Towards Anomaly Detection using Multiple Instances of Micro-Cluster Detection

Rafael Copstein
Faculty of Computer Science
Dalhousie University
Halifax, Canada
rafael.copstein@dal.ca

Bradley Niblett
Executive Consultant
2Keys Corporation
Ottawa, Canada
bniblett@2keys.ca

Andrew Johnston
VP Industry Relations
2Keys Corporation
Ottawa, Canada
ajohnston@2keys.ca

Jeff Schwartzentruber
Faculty of Computer Science
Dalhousie Univeristy
Halifax, Canada
jeffrey.schwartzentruber@gmail.com

Malcolm Heywood
Faculty of Computer Science
Dalhousie University
Halifax, Canada
mheywood@cs.dal.ca

Nur Zincir-Heywood
Faculty of Computer Science
Dalhousie University
Halifax, Canada
zincir@cs.dal.ca

***Abstract*—One of the resources used in anomaly detection on log data is graph based approaches. Connections between adjacent log entries, co-occurrence of attributes, and other relations can be easily represented using graphs. In this paper, using a state-of-the-art (SOTA) graph based anomaly detection method, we reproduce and show the limitations on publicly available log data. Then we introduce a novel method, MIMC, that improves on the detection rates without causing a considerable loss in the overall performance. In order to evaluate the performance of MIMC, we perform experiments over the same datasets used in SOTA. The results indicate that MIMC has merit as a graph-based anomaly detection system over different types of log data. We believe that this is an important achievement on the road to building an unsupervised and online approach.**

***Index Terms*—Graph based approaches, micro-clustering, anomaly detection, log analysis.**

I. Introduction

Networks and services are, commonly, targets of a myriad of cyber attacks including, but not limited to, distributed denial of service, code injection and data exfiltration. Attacks like these can result in financial losses to businesses and, possibly, the extraction of personal or sensitive data by the hands of malicious actors. For that reason, businesses have invested increasingly more in solutions for detection and prevention of these kinds of attacks [1]. Solutions that make use of pre-existing processes or data tend to spark more interest given that they require less service disruption in order to be implemented. One such kind of process that is commonly in place by most businesses is the capture and collection of logs.

Logs are used in networks and services to describe actions being performed by the system. They can be used to register attempted actions by a user, specific routines being called within the code, or describe the state of the system. Capturing logs is an essential tool for debugging, monitoring, and improving systems, and therefore it is commonly implemented. Attempting to detect attacks using log information is not new. Considering that logs describe the actions being performed on a system, detecting an attack is analogous to finding anomalous actions being performed. This concept is usually studied under the area of anomaly detection [2].

One of the resources used in anomaly detection on log data is graph based approaches [3]. Connections between adjacent log entries, co-occurrence of attributes, and other relations can be easily represented using graphs. These representations can, then, be analyzed for the presence of cliques, paths, subgraphs, among others, which are the foundation of some graph-based anomaly detection techniques [4].

In this work, using a state-of-the-art (SOTA) graph based anomaly detection method [5], we replicate the evaluations of the SOTA system using the same datasets in order to understand the wider context and identify any potential limitations. Based on the replication studies results, we introduce a novel method, MIMC, that improves on the detection rates without causing a considerable loss in the overall performance.

The remainder of the paper is organized as follows: section II summarizes the related work on graph based anomaly detection. Section III describes the SOTA method that is reproduced. Section IV introduces the proposed novel method, MIMC. Section V presents the evaluations and results. Finally, conclusions are drawn and the future work is discussed in section VI.

II. Related Works

Our work is closely related to the areas of log analysis and anomaly detection on log data – in particular, graph-based approaches for anomaly detection. In the following, we give an overview of the related works in the area.

The work of Noble & Cook [3] proposes two methods for graph-based anomaly detection making use of a method for detecting recurring substructures in graphs, namely Subdue. When tested over the 1999 KDD Cup data, the methods show reasonable results in identifying some of the attacks, albeit having extremely poor performance for other attacks.

The solution presented in Kurniawan et. al. [6], namely VloGraph, makes use of existing knowledge sources to connect logs and information collected *a priori* into a knowledge graph. This graph is, then, available for analysis using a query language, SPARQL, in order to retrieve events of interest.

The use of high-order networks compared to first-order networks for anomaly detection is explored by Saebi & Xu et. al [7]. Their work shows that first-order networks are not as effective as high-order networks when detecting high-order anomalies.

