be able to use ELK.

3. Conventional Architecture of Healthcare Data Analytics Stacks

We performed a comprehensive review of the existing healthcare data analytics stacks. Our findings indicate that most of the existing solutions encompass five main components: i.e., *Data Source*, *Transformers*, *Storage*, *Visualization*, and *Data Analytics*. A view of the reference architecture and the components are shown in Figure 1. In the following, we briefly outline the components [13].

Data Source: This component captures data from diverse channels, such as sensors, wearable devices, health records, and third-party datasets. The collected raw data are then transferred to the following components for aggregation and analysis.

Transformer: The Transformer implements pre-processing techniques, such as raw data cleaning, data curation, reformatting, aggregation, and de-identification. Heterogenous data, collected from disparate sources, often requires additional pre-processing to be prepared and unified for downstream components, such as Visualization and Analytics. Furthermore, the same information gained from a handful of different sources should be fused and unified to be considered input for data analytics. For example, the heart rate of an individual might be collected via two different wearable devices in a health monitoring application. The device's heart rate values need to be unified/merged for further processing [15, 10].

Storage: The data received from Transformer are stored in a file system, a memory, or a database. Researchers often deal with a large volume of multi-dimensional data in the healthcare domain, for which efficient ways are required to access and query the data. Storing data in the memory provides fast access to all the data, leading to a large memory footprint. Storing the data in a file system supports the large volume required and is human-readable in many cases. However, querying the data is drastically slow. Using the database in most cases is the best choice because many databases with different capabilities can support large-scale data and fast data access and query.

Visualization: Visualization provides dashboards and tools for Data Visualization in different formats (e.g., charts). It is essential for the healthcare data analytics stack, as researchers require to perform exploratory data analysis to gain insights from the data (particularly temporal data) before developing Machine Learning (ML)- or Artificial Intelligence (AI)-based models. For example, this component enables heart rate ascending/descending trend demonstration, which is required for health anomaly detection applications.

Data Analytics: This component applies different analytics techniques to the data, such as finding hidden patterns, discovering causation, pinpointing correlations, and deriving valuable insights to make predictions about future events or trends, not to mention helping to make more informed decisions. The outcome of the Data Analytics component could be demonstrated as visualization, prediction, or stored as a separate data stream [21, 14].

These five components are crucial to set up a healthcare analytics stack. These components are provided differently in the commercial and open-source categories. Some components are limited to the provider implementation, and others are flexible to customize or develop. The way that these five components are provided in both categories and their sub-categories are briefly explained as follows.

Researchers are provided with pre-built tools to create each component in the commercial category- PaaS group. For example, IBM Watson offers a customized database called IBM DB2 as the Storage component. IBM Watson also provides pre-implemented data cleaning, data transferring, and security plugins that can be connected to the Storage component. These data processing tools act as part of the Transformer component. Furthermore, IBM Watson contains various analytics tools and methods as their Data Analytics component. Researchers can use the provided methods or develop their ML/AI algorithms. A Business Analytics solution exists for the Visualization component, which the researchers can customize. Nevertheless, the IBM Watson plat- form lacks the Data source component. Therefore, users need to develop their data collection methods. In addition to the PaaS, in the SaaS group, all five components are already implemented, and they only provide some levels of freedom in either configuration or customization of each component. For example, users can develop their algorithms for the Data Analytics part [6].

These five components are already implemented for the open-source category - Platform group. However, the developers can contribute to adding new features. For example, if there are insufficient Data Sources or Visualization components' implementation for a specific use case, developers must contribute to implementing the necessary parts. In this group, the contribution is mainly at the component development level. Still, the main developer team (the platform creators) should decide, for example, what database to use, how those five components interact with each other, and which features should be provided. Therefore, the researchers' contribution is limited to the provided design. Moreover, in the open-source Protocol group, a general guideline is presented in high-level abstraction rather than the technical details. For instance, the protocol group does not compel the developers to opt for specific data storage or programming languages. At last, in the open-source framework group, some components are partially implemented, the design is already decided, and other parts are left to researchers to contribute. For instance, in ELK, the Logstash serves as the Transformer component which is a comprehensive tool for all kinds of transformation with its query language. The Storage component in ELK is enforced to be Elastic Search, and the Visualization component is selected to be Kibana with capabilities to create customized visual components such as charts [22].

