SaECG: a new FHIR Data format revision to enable continuous ECG storage and monitoring

Abdallah BENHAMIDA
Doctoral School of Applied Informatics and Applied Mathematics, BioTech Research Center, EKIK, Óbuda University
Budapest, Hungary
benhamida.abdallah@biotech.uni-obuda.hu

Alaa KANAS
BioTech Research Center, EKIK, Óbuda University
Budapest, Hungary
alaa.kanas@stud.uni-obuda.hu

Miklós VINCZE
BioTech Research Center, EKIK, Óbuda University
Budapest, Hungary
vkbubu19@stud.uni-obuda.hu

Kristof Tamas PAPP
BioTech Research Center, EKIK, Óbuda University
Budapest, Hungary
pappkrifi@stud.uni-obuda.hu

Mera ABBASSI
BioTech Research Center, EKIK, Óbuda University
Budapest, Hungary
mera.abbassi@gmail.com

Miklos KOZLOVSZKY
John von Neumann Faculty of Informatics, Institute of Biomatics, Óbuda University
Budapest, Hungary
kozlovszky.miklos@nik.uni-obuda.hu

***Abstract* — Information Technologies are enabling integrated healthcare processes with effective information exchange between the patient and the medical professionals. ECG is one of the key human vital signs, used by a vast number of clinicians. Fast Healthcare Interoperability Resources (FHIR) created by Health Level Seven International (HL7), Digital Imaging and Communications in Medicine (DICOM), and Standard communications protocol for computer assisted electrocardiography (SCP-ECG) are representing major digital health-data archiving standards. Most of these standards are using XML-based/JSON-based or binary formats. The interoperability and compatibility between the various ECG formats are valid problems since many years. ECG monitoring is a challenging process due to many factors such as time limitation, data size, or device dependent parameters such as the number of used parallel sampling channels. Nowadays' portable and wearable devices with non-invasive single-lead ECG capability are able to provide continuous long-term ECG data streaming that can last for many hours or even for weeks of monitoring. Available standards enable ECG data storage for a limited time, lasting from 30 seconds up to few minutes. FHIR is a medical data storage standard by HL7 for health data storage and exchange. In this paper, we propose a new FHIR (HL7-aECG) extension format for stream-like ECG, namely Stream-enabled annotated ECG (SaECG). The new format is validated successfully in our living lab and is capable to be used easily both by clinicians and researchers.**

Keywords— ECG, HL7, FHIR, streaming data, aECG, Monitoring, continuous vital sign monitoring

I. INTRODUCTION

Nowadays, medical data collection and digitalization are two of the most important processes in the field of IT. With the big advent of technologies and microelectronics in general, almost all hospitals want to digitalize its paper-based medical records, especially ECG records collected using traditional 6 and 12-channel ECG equipment. The raise in number of medical ECG devices that differs in terms of channels number and device types (eg.: single channel ECG mobile wearable devices which are made for the daily use during different daily activities) prompted researchers to find some ways to store the collected stream-like data in a digital format for further use in analysis and monitoring processes.

The boost in IT and in storage capacities together with the support provided for this area helped in developing further and improving the pre-existing standards and data formats (e.g.: SCP-ECG, DICOM-ECG, and HL7 aECG), as well as making whole new standards and data formats to store medical data in a digital format [1]. These standards are, mostly, non-compatible, they poses interoperability problems, and are very challenging to save stream-like ECG data from devices that sends stream-like ECG data or even for period-changing short or long term ECG data [2], streaming ECG devices could operate up to multiple days or weeks of ECG data monitoring using single-channel ECG mobile devices made for daily use during different daily activities [3]. On the other hand, further procedures, such as ECG data analysis, prediction, monitoring, or decision-making, are very hard to perform and challenging tasks due to short-period data collection procedures, and the single state monitoring where the patient is present at the hospital but not during a daily event [4].

Meditech CardioBlue and Savvy Pcard are some of the stand-alone ECG mobile devices which are used to monitor specifically the electrocardiogram activity of the heart together with a specific software for a computer or a mobile device to store or visualize the data [5]. Streaming ECG data is, in many cases, not considered while implementing medical data standards for data storage and is still an open research area for enabling storage of long-term ECG data. FHIR is a medical data storage standard by HL7 for health data storage and exchange. In this paper, we propose an extended version of HL7 standard based on FHIR (release 4, version 4.0.0) to enable long-term stream-like ECG data storage. This paper focus on the problem of data collection from streaming ECG devices and storage given the time difference between every two successive samples where the period is varying from one value to another.

