TSAR: a Time Series Assisted Relabeling Tool for Reducing Label Noise

Gentry Atkinson
gma23@txstate.edu
Texas State University
San Marcos, Texas, USA

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

ABSTRACT

Accurately detecting instances in datasets that have been mislabeled is a difficult problem with several imperfect solutions. Hand-reviewing labels is a reliable but expensive approach. Time series datasets present additional challenges because they are not as easily interpreted by reviewers. This paper introduces TSAR, as system for facilitating human review of a small portion of a dataset that it identifies as the most likely to be mislabeled. TSAR's use is demonstrated on real-world time series data.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**; • **Human-centered computing** → **Empirical studies in visualization**.

KEYWORDS

Data Curation, Label Noise, Convolutional Neural Networks, Human Activity Recognition

ACM Reference Format:
Gentry Atkinson and Vangelis Metsis. 2021. TSAR: a Time Series Assisted Relabeling Tool for Reducing Label Noise. In *The 14th PErvasive Technologies Related to Assistive Environments Conference (PETRA 2021), June 29-July 2, 2021, Corfu, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3453892.3453900

1 INTRODUCTION

Every instance in a classification dataset has one or more true, abstract classes. A picture taken of a cat will always have the abstract class "cat" regardless of the processing that is applied to it. But machine learning models do not learn the true class of an instance, but rather the label that the instance has been assigned by the annotators of a dataset [5]. Ideally, the labels in a dataset will perfectly match the true class of every instance but this is not always the case. Any disagreement between the class of an instance and its label is called label noise [6]. There is no firm consensus on the percentage of instances of real-world that5 are mislabeled but good estimates include 3% [14] and 5% [5].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
PETRA'21, June 29–July 12, 2021, Corfu, Greece
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8792-7/21/06…$15.00
https://doi.org/10.1145/3453892.3453900

There are many sources of label noise in large datasets [6]. Domain experts can assign an incorrect label to an instance when the labels are partly subjective [3] or when classes overlap in a feature space. Self-reporting by research volunteers can suffer from subjective labeling, which is a known source of label noise [7]. Electronic processing of records can also produce mislabeled instances through coding errors [7] or when a signal is segmented in a way that does not preserve the event recorded. Finally methods for automatically or semi-automatically annotating data can introduce some label noise.

Human activity recognition (HAR) has important application in assistive technologies and eldercare but is also one of the more complicated problems in bio-signal analysis [16]. Label noise is one of the factors that make HAR difficult for classification problems.

One of the simplest methods for detecting and cleaning label noise in datasets is to have a human review some or all of the instances in a dataset. Some varieties of data, such as image and video, lend themselves easily to this process because they are easily interpretable by human reviewers. However, time series are not one of the classes of data that can be easily interpreted by humans. A good feature extractor can make time series more interpretable and can make relationships in the data more apparent. This feature extractor must be able to preserve the temporal locality of the samples in an instance. Deep learning can be applied to signal data to produce a feature vector which is better suited to visualizing large datasets [18]. This approach accomplishes several goals simultaneously: the dimensionality of the data is reduced to a useful vector, the feature vectors can produce a distance matrix which reveals relationship in full-dataset visualizations like tSNE [8], and the process of domain-specific manual feature engineering is avoided.

This project has developed a system for label review called TSAR (Time Series Assisted Relabeling). This platform employs a deep learning model to identify a user-specified percentage of any time series dataset as being the most likely to be mislabeled. The system then generates a set of visualizations based on the extracted features for a human to review. The usefulness of TSAR's visualization have been evaluated by means of a survey completed by 15 non-expert volunteers and have shown that the group collectively was 81.25% accurate in identifying instances taken from real-world data as being correct or mislabeled. Furthermore the effects of label cleaning using TSAR is evaluated on several real-world datasets and it was found that the classification accuracy of models trained and tested on cleaned data improves from an average of 95.23% to 97.10%.

