Cluster-Clean-Label: An interactive Machine Learning approach for labeling high-dimensional data

David Beil
Aalen University of Applied Sciences
73430 Aalen, Germany
https://orcid.org/0000-0002-6425-7919

Andreas Theissler
Aalen University of Applied Sciences
73430 Aalen, Germany
https://orcid.org/0000-0003-0746-0424

ABSTRACT

One of the major problems of applying supervised machine learning methods in real-world problems is the absence of labeled data. Labeling huge amounts of data is time consuming and cost intensive. Moreover, in many cases, labels can only be assigned by domain experts like medical doctors or engineers, who have little time and do not necessarily have profound machine learning knowledge. In this paper, we propose an efficient interactive cluster-clean-label approach. First, to visualize the potentially huge amount of data, principal component analysis followed by t-SNE projection is applied. On the 2-dimensional representation of the data, HDBSCAN clustering is utilized to identify groups of potentially similar class membership. Subsequently, anomaly detection in form of an autoencoder is applied on each cluster, and instances that are likely to belong to different classes are suggested to the user. The user decides which of these suggested instances to include and restarts the anomaly detection process with the remaining subset of instances. This iterative process is repeated until the user is satisfied with the clusters' purity. Eventually, labels are assigned to the clusters. The approach is evaluated by a user study with 25 participants using the initially unlabeled MNIST data set, where on average users were able to label 91.59% of the data set, with an accuracy of 98.99%. A video showing the approach is available: https://youtu.be/RsLI0dg90qE.

ACM Reference Format:
David Beil and Andreas Theissler. 2020. Cluster-Clean-Label: An interactive Machine Learning approach for labeling high-dimensional data. In *The 13th International Symposium on Visual Information Communication and Interaction (VINCI 2020), December 8–10, 2020, Eindhoven, Netherlands.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3430036.3430060

1 INTRODUCTION

Methods of machine learning, especially methods of supervised learning, have achieved outstanding results in almost all areas of everyday life in recent years. The areas of application are broad and ranging from computer vision [24], predicting user behavior [8], machine translation [18] to fault detection [31]. Buettner et al. [9], for example, achieved great success in the classification of patients with mental disorders by making use of the random forest classifier [9, 25]. In [23] machine learning was used for fraud detection in the banking industry. Moreover, machine learning can be leveraged to make the production process more efficient or reducing the downtime of production systems [29, 38].

For the achievement of such accurate and robust results, methods of supervised learning are used in most cases. These methods follow a two-phase procedure, which consists of a training and a test phase [20]. In the training phase, the model is trained with training data, which is composed of the data points as well as the corresponding labels. During this phase the model learns to recognize the relevant characteristics and to map these to the assigned class labels.

Artificial neural networks are well-known examples of supervised methods achieving outstanding results in many areas [17]. However, to achieve such excellent results, these methods require an enormous amount of labeled training data [5]. This need of labeled training data is one of the major limitations of supervised methods in real-world problems, addressed in e.g. [36].

To label the data, often profound knowledge is required in the respective domain. Depending on the domain and the type of data, labeling of a whole data set can be a very time-consuming task. Considering that domain specialists like medical doctors or engineers are typically rare, it is evident that the classification of a data set is a very cost-intensive process [5, 11].

Motivated by this, a new field of research has developed in recent years. Promising approaches were published under the keywords active learning and interactive learning. Also studies in the area of semi-supervised learning deal with methods combining users and algorithms to achieve accurate results. Most of these approaches aim to make the labeling process more efficient by reducing the number of instances that have to be labeled manually. This subset of instances is subsequently used to train a model, which is then able to support the labeling process by predicting labels or proposing instances that will lead to large information gain for the model. Several of these methods suffer from the so-called cold-start-problem. It describes the situation at the beginning of the task, where no labels are known yet and therefore no predictions or proposals can be made [5]. Also processes and frameworks combining model-based and user-driven techniques have been published in the past [6, 11].

In addition to labeling for classification problems, in [32] the challenges for interactive labeling for supervised anomaly detection are addressed.

Inspired by the previous work, the research question for this paper is: "How can we combine the strengths of humans and machine learning methods to label high-dimensional data sets time-efficiently and highly accurate?".

VINCI 2020, December 8–10, 2020, Eindhoven, Netherlands

ACM ISBN 978-1-4503-8750-7/20/12…$15.00
https://doi.org/10.1145/3430036.3430060

We propose a novel *cluster-clean-label*-method to label high-dimensional data set with an interplay of user and machine learning methods. Starting with unlabeled data, the approach allows to incrementally improve the proportion and accuracy of labeled instances.

Following an initial projection onto two-dimensional space with Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE), the data is clustered to identify groups of instances with potentially similar class membership. The purity of these clusters is incrementally refined using an autoencoder in connection with the user's decision to include or exclude instances from clusters. In a final step, representatives of each clusters are presented to the user allowing the user to efficiently label clusters as classes.

In this work we make the following contributions:

- We propose a novel sequence of machine learning techniques to support the user in the process of labeling a data set.
- Our approach enables domain experts without profound knowledge of ML to efficiently label a data set.
- In a user study, we were able to improve the results on the MNIST data set achieved by Vajda et al. [30], a paper which inspired this work, with an accuracy of 98.99% on the 64,113 (91.59%) labeled instances averaged over the 25 participants.

