Improving Real-Time Anomaly Detection using Multiple Instances of Micro-Cluster Detection

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

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

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

***Abstract*—Analysis of incoming packets in deployed systems is one of the main methods used for detection of anomalous behaviour. Techniques utilizing supervised learning subject to the need of retraining if the observed behaviour in the system changes over time. Unsupervised techniques mitigate this problem but are not always capable of real-time analysis. Real-time unsupervised techniques bring to the table both the adaptability to dynamic behaviour as well as the ability to detect and alert about anomalies in real-time. A recent state-of-the-art technique, MIDAS, shows real-time capabilities while being unsupervised, but recent works have showed that it still had some shortcomings regarding its performance over more specific datasets. An alternative method has been proposed, namely MIMC, that builds on the foundation set by MIDAS. In this paper it is shown that, for the datasets of interest, there is always a way to setup MIMC that yields a higher performance than MIDAS. Furthermore, a method for determining parameters for the technique is also presented, and it is shown that it improves the yielded performance even further in a majority of cases.**

***Index Terms*—Network and service security, micro-clustering, anomaly detection, log analysis.**

I. Introduction

Analysis of incoming packets in deployed systems is one of the main methods used for detection of anomalous behaviour. This kind of strategy is especially useful for detecting when systems are misconfigured or when malicious attacks are under way.

Techniques utilizing supervised learning require not only labelled data for training but also additional effort to train and deploy in existing systems and are subject to the need of retraining if the observed behaviour in the system changes over time. Unsupervised techniques mitigate this problem by adapting to changing environments without the need for retraining, but are not always capable of real-time analysis. Real-time unsupervised techniques bring to the table both the adaptability to dynamic behaviour as well as the ability to detect and alert about anomalies in real-time.

A recent state-of-the-art technique, MIDAS [1], shows real-time capabilities while being unsupervised. Its performance is very respectable with detection rates around ninety percent over publicly available attack datasets. However, Copstein et. al. [2], [3] showed that MIDAS still had some shortcomings regarding its performance over more specific datasets. Moreover, MIDAS does not consider some of the available information that directly relates to the kinds of attacks present in the datasets of interest. In these works, an alternative method is proposed, namely MIMC, that builds on the foundation set by MIDAS.

The goal of this paper is to provide further evidence of MIMC's higher-performing capabilities when compared to MIDAS over datasets of interest. To do that, first we establish MIDAS as a state-of-the-art real-time unsupervised anomaly-detection technique by comparing it to other techniques in the literature. Then, we show how MIMC builds on top of the foundation set by MIDAS and how it improves on some of its shortcomings. Furthermore, we present a method for determining MIMC's parameters that further improves its performance, and finally we show that MIMC's results are higher than MIDAS's in a statistically significant way. Compared to previous publications covering MIMC, this paper better describes the method's foundations and better supports the claims of higher performance through new experiments and analyses.

The rest of this paper is organized as follows: section II explores related works in the literature, section III summarizes some key concepts required for understanding this study, section IV presents the main differences between MIMC and MIDAS and describes the experiments run and results obtained, and section V summarizes our conclusions and defines topics for future work.

II. Related Works

This research is closely related to log analysis and anomaly detection on log data, particularly graph-based approaches for anomaly detection. The following section gives an overview of related works in the area.

Noble and Cook [14] proposed two methods for graph-based anomaly detection. Subdue uses a method for detecting recurring substructures in graphs. When tested on the 1999 KDD Cup data, the methods show reasonable results in identifying some attacks, albeit inferior performance for others.

The solution presented by Kurniawan et al. [10], namely VloGraph, uses 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, to retrieve events of interest.

978-3-903176-66-9

Kulkarni et al. [9] explored the patterns found by creating different graphs of insider trading data. These include networks of traders, purchases, and sales of stocks. Furthermore, they explore the idea of anomaly detection using hyper-graphs, concluding that, given the complexity of the domain, it is hard to evaluate the performance of their model. However, they can confirm that the hyper-edges identified as anomalies (insider trading) result in profit for the trader in most cases.

Moreover, Mongiovi et al. proposed NetSpot [13] for finding anomalous regions on dynamic networks, including traffic networks, social networks, and knowledge networks. They show that NetSpot is up to one order of magnitude faster than an exhaustive search approach and yields results within 5%.

