A Survey of Methods for Detection and Correction of Noisy Labels in Time Series Data

Gentry Atkinson(✉) and Vangelis Metsis(✉)

Texas State University, San Marcos, TX 78666, USA
{gma23,vmetsis}@txstate.edu

Abstract. Mislabeled data in large datasets can quickly degrade the performance of machine learning models. There is a substantial base of work on how to identify and correct instances in data with incorrect annotations. However, time series data pose unique challenges that often are not accounted for in label noise detecting platforms. This paper reviews the body of literature concerning label noise and methods of dealing with it, with a focus on applicability to time series data. Time series data visualization and feature extraction techniques used in the denoising process are also discussed.

Keywords: Label noise · Machine learning · Data quality

1 Introduction

Machine learning is well established as a useful field in artificial intelligence. But poor quality training data can quickly degrade the performance of a machine learning model in terms of accuracy, time to train, and the size of the classifier [69]. Noise has been defined as any disruption in the observed relationship between the features of an instance in a dataset and its class [25]. When this disruption occurs in the features it can be called attribute noise and in the labels, label noise [26]. The focus of this work will be on label noise, which is common in real-world datasets [69], but less widely addressed by denoising approaches.

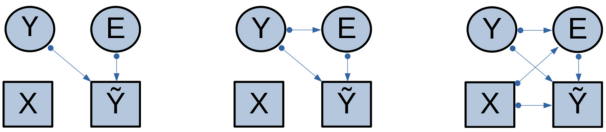
Label noise is only one of the names used to refer to degraded quality in the assigned labels of datasets. Other names include: class noise, mislabeled data, poorly annotated data, and the borderline accusatory sloppily labeled data [52]. This work has chosen to use the name label noise because it is common and because it conforms with the taxonomy presented in [25]. Following that taxonomy, every instance of data has an abstract and true identification known as its class (Y). Its label ($\tilde{Y}$) is an assigned annotation which should, but does not always, identify the instance's correct class.

Frenay's taxonomy[26] divides label noise into three categories: noise completely at random, noise at random, and noise not at random based on the

Published by Springer Nature Switzerland AG 2021
I. Maglogiannis et al. (Eds.): AIAI 2021, IFIP AICT 627, pp. 479–493, 2021.
https://doi.org/10.1007/978-3-030-79150-6_38

dependencies between Y, $\tilde{Y}$, the feature space X, and an error rate E. When noise is completely at random (NCAR), every class is equally likely to be mislabeled. Noise at random (NAR) has an error rate that is affected by the class. Noise not at random (NNAR) mislabels data at a rate that depends on both the class and the feature space. The dependencies are shown in Fig. 1. These categories were inspired by early work on missing data [46].

Time series data offer challenges to researchers which are not present in other classes of data, such as images or text. The phrase time series refers to "[a] set of observations arranged chronologically", or in more charming terms from the same author a "wiggly record" [42]. Time series can be modeled as the output of a continuous function on some set of time steps. This continuity makes time series different from other ordered, sequential data. A feature extractor working on time series data must preserve the temporal relationships between nearby samples in the data.



(a) NCAR (b) NAR (c) NNAR

Fig. 1. A visual taxonomy of label noise adapted from [25]. (a) shows noise completely at random (NCAR), (b) shows noise at random (NAR), and (c) shows noise not at random. Y represents true classes of instances, E the error (or mislabeling) rate, X the features, and $\tilde{Y}$ the assigned label. Arrows indicate a dependency between elements.

Label noise is an old and long-standing problem discussed since the early days of digital data analysis [19]. Section 2 of this paper will summarize recent works in noise detection. Section 2 is subdivided following a taxonomy of detection techniques defined in [30]. This work distinguished detection techniques based on the type of learning as: local learning, ensemble learning, or single model learning.

Section 3 will document recent approaches to feature extraction and visualization for time series data. Human review of labels is an effective technique for cleaning noisy labels [69]. But, the ability of human annotators to clean labels in time series data is largely dependent on their ability to visualize that data meaningfully.

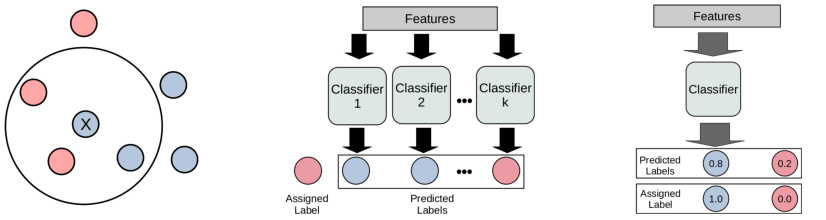
Section 4 will explore techniques for mitigating the effect of classifier performance with noisy labels. Broadly the approaches to improve a model are: data cleansing, robust learning algorithms, and model hardening [25]. Section 5 concludes with a discussion of the material presented and our main observations.

