Information Processing and Management 61 (2024) 103617

Contents lists available at ScienceDirect

Information Processing and Management

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

CAWAL: A novel unified analytics framework for enterprise web applications and multi-server environments ☆

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

[a] Sakarya University, Institute of Natural Sciences, Dept. of Computer and IT Engineering, Serdivan, 54050, Sakarya, Turkey
[b] Sakarya University of Applied Sciences, Vocational School of Sakarya, Dept. of Computer Tech., Adapazari, 54290, Sakarya, Turkey
[c] Sakarya University, Faculty of Computer and IT Engineering, Dept. of Computer Eng., Serdivan, 54050, Sakarya, Turkey

ARTICLE INFO

Keywords:
Web analytics
Application logging
Logging framework
User tracking
Data governance

ABSTRACT

In web analytics, cloud-based solutions have limitations in data ownership and privacy, whereas client-side user tracking tools face challenges such as data accuracy and a lack of server-side metrics. This paper presents the Combined Analytics and Web Application Log (CAWAL) framework as an alternative model and an on-premises framework, offering web analytics with application logging integration. CAWAL enables precise data collection and cross-domain tracking in web farms while complying with data ownership and privacy regulations. The framework also improves software diagnostics and troubleshooting by incorporating application-specific data into analytical processes. Integrated into an enterprise-grade web application, CAWAL has demonstrated superior performance, achieving approximately 24% and 85% lower response times compared to Open Web Analytics (OWA) and Matomo, respectively. The empirical evaluation demonstrates that the framework eliminates certain limitations in existing tools and provides a robust data infrastructure for enhanced web analytics.

1. Introduction

Web analytics tools enable detailed user behavior analysis, tracking user paths, time spent on individual pages, and interactions with site elements, allowing organizations to optimize their content and design (Pilz et al., 2020). With personalization gaining importance, businesses can now segment audiences based on various parameters such as demographics, geographical location, behavior, and devices used (Vamsee et al., 2023). Furthermore, machine learning and artificial intelligence integration have enabled predictive analytics, helping to anticipate future user behavior and strategize more effectively. Early web analytics implementations predominantly relied on web server logs to capture incoming HTTP requests (Kousik et al., 2021). These logs included click streams for web traffic analysis as well as potentially misleading elements such as automatic page reloads and source files. However, richer datasets also introduce greater complexity in analysis. Solving such complexities requires robust tools to process and visualize data, removing the challenges of detailed logs (Walny et al., 2020). Web log analyzers process clickstreams to offer key metrics on user behavior and site performance, utilizing arithmetic and statistical methods to enable precise analytics for informed decision-making (Sakas et al., 2021).

☆ This work was supported by Sakarya University Scientific Research Projects Coordinatorship (Project number: 2010-50-02-024).

* Corresponding author at: Sakarya University of Applied Sciences, Vocational School of Sakarya, Dept. of Computer Tech., Adapazari, 54290, Sakarya, Turkey.
E-mail addresses: canay@subu.edu.tr (Ö. Canay), umit@sakarya.edu.tr (Ü. Kocabıçak).

https://doi.org/10.1016/j.ipm.2023.103617
Received 18 October 2023; Received in revised form 16 December 2023; Accepted 20 December 2023
Available online 26 December 2023

Contemporary web analytics tools present a range of capabilities and constraints in raw data access and extraction (Kumar & Ogunmola, 2020). Cloud-based platforms, including Google Analytics 360 and Adobe Analytics, offer various levels of data access with features such as BigQuery Export. However, these often come with additional costs or limitations due to privacy regulations (Alby, 2023). Similarly, other cloud-based tools either do not provide raw data access or offer it with limitations. In contrast, on-premises open-source tools such as OWA and Matomo offer a more flexible approach that provides comprehensive control over data. Organizations should carefully examine analytics platform data policies to meet their requirements for raw data access and export (Kumar & Ogunmola, 2020).

Analytics tools that primarily rely on client-side JavaScript tracking offer a limited perspective, as they only capture browser-level user interactions. The evolution of Single Page Applications (SPAs), Progressive Web Apps (PWAs), mobile, Internet of Things (IoT), and ad blocker applications has only intensified these limitations, necessitating innovative solutions for effective web analytics (Biørn-Hansen et al., 2017). However, incorporating dynamic IPs adds further complexities in accurately tracking individual user behavior across sessions (Canay & Kocabicak, 2023). This challenge arises from the difficulty in consistently identifying a single user through their various interactions, particularly when their IP address frequently changes. Moreover, multi-server architectures, vital for scalability in high-traffic scenarios, intensify the challenges associated with web analytics. Uniform session management across servers in a web farm necessitates specialized configurations for cross-domain user tracking and data consistency.

