Future Generation Computer Systems 167 (2025) 107748

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Blending lossy and lossless data compression methods to support health data streaming in smart cities

Alexandre Andrade [a], Cristiano André da Costa [a], Alex Roehrs [a], Debora Muchaluat-Saade [b], Rodrigo da Rosa Righi [a,*]

[a] Applied Computing Program, Universidade do Vale do Rio dos Sinos, São Leopoldo, RS, Brazil
[b] Computer Science Department, Universidade Federal Fluminense, Niterói, RJ, Brazil

ARTICLE INFO

Keywords:
Vital Sign
Healthcare
Internet of Things
Smart city
Data compression

ABSTRACT

The digital transformation process has significantly boosted the widespread adoption of telemedicine and the utilization of wearable devices for vital signs remote monitoring. However, implementing a system for continuous monitoring of the population's vital signs, with data being streamed from various locations within a smart city context, faces significant challenges. These challenges are related to bandwidth consumption, communication latency, and storage capacity due to the large volume of data. To overcome these challenges, a common practice consists in modeling an edge-fog-cloud layered architecture. The literature lacks software solutions capable of managing the simultaneous transmission of various vital signs data from geographically distributed individuals while maintaining the ability to generate health notifications promptly. In this context, we propose the VSAC (Vital Sign Adaptive Compressor) model, which combines lossy and lossless data compression algorithms in a layered architecture to support healthcare demands in a smart city. The main contribution is how we blend both strategies: we first use lossy compression to collect only valuable vital sign data for everyone, applying lossless algorithms afterwards to reduce the number of bytes before sending it to higher layers. We provide a real-time processing protocol that facilitates the collection of heterogeneous data distributed across different city regions. After executing a VSAC prototype, the results indicate that orchestrating the aforementioned two data compression algorithms is more efficient than conventional data reduction methods. In particular, we obtained gains of up to 42% when measuring the compression rate metric.

1. Introduction

Digital transformation has been significantly advanced with the growing adoption of the Internet of Things (IoT). Integrating connected devices and sensors transforms various areas, such as industry, medicine, and agriculture. In this context, devices enable interaction between the physical and virtual worlds, turning ordinary objects into "smart" ones capable of collecting and transmitting data [1]. Transmission typically occurs through radio frequency-based networks, which can face issues such as packet loss and retransmission if not properly scaled. In terms of storage, the advantages of cloud computing are widely known for offering information security, ease of access, and recovery. However, we have financial costs associated with the need for appropriate data storage and retrieval strategies.

In the field of healthcare, IoT has a transformative impact, facilitating more personalized and proactive approaches to patient care. Wearable devices monitor conditions in real time, enabling faster and more accurate diagnoses. The COVID-19 pandemic accelerated this movement, allowing hospitals and healthcare institutions to monitor vital signs of individuals remotely. Instead of having a discrete approach based on regular time-interval observations, the integration of IoT and remote health systems allows continuous monitoring. This enables anticipating risks and rapid response to medical emergencies. These factors, if associated with the context of smart cities, offer the opportunity for technical innovations and social advances, with sensor networks that monitor public health and contribute to more effective public policies [2]. For example, applying predictive methods to collected data can reveal patterns of change in health conditions, anticipating problems before they worsen.

Efficient management of collected data is fundamental to the successful integration of IoT in healthcare, addressing challenges such as latency in data transmission and the need for effective compression to handle large volumes of information. Multilayer architectures, such as

* Corresponding author.
E-mail addresses: andrade@unisinos.br (A. Andrade), cac@unisinos.br (C.A. da Costa), roehrs@unisinos.br (A. Roehrs), debora@midiacom.uff.br (D. Muchaluat-Saade), rrrighi@unisinos.br (R. da Rosa Righi).

https://doi.org/10.1016/j.future.2025.107748
Received 19 September 2024; Received in revised form 23 December 2024; Accepted 1 February 2025
Available online 11 February 2025
0167-739X/

Edge-Fog-Cloud, process data closer to the source, reducing latency and network traffic [3]. In contrast, data compression techniques optimize storage and transmission by blending lossy and lossless methods to preserve the integrity of critical information [4]. Lossy compression achieves significant file size reduction by permanently discarding less critical data, introducing acceptable distortion. On the other hand, lossless compression ensures complete data preservation, allowing for the exact reconstruction of the original dataset. Adaptive compression dynamically combines these methods, tailoring the compression approach to the data type and its importance, balancing fidelity, storage, and transmission needs. This adaptability is especially advantageous in healthcare monitoring, where real-time shifts in data priorities demand flexible and efficient data management strategies.

Upon investigating how the literature addresses these topics, we perceived that research efforts focus on developing methods that improve accuracy and efficiency in the management of health data [5–8]. However, there are still opportunities, especially in adapting transmission and storage strategies to the specific characteristics of vital signs and patient needs. Developing an adaptive model that considers the health data of each individual and his/her healthcare condition is crucial to maximizing the effectiveness of IoT-based health remote monitoring systems.

To address these challenges, we envisage as important an adaptive data management model that prioritizes valuable vital sign information while minimizing data volume and bandwidth usage. This model must reduce data size through effective compression and adaptively manage different data types according to their priority and relevance to individual health conditions. For example, adults, elderly, healthy and unhealthy people can be monitored in different manners. In addition, each vital sign has its peculiarity, in which oxygen saturation can vary faster than body temperature. The literature lacks solutions that integrate lossy and lossless compression within a layered Edge-Fog-Cloud architecture to efficiently handle health data transmission and storage across geographically dispersed smart city environments. This work proposes the VSAC model (Vital Sign Adaptive Compressor) to fill this gap, looking at bandwidth consumption, data prioritization, and communication latency. Our focus is on optimizing the transmission and storage of vital sign data in a scalable, hierarchical, and adaptive manner. In summary, we highlight the VSAC contribution as follows:

- Development of a data management approach that combines lossy and lossless data compression techniques to optimize data transmission and ensure timely handling of healthcare data, so working with only valuable vital signs data can trigger pertinent alerts for each person in a smart city setting.

The aforesaid context drives us to draw the following research question: How can we efficiently combine lossy and lossless data compression algorithms to handle vital sign-based healthcare data? Here, the term efficiently refers to dealing with streaming data, so we need a protocol that analyzes, compresses, and aggregates data at the right time. Moreover, we are closing our scope to vital sign data, since healthcare is an area where time to get insights and notifications is critical.

The remainder of this article is organized as follows. First, Section 2 discusses the related work of the most recent studies focused on vital signs data acquisition and compression techniques. Next, Section 3 describes the VSAC model. Section 4 presents the model evaluation methodology, describes the prototype developed for the tests, and outlines the planned test scenarios along with the parameters, workload, and metrics. Section 5 presents the results, while Section 6 discusses the main limitation of the proposed work. Finally, Section 7 highlights our final considerations, emphasizing the actual application of the present research.

2. Related work

The literature review used in this study aims to provide an overview of data compression strategies for vital signs in health monitoring systems. The scope of bibliographic research includes the selection of relevant conference papers and journal articles. These sources are indexed in the following scientific databases: ACM Digital Library,[1] Google Scholar,[2] IEEE eXplore,[3] ScienceDirect[4] and Scopus.[5]

The research was carried out on these scientific databases, searching for articles published in 2018 onward, with the objective of identifying works containing specific keywords and terms that should appear in the title or abstract of the selected articles. For this process, the keywords and scope of the research were defined. These keywords were established to obtain search results relevant to the topic of this study. Thus, the following search string, presented in the table below, was established to search and select articles in electronic academic libraries.

```
(IoT OR "Internet Of Things" OR sensors OR
                    actuators)
                    AND
          (latency OR bandwidth OR network)
                    AND
          ("data compress" OR compression)
                    AND
(healthcare OR medical OR "vital-signs")
```

This research resulted in 437 references, with three filters applied to this base for the final selection of articles. The exclusion criteria applied to this database used three filters, the first of which consisted of removing duplicate studies, followed by a second filter consisting of analyzing the title and abstract, removing those that were not related to the acquisition and transmission of vital signs. The third filter analyzed the entire text, selecting only studies that presented solutions focused on efficient transmission of vital sign data using data compression methods. This selection process is presented in Fig. 1.

Although numerous studies are being conducted in the field of data transmission in healthcare, particularly on the Internet of Health Things (IoHT), the literature more broadly addresses various architectures and middlewares. These approaches propose models and solutions to address the challenges of low latency, bandwidth occupancy, and high throughput [20–23]. In this context, the architecture most widely adopted in health monitoring systems is the three-layered Edge-Fog-Cloud. From this layered perspective, data are processed in the first two layers (Edge-Fog) before being sent to the cloud, thus reducing the packet load on the network.

However, data traveling in fog benefit from a fixed and dedicated telecommunications infrastructure, which optimizes transmission speed and latency. However, there is a significant challenge in the volume of data traveling between the edge and fog layers, where communication between these two layers is generally slow [24–26] and often wireless. Therefore, bottlenecks may occur in packet transmission from the edge to the fog. At this point, a solution based on data packet management and optimization is necessary to maintain the scale of sensors distributed in a smart city.

Beyond an architectural approach, as verified in the analysis of related works, another method adopted to handle the high volume of data in vital sign monitoring involves the use of data compression techniques. Although less prevalent, this literature presents different methods that rely on compression techniques, combining widely adopted solutions or proposing modifications to well-known algorithms

[1] https://dl.acm.org/.
[2] https://scholar.google.com.br/.
[3] https://ieeexplore.ieee.org/.
[4] https://www.sciencedirect.com/.
[5] https://scopus.com/.

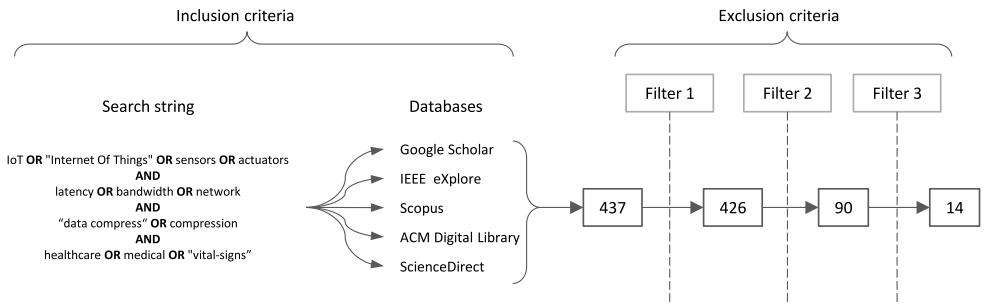


Fig. 1. Filtering process for selecting related papers.

to achieve better performance in specific situations.

As shown in Table 1, a list of related works was analyzed according to the problem that needs to be solved, the approach adopted and the technique developed to obtain the result. The works presented in [5,10,12,15,27] utilize compression techniques that, when combined, generate new methods to achieve more efficient data reduction than would be obtained using just one method. In these cases, lossy and lossless techniques are combined, with the first stage typically applying lossy. This maintains controlled compression rates to avoid losing information that could compromise the quality of the result. In these cases, since we are dealing with vital signs, this result is directly related to the accuracy of a diagnosis or health alert.

For these research efforts that combine compression techniques in two or more stages, a second reduction cycle is applied after the lossy stage to gain further efficiency, now without losses. This process is structured in this way because the first stage has already addressed compression losses at acceptable levels, according to the needs of each system where it is applied.

In the stage responsible for lossy compression, the models that combine lossy and lossless compression techniques frequently employ processes that keep this module up to date on the acceptable loss level. In [6,9,15,17], the compression rate of the module where data reduction with loss occurs is established in a way that stays within levels that do not compromise the quality of restored information. In these proposals, there are instances where information is updated in real-time by setting parameters prior to running the process. The researched literature shows a balance on the type of compression used for vital signs. In [6,9,17], lossy compression is used, with tolerance controls for the level of losses. Lossless compression is found in [13,14,16,18], and a hybrid solution in [5,10,12,15,27].

In [8], a hybrid lossy and lossless ECG compression system is designed for signal fidelity and efficient compression rate. The hybrid compressor has a lossy part based on an optimization method and a lossless part based on the Huffman entropy encoding method. The work presented in [7] proposes a method that takes advantage of the redundancy of vital signs data to provide energy conservation and compression rate. The received signal is predicted continuously, and the error resulting from the actual and expected signals is calculated. The resulting error is compressed by the Run Length Encode (RLE) algorithm and sent to the destination. The method combines Linear Predictive Coding (LPC) for data prediction and RLE for data compression.

Based on the analysis of related works, it was found that, despite various studies in the domain of data compression for vital signs, there are needs to approach that adequately address the primary needs of a health monitoring system. More specifically, there is a research gap in scalable solutions that allow to adaptive transmission strategies in runtime. Especially considering intrinsic characteristics of vital signs, such as the type of vital sign, periodicity, prioritization according to patient needs, and location.

3. VSAC model

The Vital Signs Adaptive Compression (VSAC) model aims to manage vital signs data within a health monitoring system to generate efficiency in network resource consumption and storage space while maintaining the quality of the information available for diagnostic decision-making and generating health alerts. For this purpose, an architecture was designed with components that enable data processing, applying data compression and prioritization techniques. The following subsections present the design decisions, the proposed architecture, the model's functioning, and how the model performs data aggregation and validation.

