
Validating One-Class Active Learning with User Studies – a Prototype and Open Challenges

Holger Trittenbach, Adrian Englhardt, and Klemens Böhm

Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
{holger.trittenbach, adrian.englhardt, klemens.boehm}@kit.edu

Abstract. Active learning with one-class classifiers involves users in the detection of outliers. The evaluation of one-class active learning typically relies on user feedback that is simulated, based on benchmark data. This is because validations with real users are elaborate. They require the design and implementation of an interactive learning system. But without such a validation, it is unclear whether the value proposition of active learning does materialize when it comes to an actual detection of outliers. User studies are necessary to find out when users can indeed provide feedback. In this article, we describe important characteristics and prerequisites of one-class active learning for outlier detection, and how they influence the design of interactive systems. We propose a reference architecture of a one-class active learning system. We then describe design alternatives regarding such a system and discuss conceptual and technical challenges. We conclude with a roadmap towards validating one-class active learning with user studies.

Keywords: Active learning · One-class classification · Outlier detection · User study.

1 Introduction

Active Learning is the paradigm to involve users in classifier training, to improve classification results by means of feedback. An important application of active learning is outlier detection. Here, one-class classifiers learn to discern inliers from outliers. Examples are network security [17, 37] or fault monitoring [45]. In these cases, one-class classifiers with active learning (OCAL) ask users to provide a binary label (“inlier” or “outlier”) for some of the observations. Then they use these labels in subsequent training iterations to learn an accurate decision boundary. OCAL differs from other active learning applications such as balanced binary or multi-class classification. This is because the strategies to select observations for feedback (“query strategies”) take into account that outliers are rare to non-existent.

Over the last years, several OCAL-specific query strategies have been proposed. They focus on improving classification accuracy with minimal feedback [3, 15, 16, 18, 45]. To evaluate them experimentally, authors generally rely on data sets with an available ground truth, in order to simulate user feedback. On the

© 2019 for this paper by its authors. Use permitted under CC BY 4.0.

one hand, this seems to be convenient, since it allows to evaluate algorithmic improvements without a user study. On the other hand, simulating user feedback requires some assumptions on how users give feedback. These assumptions are generally implicit, and authors do not elaborate on them, since they have become widely accepted in the research community. In particular, there are two fundamental *assumptions behind active learning*:

- (Feedback) Users provide accurate feedback independently from the presentation of the classification result and from the observation selected for feedback (the “query”).
- (Acceptance) Users do not question how the feedback provided changes the classification results. Their motivation to provide feedback, even for many observations, does not hinge on their understanding of how their feedback influences the classifier.

Think of an OCAL system that asks a user to provide a class label on a 20-dimensional real-valued vector, where features are the result of some pre-processing, such as principal component analysis. We argue that, without further information, users are unlikely to comprehend the query, and cannot provide accurate feedback. This is in line with observations from literature [14, 33]. Even if they could provide the feedback, any change in the classifier is not tangible. This is because there is no suitable visualization or description of a 20-dim decision boundary. We argue that users may question whether their feedback has a positive effect on the classifier, or even any effect at all, lose interest, and eventually discontinue to provide feedback.

This is a peculiar situation: On the one hand, the value proposition of active learning is to obtain helpful information from users that is not yet contained in the training data. On the other hand, there currently is no validation whether one can realize this value in an actual application. Such a validation would require to implement an interactive OCAL system and to conduct a user study.

However, such an experimental validation is difficult, since there are several conceptual and technical issues. We have experienced this first hand, when we have looked at smart meter data of an industrial production side [7, 42] to identify unusual observations, and to collect respective ground truth labels from human experts. In our preliminary experiments with this data, we found that both *active-learning assumptions do not hold* in practice. In particular, we have observed that domain experts ask for additional information such as visualizations and explanations that go way beyond a simple presentation of classification results. Since there is an over-reliance on active-learning assumptions, only little effort has been spent on making OCAL interpretable, comprehensible, and usable. So it is unclear what the minimal requirements behind an OCAL system are to carry out a user study. Second, there are *conceptual issues* that are in the way of implementing OCAL systems. One issue is that the design space of OCAL systems is huge. It requires to define a learning scenario, to choose a suitable classifier and a learning strategy, as well as selecting multiple hyperparameter values. In addition, there may be several conflicting objectives: One may strive to improve classification accuracy. Another objective may be to use OCAL

