Formation of a Single Vector of Attributes for Data from Monitoring and Journalism Systems

Simar Muratov, Sergey Muravyov
ITMO University
Saint-Petersburg, Russia
{symuratov, smuravyov}@itmo.ru

Abstract — **The paper presents the development of a unified feature vector for data from monitoring and journalism systems, aiming to enhance the efficiency of modern software and infrastructure monitoring. Current machine learning-based monitoring tools facilitate a balance between incident response speed and solution effectiveness but encounter challenges such as high integration complexity, limited flexibility, and suboptimal operational performance. While cloud providers offer partial solutions, their applicability is restricted by corporate deployment constraints. Consequently, the creation of a specialized log processing complex to overcome these limitations is both relevant and necessary. A key component of this complex is the single feature vector, which is constructed through feature extraction, semantic analysis of logs, and evaluation of feature significance for anomaly detection models. This approach enables the formation of an efficient feature space, characterized by an optimal number of features and high informational value, ensuring robust semantic representation and model training quality. Semantic analysis reveals that logs are neutrally polar and lack inherent subjective significance due to their artificial origin. The unified feature vector is generated using a weighted extraction method, selecting features with the highest significance for tasks such as anomaly detection, classification, and log clustering. These features are subsequently aggregated into a single vector, categorized into static, semi-static, and dynamic groups based on the constancy and variability of log field values. The implementation of this vector significantly improves the overall quality and effectiveness of the log processing complex.**

Keywords: Machine learning, anomaly detection, feature extraction, logging, monitoring and logging systems.

I. INTRODUCTION

Monitoring and journalism systems play a critical role in modern cybersecurity and system maintenance, providing essential insights into system behavior and potential anomalies. However, extracting meaningful **features** from these systems remains challenging due to the diversity and complexity of log formats and structures. This paper addresses the problem of creating a unified **feature vector** for log data, enabling more accurate and reliable anomaly detection, classification, and clustering.

The proposed approach involves developing a comprehensive framework for transforming raw log data into a structured format suitable for machine learning algorithms. By leveraging techniques such as **feature extraction**, semantic analysis, and statistical modeling, we aim to enhance the efficiency and effectiveness of log processing tools like **Global Monitoring Tool** (**GMT**).

Our contribution lies in demonstrating how a unified **feature vector** can significantly reduce errors in anomaly detection while maintaining computational feasibility. Additionally, we provide practical examples and experimental results illustrating the benefits of our methodology across various types of log data.

In summary, this paper offers valuable insights into improving the reliability and scalability of log-based monitoring systems, particularly within the context of modern cybersecurity challenges.

II. DESCRIPTION OF DATA TYPE AND SYSTEMS

The main architectural component of data lake security – **Global Monitoring Tool** (**GMT**) during the information security audit had the following results, presented in Fig. 1 (version 1.0), for the period of the previous phase of the study [1, 2].

GMT is a variant of the log processing complex; in this case logs of operations performed in the data lake. The Fig. 1 (a) shows a graph of operations classified as safe and malicious. The Fig. 1 (b) shows a graph of extraction, transformation, and loading operations in the data lake using the MABAC cybersecurity model and batch processing mode.

The Fig. 1 (c) shows the values of the metrics of the transaction log processing complex. The Fig. 1 (d) shows a confusion matrix for classifying operations into safe and malicious based on the processed logs.

One of the key results and, at the same time, a motivating factor for the subsequent stages of the study is the magnitude of the second kind of error. Its value of 3% to 3.5% on average is unacceptable for commercial operation of **GMT** [3, p. 3]. Consequently, it is necessary to carry out several modifications to improve the quality of the **GMT**. The conducted analysis of the audit results revealed a significant drawback – excessively high adaptation to data sets due to the lack of implementation of methods **feature extraction** from logs, which makes it impossible to implement **GMT** in any log-generating system. The modifications to feature handling are divided into three steps:

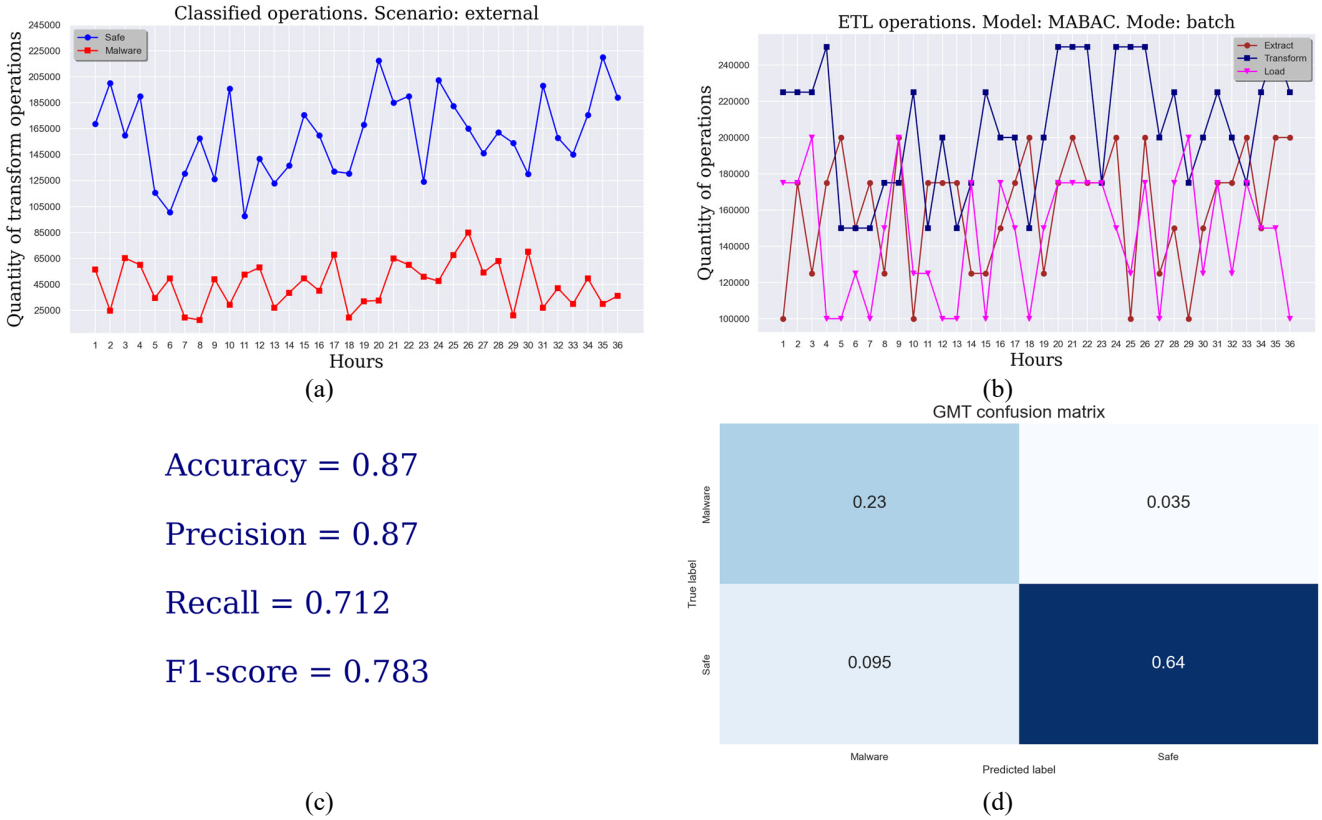


Fig. 1. Dashboard from a sample of GMT 1.0 audit results

1) formation of a single **feature vector**,

2) extraction of features for datasets based on the given vector,

3) automation of dataset generation for training, testing and validation of **GMT**.

This paper describes the first stage – the formation of a single **feature vector**, i.e., such a set of features that allows the **feature space** of the source dataset to be grafted onto the target dataset without loss of generality and semantic significance of logs. The possibility of forming such a vector is justified by the nature of logs.

