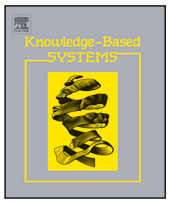
Knowledge-Based Systems 306 (2024) 112710

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Predictive modeling and anomaly detection in large-scale web portals through the CAWAL framework

Özkan Canay [a,b,*], Ümit Kocabıçak [c,d]

[a] Sakarya University of Applied Sciences, Vocational School of Sakarya, Dept. of Computer Tech., 54290, Sakarya, Turkiye
[b] Sakarya University, Institute of Natural Sciences, Dept. of Computer and IT Engineering, 54050, Sakarya, Turkiye
[c] Turkish Higher Education Quality Council, 06800, Ankara, Turkiye
[d] Sakarya University, Faculty of Computer and IT Engineering, Dept. of Computer Eng., 54050, Sakarya, Turkiye

ARTICLE INFO

Keywords:
Web usage mining (WUM)
CAWAL
User behavior prediction
Anomaly detection
Machine learning

ABSTRACT

This study presents an approach that uses session and page view data collected through the CAWAL framework, enriched through specialized processes, for advanced predictive modeling and anomaly detection in web usage mining (WUM) applications. Traditional WUM methods often rely on web server logs, which limit data diversity and quality. Integrating application logs with web analytics, the CAWAL framework creates comprehensive session and page view datasets, providing a more detailed view of user interactions and effectively addressing these limitations. This integration enhances data diversity and quality while eliminating the preprocessing stage required in conventional WUM, leading to greater process efficiency. The enriched datasets, created by cross-integrating session and page view data, were applied to advanced machine learning models, such as Gradient Boosting and Random Forest, which are known for their effectiveness in capturing complex patterns and modeling non-linear relationships. These models achieved over 92% accuracy in predicting user behavior and significantly improved anomaly detection capabilities. The results show that this approach offers detailed insights into user behavior and system performance metrics, making it a reliable solution for improving large-scale web portals' efficiency, reliability, and scalability.

1. Introduction

Web usage mining (WUM) is the process of analyzing user interactions on websites to extract meaningful and valuable insights from their behavior. Web logs play a critical role by providing essential data such as navigation paths, pages visited, and interaction durations, enabling the analysis of user behaviors on websites [1]. Accurate analysis of user interactions is crucial for strategic decision-making, especially in fields like e-commerce, online education, and security [2]. However, the growing volume of data and the increasing complexity of user interactions challenge the capacity of traditional methods to process large datasets efficiently. Hence, integrating machine learning and data mining techniques into WUM offers significant opportunities, particularly in predictive modeling and anomaly detection, while also introducing new challenges [3].

Data preprocessing, one of the most critical stages in WUM, involves cleaning and organizing weblogs to extract meaningful information. However, traditional data preprocessing methods are time-consuming and complex, especially for large datasets [4]. For example, the use of social network analysis and frequent pattern mining to discover valuable information from extensive web data was proposed [5], while fuzzy techniques and clustering were focused on to understand user behavior in large datasets [6,7]. Despite these advancements, the need for more comprehensive and automated data processing techniques is growing.

Predictive modeling, a widely used WUM application, is a crucial method for forecasting future user activities on websites. In recent years, a study successfully applied Long Short-Term Memory (LSTM) networks to predict e-commerce users' shopping intentions with high accuracy [8]. Similarly, another recent study achieved high success in web page prediction using a chicken swarm optimization model based on neural networks [9]. These models contribute to strategic decision-making by predicting users' shopping tendencies and browsing habits. However, the accuracy of such predictive models is directly linked to the scope and richness of the datasets used. Due to their limited data coverage, web server logs often constrain these models' performance.

* Corresponding author at: Sakarya University of Applied Sciences, Vocational School of Sakarya, Dept. of Computer Tech., 54290, Sakarya, Turkiye.
E-mail address: canay@subu.edu.tr (Ö. Canay).
URL: https://canay.subu.edu.tr/ (Ö. Canay).

https://doi.org/10.1016/j.knosys.2024.112710
Received 29 September 2024; Received in revised form 24 October 2024; Accepted 3 November 2024
Available online 9 November 2024

Another important application of WUM is anomaly detection, which focuses on identifying abnormal user behaviors and is commonly employed to detect security threats and performance issues. For example, a multi-behavior fusion-based security system has been shown to achieve high accuracy in detecting anomalies and misuse within computer networks [10]. In a similar vein, an automated analysis method utilizing NGINX logs has been developed to detect user anomalies in large web server logs, providing a valuable approach for enhancing cybersecurity in network analysis [11].

This study presents an approach that utilizes session and page view data collected through the CAWAL framework [12] for predictive modeling and anomaly detection in WUM process. CAWAL integrates detailed data such as session information, page view records, user profiles, and interaction history, eliminating the need for the preprocessing stage in traditional WUM and enhancing process efficiency, while also providing a robust and comprehensive dataset. In this study, session and page view data collected by CAWAL are cross-integrated using developed custom queries to create enriched datasets. These enriched datasets are then applied to advanced machine learning models, such as Gradient Boosting and Random Forest, which are known for their effectiveness in capturing complex patterns and modeling non-linear relationships. The results obtained from these models were subsequently analyzed to evaluate their performance.

The hypothesis (H1) that the user interaction data provided by the CAWAL framework will enhance the accuracy of machine learning-based prediction models in large-scale, multi-server architectures is tested in this study. CAWAL is expected to improve the performance of WUM processes by delivering more reliable results for predictive models. Additionally, the hypothesis (H2) that the framework will optimize anomaly detection in multi-server systems, thereby improving system efficiency and security, is also tested. These hypotheses are explored within this study's scope to evaluate the framework's performance in WUM processes.

The main contributions of this study are as follows:

1. The acceleration of WUM processes by eliminating the preprocessing step using CAWAL-provided data.
2. Improvement in predictive model accuracy by enriching CAWAL's session and page view data through advanced techniques.
3. Optimization of anomaly detection processes in multi-server and multi-domain architectures, such as web farms.
4. Provision of a more comprehensive data infrastructure for optimization and decision-making processes in web portals.

The remainder of this paper is structured as follows: Section 2 reviews existing approaches in web usage mining and discusses the innovations introduced by the CAWAL framework. Section 3 details the framework's architecture, data flow and processing steps, and preparations for machine learning. Section 4 analyzes the prediction models and anomaly detection using enriched data and examines the experimental results. Section 5 provides a thorough discussion of the findings, and finally, Section 6 offers a general evaluation of the study, conclusions, and suggestions for future research.

2. Related work

Web usage mining is the process of analyzing weblogs to extract meaningful patterns from users' online interactions. The data, such as users' browsing habits, pages visited, and session durations, form the core of WUM's information sources [13]. This process is supported by machine learning and data mining techniques to handle and analyze large datasets. Table 1 summarizes recent studies on web usage mining, focusing on prediction and anomaly detection, along with the methods and techniques used.

In recent years, predictive modeling and anomaly detection have emerged as two prominent application areas within WUM [14,22]. The main stages of web usage mining consist of data preprocessing, pattern discovery, and pattern analysis [23]. The first stage, data preprocessing, is essential for making raw web log data analyzable, involving tasks such as data cleaning, user identification, and session identification [24]. However, inaccuracies and errors in web log data can negatively impact the accuracy of analyses. For instance, incorrect session merging or user identification errors can lead to misleading results during modeling processes [25]. Therefore, careful execution of data cleaning and session management processes is critical to the success of WUM.

Predictive modeling is a strategic method used to forecast users' future behaviors. These models utilize large-scale data analysis and machine learning algorithms to predict users' browsing habits, interactions, or purchasing tendencies [26]. Such predictions provide valuable contributions, especially in dynamic fields like e-commerce and online services [27]. On the other hand, anomaly detection identifies activities that deviate from standard user behavior patterns, providing crucial feedback in terms of security and performance [28]. This section will examine recent developments in predictive modeling and anomaly detection within WUM, exploring studies and new approaches to enhance these processes' effectiveness.

2.1. Data preprocessing and session identification

Data preprocessing, as the initial and most crucial phase of the web usage mining process, is fundamental for systematically structuring large datasets to enable precise predictive modeling and effective anomaly detection. This phase includes key tasks such as data cleaning, user identification, and transforming raw log data into a format suitable for analysis [24]. The necessity of these tasks arises from the inherently noisy and unstructured nature of web log data, which, if left unprocessed, can impede effective pattern discovery and data mining efforts [29]. Reducing inconsistencies in the data set, filtering out irrelevant information and structuring sessions in the pre-processing phase significantly improve the accuracy of WUM results.

The complexity and time-consuming nature of preprocessing tasks make it one of the most resource-intensive stages in the WUM process. Research indicates that this phase consumes over 60% of the total time and resources allocated to WUM [30,31], with some studies showing that this figure may rise to as much as 80% [32,33]. Given its impact, preprocessing is established as a fundamental step in WUM, with various studies demonstrating its importance in enhancing the quality and reliability of data mining results [30,34].

Session identification, an essential step in data preprocessing, plays a critical role in accurately analyzing users' navigation behavior. A new method for identifying web user sessions was developed, successfully generating all possible maximal paths [35]. This approach enabled more accurate structuring of user sessions, leading to superior results in subsequent page predictions. This process plays a significant role in making predictive modeling more efficient. Similarly, the Online Web Navigation Assistant (OWNA) analyzes real-time data streams during session identification, providing recommendations to users [36]. This model optimizes user navigation behavior throughout sessions, improving the prediction of their actions on the web.

