Automating the Cybersecurity Triage Process: A Comparative Study on the Performance of Large Language Models

Pascal Bakker
p.n.bakker@student.utwente.nl
University of Twente

ABSTRACT

In Security Operation Centers, security analysts have the task to inspect cybersecurity alarms to filter false positives and identify their severity: triage. The problem with this process is that it is complicated and time-consuming, limiting the depth and speed of investigations. Whereas other proposed optimizations and automations appear to be very promising, rapid advancements in the development of Large Language Models (LLMs) opened up new possibilities to speed up the triage process. This research will identify ways in which LLMs can optimize triage, evaluate the performance of these techniques and offer a comparison between different LLMs. The findings in this study are expected to help security teams in making informed implementation decisions when optimizing the triage process.

KEYWORDS

Cybersecurity, triage, analysis, large language model, natural language processing, automation

1 INTRODUCTION

(TODO: numbers on security incidents)

The 2023 Cost of a Data Breach Report by IBM Security [12] concludes that the average cost of a security breach in 2023 was 4.45M USD, marking an increase of 2.3% since 2022 and a 15.3% increase compared to 2020. Considering that only 1 in 3 breaches are identified by an organization's security team, and that organizations with high levels of incident response planning saved 1.49M USD on average, it shows the pressing need for security investments in training, thread detection and response technologies.

One such investment is the use of Security Operation Centers (SOCs). Organizations use SOCs to respond to security incidents in real-time. SOCs consist of security analysts that investigate data from various sources such as Security Information and Event Management (SIEM) systems. The SIEM systems collect log data from a large number of sources such as network devices and applications within the organization's system. Based on rules, patterns and conditions, anomalies and suspicious activities are identified and alarms are created.

The number of alarms is large, ranging from hundreds to thousands per day, of which a large portion are false positives or low priority. The volume and complexity of the alarms causes SOCs to miss serious attacks and inadvertently contributes to mistakes in the analysis. Besides that, it leads to security teams experiencing fatigue, and it contributes to internal friction and turnover [23].

Since the SOCs cannot respond to every single alarm, identifying the severity of alarms is an important step in the incident response workflow. This process, triage, involves understanding the impact of an alarm, correlating it with other alarms and identifying potential future goals of adversaries to conclude its severity. By prioritizing alarms, SOCs can focus their resources on high-severity alarms first, thus mitigating damages and reducing costs.

There are many proposed and implemented techniques to optimize triage and the SOC workflow. For example, Security Orchestration, Automation, and Response (SOAR) platforms have streamlined parts of the process by automating routine tasks, but many steps of triage still require human judgment to make adequate decisions [7]. Consequentially, the triage process is prone to human error. This, in combination with the volume and complexity of alarms, presses the need for the automation of triage. (TODO: Other examples)

The field of Artificial Intelligence (AI) has the potential to significantly impact the automation of triage. It involves the use of machines to perform tasks that mimic human actions such as reasoning, problem-solving and learning [24]. Machine Learning (ML) allows systems to solve problems by analyzing patterns in data and making predictions and decisions without explicit programming [25]. One subfield of ML is Natural Language Processing (NLP). NLP involves using computational approaches to process and transcribe natural-language texts with further goals such as translation, summarization, sentiment assessment or generation of texts. It plays a growing role in streamlining and automating business operations, and increasing productivity [11].

One application of NLP is that of Large Language Models (LLMs). These models have been trained on immense amounts of natural-language data and are capable of understanding and generating texts to perform a wide range of tasks [10]. They are designed to be applied in any domain or industry, erasing the need to create or train a domain-specific model. Their ability to identify contextual relationships and recognize complex patterns [4] in a short amount of time makes LLMs the perfect entrypoint to automate triage and thus optimize the incident response workflow.

This research aims to explore the potential of LLMs in optimizing the triage process, establish ways to evaluate the performance of LLMs in cybersecurity, and present a comparison of different models when automating triage steps. To pursue our goal, we define the following research questions (RQ) as the basis of our research:

- **RQ1:** How can LLMs be integrated into the existing incident response workflow to streamline the triage process?
- **RQ2:** What suitable evaluation metrics should be used to assess the performance of LLMs in cybersecurity triage?
- **RQ3:** How do different LLMs compare in performance when optimizing the cybersecurity triage process?