3.1. Design principles

VSAC utilizes data compression techniques to optimize file transmission over the network. Optimization is based on the intrinsic properties of the vital signs and priority signals monitored and received from the monitoring system. VSAC does not focus on hardware (i.e., wearable devices) that capture vital sign data; however, we understand that JSON-based vital sign data are captured from devices in a streaming-like fashion. For example, we can consider the capture hardware that Facco et al. [28] present. In addition, VSAC is structured into separate components to enable this process, resulting in a modular framework. This allows these components to be implemented in microservices and enables their integration and availability via API. VSAC considers the following vital signs: body temperature (BT), blood pressure (BP), heart rate (HR), and respiratory rate (RR). They represent the most significant step in addressing the sequels of the COVID-19 pandemic [29,30].

The model adopts a two-stage data compression strategy. Given the repetitive nature of the data with slight variation, the first compression stage uses a lossy algorithm. The second stage collects data from the previous step and applies a lossless data compression algorithm to maintain the quality of the information before sending it to the monitoring system in higher layers. This work does not address the data decompression process. The first compression stage uses the Swinging Door Trending (SDT) lossy method [31,32]. This compression algorithm is characterized by allowing compression rate adjustments during runtime, has low computational complexity, and uses a linear trend to represent a certain amount of data [32].

The data received and grouped into a new structure adopted the JSON format, which was chosen because it allows data to be structured in text format for use in different types of systems. It is a simple and lightweight format that offers more processing agility. Moreover, since we have the data in text format, various algorithms are available to provide high compression rates with low CPU processing consumption. The second compression stage receives files in JSON format and applies a lossless algorithm. Among these algorithms, Huffman Coding (HC) and LZW stand out, being easy to implement and requiring little processing footprint [33]. A decision stage analyzes which of these algorithms to apply based on the size of the received file [33], using HC for files smaller than 33.2 kB and LZW otherwise.

Table 1
Related work (RW) analyzed according to the problem that needs to be solved, the approach adopted, and the technique developed to obtain the result.

Identifier	Authors	Year	Problem addressed	Approach	Technique	Results
RW1	Xiao et al.	2018	High energy consumption of IoT devices for transmitting vital signs data.	Reduce the size of data packets through compression techniques.	Alter the Compressive Sensing method to be adaptive and reduce data packet size.	Improve in accuracy recognition (90.5%) with energy consumption reduced by 28%.
RW2	Azar et al.	2019	High energy consumption in wearable devices for vital signs data.	Apply two stages of data processing, compressing before transmission and then restoring at the edge layer using ML techniques, to transmit to the cloud.	Lossy compression technique at the sensor layer and at the edge layer applies deep learning models to reconstruct the data and send it to the cloud.	Reduce the amount of data by up to 103 times and the classification accuracy obtained from training the model on features extracted from compressed data did not decrease.
RW3	Hanoune and Lysmos	2020	High storage space usage and data transfer time for health monitoring applications.	Compress the data at the gateway prior to transmission to the cloud.	Apply compression by prediction, sampling, and scheduling. Then apply another stage of lossless compression and send to cloud.	For mechanism predictive Bzip2 is best suited with CR of 8.43%, LZO and LZ4 for sampling CR between 13.6% and 15.27%. For scheduling, LZO with CR of 16.7%.
RW4	Kalaivani et al.	2020	Energy consumption of IoT devices when transmitting data and the storage space required.	Compress the acquired data before transmission at the edge layer combining compression techniques.	Apply lossless compression and another hybrid one, which includes lossy and lossless compression.	The analysis of storage space using the proposed algorithm shows a reduction of 70% storage space.
RW5	Passos et al.	2020	Energy consumption of IoT devices.	Manage data compression in real-time according to the type of collected vital signs, avoiding redundant data transmission.	Apply lossy compression according to a threshold to limit data loss within acceptable limits, then apply lossless compression.	The proposed method achieved an energy efficiency up to 53.73% with a maximum delay of 55 ms.
RW6	Lu et al.	2020	High bandwidth consumption for transmitting large volumes of data and storage space usage.	Adaptive compression mechanism that decides to either reduce data transmission time or storage space according to the type of data.	Mathematical model that considers CPU, I/O for data reading, network bandwidth, and selects the most appropriate compression mechanism.	The method outperforms the best unitary compressor lzma by 5% in CR with 17% less compression time.
RW7	Das and Rahman	2021	Storage space required for vital signs data and transmission time, especially on low-speed networks.	Create a dictionary based on the structure of vital signs data to encode them before compression.	Categorizing data based on a standard table for each type of vital sign, then compress with a standard lossless method and transmitting.	The technique produces a performance of about 18% to 55% over LZO, LZ4, BZ2, LZSS, LZMA.
RW8	Idrees et al.	2022	Latency and bandwidth occupation in sending data over an IoMT network.	Reduce the amount of data at the edge, identify the possibility of an epileptic seizure in the patient at the fog layer, and apply a new compression stage before sending to the cloud.	Combine K-means clustering and Huffman coding, then send to the fog for disease detection based on ML using Naive Bayes algorithm. Then the method from the first stage is applied again for transmission.	The approach reduces the EEG data size after the compression from 85.6% up to 89.2% compared with noncompressed EEG.
RW9	Khlief and Idrees	2022	Low efficiency in sending data in real-time with limited bandwidth, latency, and high energy consumption.	Compress data packets originating from sensors at a gateway and transmit with a distortion rate within configurable parameters.	Combine data grouping with DBSCAN clustering followed by lossless compression. In the final stage, the data is compressed with Huffman and transmitted.	EEG data compress from 66% up to 98.6%, whereas LZW, HE, HCLZW, and HCHE compress from 40% up to 54%, from 65% up to 70%, from 14% up to 56%, and from 70% up to 83%.
RW10	Hossein and Sheikh	2022	Transmitting large amounts of data for medical monitoring through limited bandwidth channels.	Employ two cascading compression methods, aiming to increase the CR and reduce the volume of data by altering the information bits.	Start compressing the data using the LZW at the application layer, then apply a rearrangement process, and finally compress again using Huffman.	The proposed algorithm reduce data by 37.85% for signal, image and text data.
RW11	Amiri et al.	2022	Clinical-level monitoring of patients that require computing, storage beyond those available in wearable devices.	Adaptive compression method for vital signs data which a collaborative process of interaction between the sensor and the server occurs through a loop.	Through a loop between sensor and server obtain a semantic representation of data. Server evaluates the current set of received coefficients and determines whether to request more or stop according to desired accuracy.	Increases by up to 37% and 26% the recall and F1 score compared to an optimized but static compression strategy.
RW12	Idrees and Khlief	2023	Delay in generating medical diagnoses due to large amounts of data sent to the cloud via the IoMT network.	Use an edge-fog-cloud architecture to compress vital signs at a gateway located in the fog layer, aggregating and compressing before sending to the cloud.	The data received at the gateway is grouped by DBSCAN, then Delta encoding is applied to provide differences for data indices, and then the resulting file is compressed by Huffman Encoding.	Data is compressed in the range 65–85% whereas other methods 40%–54% from 65 to 70% , and from 14 to 56%, for the LZW, HE, and HCLZW, respectively.

(*continued on next page*)

Table 1 (*continued*).

RW13	Chang and Sobelman	2023	Low quality of reconstructed vital sign after being compressed.	Adopt a hybrid lossy and lossless compression system to get good signal fidelity.	Lossy stage followed by a Huffman stage Use a method for evaluating the errors by the lossy compression based on the classification results from DNNs.	Hybrid compressor achieve CR of 4.91% with a mean squared error of 0.01.
RW14	Vakil and Shirmohammadi	2024	Sensors in the WBAN have a limited source of energy consumed largely in data exchange.	Use data prediction for vital signs data and apply a second stage of data compression.	Develop a method called Linear Predictive Run Length Coding that combines linear predictive coding for prediction and runtime coding for compression.	Achieved 98% energy savings and up to a 70-fold reduction in data volume compared to other algorithms.

CR - Compression Rate, LZW - Lempel–Ziv–Welch, LZMA - Lempel–Ziv–Markov, HCLZW - Hierarchical Clustering and the LZW, HCHE - Hierarchical Clustering and Huffman Encoding [19].

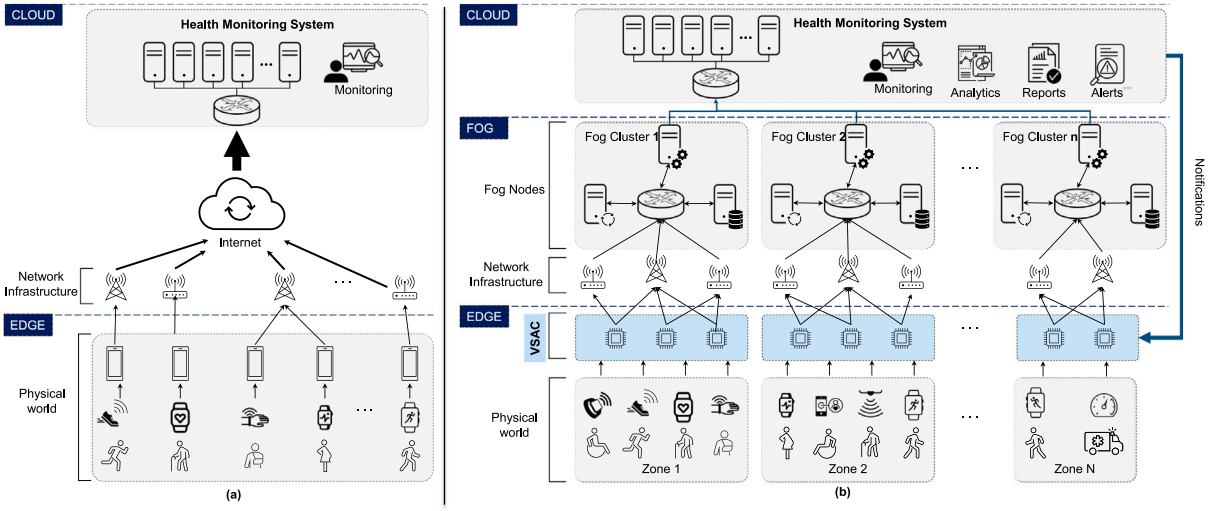


Fig. 2. Traditional (a) and proposed high-level architecture (b) to acquire and process vital sign data. In (a), data originating from wearable devices is transmitted to a mobile device, which in turn sends the files directly through the internet for further analysis. In (b), we have the same trigger for the Wearable devices, but now we envisage a data flow that goes towards VSAC components geographically distributed across different locations. Also, data are compressed and redirected to fog nodes, where preliminary processing occurs before transmitting them to the cloud layer. At the cloud, position-aware data is available to generate reports, dashboards, and alerts, which feed back to the VSAC model through notifications to calibrate the situation of each person and tune VSAC parameters.

The system prioritizes essential aspects to ensure that the VSAC model meets the design goals of energy efficiency, scalability, and time efficiency. Energy efficiency can be achieved through power-aware hardware, efficient compression algorithms, and energy-aware scheduling. In this regard, the module responsible for real-time processing of vital signs data applies lossy compression techniques to optimize energy consumption. This module can prioritize data processing tasks based on their urgency and energy impact. For example, critical health data can be processed with higher priority, while less critical data can be processed at lower priority or deferred to lower energy demand monitoring periods.

We also work with time efficiency in mind since VSAC incorporates strategies such as real-time processing, prioritizing critical health data, and ensuring a prompt response to urgent situations. Parallel processing techniques are used to distribute computational tasks across multiple cores (using Pthreads or OpenMP thread-processing library), accelerating processing times. Efficient algorithms for data compression and signal processing minimize computational overhead. In addition, we work on modeling a network protocol that reduces the network footprint. By implementing these strategies, the VSAC model ensures timely data processing and minimizes delays in critical health monitoring applications by implementing these strategies.

A scalable design capable of handling increasing numbers of users and data volumes can be achieved formerly through a modular architecture that allows independent scaling of components. By distributing processing tasks across edge devices and fog nodes, the VSAC model can efficiently handle increased workloads. We adopt a hierarchical architecture composed of edge and fog nodes and a centralized cloud on top. As DNS does, a hierarchical setting explores the best of client–server and P2P architectures: simplicity and manageability, while providing scalability and fault tolerance from P2P. Furthermore, efficient data compression techniques reduce data volume and optimize resource usage.