4. The Portable Healthcare Analytics Stack (PHAS) Framework

In this paper, we propose a new healthcare data analytics framework named PHAS. PHAS offers the minimum fundamental core for the discussed components, whose main goals are retaining integration, reducing repetitive tasks, providing a shared-knowledge environment, and maximizing contribution to developing data analytics algorithms. As an illustration, we enable the usage of specific tools such as influxDB [7], and Grafana [5] as Storage and Visualization components, respectively. PHAS empowers researchers to accelerate the development of their algorithms toward healthcare data analytics.

The primary purposes of PHAS are re-usability, collaboration, low complexity, abstraction–all, integration, and time-series data awareness. The framework not only satisfies the integration but also offers less complexity and more flexibility in terms of both learning and using the framework. Moreover, it provides a collaborative environment, thus enabling sharing of the codes and already-created visualized panels by diverse researchers.

We will discuss the components of PHAS and their distinctive features in the following. First, we describe how we have devised the general five-component architecture and its distinguishing characteristics in PHAS (see Figure 2). Data Source is placed in the Transformer component in PHAS. Therefore, the researchers can write their own codes in Transformer to read and access the data. Then, they can apply methods, for example, to clean, reformat, unify, and merge health data. The Transformer component is a simple Jupyter file in which the researchers can write their Python codes for parsing files, reading the data from wearable devices, or fetching from a dataset. Users, therefore, can take advantage of the existing Python tools to transform the data and store them into the Storage component. We decided to put a Jupyter file for the Transformer to provide flexibility especially emphasising ease of use and minimal required development knowledge for users to start their research. Users are free to use any Python tools they are familiar with to fetch the data as well as transform them. Providing Jupyter also enables them to share their files (which may be their main contribution to some data analytics algorithms) easily with others as a separate component, thus preserving re-usability and contribution capabilities. For the Storage component, we have chosen the influxDB [7], as it works well on large-scale time-series data. We provide the required codes to interact with this database in our architecture. InfluxDB is open-source and is made especially for time-series data containing efficient time-related queries. It is suitable for applications, such as the Internet of Things (IoT) and real-time analytics. This database features visual query builder that helps make queries easier. Moreover, influxDB integrates easier with Grafana, which we will introduce as our Visualization component.

For the Visualization component, we have deployed Grafana: i.e., an open-source data visualization platform, allowing users to visualize their data using graphs or charts. Grafana is a fork of Kibana. Kibana is only integrated into Elastic Search, which is our main reason for choosing Grafana over Kibana. We exploit Grafana to enhance our

