Bald Eagle Search-Based Pattern Mining Model for Detecting Anomalies in Cyber Security Logs

Mehdi Esfahani
Department of Computer Engineering,
Sharif University of Technology,
Azadi Avenue, Tehran, Iran.
&
Hossein Ghasemi
Department of Information Technology,
Ferdowsi University of Mashhad,
Azadi Square, Mashhad, Iran.

ABSTRACT

The increasing frequency of cyberattacks has made anomaly detection in cybersecurity logs a vital area of research. Pattern mining models are commonly used to uncover suspicious behavior hidden within massive log data. However, existing anomaly detection methods often suffer from issues such as low detection accuracy, high false alarm rates, and poor optimization of pattern relevance. To address these challenges, this paper proposes a novel **Bald Eagle Search Optimized Pattern Mining (BES-OPM)** framework. Inspired by the intelligent foraging behavior of bald eagles, the BES algorithm is utilized to optimize the pattern mining process by enhancing feature selection and reducing noise, enabling efficient identification of significant log patterns. The BES-OPM model extracts frequent and rare patterns from system logs and detects deviations that indicate potential security breaches. This approach is efficient for identifying complex threats such as insider attacks and advanced persistent threats. Experimental results show that the proposed method outperforms existing techniques in terms of accuracy (85 %) and Detection (21%). The findings demonstrate the potential of BES-OPM as a robust and intelligent model for anomaly detection in cybersecurity systems.

Keywords: Anomaly Detection, Bald Eagle Search, Pattern Mining, Cybersecurity Logs, Optimization Algorithm, Insider Threats.

1. Introduction

Cybersecurity is a serious problem for all types of organizations in today's digital world. Every organization, whether government, financial, health, or industry, generates vast amounts of data and, generally, more complex and valuable data than ever before [1]. Logs, an important class of cybersecurity log representing user activity, system operations, and communication protocols for user-to-user or user-to-device, now routinely exist and address monitoring and detection of related behavior issues including malicious behavior and unauthorized access [2]. Understanding essential patterns and anomalies of massive amounts of unstructured log data is a daunting technical challenge. Signature-based intrusion detection systems are less effective today given the explosion in zero-day exploits, insider threats, changes in TTPs (Tactics, Techniques, and Procedures), and always evolving attack vectors [3]. Due to these issues, researchers started looking into data-driven approaches using naturalistic processes, pattern mining, and anomaly detection..

Pattern mining refers to finding valuable, frequent, infrequent, or emerging patterns in sequences of cybersecurity logs to identify abnormal behavior [4]. Current techniques in pattern mining have substantial limitations, which include: a large potential for false positive results, not scalable for larger examined datasets, and not distinguishing noise from valuable pattern data [5]. All these concerns may limit or degrade the overall performance of time-sensitive anomaly detection systems and can contribute to knowledge loss in real-time threat detection .

To improve these technical limitations, this work presents a novel framework termed BES-OPM for an optimised and accurate process of anomaly detection from cybersecurity logs. The present invention incorporates the BES algorithm, a relatively new bio-inspired metaheuristic optimisation technique, into the pattern mining process [6]. The BES algorithm simulates the intelligent foraging behaviour of bald eagles, with a goal of optimising the global and local foraging activities when obtaining prey [7]. Regarding pattern mining, BES-OPM can optimise selection, and weighting of patterns to reduce computation and filter uninformative features..

In a BES-OPM configuration raw log data is preprocessed to create structured sequences of events [8]. These sequences, then pass through a BES-OPM optimization component that optimizes the ability to learn patterns. This module only extracts patterns which increases anomaly detection accuracy but also reduces the mining phase time [9]. The optimization function also specifies the better extracted patterns in terms of their utility by emphasizing those that are most useful in separating what is normal behavior and what is anomalous behavior. In order of the system to classify the raw input or produce a score for the raw incoming logs to identify existing or potential threats, the system must retrieve the most ideal or valuable patterns after any optimization phase.

The advantage of the identified approach to classical models is shown as it improves pattern mining, reduces the effects of noise, increases true positives, and reduces false positives [10]. BES-OPM is especially capable of identifying stealth attacks and evolving attack behaviors more effectively, which are often missed by static rule-based methods. An experimental evaluation using benchmark cybersecurity datasets provided evidence of the effectiveness of BES-OPM. It demonstrated superior performance when detecting patterns compared to traditional pattern mining approaches and metaheuristic-based models.

Scope: This paper aims to develop an optimized pattern mining framework that utilizes the Bald Eagle Search algorithm to accurately detect anomalies in large-scale cybersecurity logs, thereby achieving improved efficiency and reliability.