2 Detection of Label Noise

Following the work presented in [30], label noise detection platforms can be divided into local learning methods, ensemble learning methods, and single classifier learning methods. This division is based on the method used to distinguish one instance as being mislabeled. Local learning methods compare instances to their nearest neighbors using methods such as K-Nearest Neighbors (KNN). Ensemble learning methods train multiple classifiers and identify mislabeled points based on a vote of those classifiers on the correct label for each instance. Single model learning methods train a single classifier (often a neural network) on some data and use the labels predicted by that classifier to distinguish mislabeled instances in the data. These three methods are summarized in Fig. 2.

2.1 Local Learning

Local learning methods assume that instances that are close in the feature space should share a label [30]. They frequently employ K-Nearest Neighbors which has the advantage of not needing to be trained. KNN is identified in [20,40] as being exceptionally robust to label noise. This particular property of that model will be discussed more in Sect. 4 but it should be sufficient for now to say that this observation makes it a reliable choice for a noise detection system. A demonstration of the use of KNN for identifying label noise is shown in Fig. 2a.



(a) Local Learning (b) Ensemble Learning (c) Single Classifier

Fig. 2. A summary of the three noise detection approaches defined in [30]

Several adaptations have been made to the basic KNN algorithm (identified in some older source as Instance-based Learning [2]) to make it suited to the task of identifying mislabeled instances. Edited nearest neighbor (ENN) automatically removes all instances misclassified by KNN from a training set [58] while its cousin Repeated-ENN iteratively applies ENN until all instances share a label with their nearest neighbors [58]. An advancement on Repeated-ENN, called All-KNN, was presented in [55] that used increasing values for K across iterations.

Later work applied Gabriel Graphs [56] and Relative Neighbourhood Graphs [34] to improve on the ability of KNN to recognize mislabeled instances in overlapping class regions of a dataset. The technique used a proximity graph to refine

the set of neighbors used in order to mitigate occurrences of label noise in the set of neighbors used by a KNN classifier [45].

Biclusters, selections of features and instances that demonstrate high coherence of values across attributes and labels [21], have also been applied to the problem of identifying label noise. These selections are learned in an unsupervised manner. BicNoise [24] computed a mean square residue value (a measure of error between real and calculated values) of sets of instances as instances were experimentally inserted into highly coherent subsets from the training data. Instances that caused the MSRV to rise over some threshold were removed as noise.

Cluster validation measures provide a quantitative metric for the fit of a data partitioning. These measures have also been employed as a label noise filter [13]. This approach treated the dataset labels as clusters and calculated several cluster validity measures for each instance, following the intuition that poorly clustered points would be more likely to be mislabeled [13]. The measures employed were: the Silhouette Index, Connectivity, and the Average Intracluster gap.

2.2 Ensemble Learning

Ensemble learning can improve the performance of machine learning techniques [30]. Comparing the outputs of several classifiers can also be a useful approach for identifying mislabeled data [30]. This can either be several classifiers of different types (e.g., a decision tree, KNN, and a neural network) or several classifiers of the same type trained on different manipulations of the training data [30].

Employing sets of classifiers as a noise detection platform was shown to improve classification accuracy for noise levels up to 30% [18]. The technique presented in [18] developed a noise filter based on the residual error of classifiers trained on noisy data. This work was able to demonstrate the residual noise from ensemble classifiers was superior to residual noise from a single classifier and majority voting by ensemble classifiers.

Data partitioning is one strategy for ensemble learning approaches. The Partitioning Filter [70] was one technique that used data partitioning by dividing large datasets into several subsets, developing a good rule set to classify each partition, merging the rule sets into a single set for the full dataset, and filtering instances that are misclassified by the new ruleset. Partition Filtering was showed to be effective on datasets with up to 40% label noise [70]. Another application of data partitioning separated a full dataset into several overlapping partitions, trained several classifiers of each partition, and then tested combinations of majority voting and consensus voting on the label predictions from the classifiers [31]. Mislabeling rates of up to 40% were also tested in this paper but in some datasets, the accuracy of the noise detector fell below 50% when the mislabeling rate was higher than 30% [31].

Repeated labeling was applied in [48] to improve the label quality of chosen sets of low-quality instances. This work focused on cheap and crowd-sourced human labelers and developed analytical measures of label quality to determine

instances that should be relabeled. The output of a model trained on the labeled instances was used to compute uncertainty of labeled datasets, and that measure was used to inform the relabeling process [48].

