ASSISTED LABELING VISUALIZER (ALVI): A SEMI-AUTOMATIC LABELING SYSTEM FOR TIME-SERIES DATA

Lee B. Hinkle, Tristan Pedro, Tyler Lynn, Gentry Atkinson, and Vangelis Metsis

Department of Computer Science
Texas State University
San Marcos, TX 78866, USA

ABSTRACT

Machine learning applications can significantly benefit from large amounts of labeled data, although the task of labeling data is notoriously challenging and time-consuming. This is particularly evident in domains involving human subjects, where labeling time-series signals often necessitates trained professionals. In this work, we introduce the Assisted Labeling Visualizer (ALVI), a system that simplifies the process of labeling data by offering an interactive user interface that visualizes synchronized video, feature-map representations, and raw time-series signals. ALVI also leverages deep learning and self-supervised learning techniques to facilitate the semi-automatic labeling of large amounts of unlabeled data. We demonstrate the capabilities of ALVI on a human activity recognition dataset to showcase its potential for enhancing the labeling process of time-series sensor data.

Index Terms— Time series, sensor data, semi-automatic labeling, visualization.

1. INTRODUCTION

Data labeling is a crucial step in the process of machine learning applications. It involves assigning relevant and accurate tags to data that are used to train models. Labeling time-series data is particularly important for applications that involve continuous monitoring and tracking of data, such as in healthcare, manufacturing, and environmental monitoring. Time-series sensor data can provide valuable insights into changes in a system, environment, or individual's behavior over time. By accurately labeling such data, machine learning models can identify patterns and predict future outcomes.

The utilization of time-series sensor data and labeling can significantly enhance accessibility for people with disabilities. For instance, by deploying sensors throughout a museum and labeling the captured data, machine learning models can be trained to recognize patterns in visitor behavior [1], such as popular exhibits and frequently taken routes. This information can then be utilized to provide more accessible experiences for people with disabilities [2][3].

Manual labeling of time-series sensor data can be a daunting and challenging task, particularly when dealing with massive and complex datasets. One of the main difficulties is that humans are not naturally skilled at reading and interpreting raw time-series sensor data. Even with the use of video data to assist in the labeling process, the manual labeling of time-series data can still be tedious, time-consuming, and error-prone. The process of manual labeling requires significant human resources and can lead to inconsistencies across different human labelers. Semi-automatic labeling can make use of machine learning algorithms to pre-label the data and allow human labelers to correct any errors or inconsistencies in the pre-labeled data. This approach can significantly reduce the time and effort required for manual labeling and improve the accuracy and consistency of the labeled data.

In this work, we introduce a browser-based software framework[1] for labeling time-series sensor data that incorporates state-of-the-art visualization and machine learning techniques, enabling efficient and precise semi-automatic labeling of such data. We employ interactive visualizations of raw time-series data, as well as features extracted through contrastive self-supervised learning methods. Furthermore, we incorporate a label correction process to detect and correct any potential errors in the automatically assigned labels. The human remains in the loop to ensure the quality of the assigned labels but with a significantly reduced workload. Our system is compatible with any Jupyter-capable machine, thus enabling local execution to maintain data confidentiality in cases where the data is sensitive.

2. RELATED WORK

There have been many general systems developed to label data, such as text, images, and audio. One example is the VGG Image Annotator [4]. Crowd-sourcing platforms like Amazon Mechanical Turk [5], Apen [6], and Labelbox [7] are commonly used for general data labeling. These platforms provide a cost-effective and scalable solution for data labeling, but they have certain drawbacks, such as the difficulty in

This work is supported in part by the NSF awards 1757893 and 2149950.

[1] https://github.com/imics-lab/time-series-label-assist

ensuring the quality of labels and the potential for low-quality labels due to the lack of expertise of the workers. General data labeling systems can have issues with ambiguity, making it difficult for labelers to accurately label data.

