RISSAD: Rule-based Interactive Semi-Supervised Anomaly Detection

J. Deng and E. T. Brown

DePaul University, Chicago, IL, U.S.A.

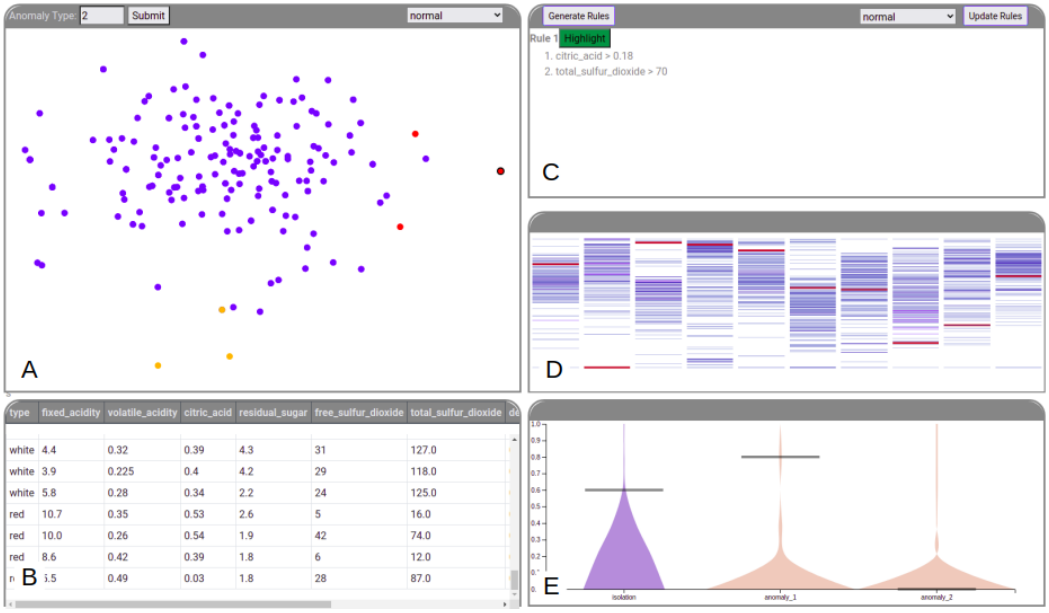

Figure 1: *RISSAD prototype: **A**) scatterplot, **B**) data table, **C**) descriptive rules, **D**) data distributions, and **E**) isolation and similarity scores.*

Abstract

Anomaly detection has gained increasing attention from researchers in recent times. Owing to a lack of reliable ground-truth labels, many current state-of-art techniques focus on unsupervised learning, which lacks a mechanism for user involvement. Further, these techniques do not provide interpretable results in a way that is understandable to the general public. To address this problem, we present RISSAD: an interactive technique that not only helps users to detect anomalies, but automatically characterizes those anomalies with descriptive rules. The technique employs a semi-supervised learning approach based on an algorithm that relies on a partially-labeled dataset. Addressing the need for feedback and interpretability, the tool enables users to label anomalies individually or in groups, using visual tools. We demonstrate the tool's effectiveness using quantitative experiments simulated on existing anomaly-detection datasets, and a usage scenario that illustrates a real-world application.

CCS Concepts

• ***Computing methodologies*** → *Interactive systems; Pattern analysis;*

1. Introduction

Anomaly detection plays an important role in many areas of research, including education [MXC*19], cyber-security [HLG14] and mechanical engineering [GMESK99]. In general, an anomaly is vaguely defined as a data point that does not share a similar pattern with the rest of the population. However, this ambiguity in the definition leads to the lack of ground-truth labels in many datasets. Because of this, and the imbalance of normal vs. anomaly points

DOI: 10.2312/evs.20211050

by definition, many traditional supervised learning algorithms such as decision trees, neural networks and multi-class support vector machines, will often perform poorly on problems in which it is expensive to obtain labels for each training case [CBK09]. Faced with those challenges, many state-of-art techniques for anomaly detection rely heavily on unsupervised learning algorithms such as Local Outlier Factor (LOF) [BKNS00], Isolation Forest [LTZ08] and One-Class SVM [SPST*01]. Despite some promising results delivered by these techniques in various situations, they generally do not provide a robust mechanism for interpretation of the results.