An adaptation of 10-fold Cross Validation is presented in [38] that used an ensemble of machine learning classifiers to filter mislabeled instances. Although the ML models employed in this approach were comparatively simple, this team reported classification accuracies in datasets with up to a 20% mislabeling rate can be increased to the same rate of accuracy as a classifier trained on data with no mislabeled instances using their filter [38].

2.3 Single Classifier Learning

Single classifier learning detection methods are more suitable for highly dynamic noisy data [30]. This approach can also be more efficient than ensemble learning approaches [30] but newer single classifier learning approaches tend to utilize deep learning models with powerful feature extraction techniques, which can easily require more time and computing power to train than an ensemble of simpler models.

It has been observed that mislabeled points do not necessarily behave like outliers in datasets [57]. Support vector machines (SVM) have been employed in a single classifier learning approach using a form of data partitioning to address this issue [57]. In this approach, a subspace of the full feature space was selected using domain expertise to train a classifier that would better identify the true class of the classified instances.

Label noise identification can also be incorporated as a portion of a classification model. The authors in [17] designed and implemented a model which included a sparse Bayesian Logistic Regression algorithm that was used to identify mislabeled instances. This training algorithm alternated between training the classifier and estimating label noise probabilities. The output of this approach is both a robust classifier and a list of suspect instances that could be addressed later.

Selecting an appropriate feature extractor can have a substantial effect on the efficacy of a classifier (a point that will be discussed further in Sect. 3). Labelfix [39] is a platform that automatically selected a deep feature extractor and trained a classifier using the learned features. Instances were then sorted based on the distance between the assigned one-hot label vector and the predicted label vector. This detection technique can also be built into a model training pipeline [44].

Our previous work has also expanded on the approach of sorting instances based on the distance between assigned and predicted labels [6,7]. A convolutional neural network (CNN) is applied as a feature extractor that is well-suited to time series data and the extracted features are used to produce interpretable visualizations for human reviewers. Human review is an effective technique for noise removal but is generally too expensive to be applied to a full dataset [69].

3 Feature Extraction and Visualization

Time series are a class of data that covers many different applications: financial, medical, engineering, scientific, social, and military [49]. Many of these fields can produce attribute vectors that can be very large [4]. Consider 10 s sound clips recorded at 44,100 Hz, or a minute of a 64-channel electroencephalogram (EEG) collected 1000 Hz. Either example would be untenable as an input layer for a neural network, or as an input vector for another model (e.g., an SVM or decision tree).

The human brain is a very powerful tool for recognizing and finding connections between time series but this ability depends on the visual appearance of the data [33]. Like the classifiers mentioned earlier, the human mind can benefit greatly from techniques of data abstraction that preserve the temporal properties of the original signal. The size of contemporary data is a significant challenge for time series visualization and analysis [49].

A good feature extractor is often enough by itself to help reduce the impact that label noise can have on a classifier [43]. What's more, extracted features that have captured the temporal relationships of samples in a time series can be processed as numerical data using many of the techniques presented in Sects. 2 and 4 that were not specifically crafted for temporal data.

Dimensionality reduction techniques can reduce the size of attributes without sacrificing the information they have captured. Principal component analysis (PCA) [35] is one approach to reduce the size of an attribute vector by selecting the most impactful channels from the full attribute set. PCA has been successfully applied to time series data [11,61]. PCA can be used to aid a task of visualization by selecting the two or three most meaningful features to plot in a flat figure [23].

Autoencoders were introduced as a method for dimensionality reduction [37] but have also been used effectively as feature extractors for signal visualization [62,63]. By mapping attributes into lower dimension embedding spaces, autoencoders can produce features that capture the relationship between instances in a way that is learn-able. This set of features can be further projected into 2 or 3 dimensions using a technique such as tSNE [32] to make the full dataset visually interpretable [7,12,63].

Convolutional neural networks have been used as part of label noise detection platforms [6,39] and have also been shown to be particularly effective as feature learners for time series data [23]. CNNs train nodes on small clusters of samples in the original attribute set. This process effectively captures the temporal relationship between samples in time series data. CNN layers can be incorporated smoothly feature learners in single model learning noise detection platforms [39] or built into autoencoders. Features extracted from signal data in [23] showed reliable separation of classes in 2-dimensional visualizations of data collected from audio sources and inertial measurement units (IMUs).

Segmentation divides time series instances into contiguous runs of samples that share some trend. A good technique for segmentation is one of the fundamental approaches to time series analyses [27]. A method for segmenting signals

based on binary tree representations was presented in [27] and a similar technique followed in [67]. Both approaches build binary trees of samples whose root is a Perceptually Important Point (PIP) [22]. A simplified demonstration of signal segmentation based on PIP is presented in Fig. 3.

