Integrating Big Data and Machine Learning for Effective Cyberattack Prediction in e-Health Information Systems

Mohamed Abdelbaki[1]*, Latif Adnane[2], Charaf Eddine Ait Zaouiat[3]
Information Technology and Modelling Team-ENSA Marrakech, Cadi Ayyad University, Marrakech, Morocco[1, 2]
Polydisciplinary Faculty of Sidi Bennour, Chouaib Doukkali University, ELJADIDA, Morocco[3]

***Abstract*—This study proposes an intrusion-prediction framework for e-Health information systems that combines structured web-log analysis, supervised machine learning, and Apache Spark-based distributed processing. A corpus of 1,000,000 labeled HTTP log instances collected from a university hospital web environment was preprocessed into security-relevant features, including request method, request/response type, packet size, status code, URL length, and parameter count. Using a stratified 80/20 train-test split and five-fold cross-validation on the training data, we compared K-Nearest Neighbors (KNN), Logistic Regression, and Decision Trees. KNN achieved the best held-out performance, with 95.66% accuracy, 91.79% precision, 93.93% recall, 92.85% F1-score, and a 3.60% false positive rate. Logistic Regression and Decision Trees reached accuracies of 85.30% and 83.20%, respectively. Spark also reduced runtime substantially at the 1,000,000-instance scale, lowering KNN processing time from 12.0 s to 6.5 s. The results show that combining big data infrastructure with carefully tuned machine learning can improve both detection quality and operational feasibility in hospital cybersecurity monitoring.**

***Keywords*—Artificial intelligence; big data; cybersecurity; hospital information systems; log files**

I. Introduction

This exponential development of information and communication technologies rests on the increased utilization of more and more connected devices and tools in recent years. Through such evolution, users have easily and rapidly accessed information [1]. However, in the process, it has also facilitated illicit access to sensitive resources by malicious actors. The big challenge in the detection and prediction of attacks on communication systems, especially for hospital information systems, electronic medical records, and electronic health records, is related to the detection and prediction of such attacks [2]. Among the most attacked resources are those provided by web applications used in healthcare environments [3]. The log files record every event happening during system operation and are large in volume and complex in structure. Analysis of such log files is, therefore, of prime importance for addressing the risks involved with system attacks that may affect patient data and overall performance. Because of the huge size of these log files, it is not possible to conduct their manual analysis; extended processing times need to be used [4]. The novelty of this work lies in designing a cyberattack prediction framework that balances accuracy, scalability, and robustness to imperfect training data. Because security log datasets may contain noisy, inconsistent, or manipulated records, robustness becomes an important consideration alongside predictive performance [35]. This work will focus on the accuracy of different predictive algorithms, along with their time complexity, to provide an overall framework that will help administrators in healthcare identify and mitigate vulnerabilities effectively. The time complexity comparison analysis taken by algorithms used secures the development of an efficient solution to be able to handle volumes and improve efficiency, while personal data on patients is protected.

Also, by embedding Apache Spark as a big data processing engine, the model increases its speed of processing information and thus allows real-time analysis of large volumes of log data; in our evaluation, Spark supports distributed processing across scales up to 1,000,000 instances, addressing the practical scalability constraints of security analytics over large healthcare log volumes. This establishes a dual emphasis on predictive accuracy and computational efficiency, positioning the research as a practical contribution to improving the security posture of healthcare web applications.

The rest of this study is organized as follows. First, some related works concerning log file analysis and cybersecurity techniques are reviewed. Following this, the advantages and challenges in log file analysis applied to healthcare are discussed. Then, Big Data technologies and their importance in regard to security issues are described, and afterwards, machine learning tools and their practical applications in log analysis. The proposed approach and the implementation follow together with discussions of results.