VSAC focuses on high-performance computing and healthcare. We consider security to be highly important, but we have not explored in depth in the present manuscript. However, in the current version of the VSAC model, we have already planned initial privacy controls. To accomplish this, the VSAC model incorporates data anonymization techniques. By removing or masking personally identifiable information, the system protects sensitive data while still allowing the main monitoring needs to be achieved. This approach aligns with the principles outlined in the General Data Protection Regulation (GDPR). Specifically, we refer to Article 6(1), Article 25, and Article 5(1)(c).[6]

3.2. Architecture

The architecture adopted for the project encompasses three layers: edge, fog, and cloud. This approach differs from traditional remote health monitoring, which often involves wearable devices transmitting data directly to a mobile device and then to the cloud, as shown in Fig. 2. For the proposed model, vital signs data acquisition occurs at the edge layer, which includes the modules for data management and

[6] General Data Protection Regulation (GDPR), available at: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

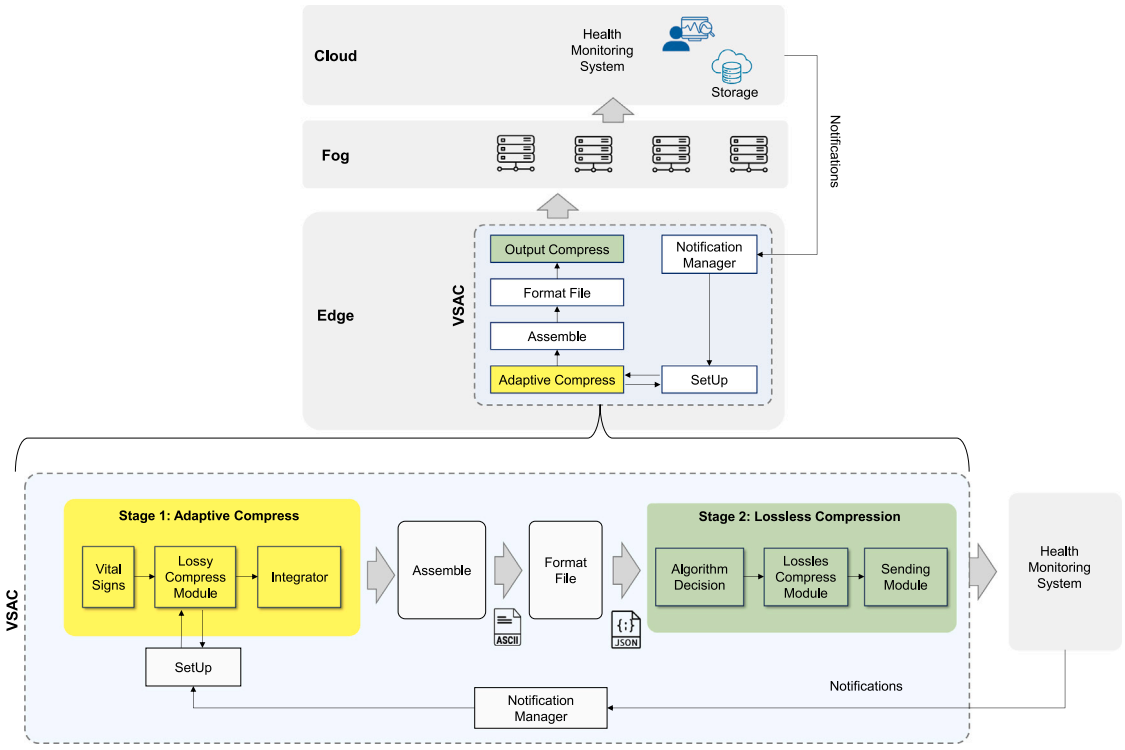


Fig. 3. VSAC architecture and functioning. Here, Adaptive Compress is in charge of performing data reduction according to SetUp parameters. This last component is adjusted based on notifications from the monitoring system via Notification Manager. After the data is compacted in Adaptive Compress, it is sent to the Assemble module, which groups them into packets and make them available in ASCII-based text for Format File. The data, now structured as JSON files, move towards the final compression stage in Output Compress. Thus, data is ready to be transmitted to the health system, where we have data decompression, storage and healthcare diagnostics.

compression, and compressed packets are sent to the fog layer, with information storage taking place in the cloud layer. In this regard, the focus of this work is on the edge layer, where the model is developed.

VSAC introduces an alternative approach to remote health monitoring compared to traditional methods. In conventional systems, data are typically transmitted directly from wearable devices to the cloud. In contrast, the VSAC model incorporates initial data processing and compression at the edge layer. This approach helps to reduce data volume, alleviate network congestion, and optimize bandwidth utilization. Using a layered architecture, the VSAC model improves efficiency and scalability, offering a well-rounded and adaptable solution for remote health monitoring.

Fig. 3 illustrates the architecture and functioning of VSAC, highlighting its key components that define the model as hybrid and adaptive. The model is considered hybrid due to its operation with distinct compression methods across two stages, and it is deemed adaptive because it allows adjustments according to notifications received from the monitoring system. In summary, below, we can see a description of each VSAC module:

- Adaptive Compress: this component is the entry point for vital signs data and the first stage of volume reduction of the data. It receives vital signs from the sensors and signals from the SetUp module. These signals define the thresholds and the data selection cadence. The compression developed in this component refers to the lossy type, meaning that repetitive values of vital signs or those with slight variation, as defined by limits set from SetUp, are going to be discarded. This factor makes the module adaptive, as it adjusts the levels of data selection and discarding according to notifications originated from the monitoring system;
- Notification Manager: this component receives notifications from the Health Monitoring System and forwards them to the SetUp module. From this module, it is possible to individualize the type of vital sign to be prioritized and the respective reading cadence for a specific individual;
- SetUp: this component translates the notifications received from the Notification Manager into commands that adjust the thresholds used in the Adaptive Compress component;
- Assemble: this module is responsible for aggregating the compressed data received from the Adaptive Compress module. It generates an output file with the reduced data to be formatted according to the standard defined by the project. This component also inserts a header in the file with prioritization information, so serving for compression decisions in later stages of the model;
- Format file: The function of this component is to format the data received from Assemble into the JSON standard, corresponding to the format of the output file of this module.
- Output compress: this is the last stage of the model in which the lossless compression method is applied. At this stage, the data are compressed without loss of information, as the first stage of compression in the model has already performed size reductions based on data discarding. The compression method applied in this stage refers to LZW or Huffman Coding. This choice depends on the file size to be compressed and identified through a header. Smaller files are directed to the Huffman algorithm and consist of data that are treated with priority and not undergoing the data grouping process in Assemble, which requires processing time. Other non-priority files are directed to LZW, which performs better, [33]. The compressed files from this module are transmitted to the cloud for storage and use by the monitoring system.

The adoption of Huffman and LZW compression methods in this study is motivated by their widespread use and proven effectiveness in various data compression applications [16,34,35]. These methods are well-suited for compressing the formatted data generated by the VSAC model, leading to significant reductions in file size and improved transmission efficiency. A comprehensive overview of the computational properties of each VSAC module is provided in Table 2. The table presents the input and output data for each module, along with

Table 2
VSAC modules and their respective properties.

Module	Input	Execution type	Data processing type	Latency		Output
				Level	Requirements	
Adaptive compress	Vital signs and parameters	Real-time	Lossy data compression, signal processing, and adaptive filtering.	Low	Millisecond-level	Compressed vital signs data
SetUp	Priority notifications	Real-time	Parameter tuning and configuration.	Low	Millisecond-level	Parameters for compression configuration
Notification manager	Health system	Real-time	Event-driven processing and data filtering	Medium	Second-level	Notifications prioritization
Assemble	Compressed vital signs data	Batch	Data aggregation, formatting, and metadata insertion	Medium	Second-level	Grouped data packets
Format file	Grouped data packets	Batch	Data serialization and formatting.	Medium	Second-level	Files in JSON format
Output compress	Vital signs files in JSON format	Batch	Lossless data compression (LZW or Huffman Coding)	Medium	Second-level	Compressed file to send to health system

Table 3
VSAC protocol: fields that comprise the data string with vital signs information from a particular monitored person.

Field name			Format	Length (bytes)	Description
Packet header	Base ID		nnnnnnnn	8	Identification of the base acquiring station
	Location		+/−dd.nnnnnn; +/−dd.nnnnnn	21	Base station geographic location
	Len		nnnnnnnn	8	Packet data length
Patient data	ID		nnnnnnnn	8	Identification of monitored person
	HR	Len	nnnnnnnn	8	Heart rate packet data length
		Ts	mmddyyyyhhmmss	14	Start time of the vital signs reading
		Te	mmddyyyyhhmmss	14	End time of the vital signs reading
		P	ssssss	6	Period/Cadence of signal readings (s)
		Data	nnn...nnn	var	Heart rate data
	BT	Len	nnnnnnnn	8	Body temperature packet data length
		Ts	mmddyyyyhhmmss	14	Start time of the vital signs reading
		Te	mmddyyyyhhmmss	14	End time of the vital signs reading
		P	ssssss	6	Period/Cadence of signal readings (s)
		Data	nnn...nnn	var	Body temperature data
	OS	Len	nnnnnnnn	8	Oxygen saturation packet data length
		Ts	mmddyyyyhhmmss	14	Start time of the vital signs reading
		Te	mmddyyyyhhmmss	14	End time of the vital signs reading
		P	ssssss	6	Period/Cadence of signal readings (s)
		Data	nnn...nnn	var	Oxygen saturation data
	RR	Len	nnnnnnnn	8	Respiratory rate packet data length
		Ts	mmddyyyyhhmmss	14	Start time of the vital signs reading
		Te	mmddyyyyhhmmss	14	End time of the vital signs reading
		P	ssssss	6	Period/Cadence of signal readings (s)
		Data	nnn...nnn	var	Respiratory rate data
CRC	Len		nnn.nnn	16	Checksum calculated for each data packet

its execution type (real-time or batch), data processing techniques (e.g., signal processing, compression), latency level (low, medium, or high), and specific latency requirements (milliseconds or seconds).

3.3. Functioning

VSAC is designed for data management and can efficiently transmit vital signs data originating from sensors located in the edge layer. Its operation is based on two compression stages, forming a hybrid and adaptive model through a modular structure, as presented in the previous section. Packets originating from distributed readers, installed in acquiring stations within the smart city, follow a structure that meets the health monitoring system's needs. These packets are derived from a data collection base containing strings of vital signs for a monitored group of people. The formatting of these packets is presented in Table 3, and is represented in Fig. 4.

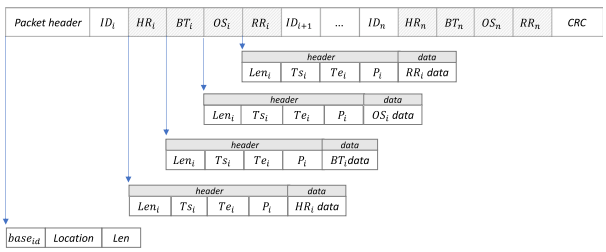


Fig. 4. Packets protocol coming from acquiring base stations.

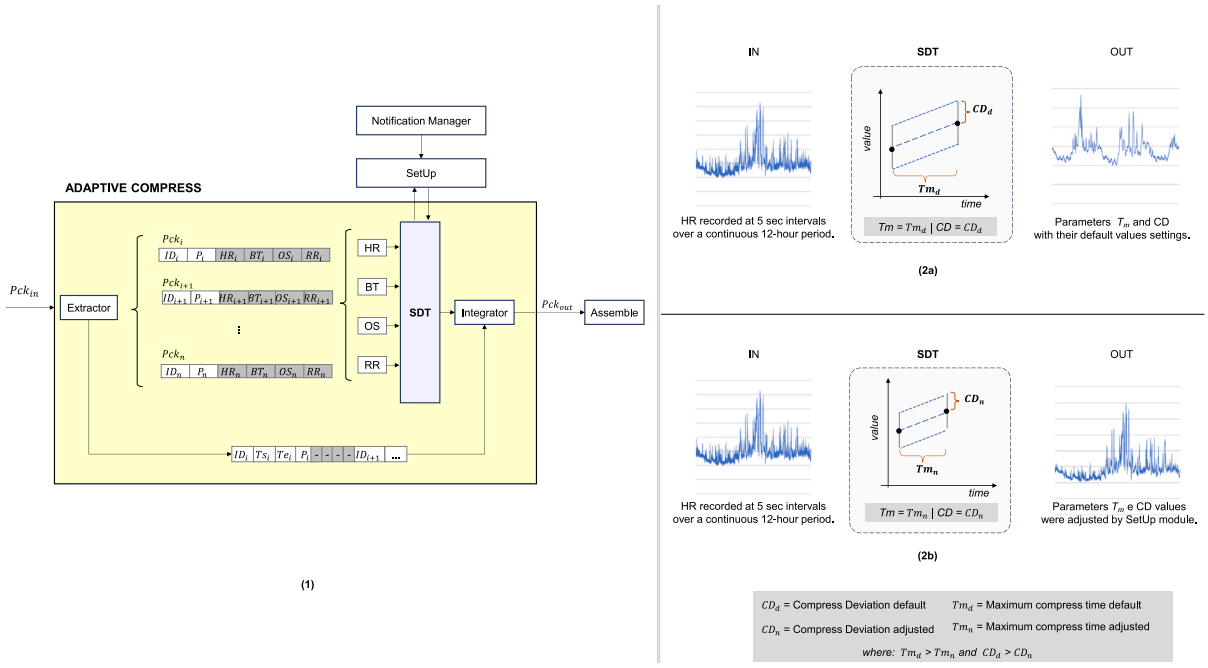


Fig. 5. In (1), the Adaptive Compress module receives packets (Pck_{in}) from the sensors denoted by the Extractor module. Here, we filter from each ID only the vital signs data and send it to be compressed in SDT according to the SetUp parameterization. The compressed data resulting from this step is then reassembled in the Integrator into the original packet, which was separated in the Extractor. In (2), the SDT compression process is emphasized, illustrating how two different outputs can be generated from the same input signal based on the CD and Tm parameter values adjusted during runtime. In (2a), the parameters are at default value; in (2b), the values have been adjusted according to notification coming from the SetUp module. In the example above, for scenario (2b), the notification received in SDT causes the parameter values to be adjusted for a new compression window with an area smaller than in (2a), leading to a lower compression rate.