To address this issue, efforts are being made in machine learning interpretability [GBY*18, MQB18, RSG16a, RSG16b], and with visualization across different application domains. For anomalies, Mu, et al. [MXC*19], introduce a system that detects abnormal behaviors of users registered in Massive Open Online Courses. Lin, et al. [LGG*17], build a visual system to identify rare categories based on active learning. Zhao, et al. [ZCW*14], contribute a timeline visualization tool to analyze anomalous user behaviors in social media platforms. Although these studies all made meaningful contributions to help users understand the data instead of simply applying a "black-box" machine learning technique, their target audiences are mostly experts.

To fill this gap, we developed an interactive anomaly detection technique that generates rules for anomaly groups that will be understandable to a broader user base. Anomalies are a technical concept, so as opposed to a general audience, we target those comfortable enough with data analysis to engage conceptually. We preserve user effort by needing only a limited number of labels, and we restrain the interface to interactions that do not require expertise in anomaly detection. Applying this technique requires two steps: in the first step, the user labels anomalous points based on their data understand using visual tools. Anomalies are grouped into clusters automatically, and in the second step of user interaction, the user may extend them before choosing to characterize the anomaly groups and remaining data each with a series of descriptive rules based on their distinctive value ranges across relevant variables. The main contributions of our work are: (1) RISSAD: an interactive anomaly detection technique for non-expert users that characterizes groups of anomalies automatically with descriptive rules, (2) a prototype implementation of this technique, (3) an evaluation by simulation of user interactions over multiple datasets, demonstrating accurate rule sets with limited user feedback, and (4) a usage scenario showing how this can be applied to discover and describe anomalies in real-world data.

2. Related Work

2.1. Anomaly Detection Algorithms

In general, most anomaly detection techniques are traced back to four categories: (1) classification-based algorithms [HHWB02, MC03, WMCW03], (2) nearest-neighbour-based algorithms [BS03, BKNS00], (3) clustering-based algorithms [MLC07, SPBW12], and (4) statistical-based algorithms [KK17, YTWM04]. To combine the advantages of various techniques, ensemble approaches have gained popularity in recent years [VC17, ZDH*17]. Dimensionality-reduction, such as multidimensional scaling (MDS) [Kru64] and principal component analysis (PCA) [SCSC03], is also used for anomaly detection given its advantage in reducing model complexity and reducing the computational cost.

2.2. Anomaly Detection Visualization

Combined with detection algorithms explained in subsection 2.1, visualizations are widely used to enhance a user's understanding of the problem and supplement the learning process of the chosen technique. For example, Arakwa, et al., present an automated visual system to detect anomalous patterns in human behaviors with a modified Gaussian mixture model (GMM) [AY19]. Lin, et al., proposes a visual system that relies on the scatterplot generated using dimensionality reduction [LGG*17]. Xu, et al., present a hybrid approach that ensembles multiple state-of-art anomaly detection algorithms and assists users in interacting with data [XXM*18]. Although all those techniques provide interpretable insights, they often require a high-level understanding of statistics, which are generally obscure to non-experts. Inspired by this, our technique offers a solution to generate comprehensible rules based on interactive feedback from users without expertise.

3. Anomaly Detection and Description Algorithm for RISSAD

RISSAD requires an anomaly detection algorithm that can use limited user labels and produce understandable, descriptive rules for the anomalies. Our algorithm is based on *ADOA*, presented in Zhang, et al. [ZLZ*18]. Its underlying assumption is that anomalies are often isolated from the rest of the population, but close to other anomalies in distinct clusters. These concepts are made concrete with the *Isolation Score* (IS) and *similarity score* (SS).

