Articulating heterogeneous data streams with the attribute-relation file format

Cite as: AIP Conference Proceedings **2173**, 020021 (2019); https://doi.org/10.1063/1.5133936
Published Online: 11 November 2019

Mario José Diván, and María Laura Sánchez Reynoso

Articulating Heterogeneous Data Streams with the Attribute-Relation File Format

Mario José Diván[1, a)] and María Laura Sánchez Reynoso[1, b)]

[1] *Economy School, National University of La Pampa, Coronel Gil 353 1st floor, Santa Rosa, CP6300, Argentina*

[a)] Corresponding author: mjdivan@eco.unlpam.edu.ar
[b)] mlsanchezreynoso@eco.unlpam.edu.ar

Abstract. The processing strategy based on measurement metadata is a data stream engine running on Apache Storm, who is able to process measures in real-time. In the data stream context, the data have no an associated limit, they are al-ways arriving. The Attribute-Relation File Format (ARFF) is used by popular software like Weka, allowing offline analysis in the machine learning and data mining area. However, the ARFF file has a finite size. The CincamimisConversor library allows exporting from the data streams organized under a measurement interchange schema to a columnar-data organization in real-time. Here, an extension to the library is introduced for supporting the real-time translating and storing from the heterogeneous data streams to the ARFF file format. This is very useful, because through the library now is possible to collect data from heterogeneous data sources (e.g. Internet-of-Thing -IoT- devices) and export them in real-time for offline analysis in Weka. Even, this could foster a lot of educational applications among IoT, the measurement process with heterogeneous sources, data stream processing strategy, and Weka. A discrete simulation was carried out, obtaining promising results. It is just required at most 0.2387 ms for translating 5000 measures, while the storing operation for them consumed less than 0.2028 ms on a Solid-State disk.

Keywords - Heterogeneous data streams, measurement, attribute-relation file format, weka, processing architecture

INTRODUCTION

The measurement allows characterizing from the industrial processes, businesses, up to different disciplines related to engineering. It takes a protagonist role when it is necessary to quantify different aspects able to be used for describing an entity under monitoring (be it tangible or not). The measures allow determining the current state of an entity jointly with its associated changes. This represents an essential aspect when a decision-making process should be carried forward, independently the application fields [1], [2].

The real-time decision making implies that the data should be processed on the fly, with the available resources at the moment in which they arrive. This constitutes a propitious field for the data stream engines, which are specially oriented to this kind of processing. In this context, the challenge is oriented to make decisions in a tolerant way, because the data processor has no control on the data sources and it should process the data such it is, without the possibility of reprocessing. The reprocessing is not a feasible alternative because new data is always arriving which replace the data before them [3]–[5].

Even, each time more the real-time data processing is related to different kinds of applications associated with the Internet-of-Things (IoT) field. In this kind of applications, measures could be collected from heterogeneous data sources, and they could be processed at the arriving time. For example, this is especially useful for monitoring of entities, such as lakes or forest in relation to floods or fire respectively [6], [7].

The Processing Architecture based on Measurement Metadata (PAbMM) is a data stream engine based on Apache Storm specialized in measurement projects. The architecture is designed as a Storm´s topology and it uses a formal framework for defining a measurement project with the aim of warrantying the repeatability and consistency related

Advances in Electrical and Electronic Engineering: From Theory to Applications (Series 2)
AIP Conf. Proc. 2173, 020021-1–020021-10; https://doi.org/10.1063/1.5133936
Published by AIP Publishing. 978-0-7354-1920-9/$30.00

to the measurement process. Thus, the comparability of the results is especially taken into account considering the heterogeneous data sources [8], [9].

The Weka is a well-known software used for analyzing and studying different kinds of algorithms related to machine learning [10], [11]. The Massive Online Analysis (MOA) and Scalable Advanced Massive Online Analysis (SAMOA) projects are projects related to WEKA but oriented to the data stream processing [4]. The Attribute-Relation File Format (ARFF) is a data format used by Weka, MOA, and SAMOA for describing a list of instances which share a set of attributes. This allows loading data from files following a given organization both Weka, MOA, SAMOA, as the rest of software able to read ARFF data files.

In PAbMM, the heterogeneous data sources are configured using the project definition, which allows associating each data source with a specific IoT device and entity under monitoring [12]. Once the data sources are configured, the data and metadata are informed to the data processor in the architecture using a measurement interchange schema for keeping traceability and consistency [13]. The *CincamimisConversor* library [14] is an extension for PAbMM which allows translating to a columnar-data organization the data and metadata coming in real-time from heterogeneous data sources.

Off course, there are other systems such as MLlib, Scikit-learn or Caret which allows load data without the necessity of a given file and using directly the data streams. In this way, if it was required, PAbMM currently is able to send directly the data stream to any data stream engine by means of an internal component named “Carriers” without the necessity of a file. However, the underlying idea here is to provide an autonomous alternative able to interpret the measurement stream and translate it to ARFF for articulating the library with Weka, MOA, and SAMOA because they are introductory systems of the discipline used by non-expert’s users.