This paper is organized as follows. In Section 2 we will review work related to this project. In Sections 3 explains the procedures for preparing the evaluation of our system, the process of using

TSAR to remove mislabeled instances, and the method used to test TSAR on several machine learning models. In Section 4 we present the findings of the survey and label cleaning tests. In Section 5 we will consider the meaning and purpose of the results presented in Section 4. In Section 6 we will summarize our work and elaborate on future updates to TSAR.

2 RELATED WORK

Detecting and correcting mislabeled instances is a large problem with many established approaches. Labelfix [14] from 2019, built on earlier work [19] that trained a learning model on datasets of several types and then used the predictions the distance between the predicted label and the assigned label as a probability of an instance being mislabeled. However time series data was not well accommodated by their choice of feature extractor. Another approach is to train several models [3] and flagging they points that they disagree on but doing this could be prohibitively expensive with deep feature extraction.

Several other works have tackled the problem of removing or relabeling mislabeled instances in time series datasets. One approach [20] worked with datasets that included videos that had been recorded of volunteers performing activities. The videos were used by reviewers on Amazon's Mechanical Turk to correct the label of instances in HAR datasets, which included some signal data. This team's previous work [2] used a the output of a machine learning model to identify mislabel instances in time series data in a similar fashion to TSAR but did not allow for the inclusion of a human reviewer in the loop. In this fashion, suspicious instances had to be removed from the dataset altogether rather than correcting their label. Other works have focused on producing learning models for time series data that are robust to label noise [10, 15], rather than removing or relabeling the noisy instances.

Data visualization is a very widely considered problem but some recent contributions have been made which specifically incorporate deep feature extraction on time series data. The work in [18] uses a sparse autoencoder as a feature extractor for signal data and then uses tSNE as a method for visualizing large datasets composed of time series data. Convolutional Neural Networks have been in use since 1998 [11] and have demonstrated an ability to preserve the temporal locality of samples when extracting features from signal data. This functionality has been adapted to HAR [4] problems with good success. A good catalog of visualizations techniques based on data mining can be found in [17].

Human Activity Recognition as a tool for assistive technologies is now commonly included in commercial products like smart watches. Our watches can now encourage us to exercise and alert medical responders that an elderly person has fallen. SmartFall [12] is one example of the latter application.

The contribution of this work is that it combines automatic detection of possibly mislabeled instances with human interpretable visualizations of time-series, so that a human can have the final decision of whether the labels is incorrect and what the new label should be. A second contribution is a short exploration of the ability of human reviewers to recognize mislabeled instances in time series datasets using combined visualization techniques.

2.1 Data Sets

This project has used two real-world datasets which deserve an introduction. The first of them is the UniMiB SHAR dataset[13] which was collected at the University of Milano Bicocca in 2017. Their data were collected from volunteers carrying commercial smartphones in their front trousers pockets at a sample rate of 50Hz. The dataset is well segmented into slices 151 samples wide, which makes them easy to use with neural networks that require a fixed-length input vector. Several label sets are provided for various applications. We have used the ADL (Activities of Daily Life) which offers inertial data for 7,359 instances with the labels: Standing Up, Getting Up, Walking, Running, Going Up, Jumping, Going Down, Lying Down, and Sitting Down.

The second dataset we have relied on is UCI HAR [1]. This set was collected at the Universitat Politecnica de Catalunya in 2012. Samples were collected at 50 Hz using commercial smartphones mounted at the waist. The instances are tri-axial inertial measurements measurements with 128 samples. Only one set of labels is offered with these being: Walking, Upstairs, Downstairs, Sitting, Standing, and Laying. Like the UniMiB set, UCI HAR is neatly segmented and ideal for processing with models that require a fixed input length.

3 METHOD

A label review survey was prepared to test the ability of human reviewers to recognize intentionally mislabeled instances in real world, time series datasets. 96 visualizations were made using the visualization tools in TSAR. Of these, 48 had labels which were intentionally altered but reassigning the label to be any other class with all classes being equally likely. This was done by selecting a new label equally at random from the set of all labels other than the original. One of three methods of feature extraction was applied to each instance before its visualization was generated. These methods were supervised using the upper layers of a trained CNN, unsupervised using the embedding layer of a convolutional autoencoder, and a vector derived from signal processing techniques. Table 2 presents the values that were included in this vector. Finally 48 of the points were produced with 5% label noise having been added to the data before feature extraction. Table 1 summarizes the number of visualization produced for each combination of the three variables.