Besides these technical considerations, the ethical and legal dimensions of tracking have recently gained significant importance. A thorough understanding of the legal and ethical framework of web analytics is crucial, particularly in devising mechanisms for user consent, data transparency, and opt-out options (Utz et al., 2023). In this context, adhering to regulatory frameworks like GDPR, CCPA, and PIPEDA is essential for enhancing data governance and ensuring ethical compliance (Regueiro et al., 2021). These regulations mandate robust security measures for data integrity and protection against vulnerabilities. Analytical design choices and factors such as ad blockers lead to data inconsistencies. Additionally, the need for real-time analytics presents significant computational challenges. Therefore, a web analytics system must effectively balance technological, ethical, and legal considerations (Demirkan & Delen, 2013).

The advantages and disadvantages of web analytics tools are directly related to their characteristics. Generic web analytics tools like Google Analytics (freemium), Open Web Analytics (OWA), and Matomo (formerly Piwik) focus on user interactions on websites, capturing metrics such as pageviews and events but bypassing server-side application logs (Alby, 2023). These application logs, rich in details about software operations and errors, are not a primary feature of these platforms, even if some integrations may allow limited log analysis. However, despite their user-friendliness and versatility, these tools may not meet all specific organizational needs in log management. Web analytics tools and methodologies face several limitations, such as the inability to fully capture user interactions and challenges with collecting data, especially in complex web applications and multi-server environments. In contrast, a custom-built logging framework, integrated directly into the corporate web application, is tailored to the organization's unique needs, ensuring a holistic and efficient log management and analysis experience. In our prior work (Canay & Kocabicak, 2023), we introduced an innovative data collection method that sidesteps the preprocessing stage in web usage mining (WUM). Building on this, our paper presents a model and framework that address these industry gaps by integrating application logging with web analytics at an enterprise level.

1.1. Research objectives

The primary objective of this research is to establish and validate the CAWAL model, which integrates web analytics with application logging to provide comprehensive data management. Subsidiary objectives include:

- Presenting the CAWAL framework as an alternative to existing tools, emphasizing its enterprise-grade capabilities and on-premises data ownership.
- Demonstrating CAWAL's proficiency in accurately capturing a wide range of user interactions, including SPAs, PWAs, and IoT platforms.
- Highlighting the framework's robust session management and cross-(sub)domain user tracking in enterprise-grade web farms.
- Conducting a comparative evaluation of CAWAL against well-known open-source tools, Open Web Analytics (OWA) and Matomo, regarding features and performance.

The multidimensional hypothesis of this research posits that the CAWAL framework can improve adaptation to multi-server environments, application usage data collection, and troubleshooting with data accuracy, diversity, ownership, and governance on an enterprise-wide scale compared to existing analytics tools.

1.2. Primary contributions

The main contributions of the study are:

1. Introducing the CAWAL model, emphasizing its role in bridging the gap between web analytics and application logging.
2. Presenting the CAWAL framework as an enterprise-grade alternative, promoting centralized data management and privacy.
3. Highlighting CAWAL's ability to capture diverse user interactions, offering an on-premises data ownership advantage.
4. Detailing the framework's proficient session management, particularly in tracking users across various subdomains effectively.
5. Conducting an empirical assessment that underscores CAWAL's superior performance against recognized open-source counterparts in terms of features and response times.

1.3. Structure of the article

Section 2 provides an overview of current web analytics solutions and highlights studies that illuminate their limitations. Section 3 delves into the design and features of the proposed model and framework. Section 4 elaborates on the framework's application to an enterprise web information system and compares it with other on-premises tools. Section 5 presents an analysis of some results derived from applying the framework to the genuine system. Section 6 engages in a comprehensive discussion of the findings. Finally, Section 7 concludes the article by summarizing the significant insights and suggesting potential directions for future research.

2. Related work

Web analytics is a complex, multidimensional domain integrating data interpretation, technological application, and user behavioral studies (Fundingsland et al., 2022). Although it originates from computer science, web analytics, with expanding applications, is extensively studied in fields such as business, marketing, social sciences, health, finance, and others (Ahadi et al., 2022; Amalina et al., 2020). It closely relates to data mining, machine learning, and artificial intelligence within computer science.