Labeling time-series sensor data presents a unique set of challenges when compared to labeling images, videos, or text. Examples of labeling systems used for time-series sensor data include Visplore [8] and Label Studio [9]. These systems use various techniques such as manual labeling, rule-based labeling, and semi-automatic labeling to overcome these challenges. However, the labeling of time-series data is often more tedious and time-consuming than other types of data.

Semi-automatic labeling is a method that combines human expertise with machine learning algorithms to improve the efficiency and accuracy of data labeling. Examples of literature that use semi-automatic labeling include VAST [10] and SALT [11]. Semi-automatic labeling has been shown to significantly reduce the time and effort required for manual labeling while maintaining high labeling accuracy. However, most available solutions so far only apply to image/video and textual data.

Data visualization tools, such as t-SNE [12] and UMAP [13], can assist in the labeling of time-series sensor data by providing visual representations of the data that can aid in identifying patterns and relationships. UMAP has been shown to be particularly useful for dimensionality reduction and visualization of high-dimensional time-series sensor data.

There are several techniques available for automatically correcting mislabeled instances of time series data. These methods generally employ deep learning systems that can recognize the correct class of mislabeled instances, despite the difficulty that machine learning models may encounter when training on noisy labels [14]. By comparing the output of a trained convolutional neural network (CNN) to the assigned labels in a dataset, it is possible to identify which instances in a sensor dataset are most likely to be mislabeled [15]. Another approach for identifying mislabeled data is to compare instances to their nearest neighbors and determine the most probable correct label either by comparison to neighbors [16] or statistical inference [17].

3. METHODOLOGY

The labeling framework proposed in this study has been realized as a web interface that is compatible with any browser by leveraging the capabilities of Jupyter Notebooks [18] and Plotly Dash [19]. The web-based tool is self-contained and does not rely on cloud hosting, thus enabling local deployment on any Jupyter-enabled machine or Google Colab. As a result, the framework allows for secure and privacy-preserving labeling of sensitive data, without the need to upload the data to a remote server for annotation.

The data labeling process comprises the following steps:

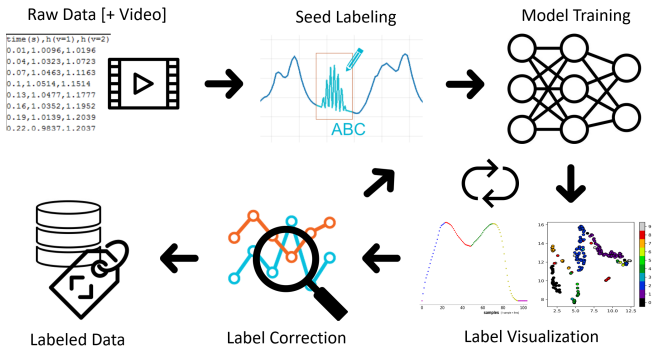


Fig. 1: ALVI semi-automatic labeling process workflow.

1. Raw time series data, and if available video, are loaded from a file.
2. Data and corresponding video are visualized as time-series plots using Plotly.
3. A small amount of labeled data are provided or manually labeled by the user.
4. A deep learning model is trained based on the initially labeled data.
5. A larger amount of data is automatically labeled using the trained model.
6. The automatically assigned labels are analyzed, and parts of the data are manually reviewed.

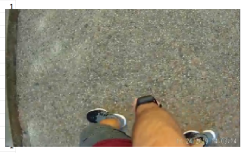
Figure 1 visually depicts the above workflow. In the following subsections, we elaborate on the methods and tools used in each one of the steps above.

