A Comparative Analysis of Traditional and Deep Learning-based Anomaly Detection Methods for Streaming Data

Mohsin Munir*†, Muhammad Ali Chattha*†‡, Andreas Dengel*†, and Sheraz Ahmed†
*Technische Universität Kaiserslautern, Kaiserslautern, Germany
†German Research Center for Artificial Intelligence (DFKI) GmbH, Kaiserslautern, Germany
‡School of Electrical Engineering and Computer Science (SEECS) – NUST, Islamabad, Pakistan
{firstname.lastname}@dfki.de

***Abstract*—With the Internet of Things (IoT) devices becoming an integral part of human life, the need for robust anomaly detection in streaming data has also been elevated. Dozens of distance-based, density-based, kernel-based, and cluster-based algorithms have been proposed in the area of anomaly detection. Recently, because of the robustness of the deep neural networks (DNN), different deep learning-based anomaly detection methods have also been proposed. With all these rapid developments, there exists a small number of comparative studies for anomaly detection methods. Even in those studies, the comparison is done only in typical anomaly detection settings without taking the streaming data into consideration. The presence of intrinsic time-series characteristics like trend, seasonality, and change-point makes it important to study the behavior of commonly used anomaly detection methods on streaming data. Moreover, the comparison of traditional methods with deep learning-based methods also brings exciting insights about the data which are generally overlooked by traditional methods. In this study, we compare 13 anomaly detection methods on two commonly used streaming data sets. We used four different evaluation metrics to evaluate the methods from different perspectives. Our analysis reveals that the deep learning-based anomaly detection methods are superior to traditional anomaly detection methods.**

***Index Terms*—anomaly detection, streaming data, deep learning**

I. Introduction

Anomaly detection has always been of great human interest. It is a common observation that an abnormal activity attracts a lot of human attention. For example, a driving car is considered an anomaly inside a park and a person with blue hairs will stand out in a shopping mall. Anomaly is an outlier which Hawkins [1] defined as *an observation that deviates so significantly from other observations as to arouse suspicion that it was generated by a different mechanism*. Detecting and mitigating anomalies is vital in the manufacturing and industrial sector and also crucial in the healthcare and surveillance sector. A timely detected anomaly can improve machine performance, avoid a machinery downtime, reduce a disease outbreak, and even save a human life [2], [3]. Due to its applicability in almost every field of life, the meaning of anomaly varies from domain to domain and depends on the context in which it is used. Detecting unauthorized interference in a network is anomaly detection in the networking domain, whereas, detecting a malfunctioned component inside a machine is referred to as fault detection.

This work was supported by the NVIDIA AI Lab (NVAIL) and IAV-FLaP programs.

Anomaly detection is an old research topic and a lot of advancements have been made in this area. There are two main modalities in which this wide research area can be divided: image-based and non-image-based. The notable modality is the image-based, in which it is easy to highlight a detected anomaly and most of the research studies are available for this modality. But, in the current era of the Internet of Things (IoT), where each device is connected to the internet and generating a bulk of data, non-image-based modality is also getting prominent for quite some time. Recent amplification in the usage of IoT devices has created a high demand for robust anomaly detection methods for streaming sensor data. The presence of the intrinsic characteristics of a sensor-based streaming data (periodicity, trend, seasonality, and irregularity) makes the process of robustly detect and represent anomalies hard.

Vital developments in the area of deep neural networks (DNN) have proved them to be a very good option for most of the classification and regression problems [4]. There exist different deep learning-based and traditional approaches for anomaly detection, but their direct comparison of non-image streaming data is missing in the literature. The focus of this study is to draw a comparative analysis for the most commonly used traditional and deep learning-based anomaly detection methods for non-image streaming data. Given the superiority of DNNs in many recent studies, we hypothesize that DNN-based anomaly detection algorithms may outperform commonly used traditional methods on streaming data.