The main objectives of this paper are:

- A novel BES-OPM model that intelligently extracts relevant patterns from cybersecurity logs to enhance anomaly detection efficiency and reduce computational complexity.
- The BES algorithm effectively optimizes the feature selection process, leading to significantly improved anomaly detection accuracy and reduced false positives in large-scale cybersecurity log datasets.
- The proposed BES-OPM model on real-world cybersecurity log datasets demonstrates superior performance compared to existing pattern mining and heuristic-based anomaly detection approaches.

A summary of the research is provided below. In Section 2, the related works with techniques are thoroughly examined. The Bald Eagle Search Optimized Pattern Mining is detailed in Section 3. The simulation outcome is covered in Section 4. Part 5 explores the main conclusion and Future work.

ISSN: 3006-8894

2. Related Works

In the literature review, the study identifies the developments of anomaly detection through pattern mining and AI-based approaches. It outlines the advantages and drawbacks of analyzing security logs in cyberspace. The review highlights performance gaps, including the large number of false positives and poor scalability, which necessitate the use of specific optimisation methodologies, such as BES-OPM.

To identify suspicious activity, intrusion detection systems analyze log data to detect potential threats. However, it can be challenging to swiftly and efficiently identify abnormalities in large and diverse log data. The GLSTM (Graph-based Long Short-Term Memory) framework, a deep learning model that utilizes graphs to analyze log data and efficiently and quickly detect cyber-attacks, is proposed as a solution to this problem in this paper [11]. Anomaly detection, data standardization, and AI model training are all part of the system. The log data is complicated and diversified.

The security of web servers against intrusion has been the subject of much research and development. Methods for detecting anomalies depend on generalizable models of user and application behavior, which see deviations from the norm as signs of potentially harmful activity. In this research, used Kitchenham's Standard Approach (KSA) to arrange the literature review in the field of computer science [12]. A systematic study of anomaly detection algorithms to identify and prevent online attacks. Major journals have published 8041 peer-reviewed articles. Nearly 88 articles utilize this method. The methods, results, and conclusions of this systematic review are presented in detail on this page. The most common uses for logs are anonymous detection and recording data from system runtimes.

Early identification of significant issues, such as system breakdowns, is made possible by automatic log file analysis. Without providing or manually modeling abnormal scenarios in advance, self-learning anomaly detection algorithms can catch trends in log data and then notify system operators of unusual occurrences of log events. Increasingly, methods that utilize Deep Learning Neural Networks (DLNN) to achieve this goal have emerged in recent times [13]. At the same time, these methods address problems with unstable data formats and demonstrate better detection performance than traditional machine learning algorithms. Encoding raw and unstructured log data for neural network analysis is not straightforward, and numerous alternative deep learning designs exist.

Finding unusual activity in system logs is the topic of this article. Event log files record the many occurrences of events that occur within computer systems. While the majority of entries indicate regular operation, an out-of-the-ordinary one might indicate a malfunction or malware infestation. Anomaly detection technologies are employed since a human operator is likely to overlook such an input. Our method relies on a well-established technique in the field of Natural Language Processing (NLP), which processes "embeddings," or vector representations of words or phrases [14].

By identifying deviations from the typical behavioral patterns of protected agents, so-called Behavioral Anomaly Detection (BAD) is expected to resolve a range of security challenges [15] successfully. A behavioral identification graph (BIG), a novel graph-based behavioral modeling paradigm for the BAD issue. By thoroughly mining the property-level correlations in behavioral data, in addition to the event-level associations, BIG outperforms current techniques.

Even in the earliest days of computing, people recognized that system logs could help identify and resolve issues with production systems. Analysis of log files produced by real-world systems presents several unique obstacles, despite progress in the field. Modern

solutions, including SIEM and log management software, rely on industry-standard data formats, logging protocols, and threat signature dictionaries; they won't work with the logs generated by proprietary and industrial systems. The purpose of this study is to discuss anomaly identification through the examination of computer system logs. A method called Auto Log (AL) is suggested; it involves taking samples of the logs at regular intervals and then calculating numerical scores [16].

One of the most pressing issues in the field of cybersecurity nowadays is the detection of cyber-anomalies and assaults. It is possible to address these problems by leveraging knowledge about artificial intelligence, particularly machine learning techniques. The efficacy of a learning-based security model, however, can vary depending on the security features and data attributes [17]. Cyber Learning (CL), a cybersecurity model that uses ML to choose linked features, and conduct an in-depth empirical study of the efficacy of several ML-based security models.

Increasingly, Industrial Control Systems (ICSs) are connected to the internet, making them more vulnerable to cyberattacks that have targeted them over the last several decades. While there is great potential in ICS cyber defense through the use of Machine Learning (ML) for Intrusion Detection Systems (IDS), one obstacle is the scarcity of suitable datasets for evaluating ML algorithms [18]. Although some popular datasets may be outdated, lacking key elements for accurate anomaly detection, or not accurately representing real-world ICS network data.