II. Related Work

Basically, log file analysis refers to the intended examination and analysis of log files [5]. Log files are very important in tracking mistakes in data transmission, as well as tracking firewall activities [6]. Medical log file analysis is one of the most vital analyses that can be used in enhancing hospital information systems and electronic health records (EHR) [7]. Log files record not only the interaction of visitors with web applications but also trace various technical errors related to networks, software, and components, most of all, though, underlying security issues regarding patient data [8, 9]. However, the analysis of security logs is complex and requires advanced technical skills related to Big Data, Machine Learning, and both [30].

*Corresponding author.

Several works have described how Big Data and Machine Learning can be applied in security log file analysis. Landauer et al. [10] proposed a dynamic anomaly detection approach using clustering mechanisms to enable a self-learning algorithm to detect anomalies related to temporal behavior. On the other hand, Zhong et al. [11] proposed a deep learning log analysis-based intrusion detection approach based on the application of LSTM networks. Skopik et al. [12] investigated a unified pipeline that realizes several machine learning algorithms to analyze the system behavior and to identify deviations from established patterns, which in turn allows identifying both known and unknown attacks. Recent IDS surveys also show that Machine Learning and Deep Learning-based intrusion detection must address dataset quality, evasion techniques, false positives, and the detection of evolving attacks, which remains highly relevant for healthcare web environments [33, 37].

In the Big Data log file analysis context, Azizi et al. [13] proposed an approach where MapReduce could be used for the analysis of security-related log files concerning certain types of attacks, such as SQL injection or DDoS. Jeon et al. [14] proposed a new Big Data-based security logging system that extends security intelligence by analyzing data events created in enterprise log files, covering systems, application services, and IT infrastructure.

Despite these developments, most of the earlier studies have shown suboptimal performance because of data pre-processing deficiencies, which reduce the accuracy and performance. Most research typically only gives general suggestions for improving security without quantifying the impact, whereby limited practical applicability can be achieved from such recommendations. In addition, many of them have not explained the very crucial issue of processing time in terms of the steps involved in pre-processing, learning, classification, and prediction. Without this, a solution will not be feasible in real-world situations. In turn, our approach tries to fill these gaps by proposing an end-to-end system that emphasizes strict data pre-processing in order to derive from log files high-quality and relevant features that will enhance accuracy and improve predictive performance after proper refinement of the data, and then the application of machine learning algorithms. Moreover, the time complexity analysis for each algorithm allows us to propose an efficient model able to perform the right balance between accuracy and speed of processing. The presence of Apache Spark accelerates data handling and makes the entire workflow smoother, therefore making our solution ideal for big log file analysis in real-time healthcare contexts. The two most important goals-precision and speed-are huge guarantees that our research is one of the great milestones within the field of cybersecurity in general and the elaboration of practical and efficient solutions concerning the identification and mitigation of potentially vulnerable points in healthcare information systems.

III. Big Data Toolkit for Log File Security Analysis

Big Data is a concept that copes with the research, analysis, capture, storage, sharing, and presentation of data. Big Data replaces traditional tools, since those are inefficient in managing and analyzing such large sets of information. Often characterized by the three "Vs": volume, velocity, and variety. Big Data integrates other important aspects such as Value and Veracity [15, 16].

Big Data finds applications across a wide range of industries operating with huge volumes of data every day and usually requires crucial speed [17]. The prominent fields where Big Data can find its central role include Marketing, Monitoring, and Security, among others [18, 19]. In security, Big Data contributes substantially to enhancing the capability that deals particularly in the capture, filtration, and analysis of millions of network events per second, log and audit files included [20]. It provides that third level of protection against cyber-attacks involving:

- First-level security: company security against external attacks [21].
- Second-level security: company protection from vulnerabilities imposed by its users [21].
- Third-level security: impact assessment of the threats detected within infrastructure, which means network traffic back-tracing capability for tracking malware steps and proper action [22].