The data packets containing the strings of each person are received in the Adaptive Compress module, as shown in Fig. 5. Data entry occurs in the component called Extractor, which extracts only the vital signs from each ID, retaining the location information and collection times in the original string. The respective identification (ID_i) and reading period of the signs (P_i) of the person being monitored are also forwarded along with the vital signs, thus forming the packet Pck_i. This packet is then transmitted to the SDT compression module, which applies the adaptive algorithm to vital signs based on the parameterization received from SetUp. This module has two-way communication with SDT, receiving the ID_i of each packet and returning the adjusted compression parameters to apply to the algorithm. Next, in Integrator, the compressed data from each type of vital sign will be aggregated with the respective source ID in such a way that HR_i, BT_i, SO_i, and FR_i are replaced by the new data packets HR'_i, BT'_i, SO'_i and FR'_i.

The operation of the SDT module is exemplified in Fig. 5(2), highlighting the parameters of maximum compression time (Tm) and compression deviation (CD). The higher the Tm and CD, the higher the compression rate. These parameters start from a default value and are adjusted through the SetUp component based on the notifications received and the desired frequency of signal monitoring.

Fig. 6 presents the algorithm for the decision process to adjust the SDT parameters. This process occurs in the SetUp module and begins with the parameters received from the Notification Manager and the SDT. From SDT, ID_i is informed, extracted from Pck_i, and from Notification Manager ID_m is received. These two pieces of information verify whether the packet to be processed needs to be treated with priority. If the result of this comparison is negative, then the default values of the compression module T_m and CD are maintained. If the result of $ID_i = ID_m$ is positive, the next step is to check if it is necessary to prioritize the compression process of Pck_i. The prioritization process involves assigning a high priority to the packet in the process that consumes the compression queue, thus promoting early sending to the Assemble module, thereby accelerating the transmission process to the Cloud. After prioritizing, the next step is to adjust the compression parameters.

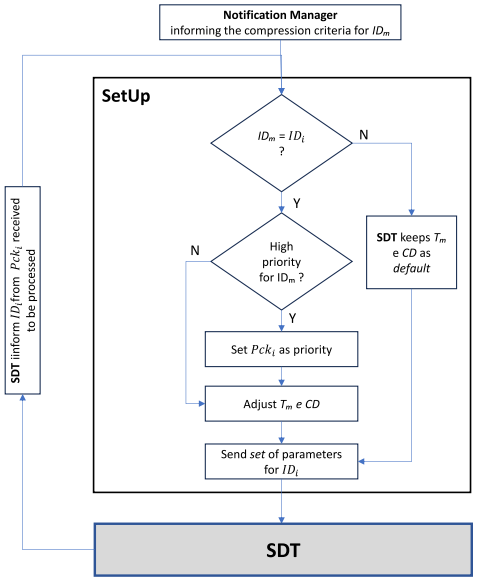


Fig. 6. Flowchart for the decision process for adjusting the SDT module parameters.

To accomplish this, the SDT Tm and CD parameters are modified according to what was received from the Notification Manager. This new set of parameters, relative to Pck_i, is sent to the compression module afterwards.

The packets resulting from Adaptive Compress are received in Assemble and can then follow two paths according to their priority level. If packets are identified as high priority, they need to be sent to the health monitoring system as quickly as possible. Thus, we do not have the file grouping process in the Assemble module. Without a high priority level, other files follow the planned grouped process. The

Table 4
Performance comparison of Huffman Coding and LZW Coding methods for different text file sizes, where CR = Compression Rate, CT = Compression Time (s), and DT = Decompression Time (s).

File size	Huffman coding			LZW coding		
	CR (%)	CT	DT	CR (%)	CT	DT
2.2 kB	54.20	0.0	0.0156	105.56	0.0	0.0
6.7 kB	54.20	0.0	0.069	84.46	0.0	0.0
33.2 kB	**54.205**	**0.015**	**0.231**	49.94	0.0156	0.0156
198.8 kB	54.204	0.0625	1.402	24.219	0.052	0.0159
994.1 kB	54.203	0.601	7.076	**12.458**	**0.284**	**0.115**
3.9 MB	54.203	8.574	30.029	6.319	1.622	1.624
7.8 MB	54.203	34.266	55.879	5.062	3.224	6.852
15.5 MB	54.203	123.664	115.858	3.604	8.560	29.734
31.1 MB	54.203	548.473	237.528	2.566	17.352	103.317
62.1 MB	54.203	1.945.389	470.933	1.827	53.198	469.634

CR (Compression Rate) = (C2/C1) *100%, where C1 = input file size before compression and C2 = output file size after compression.

Assemble component groups the received packets until the file reaches a predefined size. The size of the resulting file corresponds to the value that allows greater compression efficiency for the lossless method adopted in the second and final stage of compression of the model. According to Table 4 [33], the compression rate, compression time and decompression time for the LZW algorithm, when processing text files, the compression rate increases as the file size increases; however, a cutoff line is necessary; that is, the limit size to ensure a continuous process among all components of the model. The output files from the Assemble module have a header containing the information necessary to decide the type of compression in the second stage of the VSAC model.

The output files from the Assemble module are received in Format File, where we convert the input text to JSON format. In this file, the user data are grouped into arrays according to the base station where they were collected, with each user followed by their respective vital signs in strings. In the last module of the model, Output Compress, the second stage of compression occurs, now without losses, since the first stage of compression has already performed the discarding of redundant data. For this stage, files flagged as priority or with a size of up to 33 kB are compressed using the Huffman algorithm. The remaining files are going to be compressed using the LZW algorithm. The output of this module consists of a file ready to be sent to the cloud/fog (upper layers, representing a health monitoring system), where they will be unpacked using the restoration process of the corresponding algorithm.

The Monitoring System, hosted in the cloud, receives compressed data packets and restores vital signs information, enabling the generation of regional control dashboards and the identification of deviant health patterns or anomalous health events. This monitoring can be refined to the level of identification of a potential risk situation for a specific monitored individual, triggering an alert to the patient and their attending physician. In addition, the system can adjust the compression level of the data captured from that individual. By sending Notifications to the Notification Manager, it is possible to tailor the compression level specifically for a particular patient ID and, within the SetUp module, modify the patient's priority and adjust the compression parameters accordingly.

3.4. Data aggregation and validation

Data aggregation is a critical component of the VSAC model, streamlining the management of vital signs data across the edge, fog, and cloud layers. By minimizing redundancy and optimizing network resources, it ensures timely delivery of critical health information while supporting the scalability and real-time responsiveness required in smart city health monitoring systems. This subsection delves into the aggregation strategies within the VSAC framework and their role in enhancing system efficiency. At the edge layer, the Assemble Module plays a important role in executing the data aggregation strategy. Consolidates compressed data packets from the Adaptive Compress module into structured files, optimizing them for subsequent processing and transmission. This process not only simplifies data handling, but also ensures compatibility with the lossless compression stage, forming a seamless link in the model's data management workflow.

The aggregation process involves grouping multiple compressed data packets until a predefined file size is reached. This approach balances efficient data management with optimizing the compression rate in the subsequent stages of the model. The module also supports priority-based processing, allowing high-priority packets to bypass the grouping process for immediate transmission to the health monitoring system. This adaptability ensures responsiveness to varying urgency levels in the monitored health data. Additionally, the module includes file header generation, where metadata, such as prioritization information, is appended to aggregated files. This metadata plays a important role in later stages of the model, particularly in selecting the appropriate lossless compression algorithm (Huffman or LZW) based on the file's size or priority.

Lastly, the module contributes to the optimization of compression efficiency by regulating the size of the aggregated files. Larger files generally yield higher compression rates; however, the module enforces a size limit to maintain processing efficiency across the system. This balance ensures compatibility with the compression algorithms while preserving the model's performance and scalability. Although this paper focuses on the edge layer, the fog layer, with its greater computational capacity, offers additional opportunities for implementing advanced data aggregation strategies, particularly within the context of a smart city. One potential approach is multi-source aggregation, where data from multiple edge nodes within a defined geographical or logical area is consolidated. This strategy is particularly valuable in scenarios such as a hospital, where a fog node dedicated to a specific wing could aggregate data from numerous health monitoring devices operating in that space.

To ensure data integrity, each data packet in the VSAC model is assigned a Cyclic Redundancy Check (CRC) value, which is calculated and transmitted along with the data. This mechanism allows the receiving node to independently calculate the CRC for the received packet and compare it with the transmitted CRC. Any mismatch between these values indicates potential data corruption, enabling the system to identify and address transmission errors promptly. For device authentication, every IoT device is assigned a unique identifier (ID). This ID is verified within the health monitoring system, hosted in the cloud, during the processing of health data. This step ensures that only authorized devices contribute to the system, enhancing the overall security and reliability of the monitoring framework.

4. Evaluation methodology

To evaluate the VSAC model, we developed a methodology that includes a prototype development, definition of input workload, definition of evaluation metrics, and modeling of testing scenarios. This combination allows us to understand the behavior of the model under various conditions, offering insights that can guide design refinement and highlight optimization opportunities.

4.1. Prototype

A functional prototype was developed and implemented in Python for the validation of the VSAC model. The prototype contains the main components that influence the model's outcome, namely, the two stages of compression, the first being lossy and the second being lossless. For a better understanding of the prototype, Fig. 7 depicts the VSAC modules and their implementation details. In total, three integrated modules emulate the behavior of the model, with the entry point being the data related to vital signs obtained through datasets of time series with

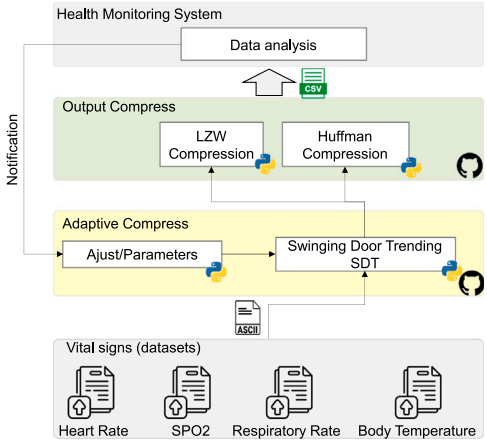


Fig. 7. VSAC prototype.

readings at different types of cadences. The Python language codes for the compression algorithms were obtained from the GitHub platform, with the following links to the respective codes:

- Swinging Door Trending (SDT): https://github.com/chelaxe/SwingingDoor/blob/main/swinging_door.py
- LZW: https://github.com/adityagupta3006/LZW-Compressor-in-Python
- Huffman: https://github.com/bhrigu123/huffman-coding

The prototype runs on a PC with an Intel(R) Core(TM) i7-8550U CPU Processor at 1.99 GHz. In addition, the operating system is Windows 11 Pro (64 bits) and we are running with 8 GB RAM memory and 256 GB SSD memory. To process the input data, a module was developed to emulate Adaptive Compression. The core of this module is the SDT algorithm. In this component, we manually adjust the deviation variable according to the output readings, simulating different scenarios of signals received from the Notification Manager module. The Output Compress compression module consists of two lossless compression algorithms: Huffman Coding and LZW Coding. The final compression stage applies the most suitable compression method to the vital signs file, achieving the best efficiency according to the size. Those smaller files marked as priority are forwarded to the Huffman Coding method. The larger files containing several packets grouped by the Cluster module are redirected to the LZW method, otherwise.

4.2. Input workload

To validate the performance of the model under different load regimes, datasets obtained from the Kaggle[7] database were used. This dataset was extracted from commercial wearable devices to measure physical activity at the population level. For this, a sample of 46 participants (26 women) wore three devices, GENEActiv, Apple Watch Series 2, and Fitbit Charge HR2. The tests consider the heart rate vital sign, as it shows a behavior of data that vary in relation to a central measure throughout the time series. Each dataset used corresponds to a sequence of periodic readings from different persons. We used three different datasets in the trials with different measurement intervals over 24 h, which corresponds to reading intervals of 15 s, 1 min, and 15 min. To test the model for different data densities, three new datasets were created, combining the three previous ones, as shown in Table 5. This strategy allows monitoring of individuals for one day, covering the minimum recommended period of 18 h to detect cardiac problems [36].

[7] https://www.kaggle.com/datasets.

The smallest dataset, with a size of 2.6 kB and a cadence of 15 min, is used to assess the model's performance on small, low-priority files. This dataset helps evaluate the model's ability to handle minimal data volumes efficiently. As the size of the data set increases, the performance of the model is tested with higher data densities and varying numbers of devices. The largest dataset, with a size of 100,000 kB and a high data density, challenges the model's ability to handle large-scale, heterogeneous data streams. To achieve the desired file sizes and characteristics, the datasets were composed of varying numbers of devices with different data cadences. For example, the 1 MB dataset was created by combining data from 34 devices with a 15 s cadence, 34 devices with a 1-min cadence, and 40 devices with a 15-min cadence, totaling 108 devices. This diverse range of datasets allows for a comprehensive evaluation of the scalability, efficiency, and accuracy of the VSAC model under various real-world conditions.

