Evaluation of MemStream: Memory-Based Streaming Anomaly Detection

Omkar Bhandare, CSE Department, IIT Kharagpur

Abstract

Anomaly detection in streaming data is crucial for many real-world applications, such as network security, fraud detection, and system monitoring. However, streaming data often exhibit concept drift, meaning the data distribution changes over time. This poses a significant challenge for anomaly detection algorithms, as they need to adapt to the evolving data to maintain high detection accuracy. Existing streaming anomaly detection algorithms lack a unified evaluation framework that can assess their performance and robustness under different concept drift and anomaly types. MemStream [13], a streaming anomaly detection framework, allow us to detect unusual events as they occur while being resilient to concept drift. MemStream leverages the power of a denoising autoencoder to learn representations and a memory module to learn the dynamically changing trend in data without the need for labels. Experimental results show the effectiveness of MemStream compared to state-of-the-art streaming baselines using 1 synthetic dataset and 6 real-world datasets.

Introduction

Anomaly detection is a fundamental and well-studied problem in many areas, such as cybersecurity [1, 2], video surveillance [3, 4], financial fraud [5] and healthcare[6]. Traditional classifiers trained in supervised learning settings do not work well in anomaly detection because of the cold-start problem; therefore, anomaly detectors are trained in an unsupervised setting where the normal data distribution is learned, and instances that appear unlikely under the current distribution are identified as anomalous. The anomaly detection problem becomes even more challenging when the data arrives in a streaming/online manner, and anomalies need to be detected in real-time. Moreover, in streaming data, there can be a drift in the distribution over time, which the existing approaches [7, 8, 9, 10, 11, 12] are unable to handle fully. To handle concept drift in a streaming setting, MemStream uses an explicit memory module. The introduction of memory module, with an appropriate update strategy, seems to tackle some of the issues in streaming anomaly detection with concept drift. MemStream, uses a denoising autoencoder [14] to extract features, and a memory module to learn dynamically changing trend, thereby avoiding the over-generalization of autoencoders (i.e. the problem of autoencoders reconstructing anomalous samples well).

Problem Definition

This report thoroughly investigates MemStream, a novel anomaly detection algorithm designed to operate efficiently on in-memory data streams. The report delves into the inner workings of MemStream and compares its effectiveness against several well-established open-source streaming anomaly detection algorithms. This comparative analysis is conducted using a variety of publicly accessible datasets, providing a robust evaluation of MemStream's relative strengths and weaknesses in real-world scenarios.

MemStream Process

MemStream is a novel algorithm that leverages a memory-augmented feature extractor. This extractor is trained beforehand on a limited sample of normal data [13]. The training process equips the extractor to capture the inherent structure of normal data points. Furthermore, MemStream incorporates a memory module specifically designed to store encodings of these normal instances.

MemStream's approach to handling concept drift refers to evolving data patterns within a stream. The algorithm continuously monitors deviations from the initial data distribution by leveraging its internal memory component. A First-In-First-Out (FIFO) replacement policy is employed within the memory to ensure temporal coherence when dealing with gradual concept drift. A key aspect of MemStream involves a selective retention process for incoming data points. This selection is determined by calculating a discounted similarity score between each new data point and existing entries within the memory. This score is compared to a predefined threshold (β) [13]. If the score falls below β, the new record is incorporated into the memory using the FIFO strategy. Notably, the analysis reveals that data points originating from the same underlying distribution tend to cluster closely based on these scores. Despite the absence of explicit anomaly labels, this clustering behaviour proves to be instrumental in anomaly detection. In essence, MemStream dynamically maintains normal data representations while utilising score variations to identify anomalies. This approach offers a robust method for anomaly identification without requiring pre-labelled anomalous data points.

Evaluation Process

MemStream assigns predicted labels to data points based on scores generated by the algorithm. These labels are determined by comparing a predefined threshold (often established empirically) with each data point's score. The threshold is typically set by iteratively comparing scores of incoming data points to those of previously confirmed non-anomalous points. These predicted labels are essential for evaluating the algorithm's performance, allowing for the identification of both false positives (incorrectly classified anomalies) and false negatives (missed anomalies).