As main contributions, i) The real-time translating from heterogeneous data sources to the ARFF data format is modeled as an extension of the *CincamimisConversor* library. This makes possible to collect data from IoT devices associated with a measurement project and to export them for its offline analysis in Weka, MOA, and SAMOA; and ii) The *CincamimisConversor* library is updated for supporting the ARFF data format as a new alternative for non-expert users, allowing the exportation (to a file or in-memory) from the columnar-data organization, or even directly from the data stream when the data arrive. In this way, a lot of educational applications could be addressed, establishing a relationship among IoT devices (i.e. heterogeneous data sources), measurement projects, data stream processing, data interoperability and measurement systems' integration (e.g. with supporting for Weka, MOA, and SAMOA). This is important because Weka provides a set of algorithms to learn the main concepts related to data mining, while MOA explains the underlying idea of data streams. Finally, SAMOA is very useful for developing portable topologies between different data stream engines.

This article is organized into seven sections. Section 2 introduces some related works. Section 3 synthesizes the ARFF data format and its uses in Weka. Section 4 describes the role of the project definition in the data stream processing strategy jointly with its incidence at the moment in which data format translation should be carried forward. Section 5 introduces the extension proposal and the integration within the *CincamimisConversor* library is shown. Section 6 outlines simulation results associated with the consumed time by the translating operation. Finally, some conclusions and future works are addressed.

RELATED WORKS

In [15] a system architecture is introduced with the aim of carrying forward a trust-worthiness analysis. It is applied in the car monitoring context based on IoT connected devices. The underlying idea is using the real-time data jointly with the previous experiences for determining whether a given driver is driving in a dangerous way or not. In this case, the Weka library is used for interchanging data under the Attribute-Relation File Format (ARFF). Our proposal is integrated with PAbMM which allows real-time data processing from heterogeneous data sources, fostering the interoperability through using a measurement interchange schema based on a measurement framework.

A proposal of integration between Apache Spark (spark.apache.org) and Cassandra (cassandra.apache.org) is introduced for describing the versatility of the real-time data processing in Spark Streaming, with the flexibility related to the distributed file system from Cassandra [16]. In this way, an approach termed Smart Cassandra Spark Integration (SCSI) is detailed with the aim of integrating NoSQL data stores (i.e. a huge but finite data repository) with the distributed systems managed through Apache Spark (e.g. unbounded data streams by means of Spark Streaming). In *CincamimisConversor* library, the columnar-data organization jointly with the ARFF data format is completely processed in memory, using the available resources and generating the output at the same time in which the data arrive.

In [17], the mldr.datasets tool is introduced as an R package with the aim of making easy the labeled data management. Among the included functions, the exportation to different kinds of data formats is present (e.g. ARFF). It is primarily thought for working on finite data sets. As a difference, our proposal is oriented to the boarding of unlimited data streams jointly with the finite data sources.

Ahmed, Ferzund, and others introduced in [18] an analysis related to data formats used in Big Data Analytics. They highlight the importance of managing different kinds of data formats thinking in the heterogeneity of related available tools. A comparative analysis of data formats (e.g. ARFF, Parquet, Avro, etc.) taking into consideration the performance, speed, and supported platforms were driven. Our proposal is oriented to carry the translation forward in memory, and then, the outcome is the ARFF data file. Our proposal is oriented to the in-memory data translating for generating a persistent outcome under the ARFF data format.

Bifet and others [19] introduce StreamDM, an open source library which incorporates a set of data mining and machine learning algorithms. It works on the top of Apache Spark and introduces a set of advanced algorithms feasible to be applied in real-time data processing. The library incorporates the concept of StreamWriter which allows generating an output from a previous data stream or related tasks. As a difference, our library allows translating from the streams organized under the measurement interchange schema stream to ARFF data format, interpreting each metric (or variable) based on the project definition.

THE ARFF DATA FORMAT AND WEKA

The ARFF data format describes a list of instances which share a set of attributes. It internally incorporates the metadata describing the data structure jointly with the data itself. The header section describes the metadata (i.e. the data structure) while the data section details the data which follows the defined data structure before.