(TODO: Structure)

2 OPTIMIZING TRIAGE USING LLMS

This section intends to answer RQ1: *How can LLMs be integrated into the existing incident response workflow to streamline the triage process?* Firstly, a rundown of the triage process is given by providing examples of the steps taken when identifying the priority of an alarm. Secondly, existing and proposed solutions of optimizing triage are summarized. Then, a general use of LLMs is given using a brief overview of the recent advances made in NLP, after which examples of LLMs performing relevant tasks are provided. Lastly, using these findings, possible automations for the triage process are identified.

2.1 Steps of Triage

The goal of triage is to follow a structured process to quickly assess and prioritize security alarms. Although the steps of triage are not set in stone and depend on the type of alarm, there are a number of basic tasks that can be followed:

(1) **Understanding the alarm.** The security analyst intends to understand the nature of the alarm. This includes the origin of the alarm (e.g., in the cloud or on-premise) and what time it was created.
(2) **Analyzing the context.** The analyst looks through the given alarm data and identifies the affected entities (e.g., users, data, systems or operations) and to what extent they are affected.
(3) **Correlating the alarm.** Based on the alarm context, the analyst searches whether the alarm has been seen before or if related alarms have occurred, either within the current network or a different one.
(4) **Identifying the position in the kill chain.** To assess potential consequences, the position of the alarm within the kill chain is determined. This is done by using the MITRE ATT&CK [32] framework or the Cyber Kill Chain [21] framework as a reference. This also depends on other potentially correlated alarms.
(5) **Prioritizing the alarm.** The analyst concludes how severe the alarm is and assigns a priority of high, medium, low or no threat.

The overall process should ideally not take longer than 30 minutes, because high-priority alarms need to be handled as urgently as possible. To meet this time constraint, it is essential to integrate tools and processes to allow analysts to perform triage efficiently.

2.2 Existing Triage Automations/Optimizations

One goal of triage is to correlate alarms, which entails identifying if the alarms are related to determine their placement within the kill chain. Ficke [9] optimizes this step by using alert trees. These trees are data structures used to organize and visualize generated alerts. The alerts are structured hierarchically, showing the relationships between alerts and the sequence of events. The proposed solution also eliminates redundancies in the graphs, thereby preventing the graphs from reaching sizes consisting of thousands of nodes. Based on academic datasets, the result is a system that quickly reconstructs paths that give insight into multistep threats in a network, which is an otherwise time-consuming task for human analysts. (TODO: How are alert paths detected (can't find full paper))

Additionally, Serketzis et al. [31] propose a model that integrates Cyber Threat Intelligence (CTI) into the process. CTI involves collecting and analyzing information about potential and existing threats. The model aims to enhance Digital Forensic Readiness (DFR), which is the preparation of digital forensics through collection and storage of data, to create relevant readily available information. Three independent but interrelated modules form the basis of the model:

(1) **IoC Collection Module.** Indicators of Compromise (IoC) consist of indicators of malicious activity observed in networks or systems. The module aggregates IoC from many internal and external sources of CTI to increase the collective knowledge. Data is evaluated and correlated to further increase its value, after which it is kept in a database.
(2) **Audit Log Processing Module.** This module aims to gather, validate and process audit log data generated by different parts within the organization. The data is stored in a dedicated database, which handles retrieval requests from the third module.
(3) **Threat Identification Module.** This module cross-matches the contents produced by the other modules and identifies threats. The evidence is stored in Intelligence Evidence Storage Systems (IESSs), which act as an entry point for analysts looking to preview potentially adverse incidents. Besides identifying suspicious activity, the module provides potential instigating factors.

The resulting model is shown to have a high accuracy of 90.73% when testing network data for malicious activity.

Besides that, Zhong et al. [37] approached the automation of triage by tracing the operations of professional security analysts. As junior analysts are typically responsible for conducting triage, this approach effectively speeds up the process. Finite state machines were constructed based on the senior analysts' patterns and were used to achieve high-speed triage with a low number of false positives. However, limitations include a high number of false negatives as well as a dependency on high-performing security analysts to maximize the performance of the automated system. Extending this approach, Lin [19] feeds contexts into a recurrent neural network which detects matching traces and presents these to novice analysts which in turn trains them in effective triage.