Logs are a set of file objects that receive objective information about the processes occurring in the system. The recording is performed by software components that control the internal part of the system [9]. Each log file is an initial log of system events, from which a dynamic description of the target system state is formed by means of monitoring systems such as Zabbix or Prometheus [10].

Logs are the only data of their kind whose creation process is completely synthetic [11]. Unlike text, sound, images and video, logs have no natural origin. All logs are created by different systems: from IoT devices to data center hardware components, from a proxy server to a heterogeneous distributed storage system [4, p. 3].

The most relevant and representative log knowledge base is Loghub by Logpai. Loghub contains a collection of system logs that are freely available for research in artificial intelligence-based log analytics. Some of the logs are production data from previous research, others are collected from real systems in our lab environment [12]. Where possible, logs are not sanitized, anonymized, or modified in any way. These journal datasets are freely available for research and scholarly work [5, p. 4].

Currently, the advanced log anomaly detection systems are the solutions of foreign cloud providers: Amazon Web Services and Microsoft Azure. These solutions are AWS CloudWatch and Azure Anomaly Detector, their structure is shown in Fig. 2 and Fig. 3. The peculiarity of the solutions is the final composition of the metrics that are used to form anomaly detection decisions [13, 14]. In this case, the logs are generated at the cloud service level, which maximizes generality of application and multiple operational scenarios. However, such solutions are difficult to use in a closed network loop due to the inaccessibility of several key components. Moreover, due to economic sanctions, enterprise-level AWS and Azure services are not available in

Russia. There are currently no alternative solutions both among source code tools and domestic cloud providers. Existing proprietary monitoring systems have anomaly detection mechanisms, but they are implemented for specialized systems, i.e. the generality of application of such solutions is limited [6, p. 2].

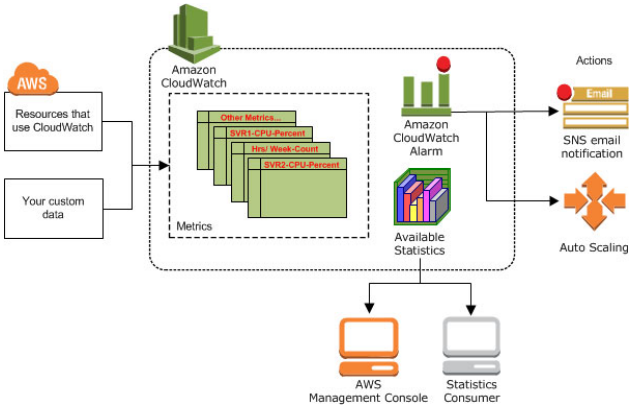


Fig. 2. AWS CloudWatch workflow

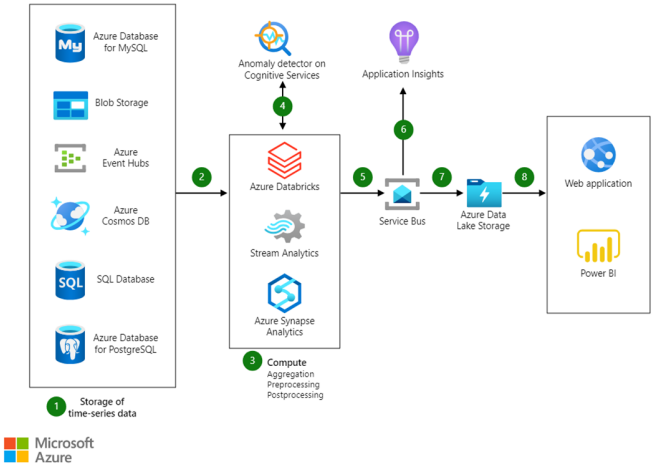


Fig. 3. Schematic of Azure Anomaly Detector operation

III. METHODS AND APPROACH

The **feature extraction** process is implemented by modifying the method described in the article on anomaly detection in data centers. The same principle of log record templating is presented in Fig. 4 [7, p. 5].

Further template enrichment was performed so that the templating process became available for logs hosted in Loghub as well. Consequently, it became possible to transform the log header into the extracted feature dataset header depicted in Fig. 5.