To assess MemStream's efficacy, comparative studies are conducted with established anomaly detection models such as KDE [18], PCA [19], LOF [20], CBLOF [21], HOBS [22], kNN [23], IForest [24], LODA [25], and LUNAR [26], all whose implementations have been adopted from PyOD [27]. Standard evaluation metrics like ROC-AUC [16] and AUC-PR [17] are employed in this evaluation process. The MemStream implementation utilised in this study is derived from the original authors' GitHub repository [15], with necessary modifications made to address specific operational requirements. Through extensive experimentation, algorithm parameters are meticulously fine-tuned to optimise performance across various datasets. Notably, a new parameter named 'limit' has been introduced to serve as a threshold for data point labelling. This addition enhances MemStream's adaptability and precision in anomaly detection tasks. This rigorous process of iterative refinement and parameterisation ensures that MemStream remains competitive and effective in real-world applications, offering robust anomaly detection capabilities particularly suited for environments with dynamic data streams.

Experimental Observations

The following tables present observations and corresponding conclusions derived from an analysis of the original MemStream papers and the outputs of benchmark models. As indicated in Table 1, MemStream might not be a universally optimal solution for streaming anomaly detection. The algorithm's core strength lies in its ability to dynamically adjust its internal memory based on data point scores, enabling it to adapt to evolving data distributions. This characteristic makes MemStream particularly effective in datasets exhibiting distinct distributions and concept drift, as evidenced by its performance on the Synthetic dataset [13]. However, relying on score-driven memory updates can lead to variable performance across different datasets. This variability may limit effectiveness in scenarios with less structured data distributions or rapid concept changes. Nonetheless, MemStream's superior

performance in specific contexts underscores its potential for applications involving datasets with identifiable patterns and gradual shifts in data characteristics.

	Synthetic	Ionosphere	Cardio	Statlog	Satimage-2	Mammography	Pima
MemStream	0.9553	0.8187	0.8720	0.7239	0.9948	0.9020	0.7407
KDE	0.5524	0.9538	0.9856	1.0000	1.0000	0.8716	1.0000
PCA	0.5188	0.8982	0.9660	0.6623	0.9784	0.8956	0.7163
LOF	0.6200	0.9574	0.9466	0.8611	0.9944	0.8638	0.6866
CBLOF	0.5491	0.9839	0.9487	0.8820	0.9980	0.8356	0.6622
HBOS	0.5483	0.7707	0.8751	0.8690	0.9713	0.8524	0.7031
KNN	0.6459	0.9781	0.9388	0.8876	0.9993	0.8809	0.7126
IForest	0.5528	0.9262	0.9472	0.8113	0.9924	0.8835	0.7431
LODA	0.5328	0.8373	0.9573	0.7032	0.9864	0.8802	0.6547
LUNAR	0.7277	0.9594	0.9971	0.8288	0.9964	0.8451	0.6144

Table 1 : ROC-AUC Values of Algorithms against Datasets

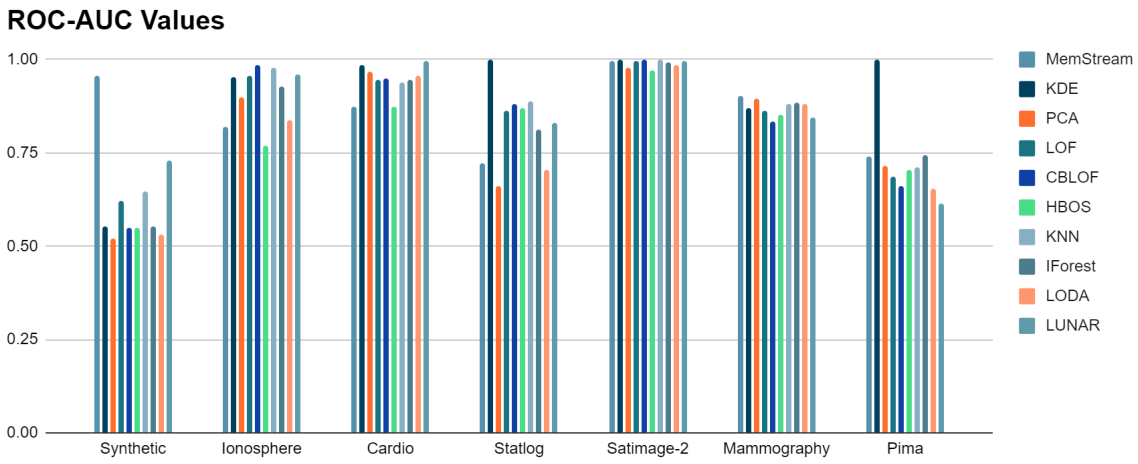
An examination of Table 2 reveals that MemStream generally exhibits lower precision scores than leading algorithms across most datasets, except for the Synthetic dataset. This observation is attributable to MemStream's higher rates of both False Positives (incorrectly identified anomalies) and False Negatives (missed anomalies), which negatively impact its precision metrics. However, it is noteworthy that MemStream consistently outperforms many other algorithms across all datasets. This pattern suggests MemStream's reliability and effectiveness as a viable choice for anomaly detection tasks. While MemStream may exhibit limitations in precision compared to specific benchmark models, its overall performance across diverse datasets underscores its robustness and applicability in real-world scenarios demanding adaptable anomaly detection solutions. Tables 3 and 4 illustrate the frequency of false positives and false negatives observed in the MemStream algorithm and other algorithms across various datasets. A consistent trend of elevated false positives emerges, with some exceptions. This propensity might be attributed to the absence of an automated threshold optimization mechanism for data point labeling within MemStream. The current approach relies on manual adjustment through iterative testing, which could potentially lead to these outcomes. Furthermore, the consistently suboptimal AUC-PR scores suggest inherent limitations in MemStream's ability to minimize false classifications. This could be a consequence of the underlying assumption that all data points stored in memory share a uniform distribution. This assumption becomes problematic in scenarios with concept drift. When the memory