II. Literature Review

Recently, Deep Learning (DL) based approaches have enjoyed significant attention, owing to their amazing performance in various domains. Hence, it is no surprise that there has been an increase in DL-based methods for anomaly detection as well. Chalapathy et al. [5] provides a review

978-1-7281-4550-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMLA.2019.00105

of DL-based methods for anomaly detection employed in different application scenarios. Techniques reviewed in their study include both image as well as time-series domain with applications ranging from fraud detection, intrusion detection to medical anomaly detection and video surveillance. Although this survey provides a comprehensive review of DL techniques, there is no comparative analysis of DL-based techniques with traditional anomaly detection techniques. Similarly, Chandola et al. [6] provides a comprehensive survey of anomaly detection techniques comprising not only of methods based on simple machine learning, like clustering and nearest neighbours, but also based on statistical approaches and information theory with a diverse range of applications. Again, there is no comparative study among the techniques discussed in the article.

Goldstein and Uchida [7] provides a comparative study of different anomaly detection techniques for a range of different data sets. Although this study does compare different techniques on the same data sets to give a better comparison, it primarily focuses on multivariate tabular data and does not incorporate recent DL-based approaches. Similarly, Gupta et al. [8] have performed an extensive survey on outlier detection techniques for temporal data. Their study provides an extensive overview of different techniques employed in multiple temporal data sets, but it lacks DL-based techniques and their comparative analysis. Similarly, Kiran et al. [9] provides a review of DL-based approaches for anomaly detection in videos. Adewumi et al. [10] provides a comprehensive survey of DL-based approaches in the domain of fraud detection. Similarly, Hodge et al. [11] provides an extensive survey on statistical and some of the earlier machine learning-based outlier detection methodologies.

Most of the surveys in the literature are general in nature that span over techniques that are built for different problems and may even belong to different domain. Moreover, most of the review articles lack quantitative comparison that can determine the efficacy of each of the techniques for a given domain. In our study, we provide a quantitative comparison of the most commonly used anomaly detection methods on streaming non-image data sets. Also, we provide a comparison between traditional anomaly detection methods and DL-based methods.

III. Anomaly Detection Methods

In this section we explain the anomaly detection methods selected for the comparative analysis. We have selected the methods which are commonly used for anomaly detection in the streaming non-image modality.

A. kNN Anomaly Detection [12]

The *k*-nearest-neighbor (*k*NN) anomaly detection is one of the most commonly used distance-based anomaly detection methods. It is a simple technique which works out-of-the-box in most of the cases and detects global anomalies precisely. For each data point in a streaming data set, the *k*-nearest-neighbors have to be found. Based on these neighbors, the anomaly score is calculated. The anomaly score depends on the average distance to all the *k* neighbors.

B. Local Outlier Factor (LOF) [13]

The LOF is also a distance-based anomaly detection method. It is used for detecting local anomalies based on the local densities. In this method, the *k*-nearest-neighbors have to be found for each data point in a given streaming data set. By using *k*-nearest-neighbors, the local density of each data point is estimated by computing the local reachability density (LRD). Finally, the anomaly score is computed by comparing the LRD of a data point with all the LRDs of its *k* neighbors.

C. Connectivity-based Outlier Factor (COF) [14]

The connectivity-based outlier factor is an improved version of LOF. In LOF, it is assumed that a given data is distributed in a spherical way around a given instance. For the cases in which this indirect condition is not fulfilled, the density estimation is incorrect and leads to poor anomaly detection. This LOF limitation is addressed in COF by estimating the local density of the neighborhood using chaining distance. Chaining distance is a shortest-path approach which is the minimum of the sum of all distances connecting all k neighbors and the instance.

D. Local Correlation Integral (LOCI) [15]

In all of the distance based anomaly detection approaches, the selection of parameter k plays a vital role in the overall performance. There is no fix rule on the basis of which the value of k can be estimated. This limitation is addressed in LOCI with the help of a maximization approach. It defines the r-neighborhood by using a radius r. The radius is expanded over time which makes this method very computational expensive.

E. Isolation Forest (iForest) [16]

This anomaly detection method is based on the concept of 'isolation' – in contrast to the widely-used distance and density measures. In this approach, the anomalies are 'isolated' from normal instances. The data instances which are few in numbers and their attribute-values are very different from the rest of the data instances are the instances that are more susceptible to be put in isolation. This method uses a binary tree structure called isolation tree $(iTree)$ to isolate such instances.