2 RELATED WORK

The problem of missing labels has been discussed in various papers and solutions have been explored. These approaches can be categorized into (a) model-centric machine learning based approaches and (b) user-centric approaches using visualization techniques for knowledge generation. Model-centric approaches are often ascribed to so-called active learning. Active learning is a special type of incremental machine learning focusing on the incorporation of user knowledge to improve the model accuracy [28]. Therefore, in general, supervised models get trained on a small subset of labeled data. These models are then used to predict labels of the other instances. Additionally, the model suggests instances whose labels could significantly improve the accuracy. The sampling criteria to suggest useful instances vary in the different approaches and can be further categorized into these five groups: uncertainty sampling, query by committee, error reduction schemes, relevance-based selection, and data-centered strategies [21, 33, 37]. In contrast, the user-centered approaches focus on a suitable visualization of underlying data in order to transfer the active selection of the instances to be labeled to the user [26, 35]. High dimensional data therefore often has to be dimensionality reduced. Agis and Pozo showed that by using a sequence of linear and non-linear methods, namely PCA and t-SNE, adequate results can be achieved [2]. In the groundbreaking work [6], Bernard et al. combined both aspects, the visualization and the active learning part, and refers to it as visual interactive labeling (VIAL). Based on this Grimmeisen et al. [13] published an approach for combining strengths of model-based active learning and user-based interactive labeling.

Peikari et al. [22] presented a significantly different way to solve the missing label problem. Instead of focusing on reducing the number of instances to be labeled by a sampling method, they developed a cluster-then-label method. For this purpose, clustering is used to find high density regions in the data space which are subsequently used by a support vector machine to find the decision boundary [22]. Another promising approach, based on clustering, was presented by Vajda et al. [30], achieving an notable accuracy of 96.37% labeling the MNIST data set, based on previously formed clusters. Therefore, they clustered 3 representations of the data set, raw images, PCA reduced data and data reduced with an autoencoder. Labels are manually annotated to each clusters and subsequently an unanimity vote defines the final label of the cluster.

Inspired by the findings of Bernard et al. [5] that multi-selection makes the process of interactive labeling more efficient, and strengthened by the great results achieved by Vajda et al. [30] with cluster-based labeling, we propose our *cluster-clean-label*-approach of cluster-based interactive labeling with an additional cleaning step.

3 APPROACH: CLUSTER-CLEAN-LABEL

The proposed *cluster-clean-label*-approach[1] has the main goal to find pure clusters with similar class memberships and thus enable the efficient assignment of labels based on clusters. The approach is demonstrated and evaluated using the well-known MNIST data set [16]. The steps of the complete labeling pipeline are shown in Fig. 1. First, in case of high dimensionality, the unlabeled data is projected onto a lower dimensional space using the linear method PCA [1].

Subsequently, the PCA-projected data is further projected onto 2-dimensional space using t-SNE [19]. This 2D-data is then clustered with HDBSCAN [10]. Following that, the user selects the desired cluster and starts cleaning (see Fig. 2). Utilizing an autoencoder per cluster, instances that appear to have a different class membership are suggested to the user (see Fig. 5). Following that, the user decides which instances should be kept in the cluster and which ones should be removed. This cleaning step is repeated until the user is convinced of the purity of the cluster.

Within the scope of this work, the developed approach is evaluated by a user study using the MNIST data set. It contains 70,000 labeled images of handwritten digits (0 to 9). Each image (instance) is represented by a 784 dimensional vector, which corresponds to a 28x28 grayscale image [16]. The MNIST data set was found to be predestined for the use in this paper, as it satisfies four prerequisites:

- no specific domain knowledge required for labeling, allowing easy recruitment of participants for user study
- data is represented by images, which is intuitive for users
- labels are available to evaluate the results
- it has been used in related work, allowing for the comparison of results

3.1 Step 1: Cluster

As shown in Fig. 1, the first step of identifying clusters in the data is subdivided into dimensionality reduction and the clustering itself.

A sequence of the linear method PCA and the computationally more expensive t-SNE method is used to project the data onto two dimensions. In recent years, a large number of dimensionality reduction methods have been developed. These can be divided into linear and non-linear methods [34]. Linear methods are often characterized by a significantly lower required computing power.

[1] A video showing the proposed approach is available: https://youtu.be/RsLI0dg90qE

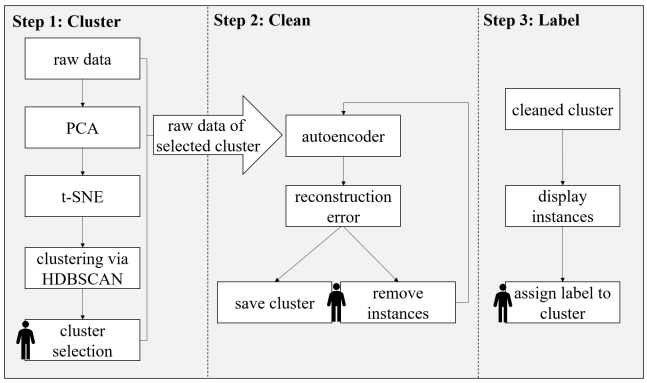

Figure 1: The *cluster-clean-label* approach: In step 1, the high-dimensional data is projected onto low-dimensional space using a PCA and a subsequent t-SNE projection in order to apply the HDBSCAN algorithm to this dimensionality reduced data. In step 2, remaining outliers are removed by the user with the help of autoencoder-based anomaly detection. In the final step, labels are assigned to the cleaned clusters by the user.