II. ECG DATA STANDARDS

ECG data formats are already available for clinical and research use; many standards are set up for this purpose, each has its proper way for data storage solutions (eg.: binary, XML based, etc.). Interoperability between these formats is a real problem for data migration from one format to another. HL7 annotated ECG (aECG), SCP-ECG, DICOM ECG are the most common ECG formats.

HL7 is an XML-based standard for medical data digitalization created by HL7' RCRIM (Regulated Clinical Research Information Management), where HL7 aECG is a sub-standard that supports ECG data storage and display [2], [3]. It was established in 2001, ANSI approved in 2004. Its format can include many time-bounded ECG waveforms annotations for the given time. HL7 RIM (Reference Information Model) presents the message model basis for the HL7 aECG standard, and aECG R-MIM (Refined Information Model) defines the basis XML schema for the format [3], [6], [7]. HL7 aECG enables the definition different annotations from the ECG record such as the P wave, QRS wave, T offset, peak R time and amplitude…etc. [3] it supports up to 12-lead ECG records with a maximum observation time of 30 seconds but unfortunately, HL7 aECG doesn't support ECG data stream-like data record and storage.

A. FHIR specification for HL7 aECG format

The following (Fig. 1) presents the basic FHIR specification of the XML structure for the observation file that is used to save up to 12-leads ECG data with a sampling time of 30 seconds at maximum. From the given structure we can extract the following components:

Fig. 1. FHIR's basic XML structure for HL7 aECG observation file.[8]

- (a) is the bigger XML node that contains the information regarding the whole observation process.
- (b) presents the set of nodes that defines basic information regarding the observation such as the "id", which defines an aECG record for HL7. The observation status, the subject patient, the performer of the given observation, and Some extra information about the used device.
- (c) is the node that defines a one lead component from the ECG record.
- (d) defines how to store an ECG observation record with multiple leads.
- (e) presents a set of nodes used to define the currently presented lead from the ECG record. It contains some parameters such as the period in milliseconds between each ECG value, the minimum and the maximum voltage within the recorded data.
- (f) is the most important XML node which is used to store the sensed ECG data for the current observation's lead. The values are stored in an XML node' attribute called "value" inside the node "<data>" and are separated with a single white character (space character).

B. The new proposed HL7 aECG (FHIR-based) format for streaming ECG data

Because we are dealing with continuously streaming data, the exact time information of the measured ECG data plays an important role in the final treatments and analysis. (Fig. 2) shows an example of the time division of many time periods of ECG data measurement. It defines an example of an observation scenario using an ECG sensor over an interval of time, called "I0" for instance, where we have some short periods of time in which, the sensor is not able to sense and send data due to some disturbances, where we have:

I: total time of the observation.

I1: sensing time interval of 3 hours length.

I2: idle time interval of 10 minutes length.

I3: sensing time interval of an hour length.

I4: idle time interval of 3 minutes length.

I5: sensing time interval of a day length.

I6: idle time interval of an hour length.

I7: sensing time interval of 40 minutes length.

This example illustrates the importance of saving all independent timestamps of measurement for each ECG data value separately. That is why we need to implement two extra fields. One contains the Unix timestamp information of each component of the sensed data, this extra element, called "<timestamp>", is the element that defines the starting timestamp of the current observation and defines the effective date time "<effectiveDateTime>" in the usual old HL7 aECG format. The second element, called "<tsdifference>", is a sequence of integer values that defines the timestamp difference in milliseconds between two sequential values; this element eliminates the element "<period>" of the usual aECG format. So, the final structure for this extra field's value will be as follows:

Fig. 2. Example of an observation timeline made using an ECG sensor

"X1 … Xi-1 Xi Xi+1 … Xn", 1 < i < n

Where:

- "X1" is the timestamp difference between the first sensed value and the starting timestamp of the current observation component and is usually equal to 0 depending on the sensor' performance.
- "Xi" is the timestamp difference between the (ith) value of the record and its previous value at index (i-1).
- "n" presents the number of the sensed ECG values.

