An LLVM-Inspired Framework for Unified Processing of Multimodal Time-Series Data

Lee B. Hinkle
leebhinkle@txstate.edu
Texas State University
San Marcos, Texas, USA

Vangelis Metsis
vmetsis@txstate.edu
Texas State University
San Marcos, Texas, USA

ABSTRACT

Groups of sensors collecting time-series data in a variety of modalities are widely used for monitoring humans, environments, and equipment. Datasets with multimodal sensor data pose several challenges not present in many image or language datasets; most notably, there are few de-facto standards on how the data should be organized and packaged. In this work, we present a framework inspired by the LLVM Compiler architecture that streamlines sensor data processing for machine learning applications. Specifically, we define standardized, intermediate representations that can be easily transformed for input to data preprocessing and model training steps. By standardizing and preserving time and subject information, our method supports robust label verification and multiple means of subject-independent cross-validation. We demonstrate the validity of our framework using seven different datasets, all containing time-series data and representing a variety of sensors, modalities, domains, and collection environments.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Applied computing** → **Consumer health**.

KEYWORDS

physiological signals, time-series, data pipeline, sensor data

ACM Reference Format:
Lee B. Hinkle and Vangelis Metsis. 2023. An LLVM-Inspired Framework for Unified Processing of Multimodal Time-Series Data. In *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '23), July 05–07, 2023, Corfu, Greece.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3594806.3594812

1 INTRODUCTION

Multimodal time-series data collected from sensors is ubiquitous, however, time-series datasets are not as readily available as the more popular image and natural language processing datasets. In addition, there are few datasets that serve as de facto standards for the publishing format in the same manner that the CIFAR-10 dataset [8] does for image datasets. One notable exception is the UCR Time-Series archive [2] which offers 128 different datasets in a common format, however, these datasets are not as complex as the ones evaluated here. The lack of standard formats results in much time spent building "converters", model evaluations using only one or two datasets, and potential misallocation of data between the testing and training sets. It is critical to preserve the subject data to ensure that no subject's data is present in the training *and* test sets. Typically this occurs when the data is partitioned into fixed time segments known as sliding windows, and the entire batch of windows is further split into training, validation, and test sets using standard libraries without consideration of the contributing subject. In this case, data from a single subject may be present in the training, validation, and test groups resulting in higher test accuracies than can be expected for a previously unseen subject [12][6]. The primary contribution of this work is the introduction of a framework with standardized intermediate representations that places the bulk of the *per dataset* work at the front of the pipeline, allowing for significant code reuse of validated transformations on the subsequent intermediate representations, and provides control over the final output for rich model experimentation and evaluation. The use of intermediate representations preceded by dataset-specific transformations is inspired by the LLVM compiler architecture [9] (https://llvm.org/). In this architecture, multiple programming languages, such as C and Fortran, are converted to intermediate representations, which are processed using common optimizations. The optimized code can then be output for multiple Instruction Set Architectures (ISAs) such as x86 and ARM. Our time-series data processing equivalent of this architecture is shown in Figure 1. By preserving the subject data during the transformations, the issue of data leakage is avoided and validation techniques from hold-one-subject-out to group-based cross-validation can be performed. This work has been validated using seven different datasets, many shared transformations, and three programs that utilize the intermediate or final representations of the data. The source code for this work is available at https://github.com/imics-lab/load_data_time_series.

2 INTERMEDIATE DATA REPRESENTATIONS

The following sections describe the processing of the datasets and the format of the intermediate representations. For *each dataset* the transformation into the first intermediate representation, IR1, is described underscoring the fact that the datasets are not published in a standard format. The second and third transformations are shared among all datasets validating the benefit of the defined intermediate representations.