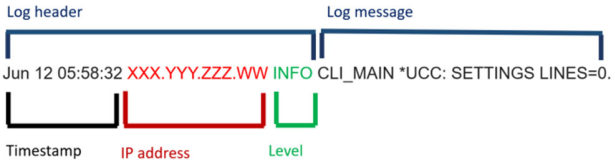


Fig. 4. Template sample of the puppet-agent service log record

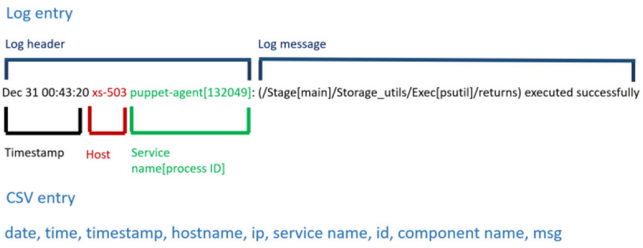


Fig. 5. Example of a log header converted to a .csv header

During the preliminary data processing stage, the format of log files is transformed into .csv using a service-specific procedure. This approach is necessitated by the differing structures of log headers across services each corresponding to a particular service. The procedure is designed to handle every log record and performs at least the following operations: adding missing fields, such as the Internet Protocol (IP) address or hostname; inserting the service name if absent; separating the service name and process identifier (ID); and extracting the component name.

Once a log file is converted to .csv format, each record contains the log message (msg) along with a set of additional variables. These variables include: date, time, timestamp, hostname, IP address, service name, ID, and component name. It should be noted that the hostname and IP address are not always available, particularly when the service operates within a virtual machine. Each file is associated with a specific service running on a known machine within the data center; its location is determined from a local database and is included in the resulting file.

IV. EXPERIMENTAL PROCESS

The experimental process was designed to validate the proposed methodology for forming a unified **feature vector** for log data, as well as to assess the effectiveness of semantic analysis in extracting meaningful features for machine learning tasks. The workflow included several key stages.

First, raw log datasets from different sources (Apache and Linux logs) were transformed according to the predefined **feature vector**. This transformation enabled the structuring of heterogeneous log data into a unified format suitable for further analysis and model training.

Next, semantic analysis was conducted on the processed logs. Word clouds were generated to visualize the most frequent terms in each log type, highlighting differences in vocabulary and event patterns between Apache and Linux logs, which is shown in Fig. 6 (a) and Fig. 6 (b). This step provided an intuitive overview of the informational content and helped identify domain-specific features.

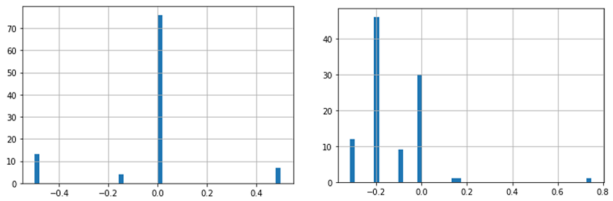
Subsequently, polarity and subjectivity analysis was performed to assess the semantic characteristics of log entries. Both Apache and Linux logs exhibited neutral polarity and low subjectivity, confirming the synthetic and objective nature of log data, which is shown in Fig. 7 (a), Fig. 7 (b) and Fig. 8 (a), Fig. 8 (b). These results support the hypothesis that

logs lack inherent semantic bias, making them suitable for automated **feature extraction**.

Thanks to the transformation, the datasets are obtained according to the given features. Further semantic analysis was performed: word clouds were constructed; polarity and subjectivity identification problems were solved.

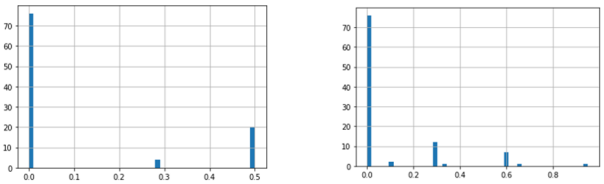
(a) (b)

Fig. 6. Word clouds for Apache log (a) and for Linux log (b)



(a) (b)

Fig. 7. Polarity for Apache log (a) and for Linux log (b)



(a) (b)

Fig. 8. Subjectivity for Apache log (a) and for Linux log (b)

The extracted features were then used to train and evaluate machine learning models for anomaly detection, classification, and clustering. The experimental results demonstrated that the unified **feature vector** significantly improves the quality and generalizability of log processing, enabling more accurate and robust analysis across diverse systems.