IS represents the isolation degree of a point from the majority of the population. The score is calculated for a point, x, using the multiple random-attribute decision trees produced by running the unsupervised algorithm *isolation forest* [LLYL02], based on the point's average depth $E(d(x))$ (in the formula below, c is a normalization constant). Conversely, $SS_i(x)$ represents the similarity of x within each anomaly cluster, i, where μ_i is the cluster center.

$$\begin{aligned} IS(x) &= 2^{-\frac{E(d(x))}{c}} \\ SS_i(x) &= e^{-(x-\mu_i)^2} \end{aligned} \tag{1}$$

Each point gets its final *SS* score as the max over i (the score from the cluster it fits best). However, we give the user the ability to override this with their labels.

The algorithm is implemented in two stages. First, (1) the labeled anomalies are clustered using k-means. For each unlabeled point, its *Isolation Score* (*IS*) and *Similarity Score* (*SS*) are computed separately. Then in stage two, (2) any unlabeled point is also automatically labeled as one of the anomaly types, or *normal* class if the weighted average of *IS* and *SS* exceeds a upper threshold or falls below a lower threshold. A weight (w) is computed as the reliability of this automatic labeling [ZLZ*18]. User labels are weighted the full value of $w = 1$. Next, a supervised learning algorithm is trained, using w as per-case weights. Although the original ADOA paper chooses SVM as the learning model, we use a decision-tree based rule learner (C5.0 R library [Rul19]), as in related interactive rule learning work [CB20].

4. The RISSAD Prototype

This section covers (1) the visual components of the prototype, (2) the workflow for providing feedback on anomalies, and (3) the features of descriptive rule generation.

4.1. Overview of the Components

Our prototype tool (Figure 1) constitutes five parts, each with a specific contribution to the workflow. In the figure, **A** is the scatterplot with a projection of multi-dimensional data points into 2D using *multidimensional scaling* (MDS) [Mea92]. We choose this straightforward method because no projection is perfect, but newer algorithms like t-SNE can reveal misleading groups due to parameter sensitivity [WVJ16], which would be problematic for anomaly detection.

Figure 1**B** is the data table, providing a detail view corresponding to **A**. **C** shows the descriptive rules learned from interactions with the tool. **D** is a *barcode* or *parallel bars* plot showing the distribution of the data variables, with highlights for individual data points as needed for context [Bos21, BLBC12]. Each attribute of the original dataset corresponds to one column, mapping the value range to the full height. Each thin line in the column represents a datapoint, drawn with transparency so that the color density represents the data distribution of that column's variable. Figure 1 **E** shows violin plots for (1) the *Isolation Scores* of the entire dataset and (2) the distributions of the *Similarity Scores* in each anomaly cluster, computed based on the algorithm explained in section 3. When reviewing points from the scatterplot, highlights in **E** help quantify the likelihood of an anomaly.

4.2. Points Labeling

In the beginning, all data points in the scatter plot (Figure 1**A**) are assumed to be normal (non-anomalies) and are marked in blue. In the violin plot (Figure 1**E**), only the *Isolation Scores* plot exists, as no anomaly clusters exist. When a user is interested in a point, they can place the mouse cursor over it to see: (1) the corresponding row placed at the top and marked in green in the table (Figure 1**B**), (2) the barcode plot showing thin lines in each column of each attribute corresponding to that specific data point marked in red (Figure 1**D**), (3) and a black line placed on the violin plot (Figure 1**E**), to show the *Isolation Score* of the moused-over sample in context. Each line of the barcode plot is also bound with a mouse-over event to provide a tool tip at the bottom of the plot with the attribute name and percentage. While the barcode plot, scatterplot and table provide views of the data in their raw context, the violin plot of the *Isolation Scores* gives an intuitive measure that may contribute to the user's consideration in labeling a point as an anomaly. Higher scores lead to higher probability of being an anomaly. Based on understanding of the data and exploration with the provided visual tools and interactions, the user can then click all potential anomalous points and those points will be marked in red.