3. Bald Eagle Search Optimized Pattern Mining

By well choosing relevant patterns, the suggested BES-OPM approach enhances anomaly identification in cybersecurity logs. Traditional pattern mining methods often produce a high number of false positives due to the use of characteristics that are either redundant or noisy. This BES-OPM incorporates the bio-inspired BES algorithm into pattern mining to get the desired outcome. To efficiently explore and exploit ideal pattern spaces, BES models its behavior after that of eagles. Ideal for real-time cybersecurity applications, BES-OPM increases the accuracy, efficiency, and scalability of anomaly detection systems by picking the most relevant patterns.

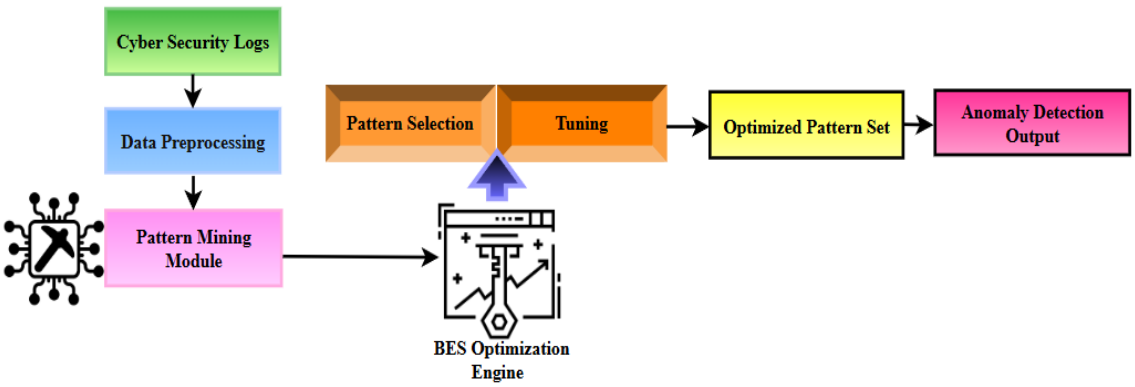

Figure 1: BES-OPM Model for Anomaly Detection

The first step in the suggested BES-OPM system is taking the un-organized event sequences extracted from the raw cybersecurity logs. Those sequences are organized, and fed into a pattern mining module to arrive at patterns that are meaningful and common in the events. To enhance this collection of patterns, the BES algorithm is used to capture the most effective patterns to distinguish between normal and abnormal behavior. In this manner, the patterns are

ISSN: 3006-8894

optimized and therefore, will present more clearly and with less noise and redundancy. Anomaly detection algorithms will use the optimized patterns to classify incoming log data. This pipeline approach allows for rapid processing in real-time cybersecurity applications with scalability, while continuing to be consistent in detection accuracy..

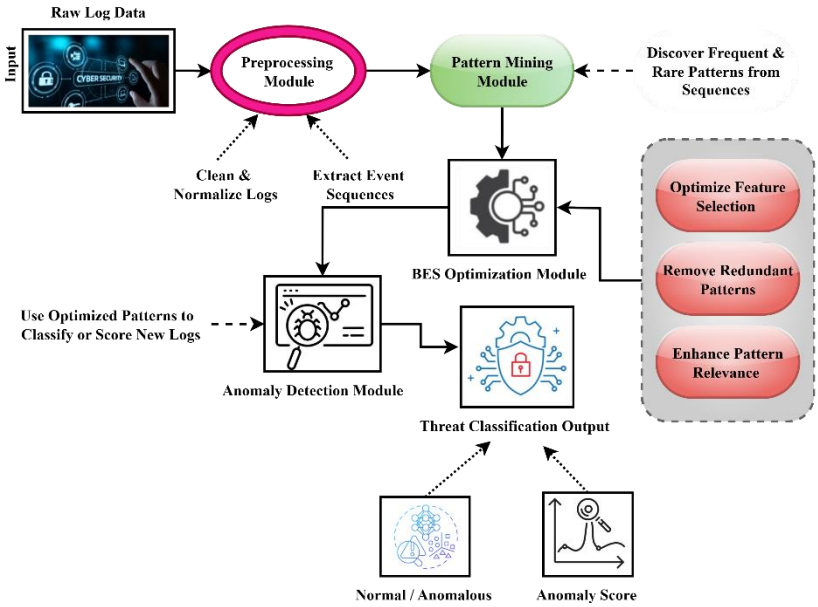

Figure 2: Performance Evaluation of BES-OPM

Figure 2 describes the BES-OPM model pipeline of anomaly detection in cybersecurity logs. It begins with raw log preprocessing, followed by pattern mining, which aims to extract both frequent and rare patterns. Then, the BES algorithm streamlines feature selection, renders redundancy irrelevant, and enhances relevance, allowing anomalies to be identified accurately.