Kulkarni et. al. [8] explore the patterns found by creating different kinds of graphs over insider trading data. These include networks of traders, purchases, and sales of stocks. Furthermore, they explore the idea of anomaly detection using hyper-graphs, reaching the conclusion that, given the complexity of the domain, it is hard to evaluate the performance of their model. However, they are able to confirm that the hyper-edges identified as anomalies (insider trading) do, in fact, result in profit for the trader in majority of the cases.

Moreover, Mongiovi et. al. proposed NetSpot [9] for finding anomalous regions on dynamic networks. These include traffic networks, social networks, or knowledge networks. They show that NetSpot is up to one order of magnitude faster than an exhaustive search approach and yielding results within 5%.

He at. al. [10] analyzed six methods for anomaly detection using log data: three supervised and three unsupervised. The three supervised methods were based on Logistic Regression, Decision Trees, and Support Vector Machines (SVM). The three unsupervised methods include Clustering, Principal Component Analysis, and Invariant Mining. In terms of accuracy, SVM achieved the highest F-Measure among the supervised methods. Out of the unsupervised methods, Invariant Mining was the one with the highest F-Measure.

In the work of Uno et. al. [11], we see an introduction to the problem of *micro-clustering* as unsupervised soft-clustering. Here, the problem is clustering highly-related entries as opposed to highly-dense ones. They propose a methodology called *data polishing*, to reduce the number of yielded clusters while maintaining the high relation between entries.

On the other hand, Farzad and Gulliver [12] propose a method for unsupervised anomaly detection in system logs. They employ an Isolation Forest algorithm and two deep Autoencoder networks. When evaluated over system logs from machines such as Blue-Gene II and Thunderbird, the proposed method outperforms comparable techniques such as Gaussian Mixture Model and One-Class Support Vector Machine.

In [13], Zhang et. al. introduce LogRobust, an anomaly detection technique that uses an extracted semantic vector to represent each log entry. It is argued that, by doing so, the method remains robust against anomalous events not previously observed in training / historical data.

LogBERT, introduced in the work of Guo et. al. [14], makes use of BERT to run a self-supervised training to learn normal sequences of log masks, that is, masks yielded by a log abstraction process. Sequences of masks that do not match the trained normal sequences are deemed anomalous.

As we can see, there is a clear preference for unsupervised methods given that they don't require prior training and, as such, don't require a training sample of data. Methods relying on machine learning models tend to be unfriendly when it comes to performing root-cause analysis over an incident. Graph-based methods are easier to analyze and, in most cases, provide a reasonable level of explainability to any flagged anomalies. The use of language models, albeit promising, relies on the semantic value of log entries which may not always be intended for readability. Therefore, in our work, we aim to develop an unsupervised, and online method that makes use of graph representations of data to detect anomalies in log data.

III. Micro-cluster Anomaly Detection

In anomaly detection, presenting potential anomalies in the form of outliers as separate micro-clusters is informative to human experts in many real world applications. Several recent works in the literature have leveraged micro-cluster based anomaly detection [15], [16]. We have employed MIDAS [5] as a representative of the state-of-the-art (SOTA) in micro-cluster based anomaly detection. MIDAS is an online method for detecting *micro-cluster anomalies*, that is, rapidly arriving groups of similar edges in a dynamic graph. This is particularly useful for detecting events such as distributed denial of service attacks (DDoS) in network traffic data. It works by processing incoming network packets as directed edges between the packet's source IP and destination IP and feeding them into a dynamic graph. By keeping track of the frequency of each edge, MIDAS makes use of a chi-squared test in order to yield an anomaly score, which is increasingly higher for edges that are considered part of a micro-cluster anomaly.

Given the online nature of MIDAS, there is an inherent concern for processing time and memory use. The method solves that by making use of a data structure named *Count-Min Sketch* (CMS) [17], which makes use of *sublinear space* to store event frequency approximation while maintaining fast retrieval times.

A. Reproducing SOTA methodology

The state-of-the-art methodology we employed – MIDAS – is run over a series of detection experiments on annotated datasets [5]. In order to facilitate the reproduction of these experiments, we have used the publicly available implementation of the method[1]. The annotated datasets used in those experiments include:

1) DARPA: The 1999 DARPA intrusion detection dataset [18] is a well-known dataset for testing intrusion detection systems. It consists of, approximately, $4.5M$ packets being exchanged between $25K$ hosts over a period of nine weeks containing a multitude of attacks including, but not limited to, dictionary attacks, ftp-write and port scans.