Big Data analytics tools actually enable cybersecurity experts to analyze a variety of data types from all different sources in real time. These tools are not limited to gathering information but also create correlations and connections [23, 24]. Amongst the major analytical tools for security are:

- Unified Analytics System: Apache Spark, a free software to process large volumes of data, with libraries for different Big Data workloads. Though Spark itself is handy for tasks that require enormous computing power, its application in anomaly detection, both in user behavior and network traffic, will provide information security [25, 26].
- IBM Security QRadar: An enterprise security information and event management product that can collect and analyze log data generated by security systems, network devices, host assets, applications, vulnerabilities, and user activities [27].
- Splunk: it is the premier software platform for search, analysis, and visualization of machine data generated. Splunk allows indexing and real-time analysis of structured, unstructured, and complex logs to enable alerts, event notifications, and reports based on the state of machines [28].
- Apache Metron: An open-source Big Data for security monitoring and analytics that is free. It allows for real-time processing of emitted data; it also has a component for long-term storage. Metron provides a centralized monitoring portal for contextual alerts and attack information [29].

A. Decision Trees

Decision Trees were included because they provide interpretable rule-based classification and can model non-linear interactions among log-derived features [34, 37, 38]. In this study, the tree recursively partitions the feature space according

to the splits that best separate normal and attack traffic, producing decision paths that are easy to inspect. This makes the model attractive for hospital security teams that need understandable alerts, although depth control is required to limit overfitting.

B. K-Nearest Neighbors

K-Nearest Neighbors (KNN) was selected as a strong non-parametric baseline for attack prediction [36, 39]. The classifier assigns a label to each new instance according to the dominant class among its nearest neighbors in feature space, allowing it to capture local and non-linear attack patterns without imposing a fixed functional form. Because KNN is sensitive to feature scaling and the value of k [36], both normalization and hyperparameter tuning were applied before final evaluation. The basic KNN decision principle is illustrated in Fig. 1.

$$Distance(x, x_i) = \sqrt{\sum_{j=1}^{d} \left(x_j - x_{ij}\right)^2}$$

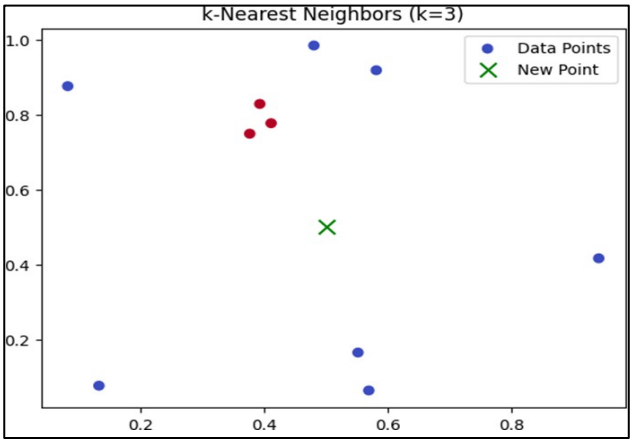

Fig. 1. Basic KNN decision principle.

C. Logistic Regression

Logistic Regression was used as a probabilistic linear baseline for binary intrusion detection [32]. The model estimates the probability that a request belongs to the attack class from a weighted combination of the input features and offers a transparent view of how features contribute to predictions. A conceptual illustration of this decision boundary is shown in Fig. 2.

Its interpretability and low computational cost make it useful in healthcare security analytics, even though purely linear decision boundaries may miss some complex attack relationships. This choice is also consistent with prior work showing that regularized logistic regression can support interpretable anomaly-oriented prediction while remaining suitable for imbalanced security-related settings [32].

$$P(X = x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_d x_d)}}$$

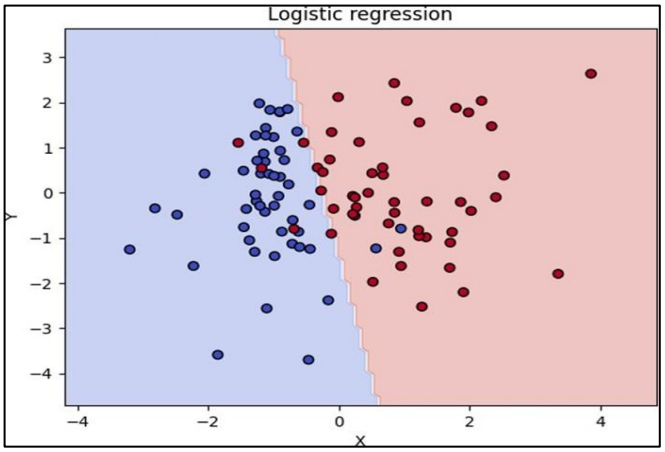

Fig. 2. Illustration of a Logistic Regression model applied to a synthetic binary classification dataset.