As a result of semantic analysis, the hypothesis that it is possible to form a single **feature vector** for logs was confirmed, since logs are as objective as possible, neutrally polar and contain synthetic language constructs with a variable number of user messages.

The vector was formed by calculating the significance of features for anomaly detection models: one class SVM, isolation forest, local outlier factor [15]. An example of the calculation result is shown in Fig. 9 and Fig. 10.

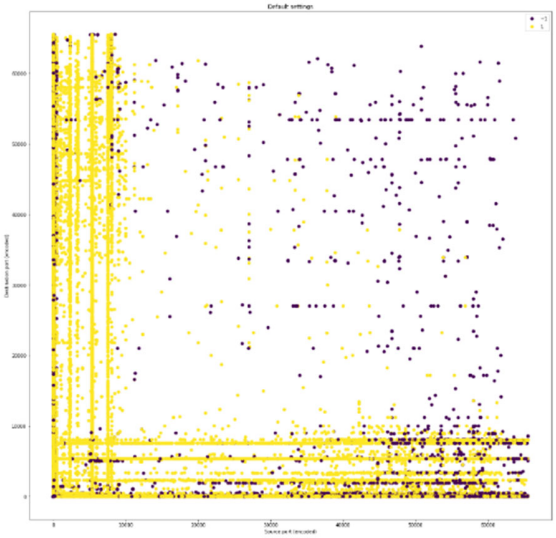


Fig. 9. Significance of the feature "port" without weights

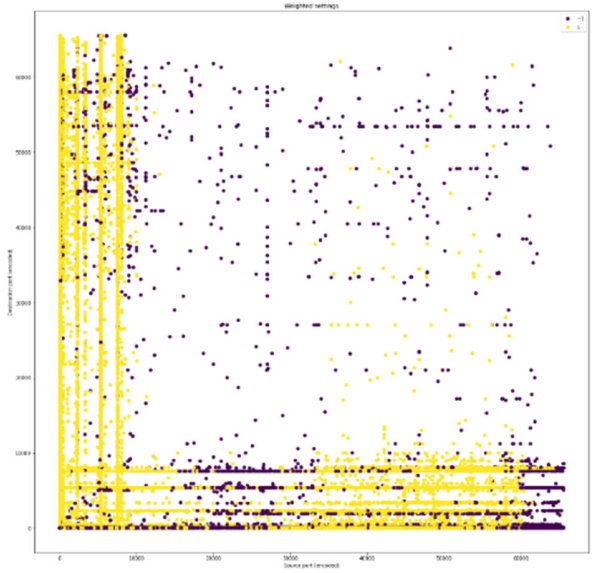


Fig. 10. Significance of the feature "port" with weights

Fig. 9 demonstrates the significance of the "port" feature calculated without applying weighting factors. The visualization reflects the raw contribution of this attribute to anomaly detection models, highlighting its relative importance in the unweighted feature space. This baseline assessment is essential for understanding the intrinsic informativeness of the feature prior to optimization.

Fig. 10 presents the significance of the same "port" feature after applying a weighted extraction method. The application of weights derived from feature importance metrics across multiple machine learning tasks: anomaly detection, classification, and clustering, which results in a refined representation of the feature's contribution. The comparison between Fig. 9 and Fig. 10 clearly shows that weighting enhances the discriminative power of the feature, thereby improving the overall quality of the feature vector.

These results confirm that the proposed weighted feature extraction approach effectively identifies and emphasizes the most relevant attributes for log analysis. Consequently, the unified feature vector not only reduces dimensionality but also preserves and amplifies the semantic and statistical significance of key features, which is critical for robust machine learning model training and generalization across diverse log-generating systems.