Recent studies in web analytics have covered diverse topics and methodologies. Alby (2023) conducted a survey to understand Google Analytics' significant influence on the sector. Kumar et al. (2022) used web metrics with Google Analytics for assessing e-commerce website usability. Cheng and Chen (2022) combined in-depth interviews with website analytics to study user behavior on government websites. The "Web Analyzer" tool, introduced by Hasan (2022), analyzes user behavior using GET/POST methods and performance monitoring. A mixed-method framework for exploring citizen science was developed by De-Groot et al. (2022). Jansen et al. (2022) compared Google Analytics and Similarweb for website interaction evaluation. Jain and Kashyap (2021) proposed an efficient algorithm for preprocessing weblog data. The use of open-source software in web analytics was explored through a case study by Gamalielsson et al. (2021).

In other research, Onder and Berbekova (2021) analyzed European Destination Management Organizations' adoption of web analytics tools. Kumar and Ogunmola (2020) systematically reviewed web analytics tools and techniques from a business perspective. Yang et al. (2022) delved into social media data analytics' potential in business decision-making and introduced the Business Decision Making System (BDMS) framework. Quintel and Wilson (2020) compared Matomo with Google Analytics in a library context. Tao et al. (2020) focused on identifying user behavior patterns through cross-domain log data fusion.

Previous research has often focused on specific aspects such as data accuracy or performance, yet it has not provided a comprehensive solution encompassing all these critical dimensions. While there are a few examples of web analytics platform development like Webalyt (Cegan & Filip, 2017), a notable gap in the literature is the lack of detailed descriptions of the technical infrastructure and working methods of these tools. This gap is primarily due to the dominance of commercial companies in the web analytics field and the extensive time required to develop such large-scale projects. Methods and algorithms in computer science are commonly applied to analyze large web datasets, understand user behavior, and make predictive analyses (Harika & Sudha, 2019). Additionally, various techniques for measuring website or application performance contribute to a deeper understanding of user behavior and trends, aiding in making informed decisions and enhancing user experience and digital strategy (Fundingsland et al., 2022).

2.1. Web analytics methods and tools

Web analytics are categorized into two types: off-site, focusing on external factors, and on-site, which concentrate on user behaviors (Rafiq, 2022). Modern on-site methods have evolved from early log-file-based metrics to use server-side logging and client-side tagging (Cahyanto et al., 2022). Advanced log analyzers like Splunk and Graylog have gained popularity, while traditional log-based methods have seen a decline. The advent of JavaScript tagging in the late 1990s allowed for more detailed metrics, such as tracking unique visitors and session paths, leading to the widespread adoption of cloud-based Software as a Service (SaaS) solutions like Google Analytics (Alby, 2023). Fig. 1 illustrates the commonly used methods for data collection and user tracking in on-site web analytics, including CAWAL's position in this landscape.

Page load speed and usability are crucial in large-scale web applications, as longer load times significantly decrease user engagement and completion of tasks (Nurshuhada et al., 2019; Szalek & Borzemski, 2018). Research indicates that over 40 percent of visitors abandon a website if the page load time exceeds three seconds (Pourghassemi et al., 2019; Stringam & Gerdes, 2019). Therefore, the integration of web analytics tools must be precise to avoid further impacting these factors. Design and software development teams focus on enhancing application performance in response to these insights. The selection of a web analytics tool involves considering data collection methods, location, cost, tracking capacity, real-time data availability, and metrics monitoring. Factors such as data segmentation capabilities, dashboard customization, interoperability, and cost-effectiveness also play a role (Al-Otaibi et al., 2018; Almatrafi & Alharbi, 2023; Boufenneche et al., 2022; Kumar & Ogunmola, 2020). The selection process should align with organizational needs and web application objectives while maintaining user experience and performance standards.

Free services like Google Analytics' basic version offer cost savings and basic insights but may lack in-depth analytics for high-traffic sites (Haaksma et al., 2018; Rafiq, 2022). Conversely, on-premises platforms such as Open Web Analytics (OWA) and Matomo provide data ownership-focused alternatives, using client-side JavaScript for user tracking similar to cloud-based services (Alby, 2023). The selection of a web analytics tool should therefore consider not only organizational needs and goals but also factors like data ownership, competitive benchmarking, and advanced reporting capabilities. This alignment is vital for organizations requiring detailed and extensive analytics for informed decision-making.