Once a user decides that a sufficient number of points have been labeled, they can then enter the number of anomaly types based on observation of the scatter plot or on prior knowledge, or they can simply leave the default of three. Then the violin plots (Figure 1**E**) will be updated with *Similarity Scores* for each anomaly type. In the previous step, the user checks points mainly based on their *Isolation Scores*. In the next step, *similarity scores* can be utilized for selecting more anomalous points while everything else remains the same. The user also has the flexibility to assign a point to any of the anomaly types by selecting the options in the drop-down menu. Each anomaly type is marked in a different color. As more points are assigned to each anomaly type, the violin plots of the *similarity scores* are adjusted accordingly. This capability does not exist in the original *ADOA* algorithm, because labels provide binary anomaly status only, and there is no opportunity to get user feedback about the clusters of anomalies.

4.3. Rules Generation

After anomaly points are labeled as described in subsection 4.2, the user can click the *Generate Rules* button on the top left corner of the descriptive rules (Figure 1**C**). The rules are created using algorithm explained in section 3 and displayed as illustrated in Figure 2b. Each anomalous class and the normal class has its own corresponding set of rules. The user can use the dropdown menu on the top right corner to check the rules of other classes. Each rule has a highlight button originally marked in green. Once the user clicks the button, the button will be switched to orange and the following two events will be activated: (1) points associated with the rule will be highlighted with black borders as shown in Figure 2a, and (2) lines associated with rules in the barcode plot (Figure 1**D**) will be marked in orange as illustrated in Figure 2. By checking those highlighted points, the user can choose to assign each point to any of the anomaly types or reassign it to any of the anomalous groups or to the normal class if they decide the point is mis-classified, by checking the dropdown menu on the top right corner of the scatterplot as shown in Figure 1**A**. After this step, the interface will start with updated information, forming a feedback loop for the user to further refine the rules by repeating these steps.

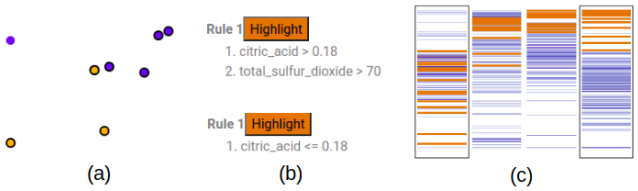

Figure 2: *An example interaction between the Scatterplot (a), the Rule Panel (in highlight mode) (b), and the Barcode Plot (c), as described in the usage scenario of subsection 5.2*

5. Evaluation

We evaluate RISSAD with (1) simulated interaction experiments to estimate expected performance on varied datasets, and (2) with a usage scenario to illustrate its capabilities on real-world data.

5.1. Simulation

To understand how RISSAD may perform with varied data and different tasks, we simulate sequences of user interactions and compare the resulting models against those of three other algorithms in the same context. We evaluate our proposed algorithm

(ADOA_Tree) against three others: (1) Isolation Forest (IF), an unsupervised approach, (2) a decision tree (Naive_Tree), which represents a fully supervised approach, and (3) the technique adopted by the original *ADOA* authors (ADOA_SVM).

We simulate a user labeling one point at a time, and since we expect labels often will be related to the *Isolation* and *Similarity* scores (see section 3), we simulate using each one half the time (always strongest scores first). In each experiment run, we create training and validation sets (70% vs. 30%). When training, we use further three-fold cross validation to tune the hyper-parameters. Rather than accuracy, we use *area under the curve* (AUC) as a performance metric. Because the classifiers are probabilistic, AUC can be calculated based on a range of thresholds.

In Figure 3, we show graphs that compare the four algorithms on four different datasets, chosen to cover a range of size and dimensionality. The x-axis represents the total number of labeled samples, and the y-axis represents the AUC score. Those experiments help understand how the model can be expected to improve with incremental human labelling. Overall, we find that in most cases, our proposed approach outperforms Naive_Tree and has similar performance to ADOA_SVM. In some cases, especially with fewer labels, unsupervised IF performs better than ADOA_Tree. However, our algorithm provides a unique advantage of generating rules to assist user comprehension of the anomalies. From the experiments, we find that this advantage can be achieved without significant trade-off on accuracy in most cases.