	Synthetic	Ionosphere	Cardio	Statlog	Satimage-2	Mammography	Pima
MemStream	0.8211	0.6544	0.4812	0.6818	0.9221	0.2258	0.5530
KDE	0.1944	0.9415	0.9561	1.0000	0.9999	0.3282	1.000
PCA	0.1553	0.8623	0.7578	0.6867	0.8821	0.2907	0.5443
LOF	0.2215	0.9283	0.6510	0.8374	0.9255	0.2431	0.5365
CBLOF	0.1597	0.9766	0.6322	0.8180	0.9657	0.1830	0.5183
HBOS	0.1263	0.6105	0.5444	0.8139	0.7343	0.1496	0.5415
KNN	0.2353	0.9679	0.6908	0.8473	0.9762	0.3098	0.5678
IForest	0.1800	0.8832	0.6566	0.7779	0.8998	0.2709	0.5832
LODA	0.1582	0.7664	0.7338	0.7281	0.8976	0.3015	0.4486
LUNAR	0.3032	0.9507	0.9742	0.7906	0.9543	0.3584	0.4443

Table 2 : AUC-PR Values of Algorithms against Datasets

contains data representing various stages of drift, anomalies might be misconstrued as drift-related fluctuations, impacting subsequent classifications. Additionally, frequent concept drift occurrences may lead the model to misclassify normal data points as anomalies during transition periods, consequently amplifying errors downstream. Addressing these challenges could significantly improve MemStream's accuracy and reliability in anomaly detection tasks characterized by dynamic data distributions.

The analysis of these tables suggests that MemStream excels at handling infrequent, well-defined concept drifts. However, its adaptation capability diminishes in datasets exhibiting variable drift rates. In comparison, the KDE algorithm emerges as the superior choice, consistently outperforming MemStream across diverse datasets. Notably, KDE achieves near-perfect results on the Pima, Statlog, and Statimage-2 datasets, highlighting its robustness in handling dynamic data distributions.



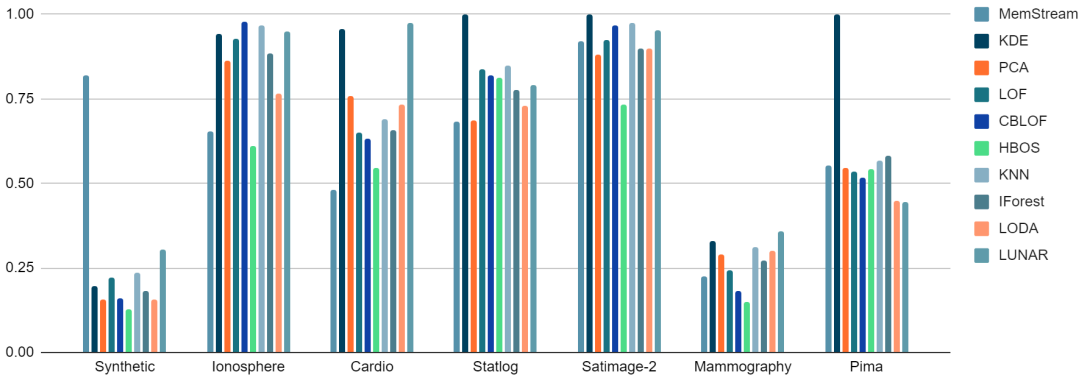
	Synthetic	Ionosphere	Cardio	Statlog	Satimage-2	Mammography	Pima
MemStream	701	163	223	1623	1310	213	0
KDE	900	23	166	0	0	1093	0
PCA	90	23	166	440	574	1093	50
LOF	734	19	143	383	495	992	49
CBLOF	900	23	166	440	574	1093	50
HBOS	504	23	166	440	574	1083	50
KNN	552	18	133	396	498	944	41
IForest	899	23	166	440	574	1093	50
LODA	680	23	166	440	574	1093	50
LUNAR	900	23	166	440	574	1093	50