D. Comparison Summary

The three classifiers were selected to represent complementary design trade-offs relevant to hospital intrusion detection. Logistic Regression offers transparency and efficiency, Decision Trees provide interpretable non-linear rules, and KNN captures local attack structures with strong discrimination when sufficient labeled examples are available. Comparing them under the same preprocessing and validation protocol allows the study to assess not only accuracy, but also operational fit for large-scale security monitoring.

IV. Design of a Combined Machine Learning and Big Data System for Cyberattack Prediction: Use Case of a University Hospital Information System

The experimental dataset consisted of 1,000,000 labeled HTTP log instances extracted selectively from a university hospital web application environment to form a 30% attack and 70% normal requests, and exported to CSV after preprocessing. The corpus was constructed from raw Apache access logs and request traces, then anonymized to remove patient and user identifiers before feature generation. Labels were assigned as normal or attack using rule-based annotation supported by known malicious payload patterns, status-code anomalies, authentication abuse signatures, and manual verification of representative samples. The retained dataset characteristics are summarized in Table I, while Fig. 3 shows the original structure of the training log data.

TABLE I. Dataset Characteristics

Item	Description
Source	Raw HTTP Log files
Total instances	1,000,000
Class distribution	700,000 normals; 300,000 attack
Retained fields	Packet size, HTTP method, request/response type, URL, status code, parameter count, URL length
Privacy handling	IP addresses and user identifiers are anonymized before modeling

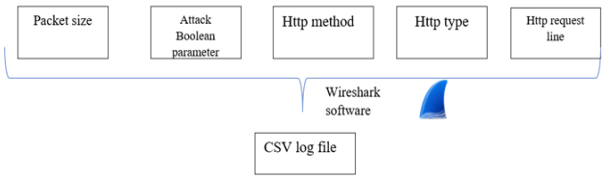

Fig. 3. Structure of the dataset used for training the prediction algorithms

A. Data Preparation and Feature Engineering

Data preparation focused on converting raw web-log events into a consistent feature space suitable for supervised learning. After removing duplicated or incomplete records, categorical fields were encoded, continuous attributes were normalized, and malformed requests were retained only when they contained security-relevant evidence. This step reduced noise while preserving the behavioral signals needed for attack prediction.

The final feature set included HTTP method, request/response type, packet size, request URL, status code, URL length, and parameter count. These features were chosen because they capture request structure and protocol behavior in a form that remains compatible with the information retained from the original server logs.

B. Predictive Modeling and Experimental Design

Three supervised classifiers were trained on the same labeled dataset: KNN, Logistic Regression, and Decision Trees. The data were partitioned with a stratified 80/20 train-test split so that the original normal/attack ratio was preserved in both subsets. All model selection steps were performed only on the training data. The overall workflow for model construction, application, and evaluation is illustrated in Fig. 4.

- Hyperparameter tuning was conducted using grid search with five-fold stratified cross-validation. For KNN, we searched k in {3, 5, 7, 9, 11}, weight schemes {uniform, distance}, and distance metrics {euclidean, manhattan}. For Decision Trees, we explored the split criterion {gini, entropy}, max depth {3, 5, 10, None}, and min samples split {2, 5, 10}. For Logistic Regression, we evaluated C in {0.01, 0.1, 1, 10} with L2 regularization using the liblinear and lbfgs solvers.
- Model performance was reported on the held-out test set using accuracy, precision, recall, F1-score, false positive rate, and processing time. Apache Spark was used to accelerate preprocessing and batch inference at larger scales, enabling a consistent comparison between traditional and distributed execution. The best configurations selected by cross-validation were KNN (k = 5, distance weighting, Euclidean distance), Decision Tree (gini, max_depth = 10, min_samples_split = 5), and Logistic Regression (C = 1, L2, liblinear). The selected hyperparameter values are summarized in Table II.