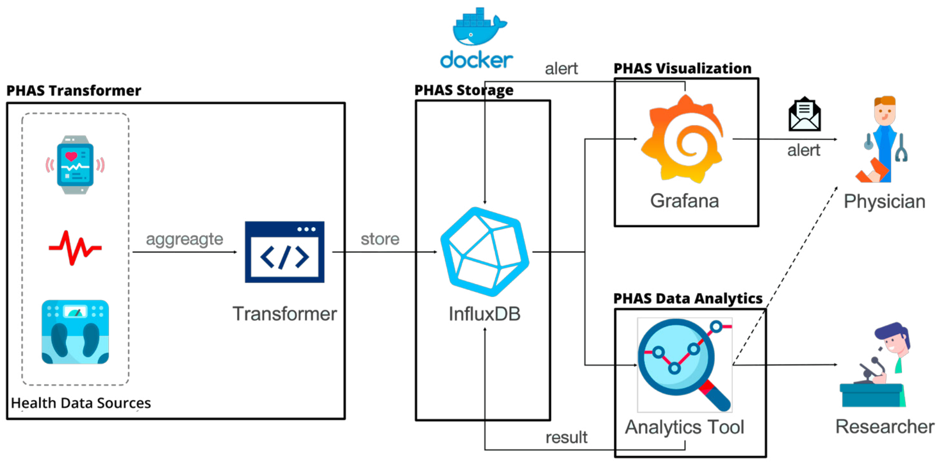

Fig. 2. PHAS Architecture

framework's strengths. A critical benefit of using Grafana is that it provides several pre-built dashboards and panels to be reused. This way, pre-existing knowledge in the visualization domain is used to create the panels time-efficiently. Moreover, we can make and share diverse panels and dashboards connected to influxDB to carry out exploratory data analysis. Researchers can create their own new panels or use other contributors' already-created panels to set up the visualization rapidly.

The final component is the data analytics component. We have added a Jupyter file in this component to read the required data from the database by which researchers can infuse their desired data and concentrate on their data analysis tasks. The same reasons mentioned in the Transformer component also apply here on why we decided to put a simple Jupyter file in this component and only provide the interfaces needed to interact with the database.

In the developing field of healthcare analytics stacks, the Transformers, Visualizations, and Analytics components are still active and open fields of research. In most health data analytics approaches, researchers first obtain the data, then transform and visualize it to gain insights about it, and spend considerable time creating different plots to investigate the relations between data points. Finally, they start implementing their data analytics algorithms. Our goal in proposing PHAS architecture is to ensure that these three components maintain flexibility while the end-to-end system maintains its integrity. Having a Jupyter file in Transformer and Analytics components helps researchers start using the Python tools they are already familiar with without extra effort. Furthermore, researchers are able to share their codes or contribute to others easily. Grafana is a modular visualization tool and encompasses many tools for visualization. We anticipate having a rapidly growing community sharing their knowledge to enable consistent progress in healthcare data analytics.

We have integrated all these components together in one docker container that is easy to run and set up locally or on a server. When this docker is run, all the components containing Jupyter lab, influxDB, and Grafana will be set up seamlessly. The container can also be pushed to a cloud or used on a local computer.

From the classification perspective, as we introduced in the related work section, PHAS is a member of the framework group. It is empowered by the strengths of the existing tools (such as Grafana and InfluxDB) to provide an environment for rapid healthcare data analytics development. Although we proposed PHAS as a healthcare longitudinal data analysis framework based on the needs of the healthcare analysis research area, it can also be used in other data analysis areas that include time-series data.

5. Case Studies

For the evaluation part, we implement anomaly detection of patients' heart rate and blood pressure as case studies. We use the Medical Information Mart for Intensive Care III (MIMIC III) dataset [17] and deploy the PHAS framework to illustrate the workflow and functionality of PHAS.

Data Source and Transformer: The MIMIC-III dataset is a large, de-identified, publicly available, and single-center collection of medical records relating to patients. We used heart rate and blood pressure data from MIMIC

III; in the MIMIC-III dataset, each vital sign data has an item-id for identifying each data; for instance, Heart-rate's item-id is 220045. However, Blood pressure has two types of data–systolic and diastolic, whose item-ids are 220179 and 220180. For our experiment, Data Source is a CSV file–extracted from the MIMIC III–comprising timestamp, heart rate, and systolic and diastolic blood pressure values. We implemented a Python script to read and parse this CSV file to pinpoint the item-id, related timestamp, and value for each heart rate and blood pressure. Afterward, the Python program creates a data stream from the parsed file and stores the data stream in the influxDB.

Data Storage: The generated data stream from the Transformer component is stored in the influxDB as time-series data for further analysis and visualization.

Data Analytics: We have implemented an anomaly detection algorithm for the Data Analytics component using the Interquartile Range rule (IQR)–which is able to detect the presence of outliers. In this component, we fetch the data from the influxDB using fast existing time-based queries. Then we apply the IQR with a moving window of a fixed size of 2 weeks. Finally, as a result of applying the IQR, labels will be generated for each data point, which will then be stored as a new data stream into the database.