He et al. [8] analyzed six methods for anomaly detection using log data: three supervised and three unsupervised. The three supervised methods are based on Logistic Regression, Decision Trees, and Support Vector Machines (SVM). The unsupervised methods include Clustering, Principal Component Analysis, and Invariant Mining. Regarding 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.

The work of Uno et al. [16] introduced the problem of micro-clustering, defined as unsupervised soft clustering. Here, the problem is clustering highly related entries instead of 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 [5] proposed 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 the Gaussian Mixture Model and One-Class Support Vector Machine.

In [17], Zhang et al. introduced 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. [7], uses BERT to run self-supervised training to learn sequences of log masks, that is, masks yielded by a log abstraction process. Sequences of masks that do not match the trained ones are deemed anomalous.

STAD, a framework introduced by Makanju et al. [12], attempts to detect alerts using system log information. It uses a clustering technique based on spatiotemporal information, in this case, nodehours. The authors report a detection of 100% of all alerts with a false positive ratio of 0.8% in the best case, a detection rate of 78% and a false positive rate of 5.4% on average. During its initial phase, STAD requires extracting message types, which, in turn, requires the analysis of multiple log messages. This aspect makes STAD unsuitable for real-time processing.

Bhati et al. [1] introduced MIDAS. It is an unsupervised method that provides anomaly detection and is optimized enough for real-time processing. Compared to related works, MIDAS performs better in all selected datasets, reaching accuracy above 98% in two of the three tested cases. Using a data structure from the family of sketches also helps reduce the method's memory usage to sublinear.

An overview of the characteristics of the works presented here can be seen in Table I. In summary, none of the presented methods have all the desired characteristics, and MIDAS is the only one planned for real-time performance. Methods relying on machine learning models tend to be harder to interpret when performing root-cause analysis over an incident. Graph-based methods are more accessible to analyze and, in most cases, provide a reasonable level of explainability to flagged anomalies. When using language models, a method relies on the semantic value of log entries, which may not always be intended for readability. Given these methods, we take MIDAS as the current state-of-the-art method that provides the desired characteristics of a real-time anomaly detector.

III. Background

A. MIDAS

As the state-of-the-art (SOTA) method, MIDAS is an efficient, unsupervised, real-time method for anomaly detection. It comprises three main stages: attribute extraction, frequency storage, and probabilistic test, as seen in Figure 1.

In the attribute extraction stage, MIDAS receives entries in the form of logs and extracts an attribute co-occurrence of the source IP address and destination IP address. This extraction assumes that the format of the logs fed into the system is previously known, which is typical for network logs.

In the following stage, addresses are registered into a graph where each node is an IP address, and the edges register the number of times two addresses have co-occurred. This is done by using a *Count-Min Sketch* [4]. The method uses a Chi-Squared Test to determine the likelihood that the information gathered in the graph follows a Chi-Squared distribution. This is used to yield an unbounded anomaly score, that is, a score with no fixed maximum.

Throughout its execution, MIDAS aims to phase out old and prioritize newly gathered information. To do that, every minute (as given by the timestamp on the log entries fed to the system), the values stored in the graph get multiplied by a *learning ratio*, a value between 0 and 1, that represents how much of the current data should be kept for the following analyses. MIDAS will only insert new data into the frequency graph if the yielded anomaly score does not exceed a set *threshold* to avoid poisoning the graph with anomalous data.

Given the use of attribute co-occurrence frequency as the main factor for determining anomalies, we can say that MIDAS focuses on anomalies that follow the property of being *bursty*, as proposed by Makanju et al [12]. The performance of MIDAS is determined by calculating the ROC-AUC over the anomaly scores yielded for each log entry.