In order to have the exact information about the sensed data time, a datatype change must be done, so that we need to replace the "gYear", and "gYearMonth" based time notation with the following: "dateTimeStamp" which is derived from "dateTime". For an exact information about the lexical representation of the dateTimeStamp, please refer to the following link: " https://www.w3.org/TR/xmlschema11-2/#dateTimeStamp ". Some extra information about the used device must be included in the observation's file; this information includes the sensor's type, name, and unique identifier…etc.

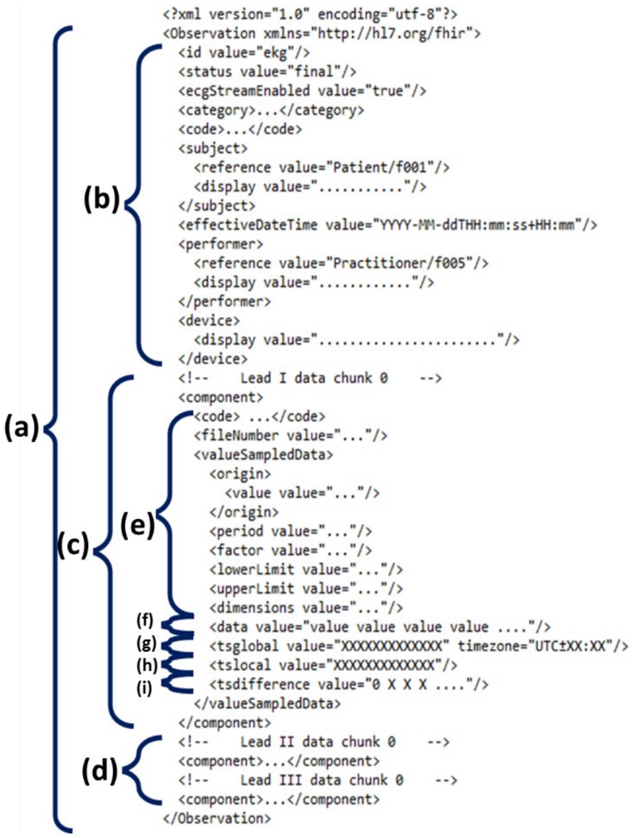

Fig. 3. The new FHIR's basic XML structure for HL7 aECG observation file.

In order to ensure interoperability between the usual old HL7 aECG format and the new proposed format, we can keep the elements "<effectiveDateTime>" which is an ISO string format of the starting time, and the "<period>" which can contain the median value of all the timestamp differences because this is a single period value which is valid for a normal observation but not for a streaming ECG data. (Fig. 3) presents the new proposed format for HL7 aECG in order to enable streaming ECG data storage for this standard.

This new proposed structure contains the same elements from (a) to (f) with respect to the preexisting aECG format, in addition to two new XML nodes (g) and (h) where:

- (g) is the "<tsglobal>" node that defines the starting Unix timestamp of the given observation. This should be stored from a trusted device other than the sensor itself. It could be used to compare with the Unix timestamp of the sensor, and to recover the correct timestamp of the beginning of the observation session if the sensor fails to give a correct time. The timestamp is included in the "value" attribute, and the current time zone is saved in the "timezone" attribute.
- (h) is the "<tslocal>" node that defines the starting Unix timestamp of the given observation. This should be stored from the used sensor itself.
- (i) is a sequence of integer values that defines the timestamp difference in milliseconds between two sequential values.

Some key different points should be taken into account while using this new format as follows:

- A new node "<ecgStreamEnabled>" with value "true" is included in the point (b) of figure 3. This gives the information that the present file is saved from a streaming ECG device.
- Depending on the used device and its data acquisition frequency, a data size limit should be used per each file in order to keep balance with the processing performance.
- A new file is created in case of reaching this limit which contains the number of the sequential file number. The last part of the file name defines the number of the file in the same observation, E.g. "observation1.part1.xml". The file number is also saved in the value attribute of the new node "<fileNumber>" in the point (e) of figure 3.
- A new file is created in case of expected or unexpected stop of the measurement process due to any reason. Either a sensor / device failure, or a connection or storage failure. The new timestamp values of the new measurement session are stored in the next file.
- Every parameter which is dependent with non-streaming observation' lead, such as "<period>" and "<factor>", should have a value of an empty string. This means that the given values are stored in the parameters of the streaming ECG presented previously in points (g), (h), and (i).