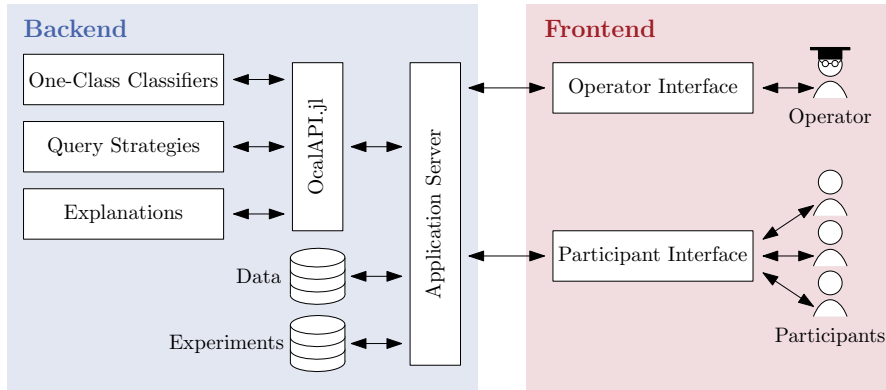


Fig. 1. System overview

as an exploratory tool, to present users with as many interesting instances as possible. Another issue is that objectives of a user study are diverse. One may want to collect a reliable ground truth for a novel data set, or to evaluate specific components of the active learning system, e.g., how well users respond to a particular visualization. Third, there are *technical issues* which we have experienced first hand when implementing an OCAL system prototype. For instance, runtimes of training state-of-the-art classifiers may be too long for interactivity. Another example is that it is unclear how to visualize decision boundaries in multi-dimensional data sets.

Although there are many difficulties, we deem user studies imperative to understand the determining factors behind realizing the value of OCAL. These factors can serve as guidelines for data mining research and can eventually lead to a more differentiated evaluation of novel query strategies and classifiers. The objective of our article is to point out important characteristics and prerequisites of OCAL and how they influence the design of interactive systems. To our knowledge, this is the first overview on conceptual and technical challenges regarding OCAL systems. We derive these challenges based on an architectural sketch on the components of an existing OCAL system, which we have implemented as a prototype. We conclude by proposing a roadmap towards validating OCAL with user studies.

2 OCAL System Architecture

The purpose of an OCAL system is to facilitate experiments with several users. An experiment is a specific technical configuration, i.e., a data set, a classifier, a query strategy, and one or more users, the participants, who provide feedback.

An OCAL system consists of several modules. Participants interact with the system through a *participant interface* that visualizes information on active

learning iterations, such as the classification result and the progress of the experiment. The training of the classifier, query selection, and the preparation of additional information such as visualizations and explanations take place in an *algorithm backend*. Finally, there is a human operator who configures, monitors and evaluates the experiments through an *operator interface*. This typically is the researcher who conducts the experiments. Figure 1 is an overview of the system architecture. In the following, we describe the different modules and link them to our prototype implementation.

Algorithm Backend: On a technical level, the algorithm backend consists of a classifier module *SVDD.jl*¹ and a module *OneClassActiveLearning.jl*², which implements active learning components such as the query strategies. A third module provides additional information, e.g., classifier visualizations. For our prototype, we have implemented the classifiers, query strategies and basic visualization information in *OcalAPI.jl*³, a ready-to-use JSON REST API. This decoupling allows to re-use the algorithm backend independent of the participant and operator interface.

Operator Interface: The operator interface allows an operator to configure so-called experiment *setups*. A setup consists of a data set, a parameterized classifier and a query strategy. Depending on the research question, the operator may also configure which information is displayed in the participant interface. This gives way to A/B tests, to, say, validate if a certain visualization has an effect on feedback quality. The operator can invite several users to participate in an *experiment run*, i.e., an instantiation of an experiment setup. He can monitor and inspect the experiment runs in an overview panel and export experiment data for further analysis.

Participant Interface: The participant interface has two functions. First, it is an input device to collect feedback during the experiment. Second, it provides the participants with information that supports them to provide educated feedback. For instance, this may be a visualization of a classifier, a view on the raw data or a history of classification accuracy over the past iterations. The participant then provides feedback for some observations. During this process, the interface captures user interactions, e.g., mouse movement and selection. When the query budget or time limit is not exhausted, the participant proceeds with the next iteration.

Our implementation of the interfaces is preliminary, since there are several open challenges, both conceptual and technical (see Section 3). We plan to make

¹ <https://github.com/englhardt/SVDD.jl>

² <https://github.com/englhardt/OneClassActiveLearning.jl>

³ <https://github.com/englhardt/OcalAPI.jl>

it publicly available in the future as well. An important takeaway from this section is an intuition about how OCAL systems can be designed, on an architectural level. This intuition may be useful to understand the following discussions on the design space of OCAL systems and the challenges related to the three modules.