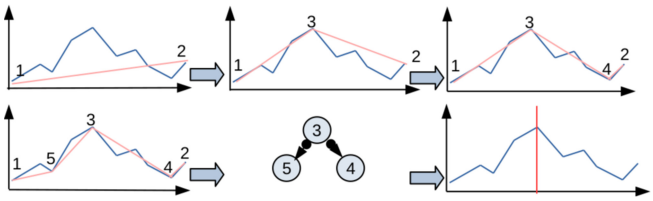


Fig. 3. A simplified demonstration of SB-Tree segmentation as outlined in [27]. PIPs are identified, built into a tree (the first and last point are generally excluded), and the roots of each tree are considered first to be a cut point for segmentation.

Prarzen notes that any time series data can be thought of as some function or sum of functions whose domain is a set of time steps [42]. Several feature extractors have been constructed around approaches that approximate the fundamental functions of time series. Fourier transforms represent arbitrary signals in the frequency domain, and [1] demonstrated that only a few fundamental frequencies can effectively abstract time series data while reducing the size of the representation. [60] presented a method for online approximation of a time series using polynomials.

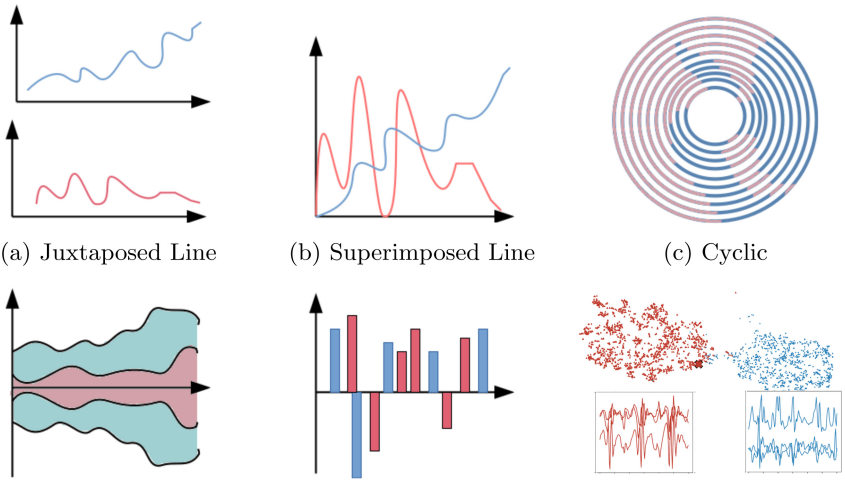
Clustering has been applied to the problem of feature extraction. VAFLE, an approach based on spectral clustering, was able to reduce the dimensionality of data while identifying points of interest [28]. Another work has used Haar wavelet transformation to inform the dimensionality of a feature vector which was then clustered using hierarchical K-means [66].

A good visualization should be easily interpretable by a user. Even with a simple graph like a line plot (or waveform, or "wiggly record" [42]) there are choices that need to be made with interpretability in mind. Bertin Indexing has been shown to produce line plots from time series financial data that is easier to use than line plots prepared using linear scale juxtaposition and log scale juxtaposition [3]. Bertin Indexing scales heterogenous time series to make comparisons easier [10].

One consideration of time series datasets is that the data are not always exclusively time datasets. An example would be geo-referenced sensor data being collected from a sensor network. One system to address this example focuses on giving the user a "big picture" understanding of the sensor network over time [51]. This system allowed the user to view a 2D map of signals which was clustered by similarity with each region have a representative waveform overlayed on it.

There are many standard techniques for visualizing time series data. A collection of these techniques are presented in the early works of [4] and [50]. VisInfo

is a more recent product whose goal is to provide visual access to time series data [9]. Some common and accepted methods for presenting time-oriented data are presented in Fig. 4.



(a) Juxtaposed Line (b) Superimposed Line (c) Cyclic

(d) River (e) Bar Chart (f) tSNE

Fig. 4. A selection of plotting techniques suited to time series data. The juxtaposed and superimposed line plots are useful for comparing linear signals, as is the bar chart. The cyclic plot is best used for finding periodicities in cyclic series. River plots can be used for percentage estimation. tSNE allows full dataset exploration. (f) shows a tSNE plot of two classes of data juxtaposed with line plots of an instance from each class.

4 Label Correction

Detecting label noise is only the first step in improving the performance of supervised classifiers when working with noisy data. Some method also needs to be adopted for dealing with that noise. Three main approaches exist for dealing with label noise: training noise-tolerant models, cleansing data sets, and crafting noise-tolerant learning algorithms [25]. Techniques do not always fit neatly into only one of the three categories so each body of work will be discussed in the section that best fits it, but might overlap with work discussed in other sections.