Table 1: A summary of the variables covered by the label review survey. An equal number of points were generated for each combination of variables: dataset, feature learner, the amount of added noise, and whether the instance was mislabeled or correctly labeled. The visualizations produced for these instances were inserted into the survey in random order.

Variable	Values
Dataset	UniMiB SHAR, UCI HAR
Feature Extractor	Traditional, Supervised, and Unsupervised
Added Noise	0%, 5%

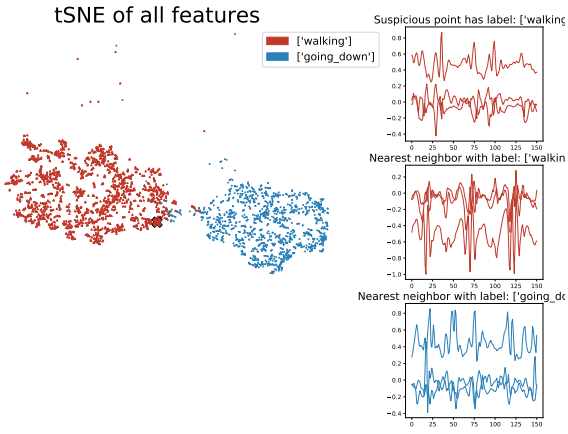


Figure 1: A visualization of one of an instances included in the candidate list generated from UniMiB SHAR. This instance was removed from the dataset by the review. The tSNE plot shows that the point is in a boundary region of the feature space so either label Walking or Going Down is likely to be correct. But the waveform of the instance looks so much more like the Going Down neighbor than the Walking neighbor that the reviewer judged this point mislabeled.

The visualizations produced by TSAR display several pieces of information for the reviewer. To review one instance, TSAR first determines the label of of the nearest point in the feature set that does not share the label of that instance using the Ball Tree algorithm to generate the list of nearest neighbors. A tSNE plot of the feature set for all points that share the instance's label and all points which share the label of the nearest differently-labeled neighbor. tSNE is a technique for dimensionality reduction based on stochastic neighbor embedding[8] which is commonly used when visualizing large datasets. Only two classes are ever represented in the plots used in our survey to keep clutter to a minimum. Alongside the tSNE plot the reviewers are shown three waveforms: the instance they are reviewing, the nearest same-label neighbor in the feature set, and the nearest differently-labeled neighbor in the feature set. An example visualization is shown in Figure 1.

A survey was produced using the 96 visualizations. For each instance the reviewer was asked to indicate if the visualization appeared to be correctly labeled or mislabeled. The reviewers were given a four paragraph explanation of their task but otherwise received no training on label review or Human Activity Recognition. They were also shown one example of an instance with the correct labeled assigned to it and one instance of a mislabeled instance. The reviewers were all students in an undergraduate computer science class.

The mislabeled instances for the survey were hand-generated rather than being automatically selected for review by TSAR. In order to test this capability of the system, TSAR was directed to generate a candidate list of 5% of the UniMiB and UCI HAR datasets using its supervised extractor. 5% was chosen as the size of the candidate list following the observation[5] that this is a rough estimate for the prevalence of label noise in real-world datasets. TSAR then generated a visualization for each instance in the candidate list. These were produced using the same procedure described for the survey visualizations. One author reviewed the candidate list and removed every instance that appeared to be mislabeled. Several machine learning models were trained and tested on uncleaned and cleaned data and their accuracies were computed. The models selected were: an SVM, a 3-Nearest Neighbors classifier, a decision tree trained using C4.5, and Naive Bayes. These models were selected because TSAR uses the output of a deep neural network to identify the instances in a feature set which are most likely to be mislabeled. Using a neural network to clean a dataset and then testing the system using another neural network might only demonstrate that our systems reinforces the biases of a single model, but for completeness a 3-layer and 6-layer dense neural network are included in the models tested. Our hypothesis is that cleaning a dataset using TSAR will produce model-agnostic improvements in accuracy.