3.1. Data Loading

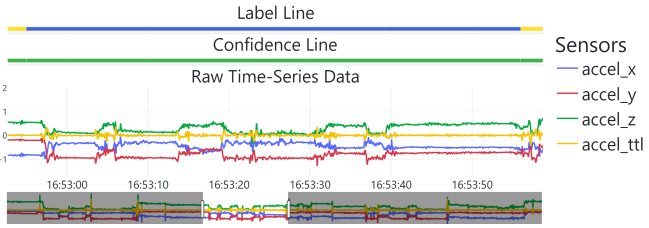
The current version of the software supports data files in CSV text format. It is assumed that files are structured as: `<timestamp>,<ch1>,...,<chn>,<label>,<sub>` where `timestamp` is the time stamp of the sensor measurement, `ch1,...,chn` represent the different data channels. For example, an accelerometer may have three channels for the three axes of acceleration, XYZ. The `label` column represents a label assigned to each time step of data. Initially, that value can be set to `''undefined''` if the data is unlabeled. The `sub` column represents the subject from which the data were collected and can be a number or a string. When data from multiple subjects are aggregated together, it is often necessary to maintain subject independence in machine learning experiments. Thus, maintaining the subject label is desirable. Each row of the file is a sensor measurement.

Along with the raw time-series data, the user can specify a video file, if available, to inform the labeling process (e.g. Fig. 2a). The loaded video file can be synchronized with a particular section of the raw data by specifying an offset with respect to the data timestamps.

	A	B	C	D	E	F	G
1	datetime	accel_x	accel_y	accel_z	accel_ttl	label	sub
2	01:54.6	-0.23438	-0.46875	0.171875	-0.44846	Walking	1
3	01:54.7	-0.26563	-0.48438	0.171875	-0.42145	Walking	1
4	01:54.7	-0.32813	-0.64063	0.140625	-0.26662	Walking	
5	01:54.7	-0.34375	-0.75	0.125	-0.16556	Walking	
6	01:54.7	-0.3125	-0.95313	0.140625	0.012857	Walking	
7	01:54.8	-0.26563	-0.96875	0.1875	0.021856	Walking	
8	01:54.8	-0.26563	-1.03125	0.21875	0.087145	Walking	
9	01:54.8	-0.29688	-1.17188	0.265625	0.237733	Walking	
10	01:54.9	-0.3125	-1.1875	0.265625	0.256332	Walking	
11	01:54.9	-0.32813	-1.3125	0.171875	0.363768	Walking	
12	01:54.9	-0.32813	-1.40625	0.15625	0.452453	Walking	
13	01:55.0	-0.35938	-1.5	0.25	0.562578	Walking	
14	01:55.0	-0.3125	-1.17188	0.1875	0.227234	Walking	
15	01:55.0	-0.29688	-1.20313	0.140625	0.247165	Walking	1



(a) A snapshot of the data file and the video captured during data collection from an arm-mounted camera.



(b) An example visualization used for seed labeling. The raw signals, color-coded assigned label line, and confidence lines are visible.

Fig. 2: Data loading and seed labeling stages.

3.2. Seed Labeling

In order to facilitate the automatic labeling of the remaining data, it is necessary to have a small amount of initial labeled data. These initial labeled data can either be directly loaded into the system if they have already been pre-labeled outside of the system or manually labeled within the system by utilizing the visualization capabilities and the video-sync feature.

The seed labeling process can use advanced visualization features and interactive graphs provided by Plotly (see Fig. 2b). For instance, the user is afforded the ability to zoom in on the signal, select specific sections, and assign labels by specifying the starting and ending points of a segment, the label, and their confidence level regarding the assigned label (i.e., low, medium, or high). Confidence levels can be utilized in model training. For example, segments of low confidence can be excluded from the training set.

3.3. Model Training and Automatic Labeling

The process of training a deep learning model involves utilizing the manually pre-labeled portion of the data. During this stage, the continuous signal is segmented into fixed-size windows, and each window is assigned a single label. This technique is a commonly employed approach for training models on time-series data. The training set can comprise either overlapping or non-overlapping segments. In the event that a window spans two or more labels, the user can elect to assign the majority of time steps as the primary label for that segment or exclude the segment entirely from the training set.

The deep learning model can be based on any neural network architecture that supports time-series classification. The current version offers the user a choice between a convolutional neural network (CNN)-based or a long short-term memory (LSTM)-based architecture. However, the module can be replaced with other architectures, such as a transformer-based architecture, if necessary.