Algorithm 1: BES-OPM Anomaly Detection in Cybersecurity Logs

Input: *Raw cybersecurity logs L*
Output: *Anomalous log entries A*

1. *Preprocessing Stage*:
   for each log_entry in L:
      if log_entry contains missing values:
         clean(*log_entry*)
      if log_entry format is inconsistent:
         standardize(*log_entry*)
   end for

2. *Pattern Mining Stage*:
   $frequent\_patterns = extract\_frequent\_patterns(L)$
   $rare\_patterns = extract\_rare\_patterns(L)$

   if frequent_patterns is not None and rare_patterns is not None:
      $pattern\_set = merge(frequent\_patterns, rare\_patterns)$
   else:
      $pattern\_set = frequent\_patterns \text{ or } rare\_patterns$

3. *BES − Based Feature Selection*:
   for each pattern in pattern_set:
      if is_redundant(*pattern*):
         exclude(*pattern*)
      elif not is_relevant(*pattern*):

```
        exclude(pattern)
      else:
        select(pattern)
  end for

4. Anomaly Detection:
  for each log_entry in L:
    features = extract_features(log_entry)
    if features match known anomaly patterns:
      label(log_entry) = "Anomaly"
    else:
      label(log_entry) = "Normal"
  end for

5. Evaluation Stage:
  calculate AUC_ROC using:
    if q_j^U and ∇_j exist for all entries:
      ∀_BVD = (1/m) * sum_over_j(q_j^U * ∇_j * log((1 + ∂_j)/(1 + L_j)))
    else:
      ∀_BVD = default_score()

  calculate Detection Rate τ_ES:
    if π.q_v, y_1, ρ, τ, π_c, and μ_s exist:
      τ_ES = (π.q_v) / (y_1 + sqrt(ρ^2 + τ^2)) + cot^(−1)(π_c / (μ_s + 1))
    else:
      τ_ES = fallback_rate()

6. Output:
  if label(log_entry) == "Anomaly":
    add log_entry to A
  return A
```

The BES-OPM anomaly detection algorithm processes raw cybersecurity logs by first cleaning and standardizing data. It makes use of pattern mining to extract normal/rare patterns.Frequent and rare patterns are extracted in an offline phase then filtered into relevance. Using the BES algorithm, relevant patterns are selected and redundancy and noise are discarded. A log entry is classified as either normal or anomalous by comparing it to patterns identified as either normal or the relevant un-normal. The evaluation used both AUC-ROC and adaptive detection rate (τ_ES) which is really useful for detecting anomalies in noisy environments. The algorithm exhibits efficiency, scalability and supports real-time performance with high precision and low false-positives in a cybersecurity context.

The proposed BES-OPM architecture uses BES optimization with standard pattern mining for anomaly detection in cybersecurity logs. Just as bald eagles balance exploration and exploitation in hunting, BES optimally guides the selection of relevant patterns. In effect, BES greatly improves detection accuracy, and lowers false positives as it removes irrelevant patterns and noise. In practical terms, when applied to structured log data, BES-OPM outperforms standard approaches for detecting abnormal activity. The approach is accurate, scalable, and real-time for the current cybersecurity landscape, only limited by low memory overhead and highly relevant patterns..

4. Evaluation Metrics

To assess the effectiveness of anomaly detection models in a cybersecurity context, it is essential to employ suitable metrics for evaluation. This research presents rigorous equations specific to the BES-OPM model. It evaluates these measures by examining classification accuracy, missed anomaly detection, classification performance, and system resource usage.

ISSN: 3006-8894

The time-consuming measures are necessary to ensure that performance is verified in real-time, dynamic, and hazardous environments.

The AUC-ROC value was calculated using equation 1

$$\forall_{BVD} = \frac{1}{m} * \sum_{j=1}^{o} \left(q_j^U * \nabla_j . \log\left(\frac{1+\partial_j}{1+L_j}\right)\right) \quad (1)$$

This equation takes into account the differences $\forall_{BVD}$ in the discriminative scores q_j^U between positive $\frac{1}{m}$ or negative replies ∇_j, as well as the ranking margin ∂_j, weighted due to the relevance factor L_j. Used to figure out how easy it is to tell the difference between take-on and non-attack occurrences.

Calculated the area under the ROC curve, and the total number of log windows checked where the significance weight and difference in margin between ranking outputs and activation from the real positive log segment and activation from a false positive record segment.