3.1 Feature Extraction

The goal of a feature extractor in this context is two-fold. First, it should minimize the overlap between classes in the visualization of the feature set. Ideally, the two-label visualization will be two crisply divided blobs on far ends of the chart. Second, the feature extractor should preserve the visual characteristics of the signal. The nearest neighbor in the feature set should be the most similar in visual appearance to the measured signal.

These requirements make preserving the chronological ordering of samples during the process of feature extraction even more important. Convolutional Neural Networks produce features based on small, sliding windows which are convoluted across an instance. This makes features which preserve the temporal locality of samples in the instance, and makes them a good choice of feature extractor for TSAR. Other works [4] have demonstrated the power of CNNs as feature learners on bio-signal data.

There are, broadly speaking, two ways that a CNN can be trained as a feature extractor. A supervised method is to train a CNN classifier on the label set and then discard the layers closest to the output. This leaves the earlier trained convolutional layers to serve as the feature extractors. An unsupervised method for creating a convolutional feature extractor is to construct an autoencoder [9] which trained to reproduce the input vector through a series of convolutional and dense layers. The decoder portion of the trained model is discarded leaving the middle (embedding) layer as the new output. The structure of the extractor is demonstrated in Figure 2.

TSAR offers both extractor training methods. The supervised extractor has generally been more effective in our experiments, but in sets with a high prevalence of label noise, it may not be possible to reliably train a supervised model. This will be discussed further in Section 5.

A feature extractor based on traditional signal processing and hand-crafted features was included in the list of feature extractors used to produce the visualizations for the survey. The types of values used by this feature extractor are listed in Table 2. This feature extractor was included only as a baseline for comparison to the two feature learners. A feature engineering approach to signal processing would require cumbersome tweaking to fit particular

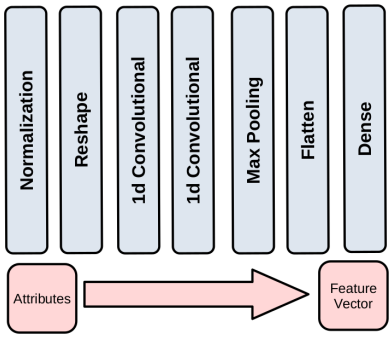


Figure 2: A visualization of TSAR's convolutional feature extractor that was prepared based on a Tensorboard of the unsupervised model during training. This network has had the lower classification layers removed. The upper layers of the two deep feature extractor (supervised or unsupervised) are identical.

data domains (e.g. EEG vs IMU data), so deep feature learners are much more portable.

Table 2: The list of features extracted from signals by the traditional feature extractor. These features were selected for being broadly used in signal processing and for being easily computed.

• Mean • Standard Deviation • Absolute Energy • Sum of Changes • Auto Correlation • Count of Values Above Mean • Count of Values Below Mean • Kurtosis • Longest Strike Above Mean • Zero Crossing Rate • Number of Peaks • Sample Entropy • Welch Spectral Density (6 coefficients)

3.2 Candidate List Generation

Manually reviewing data can be an incredibly time-consuming process. Ideally, all incorrect labels could be altered or removed by manually reviewing the smallest candidate list of uncertainly labeled points possible. TSAR implements a Single Learner method [7] to identify the instances in a dataset that are most likely to be mislabeled. A deep neural network is trained on the extracted features and used to predict a label for every instance. The dot product is calculated for the one-hot encoding of the predicted label and assigned label. This method of generating distances between assigned labels and predicted labels has been used [14] previously. The list of instances is sorted by the value of the dot product. The top of this list is now more likely to be mislabeled. The user of TSAR requests some small percentage of the list and visualizations ore generated for review. The pipeline employed by TSAR is presented in Figure 3.