Finally, it is worth noting that a high level of automation can have adverse effects on the overall performance of security analysts when performing triage because of the following reasons:

- Understanding and conducting full-time monitoring of the automation can increase workload [13].
- The level of trust in automation can lead to over-reliance or neglect [16].

Hence, it is crucial to maintain a level of human interaction when automating the steps of triage, and to prioritize user-friendliness of integrated tools.

One limitation of all the previously proposed automations is that they do not involve NLP. Triage requires understanding and interpreting content surrounding the alarm, such as logs, announcements and other forms of natural or unstructured language, which are challenging to automate through conventional methods. Therefore, the following section will discuss how LLMs can be used to optimize such tasks and ultimately automate steps of triage.

2.3 General Usage of LLMs

(TODO: check relevance of following section)

LLMs make use of Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLU aims to comprehend meaning and intent in natural language, while NLG focuses on generating original human-like texts. To achieve NLU and NLG, language models make use of so-called encoders and decoders. The purpose of encoders is to turn input texts into fixed-size vectors that act as abstract representations. Language models then utilize decoders to transform such representations into a generated target output. This approach works well on tasks that map input sequences to output sequences (sec2sec), such as language translation [6, 33].

In 2014, Bahdanau et al. [2] introduced the concept of attention which rids the encoders of creating fixed-length vectors, allowing language models to focus on the most relevant parts of texts and enabling operations on much longer input sequences. Based on this, Vaswani et al. [35] developed the transformer architecture in 2017, setting the precedent for modern LLMs. Transformers are superior in quality, more parallelizable, and take less time to train.

Early well-known examples of such transformer-based models are BERT (Bidirectional Encoder Representations from Transformers) [8], and GPT (Generative Pre-trained Transformer) [27]:

- BERT was specifically designed as a pre-trained model to be easily fine-tuned for a wide range of tasks such as answering questions and natural language inference, without the need of task-specific architecture. In the cybersecurity domain, BERT models have been fine-tuned to detect malicious software [29] and phishing emails [17], in addition to performing general cybersecurity tasks [3].
- GPT is pre-trained on a large amount of unlabeled data and designed to generate coherent context-specific text. Like BERT, it requires fine-tuning to adapt the model to specific tasks.

The disadvantage of these models is that they are relatively small and require fine-tuning to be applied to domain-specific tasks. Fine-tuning using domain-specific data is not only resource and time-intensive, but the resulting models have limited applicability and can potentially include bias. Another disadvantage is the phenomenon of catastrophic forgetting where existing models forget their original knowledge after being trained on new data.

The introduction of much larger general models such as GPT-4 [1] and Llama 3 [22] intends to eliminate these problems. The broad availability and applicability of these large general models allows organizations to easily integrate them to optimize general and specific tasks. These models are relatively new and currently have limited research available on their use, but proposed applications of fine-tuned LLMs still provide valuable insights into their potential for optimizing triage.

For example, Karlsen et al. [14] propose a system that uses LLMs to perform log analysis. Where previous methods of analysis relied on rule-based or statistical approaches, LLMs are able to learn complex patterns and relationships within log data without requiring manual feature engineering. The study focuses on fine-tuning models such as BERT and GPT-2 [28] through self-supervised learning where the LLMs automatically label data by identifying relations. Larger models like GPT-4 and Llama 2 [34] were not used due to their high parameter counts, leading to increased computational requirements during the fine-tuning process.

(TODO: more examples of applied general models)

(TODO: system prompts / user prompts)

2.4 Using LLMs to Optimize Triage

When applying LLMs to specific tasks, it is important to keep the tasks short and simple to ensure the results are consistent and easily testable. Referring to the triage steps in 2.1, all tasks that involve natural language or unstructured textual data have the potential to be automated using LLMs. Since the optimizations are ideally applicable across all kinds of alarms, the following possible automations are identified:

- **Detecting if an email is an announcement.** This means letting the LLM process the entire email and concluding if it contains any information regarding actions that could trigger alarm generation.
- **Detecting if an announcement is related to an alarm.** This involves feeding both an announcement and alarm data into the LLM and determining whether a correlation exists, meaning the alarm was triggered by an action that was announced beforehand.