TABLE I
COMPARING THE RELEVANT CHARACTERISTICS OF THE METHODS INTRODUCED IN RELATED WORKS

Method	Unsupervised	Graph-based	Explainability	Real-Time
Noble & Cook	NO	YES	YES	NO
VloGraph	YES	YES	YES	NO
Kulkarni et al.	N/A	YES	YES	NO
NetSpot	N/A	YES	YES	NO
He et al.	YES	NO	PARTIAL	NO
Farzad and Gulliver	YES	NO	NO	NO
LogRobust	NO	NO	NO	NO
LogBERT	NO	NO	NO	NO
STAD	YES	NO	PARTIAL	NO
MIDAS	YES	YES	PARTIAL	YES

B. Datasets of Interest

For this research, we refer to three datasets of interest:

- **CIC IDS 2017** [15]: contains benign data and data from attacks, such as brute-force FTP, brute-force SSH, DoS, Heartbleed, infiltration, botnet, and DDoS. The Canadian Institute of Cybersecurity at the University of New Brunswick makes this dataset available and includes approximately 2.8M entries. It consists of 5 days of captures.
- **DARPA** [11]: is well-known for testing intrusion detection systems. It consists of approximately 4.5M packets exchanged between 25K hosts over nine weeks. The attacks included in this dataset are arranged in 5 categories: Denial of Service, User to Root, Remote to Local, Probes, and Data.
- **CTU-13** [6]: dataset of botnet traffic is captured by the CTU University in the Czech Republic. It consists of thirteen distinct scenarios of botnet traffic, representing different forms of malicious behaviour. Each of the provided scenarios can be used individually or combined. This set contains 2.5M packets being exchanged between 371K hosts. For this research, similarly to the evaluation for the MIDAS technique, only scenarios 4, 10, and 11, which included some form of DDoS attack, were considered.

IV. METHODOLOGY

While MIDAS can be seen as the state-of-the-art technique for real-time anomaly detection, MIMC improves on it in three main aspects:

1) There are multiple instances of each stage. A set of *attribute co-occurrence extraction*, *frequency storage* and *probabilistic test* is called a **lane**.
2) Each lane is responsible for extracting, storing, and testing a single attribute co-occurrence and yields a **local anomaly score** regarding only that co-occurrence.
3) Local anomaly scores are combined into a single resulting anomaly score using a **combination strategy**.

Each lane of an instance of MIMC yields an unbounded anomaly score, similar to MIDAS, for each entry. Let n be the number of lanes in an instance of MIMC, L_a where $1 \leq a \leq n$ be a lane in the instance, i be a valid arbitrary input, and $S_(a, i) \in R_(\geq 0)$ the score yielded by lane L_a for input i; a combination strategy is defined as a function:

$$C : {R_{\geq 0}}^n \rightarrow R_{\geq 0}$$

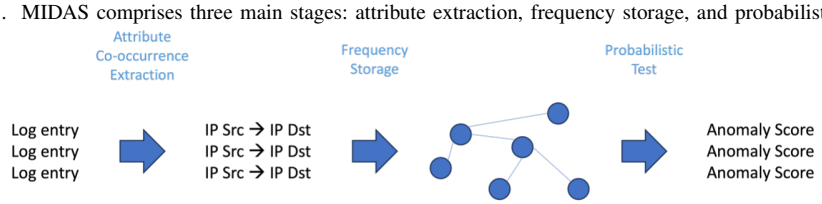
The resulting anomaly score is given by

$$C(S_{(1,i)}, S_{(2,i)}, ..., S_{(n,i)})$$

Along with the technique, fours methods are proposed to be used as combination strategies in instances of MIMC:

- **Max**: Yields the highest value of the local anomaly scores given by the lanes. This strategy considers that if any attribute co-occurrence indicates a high anomaly score, it is enough for the technique to treat an input as a potential threat.
- **Min**: Yields the lowest value of the local anomaly scores given by the lanes. This strategy says that input is only considered as much of a threat as indicated by the lowest local anomaly score, which indicates that all other scores are at least as high.
- **Median**: Yields the median value of the local anomaly scores given by the lanes. This strategy attempts to consider all of the yielded scores without prioritizing very high or low outliers.

Fig. 1. MIDAS comprises three main stages: attribute extraction, frequency storage, and probabilistic test



- **Average**: This strategy yields the average value of the local anomaly scores given by the lanes. It attempts to consider all of the yielded scores and is more susceptible to outliers.

Each combination strategy is able to provide explainability in the form of the attribute co-occurrence that influenced the final result the most. For example, the **Max** combination strategy is able to provide the attribute co-occurrence that yielded the highest anomaly score and, therefore, is the one that influenced the final score the most. The only exception to this is the **Average** combination strategy, which takes all attribute co-occurrences into consideration equally.