4 RESULTS

4.1 Human Review

A label review survey was prepared using images generated by TSAR to test the effectiveness of human review of noisy labels in

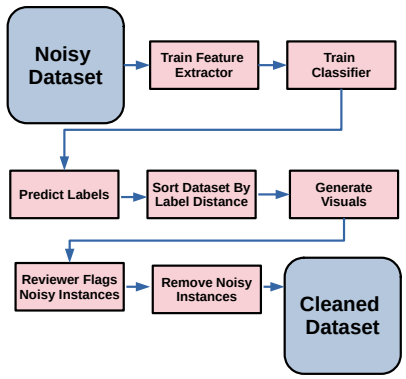


Figure 3: Process pipeline used by TSAR to identify and remove the mislabeled instances from datasets.

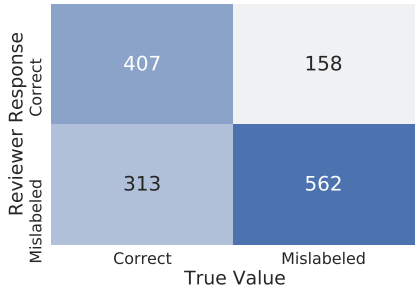
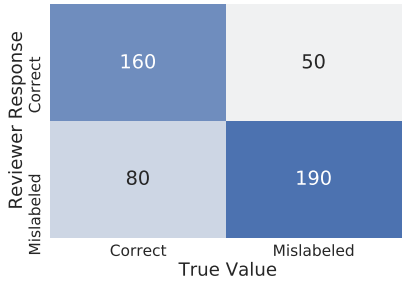


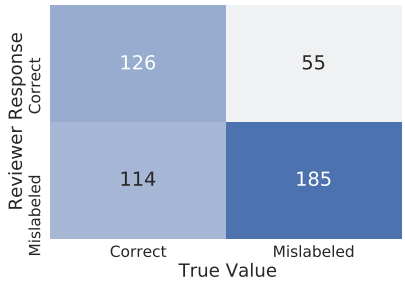
Figure 4: The confusion matrix of all individual responses to the label review survey.

time series data as described in Section 3. This section summarizes the responses of the volunteer reviewers. 15 untrained undergraduate students responded to the 96 question survey making a total of 1440 responses. Of these 562 correctly indicated that an instance was mislabeled, 407 correctly indicated that an instance was correctly labeled, 313 misidentified an instance as mislabeled, and 158 misidentified a point as being correctly labeled. A confusion matrix of these results is presented in Figure 4.

If we consider only the responses to visualizations produced with a deep feature extractor, the reviewers correctly identified 661 out of 960 visualizations giving them an accuracy of 68.9%. Out of their 960 responses to this set of instances 569 marked a point as mislabeled. 375 of these responses correctly identified the instance as mislabeled and 194 misidentified the instance as being mislabeled. This gives the individual responses a precision of 65.9%. The accuracy of the reviewers was higher using the visualizations produced with the supervised extractor: 72.9% from the supervised extractor vs. 61.9% from the unsupervised extractor. A confusion matrix for the 480 responses to the supervised instances is presented in Figure 5a and the 480 responses to the unsupervised instances in Figure 5b. The accuracies and precisions of responses to the various extractors are presented in Table 3.



(a) Supervised Feature Extraction



(b) Unsupervised Feature Extraction

Figure 5: The collected individual responses to all survey visualization generated using a deep feature extractor.

Table 3: The accuracies and precisions of subsets of the 1440 individual responses to the label review survey. Filtering the responses to represent particular variables (e.g. the feature extractor) eliminates many responses, so the number of responses used to calculate these accuracies and precisions is listed.

Feature Extractor	Added Noise	Resp.	Acc.	Prec.
All Deep	Combined	960	68.9%	65.9%
Supervised	0%	240	80.0%	77.7%
Supervised	5%	240	65.8%	63.6%
Supervised	Combined	480	72.9%	70.4%
Unsupervised	0%	240	65.4%	64.3%
Unsupervised	5%	240	64.2%	60.0%
Unsupervised	Combined	480	64.8%	61.9%

The above results are based on individual responses but we can also consider the viewers as "voting" on the correctness of each label. Of the 96 responses, 61 were marked as mislabeled by 8 or more of the 15 reviewers and 35 were marked as correctly mislabeled. The confusion matrix for the voting responses is presented in Figure 6 and the accuracies of the deep feature extractors is re-computed in Table 4. Most notably the accuracy of the reviewers on the instances with the supervised feature extractor rises to 90.6%.