PETRA '23, July 05–07, 2023, Corfu, Greece

ACM ISBN 979-8-4007-0069-9/23/07…$15.00
https://doi.org/10.1145/3594806.3594812

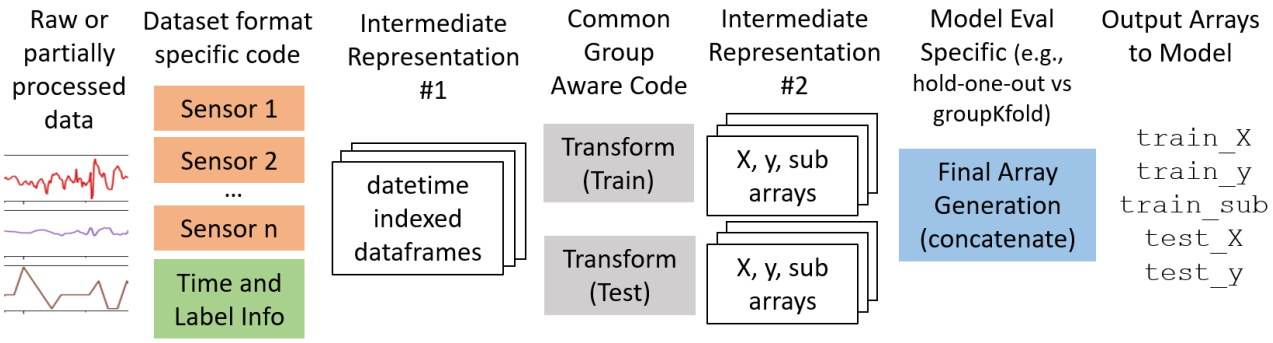

Figure 1: The LLVM Inspired Framework. Dataset-specific transformation to the first Intermediate Representation (IR1) is followed by standard transformations that can be applied to all datasets, and concludes with a final transform to generate the output arrays in the desired form to support the training and evaluation of the model.

2.1 Subject, Label, and Raw Sensor Data to IR1

The first defined intermediate representation, IR1, is a Pandas dataframe [13] indexed by UTC datetime. Each recorded session is used to generate a single IR1 with contiguous time indexing organized with each row representing a specific sample time and columns of signal data (channels), label(s), and the subject number. The datetime indexing enables the merging of data from multiple sensors with different sampling rates and start times. Having the time associated with each sample also facilitates the labeling of activities. The individual sensor columns may be typed according to the sensor's capabilities (an accelerometer with 8-bit resolution need not be left at the float64 default of many libraries). Columns are added for derived data, such as the magnitude of acceleration plus the subject number and label. Datasets often have the subject numbers and labels embedded in the filename or directory. E.g., sub1_session1.csv may be located in a "walking" directory indicating the data is from subject number 1 and the label is walking. A key benefit of using a higher-level data structure is that rich multi-character string labels can be stored in a categorical-typed column without consuming large amounts of memory. In our work, IR1 is implemented as a Pandas dataframe; however, we believe the format is compatible with a MATLAB table or an R-dataframe. A portion of an IR1 dataframe is shown in Figure 2.

The attributes for the seven datasets used are summarized in Table 1. For brevity, only significant elements of the transformation from the raw data to the IR1 dataframe will be described.

The MobiAct dataset [15] includes accelerometer and gyroscope data collected using a hip-worn smartphone with 30 subjects performing six different activities, such as standing, walking, and going up/downstairs in a scripted fashion. The label and subject number columns are populated in IR1 based on the directory and filename, respectively. The provided UTC time is used for indexing. Sample timing varies on smartphones versus the more regular sampling of dedicated devices. The datetime index allows for resampling to consistent time deltas, however, the variation has minimal effect on model accuracies [6].

UCI-HAR [1] is the oldest and most heavily pre-processed of the datasets used, so not all intermediate representations are required (resulting in some loss of configurability such as alternate sliding window sizes). After filtering the data are divided into individual sliding windows of 2.56 seconds with 50% overlap. Importantly the data is pre-partitioned *by subject* with 70% of the subjects placed into the training group and 30% placed into the test group. Similarly, the UniMIB SHAR dataset [12] ADL data are provided in MATLAB table format with windows of 151 samples centered on the peak acceleration. Due to the extensive pre-processing, the IR1 format was also not required for this dataset; many of the IR2 transformations were applied directly.