4.1 Data Cleansing

Having identified some of the mislabeled instances in a dataset, data cleansing techniques will either remove or relabel those instances. The first approach is easier to implement, but re-labeling has the advantage of not removing training data that could improve the performance of a classifier.

TimeCleanser [29], an approach based on visual analytics, capitalized on the documented efficacy of the human eye [49] to analyze large datasets. As the name suggests, this platform focuses on temporal data, which the authors note has very specific challenges which make it distinct from other classes of data [29]. Users of this product can select from several good graphics to visually interpret datasets. Instances which the user identifies as noise are removed from the data.

Another approach that relied on human analysis for data cleansing is presented in [68]. This platform used crowd-sourced reviewers to re-annotate samples of IMU data that had been selected from a human activity recognition dataset. This approach used video recorded of the instances to allow human reviewers to correct the label of instances in the dataset which an active learner had trouble labeling correctly.

A pair of techniques that did not rely on human reviewers were introduced in [41]. The first of the two was named Self-Training Correction, which uses a classification model with an integrated noise filtering algorithm. The second technique employs K-Means clustering repeatedly using weights based on the assigned label for each instance. The labels in datasets are updated using a technique which these authors have named Polishing Labels in tribute to the data polishing algorithm described by Cho-Man Teng [53].

Similar to the Self-Training Correction described above was the earlier work Automated Data Enhancement (ADE) [64]. This approach used a neural network that was repeatedly re-trained on a noisy dataset, each time assigning or updating a probability vector representing each class to each instance. As the model was re-trained the probability vector gradually drifts away from the mislabelled class and towards the true class.

Probabilistic labeling can be used as part of a label cleaning platform. By assuming that there is some hidden probability of each point having a true class distinct from the assigned label Bootkrajang and Kaban developed an analysis which they called robust Normal Discriminant Analysis (rDNA) [15]. Labels in the dataset are flipped based on a function that calculates the probability of the instance being in each class of the dataset.

4.2 Robust Learning

Boosting is a machine learning method that gives greater weight to instances that a classifier struggles with during training [47]. The method, including the popular AdaBoost optimizer, greatly improves the training rate of machine learning models but can be particularly susceptible to the presence of label noise in training data [36]. Bagging, by contrast, partitions the dataset into subsets and trains a classifier on each [8] outperforms boosting on data that have noisy labels [36]. Boosting can be modified to make it more robust to label noise by adding probabilistic factors representing uncertainty in the assigned labels [16]. The process of label flipping has been expanded by the same author in later work [14].

Ensemble classifiers can be trained with resilience to label noise in mind. One approach is to use a principle of minimum-variance during the combined training of the several classifiers [71]. This approach minimizes the error rate of

the ensemble by minimizing the sum of the variances of the collected classifiers. This technique was demonstrated to be effective in tasks of active learning from noisy, stream data [71].

Skeptical Supervised Machine Learning introduces a confidence measure that is used to represent the reliability of annotations made by human labelers [65]. Labels are generated both by human input and by the predictions of an ensemble of machine learning classifiers. The models are trained iteratively on the human input and conflict resolution is applied to decide the correct label for each instance.

Rather than removing mislabeled instances from datasets, we can omit them while training a classifier. A loss function has been proposed that skips confusing samples from training in order to improve the performance of deep neural networks on noisy data [54]. Instances with high cross-entropy error are driven into the abstention category in this platform, which the authors call a deep abstaining classifier.

Rather than completely abstaining from training on uncertain points, a model can weigh down the high uncertainty points and weight up low uncertainty points. This approach was demonstrated as reducing both the bias and variance when added as part of the loss function in neural networks [5]. The measure of certainty is calculated by means of comparison to k nearest neighbors (with $k = 5$ in the presented experiment).

4.3 Model Hardening

Some machine learning models inherently more robust to certain classes of noise. One study identified Naive Bayes and KNN as being more robust to label noise than support vector machines and decision trees [40]. More recent work has reached the conclusion that KNN and SVM were comparatively robust to label noise [20]. When considering these two results it is important to remember that there are many varieties of SVM and that the earlier result is experimentally derived while the later work is based on asymptotic analysis. Researchers working with data that are known to have some rate of mislabeling might be best served by sticking to models that are less sensitive to label noise.

As mentioned in Sect. 3, CNNs are an excellent tool for working with time series data. Expectation-Maximization has been applied to CNNs to increase their robustness to label noise and to integrate a model of noise distribution in the training of a CNN [59]. The noise model is learned as the CNN trains.