Table 3 : False Positives of Algorithms against Datasets

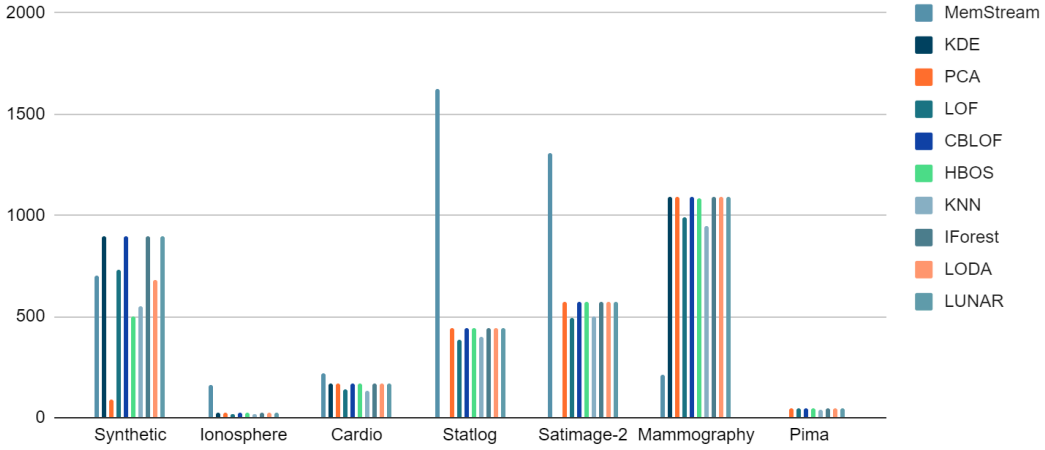
	Synthetic	Ionosphere	Cardio	Statlog	Satimage-2	Mammography	Pima
MemStream	72	20	143	533	3	204	262
KDE	826	19	9	0	0	106	0
PCA	866	40	19	952	5	79	197
LOF	798	17	31	613	2	103	184
CBLOF	816	5	23	661	0	118	180
HBOS	867	90	71	688	5	110	190
KNN	818	12	45	589	0	90	186
IForest	826	31	16	760	2	85	186
LODA	868	33	59	831	2	73	226
LUNAR	659	13	1	689	1	123	219

Table 4 : False Negatives of Algorithms against Datasets

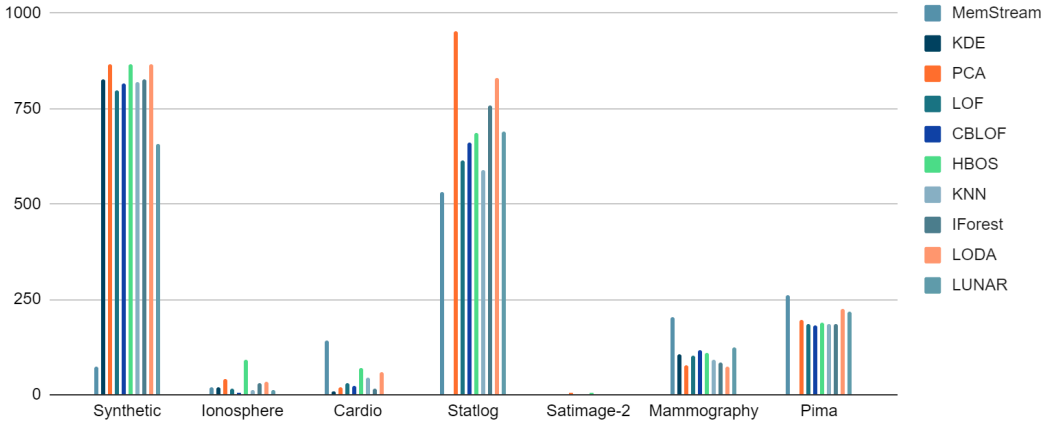
AUC-PR Values



False Positives



False Negatives



Conclusion

MemStream, a novel memory-based anomaly detection algorithm, adeptly manages concept drift in data streams. Although it does not achieve perfect results in every situation, MemStream consistently delivers satisfactory performance across diverse datasets. Empirical evaluations affirm its robustness, establishing it as a viable solution for anomaly detection tasks. The algorithm remains competitive with leading methods, though its efficacy can be influenced by dataset characteristics and the frequency of concept drift. MemStream's performance frequently approaches optimal outcomes, underscoring its reliability and applicability in practical scenarios. Future research aimed at optimising MemStream's parameters and methodologies promises to enhance its versatility and effectiveness in various anomaly detection contexts. These advancements could reinforce MemStream's status as a valuable tool in data-driven anomaly detection and concept drift management.