TABLE II. SELECTED HYPERPARAMETER VALUES

Model	Selected configuration
KNN	k = 5; distance weighting; Euclidean metric
Decision Tree	gini; max_depth = 10; min_samples_split = 5
Logistic Regression	C = 1; L2 penalty; liblinear solver

C. Results Analysis and Discussion

The evaluation examined both predictive quality and deployment-oriented efficiency. Final classification metrics were computed on a held-out test set of 200,000 instances, while processing-time measurements were recorded across dataset sizes from 100,000 to 1,000,000 instances to assess scalability under increasing log volume.

D. Classification Performance and Operational Metrics

KNN achieved the strongest overall results on the held-out test set, reaching 95.66% accuracy and substantially outperforming Logistic Regression (85.30%) and Decision Trees (83.20%). Its advantage was not limited to accuracy: KNN also produced the highest precision and F1-score, indicating that it balanced correct attack detection with reliable alert generation. The full metric breakdown is reported in Table III, and the cross-size accuracy trend is shown in Fig. 5.

From an operational perspective, the false positive rate is especially important for hospital environments because excessive false alarms can overwhelm security teams and reduce trust in automated monitoring. KNN yielded the lowest false positive rate (3.60%), whereas Logistic Regression and Decision Trees produced 8.00% and 9.00%, respectively. This result strengthens the case for KNN as the most deployment-ready classifier among the models evaluated.

TABLE III. CLASSIFICATION PERFORMANCE

Model	Accuracy	Precision	Recall	F1-score	FPR	Specificity
KNN	95.66%	91.79%	93.93 %	92.85 %	3.60 %	96.40%
Logistic Regression	85.30%	78.87%	69.67 %	73.98 %	8.00 %	92.00%
Decision Tree	83.20%	75.58%	65.00 %	69.89 %	9.00 %	91.00%

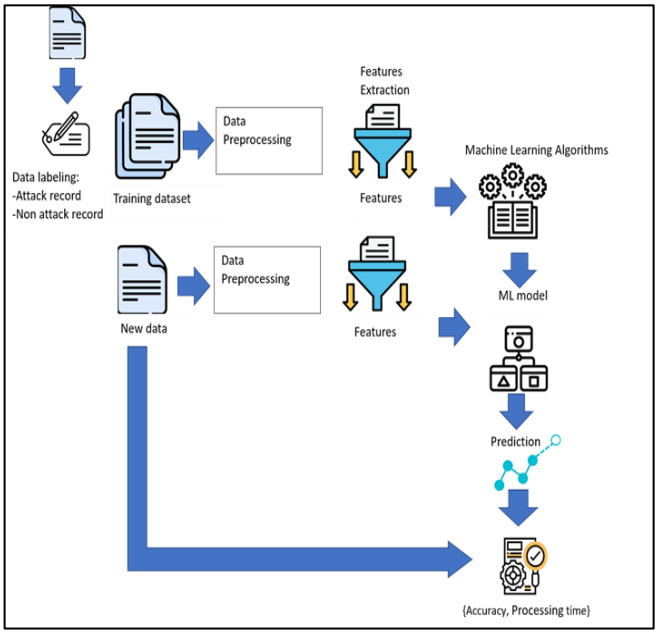

Fig. 4. Process of constructing, applying, and evaluating the predictive model for HTTP attacks prediction.