Tools used for data preprocessing in web usage mining also enhance the efficiency of processes. A hybrid approach has been developed that combines techniques like Ant Colony Optimization and Genetic Algorithm to improve classification accuracy during the data preprocessing stage [37]. Such tools filter errors and anomalies in large datasets, contributing to more successful prediction and anomaly detection outcomes.

Table 1
Recent studies on web usage mining, prediction, and anomaly detection methods.

Key contribution	Methods & techniques used	Year	Ref.
Predicts e-commerce users' shopping intentions using LSTM recurrent neural networks.	Prediction, LSTM Recurrent Neural Networks	2021	[8]
Provides an analysis of usage patterns and prediction through web usage mining techniques.	Pattern identification, Prediction, Clustering, Classification	2024	[3]
Extracts patterns from proxy logs and predicts website requests.	Prediction, Fuzzy data mining, Fuzzy frequent mining	2023	[14]
Proposes a neural network model for web page prediction using adaptive deer hunting and chicken swarm optimization.	Prediction, Apriori Algorithm, Chicken Swarm Optimization, Neural Network	2022	[9]
Develops a model for anomaly and intrusion detection using multi-demeanor fusion techniques.	Intrusion detection, Anomaly detection, Stochastic Latent Semantic Analysis	2021	[10]
Enhances next-page prediction performance using web graphs and session reconstruction techniques.	Session reconstruction, Prediction, Bayesian network, Complete Session Reconstruction	2023	[15]
Uses a context-aware cohesive Markov model and Apriori algorithm for web usage pattern discovery.	Prediction, Cohesive Markov Model, Apriori Algorithm	2022	[16]
Predicts user navigation patterns on websites using web usage mining techniques.	Prediction, Maximum frequent pattern, Classification	2021	[17]
Proposes a system to predict users' learning styles on e-learning platforms.	Prediction, Spectral Clustering, Quadratic Support Vector Machine (E-SVM)	2024	[18]
Analyzes user behavior on e-commerce platforms and develops recommendation systems based on the findings.	Prediction, Random Forest classification, Event listeners	2023	[19]
Utilizes web access logs for semantic clustering to improve web prefetching accuracy.	Web prefetching, Prediction, Semantic clustering, Thesaurus (WordNet), SPUDK	2024	[20]
Improves user session clustering and prediction using semantic-based web session clustering methods.	Session clustering, Prediction, K-Means, Hierarchical Agglomerative Clust., Semantic distance	2022	[21]

2.2. Predictive modeling techniques

Predictive modeling is one of the critical techniques used to anticipate users' next steps and predict potential actions on the web. One study demonstrated that the Compact Prediction Tree algorithm offers higher accuracy in predicting web pages than traditional methods such as k-nearest neighbor (k-NN) and decision trees [38]. Similarly, recent research has highlighted the effectiveness of hybrid machine learning methods like Random Forest and Gradient Boosting in predicting web page transitions by utilizing both static and dynamic page features [39]. Algorithms like these, employed to predict users' following pages, stand out as some of the most potent approaches in web usage mining. Another study developed a web session reconstruction algorithm using a dynamic link repository [15]. In this approach, web sessions were modeled graphically, and Bayesian networks were used to predict the next page, providing a dynamic prediction mechanism to optimize user movements across the web.

Another practical approach for predicting user behavior is the use of fuzzy logic-based algorithms. Using fuzzy data mining, one study analyzed proxy server log files to predict users' subsequent web requests [14]. By analyzing users' browsing frequency and behaviors, fuzzy association rules were created, enabling the accurate prediction of their next steps. Similarly, picture fuzzy logic was applied in a multi-criteria decision-making framework to evaluate website performance [40]. In another approach, fuzzy association rules were extracted from web data using learning automata, where trapezoidal membership functions (TMF) were used to optimize the time users spent on web pages, resulting in improved prediction accuracy [41]. When classical machine learning methods fall short, these approaches offer a more flexible and adaptive prediction mechanism.

Clustering and classification techniques are also widely used in the predictive modeling process. One study employed hybrid methods combining classification techniques such as Random Forest and genetic algorithms to improve prediction accuracy [42]. These hybrid approaches, used in the context of WUM, allow for a more accurate classification of web log data. Similarly, another study used the fuzzy C-means algorithm to cluster user behaviors, making predictions based on these clusters [43]. Following this line of research, an approach applied K-means and hierarchical clustering algorithms to group web sessions, extracting meaningful patterns from session data and predicting future user actions [21]. Clustering algorithms are particularly effective in grouping user behaviors in large datasets and predicting future trends based on these groups.

2.3. Anomaly detection approaches

One key aspect of web usage mining is the detection of anomalous user behaviors. Such anomalies can stem from various sources, including security threats, unusual user activities, or incorrect data inputs. Techniques that combine WUM with anomaly detection not only optimize user behavior analysis but also contribute to enhancing security measures. A modified hybrid method, combining PSO, GA, and K-Means, was developed for anomaly and misuse detection in computer networks [44]. This model detects abnormal behaviors in network traffic, allowing for minimizing security vulnerabilities.

Big data analytics plays a crucial role in anomaly detection within web mining. The IRPDP_HT2 algorithm, developed as a scalable data preprocessing method based on Hadoop MapReduce, enables faster and more efficient detection of anomalies in large datasets [45]. Analyzing large-scale data sets for critical tasks such as robot detection proves to be an effective method for identifying abnormal activities. Similarly, dimensionality reduction techniques have been employed to detect anomalies in large datasets, providing scalable solutions for managing large volumes of web data [46]. These approaches provide scalable solutions for managing large volumes of web data and detecting anomalies.

Hybrid methods are another approach to anomaly detection. A hybrid method combining the Grey Wolf Algorithm and CNN was developed to detect anomalous behavior in network data streams [47]. Hybrid methods offer more flexible and efficient solutions for anomaly detection by integrating machine learning techniques with traditional approaches such as data compression [48].

3. Methodology

The CAWAL model [12] was developed based on traditional application logging practices but expands this approach by integrating web analytics features. While conventional web analytics tools focus

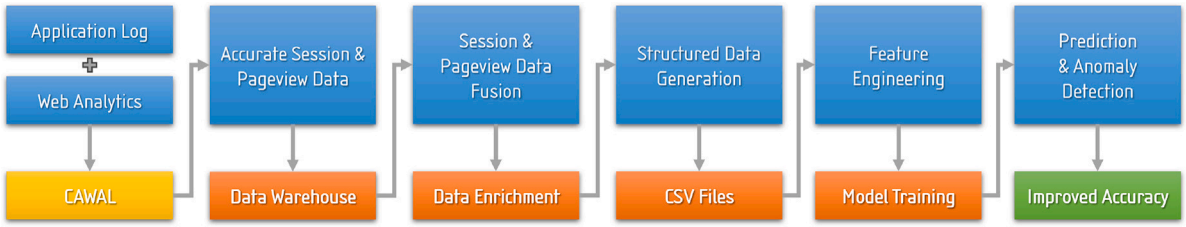


Fig. 1. Workflow of data enrichment and prediction process for web usage mining.

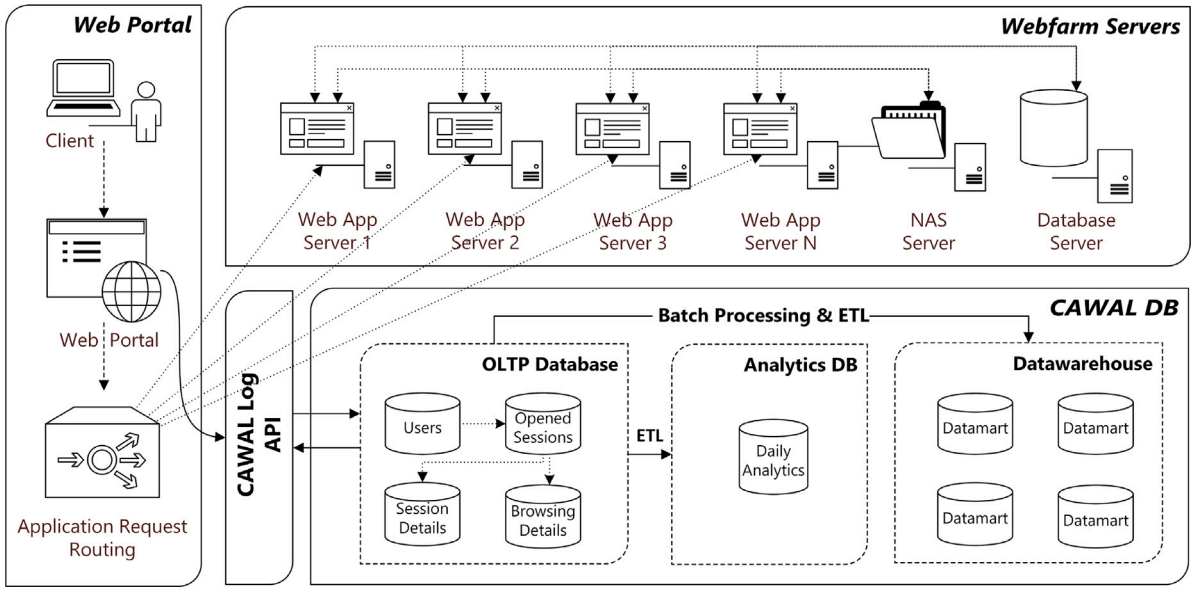


Fig. 2. CAWAL framework architecture and integration with web portal infrastructure.

on capturing user interactions, modern tools have shifted toward more comprehensive data collection [49]. The CAWAL model differentiates itself from other analytical tools by incorporating application logs and enhancing them with comprehensive log data and detailed user interaction analyses. This model was implemented on Sakarya University's institutional web application, the "Campus Automation Web Information System" (CAWIS) [50], where long-term access data was collected. The CAWIS system is architected as a web portal, utilizing separate subdomains for each service, which allows for tracking user interactions across different services and enhances the coverage and accuracy of the data gathered by the CAWAL model.