The TWristAR dataset [5] contains data from an Empatica E4 wristband [3] that contains four sensors (EDA, Temp, Accel, and PPG) having sample rates of 4Hz, 4Hz, 32Hz, and 64Hz, respectively. Each sensor's data file includes the start time and the sampling frequency, which was used to reconstruct the UTC time for the row index. The datetime indexing of IR1 facilitates the merge of all four sensor's data into a single dataframe when resampled to a common 32Hz frequency.

The "Ankle-Hip-Wrist" dataset [10] includes data from three devices with two sampling rates. Eight subjects performed 17 activities-of-daily-living (ADL), such as handwriting, face washing, eating, etc. In order to put all sensor data into a single dataframe the data from the ankle, hip, and wrist devices were concatenated by column and downsampled to 50Hz.

The PSG-Audio dataset [7] is the largest used in this work. Many elements, such as the categorical representation in IR1 were implemented due to data structure sizes that were not as cumbersome for the other datasets. Multimodal data were collected in a hospital on over 200 patients during sleep apnea evaluations of three to seven hours each. A total of 20 channels are present including Electroencephalogram (EEG), Electrooculogram (EOG - eye movement), Leg Movement, Electrocardiogram (ECG), a contact microphone, nasal airflow monitors, respiratory belts, pulse oximeter, and high-fidelity audio. The recording frequencies vary from 1Hz to 48kHz; for this work, the 200Hz signals were downsampled and merged with the 100Hz signals in IR1. The high-frequency audio channels were not used due to their large size (processing a single hour of raw audio data consumed 24GB of RAM). The sensor data is provided in European Data Format (EDF), which is a standard for medical time-series data. The labels were extracted from the provided .rml file for each

datetime	accel_x	accel_y	accel_z	accel_ttl	bvp	eda	p_temp	label	sub
2019-11-24 18:50:07.500000000	-0.31250	-1.1250	0.093750	0.171354	-5.47	8.540753	32.0	4	1.0
2019-11-24 18:50:07.531249920	-0.34375	-1.1875	0.218750	0.255457	-0.74	8.540914	32.0	4	1.0
2019-11-24 18:50:07.562500096	-0.37500	-1.2500	0.296875	0.338380	9.74	8.541073	32.0	4	1.0

Figure 2: Example of the IR1 data format as displayed by the Panda head method. This data is from the scripted portion of the TWristAR dataset.

Table 1: Summary of seven datasets that were used to develop and validate the framework. All are time-series with body-worn or attached sensors.

Name	Domain	Sensor Type(s)	# Sub	Year
UCI HAR [1]	HAR	Smartphone	30	2012
MobiAct [15]	HAR	Smartphone	50	2016
UniMiB SHAR [12]	ADL + Fall	Smartphone	30	2017
Sussex-Huawei Locomotion [4]	HAR(Transport)	4XSmartphone + Video	3	2017
Ankle-Hip-Wrist [10]	ADL	Activity Monitor	8	2021
PSG-Audio [7]	Polysomnography	Audio, ECG, EEG,+	192	2022
TWristAR [5]	HAR	Empatica E4 Wristband	3	2022

Table 2: Typical IR2 transforms that are applied. Note that the allowed transforms differ between the train and test groups. Further details and code are available on our public GitHub https://github.com/imics-lab/load_data_time_series

get_ir2_from_ir1	Train&Test	Slice into sliding windows of fixed time-steps
drop_ir2_multi_label	Train Only	Discard mixed labeled windows (for large datasets)
all_ir2_labels_to_mode	Train&Test	Assign label as the mode of labels within a sliding window
collapse_ir2_timesteps	Train&Test	Validate sub, labels match, collapse 3rd dim

patient and placed into individual Respiratory, Neurological, Limb, Nasal, Cardiac, and SpO2 categorical IR1 columns with each label category having multiple event types such as Cardiac: "Normal", "Bradycardia", "LongRR', etc.