Data Visualization: For the Visualization component, we provided different charts showing raw and analyzed data. Figure 3 shows the plots of the transformed data points directly in different time scales with varying charts. Figure 3.a shows the daily box plot of the heart rate data over a week. Figure 3.b shows the daily heart rate data points averaged hourly. The y-axis is 24 hours of the day, and the x-axis is the date. Figure 3.c also shows the daily box plot of the heart rate, and each point shows the number of measurements for each hour. Moreover, Figure 3.d depicts the average hourly data of systolic and diastolic, and Figure 3.e illustrates 12 hours average of heart rate and the population min and max.

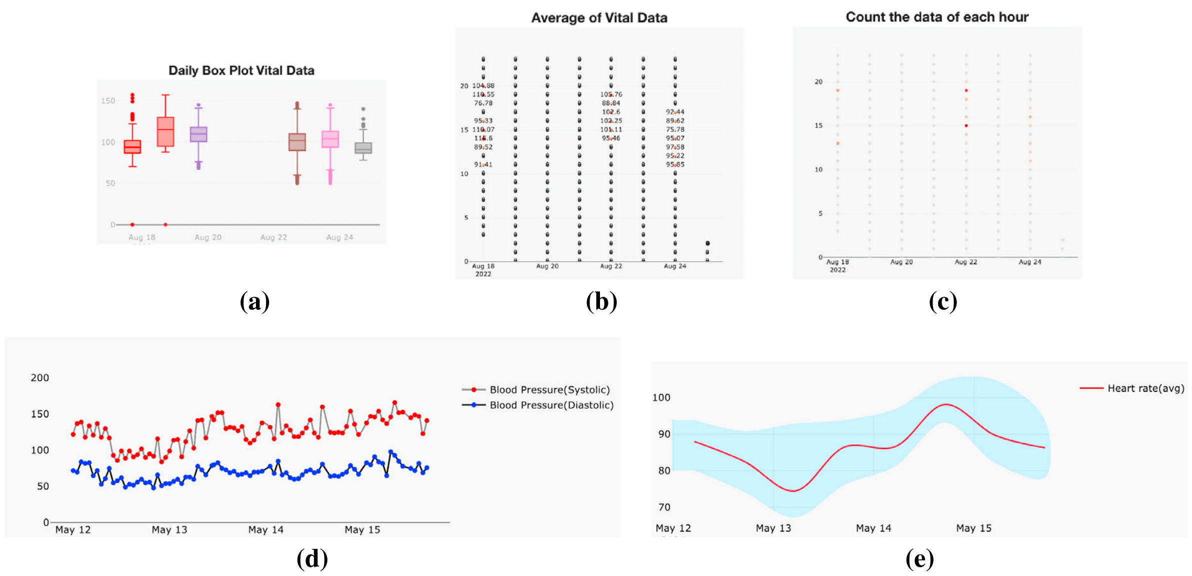

Fig. 3. **(a)** Daily Box Plot for Heart-rate of a patient **(b)** Daily heart rate data points averaged hourly **(c)** Count of the data points collected hourly every day **(d)** Hourly average data of systolic and diastolic **(e)** 12 hours average of heart rate and the population min and max

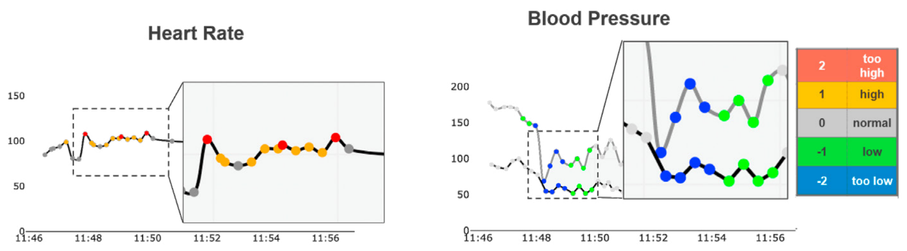

Fig. 4. (a) Anomaly Detection on Heart-rate (b) Anomaly Detection on Blood Pressure

Figure 4 contains a transformed data point and the results of the analysis. Figure 4.a shows the heart rate data points and their labels, and Figure 4.b includes both systolic and diastolic data points with their respective labels as a separate panel. As shown in Figure 4, we defined five different labels to show different types of anomalies. The charts in Figures 3 and 4 are represented as a separate panel in Grafana.