3 Design Decisions for OCAL Systems

The design and implementation of OCAL systems are inherently interdisciplinary and require expertise from several areas, including data mining, human-computer interaction, UX-design, and knowledge of the application domain. Although all disciplines are important, we now focus on the data mining perspective. We first discuss different types of interaction and elaborate on the design options for one-class classifiers and query strategies. We then present different options to prepare information for users during the learning iterations. Finally, we elaborate on several technical challenges when implementing OCAL systems.

3.1 Type of Interaction

The common definition of active learning is that a query strategy selects one or more observations for feedback. So, strictly speaking, a user does not have the option to also give feedback on other observations not selected by the system. However, there are related disciplines that do away with this restriction. For instance, one research direction is Visual Interactive Analytics (VIA) [8, 25, 43], where a user interactively explores outliers in a data set. VIA systems provide different kinds of visualization to assist users in identifying outliers, in particular with high-dimensional data sets. The unification of active learning and VIA is Visual Inter-Active Labeling (VIAL) [6, 27]. VIAL combines active learning with user-supporting visualizations from the VIA community. Variants of VIAL and active learning are conceivable as well. For instance, instead of asking for labels of specific observations, the query strategy could provide a set of observations from which users can select one or more to label.

It is an open question in which cases one should use VIAL or active learning. A user study in [5] indicates that users label more observations if they are free to choose the observations. However, the resulting classifier accuracy is higher with an AL query strategy. It is unclear whether these insights transfer to outlier detection where classes are unbalanced. In fact, we see this as one of the overarching questions to answer with user studies.

3.2 Type of Feedback

In this article, feedback is binary, i.e., users decide whether an observation belongs to the inlier or outlier class. However, other types of feedback are conceivable as well. For instance, in multi-class settings, the system may ask users to state to which classes an observation does *not* belong [10]. Another example is to

ask users for feedback on features, as opposed to instances [13]. Existing OCAL approaches in turn focus on binary feedback. It is an open question if and how OCAL can benefit from allowing for different types of feedback.

3.3 OCAL Design Space

An OCAL system consists of three building blocks: the learning scenario, the classifier, and the query strategy. In brief, a *learning scenario* are underlying assumptions about the application and user interaction. This includes the feedback type, e.g., sequential labels, the budget available for feedback, as well as assumptions on the data distribution, the objective of the learning process, e.g., to improve the accuracy of the classifier, and an initial setup, which includes how many labels are available when the active learning starts. In addition, there are several semi-supervised classifiers, such as SVDDneg [39], and *query strategies*, e.g., high-confidence sampling [3], which one can combine almost arbitrarily with any of the learning scenarios. Almost all classifiers and query strategies require to set additional hyperparameters. Their value can have significant influence on result quality, and a poor choice may bog it down. Moreover, a good query strategy and hyperparameter values may also depend on the active learning progress, i.e., the number of labels already provided by the user.

Navigating this design space is challenging, and it is generally not feasible to consider and evaluate all possible combinations. Although there is an overview and a benchmark on OCAL [41], a good solution still is application-specific and may require fine-tuning of several components.

3.4 Preparation of Information

Classifier training and query selection produce a lot of data. On a fine-granular level, this includes the parameterized decision function for the classifier and informativeness scores for the query strategy. After processing this data, query strategies select the most informative instances and predict a label for each observation. In general, this data can be processed and enriched in many ways before presenting it to a user. On a coarse level, one can provide users with additional information, such as explanations of the classifier or contextual information on the learning progress. We now discuss several types of information to present during an active learning iteration: the query, the result, black-box explanations and contextual information.

Query presentation: After selecting observations for feedback, “queries” in short, they must be presented to the user. In general, there are two representations of a query. First, the query has a *raw-data representation*. Examples are text documents, multimedia files, multi-dimensional time series of real-valued sensors, or sub-graphs of a network. Second, the data often is pre-processed to a *feature representation*, a real-valued vector that the classifier can process. In

principle, queries can be presented to users in either representation. Our experience is that domain experts are more familiar with raw data and demand it even if the feature representation is interpretable.

Next, one can provide context information for queries. For an individual instance, one can show the nearest neighbors of the query or a difference to prototypes of both classes. Another approach is to use visualization techniques for high-dimensional data [28, 32] to highlight the query. One can also visualize the score distribution over all candidate queries. Depending on the type of the query strategy, it also is possible to generate heatmaps that indicate areas in the data space with high informativeness [44] together with the query.