F. One-class SVM (OCSVM) [17]

There exist different semi-supervised and unsupervised variants of One-class support vector machine (OCSVM) based anomaly detection in literature. The basic idea of this machine learning-based approach is to learn a decision boundary that achieves the maximum separation between the points and the origin. Generally, OCSVM is sensitive to the outliers when no labels are given. To tackle this shortcoming, Amer et al. (2013) [18] enhanced OCSVM for unsupervised anomaly by proposing two modifications that make the outliers contribute less to the decision boundary as compared to the normal instances. Hu et al. (2018) [19] proposed an anomaly detection

method for detecting abnormal sub-sequences in a given time-series.

G. Principle Component Analysis (PCA) [20]

PCA is a linear dimensionality reduction method that projects data to a lower dimensional space by using singular value decomposition. The possible correlated variables are converted into a set of linearly uncorrelated variables called major and minor principal components. Shyu et al. (2003) [20] proposed an anomaly detection method based on the PCA. The predictive model is generated based on the major and minor principal components of the normal data. Kwitt and Hofmann (2006) [21] also proposed a PCA based anomaly detection method. In this method, minimum covariance determinant (MCD) is employed for the computation of covariance and correlation matrix.

H. Histogram-based Outlier Score (HBOS) [22]

It is a statistical unsupervised anomaly detection method. As the name of the method indicates, this method is based on histograms for detecting anomalies in a given streaming data. First, a histogram for each feature of the data is generated. Then the inverse height of the bins it resides of all features is multiplied for each instance of the data set. HBOS provides two histogram creation modes: i) static bin sizes with a fixed bin size and ii) dynamic bin width with a fixed amount of items in each bin. This method is far less computational expensive as compared to commonly used distance-based and clustering-based anomaly detection methods.

I. Extreme Gradient Boosting Outlier Detection (XGBOD) [23]

It is a relatively new semi-supervised method for detecting anomalies. XGBOD is an ensembling method based on extreme Gradient Boosting (XGBoost) [24]. XGBoost provides a parallel tree boosting to solve many data science problems in a fast and accurate way. XGBod combines the strengths of both supervised and unsupervised machine learning methods which exploit each of their individual performance capabilities in anomaly detection. It ensembles multiple unsupervised outlier mining methods to extract useful representations of the provided data. The predictive capabilities of this method are improved as compared to the other ensembling methods by using stacking-based outlier ensembling.

J. Autoencoder (AE) [25]

AE tries to learn an approximation to the identity function, so that the output is similar to the input. It consists of two parts, encoder and decoder. The network learns how to efficiently compress the data (encoder) and how to reconstruct the data back to a representation close to the input data (decoder). In AE-based anomaly detection, AEs are used to detect anomalous instances by calculating the reconstruction error. Schreyer et al. (2017) [26] used deep autoencoders to detect anomalies in large-scale accounting data in the area of fraud detection. Amarbayasgalan et al. (2018) [27] also proposed a novelty detection technique based on deep autoencoders. Their approach computes the error threshold from deep AE model and passes to a density-based cluster. Then, density-based clustering is applied to the compressed data to get novelty groups with low density.

K. DeepAnT [28]

DeepAnT is a deep learning-based unsupervised anomaly detection technique for streaming data. This method consists of two modules. The first module, *time-series predictor* is responsible for predicting the next timestamp. The predicted value is further passed to the *anomaly detector* module. This module is responsible for tagging a data instance as a normal or anomalous instance. The predictor module is based on a convolutional neural network (CNN). In DeepAnT, CNN is trained on raw data without removing anomalies from the training data. They have used two convolutional layers, each followed by a max-pooling layer.

L. FuseAD [29]

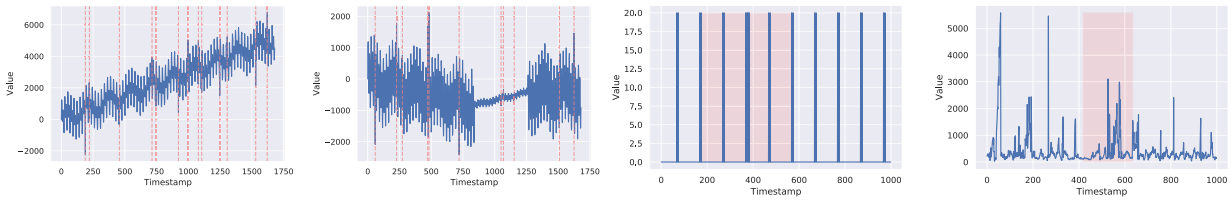
In some use-cases, statistical anomaly detection techniques robustly detect anomalies, whereas, deep learning-based anomaly detection techniques show better performance for other use-cases. To overcome the issue of choosing a better technique for a given data set, anomaly detection based on statistical models and deep learning-based models are fused in this technique. FuseAD also consists of two modules: a forecasting pipeline and an anomaly detector. In the forecasting pipeline, the forecasts of ARIMA (a statistical model) are fed to a CNN along with the actual time-series. The output of the CNN is augmented by a summation layer at the end.