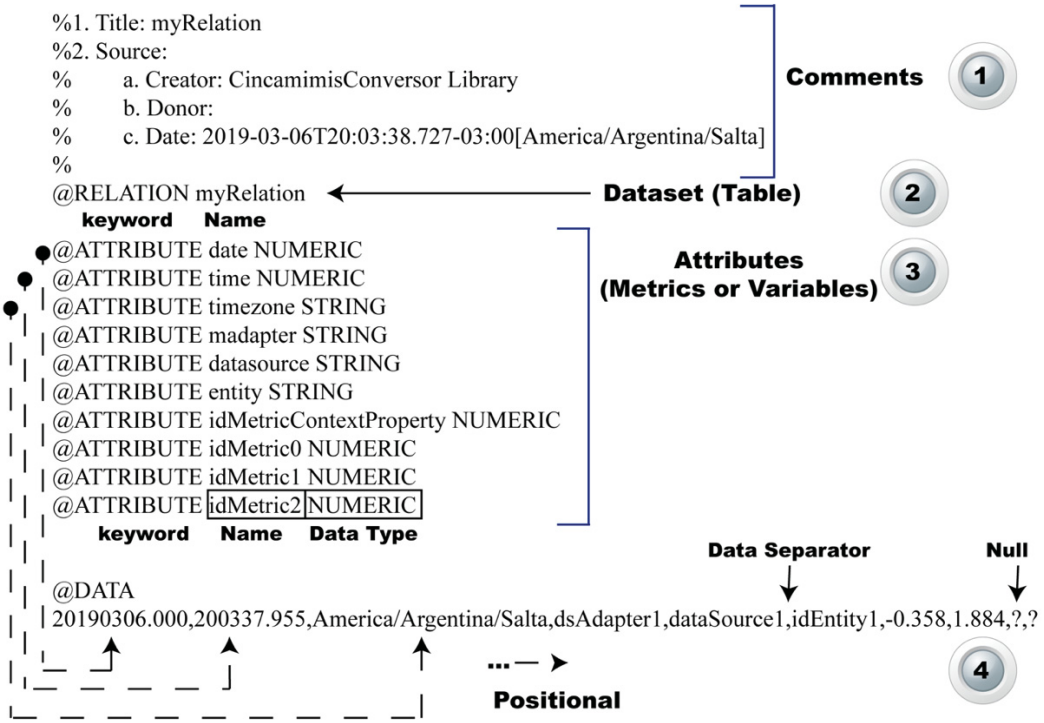

FIGURE 1. A General View of the Attribute-Relation File Format

Figure 1 synthesizes the structure of the ARFF data format. The commentary lines are started with the "%" character. They are useful for incorporating metadata describing its content and associated aim (See circle 1 in Fig. 1). The file requires that the dataset is identified by a name, what it is possible indicating the @RELATION keyword at the beginning of the line followed by the given name to the dataset (See circle 2 in Fig. 1). Each attribute (i.e. metric, column, variable, or field in this context) should be individually defined indicating the @ATTRIBUTE keyword

followed by the given name and the data type (See circle 3 in Fig. 1). Between the available data types are possible to find i) Numeric: It could represent real or integer numbers; ii) Nominal: It represents a list containing values. Each variable could assume a value but only among the listed values. In this way, the list of values corresponds with the domain-values; iii) String: It allows incorporating an arbitrary text content into a variable; iv) Date: It defines a variable able to jointly manage date and time. The default data format is ISO-8856 but it could be adapted.

Once all the attributes are defined, the starting of the data region is indicated through the @DATA keyword. Below it, each line represents an instance (i.e. a tuple, record, or file in this context) which is comma separated. When some attribute has no value for a given instance (i.e. it is null), it is represented incorporating the "?" character (See circle 4 in Fig. 1).

THE MEASUREMENT PROJECT DEFINITION AND ARFF DATA FORMAT IN THE DATA STREAM PROCESSING

PABMM is a data stream processing strategy in which the aim is oriented to automate the measurement process in a real-time data processing environment. It is based on a Measurement and Evaluation (M&E) framework termed C-INCAMI (Context-Information Need, Concept Model, Attribute, Metric and Indicator), which defines all the terms, concepts and relationships for establishing a common understanding [9], [20]. Each M&E project is defined following the agreed concepts in the framework and its definition is interchanged between heterogeneous measurement systems using the CINCAMI/Project Definition (CINCAMIPD) schema.

Basically, the M&E framework allows defining the information need (i.e. the reason which support the measurement project), the entity to monitor, the attributes which characterize the entity under monitoring, the context in which the entity is monitored, the metrics to be used to quantify each attribute, the associated measurement devices, among other aspects. Nowell, once all the concepts have been defined, they are organized under the CINCAMIPD schema with the aim of interchanging the M&E project definition between the measurement systems who need it (See Fig. 2).

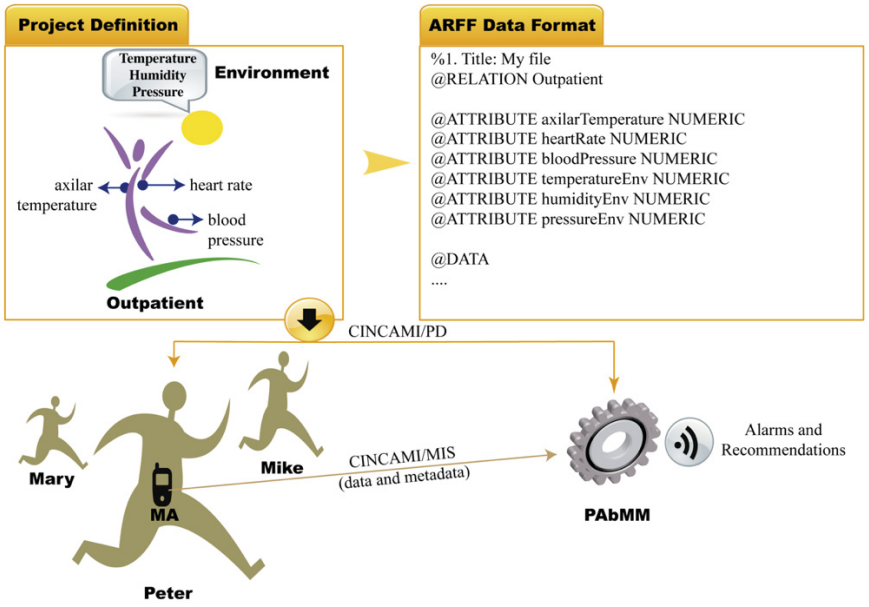

FIGURE 2. The Relationship among the Project Definition, PAbMM, and ARFF data format

Because the data sources are heterogeneous, a measurement interchange schema based on the project definition is used to foster the interoperability between the data sources and the real-time data processor (i.e. PAbMM). In this way, data (i.e. the measures coming from each data source) and metadata (i.e. tags representing the concepts previously defined in the project definition e.g. a given metric) are jointly embedded in the message. The measurement interchange schema used for doing it is termed CINCAMI/MIS (Measurement Interchange Schema) [13]. The heterogeneous data sources are connected with PAbMM through a Measurement Adapter (MA). It is responsible for

understanding each particular technology and data format from the data sources, collect their measures and translate them in a consistent CINCAMI/MIS message for sending to PAbMM (See Fig. 2).