5 Conclusion

This work summarized the most common label noise detection and correction approaches. Many of these approaches are not exclusively used with time series data, but are general platforms for use with arbitrary, numeric data. A good feature extractor is often sufficient to process time series for use with these general approaches.

Without a standardized testing procedure or data set, it is difficult to say which of the many approaches discussed here are the "best" way to process data with noisy labels. Single classifier learning can incorporate deep feature extractors which makes them a good approach for time series data. For the time it also appears that human review is going to remain an important component of data cleansing.

Analysts who are going to employ human reviewers as part of a label cleaning method should take care to use visualizations that are suited to the data. The type of data and target application should inform the choice of visualization. Segmentation and dimensionality reduction can reveal relationships in data and help produce more interpretable abstractions for reviewers.

Data cleaning comes with some risks. Removing instances from a dataset can increase classifier bias and degrade its accuracy, especially when instances near the borders between classes are removed. Cleaning a training dataset can also harm a classifier when label noise will still be present in the test data. A model that is going to be tested on noisy data will perform better with a robust learning algorithm. But this comes at the risk of knowingly leaving incorrect instances in a dataset.

Intelligent problem-solving methods with pre-collected data are impressive but truly responsive systems require some ability to process online, streaming data. Such data will always have temporal properties, and so intelligent systems (artificial or otherwise) will always have to wrestle with the problems of time series data. Data collected in the real world will always have some noise, label, or otherwise. But with attention and good practices, machine learning with noisy labels is possible and reliable.

References

1. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In: Lomet, D.B. (ed.) FODO 1993. LNCS, vol. 730, pp. 69–84. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57301-1_5
2. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Mach. Learn. **6**(1), 37–66 (1991)
3. Aigner, W., Kainz, C., Ma, R., Miksch, S.: Bertin was right: an empirical evaluation of indexing to compare multivariate time-series data using line plots. In: Computer Graphics Forum, vol. 30, pp. 215–228. Wiley Online Library (2011)
4. Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C.: Visual methods for analyzing time-oriented data. IEEE Trans. Vis. Comput. Graph. **14**(1), 47–60 (2007)
5. Almeida, M., Zhuang, Y., Ding, W., Crouter, S.E., Chen, P.: Mitigating class-boundary label uncertainty to reduce both model bias and variance. ACM Trans. Knowl. Disc. Data (TKDD) **15**(2), 1–18 (2021)
6. Atkinson, G., Metsis, V.: 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, pp. 238–243 (2020)

7. Atkinson, G., Metsis, V.: TSAR: a time series assisted relabeling tool for reducing label noise. In: 14th PErvasive Technologies Related to Assistive Environments Conference (2021)
8. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: bagging, boosting, and variants. Mach. Learn. **36**(1), 105–139 (1999)
9. Bernard, J., et al.: VisInfo: a digital library system for time series research data based on exploratory search—a user-centered design approach. Int. J. Digit. Libr. **16**(1), 37–59 (2014). https://doi.org/10.1007/s00799-014-0134-y
10. Bertin, J.: Semiology of graphics; diagrams networks maps. Technical report (1983)
11. Bingham, E., Gionis, A., Haiminen, N., Hiisilä, H., Mannila, H., Terzi, E.: Segmentation and dimensionality reduction. In: Proceedings of the 2006 SIAM International Conference on Data Mining, pp. 372–383. SIAM (2006)
12. Birjandtalab, J., Pouyan, M.B., Nourani, M.: Nonlinear dimension reduction for EEG-based epileptic seizure detection. In: 2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 595–598. IEEE (2016)
13. Boeva, V., Lundberg, L., Angelova, M., Kohstall, J.: Cluster validation measures for label noise filtering. In: 2018 International Conference on Intelligent Systems (IS), pp. 109–116. IEEE (2018)
14. Bootkrajang, J., Chaijaruwanich, J.: Towards instance-dependent label noise-tolerant classification: a probabilistic approach. Pattern Anal. Appl. **23**(1), 95–111 (2020)
15. Bootkrajang, J., Kabán, A.: Multi-class classification in the presence of labelling errors. In: ESANN, pp. 345–350. Citeseer (2011)
16. Bootkrajang, J., Kabán, A.: Boosting in the presence of label noise. arXiv preprint arXiv:1309.6818 (2013)
17. Bootkrajang, J., Kabán, A.: Classification of mislabelled microarrays using robust sparse logistic regression. Bioinformatics **29**(7), 870–877 (2013)
18. Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data. J. Artif. intell. Res. **11**, 131–167 (1999)
19. Bross, I.: Misclassification in 2 × 2 tables. Biometrics **10**(4), 478–486 (1954)
20. Cannings, T.I., Fan, Y., Samworth, R.J.: Classification with imperfect training labels. Biometrika **107**(2), 311–330 (2020)
21. Cheng, Y., Church, G.M.: Biclustering of expression data. In: ISMB, vol. 8, pp. 93–103 (2000)
22. Chung, F.L., Fu, T.C., Luk, R., Ng, V., et al.: Flexible time series pattern matching based on perceptually important points (2001)
23. Cruciani, F., et al.: Feature learning for human activity recognition using convolutional neural networks. CCF Trans. Pervasive Comput. Interact. **2**(1), 18–32 (2020). https://doi.org/10.1007/s42486-020-00026-2
24. de França, F.O., Coelho, A.L.: A biclustering approach for classification with mislabeled data. Exp. Syst. Appl. **42**(12), 5065–5075 (2015)
25. Frénay, B., Kabán, A., et al.: A comprehensive introduction to label noise. In: ESANN. Citeseer (2014)
26. Frénay, B., Verleysen, M.: Classification in the presence of label noise: a survey. IEEE Trans. Neural Netw. Learn. Syst. **25**(5), 845–869 (2013)
27. Fu, T., Chung, F., Ng, C.: Financial time series segmentation based on specialized binary tree representation. In: DMIN 2006, pp. 26–29 (2006)
28. Ghoniem, M., Shurkhovetskyy, G., Bahey, A., Otjacques, B.: VAFLE: visual analytics of firewall log events. In: Visualization and Data Analysis 2014, vol. 9017, p. 901704. International Society for Optics and Photonics (2014)