- **Correlating an alarm with other alarms and identifying if there are relationships.** Based on the alarm data, potentially related alarms are collected, after which the LLM concludes if they are in fact correlated.
- **Determining the position of an alarm in the kill chain.** Based on the MITRE ATT&CK framework, alarms have a position in the kill chain. An LLM can use alarm data and correlated alarms to identify this position.
- **Determining the priority of an alarm as high, medium, low or no threat.** This last step is to use the answers of the previous tasks to determine if the alarm should be treated as high, medium, low or no priority.

(TODO: explain what each step entails for LLM)

3 EVALUATION OF LLMS IN CYBERSECURITY TRIAGE

This section intends to answer RQ2: *What suitable evaluation metrics should be used to assess the performance of LLMs in cybersecurity triage?* Firstly, existing evaluation metrics for LLMs are identified. Finally, the most suitable metrics are determined to establish a testing framework for LLMs in the context of cybersecurity triage-related tasks.

3.1 Existing LLM Evaluation Metrics

Due to the inherent ambiguity of human language, it is challenging to evaluate the output of an LLM. Outputs of LLMs are not numerical in nature, but evaluation algorithms should produce a numerical score. This necessitates the use of sophisticated evaluation metrics. Besides simple human evaluation techniques like expert reviews and crowdsourcing, there are some notable automated metrics to measure LLM performance:

- The BLEU [26] score is specifically designed to test machine translation by matching output texts with reference texts.
- The ROUGE [18] score is used to evaluate text summaries by comparing model outputs with expected outputs.

These evaluation scores are purely statistical and thus reliable, but do not consider the nuances of semantics. They demonstrate a low correlation with human judgments, particularly in tasks related to creativity and diversity [20].

On the other hand, NLP-based evaluation techniques are more accurate but less reliable. Metrics such as BERTScore [36] and BLEURT [30] use descriptive LLMs such as BERT to provide a score by comparing generated and reference texts while taking semantics into account.

Besides that, Liu et al. [20] propose G-EVAL, a framework that uses generative LLMs such as GPT-4 or GPT-3 [5] to evaluate LLM outputs. First, evaluation steps are generated based on a given task and evaluation criteria. Then, the steps are used to assess an LLMs output given an input prompt and a score ranging from 1 to 5 is given. The resulting score takes semantics into account, and the resulting evaluation is more correlated with human judgment. However, it is unreliable due to the arbitrary nature of LLM output, and it is biased towards LLM-generated texts compared to human-written texts.

(TODO: section about Prometheus [15])

The assigned automations in the triage process are task-specific and only require tests on the correctness of the answer by the LLM. Answers are classified as true positive, false positive, true negative and false negative, depending on the actual classification of data the model's prediction. From these four classifications, different evaluation metrics can be constructed. The following task-specific metrics are suitable:

- **Accuracy** is the ratio of correct predictions to the total number of answers. It gives an overall indication of the model's ability to make correct predictions, but can be misleading if the testing or real-world data is imbalanced.
- **Precision** is the ratio of correct answers compared to all answers that were flagged positive by the LLM. A high precision indicates a low false positive rate.
- **Recall** is the ratio of correct predictions to the total number of actual positives in the test data. A high recall indicates a low false negative rate.
- **F1-score** is a harmonic mean of precision and recall. As a metric, it represents a balance between precision and recall, capturing the performance using both metrics.

3.2 Applying Evaluation Metrics to Triage

In the triage process, a large number of false positives would cause the alarm queue to be filled up, resulting in limited time for analysts to conduct thorough investigations. On the other hand, a large number of false negatives would result in critical alarms being missed. Therefore, it is important to balance these metrics when evaluating LLMs.

For the triage automations identified in section 2.4, the metrics are applied as follows:

- (TODO: apply metrics to context)

4 COMPARING LLMS IN CYBERSECURITY TRIAGE

This section intends to answer RQ3: *How do different LLMs compare in performance when optimizing the cybersecurity triage process?* (TODO: explain section)

5 CONCLUSION

Conclusions

6 DISCUSSION

Discussion. Include reflection on triage vs analysis.