The start-up process for a given measurement project happens in two stages in PAbMM. On the one hand, the first stage loads in PAbMM the project definition (i.e. CINCAMI/PD message), which prepares to the data processor for understanding each element defined in the project, staying ready to receive CINCAMI/MIS messages, process and interpret them. On the other hand, the second stage is carried out in parallel but on the measurement adapter (commonly located on mobile devices). Here, each measurement adapter is associated with each data source jointly the metric which is implementing. Thus, the MA stays ready for collecting measures, translate and send them to PAbMM in a continuous way.

Figure 2 synthesizes the relationship between the project definition, the entities under monitoring, PAbMM and the ARFF data format. The project definition is self-content, which implies that contains all the necessary information for implementing the measurement process. For example, it contains the information need, an entity under monitoring, the definition of each metric with the indication of the unit, scale, measurement method, etc. Once the project definition was done, it is loaded using a CINCAMI/PD message in PAbMM. Thus, it prepares the memory structures for receiving the measures through the CINCAMI/MIS messages from the MA. In addition, The CINCAMI/PD message is used for loading the project definition on each MA related to each outpatient, which allows setting up each sensor (e.g. corporal temperature, heart rate, etc.) with the corresponding metric on the mobile device.

Because the CINCAMI/PD is self-content in terms of the measurement project, the translation from the CINCAMI/PD to ARFF data format is transparent as you can see in Fig. 2. That is to say, the entity under monitoring is expressed like a relation in ARFF, each attribute and context property quantified by metrics correspond with attributes in ARFF, and the measures related to them are organized under the data section. However, if the translation requires the CINCAMI/PD message for converting to ARFF in each CINCAMI/MIS message, it would be necessary both. Well now, each CINCAMI/MIS message is organized based on CINCAMI/PD, and the measurement interchange here involves data (i.e. the measures) and metadata (i.e. the tags establishing the relationship of each data with a metric, context property, entity, etc.). In this way, it is possible just using CINCAMI/MIS to translate from it to ARFF because the ARFF data format requires knowing the relation name (i.e. the entity under monitoring present in CINCAMI/MIS as tag), the attributes (i.e. the attributes or context properties which inside of CINCAMI/MIS mark each measure), and the data (i.e. each measure itself). More details on CINCAMI/PD and CINCAMI/MIS jointly with the open source associated libraries could be reached in [12], [13].

SUPPORTING ARFF IN CINCAMIMISCONVERSOR LIBRARY

The measurement interchange schema (i.e. CINCAMI/MIS) is the way in which the data and metadata are interchanged between the data sources and the data processor, but this is strictly based on the project definition. The possibility of exporting the data to another data format is important but to do it at the moment in which the data arrives is better.

For that reason, the CincamimisConversor library [14] was released as an alternative in which the measures could be converted to a columnar data organization in real-time from the data streams. Here, an extension of this library is introduced for supporting the real-time translation to the ARFF data format from the data streams arriving under CINCAMI/MIS. Figure 3 shows the main concepts involved in the real-time exportation to ARFF data format inside the CincamimisConversor library. The original concepts have a white background, while the new concept has a filled background. The *ColumnFamily* class represents a grouping used in the columnar databases for putting together a set of columns. The number of columns in the column family is variable and dynamic, while the number of the column families is determined at the moment in which a table is created. Each column is represented through the *Attribute* class, which has correspondence with the attribute concept from the project definition (i.e. a quantifiable characteristic through a metric). The data of each column is managed through the *ColumnData* class. The column data is able to store nominal or quantitative values. Moreover, it is able to manage deterministic and estimated values. In the last case, a mathematical expectation is used for synthesizing a number from the likelihood distribution (See *Quantitative*, *LikelihoodDistribution*, and *Estimated* classes in Fig. 3).

Each CINCAMI/MIS message is incorporated in a queue which regulates the accessing (See *CincamimisQueue* class in Fig. 3). The *CincamimisTranslator* class progressively read the messages when they arrive for translating it. Because the *CincamimisTranslator* is an abstract class, the derived class must implement the necessary logic for translating from CINCAMI/MIS to the wished data format. The *TabularMode* class inherits from the *CincamimisTranslator* class and implements the necessary logic for expressing the measures under a two-dimensional

way (i.e. a table). In this way, the concept of rows is represented through the *Tuple* class, who has an *Attributes* class for managing the set of attributes (i.e. the column family and its associated columns). Once the translating to the columnar organization is made, the processed data streams under the shape of *Tuples* are grouped and managed in the *CincamimisWindow* class. This class implements a *TableView* interface, which define the necessary methods for accessing to the data based on a columnar organization. It is a logical organization for in-memory processing using batch, which could be viewed as a Window based on the measures.