Before being fed into the algorithm, the dataset was, first, formatted into a list of Source IP / Destination IP relations

[1] https://github.com/Stream-AD/MIDAS

accompanied by the packet timestamp. In this case, the IP addresses will be the nodes on the dynamic graph, while the edge will represent the occurrence of a communication between them. Each entry – timestamp, source IP, destination IP – is, then, fed into the algorithm individually and its anomaly score recorded. The final performance is given in terms of the *area under the receiver operating characteristic* curve (ROC-AUC). Given the randomized nature of the CMS data structure, the SOTA technique proposes that each experiment be repeated 21 times, and that the reported result be the median score of these. We follow their proposal in our experiments in order to correctly replicate the SOTA.

For the experiment with the DARPA dataset, the performance reported was **0.9873** median ROC-AUC with a standard deviation of **0.0009**. In our reproduction, we achieved a similar **0.9845** median ROC-AUC with a standard deviation of **0.002**.

Besides reproducing the experiment with the full DARPA dataset, we also performed the same experiment with its subdivisions (partitions). The dataset is originally separated into a *Training* and a *Testing* partitions. The training partition contains data from seven weeks of capture, while the testing partition includes two weeks of capture. Each week is also divided into five weekdays (Monday to Friday).

We ran the experiment for each partition of the dataset (all entries under *Training*, then all entries under *Testing*), for each week individually, and for each weekday. These experiments with the smaller subdivisions yielded results that ranged from **1%** to **99.9%** ROC-AUC. This shows that the SOTA methodology could take a considerable hit in performance if it is used with on specific scenarios represented by smaller datasets. This aspect is explored further in this research.

2) CTU-13: The CTU-13 dataset [19] is a dataset of botnet traffic captured by the CTU University in Czech Republic. It consists of 13 distinct scenarios of botnet traffic, representing different forms of malicious behaviour. Each of the provided scenarios can be used individually or be combined. Due to the focus on microcluster anomalies, this experiment was run over the combined set of entries of scenarios 4, 10, and 11 of the CTU-13, which are the scenarios containing DDoS attacks. This set contains $2.5M$ packets being exchanged between $371K$ hosts. Given the data is provided in the form of raw packets, we took a similar approach to the DARPA experiment (albeit not fully described in [5]) and formatted each packet into a tuple: timestamp, source IP, and destination IP.

In this experiment, the performance reported was **0.9843** median ROC-AUC, with a standard deviation of **0.0004**. In our reproduction, we achieved the same **0.9843** median ROC-AUC, although with a slightly higher standard deviation of **0.0005**. We suspect this might be due to the formatting of the tuples. For this dataset, we run the same experiment over the individual scenarios, as opposed to a combination of them. This is similar to the process of running the experiments over the different partitions of the DARPA dataset. In the case of the CTU-13 dataset, both scenarios 10 and 11 yield, respectively, a median ROC-AUC of **0.9937** and **0.9957**. In the meantime, scenario 4 yields a median ROC-AUC of **0.6170** and standard deviation of **0.0124**, which is considerably lower than the previous experiments.

This decrease in performance is not easily explained considering all three scenarios have packets from similar sources – all covering DDoS attacks – and only scenario 11 has a considerably smaller number of entries than the other two scenarios.

3) UNSW-NB15: The UNSW-NB15 dataset [20] was captured at the University of New South Wales and contains normal traffic data as well as synthetic modern attack behaviour. It contains approximately $2.5M$ records including, but not limited to, packets for DDoS attacks, backdoor attacks, and fuzzer attacks. This dataset is made available in four partitions.

As with the previous experiments, we formatted the dataset as tuples of timestamp, source IP address, and destination IP address. During this experiment we encountered, for the first time, a situation where some of the anomaly scores achieved by the SOTA were *inf*, that is, a value too high to be represented or simply *infinity*. In this case, the ROC-AUC could not be properly calculated as it does not support infinite values. As a workaround, we extracted the raw scores for each record and replaced all instances of *inf* with a value higher than every other (usually $1B$). This allowed us to calculate an approximate ROC-AUC.