V. RESULTS AND DISCUSSION

The single vector, which is shown in Fig. 11, is the result of the first stage of modification of work with **features** for **GMT**. According to the given vector it is possible to unify the process of feature extraction for logs of any systems and its further inclusion in the automatic **GMT** work, which is based on the methodology of automatic machine learning [8]. The resulting vector serves as the foundation for a generalized feature space, ensuring compatibility between source and target datasets without loss of semantic integrity. This unification facilitates the automation of feature extraction and dataset generation processes, which is critical for large-scale log-based monitoring systems.

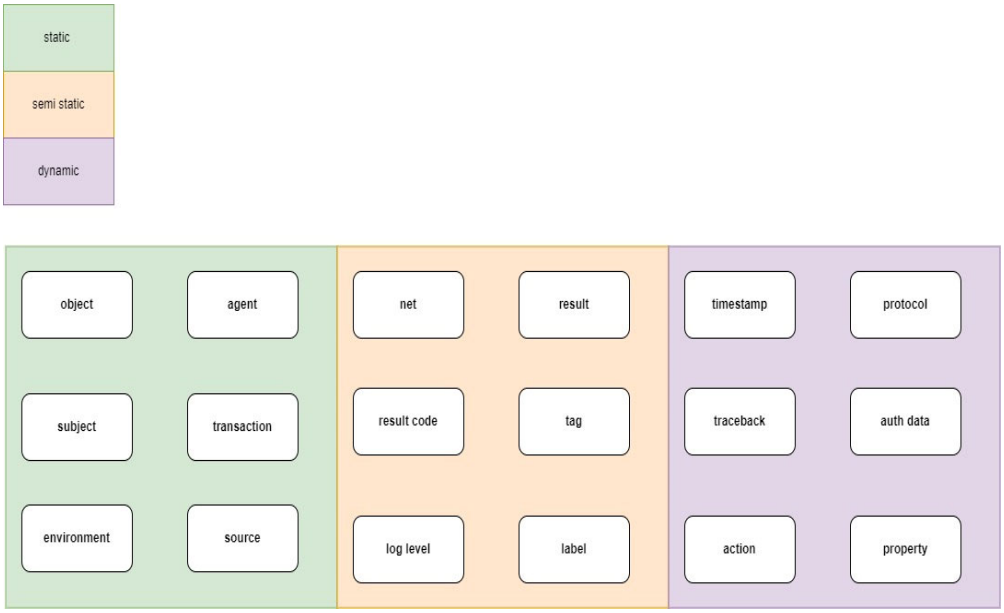
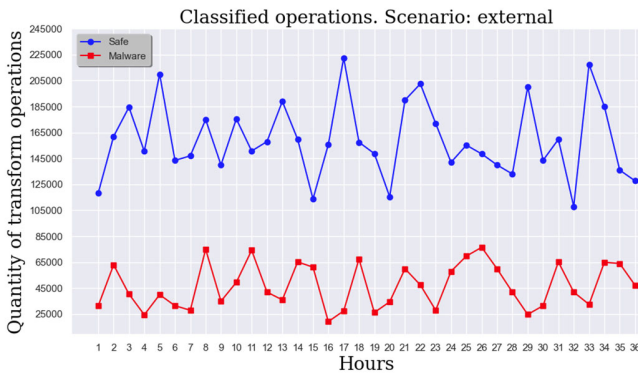
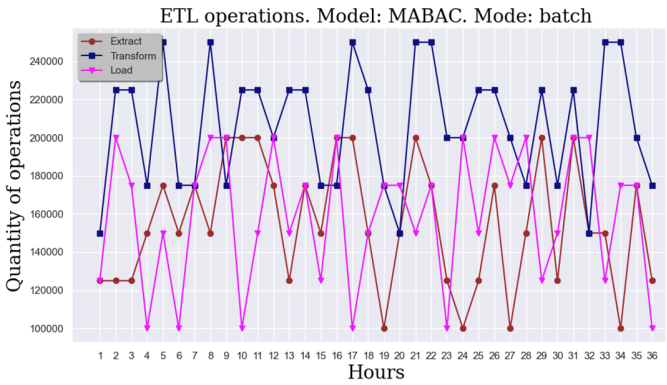