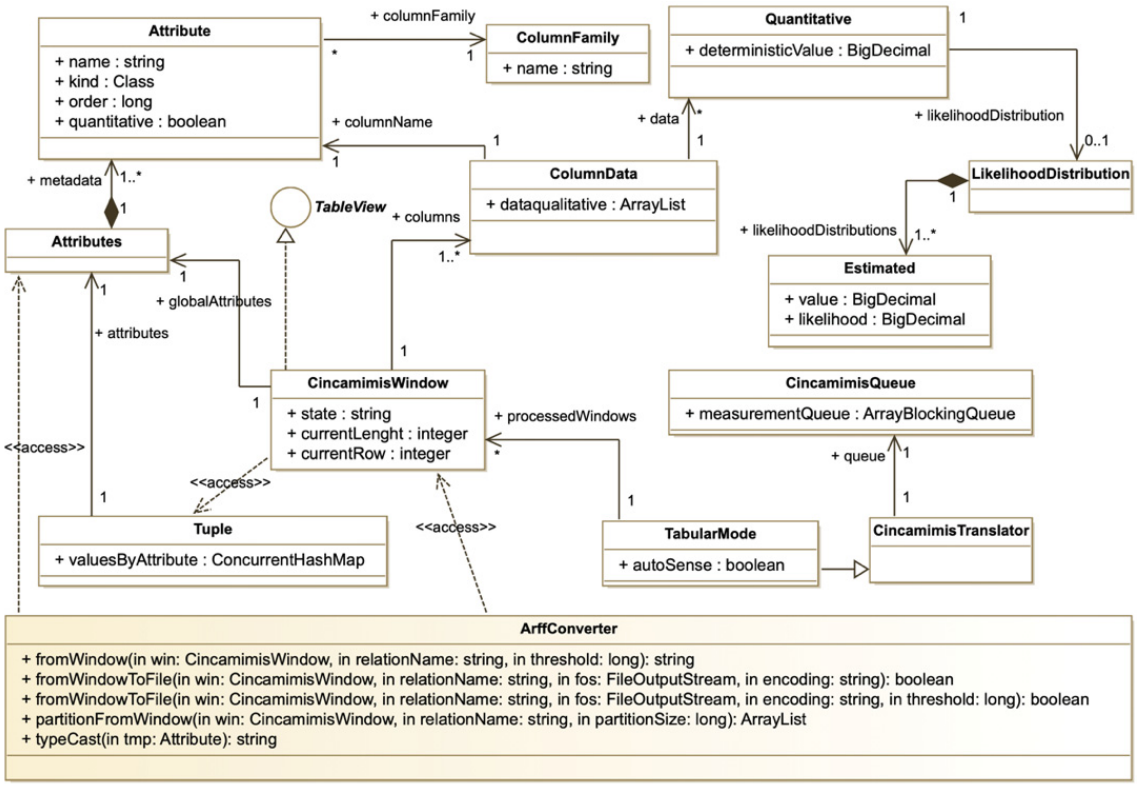

FIGURE 3. The Relationship among the Project Definition, PAbMM, and ARFF data format

Because the ARFF data format is a two-dimensional viewpoint in which the relation must be defined in terms of their attributes, here the proposed extension is based on the columnar-data organization. This kind of organization contains enough information for exporting from columnar data to the ARFF data format. For example, the *CincamimisWindow* is a grouping of *Tuples*, which each one has a specific set of *Attributes*. Each *Attribute* defines its name, data type, kind (i.e. estimated or deterministic) and the column order. The conversion logic for translating from columnar-data organization to the ARFF data format is implemented in the new *ARFFConverter* class.

The *ARFFConverter* class (See Fig. 3) introduces five methods 1) *typeCast*: It is responsible for fitting the data type from CINCAMIMIS to the supported data format in ARFF; 2) *fromWindow*: It gets an instance of *CincamimisWindow* in memory, jointly with the name to be assigned to the relation in Weka for returning the ARFF content as a UTF-8 String. The threshold parameter is useful for limiting the number of rows to be include in the translated content. When the threshold parameter is null or zero, all the content from the *CincamimisWindow* instance will be put into the output. However, when the threshold parameter is lesser than the number of measures into the Window, the number of rows in the output is limited to the defined by the parameter; 3) *fromWindowToFile*: There are two versions of this method but both methods are responsible for translating the content from the *CincamimisWindow* instance and put it into a file. On the one hand, the first method takes the *CincamimisWindow* instance and the given name for the relation in ARFF, and it makes the conversion to ARFF data format putting all the content under a specific encoding (by default UTF-8). On the other hand, the second method is similar but incorporates the threshold parameter which allows limiting the number of rows into the output; 4) *partitionFromWindow*: It allows translate all the content from the *CincamimisWindow* instance, and sequentially store it in as Strings as it needs in the measure that neither content exceeds the number of rows defined by the partition size.

In other words, the partition size is the maximum number of rows to be incorporated in each output. Finally, the partitioned content is returned as an array of Strings. It is especially useful when is necessary to limit the maximum number of rows in each partition.

Thus, the translating from heterogeneous data streams to ARFF data format can be carried out at the same time in which the data arrives. This allows an interesting integration with contexts such as Internet-of-Things, in which this kind of functionality easily would allow to export current content of devices for offline analysis later (e.g. using Weka). These changes are incorporated in the CincamimisConversor library, allowing now the simultaneous exporting to a columnar-data organization and/or ARFF data format, be it in memory or in a file. All the library components are freely available on GitHub (github.com/mjdivan/cincamimisConversor) under the terms of the Apache 2.0 license. The library is written in Java 8 and it uses the Cincamimis library (github.com/mjdivan/cincamimis) for reading and interpreting the measurement interchange schema.