The performance reported for these experiments was **0.8517** median ROC-AUC with a standard deviation of **0.0013**. In our reproduction, we achieved a similar **0.8710** median ROC-AUC with a standard deviation of **0.0002**. Like in the previous experiments, we also ran the experiment for each one of the partitions of the dataset individually. These yielded, respectively, **0.7654**, **0.9732**, **0.7556**, and **0.7244** median ROC-AUC. For the third and fourth partitions we also encountered the issue with the *inf* values. These experiments show how the distribution of normal/anomalous records considerably influences the final result of micro-clustering performed and achieved by the SOTA methodology.

IV. Proposed Model: Multiple Instances of Multi-Cluster

As seen in section III-A, the micro-clustering approach yields a good performance for the experiments conducted from a macro perspective on the datasets. However, once we have a deeper look at the reproduced results (of the SOTA) over a micro perspective, we notice that this performance can take a considerable hit on different scenarios (partitions) of the same datasets. With that in mind, our objective is to improve upon these results given that one of the aims of micro-clustering approaches is to be able to focus on rapid changes on specific situations that might be encountered on different systems.

To this end, we propose, Multiple Instances of Multi-Cluster (MIMC). In the aforementioned experiments, the micro-clustering approach only makes use of the source IP address and destination IP address of each packet, even though other meta-data is available. With MIMC, we propose to enhance this property without deviating from the micro-clustering approach, given that the meta-data is available in

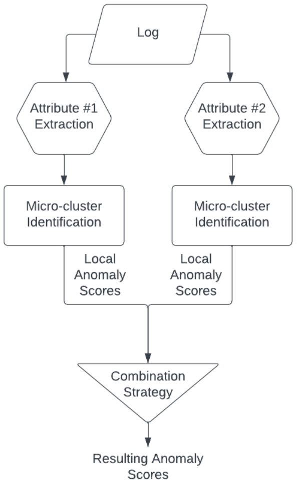

Fig. 1: Flowchart of the proposed approach MIMC

the log files. Moreover, we introduce the concept of parallel instances of micro-clustering to be able to focus on different aspects of the data. Lastly, we introduce different strategies to be able to combine the multiple instances of micro-cluster anomaly detection to calculate an overall anomaly score for the analyst.

Firstly we propose the use of alternative attributes (parameters) such as *source port* and *destination port* for the micro-clustering. Overall evaluations of the SOTA methodology using these attributes yields a ROC-AUC of **88.85%** for the DARPA dataset, **98.58%** for the CTU-13 dataset, and **78.19%** for the UNSW-NB15 dataset. For these and the experiments performed over the specific scenarios (partitions) of the datasets, the results show that using source port and destination port indicate similar or better performance (higher ROC-AUC) under certain conditions, while a decrease in the performance (ROC-AUC) under particular partitions.

Secondly, given that our goal is to improve the anomaly scores generated by each experiment with the objective of improving the overall performance, MIMC runs parallel instances of micro-clustering with different sets of attributes. Thirdly, they are combined using different strategies for calculating the anomaly scores. These are then used to calculate the ROC-AUC score as the SOTA method does over its set of scores. This process is illustrated in figure 1.

In this work, we analyze the impact of MIMC in the performance by combining the scores yielded using *Source IP Address* and *Destination IP Address*, as well as *Source Port* and *Destination Port*, using three distinct combination strategies over concurrent instances:

- **Max**: For the same entry, keep the highest of the scores.
- **Min**: For the same entry, keep the smallest of the scores.
- **Avg**: For the same entry, calculate the average of the scores.

V. Evaluations and Results

In order to test the performance of MIMC, we ran experiments over the aforementioned datasets used by the SOTA method as seen in section III. We then compared their performances for detecting anomalies over multiple scenarios (partitions) of the available data using the three aforementioned combination strategies – *max*, *min*, and *avg*. For each case compared, we classified it as an improvement if the yielded ROC-AUC was higher or equal to its counterpart reported in the literature, comparable, if the ROC-AUC was within 5 points percentual below the counterpart, or a decline, otherwise. In the following figures black coloured regions represent improvements, darker grey coloured regions represent comparable regions and light grey coloured regions represent decline performance. White coloured regions represent the parts of the data with no anomaly.

A. DARPA

For the DARPA dataset, we start by comparing the performance of both techniques over each of the five weekdays of each of the seven weeks present in the training partition of the dataset. Two of the days in this partition do not include anomalous entries, so their ROC-AUC is not calculated neither for SOTA nor for MIMC. When using the *max* combination strategy, as seen in figure 2, MIMC achieves **29** improvements, **4** comparable results, and **0** declines. For the *min* combination strategy, as seen in figure 3, MIMC achieves **20** improvements, **12** comparable results, and **1** decline. For the *avg* combination strategy, as seen in figure 4, MIMC achieves **29** improvements, **4** comparable results, and **0** declines.