Result presentation: The presentation of a classification result largely depends on the classifier. With OCAL, the classifiers predominantly used rely on the notion of Support Vector Data Description (SVDD) [39]. In a nutshell, SVDD is an optimization problem to fit a hypersphere around the data, while allowing a small percentage of the data, the outliers, to lie outside the hypersphere. By using the kernel trick, the decision boundaries can become of arbitrary shape. So a natural presentation of SVDD is a contour plot that shows distances to the decision boundary. However, when data has more than two dimensions, contour plots are not straightforward. The reason is that contour plots rely on the distance to the decision boundary for a two-dimensional grid of observations (x_1, x_2) . However, the distance depends on the full vector (x_1, x_2, \dots, x_n) and thus cannot be computed for low-dimensional projections. One remedy would be to train a classifier for each of the projections to visualize. However, the classifier trained on the projection may differ significantly from the classifier trained on all dimensions. So a two-dimensional contour plot may have very little benefit. With common implementations of one-class classifiers, one is currently restricted to present results as plain numeric values, raw data, and predicted labels.

Black-Box Explanations: Orthogonal to inspecting the queries and the classification result, there are several approaches to provide additional explanations of the classification result. The idea is to treat the classifier, or more generally any predictive model, as a black box, and generate post-hoc explanations for the prediction of individual observations. This is also called *local explanation*, since explanations differ between instances. Recently, CAIPI, a local explainer based on the popular explanation framework LIME [30], has been proposed to explain classification results in an active learning setting [40]. The idea behind CAIPI is to provide the user with explanations for the prediction of a query and ask them to correct wrong explanations. Another application of LIME is to explain why an observation has been selected as a query [29]. The idea behind this approach is to explain the informativeness of a query by its neighborhood. The authors use uncertainty sampling, and this approach may also work with other query strategies, such as high-confidence sampling [3]. However, with more complex query strategies, for instance ones that incorporate local neighborhoods [45] or probability densities [16], applying LIME may not be straightforward. For outlier detection,

there exist further, more specific approaches to generate explanations for outlier-ness. An example is to visualize two-dimensional projections for input features that contribute most to an outlier score [19]. Other examples are methods from the VIA community that allow users to explore outliers interactively [8, 25, 43].

Contextual Information: The participant interface can also provide additional information that spans several active learning iterations. For instance, the interface can give users access to the classification history, allow them to revisit their previous responses, and give them access to responses of other users, if available. This can entail several issues, such as how to combine possibly diverging responses from different users, and the question whether users will be biased by giving them access to feedback of others. Studying such issues is focus of collaborative interactive learning [9]. Others have proposed to give users access to 2D scatter plots of the data, the confusion matrix and the progress of classification accuracy on labeled data [24]. In this case, accuracy measures may be biased. For instance, after collecting a ground truth for the first few labels, accuracy may be very high. It may decrease when more labels become available, and the labeled sample covers a larger share of the data space. So it remains an open question whether contextual information will indeed support users to provide accurate feedback.

To conclude, one faces many options in the design of OCAL systems. In particular, there are many approaches to support users with information so that they can make informed decisions on the class label. However, the approaches discussed have not yet been evaluated by means of user studies. Instead, they are limited to a theoretical discussion, simulated feedback based on benchmark data, or pen and paper surveys [40]. It is largely unclear which methods do enable users to provide feedback and indeed improve the feedback collected.

3.5 Technical Challenges

Active learning induces several technical requirements to make systems interactive, and to collect user feedback. Most requirements are general for active learning systems. But their realization with one-class classifiers is difficult.

Cold Start In most cases, active learning starts with a fully unsupervised setting, i.e., there is no labeled data available. This restricts the possible combinations of classifiers and query strategies in two cases. First, some query strategies, e.g., sampling close to the decision boundary, require a trained one-class classifier to calculate informativeness. In this case, the classifier must be applicable both in an unsupervised and a supervised setting. Second, some query strategies rely on labeled data, e.g., when estimating probability densities for the inlier class [15, 16]. In this case, one cannot calculate informativeness without labels. Current benchmarks mostly avoid this issue by simply assuming that some observations from each class are already labeled. In a real system, one must think

about how to obtain the initially labeled observations [2, 21]. One option would be to start with a query strategy that does not require any label, such as random sampling, and switch to a more sophisticated strategy once there are sufficiently many labels. Another option is to let users pick the observations to label in the beginning, and then switch to an active learning strategy [2, 6]. However, deciding when to do switches between query strategies with OCAL is an open question.