2.2 IR1 dataframe to IR2 n-dimensional arrays

The conversion from an IR1 dataframe to IR2 arrays is primarily the application of a sliding window of a given size and step. This results in an additional dimension - each row now represents a window "instance" containing many time steps. Typically these windows are two to five seconds in duration, and there are many windows with a recording "session", see Figure 3. From a starting IR1 shape of (session samples, {channels, labels, sub}), the three IR2 shapes will be X (instances, time steps, channels), y (instances, labels), sub (instances, subject number). We implement IR2 as three NumPy arrays to allow for faster mathematical operations with each having a single data type. "X" contains the channels with a datatype reflective of the underlying sensor precision (rarely are the default float64 values needed). "y" arrays contain the integer encoded labels; descriptive strings consume too much memory with larger datasets. The "sub" arrays contain the subject number as an int8 or int16 (for datasets with subject numbers > 255). Additional IR2 transformations are shown in Table 2. Care must be taken to ensure that no label-aware transforms are applied to the test set data. There is certainly an argument that a mixed-type data structure such as a Pandas dataframe should be input directly into the model, with the model itself handling each datatype directly. A MATLAB table [11] has such a structure, but we have found this level of abstraction difficult when dealing with the additional complexity of sliding windows and the need to preserve subject independence during the training and evaluation.

2.3 Final Output Representation: Train/Valid/Test Allocation

The final representation is primarily a concatenation of the transformed IR2 arrays. The output format varies depending on the desired model evaluation. For defined-subject (each subject is allocated manually, often to balance gender, age, height, or other attributes) the train and test IR2s are concatenated based on a subject allocation dictionary, see 3. This enables high repeatability but cross-validation is often preferable. In order to support hold-one-subject-out or group-K-fold cross validation all train subjects may be placed into the train_* arrays and the sub-array input into standard libraries such as the Scikit-Learn GroupKFold [14] to generate the multiple folds of train and valid arrays.

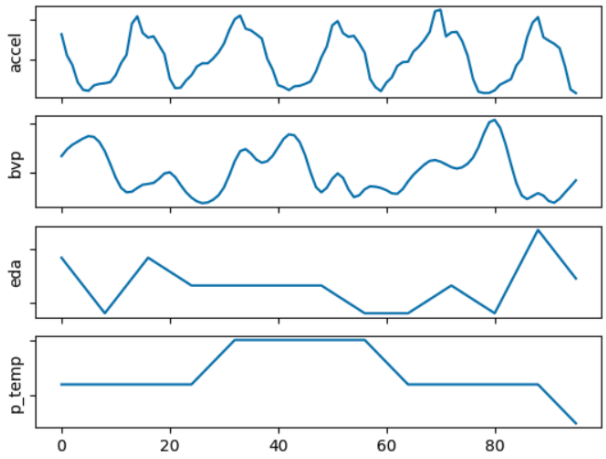

Figure 3: The multi-modal signals in a single IR2 sliding window for a TWristAR walking segment. Each three-second window at 32 Hz contains 96 samples.

Table 3: The final numpy array dimensions for the TWristAR dataset using the default arguments. The selected channels are accel_ttl, bvp, eda, p_temp

array	shape	data type
x_train	(2077, 96, 4)	float32
y_train	(2077, 1)	int8
x_test	(1091, 96, 4)	float32
y_test	(1091, 1)	int8

Table 4: The test accuracies reported by the "throwdown" which performs an A/B comparison between two models using all seven datasets. Model A is a Long-Short-Term Memory (LSTM), and Model B has two CNN layers followed by a Global Average Pooling layer. These results are presented as examples of how multiple datasets can be run easily from this framework. Neither model is fully tuned for every dataset.