SIMULATION RESULTS

A discrete simulation was carried out on a Mac Book Pro with a 2.9 GHz Intel Core i7 Processor, 16GB RAM, 500 GB SSD (Solid-State Disk), and macOS Mojave. The simulation was organized as follows 1) The number of measures into each stream was arbitrarily varied from 100 to 5000 through a cycle (i.e. dynamic window size). In statistic, a minimum of 100 measures is required by the huge numbers' theorem, while for building a decision tree at least 1000 records should be present be we want to validate through cross-validation or using a traditional approach such as dataset's splitting; 2) In each step, a CINCAMI/MIS message was generated with the number of indicated measures. The generated message had both deterministic and likelihood distributions (estimated) measures; 3) Each message is managed under the shape of a window which size is fitted from the message (i.e. CincamimisWindow in Fig. 3). The window was translated to a columnar-data organization obtaining a table with the defined number of measures; 4) The columnar-data organization was translated to the ARFF data format; 5) Each translated content in each step was stored as an ARFF data file (i.e. each translated window will have its file on disk). Once the window has been processed, the next window is generated and sent for its processing. Thus, the window processing time determines the next arrival. The sequence is available for reproducing it on GitHub on the test.java file into the CincamimisConversor library.

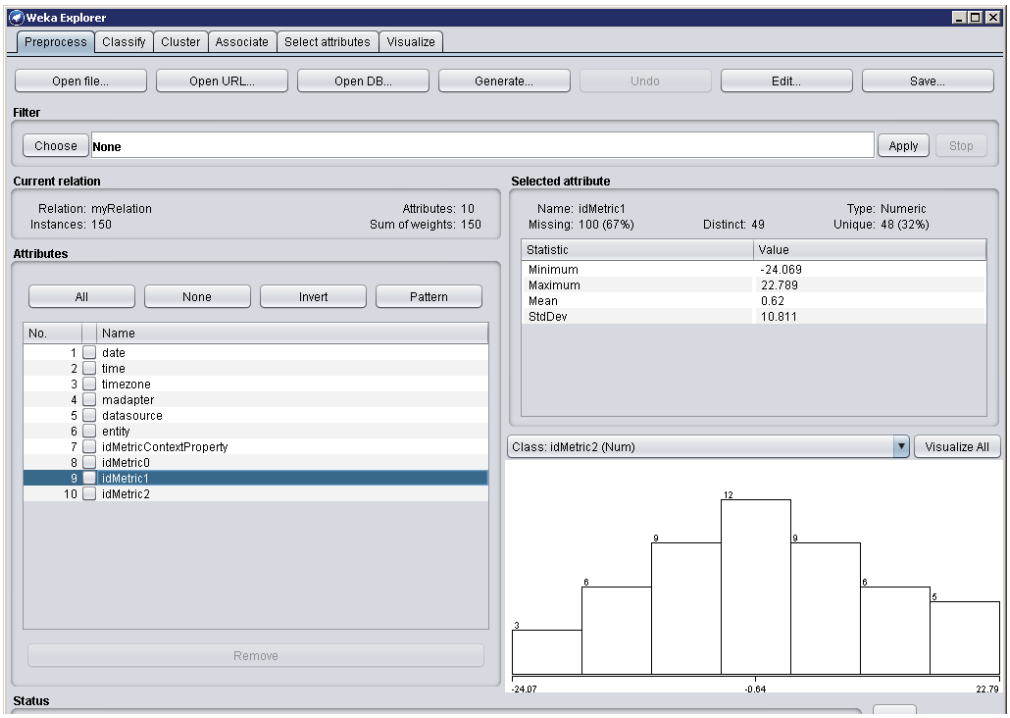

FIGURE 4. The Weka Software version 3.8 running on Windows 7 laptop, showing the metadata related to one of the translated ARFF files

One of the most important things before analyzing the simulation results is to verify the library's effectivity. That is to say, the generated ARFF file must be correctly loaded and interpreted in the Weka Software, else all the time analysis becomes abstract. Figure 4 shows the Weka Software running on a Toshiba Satellite A505 Laptop with 8GB RAM, 1.6 GHz Intel Core i7 Processor, 500GB SATA disk and 64-bit Windows 7 Home Premium (See Fig. 4), which pretends shows the interoperability related to the generated ARFF file. As it is possible to appreciate in Fig. 4, the file has ten attributes which are generated from the simulation i) *date*, *time*, and *timezone* are associated with the moment in which the measure is obtained, ii) *madapter* refers to the measurement adapter responsible for translating from the raw data format in each measure device to the measurement interchange schema (i.e. CINCAMI/MIS), iii) *datasource* indicates an identification for the measuring device, iv) entity indicates the specific entity under monitoring related to measures, v) *idMetricContextProperty* represents a measure related to the context (i.e. a way through quantifying a measurable characteristic from the context), vi) *idMetric0*, *idMetric1*, and *idMetric2* represent three attributes in general with random values. Even if it takes a look at the idMetric1attribute at the right in Fig. 4, it is possible to appreciate the distribution through the histogram, some statistical measures, and with missing values.