4.3. Evaluation metrics

The evaluation metrics used in this study were selected from related work, focusing on their relevance and suitability to effectively measure and ensure consistency in assessing the performance of the proposed model. First, we measure the Compression Rate (CR) to assess how effectively the model reduces the size of the data in its two compression stages (see Eq. (1)). In the CR equation, Tf is the final size of the file with the compressed data, and Ti is the initial size of the file with the original data. This metric is essential to understand how well the model conserves bandwidth while retaining crucial information for healthcare monitoring. Moreover, we also capture the Compression Time (CT), which is in charge of perceiving the speed to compute both lossy and lossless compression stages. This metric provides insight into the system's capacity to handle high data volumes efficiently, a necessary feature for real-time monitoring applications.

$$CR = \frac{Tf}{Ti} \times 100 \tag{1}$$

Additionally, Percentage Root-Mean-Square Difference (PRD) evaluates the level of distortion introduced by the lossy compression stage. PRD measures the accuracy of compressed data relative to the original, making it a vital metric to ensure the reliability of health information [37,38]. Commonly used metric in biomedical signal processing, PRD reflects the balance between data size reduction and data fidelity [39]; it is crucial to assess the ability of the VSAC model to preserve the quality of vital sign data while optimizing storage and transmission. These metrics provide a comprehensive view of the prototype's performance, particularly in maintaining data integrity, efficiency, and adaptability in real-time scenarios.

To measure the effect of data distortion on the original data after applying lossy compression, the PRD formula involves comparing the original data (input) with the reconstructed data (output) to quantify the differences introduced by lossy compression (see Eq. (2)). Here, N is the total number of data points in each file, X_i represents each data point in the original heart rate data file, and Y_i represents each corresponding data point in the compressed heart rate data file.

$$PRD = \frac{\sqrt{\sum_{i=1}^{N}(X_i - Y_i)^2}}{\sqrt{\sum_{i=1}^{N} X_i^2}} \times 100 \tag{2}$$

4.4. Evaluation scenarios

We modeled three scenarios to evaluate VSAC: (i) scenario 1 evaluates the complete VSAC model, incorporating its primary components; (ii) scenario 2 focuses solely on the lossless compression method; (iii) scenario 3 examines only the lossy compression method. However, the third scenario is inherently embedded within the first module of the complete model (scenario 1), making it unnecessary as a standalone test. Performance analysis emphasizes the model's behavior under diverse data conditions, such as file sizes and data densities,

Table 5
Types of files used for model testing, with measurements from each person over 24 h.

Size (kB)	Cadence	Number of readings	Number of devices	Property	Purpose
2.38	15 min	96	1	Low data density	Check performance with small/priority files.
36	1 min	1440	1	Medium data density and small file size	Analyze performance with homogeneous data in few packets.
255	15 s	5.760	1	High data density in medium-sized files.	Analyze model with few packets and high data density.
1000	1 min	40.320	28	High data density in medium-sized files.	Analyze model with standard package size.
10,000	15 s/1 min/15 min	335.222	108	High, medium and low data density combined in high-sized files	Analyze model with heterogeneous data and large number of packets.
100,000	15 s/1 min/15 min	2.707.584	661	High, medium and low data density combined in very large sized files	Analyze model with heterogeneous data and very high number of packets.

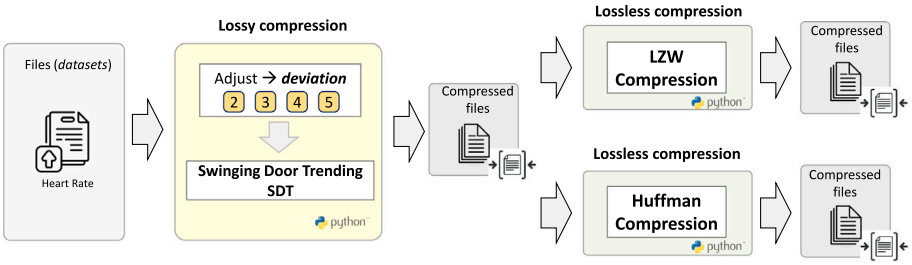


Fig. 8. Structure of scenario 1: two compression methods in sequence, with the second stage being the lossless algorithms, with Huffman Coding and LZW strategies.

to assess its effectiveness across different compression needs. Scenario 1 contains the two-stage compression processes, which integrate the lossy and lossless compression stages (see Fig. 8). Each of the six input datasets will first undergo lossy compression, with each dataset processed in four test cycles to analyze the impact of different values of the parameter *deviation*. These lossy outputs then serve as input for the lossless compression modules, where files flagged as priority or with a size of up to 33 kB are directed to Huffman Coding. In contrast, all other files are processed using the LZW algorithm as designed.

We analyze the ability of the model to achieve a high compression rate and a low compression time. It is also essential to analyze the relationship between the compression rate achieved in the lossy compression stage and its impact on the distortion rate, as measured by the PRD. The health monitoring system defines the PRD level that is deemed acceptable remotely based on medical criteria tailored to the needs of each patient. This adaptability allows for a personalized approach in which compression parameters are adjusted to maintain data quality that meets clinical standards, ensuring that essential information for accurate health monitoring is preserved. Analyzing this relationship helps determine compression settings for each patient, balancing data reduction with the retention of critical health data integrity. By monitoring PRD levels, we can quantify the extent of information retention in compressed output, enabling VSAC to achieve efficient data management without compromising the clinical relevance of vital signs. As mentioned above, this adaptability of PRD thresholds ensures that the VSAC model can respond to varying patient conditions, providing effective compression while supporting real-time patient-specific health insights.

Fig. 9 depicts the functioning of Scenario 2. Here, it is important to compare the trade-off between CT and CR and to compare it to the entire configuration of scenario 1. For this scenario there is no need to measure PRD because there is no signal distortion when compressing with lossless techniques. We designed scenario 2 to evaluate the performance of lossless compression algorithms. This scenario focuses on analyzing the trade-off between CT and CR and comparing these results with the complete configuration of scenario 1. The following tests compare the VSAC model with other data compression configurations to highlight its relative performance and efficiency. The scenarios use data from real patients collected over 24 h, focusing on different conditions, including typical heart rate variations and cases of cardiac arrhythmia.

5. Results and analysis

The following section presents the results of the test of the VSAC model within the previous evaluation scenarios. As PRD indicates, the two-stage compression process that integrates lossy and lossless algorithms is tested across six datasets with varying deviation parameters to measure how different settings impact CR, CT, and data integrity. CR and CT values were collected using different input file sizes to assess the model's performance. For the adaptive module, the parameter *deviation* was incrementally raised from 2 to 5 with each test cycle, selected to balance data reduction while preserving the level of information necessary for medical diagnosis based on the average characteristics of the monitored signals.

5.1. CR and CT evaluation

As shown in Fig. 8, the experimental setup involves a two-stage compression process. The first stage employs a lossy compression technique, while the second stage utilizes Huffman Coding or LZW for lossless compression. Each of the six input datasets will be subjected to four test cycles in the lossy compression module, with the parameter *deviation* varied in each cycle. The output files from this stage are processed by the lossless compression module, with files smaller than 33 kB being compressed using Huffman Coding and larger files using LZW. In Table 6, the results of the measurements from each stage are described, covering both compression stages, lossy and lossless, as well as for each value of the parameter *deviation*. The final values of CR and CT_{total} consider the path taken by the file from its origin to the exit of the lossless compression module.

Analyzing the test results, the CR improves as the deviation parameter increases. This is because larger deviations in the lossy compression stage allow for a more significant reduction in input data. For example, for a 100,000 kB input, the CR decreases from 8.00 (deviation = 2) to 4.73 (deviation = 5), representing a reduction of 40.8%. Smaller

Table 6
Results of test encompassing the two stages of compression in the model. For the first stage, with a lossy algorithm (SDT), different compression rate settings were tested using the *deviation* parameter. In the second stage, lossless algorithms are used, specifically LZW and Huffman Coding, depending on the size of the files. In the end, a table named *Output* presents the total Compression Rate for each configuration and the total time spent in both stages.

DEVIATION SDT = 2	STAGE 1			STAGE 2				OUTPUT	
	LOSSY			LOSSLESS					
Input (kB)	Output (kB)	CR	CT (s)	Algorithm	Output (kB)	CR	CT (s)	CR	CT total (s)
2.38	1.21	50.84	0.002	Huffman	0.781	64.55	0.009	32.82	0.011
36	12.63	35.08	0.015	Huffman	5.73	45.37	0.026	15.92	0.040
255	44.34	17.39	0.023	LZW	20.4	46.01	0.064	8.00	0.087
1000	404.93	36.38	0.336	LZW	121	29.88	0.116	10.87	0.452
10,000	3560	35.60	1.806	LZW	990	27.81	1.09	9.90	2.896
100,000	36445	36.45	18.840	LZW	8001	21.95	14.89	8.00	33.730
DEVIATION SDT = 3									
	LOSSY			LOSSLESS					
Input (kB)	Output (kB)	CR	CT (s)	Algorithm	Output (kB)	CR	CT (s)	CR	CT total (s)
2.38	1.21	50.84	0.003	Huffman	0.703	58.10	0.004	29.54	0.007
36	9.32	25.89	0.005	Huffman	4.28	45.92	0.015	11.89	0.020
255	30.69	12.04	0.021	Huffman	14.2	46.27	0.047	5.57	0.067
1000	303	27.22	0.208	LZW	94.7	31.25	0.089	8.51	0.297
10,000	2.670	26.70	2.280	LZW	762	28.54	0.809	7.62	3.089
100,000	26.650	26.65	16.320	LZW	6.020	22.59	10.915	6.02	27.235
DEVIATION SDT = 4									
	LOSSY			LOSSLESS					
Input (kB)	Output (kB)	CR	CT (s)	Algorithm	Output (kB)	CR	CT (s)	CR	CT total (s)
2.38	0.886	37.23	0.001	Huffman	0.632	71.33	0.003	26.55	0.004
36	7.59	21.08	0.007	Huffman	3.54	46.64	0.015	9.83	0.022
255	21.92	8.60	0.023	Huffman	10.2	46.53	0.031	4.00	0.054
1000	237	21.29	0.299	LZW	76.7	32.36	0.073	6.89	0.372
10,000	2.090	20.90	1.862	LZW	593	28.37	0.632	5.93	2.494
100,000	20.910	20.91	15.988	LZW	4.730	22.62	8.46	4.73	24.448
DEVIATION SDT = 5									
	LOSSY			LOSSLESS					
Input (kB)	Output (kB)	CR	CT (s)	Algorithm	Output (kB)	CR	CT (s)	CR	CT total (s)
2.38	0.774	32.52	0.001	Huffman	0.593	76.61	0.003	24.92	0.004
36	6.06	16.83	0.007	Huffman	2.88	47.52	0.014	8.00	0.021
255	16.26	6.38	0.022	Huffman	7.5	46.13	0.024	2.94	0.046
1000	188	16.89	0.278	LZW	63.2	33.62	0.061	5.68	0.339
10,000	1.660	16.60	1.823	LZW	464	27.95	0.478	4.64	2.301
100,000	16.560	16.56	14.252	LZW	3.750	22.64	6.576	3.75	20.828

datasets follow a similar trend, but differences in CR across deviations are less pronounced. Inputs of 2.38 kB or 36 kB, the limited scope for lossy compression on already compact data minimizes the variation. Additionally, Huffman compression proves less effective than LZW for larger datasets, as LZW achieves better compression efficiency in these cases.

In the lossy compression stage, CT shows minimal variation across different deviation values, indicating that file density or deviation has little impact on processing time. Regardless of the input density or the selected deviation, the algorithm processes all data points in the file. For the lossless compression stage, CT exhibits a linear relationship with file size, as expected, with processing times proportional to the dataset size and independent of data density. The results also highlight a trade-off between CR and CT. Achieving better compression typically requires more computational resources, leading to increased CT. However, larger datasets benefit significantly from the combined effects of the lossy and lossless stages, achieving greater proportional reductions in size compared to smaller datasets. This underscores the efficiency of the VSAC prototype in handling large data compression tasks.

We compare the VSAC results (two compression stages) against those obtained from lossless compression algorithms in a single stage, enabling us to analyze the model's performance with two compression stages. By applying the same vital sign datasets used in the prototype tests to these lossless methods, as shown in Fig. 9, where for each dataset, we measure CR and CT taken from data input to the completion of the compression process. Table 7 presents the results of the tests with the Huffman Coding and LZW compression algorithms. Although these

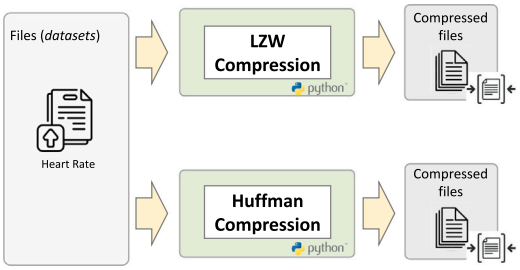


Fig. 9. Structure of scenario 2: we only have lossless algorithms, with Huffman and LZW implementations.

two algorithms are known to perform differently depending on the input file size, both were subjected to the same type of test. We observe that the full implementation of VSAC, in the final model proposed in this work, the file from the Output Compress module is directed to the appropriate compression method according to size and prioritization criteria.

The results show that the Huffman encoding demonstrates lower compression performance across all file sizes except the smallest dataset (2.38kB), indicating lower compression efficiency than LZW. As file sizes increase, the difference in CR becomes more apparent; as verified with a 100,000 kB input, Huffman achieves a CR of 49.55%, while LZW compresses more effectively with a CR of 20.10%. This trend suggests

Table 7
Result of test applying lossless compression algorithms, taking files of different sizes as input. This test put the results of LZW and Huffman Coding side by side.