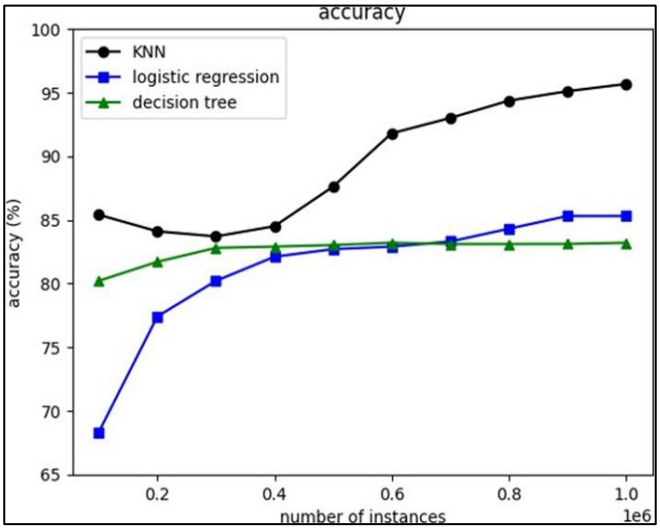

Fig. 5. Accuracy comparison of machine learning algorithms (KNN, logistic regression, and decision tree) for HTTP attack prediction across different dataset sizes.

E. Processing Time Analysis

Processing-time analysis confirmed the value of distributed execution for large-scale hospital log analysis. Across all three models, Spark reduced runtime as dataset size increased, with the strongest absolute gains observed for KNN and Decision Trees at the 1,000,000-instance scale. Although Logistic Regression remained the fastest classifier under Spark (6.0 s), KNN matched Decision Tree at 6.5 s while preserving a much stronger detection profile, making it the best accuracy-efficiency compromise in the study. Table IV summarizes the detailed timing values, and Fig. 6 highlights the corresponding processing-time trend.

TABLE IV. PROCESSING TIME COMPARISON (SECONDS)

Dataset size	Model	Without Spark (s)	With Spark (s)
100,000	KNN	1.8	1.1
100,000	Logistic Regression	1.4	1.2
100,000	Decision Tree	2.0	1.4
300,000	KNN	4.5	2.6
300,000	Logistic Regression	3.0	2.8
300,000	Decision Tree	4.7	2.9
500,000	KNN	6.8	3.7
500,000	Logistic Regression	4.2	3.9
500,000	Decision Tree	7.3	4.1
700,000	KNN	8.9	4.8
700,000	Logistic Regression	5.1	4.9
700,000	Decision Tree	9.6	5.0
1,000,000	KNN	12.0	6.5
1,000,000	Logistic Regression	6.5	6.0
1,000,000	Decision Tree	12.6	6.5

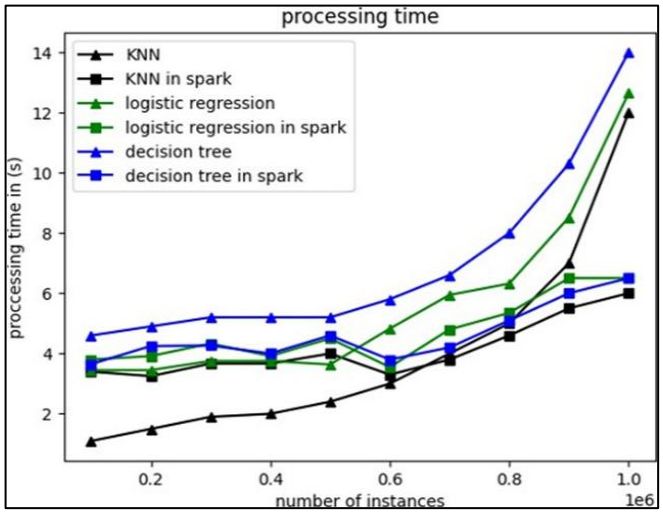

Fig. 6. Processing time comparison of KNN, logistic regression, and decision tree algorithms, both with traditional processing and using Spark.