Compliance with Sakarya University's Internet Services Usage Policy Agreement was ensured during the data collection process. All necessary permissions were obtained, and anonymization methods were applied to protect user privacy. Timestamp data was altered to anonymize users' activities over time further. This adjustment is not expected to negatively affect the analysis results, as the study focuses on trends and behavioral patterns during specific periods rather than absolute timestamps. Every research stage was conducted meticulously to maintain participant privacy and ensure data security.

This study presents an approach utilizing session and page view data collected through the CAWAL framework for prediction and anomaly detection in the field of web usage mining. The impact of enriching these datasets, stored within the CAWAL data warehouse, and integrating them into machine learning models on the accuracy of web usage predictions is examined. Fig. 1 illustrates the data enrichment and prediction workflow applied in WUM, beginning with the session and page view data obtained through the CAWAL framework. The subsequent steps for processing and analysis are outlined, demonstrating the effective use of this data for precise prediction and anomaly detection.

3.1. Integration of the CAWAL framework

The CAWAL framework, designed to integrate with the web portal, detects user information and in-app events that third-party tracking tools fail to capture, storing the data in a structured format within a relational database [12]. A data collection API continuously monitors exceptions, user flows, and state changes while also enabling the inclusion of application-specific data, such as form field entries in the tracking logs [49]. Complete session tracking is maintained through persistent monitoring of the application servers. The CAWAL data collection API is initiated at the start of the web portal's code execution and integrates seamlessly to activate automatically with each page request. By integrating the API with the portal, the complexity of logging is abstracted from software processes, allowing developers to focus on core functions without being burdened by log management. The overall architecture and integration details of the CAWAL framework with the web portal infrastructure are illustrated in Fig. 2.

While the back-end code of the portal runs, the CAWAL data collection API operates within a multi-layered framework, gathering data in the background with each request. At the end of the portal's general interface template code, details such as page load times, database query delays, and error and warning messages are updated in the page view table via the API. This code-level tracking capability provided by CAWAL offers insights into applications, servers, and connections that would be otherwise unobtainable through traditional methods. This systematic approach enables the collection of comprehensive and unique data, helping to keep applications and systems under consistent monitoring.

The deployment of the framework in a real-world corporate web portal, encompassing a web farm with ten web servers and various

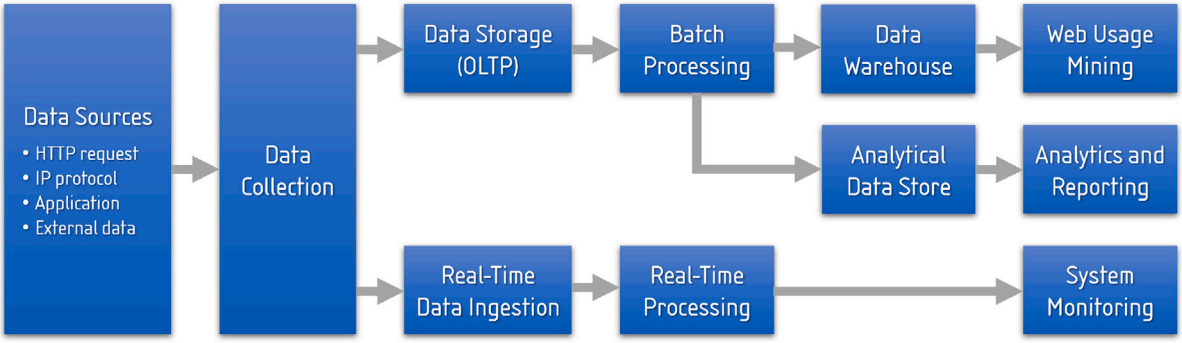


Fig. 3. Data flow and processing in the CAWAL model.

web services spread across multiple subdomains, provides a distinctive approach to managing and analyzing extensive web traffic. CAWAL works harmoniously with a load-balancing mechanism configured to track operations across different servers simultaneously, ensuring system performance is maintained even during peak user activity periods. Using a shared NAS server to direct the session and configuration folders of the servers in the web farm provides an extra layer of consistency to the CAWAL deployment. This centralized storage solution guarantees continuity across the web farm, offering uniform and structured session data management. The architecture's support from load balancing and NAS servers enhances scalability, uninterrupted service delivery, and flexible solutions for complex web applications, thus improving overall system efficiency and its ability to adapt to the demands of the applications [12].

3.2. Data flow of the CAWAL model

Operational data generated during routine transactions in web applications is stored in write-intensive OLTP databases, which handle continuous data input and output operations [49]. CAWAL implements a streamlined data model optimized for efficient analytics while minimizing storage overhead. Fig. 3 presents a schematic detailing the data flow and processing steps in the CAWAL model.

The "Data Sources" section of the schematic illustrating the data flow and processing in the CAWAL model highlights various data sources, such as HTTP requests and network protocols. These data sources provide crucial information for monitoring user interactions on the portal in detail. For example, HTTP requests reveal which pages users visit and how long they stay on them, while network protocols provide system-level metrics such as server performance and the processing time for user requests. During the data collection phase, user and usage data is fused and prepared for processing, then stored in data storage systems. Temporal data gathered through the CAWAL framework is stored in a relational database in a homogeneous structure. These data can be used in real-time for system monitoring, as well as for anomaly detection and various predictions through machine learning algorithms.

The data stored in OLTP databases is processed during the batch processing stage for analytical insights and is later transferred to the data warehouse through ETL processes [12]. The data housed in the warehouse then serves as a critical resource for future WUM analyses and predictive models. This approach minimizes the preprocessing required for WUM, ensuring the data is clean, consistent, and immediately ready for pattern discovery. The data collected through the CAWAL framework facilitates swift and accurate results in web usage mining processes, thanks to its high data accuracy and reliability.

3.3. Data preparation and enrichment

Traditional web server logs typically provide limited information, focusing primarily on page visits and clickstreams. However, more detailed and specific data, such as time spent on a page or session login status, can only be captured at the application level and through disparate data sources. The CAWAL framework addresses this limitation by integrating web analytics with application logs to generate broader datasets that are difficult to achieve with conventional methods. These comprehensive session and page view datasets, which include critical information such as the services accessed, the paths followed by users, and the time spent on each page, enable a more detailed tracking of user interactions within the web portal. In addition to these fundamental details, comprehensive session data captures a wide range of user interactions throughout the session, extending beyond simple start and end times to provide a comprehensive view of user behavior. This data includes metrics such as the number of pages visited, average time spent per page, details of services accessed, and page load times, all of which expand the scope of session data, thus enabling more profound analysis.

Structured views are used to transform the raw data collected in the CAWAL data warehouse into a format suitable for analysis. These views employ complex SQL queries to extract, sessionize, and aggregate session and page view data across specific date ranges, enriching the information with demographic and behavioral attributes. All steps, including data extraction, sessionization, and enrichment, are accomplished through SQL-based procedures. The views provide a structured format that supports comprehensive analysis, capturing metrics like session duration, login frequency, service transitions, and usage details, alongside enriched page view data such as browser type, IP location, and service usage duration. This structured approach enhances the utility of the data for web usage mining activities, enabling detailed and precise analyses.

The enriched page view data includes basic metrics as well as additional user attributes and session details. The analysis provides a comprehensive view of user behavior by linking each page view to its corresponding session, identifying accessed services, and measuring service usage duration. Additionally, attributes such as browser type, IP location, and user type allow for targeted performance evaluations across diverse user segments. Table 2 presents the fields and sample data from these enriched page view datasets, offering a clearer understanding.

Enriched session and page view datasets encompass numerical and categorical fields and provide significant advantages for researchers and developers seeking to analyze user behavior and portal performance deeply. The data serves as a rich resource for critical analyses, such as understanding how users interact with services during sessions, how they transition between services, and how these transitions impact user engagement. This comprehensive and enriched analytical data can be leveraged to maximize the effectiveness of WUM activities, improve the accuracy of analyses, and yield more precise results.

Table 2
CSV file format containing enriched page view data.

Field name	Description	Sample data
Detail_ID	Page view detail ID.	89 010 871
Session_ID	User's session ID.	83 665 107
Detail_Date_Time	Request timestamp	11.20.2022 13:01
User_ID	User ID.	184 922
Current_Login_Status	Login status at the time of request.	1
Session_Login_Status	Login status at the session.	1
User_Type	Type of portal user.	6
Sex	User's sex.	2
Age	User's age.	18
Age_Group	User's age group.	1
User_Language_TR	User's browser language.	1
User_Location	User's IP location.	1
Browser_Type	User's browser type.	1
Referer_Type	Referrer type of the request.	6
Server_ID	Requested server ID.	4
Service_ID	Requested service ID.	3
Page_Duration	Page dwell time (s).	41
Page_Load_Time	Page load (generation) time (s).	0.122

Each field in the dataset facilitates detailed monitoring and analysis of user interactions. The session ID and user ID enable tracking user movements across the portal and evaluating behavioral changes over time. Metrics such as page load time and time spent on a page directly influence the performance of the web portal and provide critical insights for improving the user experience. These metrics also play a crucial role in analyzing behavioral differences among specific user groups. Demographic data such as user type, gender, and age group allow for the segmentation of user behavior, enabling more targeted and customized analyses.