	Huffman coding			LZW		
Input (kB)	Output (kB)	CR	CT (s)	Output (kB)	CR	CT (s)
2.38	1.39	58.40	0.005	1.58	66.38	0.0149
36	17.5	48.61	0.052	12.5	34.72	0.0214
255	128	50.19	0.428	83.4	32.70	0.082
1000	550	49.41	1.709	293	26.32	0.32
10,000	4.830	48.30	15.5	2560	25.60	3.26
100,000	49.552	49.55	154.64	20.100	20.10	43.3

Table 8
Distortion rates for six input files according to the variation of parameter deviation.

Input (kB)	Deviation			
	2	3	4	5
2.38	38.24	42.77	47.89	49.87
36	50.48	55.36	56.83	56.87
255	44.52	46.00	51.12	52.63
1000	33.09	35.52	35.67	38.66
10,000	32.62	34.18	35.56	36.80
100,000	32.54	34.13	35.36	36.69

that LZW is better suited for achieving higher compression efficiency, particularly with larger datasets. Regarding CT, Huffman Coding processes data significantly faster for smaller files, with a CT of 0.005 s for the 2.38 kB input, compared to LZW's 0.0149 s. However, as input sizes grow, Huffman's CT increases, reaching 154.64 s for the 100,000 kB file, whereas LZW's CT for the same file is 43.3 s. This indicates that Huffman is more computationally expensive for large datasets, making LZW a better choice in scenarios where processing time is critical. Finally, LZW surpasses Huffman in compression efficiency, especially for larger datasets, although it requires slightly more processing time for smaller files. For most scenarios, except when dealing with tiny files, LZW is the preferred choice because of its ability to achieve higher compression rates and more efficient processing for larger datasets.

5.2. PRD evaluation

The distortion rates for compressed heart rate datasets using the swinging door trending method were analyzed for six input file sizes and four *deviation* parameters. These parameters control the compression threshold, with higher values allowing more significant data simplification but introducing increased distortion. To illustrate it, Fig. 10 preset the relationship between the parameter *deviation* and the PRD for a file of size 36 kB. In Fig. 10a, parameter *deviation* is set to 2, and the resulting PRD is 50.48%, and Fig. 10b, parameter *deviation* increases to 5, leading to a higher PRD of 56.83%. The effect of the change in the parameter is visually noticeable. Increasing the parameter value from 2 to 5, compression introduces more distortion as it grows, which is expected since larger values of *deviation* allow for greater simplification of the data representation. Fig. 10b contains fewer plotted points compared to Fig. 10a (deviation = 2).

The complete DR results for the input files are presented in Table 8. The analysis revealed that the effect of *deviation* on distortion rates across all input file sizes increased consistently with higher values of *deviation*. This behavior aligns with expectations, as larger deviations allow more aggressive compression, resulting in greater data loss. The results highlight a fundamental trade-off. Here, lower values *deviation* result in lower PRD and more accurate data representation, but require more points, leading to less compression. In contrast, higher *deviation* values improve compression efficiency by reducing the number of data points but increase the PRD, reducing the accuracy.

Regarding the impact of dataset file size, Fig. 11 demonstrates that larger datasets tend to achieve significantly lower distortion rates than smaller datasets, as indicated by the analysis. The 100 MB dataset

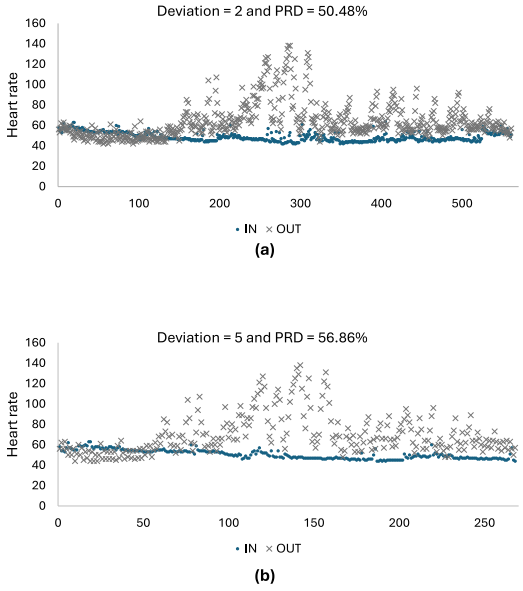


Fig. 10. Relation between distortion rate and *deviation*. For the same input file containing heart rate data, with 36 kB, (a) with a deviation of 2 results in a PRD of 50%, and (b) with a deviation of 5, the PRD results in 56%.

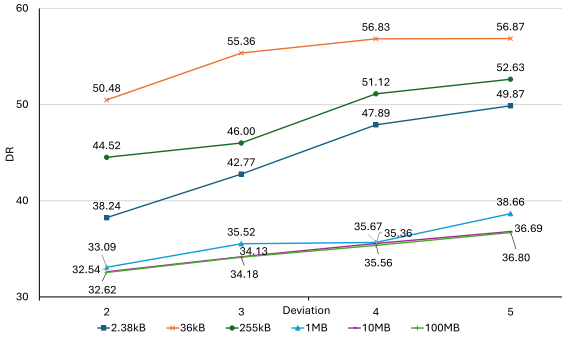


Fig. 11. DR for the six input files with different file sizes, according to parameter *deviation* variation.

achieved a distortion rate of 32.54% at a *deviation* of 2 and 36.69% at a *deviation* of 5, while the 36kB input file showed distortion rates of 50.48% and 56.87%, respectively. This suggests that the SDT method is more effective when applied to larger data sets, mainly when associated with the fact that these files have high data density, i.e. a lower data reading rate.

The sensitivity of smaller files, with low and medium data density, such as 3kB and 36kB, was susceptible to increases in the deviation parameter. These datasets generally exhibited the highest distortion rates, with the 36kB file reaching 56.87% at the largest deviation parameter. This high sensitivity indicates that lower deviation thresholds (e.g., 2 or 3) are more appropriate for such datasets to preserve data

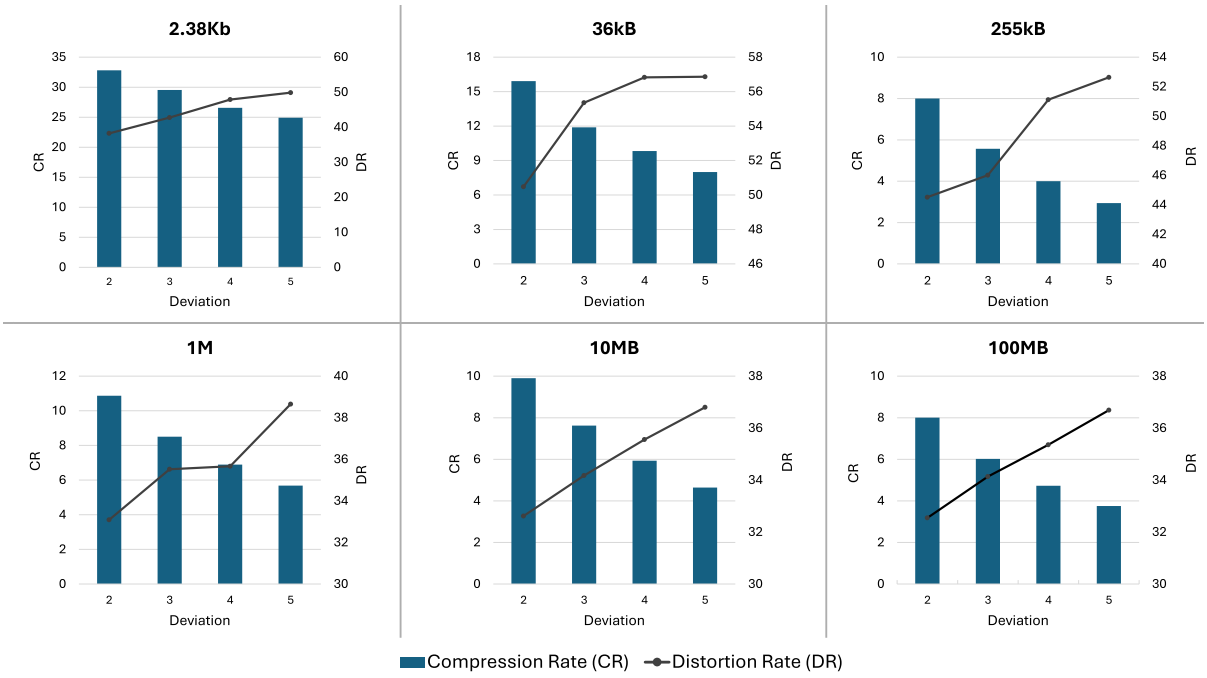


Fig. 12. Compression Rate (CR) performance versus Distortion Rate (DR) for each dataset.

accuracy. The model's performance on large datasets, particularly the 10 MB and 100 MB, demonstrated significantly lower distortion rates and less sensitivity to increasing deviation parameters. The distortion rate for the 10 MB file increased slightly, from 32.62% at a deviation of 2 to 36.80% at a deviation of 5. These findings suggest that larger datasets allow for more efficient compression with lower compromise in data integrity.

The transition point at mid-sized files, such as the 1 MB dataset, represented a threshold where the SDT method began to achieve more efficient compression. Its distortion rates, ranging from 33.09% to 38.66%, were closer to those of the larger datasets (10 MB and 100 MB) than to the smaller ones (3kB and 36kB). The results indicate that the lossy stage performs best on larger and high-density datasets, where higher deviations can be used to achieve greater compression with lower distortion. The method is more prone to distortion for smaller and lower density datasets, which requires careful selection of lower deviation parameters to maintain data fidelity. This behavior highlights the importance of tailoring compression parameters based on the size of the data set, especially in healthcare applications where accurate heart rate monitoring is critical.

Analyzing the relationship between CR and DR for each dataset and its respective deviation, revealing the trade-off between data compression and the resulting distortion, as shown in Fig. 12, we verify that for higher compression, the DR increases. This inverse relationship is consistent across all datasets and deviations. The higher the compression, the more information is lost, resulting in higher distortion. Increasing the parameter *deviation* (from 2 to 5) leads to greater compression and higher DR values. This indicates that deviations allow for more aggressive compression at the cost of data fidelity. It is possible to observe across datasets that CR and DR exhibit smaller variations than smaller datasets for large files (100 MB, 10 MB, 1 MB). CR for the 100 MB dataset ranges from 8.00 to 3.75, while the DR ranges from 32.54% to 36.69%. These data sets show better quality preservation (lower DR) for compression levels than smaller datasets. Medium datasets (255 kB, 36 kB) experience more pronounced increases in DR with compression. The 36kB dataset has a DR that increases from 50.48% to 56.87%, although the CR changes only from 15.92 to 8.00. This indicates that smaller datasets are more sensitive to compression. The smallest data set (2.38kB) shows the most significant CR values, ranging from 32.82 to 24.92, but its DR increases significantly from 38.24% to 49.87%. Smaller datasets tend to be less tolerant of compression, with a higher proportional distortion for a given CR.

Through notifications from the Health Monitor System that can adjust *deviation*, it is possible to define the perceived trade-off if the priority is compression, prioritizing storage efficiency, and scenarios with lower CR (e.g. deviation 5) are preferable. However, the DR increase must be acceptable for the use case. Prioritizing data quality, requiring minimal distortion, scenarios with higher CR (e.g. deviation 2) should be chosen, sacrificing compression efficiency. Regarding dataset sensitivity, larger datasets (e.g. 100 MB, 10 MB) offer better CR-DR trade-offs, making them more suitable for high-compression scenarios. Lower compression levels (higher CR) are advisable to limit distortion for smaller datasets.

5.3. Discussion

The results of the VSAC prototype evaluation demonstrate its efficacy in handling the compression and processing of vital signs data across diverse file sizes and compression configurations. Based on the findings of the prototype tests detailed in the previous section, the performance outcomes for each scenario were summarized and presented in Table 9. This table compares two lossless compression algorithms, Huffman and LZW, along with the hybrid compression model VSAC with a deviation threshold of 5. The evaluation focuses on two key metrics: CR, expressed as a percentage, and CT, measured in seconds, across six datasets of varying sizes.

Compression Rate: The test results demonstrate that for smaller datasets, such as 2.38 kB, Huffman achieves a CR of 58.40%, which decreases only slightly for larger datasets, such as 49.55% for 100,000 kB. This consistent behavior demonstrates Huffman's reliability for compression, but highlights its limited ability to handle data reduction compared to the other methods. In contrast, LZW achieves the lowest compression efficiency overall, with CR values that drop from 66.39% for the 2.38 kB dataset to 20.10% for the 100,000 kB dataset. Although LZW performs well for small files, it struggles with larger datasets, where it provides limited data reduction. In contrast, the model achieves the lowest CR values across all datasets, focusing on higher compression. The 2.38 kB dataset yields a CR of 24.92%, and

Table 9
Comparative table with the test scenarios applied in the prototype. The highlighted values correspond to the best results for each scenario, with those from VSAC returning the best compression rates for all input files.