C. Why HL7 aECG?

Compared with other standards such as SCP-ECG and DICOM ECG, HL7 aECG presents a very good starting point to make an ECG data converter that enables interoperability between data formats due to its XML-based format [9]. It is an easy to use standard which we can improve to enable stream-like ECG data storage. In the meanwhile, it is human readable. In the other hand, SCP-ECG [10] and DICOM ECG

[11], [12] have binary-based formats which, compared to HL7 aECG, has a very small file size but not human readable [13], [14].

The Health Level 7 (HL7) is created by HL7' Regulated Clinical Research Information Management (RCRIM), it defines a set of standards used for medical data digitalization and storage, it is used by the US Food and Drug Administration (FDA); it is mainly used for digital medical data exchange between medical service providers. Its Annotated ECG (aECG) standard is one of the standards intended for storing ECG medical data in a standard format [9].

HL7 aECG is human readable and is ASCII-based format due to the use of XML file format which offers all advantages of XML structure; It enables easy search for data mining applications and provides easy access to data while programming process or while integration with other XML tools. [9]

III. HL7 aECG Data, FHIR Storage Specification and the Programming Process

During the programming process of an HL7 aECG tool based on the FHIR specification (release 4, version 4.0.0) using the new proposed format for ECG streaming data, many details should be implemented such as the number of leads, the data file size limit in comparison with the reading or writing time consumption.

A. *Streaming ECG data storage algorithm*

Because we are dealing with continuous streaming ECG data for many days or weeks using some ECG sensors that have mostly one single ECG channel, both the sensed data and the exact time information of the measured ECG data plays an important role in the final treatments and analysis of the patient's ECG data. The proposed algorithm for this purpose consists of the following steps:

1. Collect the data and timestamps for a given number of minutes.
2. Calculate the array of time difference in milliseconds between each value and its successive value.
3. Prepare the new proposed XML file format with all the necessary information about the observation.
4. Save the collected data in the new format separated with white character (space character) to the attribute "value" of the node "data".
5. Save the starting timestamp to the attribute "value" of the node "timestamp".
6. Save the calculated array of time differences separated with white character (space character) to the attribute "value" of the node "tsdifference".
7. Repeat steps 2-6 for all channels of the ECG, each lead in a separate node "component".
8. Repeat steps 1-7 for every separate ECG observation or every observation that exceeds the time limit, each observation in a separate XML file in the new proposed format.

One question that could appear in this proposed algorithm is why to save the timestamp differences between each value and its successive value? The answer is simply the enormous file size that could reach very big sizes, which then will disturb the writing and reading processes from the XML files given that one timestamp that includes milliseconds consists from 13 characters while we can save these characters using the calculation of timestamp differences technic which will consist of 1 to 2 character for ECG data for every timestamp difference.

The other point to discuss in here regarding this algorithm is what is exactly the time limit for every observation separately? The answer is discussed later in the section of results regarding the writing and reading processes from or to XML files. The collected data is stored temporarily to an intermediate JSON format in order to make it easier for adding every newly received streamed data. This intermediate is an optional format that could be used for writing or reading processes performed on XML files, it makes it easier for further interoperability researches between all preexisting standards and could be saved in a JSON file as well.

B. *Streaming ECG data reading and display algorithm*

After having the data stored in the given new format for continuous streaming ECG data, the question now is how to deal with this data? given that timestamp information is stored in a form of time difference between every two successive values. The reading and display algorithm are given by the following steps:

1. Read the data and timestamp information from the XML file, data and timestamp difference information are separated by a white space character.
2. Calculate the array of timestamp starting from the effective date time given by the node "timestamp" and then add the time difference in milliseconds for each value and its successive value.
3. Any data treatment, analysis, or display process could be done in this step.
4. Repeat steps 2-3 for all channels of the ECG data, each lead from a separate node "component".
5. Repeat steps 1-4 for every separate XML file, each file defines a separate observation or one single observation that exceeds the time limit for an observation.