Five parameters are used to configure instances of MIMC: two of them exclusive to MIMC — the number of lanes (and their respective attribute co-occurrences) and the combination strategy (also known as the merger) — which are paired into the called **setup** — and the three parameters in MIDAS — threshold, number of columns, and learning ratio — which are combined into the called **configuration**.

A. Setups

To show that MIMC outperforms MIDAS, we must show that the main idea behind it - the combination of multiple lanes of micro-cluster detection - outperforms MIDAS's single lane in detecting anomalies. In order to do that we propose five new attribute co-occurrences to be used in experimentation:

- Source Port → Destination Port ($PortSrc \rightarrow PortDst$)
- Destination IP Address → Destination Port ($IPDst \rightarrow PortDst$)
- Destination IP Address → Bytes Sent ($IPDst \rightarrow SrcBytes$)
- Source IP Address → Destination Port ($IPSrc \rightarrow PortDst$)
- Source IP Address → Bytes Sent ($IPSrc \rightarrow SrcBytes$)

Experiemnts were run with MIMC over the datasets of interest for every possible combination of the five proposed attribute co-occurrences and the co-occurrence used originally by MIDAS ($IPSrc \rightarrow IPDst$). Each attribute co-occurrence is assigned to a lane on an instance of MIMC, and the yielded results are combined using each of the proposed combination strategies. The six attribute co-occurrences can be combined in sixty-four different ways, and there are four proposed combination strategies, which means there are 256 possible setups to use when evaluating all datasets.

The results of these experiments, as seen in Table II, were summarized to report only the best result for each setup, that is, only the best-performing combination of attribute co-occurrences and merging strategy is reported for each dataset. Along with the ROC-AUC score, the results also report the combination and strategy that was used, the equivalent performance yielded by MIDAS over the same dataset, and whether MIMC's yielded score outperformed MIDAS (>M), as well as whether it outperforms an instance of MIMC with a single lane with any of the proposed co-occurrences (>S).

B. Configurations

Even though the previous experiments show that there is always a setup where MIMC outperforms MIDAS, they do not consider the possibility of changing MIMC's configuration, that is, the three parameters used by each lane: threshold, number of columns, and learning ratio. The previous experiments used the same configuration proposed by MIDAS in their implementation.

Given that the three parameters in a configuration are non-categorical, there is no explicit limit to how many configurations can be used by an instance of MIMC. On top of that, experimenting with a multitude of configurations over the datasets of interest quickly adds up to a large number of hours of experimentation. With that in mind, experiments with configurations were run considering three values for each of the configuration's parameters, all based on the configuration proposed by MIDAS: 1K, 10K, and 100K for the threshold (10K was proposed); 512, 1024, and 2048 for the number of columns (1024 was proposed); and 0.25, 0.5, and 0.75 for the learning ratio (0.5 was proposed).

Despite using a limited number of values for each parameter in the configuration, there are still 27 different configurations that need to be evaluated for each dataset, which may contain millions of entries. To make this evaluation more feasible and less demanding in terms of resources, determining *promising parameters* is proposed.

1) Most Promising Parameters: The most promising value for a configuration parameter is the one that yields the highest performance in identifying anomalies compared to different values of the same parameter. The most promising configuration contains the most promising values for all parameters.

To improve the performance previously yielded by MIMC, it is proposed that the most promising configuration be determined over a sample of each dataset of interest. Once determined, an experiment using this newly found configuration is executed over the entire dataset of interest and compared to the previously yielded performance. In all cases, the setup used for each instance of MIMC is the same as determined to be the best performing over the entire dataset in the previous experiments.

2) Selecting a Subset of Data: With the desire for real-time operation in mind, it is crucial to establish the importance of keeping the temporal aspect of the data to be analyzed. In other words, one cannot expect consistent results from either MIMC or MIDAS by running tests with randomly sampled datasets. That being said, selecting a sequence of entries from the datasets while preserving their original ordering is sufficient to fulfill this prerequisite.

Similarly to a supervised learning step of training, the ratio of benign over malicious entries in the selected subset directly impacts the performance of each technique in detecting anomalies. Therefore, a dataset subset was chosen considering the ratio of benign over malicious entries to be higher than one, having more benign entries than malicious ones.