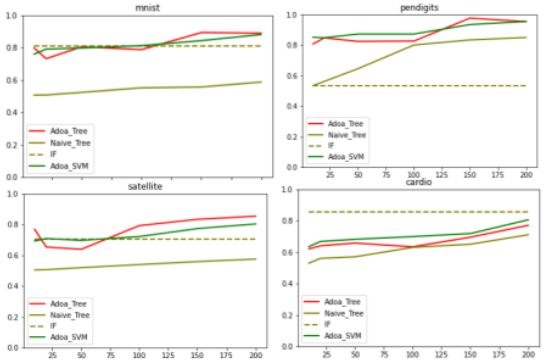

Figure 3: *Our ADOA_Tree compared to other algorithms with AUC vs. the number of labels provided. See subsection 5.1.*

5.2. Usage Scenario

Robert is a professional winemaker and he wants to explore rare, high-quality wines and understand their characteristics. An anomaly detection tool could help him find such examples, because these wines would stand out. He collects a dataset of 13 variables with technical attributes of each of 6,497 wines, which has a quality rating from 0-10 for each wine [CCA*09], and filters for quality ($quality \geq 8$). Robert is not proficient with machine learning, but our technique helps him gain a competitive advantage by finding anomalous wines and characterizing exactly what makes them unusual, so he can take action in designing his next recipe.

Robert works with the RISSAD prototype as illustrated in section 4. He begins with the scatterplot (Figure 1**A**) and notices several points that are clearly separated from the majority. The violin plots of *Isolation* and *Similarity* scores (Figure 1**E**) confirm the status of these points to be likely anomalies. To obtain optimal results, in the first round, he only selects points with *Isolation scores* over 0.8. This produces a set of three points shown in red. He notices two of those points are significantly closer to each other, implying that there may be two groups of anomalies. He sets the *Cluster Size* to two, and presses *Submit* to request a clustering.

Now, he has two groups of anomalous points and can view the *Similarity Scores* per anomaly type in addition to the *Isolation Scores* in the violin plots. He selects additional points, expanding his labels, because he notes several that are close to his original selection in the scatterplot and have high *similarity score* (above 0.6). The newly selected points were not obvious at first, but with the *Isolation* and *Similarity* scores as a guide, he decides they are different enough from the normal data to deserve a closer look. Figure 1**A** illustrates the layouts after the selection. He clicks the *Generate Rules* button and sees the result shown in Figure 2b, with rules that describe these two anomaly clusters.

He then clicks the *Highlight* buttons next to each rule to find other potential anomalous samples of each type. Out of those highlighted potential anomalies, which are previously assumed to be normal, he finds that three points have exceptionally low total sulfur dioxide by checking the barcode plot. Since total sulfur dioxide has been identified as one of the important features for the rule that describes *anomaly type 1*, he checks those three points and clicks *Generate Rules*. More refined rules are generated for both of the anomaly types, and for the normal case. Figure 4 illustrates the newly generated rules for both of the anomaly types. Compared to the original rules in Figure 2b, the new rules for *anomaly type 2* become more refined as they now include two other features (*pH* and *total sulfur dioxide*) apart from the original, *citric acid*.

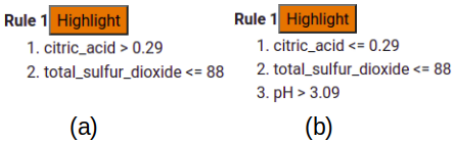

Figure 4: *Refined rules of (a) Anomaly type 1, (b) Anomaly type 2*

6. Conclusion and Future Work

In this paper, we present a technique, RISSAD, to help users detect and understand anomalies. Our prototype provides its user visual aids for finding and labeling anomalies, particularly with *isolation* and *similarity* scores. It further characterizes anomalies through descriptive rules. Through simulations, we find that the underlying machine learning can accurately label and describe anomalies with limited user intervention and without significant trade-off in accuracy. In a usage scenario, we provide an example of successful application of our technique to real data. While our technique shows promising results in generating interpretable results to anomaly detection, there are limitations. Anomaly detection algorithms can be expensive due to many similarity comparisons. These computations can run in parallel, though. Visually, scatterplots get bogged down with too many points and the barcode plot will be harder to use with many dimensions. To best push past these limitations, our future work will include a user study to understand how the limitations actually affect the task, so we can choose appropriate alternatives.