Upon acquiring a trained model, a larger quantity of data can be automatically labeled by having the model predict the correct label for each segment of the new data. Although this process saves time and effort from manual labor, it is anticipated to produce some incorrect predictions. To rectify these inaccuracies and establish a reliable ground truth, we utilize a complex approach for incorrect label detection and correction, which is detailed in the following subsections. This approach requires human intervention; however, it is substantially less labor-intensive than fully manual labeling.

3.4. Label Correction

By adopting label correction techniques, the Assisted Labeling Visualizer (ALVI) can focus a human reviewer on the portions of data most likely to be mislabeled by the Automatic Labeling process. ALVI implements a label-cleaning process based on K-Nearest Neighbors (KNN). However, such data-centric approaches are highly sensitive to the clusterability of the features being processed [17]. To account for this sensitivity, we incorporate deep feature learning using a convolutional feature extractor. The output penultimate layer of a CNN is used to produce a highly clusterable feature space.

A KNN classifier is fit to the extracted feature space and used to predict labels for each instance of data. By comparing this classification to the output of the trained CNN we identify instances that have been classified with one label but lay in a region of the feature space near instances that do not share their label. The dataset is sorted by the product of the assigned one-hot label from the CNN and the categorical label output by KNN. This allows us to identify a portion of the dataset that most needs review.

3.5. Data Visualization

Incorporating a human in the loop during the label correction process is crucial, necessitating informative visualizations to aid in decision-making. Our proposed solution involves the use of automatically labeled data that is visualized and color-coded. We employ two types of visualizations, raw signals and Uniform Manifold Approximation and Projection (UMAP) plots [13]. In the UMAP plot, each point corresponds to a signal segment. Multi-channel raw signal segments are mapped to a low dimensional vector through a model trained using self-supervised contrastive learning as presented in [20]. Such methods do not rely on the original data labels and, thus, are not affected by label noise. The low-dimensional vector is then further reduced to two dimensions for 2D visualization by UMAP.

By clicking on a point in the UMAP plot, the point is highlighted, along with the closest neighbor, irrespective of

(a) Visualization of the raw times series data.

(b) UMAP plot.

(c) Nearest neighbors.

Fig. 3: The data visualization process utilized for label review and correction. When the user clicks on a suspicious data point on the UMAP plot, the nearest neighbor overall as well as the nearest neighbor of the same label, are highlighted. The top graph shows the locations of those segments in the original signal, whereas the lower right graph shows a zoomed-in neighbor.

its class, and the closest neighbor of the same class as the one assigned to the point. Additionally, the corresponding signal segment in the raw time series data is also highlighted. If available, the synchronized video can display the corresponding position. Figure 3 shows an example of this process. This process can help the users decide if the automatically assigned label is correct or not. The combination of these visualizations enhances the effectiveness of the label correction process, ultimately leading to higher-quality labeled datasets.

4. RESULTS

In order to evaluate the tool we used TWristAR [21], a human activity recognition (HAR) dataset that contains multimodal data collected with an Empatica E4 Wristband. Three subjects performed scripted activities which were structured for easier labeling and balanced classes. For this work, the scripted activities were treated as labeled. Two subjects performed unscripted free-form walks that included a period of sitting as well as walking on flat ground and up/downstairs. A full video record is included. [22] provides additional dataset details and describes prior manual labeling work.

For the TWristAR dataset, we found that manually labeling a single subject's 11-minute free-form walk took 34 minutes. This is consistent with our experience that labeling data with frequent activity changes takes longer than real-time due to the need to stop and check the transitions. The manual labeling accuracy was 91% versus the ground truth based on prior labeling by multiple people with additional review. With the ALVI tool using the predictions of a model trained on the scripted sequences the labeling time was reduced to 9 minutes and the accuracy increased to 96%.

5. CONCLUSION

In this work, we introduced ALVI, a browser-based software framework that enables efficient and precise semi-automatic labeling of time-series sensor data. We have demonstrated the labeling time reduction and accuracy benefits when labeling portions of a HAR dataset. Future work includes the evaluation of datasets in alternate domains and the addition of more informative visualizations and feature representations.