A subset of 100,000 entries, with at least 50% benign, was extracted from the datasets of interest for the following

TABLE II
SUMMARY OF THE BEST-PERFORMING RESULT OF THE ANALYSIS OF THE DATASETS USING ALL COMBINATIONS OF ATTRIBUTE CO-OCCURRENCES AND COMBINATION STRATEGIES

DATASET	COMBINATION	STRATEGY	SCORE	MIDAS	>M	>S
CIC IDS 2017	IPSrc-IPDst IPSrc-PortDst	MIN	0.9595	0.9532	Y	Y
DARPA	IPSrc-IPDst IPSrc-PortDst PortSrc-PortDst	MED	0.9899	0.9816	Y	Y
CTU-13	IPDst-PortDst IPDst-SrcBytes IPSrc-IPDst IPSrc-PortDst IPSrc-SrcBytes PortSrc-PortDst	MIN	0.9919	0.9831	Y	Y

TABLE III
YIELDED ROC-AUC FOR EACH PARAMETER FOR EXPERIMENTS WITH THE CIC IDS 2017 DATASET. THE HIGHLIGHTED VALUE SHOWS THE HIGHEST SCORE ACHIEVED FOR EACH PARAMETER.

Parameter	Values	ROC-AUC
Learning Ratio	0.25	61.31%
	0.50	61.59%
	0.75	66.34%
Num. Columns	512	61.80%
	1024	61.69%
	2048	99.26%
Threshold	1,000	94.19%
	10,000	61.66%
	100,000	62.25%

TABLE IV
YIELDED ROC-AUC FOR EACH PARAMETER FOR EXPERIMENTS WITH THE DARPA DATASET. THE HIGHLIGHTED VALUE SHOWS THE HIGHEST SCORE ACHIEVED FOR EACH PARAMETER.

Parameter	Values	ROC-AUC
Learning Ratio	0.25	99.74%
	0.50	99.73%
	0.75	99.82%
Num. Columns	512	99.59%
	1024	99.68%
	2048	99.77%
Threshold	1,000	99.77%
	10,000	99.68%
	100,000	99.72%

TABLE V
YIELDED ROC-AUC FOR EACH PARAMETER FOR EXPERIMENT WITH THE CTU-13 DATASET. THE HIGHLIGHTED VALUE SHOWS THE HIGHEST SCORE ACHIEVED FOR EACH PARAMETER.

Parameter	Values	ROC-AUC
Learning Ratio	0.25	45.63%
	0.50	51.85%
	0.75	23.65%
Num. Columns	512	42.17%
	1024	50.43%
	2048	53.47%
Threshold	1,000	41.85%
	10,000	54.02%
	100,000	43.33%

experiments.

3) Most Promising Configuration: For the CIC IDS 2017 dataset, as seen in Table III, the most promising configuration comprises the values that yielded the highest performance for each parameter. In this case, a configuration with a threshold of 1K, 2048 columns, and a learning ratio of 0.75 is deemed the most promising.

As Table IV shows, the most promising configuration for the DARPA dataset was 1K for the threshold, 2048 columns, and a 0.75 learning ratio.

Table V shows that the most promising configuration for the CTU-13 dataset was a 10K threshold, 2048 columns, and a 0.5 learning ratio.

4) Comparing to MIDAS: With the most promising configuration determined, it is now possible to compare MIMC's performance using the most promising setup and most promising configuration to that of MIDAS.

MIMC and MIDAS use one or more instances of the Count-Min Sketch data structure, which is inherently subject to slight alterations during multiple executions due to its randomized factor that is determined during instantiation. With that in mind, it is easy to see how results could be skewed towards exceptionally high or low results that do not represent the technique's expected performance. Each of the following comparisons was run ten times to mitigate this skewing. The reported output is the mean of the values yielded in these runs.

The results from these experiments were analyzed using the Mann-Whitney U test to strengthen the validity of the claims. This non-parametric test evaluates whether there is a significant difference in the distributions of two independent data sets. A significance level of 95% was applied, meaning that a p-value of 0.05 or lower would provide sufficient evidence to reject the null hypothesis and conclude that there is a significant difference between the distributions of the two groups. The test can be adapted to be two-sided, where the total difference is considered, or one-sided, where it is considered if the values in one set are higher or lower than those in the other.