IV. Reached Results with the New Format

(Fig. 4) shows the HL7 aECG observation file size comparison between the usual HL7 aECG and the new proposed format versus observation time in minutes from 1 single minute up to one-hour observation time in which we can observe that the file size of the new proposed format (in blue) is around 150% of the file size for the same amount of ECG data saved in the old format of HL7 aECG.

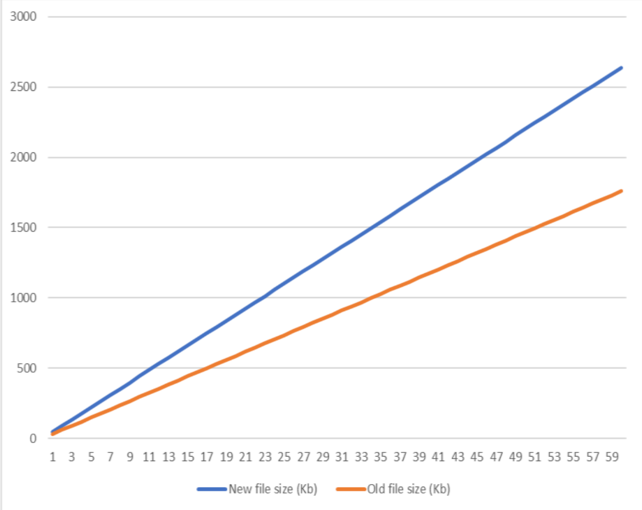

Fig. 4. aECG observation file size comparison between the usual aECG and the new proposed format versus observation time in minutes.

In the following results regarding the reading and writing time consumption from or to the old or new proposed format for HL7 aECG streaming data, every single result is repeated 10 times, then the median of the partial results is taken for that particular result; taking into consideration that these results are and will stay approximative depending on the used hardware and software for the data storage.

(Fig. 5) shows the median values for reading time from the usual old (in yellow) and new proposed format (in green) for HL7 aECG' XML file. The results are promising because of the time consumption that does not exceed 100 milliseconds, in most cases, for 60 minutes long ECG observation. The same is shown in (Fig. 6) for the writing process.

(Fig. 7) shows the set of results obtained out of the used algorithm in this research regarding the reading process from an old HL7 aECG XML file, the new proposed format, and the intermediate JSON format; where we consider the good results obtained with both the old and new HL7 aECG formats against the intermediate JSON format from time-consumption point of view. The test results are obtained from many tests performed on a windows 8.1 based environment where the used program using the proposed algorithm was developed based on .NET framework (v4.5).

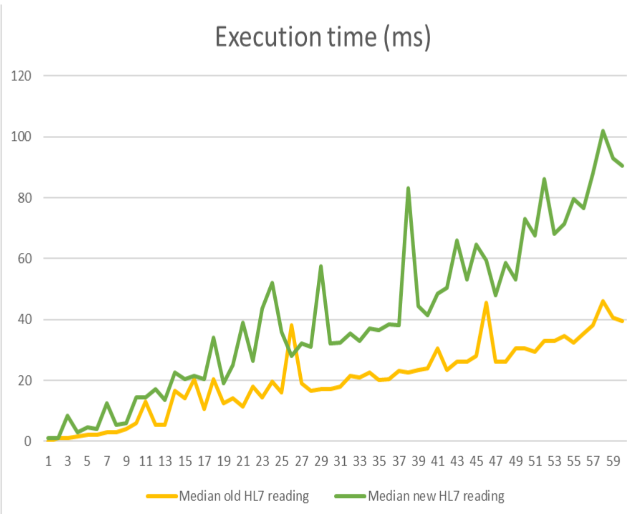

Fig. 5. Execution time for reading process of old and new proposed HL7 aECG.