IV. Data sets

In this section, we provide details of the data sets which are used for the quantitative evaluation. We have shortlisted two time-series data sets that are already labeled by the publishers. It is important to mention that only the data sets which have time-series characteristics and contains (point and contextual) anomalies are used in this study. There exist a lot of other data sets [30] which are converted to time-series from image and signal domains (e.g. *Breast Cancer Wisconsin* [30]). Such data sets are generally used for time-series classification, which is not the scope of this study.

A. Yahoo Webscope [31]

Yahoo Webscope is a publicly available streaming data set, consists of both synthetic and real data. The anonymized real data contain the Yahoo membership login data. This data set is further divided into four sets, named $A1, A2, A3$, and $A4$. Examples of time-series from this data set are shown in Fig. 1a and Fig. 1b. In these figures, red vertical lines represent the labels of anomalies. $A1, A2$, and $A3$ contain only outliers, while $A4$ also contains change-point anomalies. There are a total of 367 time-series in this data set and each sequence is comprised of $1420 - 1680$ instances.



(a) Yahoo - A3 (b) Yahoo - A4 (c) NAB - Artificial With Anomaly (d) NAB - Real Traffic

Fig. 1. Sample time-series from Yahoo Webscope (a-b) and NAB (c-d) data sets are shown in this figure. Actual streaming data are shown in blue, whereas red vertical lines and area highlight anomalous data points based on the provided labels.

B. Numenta Anomaly Benchmark (NAB) [32]

Numenta[1] open-sourced a streaming anomaly detection benchmark – NAB, in 2015. This data set consists of 58 time-series in total where each sequence is comprised of $1000 - 22000$ instances. Both, real and synthetic time-series from internet traffic, cloud services, automotive traffic, and online advertisement domains are included in this benchmark. This benchmark is labeled based on the known root cause of an anomaly and by following the labeling procedure defined by the publisher in [32]. Although this data set covers a wide range of domains, the anomaly labeling mechanism is not relevant in all the cases [33]. A window of defined size (10% of sequence size) is labeled as anomalous when an anomalous point lies within that window. In such anomalous windows, there might be only $2 - 3$ actual anomalous instances, but the whole window is labeled as anomalous. The plots shown in Fig. 1c and Fig. 1d highlight this issue of inconsistent labeling.

V. Evaluation Metrics

To analyse the anomaly detection methods from different perspectives, we have used the following evaluation metrics:

1) *Precision @ Rank n (P@n):* This simple evaluation metric evaluates an anomaly detection method by computing the True Positive Rate (TPR) for the top *n* results in a data set.
2) *Area Under ROC Curve:* A more advanced method, Receiver Operating Characteristic (ROC) curve is a graphical representation of a classification model which shows the diagnostic ability of a model. The ROC curve is created by plotting True Positive Rate (TPR) against False Positive Rate (FPR). The area under the ROC curve is reported in this study, where a value near to 1 represents a good measure of separability.
3) *Area Under Precision-Recall (PR) Curve:* An unbalanced class distribution (as in anomaly detection problems) makes the usage of the area under the ROC curve a questionable metric. To have a better performance overview of a system with an unbalanced class distribution, Precision-Recall (PR) curve is preferred. PR curve is a plot of the precision against the recall for different thresholds. In the experiments section, the area under the PR curve is reported which ranges from 0 to 1. A value close to 1 represents an accurate classifier.
4) *Inference Time:* The time required to 'infer' the results on a given test set is referred here as an inference time. The machine used to compute inference time is equipped with Intel Xeon(R) processor with 8 cores and one NVIDIA GeForce GTX 1070 GPU.