6. REFERENCES

[1] Seonghyeon Kim, Seokwoo Kang, Kwang Ryel Ryu, and Giltae Song, "Real-time occupancy prediction in a large exhibition hall using deep learning approach," *Energy and Buildings*, vol. 199, pp. 216–222, 2019.

[2] Sameh Triki and Chihab Hanachi, "A self-adaptive system for improving autonomy and public spaces accessibility for elderly," in *Agent and Multi-Agent Systems: Technology and Applications: 11th KES International Conference, KES-AMSTA 2017 Vilamoura, Algarve, Portugal, June 2017 Proceedings 11*. Springer, 2017, pp. 53–66.

[3] Kotaro Hara and Jon E. Froehlich, "Characterizing and visualizing physical world accessibility at scale using crowdsourcing, computer vision, and machine learning," *SIGACCESS Access. Comput.*, , no. 113, pp. 13–21, nov 2015.

[4] Abhishek Dutta and Andrew Zisserman, "The via annotation software for images, audio and video," in *Proceedings of the 27th ACM International Conference on Multimedia*, New York, NY, USA, 2019, MM '19, p. 2276–2279, Association for Computing Machinery.

[5] Amazon, "Amazon Mechanical Turk (MTurk)," `https://www.mturk.com/`, 2023, [Online; accessed 8-March-2023].

[6] Apen, "Apen," `https://appen.com/platform-5/`, 2023, [Online; accessed 8-March-2023].

[7] Labelbox, "The Labelbox Platform," `https://labelbox.com/product/platform/`, 2023, [Online; accessed 8-March-2023].

[8] "Visplore," `https://visplore.com/`, 2023, [Online; accessed 8-March-2023].

[9] "Label Studio," `https://labelstud.io/`, 2023, [Online; accessed 8-March-2023].

[10] Daniel R Berger, H Sebastian Seung, and Jeff W Lichtman, "Vast (volume annotation and segmentation tool): efficient manual and semi-automatic labeling of large 3d image stacks," *Frontiers in neural circuits*, vol. 12, pp. 88, 2018.

[11] Dennis Stumpf, Stephan Krauß, Gerd Reis, Oliver Wasenmüller, and Didier Stricker, "Salt: A semi-automatic labeling tool for rgb-d video sequences," *arXiv preprint arXiv:2102.10820*, 2021.

[12] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[13] Leland McInnes, John Healy, and James Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[14] Gentry Atkinson and Vangelis Metsis, "A survey of methods for detection and correction of noisy labels in time series data," in *Artificial Intelligence Applications and Innovations: 17th IFIP WG 12.5 International Conference, AIAI 2021, Hersonissos, Crete, Greece, June 25–27, 2021, Proceedings 17*. Springer, 2021, pp. 479–493.

[15] Gentry Atkinson and Vangelis Metsis, "Tsar: a time series assisted relabeling tool for reducing label noise," in *The 14th PErvasive Technologies Related to Assistive Environments Conference*, 2021, pp. 203–209.

[16] Dennis L Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, , no. 3, pp. 408–421, 1972.

[17] Zhaowei Zhu, Jialu Wang, and Yang Liu, "Beyond images: Label noise transition matrix estimation for tasks with lower-quality features," in *International Conference on Machine Learning*. PMLR, 2022, pp. 27633–27653.

[18] Project Jupyter, "Jupyter Notebooks," `https://jupyter.org/`, 2023, [Online; accessed 8-March-2023].

[19] Plotly, "Dash - A web application framework for Python," `https://github.com/plotly/dash`, 2023, [Online; accessed 8-March-2023].

[20] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman, "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9588–9597.

[21] Lee B. Hinkle, Gentry Atkinson, and Vangelis Metsis, "Twristar - wristband activity recognition dataset," https://doi.org/10.5281/zenodo.5911808, Jan. 2022.

[22] Lee Hinkle, Gentry Atkinson, and Vangelis Metsis, "An end-to-end methodology for semi-supervised har data collection, labeling, and classification using a wristband," in *Workshops at 18th International Conference on Intelligent Environments (IE2022)*, 06 2022.