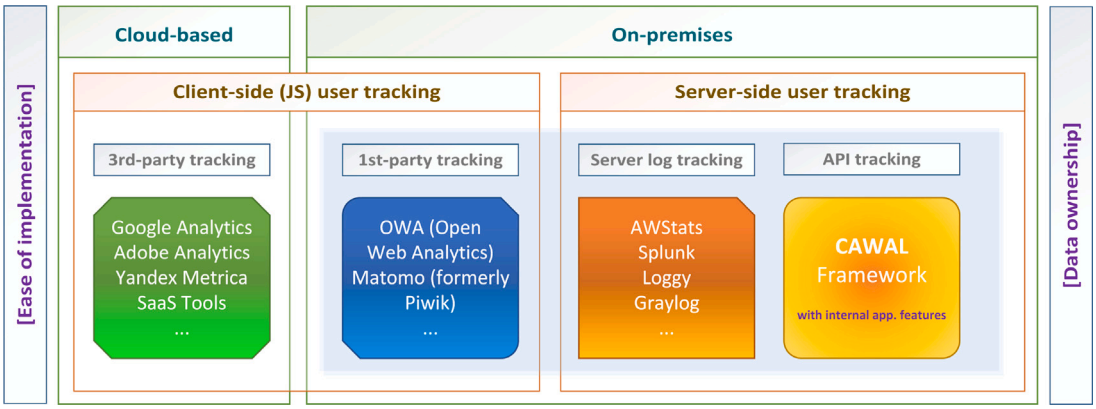

Fig. 1. Commonly used data collection and user tracking methods for web analytics.

2.2. Data issues in web analytics

Accuracy and completeness are critical for drawing meaningful conclusions and making data-driven decisions in web analytics. Jansen et al. (2022) evaluated these aspects by comparing two industry-standard approaches. Besides accuracy, real-time data analysis marks a significant advancement from previous static log file-based methods (Boufenneche et al., 2022). Current tools offer instant insights, enabling rapid responses to user behavior changes or market conditions. However, issues related to JavaScript can affect data integrity, and multi-server, multi-domain systems pose unique harmonization and analysis challenges (Biørn-Hansen et al., 2017).

Tracking users across (sub)domains in a web farm is complicated by decentralized server architectures and web protocol restrictions (Krishnan et al., 2020). In a farm with m servers, each ith server processes a subset n_i of requests, logging metrics for analysis. Naive data aggregation by simply summing these subsets $\sum_{i=1}^{m} n_i$ may miss complexities from sessions distributed across servers. Effective data aggregation and normalization are crucial, aiming to combine logs from all m servers with a worst-case time complexity of $O(m \times n)$. Load balancing complicates user tracking, as interactions may be processed by multiple servers, necessitating a unified tool for data synchronization. Session reconciliation is also key, involving the identification of fragmented sessions from a single user across servers (Mortazavi et al., 2020). This often uses session identifiers in client-side cookies, which, while essential for tracking activities, are limited by domain-specific constraints. Setting the cookie's domain attribute to the primary domain assists in tracking within subdomains, but cross-domain tracking increases in complexity. The reduced effectiveness of third-party cookies due to privacy measures has prompted the exploration of alternative tracking methods (Durántez, 2023).

Session and user identification are essential in web analytics for understanding user behaviors. Session identification can be formally modeled as a function $S(A, T, C)$, where A represents user actions, T the time intervals between these actions, and C contextual data such as IP or browser (Maslennikova et al., 2022). The primary goal is to develop a function g such that $g{:}A \times T \times C \rightarrow S$, grouping actions A into cohesive sessions S based on time T and context C. Challenges in this process include handling Session Timeout (To)—a predefined interval after which a session is considered terminated if T exceeds this limit—and identifying the end of a session due to User Inactivity (Ia) or significant Context Change (Cc). Similarly, user identification aims to recognize users over multiple interactions, represented as $I(U, M, D)$, where U is the set of users, M denotes methods of interaction (e.g., various devices or browsers), and D encompasses available data points such as IP addresses or cookies (Salminen et al., 2020). The function f, defined as $f{:}M \times D \rightarrow U$, attempts to identify a user U based on interaction methods M and data points D. This task is complicated by factors such as Dynamic IP allocation (Id), cookies deletion (Cd), browser fingerprinting ambiguities (Bf), and access from multiple devices (Dm). These complexities highlight the need for sophisticated methods to accurately capture and analyze user behavior and interactions in web analytics.