[1]https://numenta.org

VI. Experimental Setting

The initial 40% of a time-series is used to build a DL model and the rest of 60% for testing. For the distance-based and density-based anomaly detection methods that do not require a training process, we have only used the 60% of the data for consistency and fair comparison with deep learning-based methods. Since both data sets consist of multiple time-series for a particular domain, the presented results are averaged per domain/set. For all the anomaly detection methods mentioned in Section III, the default settings mentioned in [25] are used except for *DeepAnT* and *FuseAD*. For these two methods, the settings mentioned in the actual study are used.

VII. Analysis

The comparative results of distance-based, density-based, kernel-based, and deep learning-based anomaly detection methods on two streaming data sets are shown in Table I and Table II. Important evaluation metrics, the area under the ROC and PR curves are provided in Table I, while the supporting metrics, P@n and inference time are provided in Table II. For the Yahoo Webscope data set, deep learning-based anomaly detection methods are in lead by a clear margin in the terms of ROC and PR curves. For $A1$ and $A2$ data sets, kNN and LOF also show high ROC and perform on par in terms of the PR curve as compared to deep learning-based methods. It is mainly because of the relatively less complicated time-series in these data sets, also the spikes are very big in these time-series which are easily detected by distance-based methods. For $A3$ and $A4$ data sets, deep learning-based methods are way ahead of the traditional methods in terms of both ROC and PR curves. The presence of trends, seasonality, and change-points make these data sets hard for traditional anomaly detection methods. These time-series characteristics also cause low PR curve value in traditional anomaly detection settings. Deep learning-based anomaly detection methods

TABLE I
COMPARATIVE EVALUATION OF DIFFERENT ANOMALY DETECTION METHODS ON YAHOO WEBSCOPE AND NAB DATA SETS. THE AREA UNDER THE ROC AND PRECISION-RECALL (PR) CURVES ARE PROVIDED IN THIS TABLE. THE HIGHEST SCORE PER DATA SET IS SHOWN IN BOLD.

			kNN [12]	LOF [13]	COF [14]	LOCI [15]	iForest [16]	OC-SVM [18]	PCA [20]	HBOS [22]	XG-Boost [24]	XG-BOD [23]	AE [25]	DeepAnT [28]	FuseAD [29]
Yahoo Webscope	A1	ROC	0.911	0.904	0.826	0.879	0.898	0.895	0.836	0.869	0.522	0.535	0.903	0.912 ± 0.010	**0.917** ± 0.018
		PR	**0.755**	0.665	0.466	0.260	0.710	0.705	0.672	0.571	0.516	0.517	0.740	0.648 ± 0.018	0.602 ± 0.011
	A2	ROC	0.920	0.901	0.858	0.851	0.662	0.913	0.923	0.652	0.500	0.500	0.878	0.962 ± 0.006	**0.982** ± 0.010
		PR	0.742	0.742	0.671	0.150	0.435	0.733	0.698	0.434	0.503	0.503	0.719	**0.881** ± 0.006	0.833 ± 0.008
	A3	ROC	0.654	0.641	0.698	-	0.628	0.657	0.628	0.630	0.511	0.551	0.653	0.922 ± 0.007	**0.976** ± 0.014
		PR	0.264	0.251	0.309	-	0.380	0.265	0.165	0.382	0.486	0.472	0.227	**0.829** ± 0.017	0.799 ± 0.009
	A4	ROC	0.648	0.640	0.682	-	0.629	0.651	0.610	0.636	0.504	0.532	0.661	0.870 ± 0.007	**0.935** ± 0.018
		PR	0.203	0.201	0.260	-	0.235	0.200	0.175	0.270	0.485	0.419	0.196	**0.642** ± 0.010	0.632 ± 0.008
Numenta Anomaly Benchmark (NAB)	Artificial	ROC	0.560	**0.605**	0.527	0.540	0.565	0.582	0.589	0.550	0.500	0.500	0.515	0.548 ± 0.010	0.545 ± 0.006
	With Ano.	PR	0.449	0.419	0.356	0.340	0.353	0.415	0.350	0.405	**0.609**	**0.609**	0.290	0.209 ± 0.005	0.196 ± 0.003
	Real Ad	ROC	0.530	0.504	0.503	0.499	0.520	0.519	0.415	0.510	0.511	0.532	0.477	0.562 ± 0.004	**0.590** ± 0.006
	Exchange	PR	0.146	0.003	0.171	0.143	0.139	0.143	0.120	0.149	**0.396**	0.215	0.134	0.156 ± 0.004	0.154 ± 0.001
	Real AWS	ROC	0.539	0.525	0.512	-	0.539	0.561	0.520	0.554	0.507	0.486	0.540	**0.583** ± 0.006	0.572 ± 0.002
	Cloud	PR	0.211	0.189	0.209	-	0.240	0.241	0.193	0.267	**0.466**	0.345	0.333	0.197 ± 0.003	0.190 ± 0.007
	Watch														
	Real	ROC	0.632	0.578	0.493	-	**0.647**	0.637	0.596	0.641	0.519	0.544	0.613	0.608 ± 0.009	0.595 ± 0.012
	Know. Cause	PR	0.376	0.291	0.206	-	0.340	0.325	0.291	0.373	**0.515**	0.453	0.411	0.259 ± 0.011	0.236 ± 0.013
	Real	ROC	0.578	0.561	0.522	-	0.572	0.577	0.517	0.585	0.503	0.529	0.569	**0.630** ± 0.009	0.620 ± 0.004
	Traffic	PR	0.363	0.296	0.291	-	0.324	0.333	0.290	0.338	**0.570**	0.544	0.377	0.269 ± 0.004	0.262 ± 0.004
	Real	ROC	0.524	0.521	0.526	-	**0.553**	0.550	0.478	0.520	0.511	0.524	0.539	0.551 ± 0.002	0.547 ± 0.001
	Tweets	PR	0.207	0.169	0.197	-	0.138	0.140	0.114	0.183	**0.342**	0.179	0.193	0.123 ± 0.123	0.119 ± 0.119