Therefore, the runtime is often significantly shorter than those of non-linear methods. For that reason, as a first step the linear method PCA is applied on the data set. The idea of PCA is to extract the most relevant information on the basis of the data's variance and to represent it in a new set of coordinates – the so-called principal components. Using an appropriate number of principal components, high-dimensional data sets can be represented in a significantly lower dimensional space without loss of much information [1]. In this work the threshold was set to 90% explained variance, which yields 86 dimensions for the MNIST data.

This dimensionality reduced data is further projected to two-dimensional space using t-SNE [19]. Agis and Pozo [2] showed that the combination of PCA and t-SNE works well in practice. While linear methods of dimensionality reduction often focus on displaying dissimilar data points in low-dimensional space far apart, for high-dimensional data that lie on or near a low-dimensional nonlinear manifold, it is more important to keep similar points in low-dimensional space close together. Hence, t-SNE allows to represent much of the local structure of high-dimensional data as well as global structures such as the presence of clusters. Laurens van der Maaten et al. [19] also state that t-SNE is better at creating a single map that reveals structure at many different scale, than the previously proposed SNE [14].

In Fig. 3 the projections resulting from the sequence of PCA and t-SNE applied on the MNIST data set are shown. In Fig. 3a the instances are colored according to their class labels, which are not available to the user in the process of labeling.

Following the projection, the next step is to enable the user to select the desired instances. Bernard et al. showed in [5] that the selection of several instances makes the assignment of labels much more efficient. Many approaches use the so-called lasso-selector to select the instances. For data sets with a high number of instances, however, clusters may blur into each other making manual selection of instances a difficult task. Also, the required number of interactions is much higher by selecting instances with the lasso-selector than it would be by selecting entire clusters. For that reason, to facilitate the selection of instances, a clustering algorithm is applied on the projected data. In Fig. 3b the projected and clustered data is depicted the way it is presented to the user. The clusters are encoded by colors showing that the identified clusters resemble the original class labels. In addition, instances that were not assigned to clusters by the HDBSCAN clustering algorithm are visible.

While many former studies use the well-known k-means clustering algorithms, in this study we have chosen the HDBSCAN algorithm. The projection in Fig. 3a shows the presence of differently dense areas in the projected data. To take this property of the data into account, a density-based clustering algorithm is used in this approach. In contrast to k-means, where the number of desired clusters must be known a priori, density-based methods divide the data set into areas of high density regions separated by areas of low density, without requiring a previously determined number of clusters. One of the most cited approaches in this area is the DBSCAN algorithm proposed by Ester et al. [12] which is suitable for data sets with clusters of arbitrary shapes. However, clusters with large differences in their densities cannot be detected. Either way, as shown in Fig. 3, the densities in the projected data space may differ, which makes the use of the DBSCAN algorithm only possible to a limited extent. To overcome this limitation, hierarchical density-based spatial clustering of applications with noise (HDBSCAN) was presented by Campello, Oulavi and Sander [10], enhancing DBSCAN [12] to hierarchical clustering. The different densities of the areas are determined, and the most relevant clusters are identified via the hierarchical component. This makes the algorithm suitable for use on data sets with clusters of very different densities. In addition, HDBSCAN shares the DBSCAN's characteristic that instances can be classified as noise and thus not assigned to a cluster. Another advantage of HDBSCAN is that the clustering result is predominantly influenced by one parameter (the minimum cluster size), keeping the parameterization effort relatively low. The parameter determines the minimum number of instances in a cluster. The higher the number, the more instances are potentially detected as noise. The parameter value is set based on the characteristics of the specific data set. The remaining parameters were kept at their default settings. For the MNIST data set good results are achieved with a minimum cluster size of 99, which can be seen in Fig. 3b.

3.2 Step 2: Clean

The ideal case would be a clustering result with each cluster solely containing instances of similar class membership. Except for very simple data sets, this is not likely to happen. In most cases the resulting clusters will contain instances of more than one class, where one class occurs predominantly in each cluster. As shown for the MNIST data in Fig. 3b, although the clustering is very accurate, it is not sufficient to label on this basis. In order to assign labels on a cluster basis, it is essential to further increase the purity of the clusters. In this step of the labeling process, the user is incorporated to fulfill the task of data cleaning in collaboration with the autoencoder, due to the ability of humans to intuitively recognize patterns. However,