Input (kB)	LOSSLESS				VSAC	
	Huffman		LZW		SDT / deviation = 5	
	CR	CT (s)	CR	CT (s)	CR	CT (s)
2.38	58.40	0.005	66.39	0.015	**24.92**	0.004
36	48.61	0.052	34.72	0.021	**8.00**	0.021
255	50.20	0.428	32.71	0.082	**2.94**	0.046
1000	49.42	1.709	26.33	0.320	**5.68**	0.339
10,000	48.30	15.5	25.60	3.260	**4.64**	2.301
100,000	49.55	154.64	20.10	43.300	**3.75**	20.828

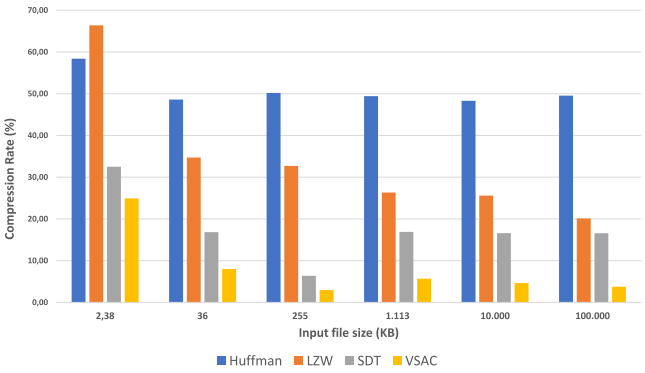


Fig. 13. Graph comparing the Compression Rate results in different test scenarios. The best scenario performance that emulates the two compression stages according to the VSAC model is verified.

for the 100,000 kB dataset, it drops to 3.75%, reflecting the suitability of VSAC for scenarios that require high compression rates.

Compression Time: Analyzing the CT of the algorithms, Huffman consistently requires the most processing time for larger datasets. The CT for the 100,000 kB dataset reaches 154.64 s, compared to 43.3 s for LZW and 20.83 s. LZW performs better than Huffman in terms of speed, but lags behind hybrid VSAC for most input sizes. VSAC demonstrates a significant advantage in terms of compression time, particularly for larger datasets, with a CT of 20.83 s for the 100,000 kB dataset. This efficiency makes VSAC especially well suited for applications where computational speed is critical.

Trade-off: The results highlight the trade-offs between CR and CT among the three algorithms. Although Huffman offers consistent CR values, its compression times are the longest. LZW performs faster than Huffman, but provides the least efficient compression rates overall. Despite achieving the lowest CR values, the VSAC model stands out for its significantly faster processing times and high compression efficiency, notably for large datasets. These characteristics make VSAC a practical choice for applications that prioritize speed and compact data representation, such as trend data in IoT or healthcare monitoring systems.

Fig. 13 highlights the compression rate among scenarios, and through this graph, it is possible to verify the direct relationship of this metric with the file size. Another observation from this graph is the model's performance with tiny file sizes, such as the 2.38 kB file used in the tests. For this file size, note that in all scenarios, the lowest compression performance is obtained, which does not pose an obstacle to the final goal of the model, as smaller files should not significantly impact bandwidth consumption when transmitted in the communication network. The results demonstrate that the model achieves the best compression efficiency and time processing for files of approximately 1 MB. Evaluating the lossless compression scenario, which evaluated the Huffman Coding algorithm across different file sizes, yielded insights. Although initial project assumptions suggested that Huffman Coding would outperform the LZW algorithm for files up to 33 kB, the experimental results revealed the contrary. The LZW algorithm was the most practical option within this size range. Specifically, for 36 kB files, the LZW algorithm produced compressed files that were 29% smaller than those generated using Huffman Coding.

As shown in Fig. 14, the test scenarios for different input file sizes demonstrate that the model presents a better compression time compared to the sum of the separated algorithms. Finally, the results obtained from the prototype trials demonstrate the high potential of the VSCA model. By combining two distinct compression techniques, the model takes advantage of each other's strengths, achieving superior performance compared to their separate applications.

Communication Time and Data Reduction: A significant factor affecting overall system performance is the communication time, which is directly proportional to the amount of data transmitted. Both lossy and lossless compression techniques reduce data volumes, thus decreasing latency and enabling faster data transmission. This reduction exhibits linear behavior, highlighting the proportional relationship between data size and communication efficiency. However, communication time also depends on the robustness of the fog layer in receiving data from multiple edge nodes. Load balance and elasticity in the fog layer can help mitigate potential bottlenecks [40,41], ensuring smooth data flow and low latency, especially in high-traffic scenarios.

Distortion Rate: The PRD analysis reveals an expected trade-off between compression and data fidelity. Higher deviation values increase the PRD, indicating greater signal distortion, while larger datasets exhibit lower PRD due to their higher data density; smaller datasets are more sensitive to compression. However, the smallest dataset (2.38 kB) showed a notable increase in PRD as the deviation increased, which requires parameter selection to preserve diagnostic quality. However, the VSAC model enables real-time adjustment of the distortion rate, allowing the deviation parameter to be fine-tuned during execution to align with the specific needs of the patient and the guidelines set by the medical board. For large datasets, the VSAC prototype proves robust, achieving low PRD even at higher deviations. This makes it ideal for scenarios that prioritize storage efficiency over data precision. In contrast, lower deviations are recommended for smaller data sets to obtain high-fidelity data. As shown in Fig. 12, there is an inverse relationship between CR and PRD across datasets, underscoring the need to balance these metrics based on monitoring requirements. Large datasets demonstrate a favorable trade-off, achieving high CR with minimal PRD increase. In contrast, smaller datasets exhibit more pronounced distortion for similar CR improvements, emphasizing the need for careful configuration when dealing with low-density data.

Initialization and **Setup times**: Key metrics, such as Initialization Time for each module and Overall System Startup Time, evaluate the system's responsiveness from startup to entire operation. At the same time, Parameter Change Time reflects the model's capability to quickly adapt to real-time notifications by adjusting operational parameters. These metrics collectively validate the effectiveness of the VSAC model in meeting its primary objectives of reducing data bandwidth usage and minimizing latency while highlighting its adaptability to the dynamic requirements of healthcare monitoring environments. The initialization and setup analysis further examines the readiness of the VSAC system modules under real-world conditions. Using timestamp logging to capture key performance indicators without introducing overhead, the tests were conducted in realistic environments and repeated 10 times for accuracy. The results reveal variations in startup and configuration times influenced by each module's unique roles and computational demands. For instance, the Adaptive Compression module shows the longest initialization time (1.211 s) due to the complexity of parameter configurations. In comparison, the Notification Manager demonstrates the shortest initialization (0.215 s) and setup times (0.150 s). Modules like Assemble and Format File exhibit low initialization times and

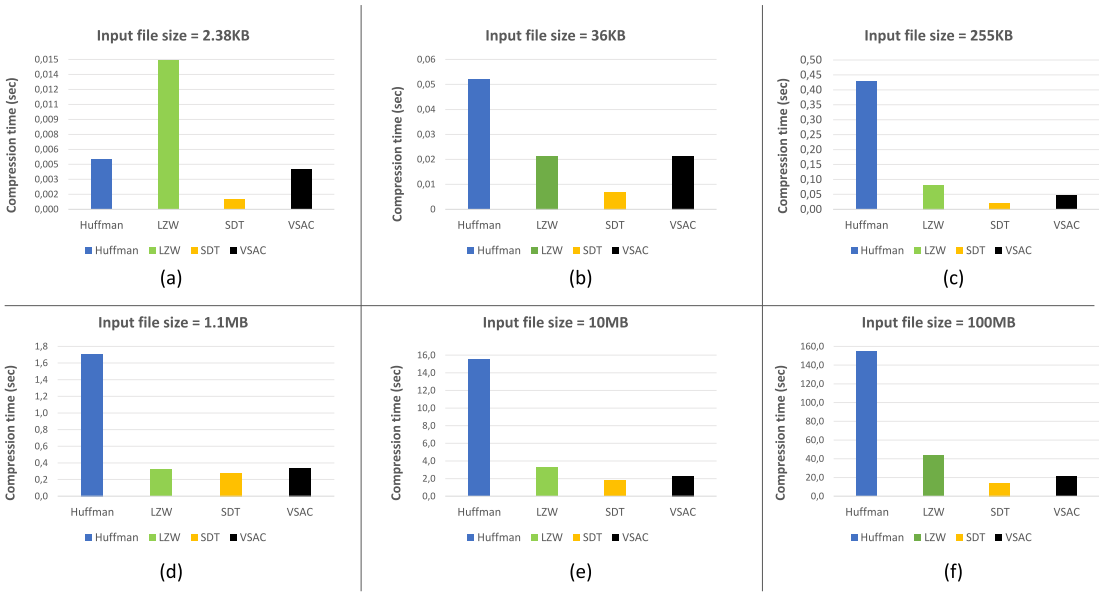


Fig. 14. Graph comparing the Compression Time for each input file size. In test (a), LZW presents the worst performance, as expected for this file size, and VSAC has a time less than the sum of the Huffman time plus SDT. For tests (b), (c), (d), (e) and (f), Huffman has the worst performance, which becomes more pronounced as the size of the input file increases. For all these cases, VSAC has a time less than the sum of the LZW time plus SDT.

require no setup, underscoring their streamlined design. These findings refine the model's processing efficiency, offering critical insights into its implementation and readiness for future enhancements.

In conclusion, the VSAC prototype demonstrates significant potential in healthcare monitoring systems by effectively compressing and transmitting vital signs data with customization trade-offs in CR, CT, and PRD. These findings underscore the model's adaptability to varying data sizes and clinical priorities, offering a reliable solution for real-time health monitoring in smart city environments. Further research could refine the adaptive compression module to optimize configurations based on input data characteristics and real-time system demands.

5.4. VSAC results comparison

According to Table 1, which describes for each related work, the respective problem addressed, the approach, the technique used, and the results obtained, it can be observed that the main problems they focus on solving are related to energy savings, storage space reduction, and information quality. The same approach and criteria were used to describe VSAC for comparative analysis, as shown in Table 10. In the context of quality of service and restoring data, as one of the focuses presented in the proposals of RW1, RW2, RW3, RW11 and RW13, the VSAC model allows adjusting the quality of the restored information through the parameter *deviation* in the SDT. This strategy allows for the loss of information in the lossy phase of the model to the desired extent to meet the expectations of the monitoring system.

Regarding energy savings, as discussed in RW1, RW2, RW4, RW5, RW9, and RW14, the VSAC model addresses this issue, as the total processing time spent in the two compression steps and the superior compression ratio result are more efficient than the compression approaches analyzed in scenarios 1 and 2. Finally, in terms of data transmission and storage efficiency, focusing on RW3, RW4, RW6, RW7, RW8, RW10, and RW12, the VSAC model presents consistent results and has compression rate gains of up to 46% higher than the best algorithm tested.

6. Limitations

Although the VSAC model significantly advances in optimizing the management and transmission of vital signs data within healthcare monitoring systems, its focus has been primarily on efficiency, scalability, and real-time processing. Data security, although acknowledged as a critical aspect of any healthcare system, particularly one that involves sensitive personal health information, has yet to be explored in depth in this paper. As a result, potential vulnerabilities, such as data interception, unauthorized parameter manipulation, or breaches at the Edge layer, still need to be addressed. These limitations underscore the importance of integrating robust security measures into the system to ensure the privacy and integrity of patient data.

Future iterations of the VSAC model will prioritize the implementation of advanced security mechanisms to address these concerns comprehensively. The proposed enhancements include adopting end-to-end encryption protocols, such as homomorphic encryption, to safeguard data throughout all stages of processing and transmission. In addition, secure API gateways, anomaly detection systems, and tamper-resistant mechanisms for compression parameters will be evaluated and integrated to reinforce the resilience of the system against cyberattacks. Using these security techniques, the model aims to ensure the confidentiality and integrity of sensitive data without compromising its efficient and adaptive framework.

Although this work has demonstrated the potential of modular architectures and hybrid compression strategies to improve the efficiency and scalability of healthcare data systems, integration of security measures will be a key focus in future research. The authors recognize that a robust information security plan is essential to protect patient privacy and maintain trust in remote health monitoring applications. Here, authentication, authorization, cryptography, and privacy strategy frameworks should be combined to be aggregated in VSAC in the future.

7. Conclusion

Consider how valuable it would be to continuously monitor the vital signs of the entire population for the municipal health system. Not only can individuals be alerted about healthcare problems, but

Table 10
Taking Table 1 as benchmark, here we present VSAC properties and novelties when compared to the analyzed related work.

Problem adressed	Approach	Technique	Results
High bandwidth consumption and latency in the transmission of large volumes of vital signs data	Combine lossy and lossless compression methods in two steps. The first step is lossy and adaptive compression, adjusting the desired compression rate. It then sends it to the second round of compression, which is lossless.	Using the Swinging Door Trending compression algorithm as a lossy method, adjustable according to notifications received from the monitoring system, the compressed package is sent to the second compression stage, which can be Huffman Coding or LZW, depending on the file size.	The compression rates obtained with the prototype perform up to 42% higher when compared to lossless algorithms and up to 46% higher than lossy algorithm.

public policies can also be orchestrated to attack social and regional problems. To support these affirmations, we have proposed a VSAC model that adds to the state-of-the-art a novel architecture that combines compression techniques as a transmission optimization strategy for streaming healthcare data on the network. Again, we understand that sometimes gaining short milliseconds in a single transmission does not matter, but this fact gains relevance when dealing with thousands of citizens, several monitored vital signs, and periodic transmissions. Therefore, reducing network bandwidth is essential for data reliability and reducing data loss in a smart city infrastructure setting.