showed around 29% improvement in the ROC as compared to the best traditional anomaly detection method (COF) for $A3$ data set and 25% improvement for $A4$ data set. There are also noticeable improvements in terms of the PR curve for $A3$ and $A4$ data sets which actually shows the robustness of deep learning-based methods in the streaming data. The PR curve increased 34% in deep learning-based methods as compared to $XGBoost$ for $A3$ data set and 16% for $A4$ data set. In terms of P@n metric, the same improvement trend is observed for deep learning-based methods. For $A4$ data set, even an improvement of 50% in P@n is observed in deep learning-based methods. One downside of using deep learning-based anomaly detection methods for this data set is a relatively high inference time. Although the inference time of deep learning-based methods is on par or even better than some statistical and distance-based methods, it is not the minimum inference time. For the Yahoo Webscope data set, PCA detects anomalies in minimum time. In comparison with other traditional anomaly detection methods, PCA performance is quite good. It's ROC, PR curve, and P@n is on par with other traditional methods, but it is far more superior than others in terms of producing results. $LOCI$ turns out to be a bad choice for anomaly detection because of its moderate results and very high time complexity. Due to its high time complexity, we are unable to report the results for all of the data sets.

The overall performance of all anomaly detection methods is not very convincing on the NAB data set. It is not due to the incompetence of these methods, but the labeling mechanism used in this data set. For this data set, a mix performance of anomaly detection methods is observed. There is no clear winner for this data set as all methods perform on par. For some domains of this data set, deep learning-based methods perform better in terms of ROC, whereas the traditional methods perform better in terms of other evaluation metrics. For this data set, $HBOS$ provides anomaly results in minimum time and kNN has maximum P@n in most of the cases.

VIII. CONCLUSION

The recent advancements in DL have also initiated the development of deep learning-based anomaly detection methods. In this study, we have compared 13 anomaly detection methods on two streaming data sets. Our contribution in this study is the comparison and analysis of different anomaly detection methods on streaming data sets. In other comparative studies available in literature [7], the analysis of anomaly detection methods is performed on the data sets which are converted from image and signal domains. Furthermore, we have included deep learning-based anomaly detection in the comparison in addition to the commonly used distance-based and density-based anomaly detection methods. The results show that deep learning-based anomaly detection methods are superior to other methods in most of the evaluation metrics for the Yahoo Webscope data set. These methods perform way better than other methods for time-series which have trends and seasonality in them. In terms of time complexity, PCA performed better than deep learning-based methods. For the NAB data set, there is no clear winner because of the labeling issue in this data set. For both data sets, $LOCI$ is not a good choice because of its high time complexity. In the future, we aim to extend the comparison to other anomaly detection methods on streaming data sets, especially the methods which are performing good in the image modality.