Next, we compare the performance of both techniques over each of the five weekdays of each of the two weeks present in the testing partition of the dataset. For the *max* combination strategy, as seen in figure 5, MIMC achieves **9** improvements, **1** comparable results, and **0** declines. For the *min* combination strategy, as seen in figure 6, MIMC achieves **8** improvements, **2** comparable results, and **0** declines. For the *avg* combination strategy, as seen in figure 7, MIMC achieves **10** improvements, and **0** comparable results or declines.

Following that, we combine all the data available of each weekday to make partitions for each of the seven training weeks on the dataset. While this provides us direct comparion to SOTA methodology, it also enables us to study macro (all) versus micro (specific) focus on the data. For the *max* combination strategy, as seen in figure 8, MIMC achieves **7** improvements, and **0** comparable results or declines. For the *min* combination strategy, as seen in figure 9, MIMC achieves **2** improvements, **5** comparable results, and **0** declines. For the *avg* combination strategy, as seen in figure 10, MIMC achieves **7** improvements, and **0** comparable results or declines.

We, then, identify the partitions for each of the two testing weeks on the dataset. For the *max* combination strategy, as seen in figure 11, MIMC achieves **1** improvement, **1** comparable, and **0** decline result. For the *min* combination strategy, as seen in figure 12, MIMC achieves **1** improvement, **0** comparable, and **1** decline result. For the *avg* combination

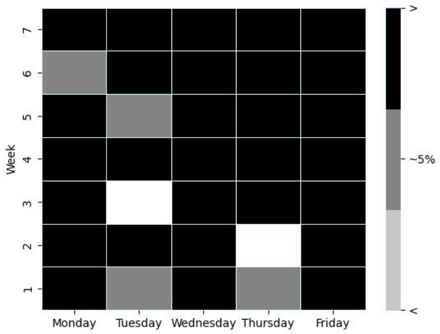

Fig. 2: Comparison between MIMC using the *max* combination strategy and SOTA over DARPA training weekdays

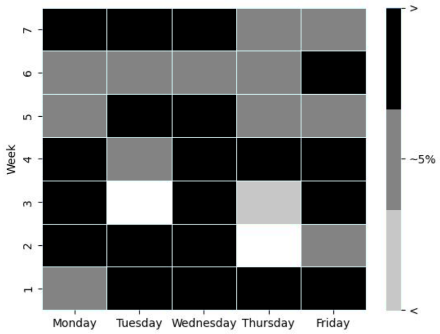

Fig. 3: Comparison between MIMC using the *min* combination strategy and SOTA over DARPA training weekdays

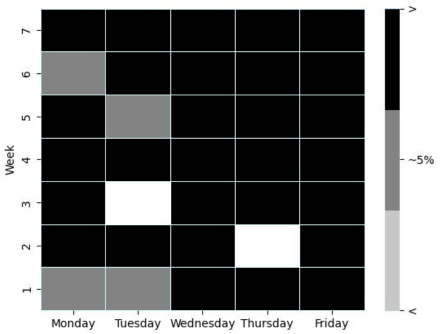

Fig. 4: Comparison between MIMC using the *avg* combination strategy and SOTA over DARPA training weekdays

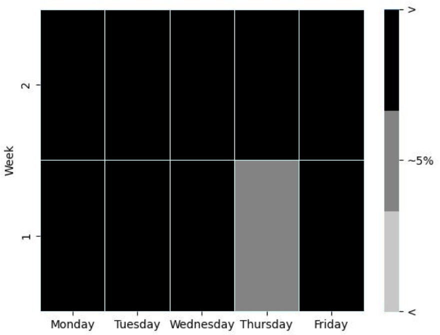

Fig. 5: Comparison between MIMC using the *max* combination strategy and SOTA over DARPA testing weekdays

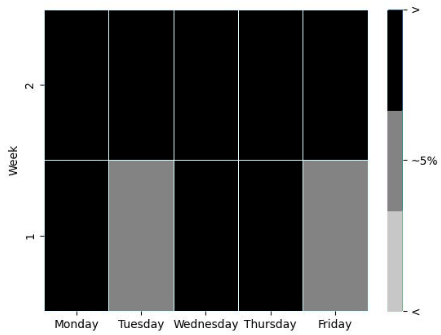

Fig. 6: Comparison between MIMC using the *min* combination strategy and SOTA over DARPA testing weekdays