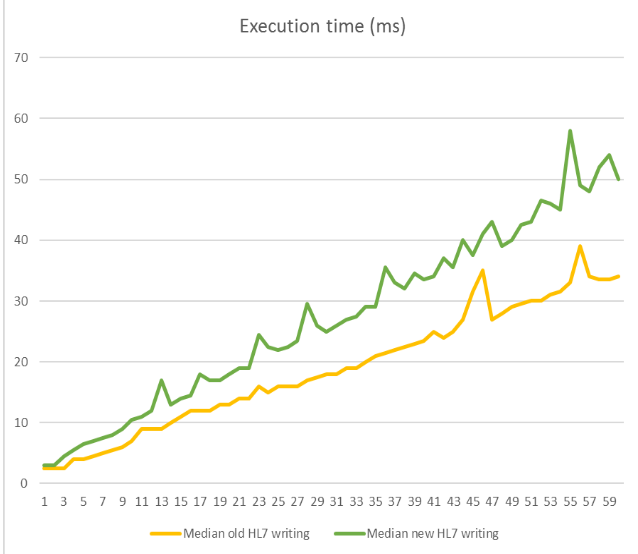

Fig. 6. Execution time for writing process of old and new proposed HL7 aECG.

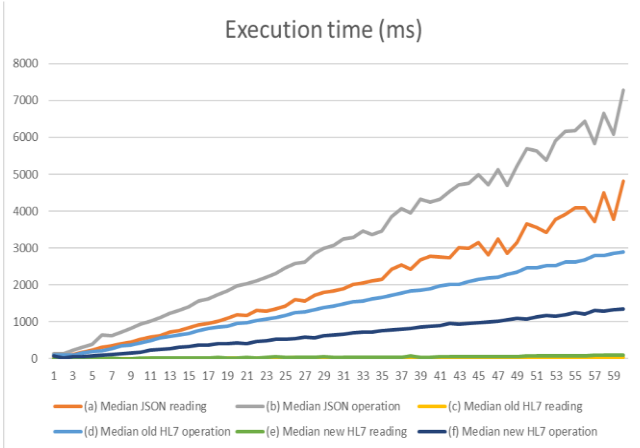

Fig. 7. Execution time for reading, treatment, and display process of old and new proposed HL7 aECG and the used intermediate JSON format.

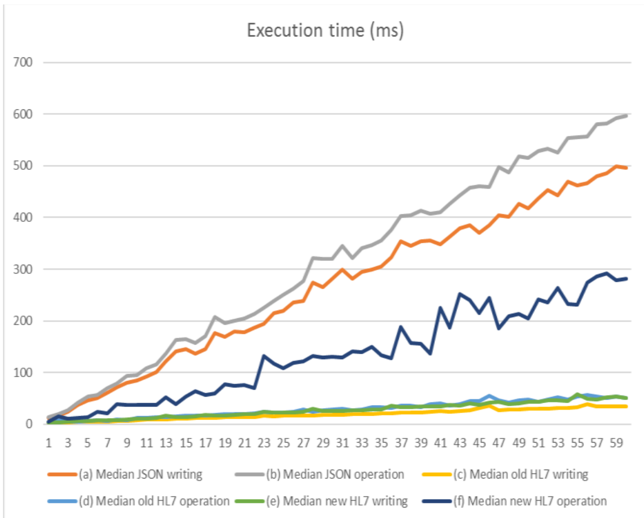

Fig. 8. Execution time for preparing and writing process of old and new proposed HL7 aECG and the used intermediate JSON format.

The new proposed format showed good results in comparison with the HL7 aECG format taking into consideration that the new format needs extra time for timestamp calculation for each value. Each tested format contains both the reading time from the associated file and the whole execution time for a reading operation including the treatment and display time. (Fig. 8) shows the same explained results for the preparation and writing process in the associated file format.

V. Summary

In the past, ECG data collection was made mainly in paper-based format where medical staff has to make analysis, extract all the main information from the ECG manually. Nowadays, many efforts and research support are provided for medical data digitalization and standardization in general and for ECG data specifically. While ECG data monitoring is done in a non-daily routine basis, many medical equipment manufacturers provide mobile devices for daily-use basis, such as single channel ECG devices (e.g.: Savvy Pcard) which can work for many days or weeks and provides streaming ECG data. Most of the used standards, both the binary and XML formats, does not support stream-like ECG data storage for long term monitoring. In this paper, we proposed a new XML-based format for stream-like ECG data based on FHIR specification for HL7 aECG format. The new proposed format enables streaming ECG data storage where the main difference is to enable saving changing time-differences between every two successive data values. This technic provides interoperability with previous versions of this standards. The new updates to the FHIR aECG could be used in both XML and JSON formats. It enables storing infinite streaming ECG data from different sensor types and different acquisition frequencies even in the same observation. The seek between different measurement files is upper bounded. This gives high performance while changing searching or viewing the data of same observation from multiple files due to the use of the timestamp component of every file. The Stream-enabled annotated ECG (SaECG)is compatible with different ECG streaming sensors and is capable to use many and independent channels. The new format is tested in our living lab and showed very good results compared to the old version. These results could be a good starting point for further research on ECG data monitoring in order to enable further procedures (eg.: analysis, monitoring, or even prediction) and to ensure interoperability between preexisting formats or standards.