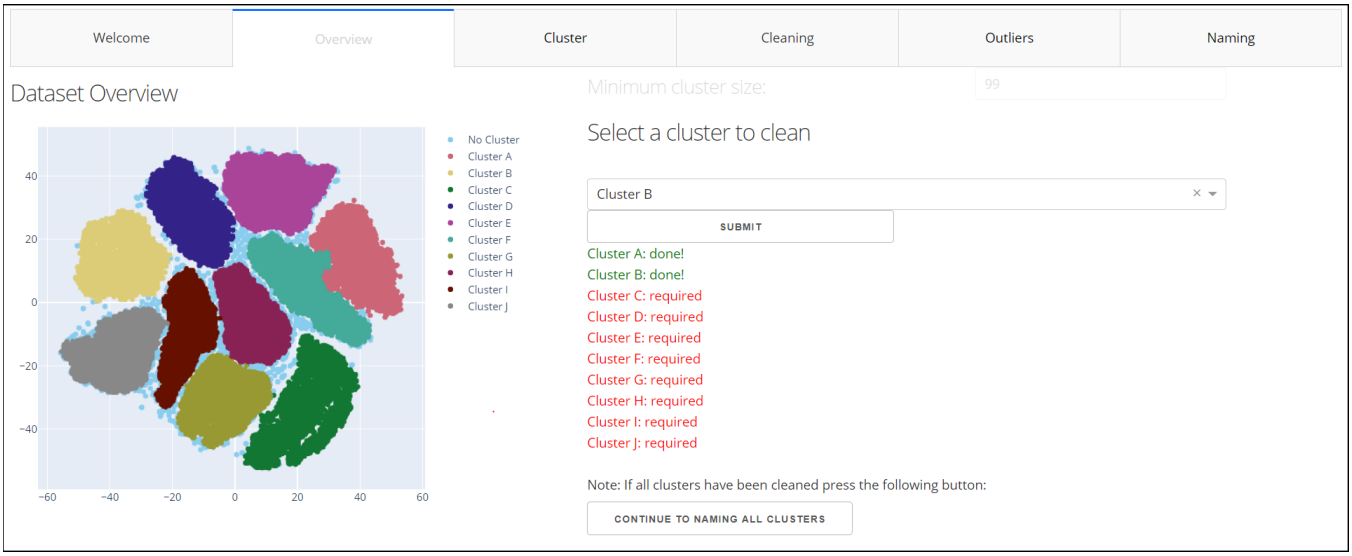

Figure 2: Overview screen showing the clustered 2D-protection of the entire data set and dropdown menu for cluster selection.

an instance-based removal of incorrectly assigned instances would be infeasible due to the high number of data points. For example, for the MNIST data set 6,000 to 7,000 instances are present in a cluster. To reduce the selection effort unsupervised anomaly detection is applied on each cluster. One promising approach is the so-called autoencoder, which is an unsupervised variant of artificial neural networks. An autoencoder consists of several layers and is trained such that the input vectors are reproduced at the output layer with the lowest possible reconstruction error [4]. In principle, the structure can be described as two funnels placed next to each other, where in the encoding part the number of nodes decreases with each layer and in the decoding part increases with each layer. In the middle of the funnels the so-called code is created, which is a dimensionality-reduced representation of the input data [15].

One autoencoder is trained per cluster. The autoencoder will thereby model the predominant class, for instances of other classes the reconstruction error between input and the output is significantly higher. To ensure that the entire information content of the data is taken into account, instead of the reduced data, the raw high dimensional data is used as input (784 dimensions for the MNIST data set). Since the aim is to assign a label to each cluster, the 'normal' data is associated with the predominately occurring digit and the 'abnormal' data thus is all instances of other digits in the cluster. Due to cluster impurity, it is not likely to train the autoencoder solely on instances from one class. However, since the proportion of instances of the predominant class in a cluster is significantly higher than the number of instances from other classes, it can reasonably be assumed that the autoencoder is optimized for the reproduction of the predominant class [3]. Based on this consideration, autoencoders seem to be predestined for this approach.

Fig. 4 shows examples of inputs and outputs from an autoencoder trained on the cluster predominantly containing digit 4. It can be seen that digit 4 is reproduced with a low reconstruction error, while digits other than 4 result in outputs with higher errors.

As the reconstruction error, the mean squared error (MSE) is used, given by

$$MSE(y, \hat{y}) = \frac{1}{d} \sum_{i=1}^{d} (y_i - \hat{y}_i)^2 \tag{1}$$

where y is one instance at the input layer, $\hat{y}$ the output, and d the number of dimensions.

The reconstruction error can subsequently be used to detect instances, that were potentially incorrectly assigned to the cluster. However, also instances that are correctly mapped may have increased errors, if they have different characteristics. An example in the MNIST data set are digits written in a different or unreadable way. Thus, a purely mathematical distinction is not possible for all instances. The idea is to incorporate the user into the process for those instances with unclear class membership which are determined by the autoencoder's reconstruction error given by eq. (1). These instances are displayed to the user, the decision whether to remove an instance is taken by the user. Since clusters may contain a larger number of incorrect assigned instances, this step is repeated until the user is satisfied. Instances can be removed by setting a threshold or by selecting specific instances from the displayed instances with the highest MSE.

The user controls the trade-off between pure clusters and a high proportion of labeled data. By simply lowering the threshold, the data can be labeled "conservatively" by removing more instances, resulting in pure clusters but a higher number of unlabeled instances. To support the selection of the threshold, the reconstruction error is plotted against the instances. By ordering the instances w.r.t. to the MSE in an ascending manner, the graph shows an exponential course, which is predestined for an intuitive selection of the threshold. To keep the task easy, selection is possible directly by clicking on the graph. More refined adjustments are made by instance-based