DISCLAIMER ON THE USE OF AI

Besides the use of LLMs as needed for this research, during the preparation of this work, the authors will use generative AI tools such as ChatGPT, LLama3, Grammarly, JetBrains

Grazie and JetBrains full line code completion for the following purposes:

- Find definitions of terms and concepts when conventional tools and search engines are unsatisfactory.
- Check and correct the spelling of words and grammar of sentences.
- Improve the readability of sentences and paragraphs through rewording and restructuring.
- Use code completion functionality to speed up programming tasks.

After using these tools/services, the authors will review and edit the content as needed and take full responsibility for the content of the work. The services will not be used to produce scientific insights, create figures, draw conclusions or provide recommendations.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Markus Bayer, Philipp Kuehn, Ramin Shanehsaz, and Christian Reuter. 2024. Cysecbert: A domain-adapted language model for the cybersecurity domain. *ACM Transactions on Privacy and Security* 27, 2 (2024), 1–20.

[4] Sebastian Bordt, Ben Lengerich, Harsha Nori, and Rich Caruana. 2024. Data Science with LLMs and Interpretable Models. *arXiv preprint arXiv:2402.14474* (2024).

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[7] Anton Chuvakin. 2019. *The Uphill Battle of Triaging Alerts*. Dark Reading. https://www.darkreading.com/threat-intelligence/the-uphill-battle-of-triaging-alerts

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Eric Ficke. 2022. *Reconstructing Alert Trees for Cyber Triage*. Ph. D. Dissertation. The University of Texas at San Antonio.

[10] IBM Corporation. n.d.. *What are large language models (LLMs)?* IBM Corporation. https://www.ibm.com/topics/large-language-models

[11] IBM Corporation. n.d.. *What is natural language processing (NLP)?* IBM Corporation. https://www.ibm.com/topics/natural-language-processing

[12] IBM Security. 2023. Cost of a Data Breach Report 2023. https://www.ibm.com/reports/data-breach

[13] David B Kaber and Mica R Endsley. 2004. The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. *Theoretical issues in ergonomics science* 5, 2 (2004), 113–153.

[14] Egil Karlsen, Xiao Luo, Nur Zincir-Heywood, and Malcolm Heywood. 2024. Large language models and unsupervised feature learning: implications for log analysis. *Annals of Telecommunications* (2024), 1–19.

[15] Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491* (2023).

[16] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. *Human factors* 46, 1 (2004), 50–80.

[17] Younghoo Lee, Joshua Saxe, and Richard Harang. 2020. CATBERT: Context-aware tiny BERT for detecting social engineering emails. *arXiv preprint arXiv:2010.03484* (2020).

[18] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[19] Tao Lin. 2018. A Data Triage Retrieval System for Cyber Security Operations Center. (2018).

[20] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634* (2023).

[21] Lockheed Martin. 2011. Cyber Kill Chain. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[22] Meta. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date. https://ai.meta.com/blog/meta-llama-3/

[23] Orca Security. 2022. 2022 Cloud Security Alert Fatigue Report. https://orca.security/wp-content/uploads/2022/03/Orca-2022-Cloud-Security-Alert-Fatigue-Report.pdf

[24] Oxford English Dictionary 2024. artificial intelligence, n. In *Oxford English Dictionary*. Oxford University Press. https://doi.org/10.1093/OED/3194963277

[25] Oxford English Dictionary 2024. machine learning, n. In *Oxford English Dictionary*. Oxford University Press. https://doi.org/10.1093/OED/2166790335

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[27] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. (2018).

[28] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[29] Abir Rahali and Moulay A Akhloufi. 2021. MalBERT: Using transformers for cybersecurity and malicious software detection. *arXiv preprint arXiv:2103.03806* (2021).

[30] Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. BLEURT: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696* (2020).

[31] Nikolaos Serketzis, Vasilios Katos, Christos Ilioudis, Dimitrios Baltatzis, and Georgios Pangalos. 2019. Improving forensic triage efficiency through cyber threat intelligence. *Future Internet* 11, 7 (2019), 162.

[32] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. 2018. Mitre att&ck: Design and philosophy. In *Technical report*. The MITRE Corporation.

[33] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems* 27 (2014).

[34] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[36] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).

[37] Chen Zhong, John Yen, Peng Liu, and Robert F Erbacher. 2018. Learning from experts' experience: toward automated cyber security data triage. *IEEE Systems Journal* 13, 1 (2018), 603–614.