VSAC first employs lossy compression in such a way that the most relevant information from each signal is preserved. This is crucial for healthcare applications, as a problem here can compromise the quality of analysis and diagnosis. In other words, we adjust the compression rate to find the balance between the amount of data discarded and the necessary for analysis. In the second compression stage, a lossless strategy is performed. STD (Swinging Door Trending) was employed in the first stage, while Huffman and LZW algorithms were orchestrated to support our second stage. This combination was helpful in achieving compression rates as follows: up to 42% higher compared to lossless algorithms and up to 46% higher than lossy algorithms.

As future work, the prototype could evolve, including support for more vital signs such as body temperature, SpO2, and respiratory rate, and observation of different citizen conditions (combining ages and health situations). Moreover, we intend to explore the potential of ZIP and LZ77 compression methods to enhance overall compression efficiency. Finally, we plan to develop a smart city prototype in São Leopoldo city, Brazil, to observe the VSAC model and other processing components in an edge-fog-cloud architecture.

CRediT authorship contribution statement

Alexandre Andrade: Writing – review & editing, Writing – original draft, Visualization, Validation, Funding acquisition, Conceptualization. **Cristiano André da Costa:** Writing – review & editing, Writing – original draft, Methodology, Data curation. **Alex Roehrs:** Writing – review & editing, Writing – original draft, Software, Formal analysis. **Debora Muchaluat-Saade:** Writing – review & editing, Writing – original draft, Validation, Formal analysis. **Rodrigo da Rosa Righi:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank to the following Brazilian agencies: CNPq (processes 402339/2024-0 and 305263/2021-8), FAPERGS (23/2551-0002202-8), and CAPES (process 001).

Data availability

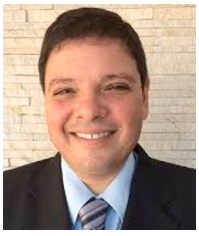
Data will be made available on request.

References

[1] H.M. Kamdjou, D. Baudry, V. Havard, S. Ouchani, Resource-constrained extended reality operated with digital twin in industrial internet of things, IEEE Open J. Commun. Soc. (2024).
[2] M.B. Dowlatshahi, M.K. Rafsanjani, B.B. Gupta, An energy aware grouping memetic algorithm to schedule the sensing activity in WSNs-based IoT for smart cities, Appl. Soft Comput. 108 (2021) 107473.
[3] J. Chen, X. Zhang, L. Xu, V.H.C. de Albuquerque, W. Wu, Implementing the confidence constraint cloud-edge collaborative computing strategy for ultra-efficient arrhythmia monitoring, Appl. Soft Comput. 154 (2024) 111402.
[4] H. Zhang, S.-x. Nan, Z.-h. Liu, J. Yang, X.-f. Feng, Lossless and lossy remote sensing image encryption-compression algorithm based on DeepLabv3+ and 2D CS, Appl. Soft Comput. 159 (2024) 111693.
[5] S. Kalaivani, C. Tharini, K. Saranya, K. Priyanka, Design and implementation of hybrid compression algorithm for personal health care big data applications, Wirel. Pers. Commun. 113 (1) (2020) 599–615.
[6] J. Azar, A. Makhoul, M. Barhamgi, R. Couturier, An energy efficient IoT data compression approach for edge machine learning, Future Gener. Comput. Syst. 96 (2019) 168–175.
[7] M.H. Vakil, Z. Shirmohammadi, LPRLC: Linear predictive run length coding to improve energy consumption of WBANs, 2024.
[8] Y. Chang, G.E. Sobelman, Lightweight lossy/lossless ECG compression for medical IoT systems, IEEE Internet Things J. (2023).
[9] L. Xiao, Y. Meng, K. Wu, Adaptive compressed classification for energy efficient activity recognition in wireless body sensor networks, in: 2018 4th International Conference on Big Data Computing and Communications, BIGCOM, IEEE, 2018, pp. 41–45.
[10] M. Hanoune, M.E.G. Lysmos, Data compression mechanisms in an intelligent e-health gateway for medical monitoring applications, Procedia Comput. Sci. 175 (2020) 578–584.
[11] C. Passos, C. Pedroso, A. Batista, M. Nogueira, A. Santos, GROWN: Local data compression in real-time to support energy efficiency in WBAN, in: 2020 IEEE Latin-American Conference on Communications, LATINCOM, 2020, pp. 1–6, http://dx.doi.org/10.1109/LATINCOM50620.2020.9282319.
[12] T. Lu, W. Xia, X. Zou, Q. Xia, Adaptively compressing IoT data on the resource-constrained edge., in: HotEdge, 2020.
[13] S.K. Das, M.Z. Rahman, A compression technique for electronic health data through encoding, in: 2021 International Conference on Electrical, Communication, and Computer Engineering, ICECCE, IEEE, 2021, pp. 1–6.
[14] A.K. Idrees, S.K. Idrees, R. Couturier, T. Ali-Yahiya, An edge-fog computing-enabled lossless EEG data compression with epileptic seizure detection in IoMT networks, IEEE Internet Things J. 9 (15) (2022) 13327–13337.
[15] M.S. Khlief, A.K. Idrees, Efficient EEG data compression technique for internet of health things networks, in: 2022 IEEE World Conference on Applied Intelligence and Computing, AIC, IEEE, 2022, pp. 403–409.
[16] A. Hossein, A. Sheikh, A novel hybrid medical data compression using huffman coding and LZW in IoT, IETE J. Res. (2022) 1–15.
[17] D. Amiri, J. Takalo-Mattila, L. Bedogni, M. Levorato, N. Dutt, Sic-edge: Semantic iterative ecg compression for edge-assisted wearable systems, in: 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, 2022, pp. 377–385.
[18] A.K. Idrees, M.S. Khlief, Efficient compression technique for reducing transmitted EEG data without loss in IoMT networks based on fog computing, J. Supercomput. (2023) 1–26.
[19] S.K. Idrees, A.K. Idrees, New fog computing enabled lossless EEG data compression scheme in IoT networks, J. Ambient. Intell. Humaniz. Comput. (2022) 1–14.

[20] M.H. Kashani, M. Madanipour, M. Nikravan, P. Asghari, E. Mahdipour, A systematic review of IoT in healthcare: Applications, techniques, and trends, J. Netw. Comput. Appl. 192 (2021) 103164.
[21] C. Sobin, A survey on architecture, protocols and challenges in IoT, Wirel. Pers. Commun. 112 (3) (2020) 1383–1429.
[22] S. Bansal, D. Kumar, IoT ecosystem: A survey on devices, gateways, operating systems, middleware and communication, Int. J. Wirel. Inf. Netw. 27 (3) (2020) 340–364.
[23] S. Krishnamoorthy, A. Dua, S. Gupta, Role of emerging technologies in future IoT-driven healthcare 4.0 technologies: A survey, current challenges and future directions, J. Ambient. Intell. Humaniz. Comput. (2021) 1–47.
[24] H.A. Alharbi, B.A. Yosuf, M. Aldossary, J. Almutairi, Energy and latency optimization in edge-fog-cloud computing for the internet of medical things, Comput. Syst. Sci. Eng. 47 (1) (2023).
[25] S. Ahmad, I. Shakeel, S. Mehfuz, J. Ahmad, Deep learning models for cloud, edge, fog, and IoT computing paradigms: Survey, recent advances, and future directions, Comput. Sci. Rev. 49 (2023) 100568.
[26] G.P. Mattia, A. Pietrabissa, R. Beraldi, A load balancing algorithm for equalising latency across fog or edge computing nodes, IEEE Trans. Serv. Comput. 16 (5) (2023) 3129–3140.
[27] C. Passos, C. Pedroso, A. Batista, M. Nogueira, A. Santos, GROWN: Local data compression in real-time to support energy efficiency in WBAN, in: 2020 IEEE Latin-American Conference on Communications, LATINCOM, IEEE, 2020, pp. 1–6.
[28] V.F. Rodrigues, R.d.R. Righi, L.M. Ceschini, B.C.L. Bellini, B. Donida, C.A. da Costa, On revisiting vital signs IoT sensors for COVID-19 and long COVID-19 monitoring: a condensed updated review and future directions, J. Ideas Heal. 4 (Special4) (2021) 604–614, http://dx.doi.org/10.47108/jidhealth.Vol4.Iss4.192, URL https://www.jidhealth.com/index.php/jidhealth/article/view/192.
[29] A. Youssef Ali Amer, F. Wouters, J. Vranken, P. Dreesen, D. de Korte-de Boer, F. van Rosmalen, B.C. van Bussel, V. Smit-Fun, P. Duflot, J. Guiot, et al., Vital signs prediction for COVID-19 patients in ICU, Sensors 21 (23) (2021) 8131.
[30] I. Kostakis, G.B. Smith, D. Prytherch, P. Meredith, C. Price, A. Chauhan, et al., Impact of the coronavirus pandemic on the patterns of vital signs recording and staff compliance with expected monitoring schedules on general wards, Resuscitation 158 (2021) 30–38.
[31] Y. Yang, X. Zhao, T. Han, Z. Li, F. Pan, Compression of electrical code violation recognition data using the improved swinging door trending algorithm, Appl. Math. Nonlinear Sci. 9 (1) (2024).
[32] J.D.A. Correa, A.S.R. Pinto, C. Montez, E.M. Leao, Swinging door trending compression algorithm for iot environments, in: Anais Estendidos Do IX SimpÓSio Brasileiro de Engenharia de Sistemas Computacionais, SBC, 2019, pp. 143–148.
[33] G. Sharma, Analysis of huffman coding and Lempel–Ziv–Welch (LZW) coding as data compression techniques, Int. J. Sci. Res. Comput. Sci. Eng. 8 (1) (2020) 37–44.
[34] K. Sharma, K. Gupta, Lossless data compression techniques and their performance, in: 2017 International Conference on Computing, Communication and Automation, ICCCA, IEEE, 2017, pp. 256–261.
[35] N. Beemkumar, S. Riyat, D. Kumar, An investigation of lossless and lossy data compression & source coding, in: 2024 15th International Conference on Computing Communication and Networking Technologies, ICCCNT, IEEE, 2024, pp. 1–6.
[36] I. Cygankiewicz, W. Zareba, Heart rate variability, Handb. Clin. Neurol. 117 (2013) 379–393.
[37] A.M.A. Hassan, S. Mohsen, Compression of electrocardiogram signals using compressive sensing technique based on curvelet transform toward medical applications, Multimedia Tools Appl. (2024) 1–17.
[38] S. Rajankar, O. Rajankar, S. Talbar, V. Raut, Signal adaptive threshold for ECG signal compression using false discovery rate approach, Circuits Systems Signal Process. (2024) 1–25.
[39] T. Moein, M.A. Keramati, H. Moinzad, Optimizing the performance reliability of diagnostic equipment and wearable sensors and medical devices in IOMT, Int. J. Nonlinear Anal. Appl. (2024).
[40] Y. Chen, S. Niu, Y. Wang, S. Xu, H. Song, M. Tan, Towards robust and efficient cloud-edge elastic model adaptation via selective entropy distillation, 2024, arXiv preprint arXiv:2402.17316.
[41] M.K. Upadhyay, M. Alam, et al., Load balancing techniques in fog and edge computing: Issues and challenges, in: 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), 5, IEEE, 2024, pp. 210–215.

Alexandre Andrade is Ph.D. student at Universidade do Vale do Rio dos Sinos, Brazil. He has expertise of finantial systems, load balancing, Internet of Things and cloud computing. In the last years, he is putting efforts towards enabling systems that join healthcare and smart cities areas. Alexandre has several articles in the computer science community, being published with IEEE, ACM and Elsevier.



Cristiano Andre da Costa received the Ph.D. degree in computer science from UFRGS University, Brazil, in 2008. He is a full professor at the Universidade do Vale do Rio dos Sinos, Brazil, and a researcher on productivity at CNPq (National Council for Scientific and Technological Development). His research interests include ubiquitous, mobile, parallel, and distributed computing. He is a member of the ACM, IEEE, IADIS, and the Brazilian Computer Society.

Alex Roehrs is a professor of undergraduate and graduate courses at the Universidade do Vale do Rio dos Sinos (UNISINOS). He holds a Master's (2012) and a Ph.D. (2019) in Applied Computing from the PPGCA (Graduate Program in Applied Computing), as well as a Lato Sensu Specialization in Computer Networks and Internet Applications (2005) from UNISINOS. He also works professionally as a Project Manager, Analyst and Systems Architect.

Debora Muchaluat-Saade holds a degree in Computer Engineering (1992), a master's degree in Computer Science (1996) and a Ph.D. in Computer Science from the Pontifical Catholic University of Rio de Janeiro (2003). She is a full professor at the Fluminense Federal University (UFF). Her areas of interest are multimedia, mulsemedia, computer networks, wireless networks, smart grids, the Internet of Things, interactive digital television, and digital health.

Rodrigo da Rosa Righi is assistant professor and researcher at University of Vale do Rio dos Sinos, Brazil, where he advises master and Ph.D. students. Rodrigo concluded his postdoctoral studies at KAIST — Korea Advanced Institute of Science and Technology, under the following topics: RFID and cloud computing. He obtained his Ph.D. degree in Computer Science from the UFRGS University, Brazil, in 2005. His research interests include load balancing and process migration. Finally, he is a senior member of the IEEE and ACM.