Acknowledgment

The authors hereby thank the EFOP-3.6.1-16-2016-00010 project and furthermore the GINOP-2.2.1-15-2017-00073 "Telemedicina alapú ellátási formák fenntartható megvalósítását támogató keretrendszer kialakítása és tesztelése" project for the financial support. We would like to thank AIAM (applied informatics and applied mathematics) doctoral school of Óbuda University, and the Institute of Biomatics, Budapest Hungary for their support in this research.

References

[1] A. Bánhalmi, J. Borbás, M. Fidrich, V. Bilicki, Z. Gingl, and L. Rudas, "Analysis of a Pulse Rate Variability Measurement Using a Smartphone Camera," J. Healthc. Eng., vol. 2018, p. 4038034, 2018.

[2] P. Bakucz, S. Willems, and B. A. Hoffmann, "Universal Fluctuations in Very Short ECG Episodes," Acta Polytech. Hung., vol. 11, no. 7, p. 10, 2014.

[3] A. BENHAMIDA, A. ZOUAOUI, G. SZÓCSKA, K. KARÓCZKAI, G. SLIMANI, and M. KOZLOVSZKY, "Problems in archiving long-term continuous ECG data – a review," in 2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 2019, pp. 263–268.

[4] S. Khoor et al., "Heart Rate Analysis and Telemedicine: New concepts & Maths," in 2007 5th International Symposium on Intelligent Systems and Informatics, 2007, pp. 39–43.

[5] P. J. Lees, C. E. Chronaki, and F. Chiarugi, "Standards and Interoperability in Digital Electrocardiography. The OpenECG Project.," p. 6.

[6] B. D. Brown and F. Badilini, "HL7 Version 3 implementation guide: regulated studies–annotated ECG, Release 1," Health Level Seven Int., 2005.

[7] G. D. Clifford, F. Azuaje, and P. McSharry, Advanced Methods And Tools for ECG Data Analysis. Norwood, MA, USA: Artech House, Inc., 2006.

[8] "Observation-example-sample-data - FHIR v3.0.1." [Online]. Available: https://www.hl7.org/fhir/observation-example-sample-data.html. [Accessed: 12-Dec-2018].

[9] R. R. Bond, D. D. Finlay, C. D. Nugent, and G. Moore, "A review of ECG storage formats," Int. J. Med. Inf., vol. 80, no. 10, pp. 681–697, Oct. 2011.

[10] P. Rubel et al., "SCP-ECG V3. 0: An enhanced standard communication protocol for computer-assisted electrocardiography," in Computing in Cardiology Conference (CinC), 2016, 2016, pp. 309–312.

[11] T. Hilbel, B. D. Brown, J. de Bie, R. L. Lux, and H. A. Katus, "Innovation and advantage of the DICOM ECG standard for viewing, interchange and permanent archiving of the diagnostic electrocardiogram," in 2007 Computers in Cardiology, 2007, pp. 633–636.

[12] "DICOM Standard." [Online]. Available: https://www.dicomstandard.org/current/. [Accessed: 02-Nov-2018].

[13] C. E. Chronaki et al., "Open ECG: A European project to promote the SCP-ECG standard, a further step towards interoperability in electrocardiography," in Computers in Cardiology, 2002, pp. 285–288.

[14] M. J. B. van Ettinger, J. A. Lipton, M. C. J. de Wijs, N. van der Putten, and S. P. Nelwan, "An open source ECG toolkit with DICOM," in 2008 Computers in Cardiology, 2008, pp. 441–444.