The session boundary issue notably affects the accuracy of web analytics, especially in measuring user interactions. Traditional methods, using a fixed 30-minute idle time to define session ends (Jansen et al., 2022), often result in misclassified activities. For instance, a user reading an article during this period might incorrectly be counted as inactive (Maslennikova et al., 2022). This misclassification distorts key performance indicators and introduces inaccuracies into the data, affecting decision-making and user experience strategies. To improve session identification, probabilistic models and machine learning consider variables like user behavior and dwell time (Alhlou et al., 2016). The midnight boundary problem, where sessions crossing midnight are counted as separate, further complicates metrics such as session duration and count. Adaptive session timeouts offered by some tools partially address this, but complexities like varying time zones continue to pose challenges (Jansen et al., 2022). Thus, resolving session and midnight boundary issues remains critical for accurate web analytics.

Web analytics is also considered the first stage of business intelligence development (Król & Zdonek, 2020). However, cloud-based solutions may not suit every organization, mainly due to data privacy and sovereignty concerns (Samarasinghe et al., 2022). With growing emphasis on privacy in information societies, control over user data, especially across borders, becomes a significant

Table 1
A sample tuple from the "log_session" table.

Column	Source	Sample data
log_id	Database	19423375
log_opn_id	Application	21091539
log_datetime	Web server	10.05.2020 22:42
log_uid	Database	540
log_username	Database	canay
log_ip	HTTP	193.140.253.81
log_proxy	HTTP	185.169.181.13
log_os_name	User-agent	Windows
log_os_ver	User-agent	11
log_brow_name	User-agent	Chrome
log_brow_ver	User-agent	114
log_brow_type	User-agent	1
log_lang	HTTP	tr
log_country	GeoIP DB	Turkiye
log_cookie_check	HTTP	1
log_landing_url	Web server	https://www.gate.sakarya.edu.tr/
log_ref	HTTP	Google
log_ref_host	HTTP	www.google.com.tr
log_ref_search_key	HTTP	sau gate
log_ref_type	HTTP	4

issue (Everest-Phillips, 2019). Adhering to data privacy laws and regulations, while critical for protection, demands substantial expertise (Binns, 2017). The use of open-source tools like Matomo in academic libraries exemplifies the shift towards alternatives that prioritize user privacy, as opposed to traditional tools like Google Analytics (Quintel & Wilson, 2020). In the context of open government, open-source solutions like Matomo are suggested to ensure data privacy and avoid vendor dependency, paving the way for sustainable web analytics practices (Gamalielsson et al., 2021).

3. Methodology

Web developers commonly log user actions within applications for various purposes. Expanding on this standard practice, we have developed the CAWAL (Combined Analytics and Web Application Log) model and framework, providing integrated web analytics and application logging. This approach, building on the data collection method introduced in our previous work (Canay & Kocabicak, 2023), takes a comprehensive view of enterprise web systems. The practical implementation of this model, the CAWAL framework, encompasses an API for data collection, a database model for structured data storage, and mechanisms for analytical output generation.

CAWAL represents more than just a marketed tool; it is a model applied in a real-world environment. Thus, it is more fitting to describe it as a conceptual model and framework rather than a tool or platform. In the model's context, which represents an abstraction of specific processes or phenomena, CAWAL outlines a comprehensive structure encompassing data collection, storage, analysis, and reporting for web analytics and log management. The term "CAWAL" also refers to the framework we developed to empirically test our research hypothesis. This framework demonstrates the model's superiority over existing web analytics tools in critical areas such as data accuracy, ownership and governance, efficiency, and compatibility in multi-server environments. It operationalizes the conceptual model by integrating a data collection API, a robust database structure, analytics generation methods, and other components suited for advanced web analytics operations in enterprise settings.

3.1. Design of the model

CAWAL is designed as a comprehensive architecture for web analytics in a multi-server environment. It operates server-side and utilizes a load balancer to distribute client requests across a web farm, as shown in Fig. 2. This diagram details the architecture's layers and functionalities. Developed to run on Linux servers, CAWAL leverages Linux's robust and secure environment for enhanced performance and scalability. In this architecture, client requests are distributed evenly among servers in the web farm by a load balancer. These servers access shared resources like configuration files and session directories stored on a Network-Attached Storage (NAS) server. The web application, equipped with a logging API, records user interactions into log tables. Nightly, an analytics extractor, activated by a cron job, processes this data within a dedicated analytics database.