V. CONCLUSION

This study presented a coherent framework for cyberattack prediction in e-Health information systems by combining structured log analysis, supervised machine learning, and Apache Spark-based distributed processing. Using a labeled corpus of 1,000,000 HTTP log instances from a university hospital context, the study demonstrated that rigorous preprocessing and feature engineering can support reliable detection of malicious web activity against healthcare services.

Among the evaluated classifiers, KNN delivered the best overall performance, achieving 95.66% accuracy, 91.79% precision, 93.93% recall, a 92.85% F1-score, and the lowest false positive rate (3.60%). These results are especially important for operational deployment because they indicate a lower alert burden on hospital security teams compared with Logistic Regression and Decision Trees.

The processing-time experiments further showed that Apache Spark materially improves scalability, cutting KNN runtime from 12.0 s to 6.5 s at 1,000,000 instances and enabling practical large-scale analysis without sacrificing detection quality. Taken together, the results suggest that KNN plus distributed execution offers the most favorable balance of accuracy, responsiveness, and operational usability for intrusion monitoring in e-Health environments.

Future work can extend this framework toward multi-class attack labeling, streaming analytics for continuous monitoring, and external validation on logs collected from additional hospital systems. These steps would further strengthen generalizability while preserving the deployment-oriented focus established in the present study.

REFERENCES

[1] Aceto G, Persico V, Pescapè A (2018) The role of information and communication technologies in healthcare: taxonomies, perspectives, and challenges. J Netw Comput Appl 107:125–154

[2] Husák M, Komárková J, Bou-Harb E, Čeleda P (2018) Survey of attack projection, prediction, and forecasting in cyber security. IEEE Commun Surv Tutor 21(1):640–660.

[3] OWASP (2024) OWASP top ten web application security risks - 2024. Available from: https://owasp.org/www-project-top-ten/

[4] Gzar DA, Mahmood AM, Abbas MK (2022) A comparative study of regression machine learning algorithms: tradeoff between accuracy and computational complexity. Math Model Eng Prob 9(5):1379–1384

[5] Abdalla RR, Jumaa AK (2022) Log file analysis based on machine learning: a survey. UHD J Sci Technol 6(2):77–84

[6] Hubballi N, Khandait P (2023) Event log analysis and correlation: a digital forensic perspective. In: Artificial Intelligence (AI) in Forensic Sciences, p 195

[7] Sittig DF, Wright A (2023) A guide to mitigating audit log-related risk in medical professional liability cases. J Healthc Risk Manag

[8] Prabha P, Kumar S, Shree N, Sundaram R, Nishanthi HM, Pranesh D (2024) Cybersecurity in healthcare: safeguarding patient data

[9] Wickramage C, Sahama T, Fidge CJ (2016) Anatomy of log files: implications for information accountability measures

[10] Landauer M, Wurzenberger M, Skopik F et al (2018) Dynamic log file analysis: an unsupervised cluster evolution approach for anomaly detection. Comput Secur 79:116

[11] Zhong M, Zhou Y, Chen G (2021) A security log analysis scheme using deep learning algorithm for IDSS in social network. Secur Commun Netw 2021:5542543

[12] Skopik F, Wurzenberger M, Landauer M (2022) Detecting unknown cyber security attacks through system behavior analysis. In: Cybersecurity of Digital Service Chains: Challenges, Methodologies, and Tools, Springer, Cham, p 103–119

[13] Azizi Y, Azizi M, Elboukhari M (2019) Log files analysis using MapReduce to improve security. Procedia Comput Sci 148:37–44

[14] Jeon D, Tak B (2022) Blackeye: automatic IP blacklisting using machine learning from security logs. Wireless Netw 28(2):937–948

[15] Khan N, Yaqoob I, Hashem IAT et al (2014) Big data: survey, technologies, opportunities, and challenges. Sci World J 2014:712826

[16] Bhadani AK, Jothimani D (2016) Big data: challenges, opportunities, and realities. Effective Big Data Manage Opp Implement 1–24

[17] Katal A, Wazid M, Goudar RH (2013) Big data: issues, challenges, tools and good practices. In: Sixth Int Conf Contemp Comput (IC3), IEEE, p 404–409