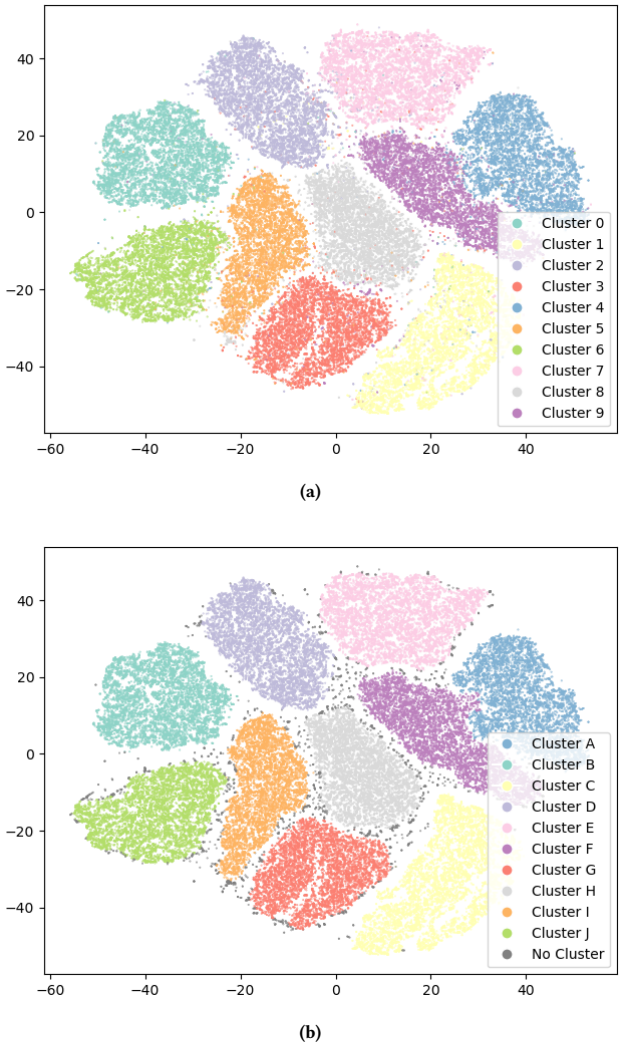

Figure 3: MNIST data set projected to 2-dimensional space with PCA and subsequent t-SNE: (a) colored according to original label, and (b) colored according to HDBSCAN clustering result.

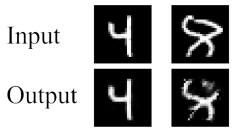

Figure 4: Examples of inputs and outputs of an autoencoder trained on the MINST data on the cluster predominantly containing digit 4.

selection in the set of suggested instances. Therefore, the 100 instances (images for MNIST) with the highest MSE are displayed (see Fig. 5, right), where instance-based selection is possible by clicking on the corresponding image. Fig. 5 shows the implemented screens for the selection process. Summarized, the cleaning step is composed of the following sequence of steps:

(1) an autoencoder is trained and creates reconstructed outputs
(2) for each instance, the mean squared error (MSE) between input and output is calculated and the instances are sorted w.r.t. the MSE in ascending order
(3) the error curve is plotted w.r.t. the sorted instances and the 100 instances with highest MSE are displayed (see Fig. 5)
(4) the user decides which instances should be removed from the cluster based on a threshold or on instance-based selection from the suggested candidates

3.3 Step 3: Label

After iterative cleaning of the individual clusters by an interplay of user and machine learning models, the impurity in the clusters has been reduced.

This leads to the fact that labels can be assigned to whole clusters. Thereby it is possible to label a huge amount of instances at a time with a high accuracy. To let the user determine the class of a cluster, five instances per cluster are displayed as representatives, enabling the user to assign a label.

4 EVALUATION: USER STUDY

In order to evaluate the developed tool and to get further insights for improving the approach, we conducted a user study. As the developed tool aims to integrate the user into the labeling process, it is essential that it is tested and evaluated by real users.

4.1 Participants

We recruited 25 participants for the user study. The age of the participants ranged from 21 to 66. The average age was 31.5 years with a standard deviation of 13.7. The group was composed of 11 women and 14 men. All participants received a short introduction into the topic of this work. Thereby, the concept of labeling was explained. After this introduction, the tool was demonstrated in an exemplary way to show the functionality. Special attention was paid not to give any recommendations for actions and to limit the introduction to the functionality of the tool. The participants had no machine learning knowledge, used the tool for the first time and had never worked with the MNIST data set. In addition, care was taken to ensure that the test persons are sufficiently familiar with the use of computer programs so that biased results due to general operating errors can be excluded.

4.2 Task

For the realization of a user study it was necessary to break down the tasks into the essential components. Since the main focus of this work is the human-machine interaction in the area of cluster cleaning, the projection onto 2-dimensional space as well as the clustering algorithm were applied in advance. For this purpose, the clustering parameters were set such that the high-density regions are mapped. Accordingly, each participant started in the overview