3.4. Performance evaluation

The time complexity of the SQL queries used for processing indexed, enriched session, and page view data has been carefully evaluated, considering the efficient utilization of data structures and optimization of query design. For enriched session data, the queries typically join session records with user demographics and behavioral information. This process involves operations such as LEFT JOIN and WHERE clauses that filter by date ranges and session IDs. With proper indexing, the time complexity of these queries generally remains at $O(n \log n)$. However, in large datasets and complex joins, the theoretical complexity has the potential to reach $O(n^2)$.

Similarly, for enriched page view data, the SQL queries link page view records with detailed user and service information. Given the presence of multiple entries per page and the need to aggregate high volumes of page view data, the complexity of the queries is influenced accordingly. With optimized indexing, these queries typically maintain a complexity of $O(n \log n)$, although as the data volume scales, the complexity can theoretically approach $O(n^2)$. In both cases, the use of indexed fields and optimization techniques significantly enhances the performance of the data enrichment processes, ensuring efficient and scalable data handling.

3.5. Generation of data files

The data collected by the CAWAL framework underwent data selection and enrichment procedures using complex SQL queries to prepare it for the web usage mining process. Approximately 8.5 GB of session and page view data, spanning one month, were structured and enriched within data warehouses through SQL-based view formats. These views facilitated the efficient extraction, sessionization, and enrichment of the raw data. The enriched data was subsequently exported as comma-separated CSV files, ensuring compatibility with Python-based analysis workflows. All steps, including data transformation and CSV generation, were executed through SQL queries, optimizing the process for effective data management. The average query execution times and CSV write durations were recorded in seconds for each time interval specified in Table 3, demonstrating that the indexed queries and CSV generation times performed well relative to the record count, ensuring efficient data processing and performance monitoring.

These structured datasets are essential for analyzing users' interactions with the portal's various services. The daily page view dataset contains 787,637 records, while the monthly session dataset consists of 1,220,916 records. Enriching these datasets is crucial for a detailed analysis of user behavior, enabling the identification of varying needs across different user segments and allowing for targeted service optimizations. Metrics such as page load time provide valuable insights into performance indicators directly influencing user engagement. These large datasets, which are vital for examining user behaviors over specific periods and analyzing usage patterns during special events, can be used to identify performance bottlenecks and develop optimization strategies. Storing the data in CSV format allows researchers to process it quickly and efficiently, enhancing accessibility for web usage mining and other analytical applications.

3.6. Feature engineering and model training

The success of data mining processes relies heavily on effectively processing raw data and transforming it into meaningful features. This process involved comprehensive tasks such as cleaning session data, selecting relevant features, and preparing these features for modeling. Fig. 4 conceptually illustrates how usage data is transformed through feature engineering into prediction models.

During the feature engineering phase, attributes that accurately reflect the complexity of user interactions were identified, and these features were used to train prediction models to reveal patterns and trends in the data. The specific features selected for the models are detailed in the analysis section of the study. In the modeling phase, a Random Forest Classifier was employed to predict users' likelihood of abandoning the system, chosen for its robustness in handling complex interactions within large datasets [51]. Similarly, a Gradient Boosting Classifier, known for its effectiveness in capturing non-linear relationships, was used to predict the last service accessed before abandonment [52]. These models were trained and tested on enriched session and page view datasets.

Four different models were applied to predict the probability of a user accessing a specific service: Gradient Boosting, Random Forest, Support Vector Machine (SVM), and Logistic Regression. Each model was selected for particular strengths in handling the dataset characteristics. Gradient Boosting was chosen for its accuracy in modeling non-linear relationships, Random Forest for its ability to manage high-dimensional data without overfitting, SVM with an RBF kernel for its capability in handling non-linear separations, and Logistic Regression for its simplicity and interpretability, making it a suitable benchmark model for comparison [53,54]. The performance of these models was enhanced through hyperparameter optimization, ensuring the best parameters were selected for each model.

The Isolation Forest algorithm was employed for anomaly detection based on server and page load times due to its efficiency in isolating anomalies without requiring prior information on anomaly ratios [55]. This model was selected over alternatives such as One-Class SVM and Local Outlier Factor because of its computational efficiency and suitability for high-dimensional, large-scale datasets. The feature pool was structured to include extensive attributes, such as enriched session and page data, user demographic information, browser and device types, and system performance metrics. This centralized repository played a crucial role in the process of cleaning, selecting, and preparing data for analysis, ensuring consistent and reliable results during the modeling phase.

Table 3
Enriched session and page view data stored in CSV files and their properties.

Time frame	Time range	File name .CSV	Number of records	File size (MB)	Query time (s)	CSV write (s)
1-day	2022-11-22 00.00 - 23.59	va_page2	787,637	66.3	42.8	2.9
1-week	2022-11-21 - 2022-11-27	va_sess4	514,879	99.2	35.7	1.8
1-month	2022-11-01 - 2022-11-30	va_sess5	1,220,916	235.0	66.2	4.1

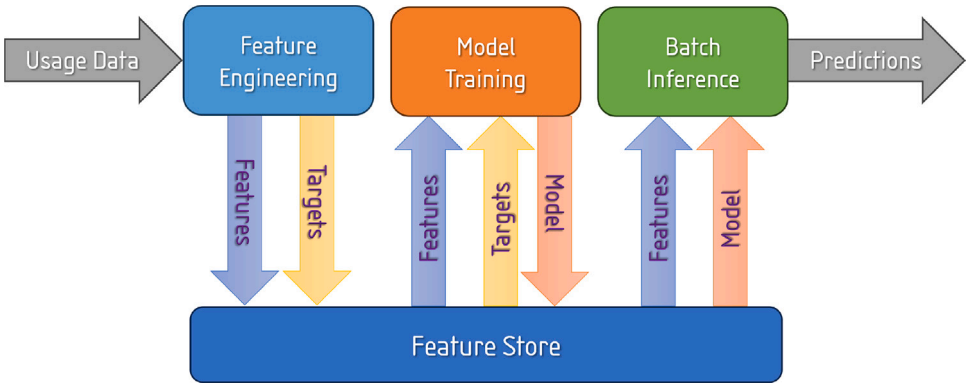


Fig. 4. Transformation of CAWAL web usage data into predictions.

Table 4
Class-wise and overall performance metrics of the multiple classification model.

Class	Precision	Recall	F1 score	Support
0 (direct leave)	0.96	0.91	0.93	163,187
1 (logout button)	0.92	0.96	0.94	192,866
2 (notification window)	0.77	0.86	0.81	10,222
Weighted avg.	0.93	0.93	0.93	366,275

4. Predictive analysis and findings

Four analyses were conducted within the scope of this research using enriched session and page view data obtained through the CAWAL framework, including three aimed at prediction and one at anomaly detection. The first three analyses, aimed at enhancing the efficiency of the web portal, focus on predicting users' exit methods, the last services they abandon, and their access to a specific portal service. On the other hand, the analysis for anomaly detection focuses on identifying abnormalities based on page load times on web farm servers, providing essential insights into the security and performance of the portal. The findings obtained from these analyses provide a strong foundation for strategies to enhance the portal's effectiveness by enabling a better understanding of user behavior.

4.1. Prediction of methods for leaving the system

This analysis, which focuses on predicting how users will leave the system, is hugely significant in understanding interactions and behavioral patterns. The prediction model used in the study was trained using a Random Forest Classifier [56] with enriched session data. The training utilized several user and session-related attributes, including User_Type, Sex, Age, User_Language_TR, User_Location, Browser_Type, Landing_Srv_ID, Exit_Srv_ID, Session_Login_Status, Page_Count, Service_Count, Total_Session_Duration, Avg_Page_Duration, Total_Page_Load, p_gate, p_mail, p_obis, p_abis, p_pbis, and p_menu.

A one-month dataset of 1,220,916 enriched session data rows was split into 70% for training and 30% for testing, yielding highly successful prediction results. Table 4 displays the precision, recall, and F1 score values for each class, as well as the support counts. The weighted average represents the model's overall performance, calculated based on all classes' proportions.

When the model's performance is examined to identify individual classes and overall predictions, it demonstrates remarkable accuracy. This finding reflects the model's ability to distinguish specific classes and its overall predictive success across the dataset. The high precision and good recall values obtained for direct departures (Class 0) indicate the model's effectiveness in identifying this class. Similarly, for secure exits via the exit button (Class 1), the high precision and even higher recall values demonstrate the model's strong performance in predicting this class.

The F1 scores for both classes indicate that the model performs a balanced performance in predicting these classes. On the other hand, the precision and recall values for Class 2 (timeout notification window) are lower than other classes. However, the F1 score obtained for this class suggests the model's performance in predicting this class is acceptable. It can be inferred that the relatively minor number of examples for this class leads to slightly lower performance results compared to the other classes.

The evaluation of the model's performance, mainly through the weighted average F1 score, demonstrates that it achieves balanced and high accuracy across all classes. This effectiveness is further supported by the enriched data provided by CAWAL, which enhances the model's predictive capabilities for different exit methods. These findings highlight the model's robustness in handling diverse user behaviors and its potential application for improving system design and user experience.