References

[AY19] ARAKAWA R., YAKURA H.: REsCUE: A framework for REal-time feedback on behavioral CUEs using multimodal anomaly detection. In *2019 Proceedings of CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–13. 2

[BKNS00] BREUNIG M. M., KRIEGEL H.-P., NG R. T., SANDER J.: LOF: identifying density-based local outliers. In *2000 Proceedings of the ACM SIGMOD international conference on Management of data* (2000), pp. 93–104. 2

[BLBC12] BROWN E. T., LIU J., BRODLEY C. E., CHANG R.: Dis-Function: Learning distance functions interactively. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2012), IEEE, pp. 83–92. 3

[Bos21] BOSTOK M.: Barcode plot / d3 / observable, April 2021. URL: https://observablehq.com/@d3/barcode-plot. 3

[BS03] BAY S. D., SCHWABACHER M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *2003 Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003), pp. 29–38. 2

[CB20] CAO F., BROWN E. T.: DRIL: Descriptive rules by interactive learning. In *2020 IEEE Visualization Conference (VIS)* (2020), IEEE, pp. 256–260. 2

[CBK09] CHANDOLA V., BANERJEE A., KUMAR V.: Anomaly detection: A survey. *ACM computing surveys (CSUR) 41*, 3 (2009), 1–58. 2

[CCA*09] CORTEZ P., CERDEIRA A., ALMEIDA F., MATOS T., REIS J.: Modeling wine preferences by data mining from physicochemical properties. *Decision support systems 47*, 4 (2009), 547–553. 4

[GBY*18] GILPIN L. H., BAU D., YUAN B. Z., BAJWA A., SPECTER M., KAGAL L.: Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE International Conference on data science and advanced analytics (DSAA)* (2018), IEEE, pp. 80–89. 2

[GMESK99] GUTTORMSSON S. E., MARKS R., EL-SHARKAWI M., KERSZENBAUM I.: Elliptical novelty grouping for on-line short-turn detection of excited running rotors. *IEEE Transactions on Energy Conversion 14*, 1 (1999), 16–22. 1

[HHWB02] HAWKINS S., HE H., WILLIAMS G., BAXTER R.: Outlier detection using replicator neural networks. In *2002 International Conference on Data Warehousing and Knowledge Discovery* (2002), Springer, pp. 170–180. 2

[HLG14] HONG J., LIU C.-C., GOVINDARASU M.: Integrated anomaly detection for cyber security of the substations. *IEEE Transactions on Smart Grid 5*, 4 (2014), 1643–1653. 1

[KK17] KWAK S. K., KIM J. H.: Statistical data preparation: management of missing values and outliers. *Korean journal of anesthesiology 70*, 4 (2017), 407. 2

[Kru64] KRUSKAL J. B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika 29*, 1 (1964), 1–27. 2

[LGG*17] LIN H., GAO S., GOTZ D., DU F., HE J., CAO N.: RClens: Interactive rare category exploration and identification. *IEEE transactions on visualization and computer graphics 24*, 7 (2017), 2223–2237. 2

[LLYL02] LIU B., LEE W. S., YU P. S., LI X.: Partially supervised classification of text documents. In *ICML* (2002), vol. 2, pp. 387–394. 2

[LTZ08] LIU F. T., TING K. M., ZHOU Z.-H.: Isolation forest. In *2008 IEEE International Conference on Data Mining* (2008), IEEE, pp. 413–422. 2

[MC03] MAHONEY M. V., CHAN P. K.: Learning rules for anomaly detection of hostile network traffic. In *2003 IEEE International Conference on Data Mining* (2003), IEEE, pp. 601–604. 2

[Mea92] MEAD A.: Review of the development of multidimensional scaling methods. *Journal of the Royal Statistical Society: Series D (The Statistician) 41*, 1 (1992), 27–39. 3