4.2 Label Cleaning

Six common machine models were trained and tested on the cleaned and uncleaned supervised feature sets: a Support Vector Machine, K-Neareast Neighbors, a C4.5 decision tree, Naive Bayes, a three-Layer ANN, and a six-Layer ANN. The accuracies of these models are presented in Table 5. The values presented are the averages from repeating the same process of training and testing on features generated from the UniMiB SHAR and UCI HAR datasets using TSAR's generated features. In no case did the performance of the model degrade after label cleaning. The arithmetic mean accuracy of the six models was 95.2% before cleaning and 97.1% after cleaning. This is an improvement of 1.9% in the average accuracy which is indicates that the error rate of the models is 60.4% of what it was before cleaning.

As described in Section 3, the feature sets was extracted from UniMiB SHAR and UCI HAR using the supervised, deep feature learner. The nine-class ADL label set was used for UniMiB SHAR in this experiment. These feature sets were then used to generate visualizations which were identified as correctly labeled or mislabeled by a human reviewer. The six listed models were trained and tested on the noisy feature set, and then six new models were trained and tested in the cleaned feature set. The improved accuracies are not a consequence of re-train any models.

It is a curious artifact of this experiment that the SVM and KNN classifiers outperform the two neural networks. When interpreting this result, it is important to keep in mind that the feature sets being classified are the output of deep, convolutional networks. The high accuracy of the KNN classifier demonstrates that TSAR's feature

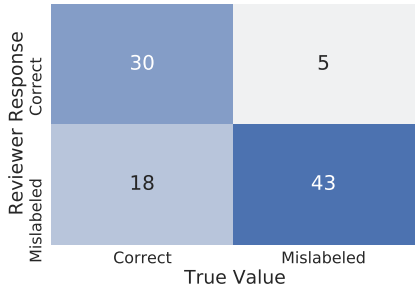


Figure 6: The confusion matrix of the collaborative responses to the survey instances. All three feature extractors are represented in these results.

Table 4: The accuracies and precisions of collective responses. These values are computed based on whether a majority of reviewers believed a point was mislabeled.

Extractor	Noise	Instances	Acc.	Prec.
All	Combined	96	76.0%	70.5%
All Deep	Combined	64	81.3%	76.3%
Supervised	0%	16	100.0%	100.0%
Supervised	5%	16	81.3%	77.8%
Supervised	Combined	32	90.6%	88.2%
Unsupervised	0%	16	75.0%	75.0%
Unsupervised	5%	16	68.8%	61.5%
Unsupervised	Combined	32	71.9%	66.7%

Table 5: The accuracies of several machine learning models trained and tested on uncleaned feature sets compared to identical models trained and tested on cleaned feature sets.

Model	Uncleaned Acc.	Cleaned Acc.
SVM	96.4%	97.8%
KNN	96.7%	97.8%
Decision Tree	94.6%	96.6%
Naive Bayes	93.3%	96.2%
3-Layer NN	95.2%	97.3%
6-Layer NN	95.3%	97.1%

extractor is generating a feature set wherein instances generally share a label with their nearest neighbors.

5 DISCUSSION

There is always some risk involved in label cleaning. Cleaning a training set can degrade the performance of a model if the test set also contains label noise [3]. For this reason we have chosen to clean both the training and testing data for our experimental model evaluations. This may not be appropriate for some circumstances but shows that TSAR will be useful for models that are going to be trained and then deployed in the real world (e.g. a fall detecting smart watch). However that are going to be trained and then tested on pre-collected data, it would be better to clean either both or neither.

Another risk of label noise detection is that if poorly implemented it can degrade to outlier detection. An instance might fall into an unusual region of the feature space but still not be mislabeled. To help avoid the the candidate list generated by TSAR considers only the distances between assigned labels and the predicted labels. But of course the predicted labels are dependent on the features, so the candidate list is in no way independent of the feature set. Some good judgement will have to be exercised by reviewers to avoid removing correctly labeled but unusual examples from the feature set. Figure 7 shows a good example of an instance that will come down to a judgement call on the part of the reviewer.