Fig. 11. Unified feature vector for logs

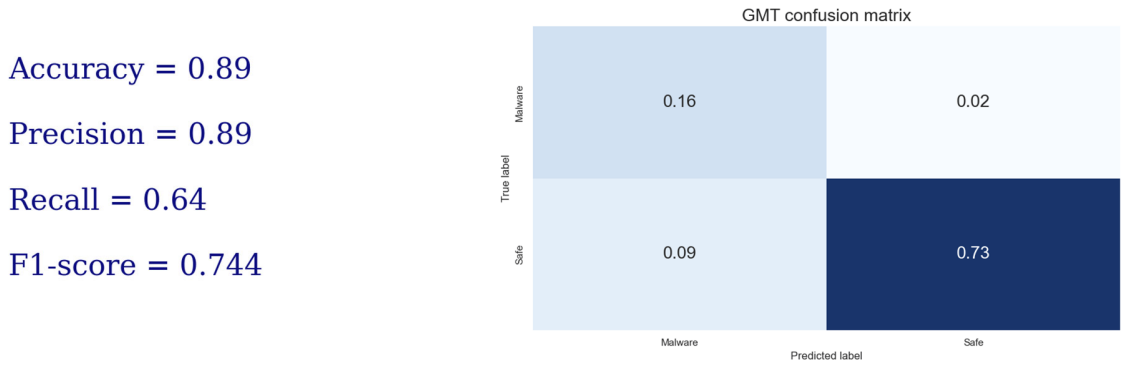
Fig. 12 (a-d), which is same type dashboard as shown in Fig. 1, presents the set of metrics for the complex version 1.5. The metric values have increased, while the rate of type II error has decreased. This increment includes the implementation of the developed algorithm for automated feature extraction. The positive dynamics of the increment are attributed to the improved adaptability of the complex's models to various generating systems, as well as to the reduction of feature space dimensionality without loss of semantic significance and relevance.



(a)



(b)



(c) (d)

Fig. 12. Dashboard from a sample of GMT 1.5 audit results with a generalized feature space based on a single feature vector

The experimental results confirm the central hypothesis of the paper: it is feasible to construct a unified feature vector for log data, enabling effective anomaly detection, classification, and clustering across heterogeneous systems. Semantic analysis demonstrated that logs are inherently objective, exhibit neutral polarity, and contain synthetic language constructs with variable user messages, which makes them particularly suitable for automated **feature extraction** and machine learning applications.

VI. CONCLUSION AND FUTURE WORK

This work demonstrates the successful development of a unified **feature vector** for log data, addressing the challenge of extracting meaningful features from diverse and complex log formats. Through the integration of **feature extraction**, semantic analysis, and statistical modeling, we have established a framework capable of enhancing the accuracy and reliability of anomaly detection, classification, and clustering tasks.

The unified **feature vector** enables the construction of an optimized **feature space**, ensuring compatibility between source and target datasets without compromising generalizability or semantic integrity. Experimental results confirm the neutral polarization and subjectivity of logs, validating the appropriateness of the proposed methodology.

Furthermore, the findings highlight the necessity of incorporating automated **feature extraction** and dataset generation processes to support large-scale log-based monitoring systems. Future research will focus on refining these methodologies and exploring additional applications beyond traditional cybersecurity contexts.

Overall, this work contributes valuable insights towards improving the reliability and scalability of log-based monitoring systems, advancing the capabilities of global monitoring tools and contributing to broader advancements in cybersecurity practices.

The unified **feature vector** of log data forms a generalized **feature space**, serving as the target for the log processing complex. Within the scope of this work, a **feature space** is considered generalized if all the following conditions are met:

1) the set of features represents an optimum quantity of features with maximum relevance for solving tasks related to anomaly detection, classification, and clustering of logs,

2) transition to this space does not entail loss of semantic significance of field values for log records,

3) log entries in this feature representation are equally relevant to the generating system as they are in the original **feature space**.