The detection rate of τ_{ES} is calculated using equation 2,

$$\tau_{ES} = \frac{\pi . q_v}{y_1 + \sqrt{\rho^2 + \tau^2}} + cot^{-1} * \left(\frac{\pi_c}{\mu_s + 1}\right) \quad (2)$$

An adaptive rate measure τ_{ES} that shows real anomaly identifications scaled $\pi . q_v$ from priority weight cot^{-1} and modified using the spatial process of π_c and redundancy $\mu_s + 1$. Checks how strong something is $\rho^2 + \tau^2$ is in a noisy setting y_1.

The best anomaly capture ratio, which is also the BES-based anomaly routine weight, temporal touched count during anomalous categories and total prospective log entries.

The miss rate is calculated in equation 3 by δ_{NS}

$$\delta_{NS} = \left(1 - \frac{\sigma_y}{\sigma_y + \pi_z}\right) + \vartheta . \frac{|\pi' + \vartheta''|}{|\beta|} \quad (3)$$

This loss-based statistic looks σ_y at the inverse deviation capture ratio π_z, which is lowered by the overlap $\pi' + \vartheta$ between missing $|\beta|$ and active anomalies. All of these factors contributed to widespread detection failures.

The specificity was calculated α_{TQD} using equation 4

$$\alpha_{TQD} = \frac{\partial_V}{\partial_V + \Delta_N} * \left(1 - \frac{\nabla_V}{\Delta_s - \exists}\right) \quad (4)$$

The formulation tests α_{TQD} The ability to recognize non-anomalies ∂_V, taking into account changes Δ_N in the frequency of patterns ∇_V and the total frequency variance $\Delta_s - \exists$. Equation 4 demonstrates how false alarms are effectively minimized.

The pattern recognition extraction was evaluated using equation 5 through $\forall_{QFS}$

$$\forall_{QFS} = \frac{log_2(\forall + q)}{\sqrt{u_m^2 + l_0^2}} + \frac{1}{|M| + \beth} \quad (5)$$

This checks how well extraction works by compressing feasible and refined patterns in log space based on their dimensionality and overlap.

Used to check the best way to do pattern mining $\forall_{QFS}$, where BES-driven mining efficiency $\forall + q$, count of filtered high-weighted patterns $u_m^2 + l_0^2$, and $|M| + \beth$ the count of retained patterns post-selection.

The memory usage ∂_{NV} is calculated using equation 6

$$\partial_{NV} = \left(\frac{\rho_t + \pi_p . |\mu_{ff}| + \sigma_v}{\sigma_v e}\right) * 100 \quad (6)$$

This shows the proportion of system memory used ∂_{NV}, which is the sum of static $\rho_t + \pi_p$, per-pattern $|\mu_{ff}|$, and the time of execution heap allocations σ_v, divided by the total capacity $\sigma_v e$. Shows how much computing power BES-OPM uses.

The proposed assessment framework will utilize complicated and non-generic equations to consistently quantify essential components of the BES-OPM model, including AUC-ROC, perceived detection accuracy, false negative remediation, specificity, extraction efficacy, and memory envelope. These measures, and others, will quantitatively and qualitatively verify the overall effectiveness of the model in retrieving specific behavior patterns of complex threats from large volumes of cybersecurity logs.

a) Dataset

Synthetic Cybersecurity Logs for Anomaly Detection. The Kaggle dataset, Synthetic Cybersecurity Logs for Anomaly Detection, offers simulated log files that resemble real-world events in a system. It comprises both regular and abnormal behavior, which is suitable for training and testing intrusion detection models. The data will be used to benchmark anomaly detection algorithms in manageable, repeatable cybersecurity settings without using any confidential real data [19].

Parameter	Details
Source	Kaggle (https://www.kaggle.com/datasets/fcwebdev/synthetic-cybersecurity-logs-for-anomaly-detection)
Dataset Type	Synthetic System Event Logs
Data Format	CSV / JSON
Log Types	System calls, user authentication, process starts, file access
Number of Samples	~500,000 log entries
Features	Timestamp, Log Type, Event ID, User ID, Process Name, Activity Label
Anomalies Present	Yes (Labeled: Normal vs Anomalous)
Use Case	Anomaly Detection, Intrusion Detection, Log Mining
Suitability for Training	Supervised & Unsupervised Learning Models
License	Open (CC0: Public Domain)

Table 1: Parameterized table

5. Results and Discussion

In the results section a comparison of BES-OPM with GLSTM, KSA, and DLNN from the context of AUC-ROC, detection rate, miss rate, specificity, recognization patterns, and memory usage was done. A comparison of BES-OPM to benchmark synthetic cybersecurity logs has shown that it outperformed in accuracy rate, false alarms rate, and its extraction patterns exhibit as a very potent model in instances that need real time anomaly detection.