CAWAL's framework incorporates an ETL (Extract-Transform-Load) process (Al-Rahman et al., 2023) for data analytics, storing results in data marts within a data warehouse. This setup allows CAWAL to combine an OLTP (Online Transaction Processing) database (Li et al., 2022) for immediate user data processing with a data warehouse for long-term storage. This integration ensures real-time processing and preservation of data for future analysis, offering a dynamic platform suitable for modern web analytics demands. CAWAL employs object-oriented PHP and integrates with a MySQL relational database, creating a robust analytics solution. PHP manages server-side operations such as session management, while MySQL, optimized for high-speed data retrieval, serves as the main database for data storage and query execution (Alya & Ikhwan, 2022). MySQL's ACID-compliant transactional features

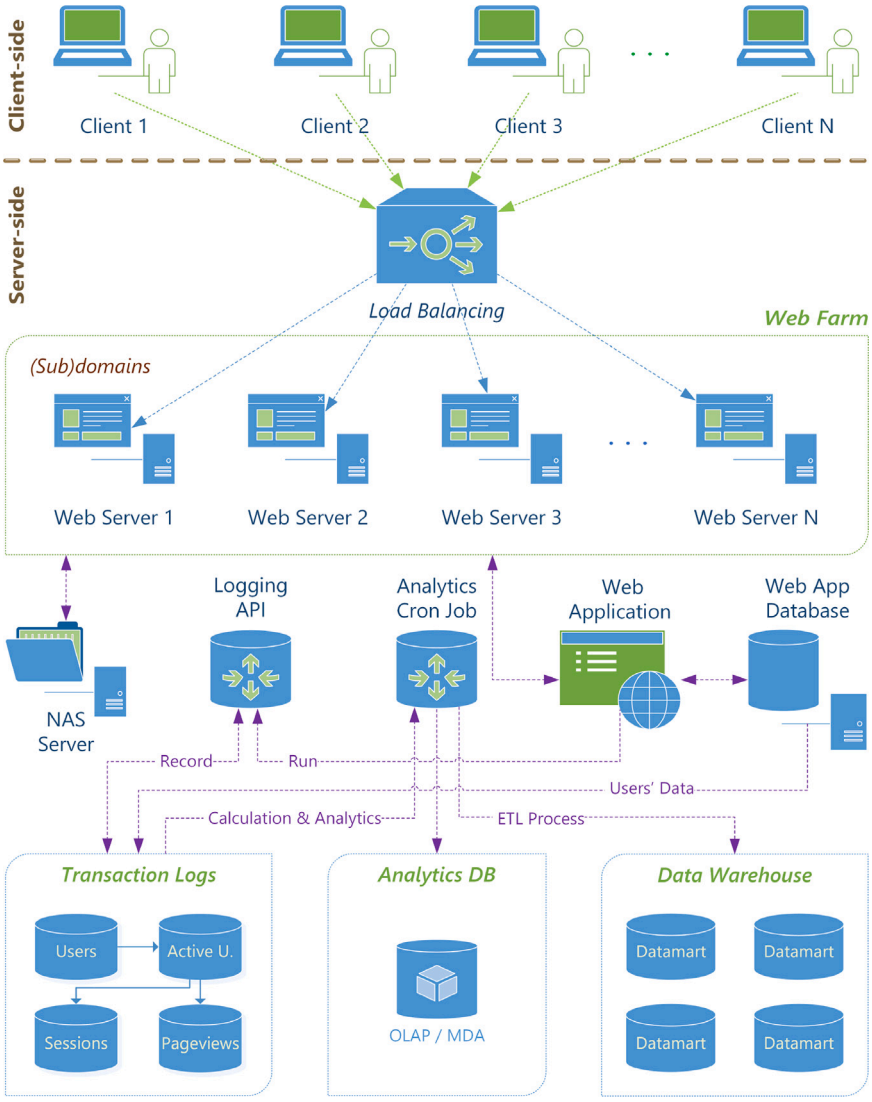

Fig. 2. Full model of multi-server supported web analytics framework architecture.

ensure reliable data persistence, and its SQL functions facilitate data processing, statistical analysis, and time series evaluation. The architecture also supports distributed capture agents via CAWAL's API, resulting in a balanced and comprehensive web analytics solution.