Batch Query Selection Currently, query selection for one-class classifiers is sequential, i.e., for one observation at a time. However, this sequentiality may have several disadvantages, such as frequent updating and re-training of the one-class classifier. Further, it might be easier for users to label several observations in a batch than one observation at a time [34]. This may be the case when showing a diverse set of observations helps a user to develop an intuition regarding the data set. However, there currently are no strategies to select multiple observations in batches with one-class classifiers. An open question is whether strategies that have been proposed for other use cases, such as multi-class classification [12], are applicable with one-class classifiers.

Incremental Learning The runtime for updating a classifier constrains the frequency of querying the user. In particular, excessive runtimes for classifier training result in long waiting times and do away with interactivity. Intuitively, there is an upper limit that users are willing to wait, but the specific limit depends on the application.

Several strategies are conceivable to mitigate runtime issues. First, one can rely on incremental learning algorithms [20]. However, state-of-the-art one-class classifiers like SSAD [18] have been proposed without any feature for incremental learning. Second, one can sub-sample to reduce the number of training observations. Several strategies have been proposed explicitly for one-class classifiers [23, 26, 38]. But to our knowledge, there are no studies that combine sub-sampling with OCAL. Finally, one can use speculative execution to pre-compute the classifier update for both outcomes (inlier or outlier) while the user is deciding on a label [36]. While such a strategy requires additional computational resources, it might reduce waiting times significantly and improve interactivity. The open question is how to proceed with pre-computing when the look-ahead l is more than one feedback iteration. This is a combinatorial problem, and pre-computing all 2^l learning paths is intractable. Instead, one may use conditional probabilities to pre-compute only the most likely search paths. However, there currently is no method to plan pre-computation beyond $l = 1$. If users select observations to label by themselves, pre-computation would require to compute classifier update for all observations and outcomes, which is infeasible. Thus, there is a trade-off between giving users flexibility to decide freely on which observations to label, and the capabilities of pre-computation.

Evaluation at Runtime Without a good quality estimate, it is impossible to know whether the feedback obtained from a user already is sufficient [2], i.e., the one-classifier has converged, and additional feedback would not alter the decision boundary any further. However, evaluating the classification quality of OCAL at runtime is difficult [22]. This issue exists in both, when benchmarking with simulated feedback, and in real systems – here, we focus on the latter. Users may become frustrated if they face periods where their feedback does not have any effect.

However, showing users any estimated classification quality is difficult for two reasons. First, there might be a short term bias, i.e., the classifier performance might fluctuate significantly. This may be irritating, and it may be difficult to assess for the user. Second, the number of observations in the ground truth increases over time. With only a few labeled observations, the quality estimates may have a large error. This error may reduce with more iterations. So the open question is how to estimate classification quality reliably, and how to adapt these quality estimates during learning. One conceivable option is to switch between exploration and exploitation, i.e., switch from querying for examples that improve classification quality to selection strategies that improve the quality estimate of the classifier. However, there currently is no such switching method for OCAL.

Management of Data Flows Developing an active learning system also requires a sound software architecture. Although this is not a research challenge per se, there are several aspects to consider when implementing OCAL systems. One key aspect is the management of data flows. In particular, with a distributed application, see Section 2, there are several locations where one has to retain the data set, the classifier, the predictions, and the informativeness scores. For large data sets in particular, transferring data between a client and a backend or loading data sets from disc may affect runtimes significantly. This calls for efficient data caching. Further, one must decide where computations take place. For instance, to visualize contour plots, one must predict the decision boundary for a grid of observations, possibly in multiple projections of the data. In this case, transferring the model over the network may be very little overhead. This can be an efficient strategy when evaluating the model for an observation is cheap. This is the case with SVDD, since the model consists of only a few support vectors. With multi-user studies, one may even reuse trained classifiers and informativeness scores from other user sessions with an equivalent feedback history. In this case, it might be more efficient to pre-compute grid predictions in the backend. So there are several trade-offs and factors that determine an efficient data flow. There currently is no overview on these trade-offs. It also is unclear how they affect design decisions for OCAL systems.

4 Validating OCAL with User Studies

There are a few active learning user studies which have been conducted for special use cases, such as text corpus annotation [1, 11, 31] and network security [4]. However, it is unclear how findings relate to outlier detection with OCAL – the previous sections illustrate the peculiarities of this application. Further, the plethora of design options make user studies with OCAL systems particularly challenging.

Addressing all of the design options at once is not feasible, since there are too many combinations of classifiers, query strategies and ways to prepare information for users. So we propose to start with a narrow use case and to increase the complexity of the OCAL system step-wise. Specifically, we have identified the following steps towards a validation of OCAL in real applications.