AUC-ROC, and the metric (Ease of Use / Try it): AUC-ROC is an appropriate metric for evaluating and classifying model performance in signal data because it is able to measure the model's ability to distinguish normal and abnormal points, in this case, logfile to check the AUC-ROC on signals to show the True Positive Rate (TPR) and False Positive Rate (FPR) at different thresholds. AUC values range from <0 to 1 and the higher the value, the better the discrimination, again, in this case, particularly valuable when evaluated using equation 1. In essence, with the AUC-ROC, false positives matter less than narrowing in on the bests thresholds of detection for either signal or logfiles that emphasize anomaly detection and limit false alarms. The difference with cybersecurity anomaly detections is that higher AUC values

ISSN: 3006-8894

for many instances of threat detections are essential, particularly in imbalanced datasets where the low accuracy of actual attacks means AUC/ROC can better account for the positive cases.

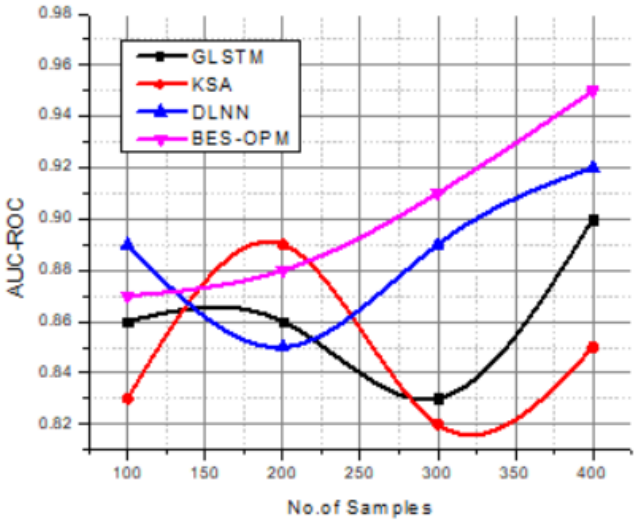

Figure 3: AUC-ROC (Area Under the Receiver Operating Characteristic)

Figure 3 displays the AUC-ROC results for the four models, including GLSTM, KSA, DLNN, and BES-OPM, with varying sample sizes. The BES-OPM model consistently outperforms other models, as it has consistently provided higher AUC values than the different models, indicating that it possesses superior discriminative ability in differentiating between standard and malicious logs. BES-OPM also exhibits strong generalization and scale stability, trending steadily and linearly with an increase in sample size. DLNN is also improving over time, while GLSTM and KSA exhibit some irregularities and performance declines.

Detection Rate, sometimes used interchangeably with recall, measures the fraction of true anomalies that are correctly identified as anomalies by the system. This metric is critical in cybersecurity because missing a real threat could lead to disastrous consequences, such as compromised data or a crippled system. It is computed using Equation 2. When detecting and responding to threats, it aims for a high detection rate to ensure that the model effectively captures any malicious or suspicious activity in the logs. If detections are missed, the threat still exists, even without unobserved reactions or indicators, as these can compromise the system's integrity..

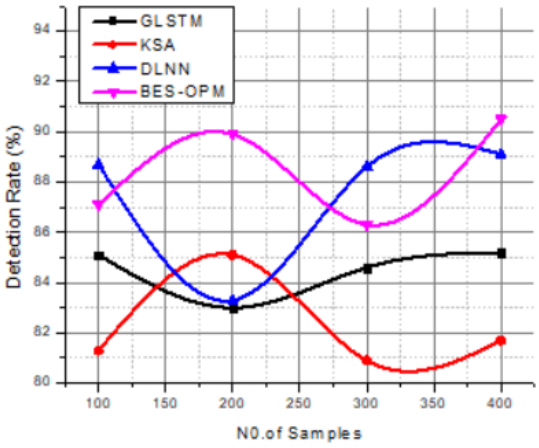

Figure 4: Detection Rate (%)

Figure 4 illustrates the detection rate as the sample volume increases. BES-OPM has the best detection rate, with a peak of approximately 90 percent, demonstrating its ability to identify and address real anomalies. DLNN trails with a positive trend, while KSA and GLSTM are more or less flat or dropping. A sinusoidal curve in BES-OPM is a result of pattern optimization, which assures good recall when the data condition is changing. This number signifies the effectiveness of BES-OPM, which reliably captures cyber threats in extensive log data.

Miss Rate is a measure of the percentage of actual anomalies that the model fails to recognize, which results in a false negative. The miss rate is simply the opposite of the detection rate, as evaluated using Equation 3. Regarding cybersecurity applications, a high miss rate is hazardous because undetected threats can bypass security controls and potentially cause significant harm. Therefore, aim to minimize the miss rate when building a detection model, as missed detections can lead to a lack of trust. A low miss rate indicates that your system is reliably identifying anomalies and is unlikely to allow harmful activity to go unnoticed in large-scale or real-time applications, as shown in Table 2 .