Dataset	LSTM	CNN+GAP
UCI HAR	37.8%	84.0%
MobiAct	89.6%	97.7%
UniMiB SHAR	92.0%	89.5%
Sussex-Huawei Locomotion	20.8%	24.4%
Ankle-Hip-Wrist	17.0%	63.9%
PSG-Audio	81.2%	80.7%
TWristAR	66.5%	66.4%

3 RESULTS AND CONCLUSION

In order to confirm the portability of the intermediate representations three programs were created and are available on our public repository at https://github.com/imics-lab/load_data_time_series. The *ts_demo* program is a minimal example Jupyter Notebook data loader which evaluates a single 1D-CNN model using any one of the currently supported datasets. *ts_visualize* uses the IR1 representations to examine and visualize the dataset attributes using the IR1 format. To demonstrate how this work can broadly enable the inclusion of multiple datasets using a subject-independent evaluation, *ts_throwdown* performs an A/B comparison of two models by running a train/test pass using the seven supported datasets. The results are summarized in Table 4.

We have presented an LLVM-inspired architecture with defined intermediate representations of time-series data to facilitate rich model evaluation. We have validated our framework using seven datasets of varied formats which were converted into a standard representation enabling subsequent common transformations and multiple final output configurations. Subject data is preserved throughout ensuring that there is no test data leakage into the training set. Accuracies for LSTM and CNN models running each of the seven datasets from multiple domains are shown.

REFERENCES

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra Perez, and Jorge Luis Reyes Ortiz. 2013. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*. 437–442.

[2] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. 2018. The UCR Time Series Classification Archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[3] Empatica. 2020. E4 Wristband User's Manual, Rev. 2.0. https://empatica.app.box.com/v/E4-User-Manual. Accessed: 2022-06-09.

[4] Hristijan Gjoreski, Mathias Ciliberto, Francisco Javier Ordoñez Morales, Daniel Roggen, Sami Mekki, and Stefan Valentin. 2017. A versatile annotated dataset for multimodal locomotion analytics with mobile devices. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Sys.* 1–2.

[5] Lee B. Hinkle, Gentry Atkinson, and Vangelis Metsis. 2022. *TWristAR - wristband activity recognition.* https://doi.org/10.5281/zenodo.5911808

[6] Lee B Hinkle and Vangelis Metsis. 2021. Model Evaluation Approaches for Human Activity Recognition from Time-Series Data. In *International Conference on Artificial Intelligence in Medicine*. Springer, 209–215.

[7] Georgia Korompili, Anastasia Amfilochiou, Lampros Kokkalas, Stelios A Mitilineos, Nicolas-Alexander Tatlas, Marios Kouvaras, Emmanouil Kastanakis, Chrysoula Maniou, and Stelios M Potirakis. 2021. PSG-Audio, a scored polysomnography dataset with simultaneous audio recordings for sleep apnea studies. *Scientific Data* 8, 1 (2021), 197.

[8] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[9] Chris Lattner and Vikram Adve. 2004. LLVM: A compilation framework for lifelong program analysis & transformation. In *International symposium on code generation and optimization, 2004. CGO 2004.* IEEE, 75–86.

[10] Maurizio Leotta, Andrea Fasciglione, and Alessandro Verri. 2021. Daily Living Activity Recognition Using Wearable Devices: A Features-Rich Dataset and a Novel Approach. In *Pattern Recognition. ICPR International Workshops and Challenges*, Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani (Eds.). Springer International Publishing, Cham, 171–187.

[11] MathWorks. 2022. trainNetwork. https://www.mathworks.com/help/deeplearning/ref/trainnetwork.html. Accessed: 2022-06-02.

[12] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. 2017. UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones. *Applied Sciences* 7, 10 (2017). https://doi.org/10.3390/app7101101

[13] The pandas development team. 2020. *pandas-dev/pandas: Pandas.* https://doi.org/10.5281/zenodo.3509134

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[15] George Vavoulas, Charikleia Chatzaki, Thodoris Malliotakis, Matthew Pediaditis, and Manolis Tsiknakis. 2016. The mobiact dataset: Recognition of activities of daily living using smartphones. In *International Conference on Information and Communication Technologies for Ageing Well and e-Health*, Vol. 2. SciTePress, 143–151.