One curious result from the survey is worth bringing up. Out of the 1440 responses from the 15 reviewers, 875 responses indicated that the instance was mislabeled as compared to 565 indicating that an instance was correctly labeled. Keeping in mind that an equal number on the instances in the survey were correctly labeled and mislabeled (48 of each), it is surprising to see that the reviewers were about 50% more clicks on the 'Mislabeled' button as compared to the 'Correct' button. This effect is even more pronounced when the reviewers vote on instances with 61 out of the 96 being identified as mislabeled by 8 or more of the reviewers, or 63.5% of the 96 instances. This might be an artifact of our visualizations, an expression of some psychological phenomenon, or a coincidence born of a small sample size. In any case, this effect will be considered as TSAR is refined.

It was mentioned in Section 3 that both a supervised and an unsupervised feature extractor are included in TSAR, but that only the supervised feature extractor was used for label cleaning before model testing. This decision was motivated by the results of the

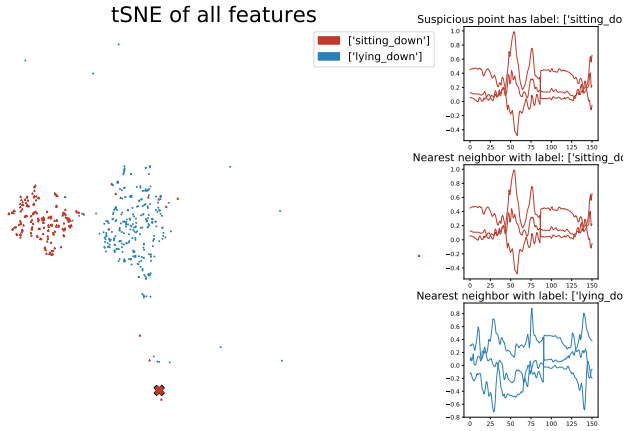


Figure 7: An example point from the UniMiB SHAR dataset. The tSNE plot shows that this instance is in a sparse region of the feature and the waveforms show that this signal shares similarties to the Sitting Down and Lying Down examples. Difficult instances like this should be left in during label cleaning.

survey, which indicated that the responses on the visualizations prepared using the unsupervised extractor were at best 71.9% accurate as compared to the 90.6%. It was assumed that the unsupervised feature extractor would be less affected by added label noise in the datasets, and that this might make it overall the more reliable feature extractor. While the survey results show that the unsupervised feature extractor is indeed less affected by the addition of 5% label noise, the supervised feature extractor was the more reliable tool even with the added noise. So there might be some amount of label noise where the unsupervised extractor becomes the more reliable extractor (a linear interpolation from the survey results indicates that that is 34.3% of instances being mislabeled) but at the levels of label noise that can reasonably be expected in the UniMiB and UCI datasets, the supervised extractor was clearly a better choice. But that's only meaningful for these feature extractors on these datasets with these visualization techniques. That choice is by no means a broad conclusion.

6 CONCLUSION AND FUTURE WORK

TSAR is very much still a work in progress but the initial results show that can be used to reliably improve the performance of machine learning classification models on real-world data. In every case the performance of a model that was trained and tested on cleaned data is better than the same model trained and tested on uncleaned data. Human Activity Recognition is one of the more difficult classification problem [16] in biosignal analysis and researchers should be conscious of any technique that could make their models more reliable. The usefulness of HAR classifiers in assistive technologies is well understood. Any decrease in the error rate of the models that drive these systems will have a real world impact.

This project has also demonstrated the limitation of human review in label cleaning. Human review is a powerful tool but we can

never assume that a human reviewer will perfectly accurate. We have shown that our 15 reviewers were colllectively 90.6% accurate in identifying mislabeled instances in real world data using a supervised extractor but individually only 72.9% accurate. Research teams who are using any system for label review that involves human review should be conscious that using several reviewers in a voting system could be a much better choice than one reviewer acting alone. Having said that, our models were tested using features that were cleaned by a single reviewer and still showed a reliable increase in performance.