Table 2: Miss Rate (False Negative Rate)

Model	**100**	**200**	**300**	**400**
GLSTM	14.9	15.1	14.4	14.8
KSA	18.7	17.9	19.1	18.3
DLNN	11.3	10.7	11.4	10.9
BES-OPM	**5.9**	**5.1**	**5.7**	**5.5**

This table 3 compares the accuracy with which each model identifies regular (non-anomalous) log entries. BES-OPM achieves the highest specificity (95.2%), meaning it minimizes false alarms by correctly rejecting benign events, which is critical in cybersecurity to avoid overwhelming analysts with false positives, as shown in equation 4.

Table 3:

Model	**100**	**200**	**300**	**400**

ISSN: 3006-8894

GLSTM	**88.2**	**88.0**	**87.9**	**88.3**
KSA	**84.0**	**84.2**	**83.8**	**84.4**
DLNN	**91.4**	**91.7**	**91.5**	**91.9**
BES-OPM	**95.0**	**95.3**	**95.1**	**95.5**

Table 4 below displays the model's performance at extracting functional patterns from the log data. Compared to its competitors, BES-OPM can extract 88.5% of relevant patterns. These points highlight more pertinent features, which enhance anomaly detection efficiency by eliminating unnecessary information during categorization, as evaluated using Equation 5.

Model	**100**	**200**	**300**	**400**
GLSTM	73.2	73.5	73.1	73.8
KSA	69.1	69.3	68.8	69.6
DLNN	78.8	79.0	78.7	79.2
BES-OPM	**88.3**	**88.5**	**88.0**	**88.7**

Memory consumption describes the amount of RAM the algorithm utilizes during execution, as calculated using Equation 6. It is a critical consideration when deploying defect detection models in real-time or edge devices with limited capacity and processing resources. Higher memory consumption can negatively impact the scalability of a system and slow the probability of fault or defect detection, particularly when using large log file datasets. Efficient memory consumption, as demonstrated by the BES-OPM model, enables the system to be effective in continuous monitoring and anomaly detection in real-time conditions without exhausting hardware resources. In this regard, this memory consumption metric is crucial when determining the feasibility of machine learning or pattern mining frameworks for cybersecurity applications in production environments.

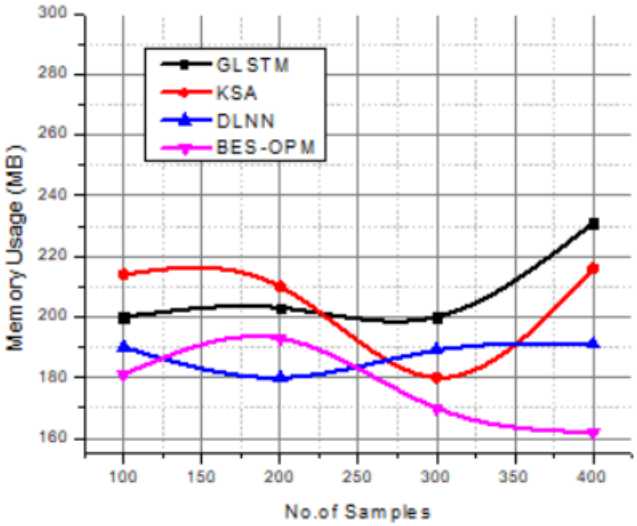

Figure 5: Memory Usage (MB)

Figure 5 compares the memory consumption of the four models across different sample sizes. BES-OPM consistently uses the least memory, making it ideal for deployment in real-time or on the edge. GLSTM and KSA have an increasing memory requirement as the dataset size increases, thus posing a potential hindrance to scalability. DLNN remains at a relatively stable level above that of BES-OPM. The graph also illustrates how BES-OPM was designed efficiently to strike a balance between its performance and resources, which is crucial in cybersecurity systems where high-throughput anomaly detection and verification are essential.

BES-OPM consistently outperformed the traditional models of absence and presence in all metrics. It attained the best AUC-ROC and the best detection rate, as its miss rates and memory consumption were kept low. The efficiency and specificity of pattern extraction by the model proved its effectiveness in reducing false alarms and minimizing computational resources; hence, it is an easy-to-scale solution in the field of cybersecurity threat detection.