29. Gschwandtner, T., et al..: Timecleanser: a visual analytics approach for data cleansing of time-oriented data. In: Proceedings of the 14th International Conference on Knowledge Technologies and Data-Driven Business, pp. 1–8 (2014)
30. Guan, D., Yuan, W.: A survey of mislabeled training data detection techniques for pattern classification. IETE Tech. Rev. **30**(6), 524–530 (2013)
31. Guan, D., Yuan, W., Ma, T., Lee, S.: Detecting potential labeling errors for bioinformatics by multiple voting. Knowl. Based Syst. **66**, 28–35 (2014)
32. Hinton, G., Roweis, S.T.: Stochastic neighbor embedding. In: NIPS, vol. 15, pp. 833–840. Citeseer (2002)
33. Höppner, F.: Time series abstraction methods-a survey. Informatik bewegt: Informatik 2002–32. Jahrestagung der Gesellschaft für Informatik ev (GI) (2002)
34. Jaromczyk, J.W., Toussaint, G.T.: Relative neighborhood graphs and their relatives. Proc. IEEE **80**(9), 1502–1517 (1992)
35. Jolliffe, I.: Principal component analysis. Technometrics **45**(3), 276 (2003)
36. Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A.: Comparing boosting and bagging techniques with noisy and imbalanced data. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **41**(3), 552–568 (2010)
37. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. AIChE J. **37**(2), 233–243 (1991)
38. Li, Y., Cui, W.: Identifying the mislabeled training samples of ECG signals using machine learning. Biomed. Signal Process. Control **47**, 168–176 (2019)
39. Müller, N.M., Markert, K.: Identifying mislabeled instances in classification datasets. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2019)
40. Nettleton, D.F., Orriols-Puig, A., Fornells, A.: A study of the effect of different types of noise on the precision of supervised learning techniques. Artif. Intell. Rev. **33**(4), 275–306 (2010)
41. Nicholson, B., Zhang, J., Sheng, V.S., Wang, Z.: Label noise correction methods. In: 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 1–9. IEEE (2015)
42. Parzen, E., et al.: An approach to time series analysis. Annals of Math. Stat. **32**(4), 951–989 (1961)
43. Pechenizkiy, M., Tsymbal, A., Puuronen, S., Pechenizkiy, O.: Class noise and supervised learning in medical domains: the effect of feature extraction. In: 19th IEEE Symposium on Computer-Based Medical Systems, CBMS 2006, pp. 708–713. IEEE (2006)
44. Rädsch, T., Eckhardt, S., Leiser, F., Pandl, K.D., Thiebes, S., Sunyaev, A.: What your radiologist might be missing: using machine learning to identify mislabeled instances of x-ray images. In: Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS)
45. Sánchez, J.S., Pla, F., Ferri, F.J.: Prototype selection for the nearest neighbour rule through proximity graphs. Pattern Recogn. Lett. **18**(6), 507–513 (1997)
46. Schafer, J.L., Graham, J.W.: Missing data: our view of the state of the art. Psychol. Meth. **7**(2), 147 (2002)
47. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S., et al.: Boosting the margin: a new explanation for the effectiveness of voting methods. Ann. Stat. **26**(5), 1651–1686 (1998)
48. Sheng, V.S., Provost, F., Ipeirotis, P.G.: Get another label? Improving data quality and data mining using multiple, noisy labelers. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 614–622 (2008)