- (i) *Simplified Use Case*: Much of the value of active learning is in domains where obtaining labels is difficult, even for domain experts. However, we argue that one should identify a use case that many people can easily relate to. This has several advantages. First, we deem reproducibility more important than to obtain sophisticated insights on very special use cases. User studies are easier to reproduce when they do not depend on specific domain expertise. Further, when relationships in data are well understood, one can more easily judge whether the presentation of queries and results is accurate. So we argue to base a validation of OCAL on standard benchmark data, for instance the hand-written digit image data set MNIST⁴. Such a simplification also includes to fix the details of the feedback process, for instance to “sequential feedback” and “no initial labels”. If necessary, one should downsample data sets so that runtimes of classifiers and query strategies are not a bottleneck.
- (ii) *Validation of Information Presented*: The next step is to identify situations when users can give accurate feedback. Since the focus is to validate a learning system with users, one should start with a data set with available ground truth and select the best combination of classifier and query strategy in an experimental benchmark. This might seem counter-intuitive at first sight. In a real application, there generally are not sufficiently many labels available to conduct such a benchmark – in fact, this may even be the motivation for active learning in the first place [2, 35]. However, we argue that this is a necessary step to break the mutual dependency between selecting a good setup and collecting labels. Given a combination of classifier and query strategy, one can then apply different query and result presentations and work with explanations and contextual information. By evaluating this step with user experiments, one can derive assumptions which, if met, enable users to provide accurate feedback.
- (iii) *Validation of Classifier and Learning Strategy*: Based on these assumptions, one can vary the dimensions that have been fixed beforehand. This

⁴ <http://yann.lecun.com/exdb/mnist/>

is, one fixes the information presented to the user and varies the query strategies and classifiers. Further, one may validate specific extensions such as batch query strategies.

- (iv) *Generalization*: The first step of generalization is to scale the experiments to a large number of observations, using the techniques discussed in Section 3.5. Finally, one can then validate the approach on similar data sets, e.g., on different image data.

We expect the findings from these steps to be two-fold. On the one hand, we expect insights that are independent from the use case. For instance, whether scalability techniques are useful is likely to be use-case independent. On the other hand, many findings may depend on the type of data at hand. Explanations based on image data may be very different from the ones for, say, time series data.

Our OCAL system prototype already includes different classifiers and query strategies, see Section 2. So, in general, any researcher can already use our system to conduct Step (i) and the pre-selection of the query strategy and classifier information required for Step (ii). Regarding our prototype, the next steps are to select and implement a working set of query and result presentations, as well as to include black-box explainers and contextual information.

5 Conclusions

Validating One-Class Active Learning through user studies is challenging. One reason is that there are several open conceptual and technical challenges in the design and implementation of interactive learning systems. This article has featured a systematic overview of these challenges, and we have pointed out open research questions with one-class active learning. Next, we have sketched an architecture of a one-class active learning system, which we have implemented as a prototype. Based on it, we propose a roadmap towards validating one-class active learning with user studies.

Acknowledgement This work was supported by the German Research Foundation (DFG) as part of the Research Training Group GRK 2153: *Energy Status Data – Informatics Methods for its Collection, Analysis and Exploitation*.

References

1. Arora, S., Nyberg, E., Rosé, C.P.: Estimating annotation cost for active learning in a multi-annotator environment. In: NAACL Workshop. pp. 18–26. ACL (2009)
2. Attenberg, J., Provost, F.: Inactive learning?: Difficulties employing active learning in practice. SIGKDD Explor. Newsl. **12**(2), 36–41 (2011). <https://doi.org/10.1145/1964897.1964906>
3. Barnabé-Lortie, V., Bellinger, C., Japkowicz, N.: Active learning for One-Class classification. In: ICMLA. pp. 390–395. IEEE (2015). <https://doi.org/10.1109/ICMLA.2015.167>