[MLC07] MÜNZ G., LI S., CARLE G.: Traffic anomaly detect. using k-means clust. In *2007 GI/ITG Workshop MMBnet* (2007), pp. 13–14. 2

[MQB18] MING Y., QU H., BERTINI E.: Rulematrix: Visualizing and understanding classifiers with rules. *IEEE transactions on visualization and computer graphics 25*, 1 (2018), 342–352. 2

[MXC*19] MU X., XU K., CHEN Q., DU F., WANG Y., QU H.: MOOCad: Visual analysis of anomalous learning activities in massive open online courses. In *2019 EuroVis (Short Papers)* (2019), pp. 91–95. 1, 2

[RSG16a] RIBEIRO M. T., SINGH S., GUESTRIN C.: Model-agnostic interpretability of m.l. *arXiv preprint arXiv:1606.05386* (2016). 2

[RSG16b] RIBEIRO M. T., SINGH S., GUESTRIN C.: "why should i trust you?": Explaining the Predictions of Any Classifier. In *2016 Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining* (2016), pp. 1135–1144. 2

[Rul19] RULEQUEST RESEARCH: Information on see5/c5.0, April 2019. Retrieved from https://www.rulequest.com/see5-info.html August 24, 2020. URL: https://www.rulequest.com/see5-info.html. 2

[SCSC03] SHYU M.-L., CHEN S.-C., SARINNAPAKORN K., CHANG L.: *A novel anomaly detection scheme based on principal component classifier*. Tech. rep., Miami University Coral Gables, Florida Department of Electrical and Computer Engineering, 2003. 2

[SPBW12] SYARIF I., PRUGEL-BENNETT A., WILLS G.: Unsupervised clustering approach for network anomaly detection. In *2012 International conference on networked digital technologies* (2012), Springer, pp. 135–145. 2

[SPST*01] SCHÖLKOPF B., PLATT J. C., SHAWE-TAYLOR J., SMOLA A. J., WILLIAMSON R. C.: Estimating the support of a high-dimensional distribution. *Neural computation 13*, 7 (2001), 1443–1471. 2

[VC17] VANERIO J., CASAS P.: Ensemble-learning approaches for network security and anomaly detection. In *2017 Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks* (2017), pp. 1–6. 2

[WMCW03] WONG W.-K., MOORE A. W., COOPER G. F., WAGNER M. M.: Bayesian network anomaly pattern detection for disease outbreaks. In *2003 Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (2003), pp. 808–815. 2

[WVJ16] WATTENBERG M., VIÉGAS F., JOHNSON I.: How to use t-sne effectively. *Distill* (2016). URL: http://distill.pub/2016/misread-tsne, doi:10.23915/distill.00002. 3

[XXM*18] XU K., XIA M., MU X., WANG Y., CAO N.: Ensemblelens: Ensemble-based visual exploration of anomaly detection algorithms with multidimensional data. *IEEE transactions on visualization and computer graphics 25*, 1 (2018), 109–119. 2

[YTWM04] YAMANISHI K., TAKEUCHI J.-I., WILLIAMS G., MILNE P.: On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery 8*, 3 (2004), 275–300. 2

[ZCW*14] ZHAO J., CAO N., WEN Z., SONG Y., LIN Y.-R., COLLINS C.: #FluxFlow: Visual analysis of anomalous information spreading on social media. *IEEE transactions on visualization and computer graphics 20*, 12 (2014), 1773–1782. 2

[ZDH*17] ZHANG X., DOU W., HE Q., ZHOU R., LECKIE C., KOTAGIRI R., SALCIC Z.: LSHiForest: A generic framework for fast tree isolation based ensemble anomaly analysis. In *2017 IEEE International Conference on Data Engineering (ICDE)* (2017), IEEE, pp. 983–994. 2

[ZLZ*18] ZHANG Y.-L., LI L., ZHOU J., LI X., ZHOU Z.-H.: Anomaly detection with partially observed anomalies. In *2018 Companion Proceedings of the The Web Conference* (2018), pp. 639–646. 2