49. Shurkhovetskyy, G., Andrienko, N., Andrienko, G., Fuchs, G.: Data abstraction for visualizing large time series. In: Computer Graphics Forum, vol. 37, pp. 125–144. Wiley Online Library (2018)
50. Silva, S.F., Catarci, T.: Visualization of linear time-oriented data: a survey. In: Proceedings of the 1st International Conference on Web Information Systems Engineering, vol. 1, pp. 310–319. IEEE (2000)
51. Steiger, M., et al.: Visual analysis of time-series similarities for anomaly detection in sensor networks. In: Computer Graphics Forum, vol. 33, pp. 401–410. Wiley Online Library (2014)
52. Stempfel, G., Ralaivola, L.: Learning SVMs from sloppily labeled data. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009. LNCS, vol. 5768, pp. 884–893. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04274-4_91
53. Teng, C.M.: Correcting noisy data. In: ICML, pp. 239–248. Citeseer (1999)
54. Thulasidasan, S., Bhattacharya, T., Bilmes, J., Chennupati, G., Mohd-Yusof, J.: Combating label noise in deep learning using abstention. arXiv preprint arXiv:1905.10964 (2019)
55. Tomek, I., et al.: An experiment with the edited nearest-nieghbor rule (1976)
56. Tüceryan, M., Chorzempa, T.: Relative sensitivity of a family of closest-point graphs in computer vision applications. Pattern Recogn. **24**(5), 361–373 (1991)
57. Venkataraman, S., Metaxas, D., Fradkin, D., Kulikowski, C., Muchnik, I.: Distinguishing mislabeled data from correctly labeled data in classifier design. In: 16th IEEE International Conference on Tools with Artificial Intelligence, pp. 668–672. IEEE (2004)
58. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cybern. **3**, 408–421 (1972)
59. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2691–2699 (2015)
60. Xu, Z., Zhang, R., Kotagiri, R., Parampalli, U.: An adaptive algorithm for online time series segmentation with error bound guarantee. In: Proceedings of the 15th International Conference on Extending Database Technology, pp. 192–203 (2012)
61. Yang, K., Shahabi, C.: A PCA-based similarity measure for multivariate time series. In: Proceedings of the 2nd ACM International Workshop on Multimedia Databases, pp. 65–74 (2004)
62. Yuan, Y., Xun, G., Suo, Q., Jia, K., Zhang, A.: Wave2Vec: learning deep representations for biosignals. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 1159–1164. IEEE (2017)
63. Yuan, Y., Xun, G., Suo, Q., Jia, K., Zhang, A.: Wave2Vec: deep representation learning for clinical temporal data. Neurocomputing **324**, 31–42 (2019)
64. Zeng, X., Martinez, T.R.: An algorithm for correcting mislabeled data. Intell. Data Anal. **5**(6), 491–502 (2001)
65. Zeni, M., Zhang, W., Bignotti, E., Passerini, A., Giunchiglia, F.: Fixing mislabeling by human annotators leveraging conflict resolution and prior knowledge. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 3, no. 1, pp. 1–23 (2019)
66. Zhang, H., Ho, T.B., Zhang, Y., Lin, M.S.: Unsupervised feature extraction for time series clustering using orthogonal wavelet transform. Informatica **30**(3), 305–319 (2006)

67. Zhang, Z., Jiang, J., Wang, H.: A new segmentation algorithm to stock time series based on pip approach. In: 2007 International Conference on Wireless Communications, Networking and Mobile Computing, pp. 5609–5612. IEEE (2007)
68. Zhao, L., Sukthankar, G., Sukthankar, R.: Incremental relabeling for active learning with noisy crowdsourced annotations. In: 2011 IEEE 3rd International Conference on Privacy, Security, Risk and Trust and 2011 IEEE 3rd International Conference on Social Computing, pp. 728–733. IEEE (2011)
69. Zhu, X., Wu, X.: Class noise vs. attribute noise: a quantitative study. Artif. Intell. Rev. **22**(3), 177–210 (2004)
70. Zhu, X., Wu, X., Chen, Q.: Eliminating class noise in large datasets. In: Proceedings of the 20th International Conference on Machine Learning, ICML 2003, pp. 920–927 (2003)
71. Zhu, X., Zhang, P., Lin, X., Shi, Y.: Active learning from stream data using optimal weight classifier ensemble. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **40**(6), 1607–1621 (2010)