At this time, TSAR completely removes the instances that the human reviewers indicates from the training and testing set. A more sophisticated approach might be to relabel those instances rather than removing them. Machine learning models generally perform better when they have more data to train on, so it would be reasonable to assume that an approach that leaves more instances in the training set will be better. Using this approach will require that safeguards be put in place to avoid biasing the model towards the researcher's assumptions about the data.

ACKNOWLEDGMENTS

The authors would like to thank the Texas State University IMICS lab for their support and insight.

REFERENCES

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. 2013. A public domain dataset for human activity recognition using smartphones.. In *Esann*, Vol. 3. 3.

[2] Gentry Atkinson and Vangelis Metsis. 2020. Identifying label noise in time-series datasets. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*. 238–243.

[3] Carla E Brodley and Mark A Friedl. 1999. Identifying mislabeled training data. *Journal of artificial intelligence research* 11 (1999), 131–167.

[4] Federico Cruciani, Anastasios Vafeiadis, Chris Nugent, Ian Cleland, Paul McCullagh, Konstantinos Votis, Dimitrios Giakoumis, Dimitrios Tzovaras, Liming Chen, and Raouf Hamzaoui. 2020. Feature learning for Human Activity Recognition using Convolutional Neural Networks. *CCF Transactions on Pervasive Computing and Interaction* 2, 1 (2020), 18–32.

[5] Benoît Frénay, Ata Kabán, et al. 2014. A comprehensive introduction to label noise.. In *ESANN*.

[6] Benoît Frénay and Michel Verleysen. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems* 25, 5 (2013), 845–869.

[7] Donghai Guan and Weiwei Yuan. 2013. A survey of mislabeled training data detection techniques for pattern classification. *IETE Technical Review* 30, 6 (2013), 524–530.

[8] Geoffrey E Hinton and Sam Roweis. 2002. Stochastic neighbor embedding. *Advances in neural information processing systems* 15 (2002), 857–864.

[9] Mark A Kramer. 1991. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal* 37, 2 (1991), 233–243.

[10] Hyeokhyen Kwon, Gregory D Abowd, and Thomas Plötz. 2019. Handling annotation uncertainty in human activity recognition. In *Proceedings of the 23rd International Symposium on Wearable Computers*. 109–117.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[12] Taylor R Mauldin, Marc E Canby, Vangelis Metsis, Anne HH Ngu, and Coralys Cubero Rivera. 2018. SmartFall: A smartwatch-based fall detection system using deep learning. *Sensors* 18, 10 (2018), 3363.

[13] 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), 1101.

[14] Nicolas M Müller and Karla Markert. 2019. Identifying Mislabeled Instances in Classification Datasets. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[15] Sudipta Paul, Shivkumar Chandrasekaran, BS Manjunath, and Amit K Roy-Chowdhury. 2020. Exploiting Context for Robustness to Label Noise in Active Learning. *arXiv preprint arXiv:2010.09066* (2020).

[16] Sreenivasan Ramasamy Ramamurthy and Nirmalya Roy. 2018. Recent trends in machine learning for human activity recognition—A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1254.

[17] Georgiy Shurkhovetskyy, N Andrienko, G Andrienko, and Georg Fuchs. 2018. Data abstraction for visualizing large time series. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 125–144.

[18] Ye Yuan, Guangxu Xun, Qiuling Suo, Kebin Jia, and Aidong Zhang. 2017. Wave2vec: Learning deep representations for biosignals. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1159–1164.

[19] Xinchuan Zeng and Tony R Martinez. 2001. An algorithm for correcting mislabeled data. *Intelligent data analysis* 5, 6 (2001), 491–502.

[20] Liyue Zhao, Gita Sukthankar, and Rahul Sukthankar. 2011. Incremental relabeling for active learning with noisy crowdsourced annotations. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE, 728–733.

A ONLINE RESOURCES

Code and collected data for this project can be found at https://github.com/imics-lab/TSAR