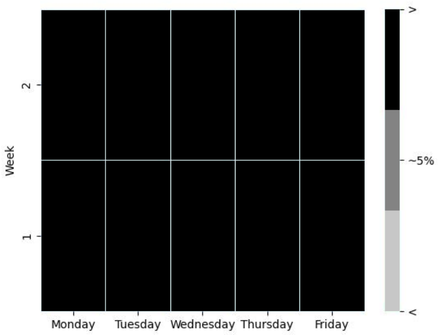

Fig. 7: Comparison between MIMC using the *avg* combination strategy and SOTA over DARPA testing weekdays

strategy, as seen in figure 13, MIMC achieves **1** improvement, **1** comparable, and **0** decline result.

When analyzing the entire training partition, SOTA method achieves **99.33%** ROC-AUC, whereas MIMC achieves – for the *max*, *min*, and *avg* strategies respectively – **99.28%**, **98.79%**, and **99.32%**. All are comparable results. For the entire testing partition, SOTA method achieves **93.45%**, whereas MIMC achieves **93.35%** and **89.64%** for the *max* and *min* strategies respectively – which are comparable results – and **93.53%** for the *avg* strategy, which is an improvement result. Finally, when comparing the performance over the entire dataset, SOTA method achieves **98.43%** ROC-AUC (as seen previously), whereas MIMC achieves **97.82%**, **97.33%**, and **97.95%** for the *max*, *min*, and *avg* strategies, respectively. These fall under comparable results.

B. CTU-13

As seen in section III-A, the performance of SOTA methodology over the individual scenarios of the CTU-13 dataset (as opposed to the combination of them) showed that the performance over scenario 4 was not as high as scenarios 10 and 11. To further analyze and improve the performance over these individual scenarios, we run MIMC over each one with the three aforementioned combination strategies. As seen in table I, MIMC is able to improve the performance of every scenario with at least one technique. Scenario 4, especially, gets an improvement of over 6 points percentual with the *min* strategy. All other results fall under comparable results.

	ROC-AUC			
Technique	**Scenario 4**	**Scenario 10**	**Scenario 11**	**All**
SOTA	62.06%	99.35%	99.62%	98.43%
MIMC (max)	62.45%	99.33%	**99.62%**	98.39%
MIMC (min)	**68.82%**	**99.51%**	99.30%	**98.74%**
MIMC (avg)	62.39%	**99.35%**	99.61%	98.42%

TABLE I: Comparison between performances of SOTA and MIMC over scenarios 4, 10, and 11, and their combination of CTU-13 dataset. Bold values represent improvements

C. UNSW-NB15

Our experimentation with the SOTA methodology showed that the performance of Partition 2 of the UNSW-NB15 dataset was considerably higher than the other three parts. To further analyze and improve the performance over these individual scenarios, we ran MIMC over each one with the three aforementioned strategies. As seen in table II, MIMC shows improvement in particular over Partition 4 as well as for the entire dataset using the *min* combination strategy. Besides that, it shows 4 comparable and 5 decline results.

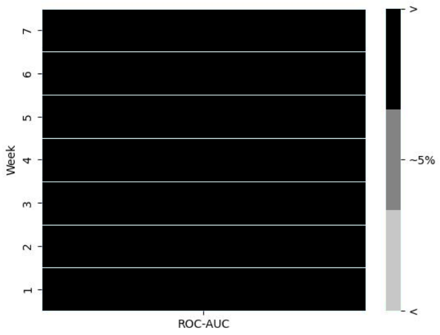

Fig. 8: Comparison between MIMC using the *max* combination strategy and SOTA over DARPA training weeks

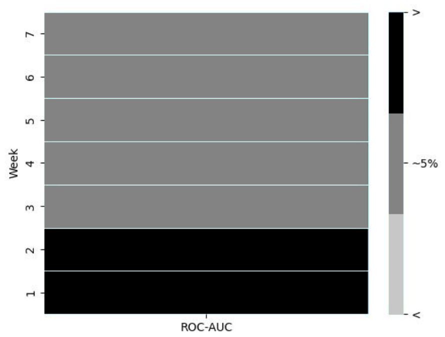

Fig. 9: Comparison between MIMC using the *min* combination strategy and SOTA over DARPA training weeks