4.2. Prediction of the last abandoned service

This part of the study focuses on predicting the service through which users will leave the system. For this purpose, the model is trained with a gradient boosting classifier [57] using various user and session features from a comprehensive dataset containing one-week session information. In the model's training, 0.1 was chosen as the learning rate, three as the maximum depth, and 100 as the number of predictors. The features used in the model and their importance on the predictive ability of the model are presented in detail in Fig. 5.

The details of the features used in the model and their impact on the model's predictive ability highlight the depth and comprehensiveness of these analyses. Notably, the features p_obis (0.4033), p_mail (0.1304), and p_gate (0.0969) stand out as the most influential factors that significantly enhance the model's prediction accuracy. These p_ prefixed features represent the number of pages users visit within specific services. These findings indicate that users' interactions with the

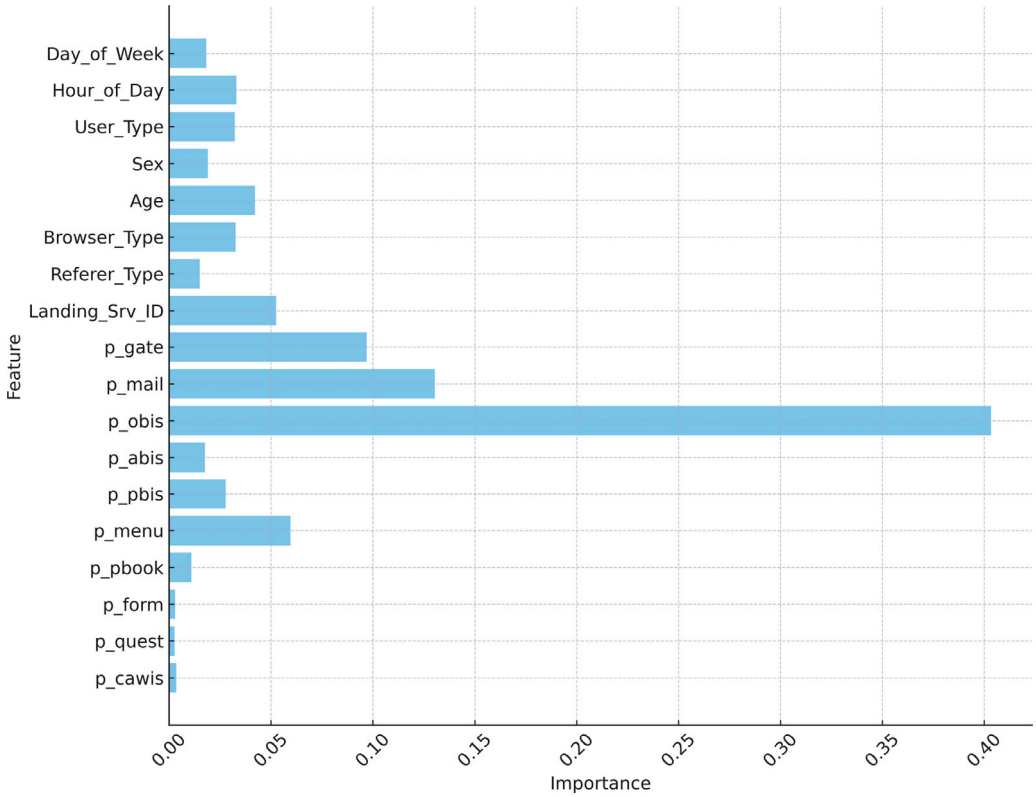


Fig. 5. Feature importance scores in the predictive model.

"obis", "mail", and "gate" services play a crucial role in determining their exit points from the web portal. The high importance of these features dramatically increases the predictability of users' interactions with the system, providing valuable insights for optimizing the web portal's performance and enhancing the user experience. According to the model's test results, the accuracy is measured at 0.9557, precision at 0.9561, recall at 0.9557, and the F1 score at 0.9555. These results demonstrate that the model is highly effective in predicting through which service users will exit the system. These findings prove that using the data provided by the CAWAL framework, it is possible to predict with high accuracy the last service through which users will leave the portal.

4.3. Prediction of access to a specific service

WUM methods are crucial for gaining deeper insights into portal interactions. Prediction models can be used to determine whether a user will access a particular portal service in a session. Various session information such as Log_Date_Time, Log_Date, User_Type, Sex, Age, Avg_Page_Duration, User_Language_TR, User_Location, Browser_Type, and Referer_Type was trained with four different models to perform service access prediction. Hyperparameter optimization was performed to achieve the best performance, and the optimal parameters of the four models were determined. Table 5 presents the performance of the classification models and their parametric configurations in a comparative manner.

This representation reveals the models' differences by demonstrating the best parameter combinations and performance metrics, such as average cross-validation (CV) score, accuracy, precision, recall, and F1 score side-by-side. When the model performance metrics are analyzed, it is seen that all four models exhibit high accuracy rates and balanced F1 scores. The fact that the Random Forest and Gradient Boosting models stand out in terms of both accuracy and F1 score indicates that these two models have a better generalization capability on the dataset. The combination of values identified as the best parameters of the Random Forest model shows that the model manages its complexity and learning ability in a balanced way.

The Logistic Regression model [58] achieved a high Average Cross-Validation Score with the best C parameter but had a slightly lower F1 score, indicating potential difficulty in discriminating between certain classes. In contrast, the Support Vector Machine (SVM) model [59] demonstrated high generalization capability with its RBF kernel and C parameter. Both models offer a balanced approach to the classification problem. The Gradient Boosting model's performance improved by carefully tuning the learning rate, maximum depth, and number of predictors, successfully capturing the dataset's complex structures with high accuracy and F1 score. While all models show high accuracy, Gradient Boosting and Random Forest particularly excel in capturing complex patterns. Despite strong cross-validation performance, the Logistic Regression model may slightly struggle with class separability, suggesting that tree-based models might better fit scenarios where class distinction is critical.

The four models applied have produced effective results for this classification problem and have accurately predicted whether a user will access a service based on specific features. The fact that each model's parameter settings were adjusted to provide the best results suited to the system's data structure demonstrates the significant role of hyperparameter optimization. This optimization has enhanced each model's performance, leading to more precise and reliable predictions. Specifically, Gradient Boosting and Random Forest models have effectively utilized the selected session-specific features to capture the complex interactions in the dataset, achieving high accuracy and balanced F1 scores. These results indicate that the detailed information provided by CAWAL, including user demographics, behavioral patterns, and contextual factors, has significantly improved prediction performance. Beyond the primary interaction data provided by traditional web server logs, these enriched features have enabled the models to understand the complex aspects of user behavior better and make more accurate predictions.

Table 5
Performance comparison of various classification models for predicting mail service usage.

Metric/Model	Gradient boosting	Random forest	Support vector Mach.	Logistic regression
Best parameters	Learning Rate: 0.1, Max Depth: 3, N Estimators: 100	Max Depth: 10, Min Samp. Leaf: 2, Min Samp. Split: 5, N Estimators: 100	C: 10, Kernel: 'rbf'	C: 10
Average CV score	0.9182	0.8947	0.9184	0.9143
Accuracy	0.9252	0.9240	0.9238	0.9178
Precision	0.9234	0.9218	0.9214	0.9156
Recall	0.9252	0.9240	0.9238	0.9178
F1 score	0.9188	0.9176	0.9174	0.9094

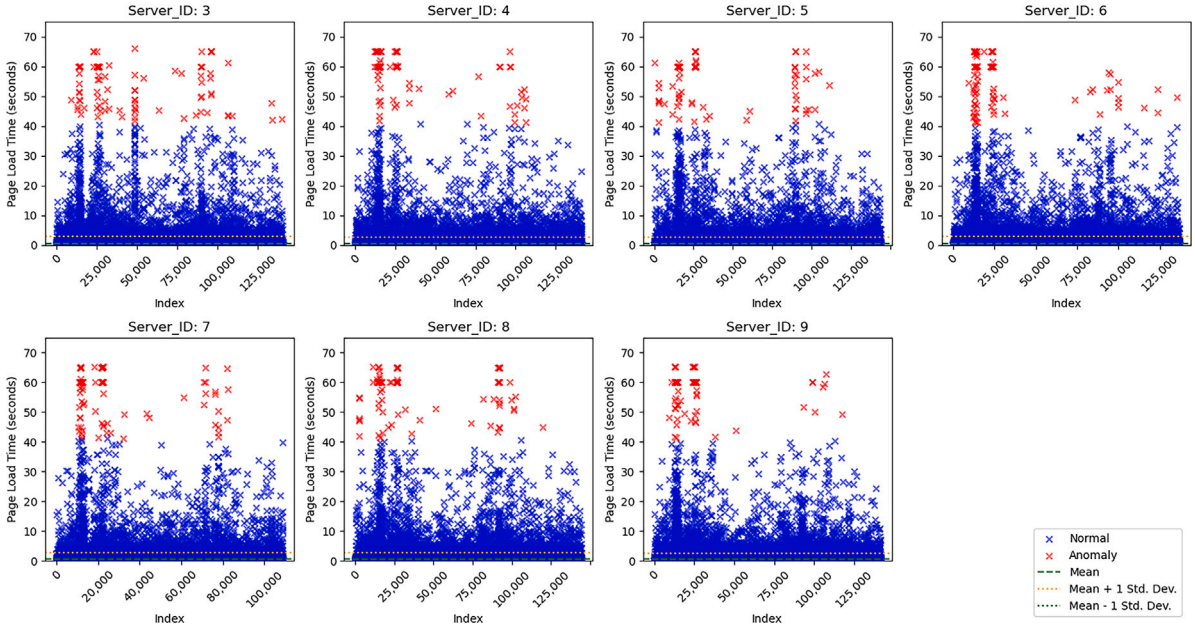


Fig. 6. Visualization of anomalies based on page load times across seven servers in the web farm.