4. Beaunon, A., Chifflier, P., Bach, F.: End-to-end active learning for computer security experts. In: AAAI Workshop (2018)
5. Bernard, J., Hutter, M., Zeppelzauer, M., Fellner, D., Sedlmair, M.: Comparing visual-interactive labeling with active learning: An experimental study. *Trans. Vis. Comput. Graph.* **24**(1), 298–308 (2017)
6. Bernard, J., Zeppelzauer, M., Sedlmair, M., Aigner, W.: Vial: a unified process for visual interactive labeling. *The Visual Computer* **34**(9), 1189–1207 (2018)
7. Bischof, S., Trittenbach, H., Vollmer, M., Werle, D., Blank, T., Böhm, K.: Hipe: An energy-status-data set from industrial production. In: Proceedings of the Ninth International Conference on Future Energy Systems. pp. 599–603. ACM (2018)
8. Bögl, M., Filzmoser, P., Gschwandtner, T., Lammarsch, T., Leite, R.A., Miksch, S., Rind, A.: Cycle plot revisited: Multivariate outlier detection using a distance-based abstraction. *Computer Graphics Forum* **36**(3), 227–238 (2017)
9. Calma, A., Leimeister, J.M., Lukowicz, P., Oeste-Reiss, S., Reitmaier, T., Schmidt, A., Sick, B., Stumme, G., Zweig, K.A.: From active learning to dedicated collaborative interactive learning. In: ARCS 2016; 29th International Conference on Architecture of Computing Systems. pp. 1–8 (Apr 2016)
10. Cebron, N., Richter, F., Lienhart, R.: “I can tell you what it’s not”: active learning from counterexamples. *Prog. Artif. Intell.* **1**(4), 291–301 (2012). <https://doi.org/10.1007/s13748-012-0023-9>
11. Choi, M., Park, C., Yang, S., Kim, Y., Choo, J., Hong, S.R.: Aila: Attentive interactive labeling assistant for document classification through attention-based deep neural networks. In: CHI. ACM (2019). <https://doi.org/10.1145/3290605.3300460>
12. Demir, B., Persello, C., Bruzzone, L.: Batch-Mode Active-Learning methods for the interactive classification of remote sensing images. *Trans. Geosci. Remote Sens.* **49**(3), 1014–1031 (2011). <https://doi.org/10.1109/TGRS.2010.2072929>
13. Druck, G., Settles, B., McCallum, A.: Active learning by labeling features. In: EMNLP. pp. 81–90. ACL (2009)
14. Endert, A., Hossain, M.S., Ramakrishnan, N., North, C., Fiaux, P., Andrews, C.: The human is the loop: new directions for visual analytics. *J. Intell. Inf. Syst.* **43**(3), 411–435 (2014). <https://doi.org/10.1007/s10844-014-0304-9>
15. Ghasemi, A., Manzuri, M.T., Rabiee, H.R., Rohban, M.H., Haghiri, S.: Active one-class learning by kernel density estimation. In: MLSP Workshop. pp. 1–6 (2011). <https://doi.org/10.1109/MLSP.2011.6064627>
16. Ghasemi, A., Rabiee, H.R., Fadaee, M., Manzuri, M.T., Rohban, M.H.: Active learning from positive and unlabeled data. In: ICDM Workshop. pp. 244–250 (2011). <https://doi.org/10.1109/ICDMW.2011.20>
17. Görnitz, N., Kloft, M., Rieck, K., Brefeld, U.: Active learning for network intrusion detection. In: AiSec Workshop. ACM (2009). <https://doi.org/10.1145/1654988.1655002>
18. Görnitz, N., Kloft, M., Rieck, K., Brefeld, U.: Toward supervised anomaly detection. *J. Artif. Intell. Res.* **46**, 235–262 (2013)
19. Gupta, N., Eswaran, D., Shah, N., Akoglu, L., Faloutsos, C.: Beyond outlier detection: LOOKOUT for pictorial explanation. In: ECML. pp. 122–138. Springer (2018). <https://doi.org/10.1145/1235>
20. Kefi-Fatteh, T., Ksantini, R., Kaàniche, M.B., Bouhoula, A.: A novel incremental one-class support vector machine based on low variance direction. *Pattern Recognit.* **91**, 308–321 (2019). <https://doi.org/10.1016/j.patcog.2019.02.027>
21. Kottke, D., Calma, A., Huseljic, D., Krempl, G.M., Sick, B.: Challenges of reliable, realistic and comparable active learning evaluation. In: Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning. pp. 2–14 (2017)