TABLE II

COMPARATIVE EVALUATION OF DIFFERENT ANOMALY DETECTION METHODS ON THE BASIS OF PRECISION @ RANK n (P) AND THE INFERENCE TIME (T). THE TIME IS REPORTED IN SECONDS. FOR EACH DATA SET, THE MAXIMUM P@n AND THE MINIMUM INFERENCE TIME ARE SHOWN IN BOLD.

			kNN [12]	LOF [13]	COF [14]	LOCI [15]	iForest [16]	OC-SVM [18]	PCA [20]	HBOS [22]	XG-Boost [24]	XG-BOD [23]	AE [25]	DeepAnT [28]	FuseAD [29]
Yahoo Webscope	A1	P	0.7275	0.6517	0.4645	0.2491	0.5278	0.6166	0.6678	0.2327	0.0000	0.0345	**0.7387**	0.5656 ± 0.0087	0.5340 ± 0.0015
		T	0.0782	0.0025	1.8057	2155.1	0.0604	0.0074	**0.0002**	0.0245	0.0012	3.4775	0.0362	0.0457	0.0486
	A2	P	0.7200	0.7200	0.6472	0.0835	0.0100	0.6700	0.6817	0.0000	0.0000	0.0000	0.7000	**0.8738** ± 0.0045	0.8147 ± 0.0074
		T	0.0757	0.0022	1.7474	2196.7	0.0560	0.0078	**0.0003**	0.0011	0.0011	3.4474	0.1563	0.0466	0.0491
	A3	P	0.2570	0.2498	0.3136	-	0.1753	0.2579	0.1628	0.2030	0.0084	0.1052	0.2210	**0.8136** ± 0.0190	0.7723 ± 0.0069
		T	0.0907	0.0025	2.6382	-	0.0604	0.0108	**0.0002**	0.0012	0.0017	3.8734	0.3721	0.0608	0.0639
	A4	P	0.2022	0.2012	0.2670	-	0.1207	0.2011	0.1731	0.0889	0.0306	0.0486	0.2029	**0.6178** ± 0.0102	0.6122 ± 0.0003
		T	0.0911	0.0026	2.5313	-	0.0624	0.0108	**0.0003**	0.0011	0.0017	3.8460	0.5825	0.0584	0.0607
Numenta Anomaly Benchmark (NAB)	Artificial	P	0.3302	0.3722	0.2637	0.3076	0.3654	0.3518	0.3469	**0.3926**	0.0000	0.0000	0.2544	0.1972 ± 0.0142	0.1882 ± 0.0033
	With Ano.	T	0.2107	0.0105	9.4820	44322.9	0.0906	0.0341	0.0003	**0.0002**	0.0017	6.7487	0.3467	0.1336	0.1420
	Real Ad	P	0.1231	0.1213	0.0719	0.0877	0.1177	0.1228	0.0842	0.1274	0.0725	0.0724	0.0567	**0.1641** ± 0.0044	0.1603 ± 0.0044
	Exchange	T	0.0913	0.0057	1.6342	1533.4	0.0671	0.0063	0.0003	**0.0001**	0.0022	2.7408	0.1607	0.0603	0.0664
	Real AWS	P	**0.2171**	0.1718	0.1296	-	0.1792	0.1745	0.1998	0.1880	0.0574	0.0279	0.1391	0.1974 ± 0.0031	0.1901 ± 0.0080
	Cloud Watch	T	0.2176	0.0088	10.549	-	0.0883	0.0417	0.0003	**0.0002**	0.0033	7.1031	0.4798	0.1407	0.1537
	Real	P	**0.3375**	0.2674	0.1131	-	0.2881	0.3056	0.2848	0.3368	0.0716	0.0876	0.1115	0.2705 ± 0.0144	0.2544 ± 0.0118
	Know. Cause	T	0.4998	0.0168	101.82	-	0.1612	0.3609	0.0004	**0.0003**	0.0060	20.2553	1.3271	0.2830	0.2997
	Real	P	0.3746	0.3159	0.2740	-	0.3126	0.3217	0.2643	**0.3843**	0.0255	0.0700	0.1646	0.2493 ± 0.0041	0.2495 ± 0.0019
	Traffic	T	0.1253	0.0038	3.3476	-	0.0718	0.0124	0.0003	**0.0002**	0.0014	3.7763	0.3248	0.0888	0.0946
	Real	P	**0.3004**	0.2331	0.2204	-	0.1633	0.1644	0.1480	0.2322	0.1937	0.1616	0.0960	0.1485 ± 0.0024	0.1416 ± 0.0015
	Tweets	T	0.9121	0.0734	155.90	-	0.2242	0.6048	0.0004	**0.0003**	0.0160	34.7278	2.4480	0.4569	0.5019