3.2. Data model

CAWAL employs a lean data model characterized by minimal table design to facilitate efficient analytics, rather than minimizing data storage. The framework, adhering to this model, incorporates an Application Programming Interface (API) for structured data acquisition and storage, alongside a database schema optimized for data retention (Ofoeda et al., 2019). The framework's underlying data collection infrastructure, including the primary database tables "log_session" and "log_page", is extensively detailed in our previous paper (Canay & Kocabicak, 2023). The UML diagram in Fig. 3 visually presents the logical data model, showcasing the analytical database, which further elaborates on our earlier research.

Central to the CAWAL framework, the "log_session" table is designed to comprehensively capture each user session. It contains detailed information about user interactions and the technical environment, including data fields such as session ID, user ID, IP address, proxy address, operating system and version, browser details, language, and country (sourced from the GeoIP table). The table also records the initial URL accessed, referring site, search engine keywords, and processing time. A sample tuple from the "log_session" table, depicted in Table 1, illustrates the depth of data collected by the CAWAL framework regarding user sessions. This

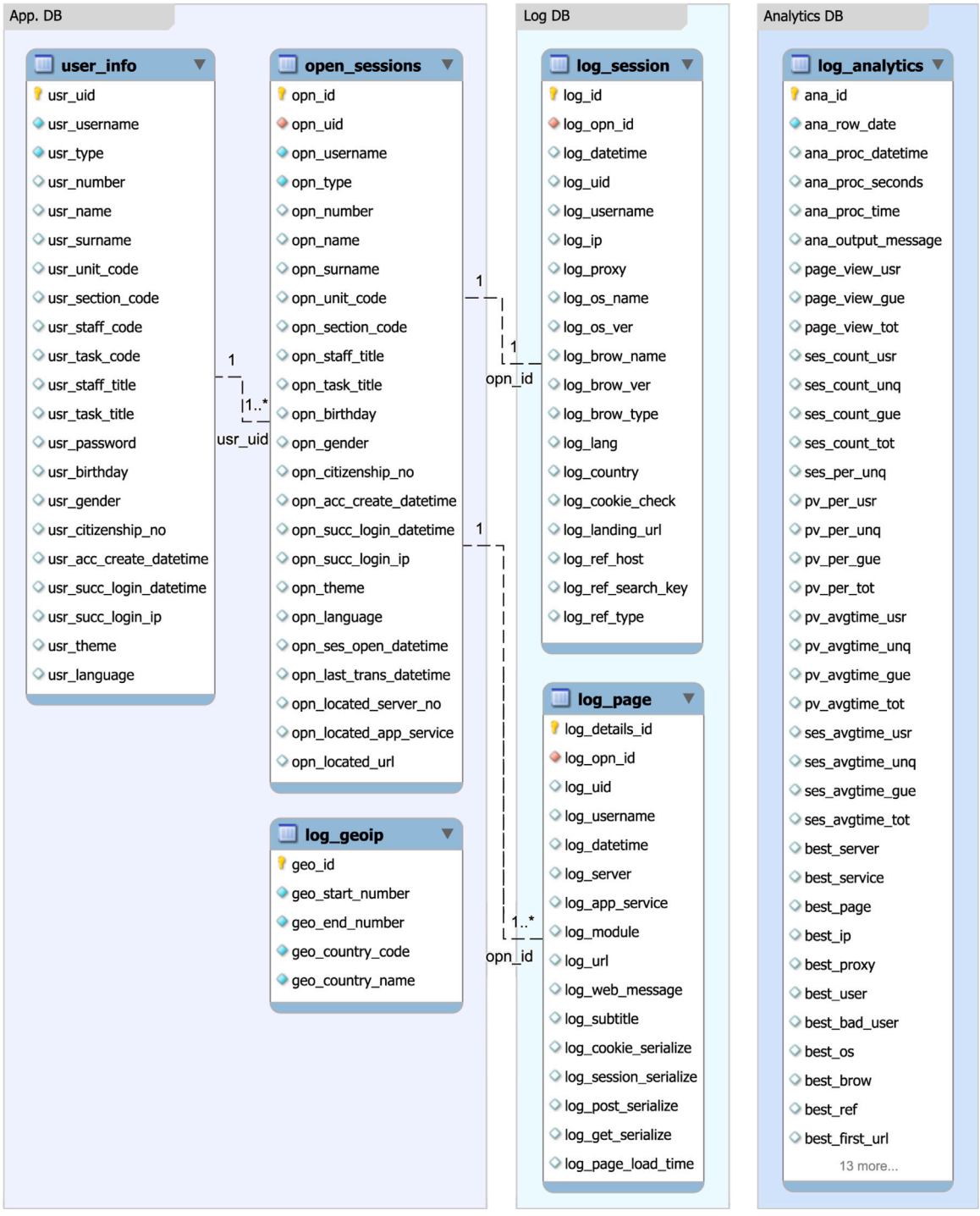

Fig. 3. UML diagram of the logical data model.

comprehensive data collection is vital for identifying patterns in user behavior, demographic distributions, and technical preferences like browser and operating system usage. The inclusion of initial URL and referring site data enhances the understanding of user navigation patterns and source preferences, adding a layer of complexity to the web analytics capabilities of CAWAL.