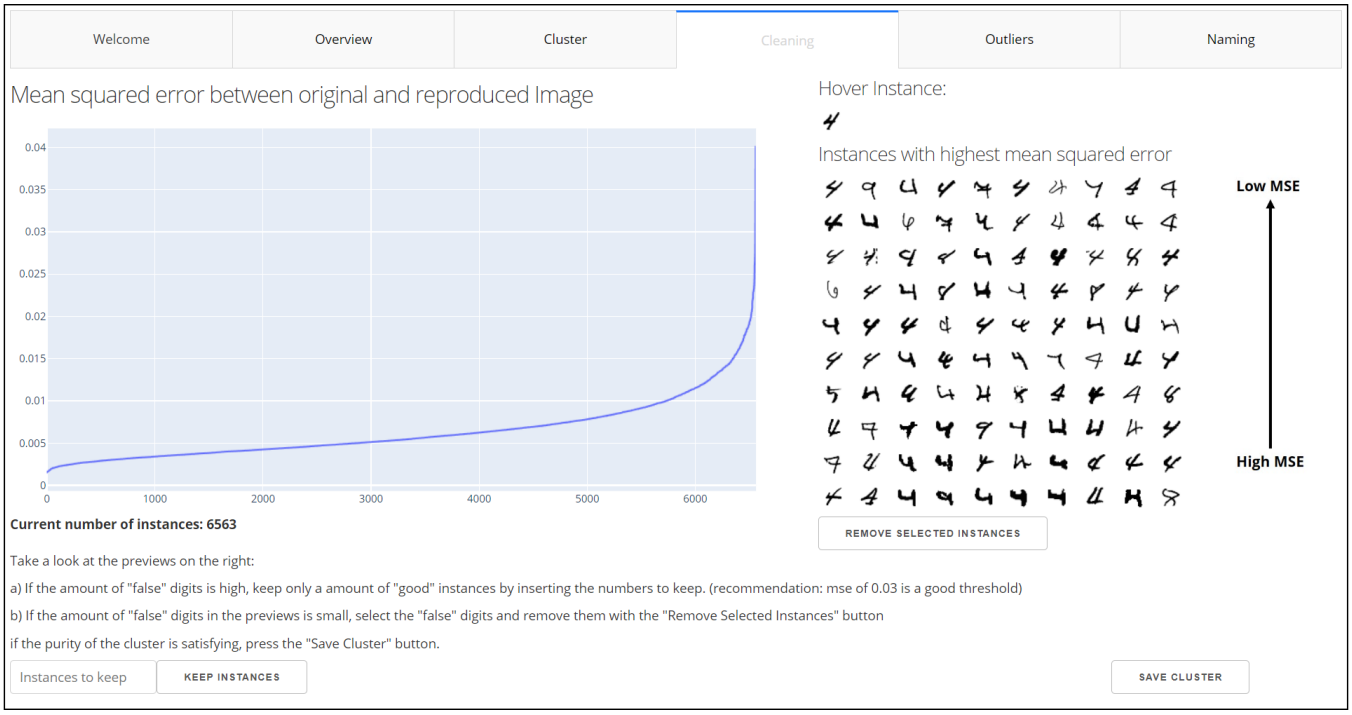

Figure 5: Cleaning screen: On the left the MSE is plotted over number of instances. On the right 100 instances with highest MSE are displayed.

screen shown in Fig. 2. The participants were then given the task to label the entire MNIST data set, where the starting point was the initial clustering shown in the 2D-projection. Instances not belonging to the cluster were to be removed. Users had the choice to select individual instances or to set a threshold in each run. Participants were told that they can stop the cleaning process for each cluster once they are satisfied with the cluster purity. This should correspond to a real world scenario in which the decision about sufficient purity is most likely made by the domain experts. No time limit was set in order not to influence the results by putting the participant under pressure. The elapsed processing time was, however, measured. All participants started with the same initial state. After having cleaned all clusters, the participants assigned labels to the individual clusters. Based on the resulting labeled data set the accuracy was calculated.

4.3 Experimental Results

For a data set D, the subset of instances labeled by the user is denoted as L, the unlabeled instances as U. To evaluate the approach, the class labels – which are obviously not available in a real-world scenario – are used to determine the number of correctly labeled instances denoted as C, i.e. $C \subseteq L \subseteq D$. We denote $|D|$, $|L|$, $|U|$, and $|C|$ as the respective number of instances.

We introduce the proportion of labeled data points as

$$prop_{labeled} = \frac{|L|}{|D|} \tag{2}$$

and the proportion of correctly labeled instances as

parameters	value
participants in user study	25
instances to be labeled $\|D\|$	70,000
labeled instances $\|L\|$	64,113
$prop_{labeled}$	91.59%
$acc_{labeled}$	98.99%
standard deviation of $acc_{labeled}$	0.0016

Table 1: Results on MNIST data set averaged over the 25 participants of the user study.

$$acc_{labeled} = \frac{|C|}{|L|} \tag{3}$$

which is used as the measure for the achieved accuracy.

Table 1 shows the achieved results averaged over all participants. On average $|L| = 64,113$ instances were labeled during the user study, which corresponds to $prop_{labeled} = 91.59\%$ of the whole data set D with 70,000 instances. An accuracy of $acc_{labeled} = 98.99\%$ was achieved, i.e. the vast majority of the labeled instances L corresponded to the true class label.

In Fig. 6 the results of all participants are shown. It is noticeable that the variance of the achieved accuracy of the cluster containing mainly digit 9 is significantly higher compared to the other clusters. Moreover, in this cluster the mean achieved accuracy is lower.

The results show that we are able to improve state-of-the-art approaches. Vajda et al. [30] for instance reached an accuracy of 96.37% for 54.76% of the training data of the MNIST data set in their

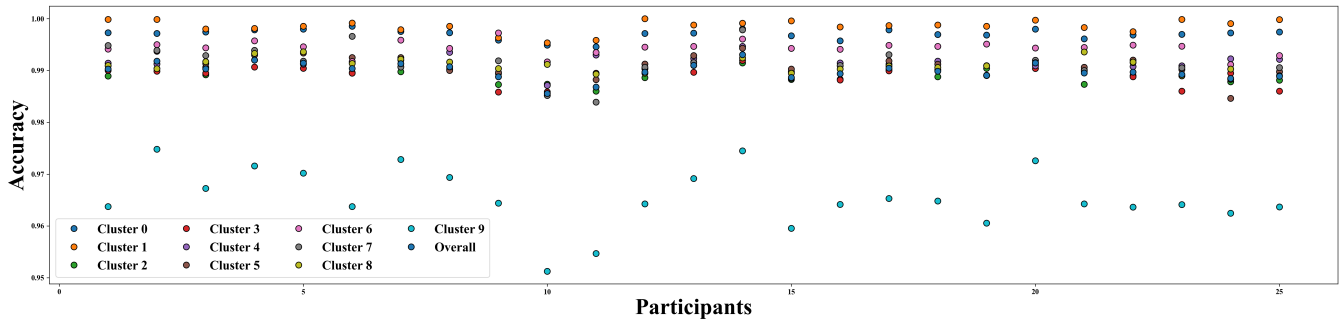

Figure 6: Accuracy of labeled instances over all clusters.

interactive labeling approach. Nevertheless, even this smaller subset was enough to train a k-nearest neighbor classifier that achieved an accuracy of 94.81% on test data. Moreover, the conducted user study showed that even users with almost no machine learning knowledge were able to achieve notable results with our approach. Especially in real-world problems this is a point not to be neglected. To interpret data of real-world problems, the knowledge of domain specialists is often required. However, they do not always have expertise to handle complex machine learning software.