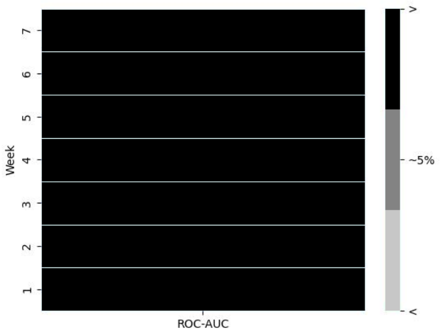

Fig. 10: Comparison between MIMC using the *avg* combination strategy and SOTA over DARPA training weeks

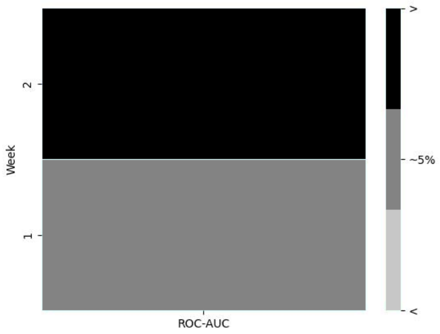

Fig. 11: Comparison between MIMC using the *max* combination strategy and SOTA over DARPA testing weeks

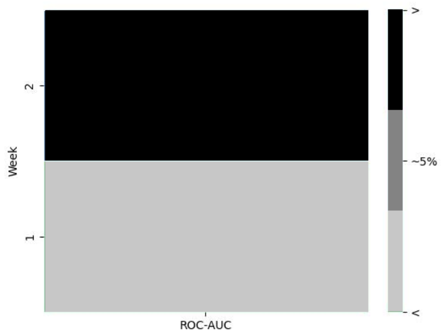

Fig. 12: Comparison between MIMC using the *min* combination strategy and SOTA over DARPA testing weeks

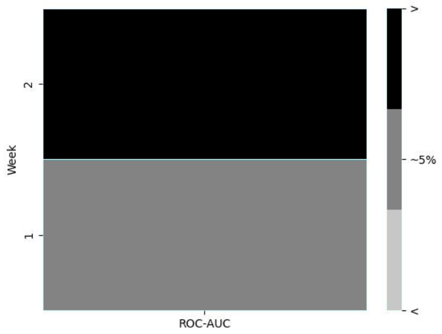

Fig. 13: Comparison between MIMC using the *avg* combination strategy and SOTA over DARPA testing weeks

			ROC-AUC		
Technique	**Part 1**	**Part 2**	**Part 3**	**Part 4**	**All**
SOTA	76.99%	97.33%	75.56%	72.44%	87.66%
MIMC (max)	54.57%	**97.33%**	64.83%	**72.51%**	83.77%
MIMC (min)	75.29%	94.40%	68.52%	**75.46%**	**89.74%**
MIMC (avg)	63.03%	**97.33%**	67.00%	**74.23%**	85.94%

TABLE II: Comparison between performances of SOTA and MIMC over partitions 1 to 4, and their combination of UNSW-NB15 dataset. Bold values represent improvements

VI. Conclusion & Future Work

In this work, we analyzed a SOTA graph-based methodology for detecting multi-cluster anomalies in log data using publicly available MIDAS. Based on the performance of SOTA, we proposed a novel method, MIMC, that makes use of (i) enhancing attributes, (ii) executing multiple instances of micro-clustering over different attributes of the log in parallel, and (iii) combining these multiple instances of micro-cluster anomaly detection with an appropriate strategy. In order to evaluate the performance of MIMC, we ran experiments over the same datasets used in SOTA and compared the results. Under many scenarios, we observed an improvement by MIMC in performance over the results yielded by SOTA when experimenting with specific scenarios (partitions) of the datasets. This is achieved by MIMC without getting a hit on the performance when evaluating with the entire datasets. It should be noted here that SOTA evaluations were done over the entire datasets. In summary, this demonstrates that the results obtained by the proposed method MIMC have merit in the area of graph-based anomaly detection over different types of log data. The impact seems to focus on scenarios and partitions where the data is sparse and therefore makes it more challenging to differentiate normal vs anomaly for SOTA. This indicates that the steps proposed in MIMC improve the reliability of detection on scenarios where data is in short supply. We believe that this is an important achievement on the road to building an unsupervised and online approach. The exploration of using other attributes and using more instances of multi-cluster identification are the next steps for future research directions.