4.4. Server and page load time-based anomaly detection

In a web farm architecture, the systematic analysis of server performance and page load times is crucial for detecting anomalies within applications and servers. These issues can arise from software bugs, hardware malfunctions, network latency, interactions with databases or other connected systems, denial-of-service (DoS) attacks, or server configuration problems. In this analysis, anomaly detection was performed using the Isolation Forest (iForest) algorithm, an effective method for identifying outliers in a dataset based on isolation principles, as demonstrated in a recent study [60]. The Isolation Forest works by recursively partitioning the data space through multiple binary trees, known as isolation trees or iTrees. The main idea behind this method is that anomalies are relatively sparse and distinct, which makes them easier to isolate. To achieve this, the mathematical techniques used in the anomaly detection process are explained in detail below.

One such technique is the Isolation Forest algorithm, which detects anomalies by isolating data points using random partitioning. The isolation process begins by randomly selecting a feature and then selecting a split value between the minimum and maximum values of that feature. This process recursively repeats, creating partitions that progressively isolate individual data points. The key metric in Isolation Forest is the *path length* $h(x)$, defined as the number of edges traversed from the root node to the terminating node for a specific data point x in an isolation tree. Anomalous data points tend to have shorter path lengths because they are easier to isolate compared to normal points.

For the dataset with n instances, the average path length $c(n)$ of unsuccessful searches in a Binary Search Tree can be approximated by:

$$c(n) = 2 \cdot H(n-1) - \frac{2(n-1)}{n}$$

where $H(n-1)$ is the harmonic number, estimated as:

$$H(n-1) = \ln(n-1) + \gamma$$

and γ is the Euler–Mascheroni constant ($\gamma \approx 0.5772$).

The anomaly score for a data point x is calculated based on its average path length $h(x)$ across all isolation trees in the forest. The score $s(x,n)$ is given by:

$$s(x,n) = 2^{-\frac{h(x)}{c(n)}}$$

Specifically, $s(x,n)$ ranges from 0 to 1, where points with a score close to 1 are considered anomalies due to their shorter path lengths. Isolation Forest is advantageous because of its linear time complexity with a low constant and its ability to handle high-dimensional data efficiently. Moreover, it does not require prior knowledge of the anomaly ratio within the dataset, making it a versatile tool for various applications.

Considering these advantages, we applied Isolation Forest to analyze the 24-h page view data and load times of requests distributed through load balancing in a multi-server architecture. This approach enabled us to detect anomalies in web traffic patterns, which could indicate issues such as server overloads, network delays, or potential security threats. The results of the analysis performed with the model trained using the Isolation Forest algorithm are presented in Fig. 6.

The visualization illustrates anomalies detected based on page load times across seven diverse servers. Each graph shows the distribution

of page load times, which refers to the time the back-end code takes to generate the page over time, and highlights anomalies detected for the web server identified by its Server_ID. The "Y" axis of the subgraphs represents the page load times, while the "X" axis represents the "Index" value, corresponding to the number of page views observed over time on each server. In the graphs, regular observations are marked by blue 'x' symbols, while anomalies, where the load time is significantly longer than expected, are indicated by red markers.

Data analysis from a weekday with heavy system usage reveals several requests with high page load times. While the number of anomalies varies across the servers, the distribution appears generally balanced. Anomalies were detected when page load times deviated more than one standard deviation from the average. An exceptionally high number of anomalies observed on servers 3 and 6 suggests that these servers experienced more performance issues than others, especially in their interactions with related components such as other web servers, database servers, LDAP, and mail servers. These problems may have led to noticeable delays in page load times and even page timeout occurrences.

The delays in connections to heavily loaded servers and disruptions in queries performed on other application databases are among the primary causes of the anomalies. Performance issues in interactions with other components, especially on specific servers, cause significant delays and timeouts in page load times. The data provided by the CAWAL framework plays a critical role in detecting and analyzing these issues, contributing significantly to the overall evaluation of system performance.

5. Discussion

The CAWAL framework has successfully overcome the limitations of conventional Web usage mining methods by analyzing user behaviors in web portals. Traditional approaches rely solely on server logs, resulting in superficial and limited analysis of user interactions. Due to the limited data sources, these constraints make it difficult to understand user behavior. For instance, one study [32] attempted to model user behavior by preprocessing web data, but the diversity of the data remained restricted. Similarly, a new method for constructing user sessions was proposed [35], but it also faced the limitations of data richness from weblogs.

The CAWAL framework, developed as a web analytics solution, combines application logs with web analytics to provide a more comprehensive and robust dataset for large-scale web portals. This innovative approach offers significant advantages in multidimensional data integration and richness compared to weblogs, widely used as a data source in the literature for WUM. The framework enhances data quality by addressing the limitations of relying solely on weblogs. Improved data quality and diversity enable detailed and precise analysis of user interactions, resulting in impressive outcomes in analyses, where predictive models achieved over 92% accuracy and server-based anomaly detection yielded significant findings.

In this study, predicting the last service accessed before users exit the system was successfully achieved using the Gradient Boosting algorithm, with an accuracy of 95.61% and an F1 score of 95.55%. Detailed user and session data significantly enhanced the model's capacity to capture complex behavioral patterns. In a similar study, LSTM networks were employed to predict e-commerce users' shopping intentions [8], but the generalization capacity was limited due to insufficient data integration. The comprehensive data integration offered by CAWAL addresses this gap in the literature by improving the accuracy of predictive models. These results enable strategic decisions to enhance critical services in the system by accurately predicting the points at which users exit the portal.

The prediction of users' exit methods was effectively achieved using the Random Forest model, which demonstrated a weighted average F1 score of 93%, accurately forecasting different exit methods. The richness of the data achieved through the integration and processing of CAWAL-collected session and page view data enhanced the model's capacity for accurate predictions. A similar approach aimed to automatically extract users' browsing patterns [38], but the accuracy achieved did not reach the levels provided by CAWAL. These predictions can be considered a strategic tool for system design and user experience optimization. The obtained results demonstrate that CAWAL's depth and accuracy in analyzing user behavior contribute significantly to improving the performance of web portals.

The success of models predicting users' access to specific services was also remarkable, thanks to CAWAL's rich data sources. Comparing model results revealed that hyperparameter tuning was critical in accurately capturing the complex structures within the dataset. The Gradient Boosting model, with carefully tuned parameters such as learning rate, maximum depth, and the number of predictors, captured complex patterns with 91.88% accuracy and a 91.76% F1 score. The Random Forest model also performed well, with an accuracy of 92.40% and a 91.76% F1 score. Although the Logistic Regression model delivered effective results with a high cross-validation score, it performed relatively lower in the F1 score, indicating its limitations in distinguishing certain classes. The SVM model demonstrated balanced success with an RBF kernel and C parameter. These findings suggest that model selection and parameter tuning are crucial for obtaining results in classification problems, particularly with complex datasets. A study applying machine learning approaches to predict learning styles in e-learning platforms [18] achieved lower accuracy due to limited data diversity.

The high predictive success achieved across these three analyses strongly supports the initial hypothesis. The enriched datasets, created using the raw data collected through the CAWAL framework, have improved the accuracy of machine learning-based predictive models in large-scale architectures. The models' high accuracy and F1 scores demonstrate how the detailed session and page view data collected and integrated by the CAWAL framework offer the necessary depth and variety, enabling the accurate capture and modeling of complex user behaviors. While previous studies used various methods to determine user access behaviors on the web [14,61,62], these approaches were limited in terms of data diversity and richness because they mainly relied on weblogs. The data collected and processed through the CAWAL framework provides deeper analysis capabilities than traditional WUM methods, overcoming these limitations and enabling higher accuracy rates. Analyzing details such as session ID and page load times enabled more accurate predictions of user behavior, resulting in significant improvements in the system. These results indicate that CAWAL is highly effective in performance optimization and enhancing user engagement in high-traffic web portals.

The successful outcomes of anomaly detection further confirm the second hypothesis. The CAWAL framework enhances anomaly detection in web farms and multi-server architectures, helping to detect operational disruptions early and maintain system stability. The anomalies detected during the analysis could be due to factors such as insufficient server resources, software errors, or high traffic volumes. For example, the delays observed in email and database servers are thought to be caused by resource bottlenecks or excessive demand. Such analyses are valuable for improving system efficiency and developing proactive solutions for future needs. While a previous study examined anomaly detection in networks using WUM techniques [10], CAWAL's success in large-scale, multi-server environments provides a broader scope. The enriched datasets, developed from CAWAL-collected data, accelerate anomaly detection, allowing potential issues to be identified earlier.

However, the CAWAL framework does have some limitations. One limitation is that the datasets used in the study focus on a specific period and user group, which may limit the generalizability of the findings. The framework's effectiveness has not been thoroughly tested in environments demonstrating diverse user behaviors, such as various

industries, e-commerce, and mobile platforms. This situation highlights the need for further investigation into how CAWAL handles broad data diversity and models various user behaviors.

Additionally, although the CAWAL framework can process large datasets, aspects of the data processing pipeline require optimization in terms of time and computational costs. Specifically, multi-server systems' data collection and analysis processes demand substantial computational power and time, which may pose challenges for real-time applications. Furthermore, the performance of the machine learning models heavily depends on the scope and quality of the datasets. The effectiveness of these models may decrease with limited or imbalanced datasets.