A second insight is that the performance of the newly developed method varies over the clusters. Table 2 shows the average accuracy and standard deviation for each cluster. Approximately 3.5% of the instances labeled as digit 9 are in fact different digits. This can be explained by the similarity of digit 9 to many others like 4. Nevertheless, the performance over all clusters is very promising and was efficiently achieved within an averaged processing time of 1 hour and 42 minutes (median: 1 h 38 min, Q1: 1 h 16 min, Q3: 2 h).

4.4 Usability score

To evaluate the usability of the developed tool, we conducted a user survey among the participants of the user study, where 21 of the participants responded. We used the System Usability Scale (SUS) proposed by Brooke [7], which is a questionnaire comprising 10 pre-defined questions. As a result, the systems usability can be represented by a standardized evaluation procedure. The SUS score is in the range of 0..100, where 100 represents perfect usability. The mean score, which demonstrate an acceptable level of usability, is reported to be 68 [27]. The survey yielded an SUS score of 81.1 for our tool, which according to [27] can be mapped to the letter grade *A* or the adjective *excellent*.

4.5 Discussion

The results achieved during the user study clearly show the possibility of labeling a data set with the developed tool, even without having a profound knowledge of machine learning. In addition, we have also gained insights that require further discussion. For example, we revealed that the performance of the developed application varies over the classes, in the used data set the different digits. In the user study, cluster cleaning performed most ineffectively for the cluster that predominantly contains digit 9. Considering the images available in the data set, intuitive similarities between digits 9 and 4 can often be recognized. A comparable assessment of the difficulties in distinguishing between digits 9 and 4, has already been published in [30]. Due to this similarity, among the instances that were suggested by the autoencoder by means of a low reconstruction error (eq. (1)) as not belonging to the class, untypical instances of digit 9 were present together with some instances of digits 4 and others. In order to investigate this problem further, in future work various methods of anomaly detection will be tested and the autoencoder used will be further improved.

Furthermore, while conducting the user study we noticed that participants tended to remove instances from the cluster, which should have remained in cluster. On the other hand, instances which were suggested as not belonging to the cluster were left in the cluster, although they actually did not match the later assigned class label. In a later informal interview, many participants stated that they had difficulties in reaching a decision due to the unclear handwriting of the respective MNIST digits. This shows that the measured accuracy not only measures the effectiveness of the anomaly detection, but also strongly depends on the individual perception and the accuracy of the participants.

It should be noted that not the entire data set was labeled, which is consistent to the results reported in [30]. In the user study 8.41% of the instances remained unlabeled. However, those instances can be labeled using label propagation e.g. with supervised models as shown in [30].

5 CONCLUSION

In this paper the novel approach *cluster-clean-label* to interactively label high-dimensional data was proposed. For efficient labeling, we combined the strengths of humans and unsupervised machine learning methods. We reduced the dimensionality of the data to two dimensions using a PCA and a t-SNE projection. By applying the HDBSCAN clustering algorithm clusters of instances with potentially similar class membership are found. The user is supported in cleaning the clusters by an anomaly detection algorithm based on autoencoders. In a final step, the user labels a huge amount of data by assigning labels to the cleaned cluster. We evaluated the approach with a user study where an average of 91.59% of the instances were labeled with an accuracy of 98.99%.

In future work we plan to evaluate the user interactions taking place when operating the tool and the evolvement of accuracies and proportions of labeled data.

Cluster	0	1	2	3	4	5	6	7	8	9
$acc_{labeled}$	99.70%	99.86%	98.93%	98.93%	99.13%	99.09%	99.44%	99.12%	99.11%	96.55%
standard deviation	0.0010	0.0012	0.0016	0.0017	0.0016	0.0023	0.0014	0.0030	0.0013	0.0057

Table 2: Accuracies per cluster averaged over all 25 participants.

Additionally, there is optimization potential by improving the accuracy of the cleaning step. Promising solutions could be deeper or alternative network architectures which, however, need to be computationally efficient enough in order not to slow down the interactive process. Furthermore, machine learning models could be used to classify the remaining set of unlabeled instances based on the labeled data and hence increase the total amount of labeled data. Moreover, the approach of Vajda et al. could be implemented to increase comparability and to measure the processing time. Although we suspect that the approach is at least applicable to clusterable image data, which can be displayed in thumbnail view, and to a subsequence of time series data, the approach should be tested on real-world data sets in future work. This would increase the external validity of our results and also verify the applicability in real-world scenarios.

REFERENCES

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[2] David Agis and Francesc Pozo. A Frequency-Based Approach for the Detection and Classification of Structural Changes Using t-SNE. *Sensors*, 19(23):5097, 2019.