Acknowledgement

This research is supported by an NSERC Alliance Grant. The first author gratefully acknowledges the support by the province of Nova Scotia. The research is conducted as part of the Dalhousie NIMS Lab[2].

[2] https://projects.cs.dal.ca/projectx/

References

[1] Yuchong Li and Qinghui Liu. A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. *Energy Reports*, 7:8176–8186, 2021.

[2] Monowar H Bhuyan, Dhruba Kumar Bhattacharyya, and Jugal K Kalita. Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials*, 16(1):303–336, 2013.

[3] Caleb C Noble and Diane J Cook. Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636, 2003.

[4] Debajit Sensarma and Samar Sen Sarma. A survey on different graph based anomaly detection techniques. *Indian J Sci Technol*, 8(31):1–7, 2015.

[5] Siddharth Bhatia, Rui Liu, Bryan Hooi, Minji Yoon, Kijung Shin, and Christos Faloutsos. Real-time anomaly detection in edge streams. *ACM Transactions on Knowledge Discovery from Data*, 16(4):1–22, jan 2022.

[6] Kabul Kurniawan, Andreas Ekelhart, Elmar Kiesling, Dietmar Winkler, Gerald Quirchmayr, and A Min Tjoa. Vlograph: a virtual knowledge graph framework for distributed security log analysis. *Machine Learning and Knowledge Extraction*, 4(2), 2022.

[7] Mandana Saebi, Jian Xu, Lance M Kaplan, Bruno Ribeiro, and Nitesh V Chawla. Efficient modeling of higher-order dependencies in networks: from algorithm to application for anomaly detection. *EPJ Data Science*, 9(1):15, 2020.

[8] Adarsh Kulkarni, Priya Mani, and Carlotta Domeniconi. Network-based anomaly detection for insider trading. *arXiv preprint arXiv:1702.05809*, 2017.

[9] Misael Mongiovi, Petko Bogdanov, Razvan Ranca, Evangelos E Papalexakis, Christos Faloutsos, and Ambuj K Singh. Netspot: Spotting significant anomalous regions on dynamic networks. In *Proceedings of the 2013 Siam international conference on data mining*, pages 28–36. SIAM, 2013.

[10] Shilin He, Jieming Zhu, Pinjia He, and Michael R Lyu. Experience report: System log analysis for anomaly detection. In *2016 IEEE 27th international symposium on software reliability engineering (ISSRE)*, pages 207–218. IEEE, 2016.

[11] Takeaki Uno, Hiroki Maegawa, Takanobu Nakahara, Yukinobu Hamuro, Ryo Yoshinaka, and Makoto Tatsuta. Micro-clustering: finding small clusters in large diversity. *arXiv preprint arXiv:1507.03067*, 2015.

[12] Amir Farzad and T Aaron Gulliver. Unsupervised log message anomaly detection. *ICT Express*, 6(3):229–237, 2020.

[13] Xu Zhang, Yong Xu, Qingwei Lin, Bo Qiao, Hongyu Zhang, Yingnong Dang, Chunyu Xie, Xinsheng Yang, Qian Cheng, Ze Li, et al. Robust log-based anomaly detection on unstable log data. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 807–817, 2019.

[14] Haixuan Guo, Shuhan Yuan, and Xintao Wu. Logbert: Log anomaly detection via bert. In *2021 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2021.

[15] Xiaolan Wang, Md Manjur Ahmed, Mohd Nizam Husen, Hai Tao, and Qian Zhao. Dynamic micro-cluster-based streaming data clustering method for anomaly detection. In *Soft Computing in Data Science: 7th International Conference, SCDS 2023, Virtual Event, January 24–25, 2023, Proceedings*, pages 61–75. Springer, 2023.

[16] Ling Lin and Jinshan Su. Anomaly detection method for sensor network data streams based on sliding window sampling and optimized clustering. *Safety science*, 118:70–75, 2019.

[17] Graham Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[18] RP Lippman, RK Cunningham, DJ Fried, I Graf, KR Kendall, SE Webster, and MA Zissman. Results of the darpa 1998 offline intrusion detection evaluation. In *Slides presented at RAID 1999 Conference*, 1999.

[19] Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino. An empirical comparison of botnet detection methods. *computers & security*, 45:100–123, 2014.

[20] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE, 2015.