References

[1] Siddharth Bhatia, Bryan Hooi, Minji Yoon, Kijung Shin, and Christos Faloutsos. 2020. MIDAS: Microcluster-Based Detector of Anomalies in Edge Streams. In AAAI.

[2] Swee Chuan Tan, Kai Ming Ting, and Tony Fei Liu. 2011. Fast Anomaly Detection for Streaming Data. In IJCAI.

[3] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. 2010. Anomaly detection in crowded scenes. CVPR (2010

[4] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. S. 2019. Training Adversarial Discriminators for Cross-Channel Abnormal Event Detection in Crowds. WACV (2019).

[5] Abhinav Srivastava, Amlan Kundu, Shamik Sural, and Arun Majumdar. 2008. Credit card fraud detection using hidden Markov model. TDSC (2008).

[6] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In IPMI

[7] Siddharth Bhatia, Arjit Jain, Pan Li, Ritesh Kumar, and Bryan Hooi. 2021. MSTREAM: Fast Anomaly Detection in Multi-Aspect Streams. TheWebConf (WWW) (2021).

[8] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. 2016. Robust Random Cut Forest Based Anomaly Detection on Streams. In ICML.

[9] S. Hariri, M. Kind, and R. Brunner. 2021. Extended Isolation Forest. TKDE (2021).

[10] Emaad Manzoor, Hemank Lamba, and Leman Akoglu. 2018. xStream: Outlier Detection in Feature-Evolving Data Streams. KDD (2018).

[11] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. NDSS (2018).

[12] Gyoung S Na, Donghyun Kim, and Hwanjo Yu. 2018. DILOF: Effective and Memory Efficient Local Outlier Detection in Data Streams. KDD (2018).

[13] Siddharth Bhatia, Arjit Jain, Shivin Srivastava, Kenji Kawaguchi, and Bryan Hooi. 2022. MemStream: Memory-Based Streaming Anomaly Detection.

[14] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. 2008. Extracting and Composing Robust Features with Denoising Autoencoders. In ICML.

[15] Stream-AD/MemStream: MemStream: Memory-Based Streaming Anomaly Detection (github.com)

[16] Andrew P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, 1996

[17] Kendrick Boyd, Kevin H. Eng, and C. David Page, Area Under the Precision-Recall Curve: Point Estimates and Confidence Intervals, Springer-Verlag Berlin Heidelberg 2013

[18] Latecki, L.J., Lazarevic, A. and Pokrajac, D., 2007, July. Outlier detection with kernel density functions. In International Workshop on Machine Learning and Data Mining in Pattern Recognition (pp. 61-75). Springer, Berlin, Heidelberg.

[19] Shyu, M.L., Chen, S.C., Sarinnapakorn, K. and Chang, L., 2003. A novel anomaly detection scheme based on principal component classifier. *MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING*.

[20] Breunig, M.M., Kriegel, H.P., Ng, R.T. and Sander, J., 2000, May. LOF: identifying density-based local outliers. *ACM Sigmod Record*, 29(2), pp. 93-104.

[21] He, Z., Xu, X. and Deng, S., 2003. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10), pp.1641-1650.

[22] Goldstein, M. and Dengel, A., 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. In *KI-2012: Poster and Demo Track*, pp.59-63.

[23] Ramaswamy, S., Rastogi, R. and Shim, K., 2000, May. Efficient algorithms for mining outliers from large data sets. *ACM Sigmod Record*, 29(2), pp. 427-438.

[24] Liu, F.T., Ting, K.M. and Zhou, Z.H., 2008, December. Isolation forest. In *International Conference on Data Mining*, pp. 413-422. IEEE.

[25] *(1, 2)* Pevný, T., 2016. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2), pp.275-304.

[26] Goodge, A., Hooi, B., Ng, S.K. and Ng, W.S., 2022, June. Lunar: Unifying local outlier detection methods via graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence.

[27] Zhao, Y., Nasrullah, Z. and Li, Z., 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. Journal of machine learning research (JMLR), 20(96), pp.1-7.