[3] Jinwon An and Sungzoon Cho. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. *Special Lecture on IE*, 2(1), 2015.

[4] Pierre Baldi. Autoencoders, Unsupervised Learning, and Deep Architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, Bellevue, Washington, USA, 2012. PMLR.

[5] Jürgen Bernard, Marco Hutter, Matthias Zeppelzauer, Dieter Fellner, and Michael Sedlmair. Comparing Visual-Interactive Labeling with Active Learning: An Experimental Study. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):298–308, 2018.

[6] Jürgen Bernard, Matthias Zeppelzauer, Michael Sedlmair, and Wolfgang Aigner. VIAL: a unified process for visual interactive labeling. *The Visual Computer*, 34(9):1189–1207, 2018.

[7] John Brooke. SUS: a "quick and dirty" usability. *Usability evaluation in industry*, page 189, 1996.

[8] Ricardo Buettner. Predicting user behavior in electronic markets based on personality-mining in large online social networks. *Electronic Markets*, 27(3):247–265, 2017.

[9] Ricardo Buettner, Annika Grimmeisen, and Anne Gotschlich. High-performance diagnosis of sleep disorders: a novel, accurate and fast machine learning approach using electroencephalographic data. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.

[10] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-Based Clustering Based on Hierarchical Density Estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 160–172. Springer, 2013.

[11] Mohammad Chegini, Jürgen Bernard, Philip Berger, Alexei Sourin, Keith Andrews, and Tobias Schreck. Interactive labelling of a multivariate dataset for supervised machine learning using linked visualisations, clustering, and active learning. *Visual Informatics*, 3(1):9–17, 2019.

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, volume 96, pages 226–231, 1996.

[13] Benedikt Grimmeisen and Andreas Theissler. The Machine Learning Model as a Guide: Pointing Users to Interesting Instances for Labeling through Visual Cues. In *The 13th International Symposium on Visual Information Communication and Interaction (VINCI 2020), December 8–10,2020, Eindhoven, Netherlands*. ACM, 2020.

[14] Geoffrey E Hinton and Sam T Roweis. Stochastic Neighbor Embedding. In *Advances in Neural Information Processing Systems*, pages 857–864, 2003.

[15] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, 2006.

[16] Yann LeCun. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

[17] Yann LeCun, Y Bengio, and Geoffrey Hinton. Deep Learning. *Nature*, 521:436–44, 05 2015.

[18] Yang Liu and Jiajun Zhang. *Deep Learning in Machine Translation*, pages 147–183. Springer Singapore, Singapore, 2018.

[19] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[20] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[21] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. 2009.

[22] Mohammad Peikari, Sherine Salama, Sharon Nofech-Mozes, and Anne L. Martel. A Cluster-then-label Semi-supervised Learning Approach for Pathology Image Classification. *Scientific Reports*, 8(1), 2018.

[23] S Benson Edwin Raj and A Annie Portia. Analysis on credit card fraud detection methods. In *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, pages 152–156. IEEE, 2011.

[24] Waseem Rawat and Zenghui Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29:1–98, 06 2017.

[25] Thilo Rieg, Janek Frick, Marius Hitzler, and Ricardo Buettner. High-performance detection of alcoholism by unfolding the amalgamated EEG spectra using the Random Forests method. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[26] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A Keim. Knowledge Generation Model for Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1604–1613, 2014.

[27] Jeff Sauro and James R. Lewis. Chapter 8 - standardized usability questionnaires. In Jeff Sauro and James R. Lewis, editors, *Quantifying the User Experience (Second Edition)*, pages 185 – 248. Morgan Kaufmann, Boston, second edition edition, 2016.

[28] Burr Settles. Active Learning Literature Survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[29] Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820, 2014.

[30] Akmal Szil´rd Vajda, Junaidi and Gernot A. Fink. A Semi-supervised Ensemble Learning Approach for Character Labeling with Minimal Human Effort. In *2011 International Conference on Document Analysis and Recognition*, pages 259–263. IEEE, 2011.

[31] Andreas Theissler. Detecting Known and Unknown Faults in Automotive Systems Using Ensemble-based Anomaly Detection. *Knowledge-Based Systems*, 123(C):163–173, May 2017.

[32] Andreas Theissler, Anna-Lena Kraft, Max Rudeck, and Fabian Erlenbusch. VIAL-AD: Visual Interactive Labelling for Anomaly Detection – An approach and open research questions. In *4th International Workshop on Interactive Adaptive Learning (IAL2020)*. CEUR-WS, 2020.

[33] Devis Tuia, Michele Volpi, Loris Copa, Mikhail Kanevski, and Jordi Munoz-Mari. A Survey of Active Learning Algorithms for Supervised Remote Sensing Image Classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):606–617, 2011.

[34] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality Reduction: A Comparative Review. *J Mach Learn Res*, 10(66-71):13, 2009.

[35] Jarke J Van Wijk. The value of visualization. In *VIS 05. IEEE Visualization, 2005.*, pages 79–86. IEEE, 2005.

[36] D. Wang and Y. Shang. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 112–119, 2014.

[37] Meng Wang and Xian-Sheng Hua. Active Learning in Multimedia Annotation and Retrieval: A Survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(2):1–21, 2011.

[38] Thorsten Wuest, Christopher Irgens, and Klaus-Dieter Thoben. An approach to monitoring quality in manufacturing using supervised machine learning on product state data. *Journal of Intelligent Manufacturing*, 25(5):1167–1180, 2014.