Moreover, in environments where comprehensive and detailed user information is tracked, such as with CAWAL, stricter measures should be taken to ensure the security and privacy of the data, considering the associated risks. Handling such sensitive data requires adopting robust encryption methods and compliance with data protection regulations, such as GDPR [63], to prevent data breaches and unauthorized access, particularly in multi-server systems where vulnerabilities may increase.

6. Conclusion

The machine learning models utilizing enriched session and page view data collected through the CAWAL framework have shown superior performance in predicting user behaviors and detecting anomalies. By integrating data from multiple sources, including web analytics and application logs, the framework enables precise and in-depth analysis of user interactions, particularly in large-scale, multi-server architectures. This provides an effective solution for organizations that find traditional web server logs insufficient or prefer not to share user access data.

The CAWAL framework's secure and structured data collection mechanism serves as a valuable resource for web usage mining and machine learning applications, even in high-traffic systems. The diversity and integration capabilities of the framework enhance both the accuracy of predictive models and the efficiency of anomaly detection processes. The accelerated preprocessing stage achieved through CAWAL-collected data further strengthens the performance of these models.

The results confirm the framework's effectiveness in improving system performance and security, especially in web farms and multi-server environments. Anomaly detection analyses demonstrate that CAWAL enables early detection of issues, reducing operational risks and enhancing overall system efficiency. These findings validate the CAWAL framework as a practical and reliable approach for optimizing performance and security in complex, large-scale web portals.

Future research should focus on extensive testing of the model across different industries and larger datasets. These tests, aimed at increasing CAWAL's generalizability and applicability to diverse user groups, will further strengthen its flexibility and versatility, demonstrating its effectiveness across various data environments and architectural structures. Particularly in the e-commerce, finance, and healthcare sectors, the concrete effects of the framework's data integration, model accuracy, and system optimization in large-scale, multi-server systems should be evaluated. Such studies will validate CAWAL's broad application potential and contribute to identifying new approaches to predictive modeling and anomaly detection within the scope of web usage mining.

CRediT authorship contribution statement

Özkan Canay: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ümit Kocabıçak:** Supervision, Conceptualization.

Consent statement

Consent for data usage was secured through the Internet Services Usage Policy Agreement, which all portal users approved.

Ethics statement

The data utilized in this research were collected from the CAWIS web portal following legal statutes and Sakarya University's regulations. Necessary permissions were secured from the institution, and various data anonymization techniques were applied throughout the study to ensure user privacy and data security.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used DeepL, ChatGPT, and Grammarly in order to English translation and editing. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

[1] N. Hopipah, N. Jaman, N. Enri, Web usage mining guna analisis pola akses pengunjung website dengan association rule, SATIN 2 (7) (2021) 53–63, http://dx.doi.org/10.33372/stn.v7i2.735.

[2] B. Kumar, E-Commerce website usability analysis using the association rule mining and machine learning algorithm, Math. 11 (1) (2022-12) 25, http://dx.doi.org/10.3390/math11010025, Crossref.

[3] S. Dubey, G. Tiwari, P. Narwaria, Server access pattern analysis based on weblogs classification methods, Lect. Notes Electr. Eng. 1116 (2024) 183 0–195 0, http://dx.doi.org/10.1007/978-981-99-8646-0_16.

[4] E. Alshdaifat, D. Al-Shdaifat, A. Alsarhan, F. Hussein, S. El-Salhi, The effect of preprocessing techniques, applied to numeric features, on classification algorithms' performance, Data 6 (2) (2021) 11, http://dx.doi.org/10.3390/data6020011.

[5] C. Leung, E. Madill, S. Singh, A web intelligence solution to support recommendations from the web, in: ACM International Conference Proceeding Series, 2021, pp. 160 0–167 0, http://dx.doi.org/10.1145/3498851.3498966.

[6] H. Gangadwala, R. Gulati, Analysis of web usage mining using various fuzzy techniques and cluster validity index, in: 2022 1st International Conference on Electrical, Electronics, Information and Communication Technologies, ICEEICT 2022, 2022, pp. 1–7, http://dx.doi.org/10.1109/ICEEICT53079.2022.9768580.

[7] C. Za'in, M. Pratama, E. Lughofer, S.G. Anavatti, Evolving type-2 web news mining, Appl. Soft Comput. 54 (2017) 200–220, http://dx.doi.org/10.1016/j.asoc.2016.11.034.

[8] K. Diamantaras, M. Salampasis, A. Katsalis, K. Christantonis, Predicting shopping intent of e-commerce users using LSTM recurrent neural networks, in: Proceedings of the 10th International Conference on Data Science, Technology and Applications, DATA 2021, 2021, pp. 252 0–259 0, http://dx.doi.org/10.5220/0010554102520259.

[9] R. Gangurde, Web page prediction using adaptive deer hunting with chicken swarm optimization based neural network model, Int. J. Model. Simul. Sci. Comput. 13 (6) (2022) 2250064, http://dx.doi.org/10.1142/S1793962322500647.

[10] A. Gupta, J. Agrawal, The multi-demeanor fusion based robust intrusion detection system for anomaly and misuse detection in computer networks, J. Ambient Intell. Humaniz. Comput. 12 (1) (2021) 303–319, http://dx.doi.org/10.1007/s12652-020-01974-4.

[11] L. Benova, L. Hudec, Using web server logs to identify and comprehend anomalous user activity, in: 2023 17th International Conference on Telecommunications, ConTEL, IEEE, 2023-07, pp. 1–8, http://dx.doi.org/10.1109/ConTEL58387.2023.10199092.