If any of these three conditions is violated during the process of **feature extraction** and selection aimed at transitioning to the generalized **feature space**, then such a space should be deemed weakly generalized. Furthermore, if two or all three conditions are breached, the target space cannot be regarded as generalized, making the transition impractical under those circumstances.

For each of the three tasks: anomaly detection, classification, and clustering, a specific algorithm must exist to facilitate the transition from the desired **feature space** to the target space. Such an algorithm compares overall **feature importance** in the sought-after space with similarly calculated importance in the target space, utilizing a predetermined level of significance. Generalization success of the target **feature space** is achieved when either preserving or improving total feature significance. Metrics evaluating model performance in machine learning serve as secondary indicators of transition quality, as the primary goal itself is bringing the searched-for log dataset into alignment with the target representation.

REFERENCES

[1] Muratov S.Y., Lukashin A.A. Development of architecture of protected big data lake framework // Modern technologies in the theory and practice of programming. 2022. p. 167-168.

[2] Muratov S.Y. Framework architecture of the protected big data lake // Almanac of scientific works of young scientists of University ITMO. 2023. T. 1. p. 112-117.

[3] Jones R., Omar M., Mohammed D., Nobles C., Dawson M. Harnessing the Speed and Accuracy of Machine Learning to Advance Cybersecurity. 2023 P. 418–421.

[4] Korzeniowski L., Goczyla K. Landscape of Automated Log Analysis: A Systematic Literature Review and Mapping Study. 2022. doi: 10.1109/ACCESS.2022.3152549

[5] Zhu J., He S., He, P., Liu, J., Lyu, M.R.: Loghub: A Large Collection of System Log Datasets for AI-driven Log Analytics. V. 10. 2020. P. 241-250. doi:10.1109/ACCESS.2022.3152549

[6] Monni C., Pezzè M. Energy-based anomaly detection a new perspective for predicting software failures. In Proceedings of the 41st International Conference on Software Engineering: New Ideas and Emerging Results. 2019. P. 69–72. doi: 10.1109/ICSE-NIER.2019.00026

[7] Viola L. Ronchieri E., Cavallaro C. Combining log files and monitoring data to detect anomaly patterns in a data center. 2022. V. 11. N. 8. P. 117–227. doi: 10.3390/computers11080117

[8] GitHub Homepage, https://github.com/HelenGuohx/logbert, last accessed 2024/03/04.

[9] Aharoni, E. & Fine, Shai & Goldschmidt, Yaara & Lavi, Ofer & Margalit, Oded & Rosen-Zvi, Michal & Shpigelman, Lavi. (2011). Smarter log analysis. IBM Journal of Research and Development. 55. 10:1 - 10:10. 10.1147/JRD.2011.2165675.

[10] Bilobrovets, I.. (2023). Network threat detection technology using Zabbix software. Modern Information Security. 54. 10.31673/2409-7292.2023.020003.

[11] Zhang, D. & Chen, Yuntian & Meng, Jin. (2018). Synthetic well logs generation via Recurrent Neural Networks. Shiyou Kantan Yu Kaifa/Petroleum Exploration and Development. 45. 598-607. 10.11698/PED.2018.04.06.

[12] Karanjai, Rabimba & Lu, Yang & Alsagheer, Dana & Kasichainula, Keshav & Xu, Lei & Shi, Weidong & Huang, Stephen. (2024). LogBabylon: A Unified Framework for Cross-Log File Integration and Analysis. 10.48550/arXiv.2412.12364.

[13] Piper, Ben & Clinton, David. (2019). CloudTrail, CloudWatch, and AWS Config. 10.1002/9781119560395.ch7.

[14] Khan, Sardar Mudassar. (2023). Microsoft Azure Anomaly Detector.

[15] Martinez Alonso, Rodney & Plets, David & Pollin, S. & Martens, Luc & Joseph, Wout. (2023). Outlier Detection and Spectrum Feature Extraction Based on Nearest-Neighbors Correlation and Random Forest Algorithm. 10.36227/techrxiv.24416629.