6. Conclusion and Future Work

Delivering an efficient healthcare data analytics full-stack is one of the main debilitating challenges in today's digital health. In this paper, we proposed a new healthcare data analytics stack called Open-Source Portable Healthcare Analytics Stack (PHAS). PHAS incorporated the merits of commercial and open-source health stack products while also paying attention to temporal characteristics of healthcare data and helping researchers to focus on developing data analytics algorithms. PHAS introduced a five-layer architecture and established the required data storage, processing, and visualization components for researchers. Furthermore, PHAS framework encompasses features, including a collaborative knowledge-sharing space, easy-to-use, reproducibility, and cost-efficiency. By employing PHAS, researchers can develop data analytics algorithms rapidly and efficiently. Our experimental results demonstrated the applicability and workflow of PHAS on the MIMIC III dataset focusing on a heart rate and blood pressure data anomaly detection application. We plan to include causal AI and event mining methods for future work to enhance our Data Analytics component. Moreover, we intend to involve Apache NiFi for the ingester part.

References

[1] AWS. https://aws.amazon.com/health/healthcare/solutions/, 2023.
[2] Azure. https://azure.microsoft.com/en-us/solutions/industries/healthcare/, 2023.
[3] BioT. https://www.biot-med.com/, 2023.
[4] Flywheel. https://flywheel.io/flywheel-is-data-discovery/, 2023.
[5] Grafana. https://grafana.com/, 2023.
[6] IBM. https://www.ibm.com/resources/watson-health/health-data-connect/#/, 2023.
[7] InfluxDB. https://www.influxdata.com/, 2023.
[8] MD2K. https://github.com/BD2K/MD2K, 2023.
[9] L. Adamowicz et al. Scikit digital health: Python package for streamlined wearable inertial sensor data processing. *JMIR mHealth and uHealth*, 10(4):e36762, 2022.
[10] B. Bent et al. Cgmquantify: Python and R software packages for comprehensive analysis of interstitial glucose and glycemic variability from continuous glucose monitor data. *IEEE Open Journal of Engineering in Medicine and Biology*, 2:263–266, 2021.
[11] B. Bent et al. The digital biomarker discovery pipeline: An open-source software platform for the development of digital biomarkers using mhealth and wearables data. *Journal of clinical and translational science*, 5(1), 2021.
[12] B. Bent et al. Digital medicine community perspectives and challenges: survey study. *JMIR mHealth and uHealth*, 9(2):e24570, 2021.
[13] Brinnae Bent. *Discovering Digital Biomarkers of Glycemic Health from Wearable Sensors*. PhD thesis, Duke University, 2021.
[14] R. Indrakumari et al. Heart disease prediction using exploratory data analysis. *Procedia Computer Science*, 173:130–139, 2020.
[15] Md S. Islam et al. A systematic review on healthcare analytics: application and theoretical perspective of data mining. In *Healthcare*, volume 6, page 54. MDPI, 2018.
[16] L. Jalali et al. Personicle: personal chronicle of life events. In *Workshop on Personal Data Analytics in the Internet of Things (PDA@ IOT) at the 40th International Conference on Very Large Databases (VLDB), Hangzhou, China*, 2014.
[17] A. Johnson et al. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.
[18] D. Rani et al. A comparative study of saas, paas and iaas in cloud computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(6), 2014.
[19] Y. Ranjan et al. Radar-base: open source mobile health platform for collecting, monitoring, and analyzing data using sensors, wearables, and mobile devices. *JMIR mHealth and uHealth*, 7(8):e11734, 2019.
[20] Md M. Shandhi et al. Recent academic research on clinically relevant digital measures: Systematic review. *Journal of medical Internet research*, 23(9):e29875, 2021.
[21] Trevor L Strome. *Healthcare analytics for quality and performance improvement*. John Wiley & Sons, 2013.
[22] D. Uday et al. An analysis of health system log files using elk stack. In *2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, pages 891–894. IEEE, 2019.