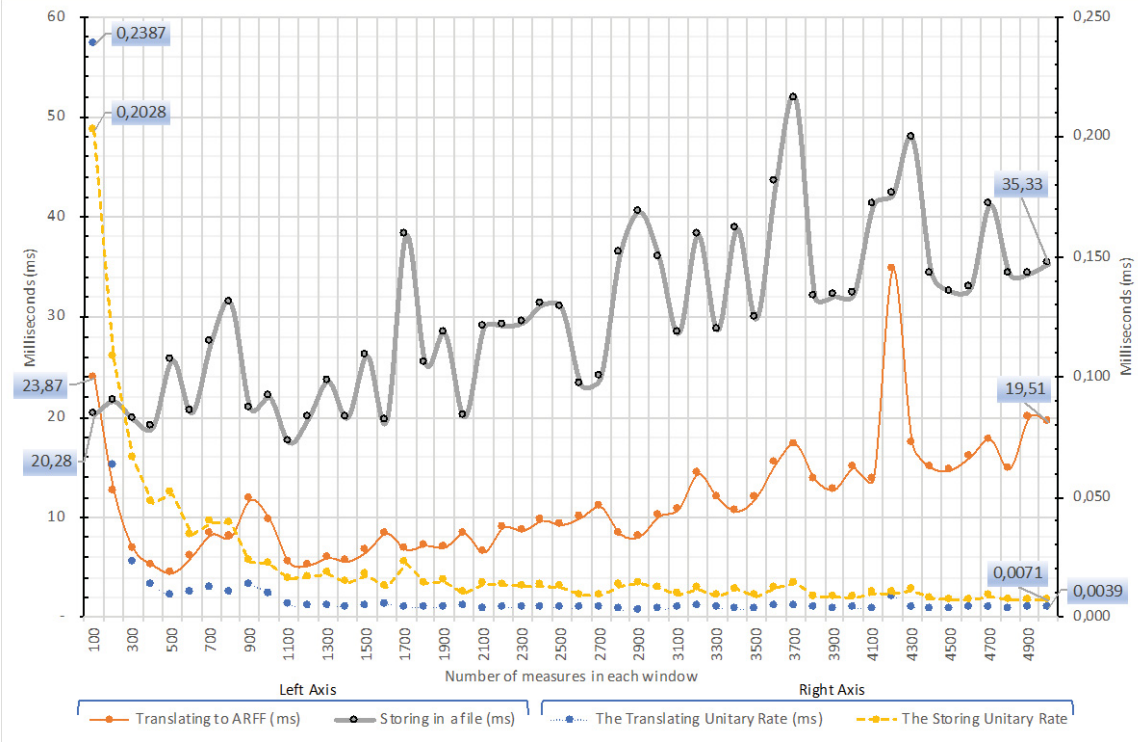

FIGURE 5. The Time Evolution related to Translating and Storing Operations in the Simulation

Figure 5 shows the time evolution (expressed in milliseconds) related to consumed time for 1) Translating from CINCAMI/MIS to ARFF data format in memory (The slim continuous line in Fig. 5); 2) Storing the ARFF data in a file on a persistent SSD device (The thick continuous line in Fig. 5); 3) The translating unitary rate (The slim dotted line in Fig. 5); and 4) The Storing unitary rate (The thick dotted line in Fig. 5). Figure 5 presents two vertical axes expressed in milliseconds due to the scale difference.

From Fig. 5 it is possible to appreciate a growing trend for translating and storing operations. On the one hand, the translating operation pass from 23.87 (ms/100 measures) to 19.51 (ms/5000 measures). It had a maximum peak when the measures reached 4200 (34.69 ms) which was related to additional consumed time for releasing memory by the garbage collector, due to data for the next window are generated immediately after the previous window has been processed. The minimum reached time was 4.40 ms with 500 measures. On the other hand, the storing operation one pass from 20.28 (ms/100 measures) to 35.33 (ms/5000 measures) with a maximum peak of 51.85 ms with 3700 measures and a minimum of 17.48 ms with 1100 measures. This is due to the scale economy and the reached parallelism through physical threads by the processing architecture.

Even when the mentioned operations have a growing trend, the respective unitary rates tend to stabilize. The unitary rates are obtained from the division between the total consumed processing time and the number of measures in the window (e.g. 34.69 ms for 4200 measures gives 0.0083 ms in the translating unitary rate). The translating unitary rate starts with 0.2387 (ms/measure) while at the end is around 0.0039 (ms/measure), with a maximum peak of 0.2387 ms and a minimum of 0.0028 ms. Something similar happens with the storing unitary rate who starts with 0.2028 (ms/measure) and ends with 0.0071 (ms/measure), which are respectively the maximum and minimum peaks. This is a typical situation to take advantage of the parallelism because measures have a specific date, time, entity, etc. which allows separating and joining them using threads without problems.