[12] O. Canay, U. Kocabicak, CAWAL: A novel unified analytics framework for enterprise web applications and multi-server environments, Inf. Process. Manage. 61 (3) (2024) http://dx.doi.org/10.1016/j.ipm.2023.103617.
[13] N. Yau, W. Zainon, Understanding web traffic activities using web mining techniques, Int. J. Eng. Technol. Manag. Res. 4 (9) (2020) 18–26, http://dx.doi.org/10.29121/ijetmr.v4.i9.2017.96.
[14] H. Gangadwala, R. Gulati, Prediction and analysis of next website request by using fuzzy approach, in: Proceedings of the 2023 1st International Conference on Advances in Electrical, Electronics and Computational Intelligence, ICAEECI 2023, IEEE, 2023, pp. 1–6, http://dx.doi.org/10.1109/ICAEECI58247.2023.10370912.
[15] J. Jors, E. Luca, Predictive behavior modeling through web graphs: Enhancing next page prediction using dynamic link repository, in: Proceedings of the 2023 22nd IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT, Vol. 2023, 2023, pp. 415–420, http://dx.doi.org/10.1109/WI-IAT59888.2023.00068.
[16] V. Luckose, J. Chembath, J. Ponnusamy, S. Sharma, P. Kaur, S. Smiley, Web usage pattern detection using cohesive Markov model with apriori algorithm, in: Proceedings of the 2022 IEEE International Conference on Automatic Control and Intelligent Systems, 2022, pp. 226–229, http://dx.doi.org/10.1109/I2CACIS54679.2022.9815465.
[17] P. Om, S. Ananthakumaran, M. Sathishkumar, R. Ganeshan, Analyzing the user navigation pattern from web logs using maximum frequent pattern approach, in: Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT, 2021, pp. 877–883, http://dx.doi.org/10.1109/ICICT50816.2021.9358751.
[18] K. Prashanth Kumar, B. Harish Kumar, A. Bhuvanesh, Spectral clustering algorithm based web mining and quadratic support vector machine for learning style prediction in e-learning platform, Meas. Sens. 31 (2024) 100962, http://dx.doi.org/10.1016/j.measen.2023.100962.
[19] P. Rajapaksha, D. Asanka, Recommendations to increase the customer interaction of e-commerce applications with web usage mining, in: Proceedings of IEEE InC4 2023 - 2023 IEEE International Conference on Contemporary Computing and Communications, Vol. 1, 2023, pp. 1–6, http://dx.doi.org/10.1109/InC457730.2023.10263131.
[20] S. Setia, D. Jyoti, N.A. Anand, N. Verma, Semantically enriched keyword prefetching based on usage and domain knowledge, J. Web Eng. 23 (3) (2024) 341–376, http://dx.doi.org/10.13052/jwe1540-9589.2332.
[21] H. Sowmya, R. Anandhi, Semantic based weighted web session clustering using adapted K-means and hierarchical agglomerative algorithms, J. Web Eng. 21 (2) (2022) 239–263, http://dx.doi.org/10.13052/jwe1540-9589.2125.
[22] B. Marcin, R. Tomasz, Advanced examination of user behavior recognition via log dataset analysis of web applications using data mining techniques, Electronics (2023) http://dx.doi.org/10.3390/electronics12214408.
[23] K. Suguna, K. Nandhini, Frequent pattern mining of web log files working principles, Int. J. Comput. Appl. 157 (3) (2017) 1–5, http://dx.doi.org/10.5120/ijca2017912642.
[24] S. Panwar, Analysis of web server log file using hadoop, Int. J. Res. Appl. Sci. Eng. Technol. 6 (4) (2018) 1059–1062, http://dx.doi.org/10.22214/ijraset.2018.4178.
[25] D. Sisodia, R. Singhal, V. Kandal, Comparative performance of interestingness measures to identify redundant and non-informative rules from web usage data, Int. J. Technol. 9 (1) (2018) 201, http://dx.doi.org/10.14716/ijtech.v9i1.1510.
[26] L. Choudhary, S. Swami, Exploring the landscape of web data mining: an in-depth research analysis, Curr. J. Appl. Sci. Technol. 42 (24) (2023) 32–42, http://dx.doi.org/10.9734/cjast/2023/v42i244179.
[27] M. Ashraf, S. Ouf, Y. Helmy, A proposed paradigm for enhancing customer retention using web usage mining, Int. J. Comput. Appl. 177 (29) (2020) 32–35, http://dx.doi.org/10.5120/ijca2020919772.
[28] R. Ilieva, M. Ivanova, T. Peycheva, Y. Nikolov, Modelling in support of decision making in business intelligence, Adv. Bus. Inf. Syst. Anal. (2021) 115–144, http://dx.doi.org/10.4018/978-1-7998-5781-5.ch006.
[29] P. Nithya, P. Sumathi, Novel pre-processing technique for web log mining by removing global noise, cookies and web robots, Int. J. Comput. Appl. 53 (17) (2012) 1–6, http://dx.doi.org/10.5120/8510-1684.
[30] M. Srivastava, A. Srivastava, R. Garg, Data preprocessing techniques in web usage mining: a literature review, SSRN Electron. J. (2019) http://dx.doi.org/10.2139/ssrn.3352357.
[31] S.P. Singh, Analysis of web site using web log expert tool based on web data mining, in: 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS, IEEE, 2017, pp. 1–5, http://dx.doi.org/10.1109/ICIIECS.2017.8275961.
[32] N. Ali, A. Gadallah, H. Hefny, B. Novikov, An integrated framework for web data preprocessing towards modeling user behavior, in: Proceedings of the 2020 International Multi-Conference on Industrial Engineering and Modern Technologies, 2020, pp. 1–8, http://dx.doi.org/10.1109/FarEastCon50210.2020.9271467.
[33] P. Verma, N. Kesswani, Comparitive analysis of algorithms for identification of session on the basis of threshhold value, in: 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom, IEEE, 2016, pp. 3724–3730.
[34] A. Alcalde-Barros, D. García-Gil, S. García, F. Herrera, Dpasf: a flink library for streaming data preprocessing, Big Data Anal. 4 (1) (2019) http://dx.doi.org/10.1186/s41044-019-0041-8.
[35] M. Bayir, I. Toroslu, Maximal paths recipe for constructing web user sessions, World Wide Web 25 (6) (2022) 2455–2485, http://dx.doi.org/10.1007/s11280-022-01024-3.
[36] N. Ali, A. Gadallah, H. Hefny, B. Novikov, Online web navigation assistant, Vestnik Udmurtskogo Univ. Matematika, Mekhanika, Komp'yuternye Nauki 31 (1) (2021) 116 0–131 0, http://dx.doi.org/10.35634/VM210109.
[37] V. Malik, R. Mittal, J. Singh, V. Rattan, A. Mittal, Feature selection optimization using ACO to improve the classification performance of web log data, in: 2021 8th International Conference on Signal Processing and Integrated Networks, SPIN, IEEE, 2021-08, pp. 671–675, http://dx.doi.org/10.1109/SPIN52536.2021.9566126.
[38] K. Mani, K. Suneetha, Performance evaluation of compact prediction tree algorithm for web page prediction, in: International Conference on Emerging Trends in Information Technology and Engineering, Ic-ETITE 2020, 2020, pp. 1–7, http://dx.doi.org/10.1109/ic-ETITE47903.2020.166.
[39] T.K.N. Dang, D. Bucur, B. Atil, G. Pitel, F. Ruis, H. Kadkhodaei, N. Litvak, Look back, look around: A systematic analysis of effective predictors for new outlinks in focused web crawling, Knowl.-Based Syst. 260 (2023) 110126, http://dx.doi.org/10.1016/j.knosys.2022.110126.
[40] K. Kara, G.C. Yalcin, E.G. Kaygisiz, V. Simic, A.S. Ornek, D. Pamucar, A picture fuzzy CIMAS-ARTASI model for website performance analysis in human resource management, Appl. Soft Comput. 162 (2024) 111826, http://dx.doi.org/10.1016/j.asoc.2024.111826.
[41] Z. Anari, A. Hatamlou, B. Anari, Finding suitable membership functions for mining fuzzy association rules in web data using learning automata, Int. J. Pattern Recognit. Artif. Intell. 35 (07) (2021) 2159026, http://dx.doi.org/10.1142/S0218001421590266.
[42] V. Malik, R. Mittal, J. Singh, V. Rattan, A hybrid approach to improve classification performance using WMOT tool, in: 2021 International Conference on Emerging Smart Computing and Informatics, ESCI, IEEE, 2021-03, pp. 688–691, http://dx.doi.org/10.1109/ESCI50559.2021.9396872.
[43] J. Serin, J. SatheeshKumar, T. Amudha, Efficient fuzzy C-means based reduced feature set association rule mining approach for predicting the user behavioral pattern in web usage mining, J. Internet Technol. 23 (7) (2022) 1495–1503, http://dx.doi.org/10.53106/160792642022122307005.
[44] Y. Yuan, Y. Li, A modified hybrid method based on pso, ga, and k-means for network anomaly detection, Math. Probl. Eng. (2022) 1–10, http://dx.doi.org/10.1155/2022/5985426.
[45] X. Zhang, P. Wei, Q. Wang, A hybrid anomaly detection method for high dimensional data, PeerJ Comput. Sci. 9 (2023) 1199, http://dx.doi.org/10.7717/peerj-cs.1199.
[46] H. Liu, K. Li, X. Li, Y. Zhang, Unsupervised anomaly detection with self-training and knowledge distillation, in: 2022 IEEE International Conference on Image Processing, ICIP, 2022, pp. 2102–2106, http://dx.doi.org/10.1109/icip46576.2022.9897777.
[47] L. Wang, Q. Chen, C. Song, Anomaly detection model of network dataflow based on an improved grey wolf algorithm and cnn, Electronics 12 (18) (2023) 3787, http://dx.doi.org/10.3390/electronics12183787.
[48] A. Prasanth, M. Hemalatha, Intelligent web information retrieval based on user navigational patterns, Int. J. Comput. Appl. 109 (5) (2015) 26–32, http://dx.doi.org/10.5120/19186-0673.
[49] O. Canay, U. Kocabicak, An innovative data collection method to eliminate the preprocessing phase in web usage mining, Eng. Sci. Technol. Int. J. 40 (2023) http://dx.doi.org/10.1016/j.jestch.2023.101360.
[50] O. Canay, S. Meric, H. Evirgen, M. Varan, Realization of campus automation web information system in context of service unity architecture, in: International Symposium on Computing in Science & Engineering, ISCSE, Izmir, Turkey, 2011, pp. 173–179.
[51] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324.
[52] J.H. Friedman, Greedy function approximation: A gradient boosting machine, Ann. Statist. 29 (5) (2001) 1189–1232, http://dx.doi.org/10.1214/aos/1013203451.
[53] D.W. Hosmer, S. Lemeshow, Applied Logistic Regression, third ed., John Wiley & Sons, 2013, http://dx.doi.org/10.1002/9781118548387.
[54] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297, http://dx.doi.org/10.1007/BF00994018.
[55] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in: Proceedings of the 2008 IEEE International Conference on Data Mining, IEEE, 2008, pp. 413–422, http://dx.doi.org/10.1109/ICDM.2008.17.
[56] Y. Manzali, M. Elfar, Random forest pruning techniques: a recent review, Oper. Res. Forum 4 (2) (2023-05) 43, http://dx.doi.org/10.1007/s43069-023-00223-6.
[57] K. Omari, Phishing detection using gradient boosting classifier, Procedia Comput. Sci. 230 (2023) 120–127, http://dx.doi.org/10.1016/j.procs.2023.12.067.
[58] Y. He, K. Shen, H. Zhang, W. Duan, Z. Gong, R. Jia, H. Wang, A study based on logistic regression algorithm to teaching indicators, in: International Artificial Intelligence Conference, Springer Nature Singapore, Singapore, 2023-11, pp. 219–227, http://dx.doi.org/10.1007/978-981-97-1280-9_17.

[59] Y. Guo, W. Zhan, W. Li, Application of support vector machine algorithm incorporating slime mould algorithm strategy in ancient glass classification, Appl. Sci. 13 (6) (2023) 3718, http://dx.doi.org/10.3390/app13063718.

[60] T. Al-Shehari, M. Al-Razgan, T. Alfakih, R. Alsowail, S. Pandiaraj, Insider threat detection model using anomaly-based isolation forest algorithm, IEEE Access (2023) http://dx.doi.org/10.1109/ACCESS.2023.3326750.

[61] F. Alhaidari, S. Alwarthan, A. Alamoudi, User preference based weighted page ranking algorithm, in: ICCAIS 2020 - 3rd International Conference on Computer Applications and Information Security, 2020, pp. 1–6, http://dx.doi.org/10.1109/ICCAIS48893.2020.9096823.

[62] B. Soewito, J. Johan, Website personalization using association rules mining, in: Innovative Technologies in Intelligent Systems and Industrial Applications, Springer Nature Switzerland, 2023, pp. 689–698, http://dx.doi.org/10.1007/978-3-031-29078-7_60.

[63] C. Negri-Ribalta, M. Lombard-Platet, C. Salinesi, Understanding the GDPR from a requirements engineering perspective—a systematic mapping study on regulatory data protection requirements, Requir. Eng. (2024) 1–27, http://dx.doi.org/10.1007/s00766-024-00423-4.