[18] Bello-Orgaz G, Jung JJ, Camacho D (2016) Social big data: recent achievements and new challenges. Inf Fusion 28:45–59

[19] Sagiroglu S, Sinanc D (2013) Big data: a review. In: Int Conf Collabor Technol Syst (CTS), IEEE, p 42–47

[20] Amanullah MA, Ariyaluran RAH, Nasaruddin FH et al (2020) Deep learning and big data technologies for IoT security. Comput Commun 151:495–517

[21] Kwan P, Ho CJ (2003) Multiple tiered network security system, method and apparatus

[22] Griffin M (2013) Assessment of run-time malware detection through critical function hooking and process introspection against real-world attacks. The University of Texas at San Antonio

[23] Rassam MA, Maarof M, Zainal A et al (2017) Big data analytics adoption for cybersecurity: a review of current solutions, requirements, challenges and trends. J Inf Assur Secur 12(4)

[24] Mahdavi P (2014) Google correlations: new approaches to collecting data for statistical network analysis. University of California, Los Angeles

[25] Arif Z, Zeebaree SRM (2024) Distributed systems for data-intensive computing in cloud environments: a review of big data analytics and data management. Indones J Comput Sci 13(2)

[26] Saeed MA, Saeed MA (2024) Real-time diabetes detection using machine learning and Apache Spark. In: 4th Int Conf Emerg Smart Technol Appl (eSmarTA), IEEE, p 1–6

[27] Gnatyuk S, Berdibayev R, Aleksander M et al (2024) Software system for cybersecurity events correlation and incident management in critical infrastructure. In: Data-Centric Business and Applications: Advancements in Information and Knowledge Management, Springer, Cham, p 247–269

[28] Razak AA, Ruzaili HH, Zolkifli MF (2024) Study on machine learning implementation in cybersecurity for security defend and attack. Borneo Int J 7(2):27–40

[29] Franzén MF, Tyrén N (2021) Anomaly detection for automated security log analysis: comparison of existing techniques and tools

[30] Langone R, Cuzzocrea A, Skantzos N (2020) Interpretable anomaly prediction: predicting anomalous behavior in Industry 4.0 settings via regularized logistic regression tools. Data Knowl Eng 130:101850

[31] Huang C (2018) Featured anomaly detection methods and applications. University of Exeter, UK

[32] Mahbooba B, Timilsina M, Sahal R, Serrano M (2021) Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model. Complexity 2021:6634811

[33] Liu C, Li B, Vorobeychik Y, Oprea A (2017) Robust linear regression against training data poisoning. In: Proc 10th ACM Workshop Artif Intell Secur, p 91–102

[34] Halder RK, Uddin MN, Uddin MA, Aryal S, Khraisat A (2024) Enhancing k-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. J Big Data 11(1):113

[35] Azam Z, Islam MM, Huda MN (2023) Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree. IEEE Access

[36] Lakshminarasimman S, Ruswin S, Sundarakantham K (2017) Detecting DDoS attacks using decision tree algorithm. In: 4th Int Conf Signal Process, Commun Netw (ICSCN),IEEE, p 1–6

[37] Yusof AR, Udzir NI, Selamat A (2016) An evaluation on KNN-SVM algorithm for detection and prediction of DDoS attack. In: Trends Appl Knowl-Based Syst Data Sci: 29th Int Conf Ind Eng Other Appl Appl Intell Syst (IEA/AIE 2016), Springer, Morioka, Japan, p 95–102

[38] Sannigrahi M, Thandeeswaran R (2024) Predictive analysis of network based attacks by hybrid machine learning algorithms utilizing Bayesian optimization, logistic regression and random forest algorithm. IEEE Access

[39] Chen JJ, Tsai C-A, Moon H, Ahn H, Young JJ, Chen C-H (2006) Decision threshold adjustment in class prediction. SAR QSAR Environ Res 17(3):337–352