CONCLUSIONS

The ARFF data format is very useful for allowing offline analysis in software like Weka which is popular along the machine learning and data mining area. The IoT and data stream engines collect and process data continuously but there is not an associated limit in the data. Thus, an extension of the CincamimisConversor library was introduced here for allowing to export in real-time data coming heterogeneous data sources to ARFF files. Now it is possible to get a given number of measures in an ARFF data format, be it in memory or even on a file. The updated CincamimisConversor library was submitted to a discrete simulation, which shows promising results in relation to the obtained times and the natural possibility of using the parallelism in both translating and storing operations. In the end, just 19.51 ms was required for translating 5000 measures, while the associated storing operation consumed 35.33 ms, while the worst times were 34.69 ms and 52.85 ms respectively. Because the translating and storing threads are independent of the processing thread in PAbMM, the library could be used with PAbMM without engaging the global performance.

As future work, the RData's organization will be incorporated for compatibility with R. Next, other data formats (e.g. data interchange format -dif-) will be progressively incorporated in the CincamimisConversor library for fostering its extensibility and interoperability between the world of the data stream and Big Data.

ACKNOWLEDGMENTS

This research is partially supported by the projects Res.CD 278/2016 and 312/18 of the Economy School of the National University of La Pampa.

REFERENCES

[1] R. S. Samosir, H. L. Hendric, F. L. Gaol, E. Abdurachman, and B. Soewito, “Measurement Metric Proposed For Big Data Analytics System,” in *Proceedings of the 2017 International Conference on Computer Science and Artificial Intelligence - CSAI 2017*, 2017, pp. 265–269.

[2] S. K. Milligan, “Methodological foundations for the measurement of learning in learning analytics,” in *Proceedings of the 8th International Conference on Learning Analytics and Knowledge - LAK ’18*, 2018, pp. 466–470.

[3] A. Pareek, B. Khaladkar, R. Sen, B. Onat, V. Nadimpalli, M. Agarwal, and N. Keene, “Striim: A Streaming Analytics Platform for Real-time Business Decisions,” in *Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics*, 2017, pp. 4:1--4:8.

[4] G. De Francisci Morales and A. Bifet, “SAMOA: Scalable Advanced Massive Online Analysis,” *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 149–153, Jan. 2015.

[5] S. A. Shaikh and H. Kitagawa, “StreamingCube: A Unified Framework for Stream Processing and OLAP Analysis,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 2527–2530.

[6] M. Rezaei, M. A. Shirazi, and B. Karimi, “IoT-based framework for performance measurement A real-time supply chain decision alignment,” *Ind. Manag. Data Syst.*, vol. 117, no. 4, pp. 688–712, 2017.

[7] P. Rathore, A. S. Rao, S. Rajasegarar, E. Vanz, J. Gubbi, and M. Palaniswami, “Real-Time Urban Microclimate Analysis Using Internet of Things,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 500–511, Apr. 2018.

[8] M. J. Divan, "Processing architecture based on measurement metadata," in *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2016, pp. 6–15.

[9] M. J. Divan and M. de los Angeles Martin, "A new storm topology for synopsis management in the processing architecture," in *2017 XLIII Latin American Computer Conference (CLEI)*, 2017, vol. 2017-Janua, pp. 1–10.

[10] S. Roulston, U. Hansson, S. Cook, and P. McKenzie, "If you are not one of them you feel out of place: understanding divisions in a Northern Irish town," *Child. Geogr.*, vol. 15, no. 4, pp. 452–465, Jul. 2017.

[11] E. Frank, M. A. Hall, and I. H. Witten, "The WEKA workbench," in *Data Mining*, Elsevier, 2017, pp. 553–571.

[12] M. Diván, M.; Sánchez Reynoso, "Fostering the Interoperability of the Measurement and Evaluation Project Definitions in PAbMM," in *2018 7th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2018*, 2018, pp. 228–234.

[13] M. Diván and M. de los Ángeles Martín, "Towards a Consistent Measurement Stream Processing from Heterogeneous Data Sources," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 6, pp. 3164–3175, Dec. 2017.

[14] M. Diván and M. Sánchez Reynoso, "A Library for Articulating the Measurement Streams with Columnar Data," *Int. J. Eng. Technol.*, vol. 7, no. 4, pp. 234–241, 2018.

[15] K.-W. Hong and D.-H. Park, "SLICE-based Trustworthiness Analysis system," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018, pp. 1389–1390.

[16] A. A. Chaudhari and P. Mulay, "SCSI: Real-Time Data Analysis with Cassandra and Spark," in *Big Data Processing Using Spark in Cloud*, M. Mittal, V. E. Balas, L. M. Goyal, and R. Kumar, Eds. Springer Singapore, 2019, pp. 237–264.

[17] F. Charte, A. J. Rivera, D. Charte, M. J. del Jesus, and F. Herrera, "Tips, guidelines and tools for managing multi-label datasets: The mldr.datasets R package and the Cometa data repository," *Neurocomputing*, vol. 289, pp. 68–85, May 2018.

[18] S. Ahmed, M. Usman, J. Ferzund, M. Atif, A. Rehman, and A. Mehmood, "Modern Data Formats for Big Bioinformatics Data Analytics," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 4, 2017.

[19] A. Bifet, S. Maniu, J. Qian, G. Tian, C. He, and W. Fan, "StreamDM: Advanced Data Mining in Spark Streaming," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, pp. 1608–1611.

[20] L. Olsina, F. Papa, and H. Molina, "How to Measure and Evaluate Web Applications in a Consistent Way," in *Web Engineering: Modelling and Implementing Web Applications*, G. Rossi, O. Pastor, D. Schwabe, and L. Olsina, Eds. London: Springer London, 2008, pp. 385–420.