22. Kottke, D., Schellinger, J., Huseljic, D., Sick, B.: Limitations of assessing active learning performance at runtime. arXiv:1901.10338 (2019)
23. Krawczyk, B., Triguero, I., García, S., Woźniak, M., Herrera, F.: Instance reduction for one-class classification. *Knowl. Inf. Syst.* **59**(3), 601–628 (2019). <https://doi.org/10.1007/s10115-018-1220-z>
24. Legg, P., Smith, J., Downing, A.: Visual analytics for collaborative human-machine confidence in human-centric active learning tasks. *Hum. Cent. Comput. Inf. Sci.* **9**(1), 5 (2019). <https://doi.org/10.1186/s13673-019-0167-8>
25. Leite, R.A., Gschwandtner, T., Miksch, S., Kriglstein, S., Pohl, M., Gstrein, E., Kuntner, J.: Eva: Visual analytics to identify fraudulent events. *Trans. Vis. Comput. Graph.* **24**(1), 330–339 (2017). <https://doi.org/10.1109/TVCG.2017.2744758>
26. Li, Y.: Selecting training points for one-class support vector machines. *Pattern Recognit. Lett.* **32**(11), 1517–1522 (2011). <https://doi.org/10.1016/j.patrec.2011.04.013>
27. Lin, H., Gao, S., Gotz, D., Du, F., He, J., Cao, N.: Rclens: Interactive rare category exploration and identification. *Trans. Vis. Comput. Graph.* **24**(7), 2223–2237 (2017). <https://doi.org/10.1109/TVCG.2017.2711030>
28. Liu, S., Maljovec, D., Wang, B., Bremer, P.T., Pascucci, V.: Visualizing high-dimensional data: Advances in the past decade. *Trans. Vis. Comput. Graph.* **23**(3), 1249–1268 (2016). <https://doi.org/10.1109/TVCG.2016.2640960>
29. Phillips, R.L., Chang, K.H., Friedler, S.A.: Interpretable active learning. arXiv preprint arXiv:1708.00049 (2017)
30. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why Should I Trust You?”: Explaining the predictions of any classifier. In: SIGKDD. pp. 1135–1144 (2016). <https://doi.org/10.1145/2939672.2939778>
31. Ringger, E.K., Carmen, M., Haertel, R., Seppi, K.D., Lonsdale, D., McClanahan, P., Carroll, J.L., Ellison, N.: Assessing the costs of machine-assisted corpus annotation through a user study. In: LREC. pp. 3318–3324 (2008)
32. Sacha, D., Zhang, L., Sedlmair, M., Lee, J.A., Peltonen, J., Weiskopf, D., North, S.C., Keim, D.A.: Visual interaction with dimensionality reduction: A structured literature analysis. *Trans. Vis. Comput. Graph.* **23**(1), 241–250 (2016). <https://doi.org/10.1109/TVCG.2016.2598495>
33. Seifert, C., Granitzer, M.: User-based active learning. In: ICDM Workshop. pp. 418–425. IEEE (2010). <https://doi.org/10.1109/ICDMW.2010.181>
34. Settles, B.: From theories to queries: Active learning in practice. In: AISTATS Workshop. pp. 1–18 (2011)
35. Settles, B.: Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* pp. 1–114 (2012)
36. Sperrle, F., Bernard, J., Sedlmair, M., Keim, D., El-Assady, M.: Speculative execution for guided visual analytics. In: VIS Workshop (2018)
37. Stokes, J.W., Platt, J.C., Kravis, J., Shilman, M.: Aladin: Active learning of anomalies to detect intrusions. Tech. rep., Microsoft Research (2008)
38. Sun, W., Qu, J., Chen, Y., Di, Y., Gao, F.: Heuristic sample reduction method for support vector data description. *Turkish Journal of Electrical Engineering & Computer Sciences* **24**(1), 298–312 (2016)
39. Tax, D.M.J., Duin, R.P.W.: Support vector data description. *Mach. Learn.* **54**(1), 45–66 (2004). <https://doi.org/10.1023/B:MACH.0000008084.60811.49>
40. Teso, S., Kersting, K.: Explanatory interactive machine learning. In: AAAI (2019)
41. Trittenbach, H., Enghardt, A., Böhm, K.: An overview and a benchmark of active learning for outlier detection with One-Class classifiers. arXiv:1808.04759 (2018)

42. Vollmer, M., Englhardt, A., Trittenbach, H., Bielski, P., Karrari, S., Böhm, K.: Energy time-series features for emerging applications on the basis of human-readable machine descriptions. In: Proceedings of the Tenth ACM International Conference on Future Energy Systems. pp. 474–481. ACM (2019)
43. Wilkinson, L.: Visualizing big data outliers through distributed aggregation. *Trans. Vis. Comput. Graph.* **24**(1), 256–266 (2017)
44. Yang, Y., Loog, M.: A benchmark and comparison of active learning for logistic regression. *Pattern Recognit.* **83**, 401–415 (2018). <https://doi.org/10.1109/TVCG.2017.2744685>
45. Yin, L., Wang, H., Fan, W.: Active learning based support vector data description method for robust novelty detection. *Knowl. Based. Syst.* **153**, 40–52 (2018). <https://doi.org/10.1016/j.knosys.2018.04.020>