REFERENCES

[1] D. M. Hawkins, *Identification of outliers*. Springer, 1980, vol. 11.

[2] M. Goldstein, "Anomaly detection in large datasets," PhD-Thesis, University of Kaiserslautern, München, Germany, 2 2014.

[3] M. Munir, S. Erkel, A. Dengel, and S. Ahmed, "Pattern-based contextual anomaly detection in hvac systems," in *ICDMW*. IEEE, 2017.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[5] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[7] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PloS one*, 2016.

[8] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2013.

[9] B. Kiran, D. Thomas, and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *Journal of Imaging*, vol. 4, no. 2, p. 36, 2018.

[10] A. O. Adewumi and A. A. Akinyelu, "A survey of machine-learning and nature-inspired based credit card fraud detection techniques," *International Journal of System Assurance Engineering and Management*, vol. 8, no. 2, pp. 937–953, 2017.

[11] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[12] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2002, pp. 15–27.

[13] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *ACM sigmod record*. ACM, 2000.

[14] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *PAKDD*. Springer, 2002, pp. 535–548.

[15] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *International Conference on Data Engineering*. IEEE, 2003, pp. 315–326.

[16] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, p. 3, 2012.

[17] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[18] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *ACM SIGKDD Workshop on Outlier Detection and Description*. ACM, 2013.

[19] M. Hu, Z. Ji, K. Yan, Y. Guo, X. Feng, J. Gong, X. Zhao, and L. Dong, "Detecting anomalies in time series data via a meta-feature based approach," *IEEE Access*, vol. 6, pp. 27 760–27 776, 2018.

[20] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," Miami University Coral Gables FL, Tech. Rep., 2003.

[21] R. Kwitt and U. Hofmann, "Robust methods for unsupervised PCA-based anomaly detection," *Proc. of IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation*, pp. 1–3, 2006.

[22] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm," *KI-2012*, 2012.

[23] Y. Zhao and M. K. Hryniewicki, "XGBOD: improving supervised outlier detection with unsupervised representation learning," in *IJCNN*, 2018.

[24] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *KDD*. ACM, 2016, pp. 785–794.

[25] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python Toolbox for Scalable Outlier Detection," *Journal of Machine Learning Research*, 2019.

[26] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer, "Detection of anomalies in large scale accounting data using deep autoencoder networks," *arXiv preprint arXiv:1709.05254*, 2017.

[27] T. Amarbayasgalan, B. Jargalsaikhan, and K. Ryu, "Unsupervised novelty detection using deep autoencoders with density based clustering," *Applied Sciences*, vol. 8, no. 9, p. 1468, 2018.

[28] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2018.

[29] M. Munir, S. A. Siddiqui, M. A. Chattha, A. Dengel, and S. Ahmed, "FuseAD: Unsupervised Anomaly Detection in Streaming Sensors Data by Fusing Statistical and Deep Learning Models," *Sensors*, 2019.

[30] G. Casalicchio, J. Bossek, M. Lang, D. Kirchhoff, P. Kerschke, B. Hofner, H. Seibold, J. Vanschoren, and B. Bischl, "OpenML: An R package to connect to the machine learning platform OpenML," *Computational Statistics*, vol. 32, no. 3, 2017.

[31] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *International Conference on Knowledge Discovery and Data Mining*. ACM, 2015.

[32] A. Lavin and S. Ahmad, "Evaluating Real-Time Anomaly Detection Algorithms–The Numenta Anomaly Benchmark," in *ICMLA*, 2015.

[33] N. Singh and C. Olinsky, "Demystifying numenta anomaly benchmark," in *IJCNN*. IEEE, 2017, pp. 1570–1577.