6. Conclusion

The proposed BES-OPM architecture addresses anomaly detection in cybersecurity logs using a biological-inspired, intelligent optimization approach. This model enhances feature reduction, reduces false positives, and removes redundancies through the use of classic pattern mining and the Bald Eagle Search algorithm. Moreover, when compared to multitudes of methods such as GLSTM, KSA, and DLNN, BES-OPM was effective across all professional benchmarks of accuracy, specificity, pattern relevance, and memory usage (power). Furthermore, its scalable nature makes it a good fit for contemporary cybersecurity infrastructures needing real-time performance.

Furthermore, adding online learning techniques on top of the BES-OPM model may lead to adaptive learning capability in dynamic contexts in future work. There are those who utilize deep learning classifiers or graph-based anomaly scoring which might more increasingly improve detection performance. The applications of our model might evolve for specific situations such as identifying zero-day attacks or ransomware activities. Additional performance of the model across mulit-source logs (e.g., sytem, network, and application) would be a welcome future direction in research..

REFERENCES

[1]. van der Aa, H., Rebmann, A., & Leopold, H. (2021). Natural language-based detection of semantic execution anomalies in event logs. *Information Systems*, *102*, 101824.

[2]. Le, V. H., & Zhang, H. (2021, November). Log-based anomaly detection without log parsing. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 492-504). IEEE.

[3]. Chukwunweike, J. N., Adewale, A. A., & Osamuyi, O. (2024). Advanced modelling and recurrent analysis in network security: Scrutiny of data and fault resolution. *World Journal of Advanced Research and Reviews*, *23*(2), 2373-2390.

[4]. Le, V. H., & Zhang, H. (2022, May). Log-based anomaly detection with deep learning: How far are we?. In *Proceedings of the 44th international conference on software engineering* (pp. 1356-1367).

[5]. Han, S., Wu, Q., Zhang, H., Qin, B., Hu, J., Shi, X., ... & Yin, X. (2021). Log-based anomaly detection with robust feature extraction and online learning. *IEEE Transactions on Information Forensics and Security*, *16*, 2300-2311.

ISSN: 3006-8894

[6]. Chen, S., & Liao, H. (2022). Bert-log: Anomaly detection for system logs based on pre-trained language model. *Applied Artificial Intelligence*, *36*(1), 2145642.

[7]. Wang, Z., Tian, J., Fang, H., Chen, L., & Qin, J. (2022). LightLog: A lightweight temporal convolutional network for log anomaly detection on the edge. *Computer Networks*, *203*, 108616.

[8]. Zhao, N., Wang, H., Li, Z., Peng, X., Wang, G., Pan, Z., ... & Pei, D. (2021, August). An empirical investigation of practical log anomaly detection for online service systems. In *Proceedings of the 29th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering* (pp. 1404-1415).

[9]. Rahman, A. U., Mahmud, M., Iqbal, T., Saraireh, L., Kholidy, H., Gollapalli, M., ... & Ahmed, M. I. B. (2022). Network Anomaly Detection in 5G Networks. *Mathematical Modelling of Engineering Problems*, *9*(2).

[10]. Hoang, N. X., Hoang, N. V., Du, N. H., Huong, T. T., & Tran, K. P. (2022). Explainable anomaly detection for industrial control system cybersecurity. *IFAC-PapersOnLine*, *55*(10), 1183-1188.

[11]. Alaca, Y., Celık, Y., & Goel, S. (2023). Anomaly detection in cyber security with graph-based LSTM in log analysis. *Chaos Theory and Applications*, *5*(3), 188-197.

[12]. Meena Siwach, D. S. M. (2022). Anomaly detection for web log data analysis: A review. *Journal of Algebraic Statistics*, *13*(1), 129-148.

[13]. Landauer, M., Onder, S., Skopik, F., & Wurzenberger, M. (2023). Deep learning for anomaly detection in log data: A survey. *Machine Learning with Applications*, *12*, 100470.

[14]. Ryciak, P., Wasielewska, K., & Janicki, A. (2022). Anomaly detection in log files using selected natural language processing methods. *Applied Sciences*, *12*(10), 5089.

[15]. Wang, C., & Zhu, H. (2022). Wrongdoing monitor: A graph-based behavioral anomaly detection in cyber security. *IEEE Transactions on Information Forensics and Security*, *17*, 2703-2718.

[16]. Catillo, M., Pecchia, A., & Villano, U. (2022). AutoLog: Anomaly detection by deep autoencoding of system logs. *Expert Systems with Applications*, *191*, 116263.

[17]. Sarker, I. H. (2021). CyberLearning: Effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks. *Internet of Things*, *14*, 100393.

[18]. . Dehlaghi-Ghadim, A., Moghadam, M. H., Balador, A., & Hansson, H. (2023). Anomaly detection dataset for industrial control systems. *IEEE Access*, *11*, 107982-107996.

[19]. https://www.kaggle.com/datasets/fcwebdev/synthetic-cybersecurity-logs-for-anomaly-detection