Table VI summarizes the results of these experiments. For the CIC IDS 2017 dataset, using the most promising configuration improves the median result over MIDAS by 5.47 points percentual. With a p-value of approximately 0.0000829, the results yielded by MIMC are shown to be statistically greater (one-sided test) than those of MIDAS. For the DARPA dataset

TABLE VI
MEDIAN ROC-AUC FOR EXPERIMENTS WITH THE DATASETS OF INTEREST USING THE MOST PROMISING CONFIGURATION COMPARED TO MIDAS.

Dataset	Technique	ROC-AUC
CIC IDS 2017	MIMC	99.01%
	MIDAS	93.54%
DARPA	MIMC	98.39%
	MIDAS	98.02%
CTU-13	MIMC	98.52%
	MIDAS	97.44%

the result for MIMC's median result improves over MIDAS by 0.37 points percentual. With a p-value of approximately 0.000898, the results yielded by MIMC are shown to be statistically greater (one-sided test) than those of MIDAS. The results for the CTU-13 dataset show that MIMC's median result improves over MIDAS by 1.08 points percentual. With a p-value of approximately 0.0000843, the results yielded by MIMC are shown to be statistically greater (one-sided test) than those of MIDAS.

V. CONCLUSION & FUTURE WORK

Anomaly detection is an essential part of the security strategy of any deployed system. Methods that provide high levels of performance when identifying anomalies facilitate the operation of large-scale systems and reduce the workload of security specialists. This research aimed to provide further evidence of the high-performance capabilities of the previously introduced real-time anomaly detection technique MIMC and its improvements over its state-of-the-art counterpart MIDAS. These differences provide the technique with the capability of exploring data more deeply and, in turn, performing better at detecting anomalies.

Experiments presented in this work show that the core concept behind MIMC - the combination of local results into a global anomaly score - already produces better results than the state-of-the-art counterpart. On top of that, by making use of the proposed method of determining the *most promising configuration*, MIMC is able to further improve its performance by analyzing only a sample of the available data *a priori*.

Finally, by performing executions of both techniques multiple times, this work provides evidence of the performance improvements of MIMC despite the non-deterministic nature of both methods. Statistical significance is also shown with the use of the Mann-Whitney U test over the yielded results.

Methods for determining which attribute co-occurrences to use for each dataset of interest, determining which attacks are more susceptible to detection by MIMC, and providing guidelines for selecting parameters for novel datasets are left as future work.

ACKNOWLEDGEMENT

This research is supported partially by the Natural Sciences and Engineering Research Council of Canada. The first author also gratefully acknowledges the support by the province of Nova Scotia. The research is conducted as part of the Dalhousie NIMS Lab[1].

REFERENCES

[1] Siddharth Bhatia, Bryan Hooi, Minji Yoon, Kijung Shin, and Christos Faloutsos. Midas: Microcluster-based detector of anomalies in edge streams. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3242–3249, 2020.

[2] Rafael Copstein, Bradley Niblett, Andrew Johnston, Jeff Schwartzentruber, Malcolm Heywood, and Nur Zincir-Heywood. Mimc: Anomaly detection in network data via multiple instances of micro-cluster detection. In *2023 19th International Conference on Network and Service Management (CNSM)*, pages 1–7, 2023.

[3] Rafael Copstein, Bradley Niblett, Andrew Johnston, Jeff Schwartzentruber, Malcolm Heywood, and Nur Zincir-Heywood. Towards anomaly detection using multiple instances of micro-cluster detection. In *2023 7th Cyber Security in Networking Conference (CSNet)*, pages 185–191, 2023.

[4] 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.

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

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

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] Adetokunbo Makanju, A Nur Zincir-Heywood, Evangelos E Milios, and Markus Latzel. Spatio-temporal decomposition, clustering and identification for alert detection in system logs. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 621–628, 2012.

[13] 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.

[14] 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.

[15] Iman Sharafaldin, Arash Habibi Lashkari, Ali A Ghorbani, et al. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116, 2018.

[16] 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.

[17] 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.

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