Another fundamental database table in the CAWAL framework is “log_page”, which serves as an extensive mechanism for tracking page transactions and includes application-specific data. This data encompasses elements such as application header, application

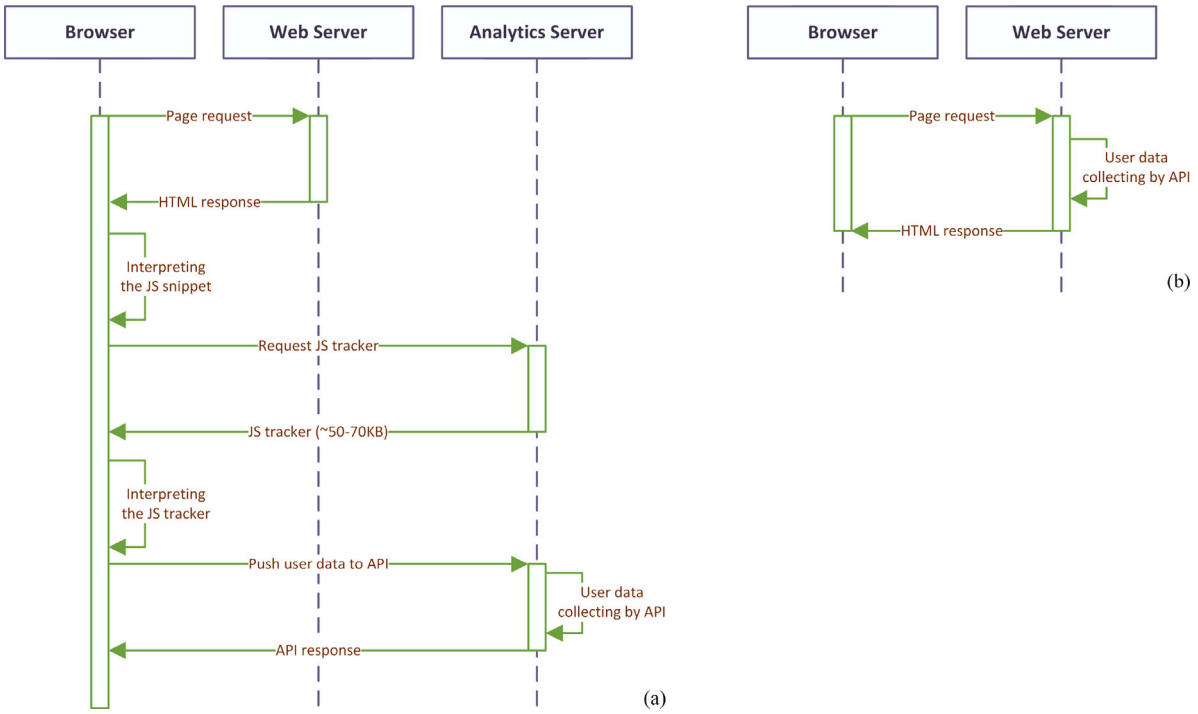

Fig. 4. Sequence diagram of client-side (a) and server-side (b) user tracking.

message, cookies, session details, POST and GET requests, and page load times. Specifically tailored to augment the framework's capabilities, the "log_page" table is instrumental in enhancing software quality. It facilitates error detection and ensures robust application performance by providing intricate details on various application aspects. Web administrators can utilize these database tables for system usage monitoring, drawing valuable insights from real-time analytics information. A key aspect of the "log_page" table is its focus on capturing page production times, an essential metric for web developers. This information is crucial for optimizing web performance, as it helps in identifying slow database queries or code inefficiencies. Therefore, it plays a vital role in enabling developers to fine-tune their applications, ultimately leading to improved user experiences.

The "log_analytics" table within the framework is another critical component, storing complex analytics data processed for each day in a single tuple. This table includes around 50 fields, with the last six fields consisting of serialized multidimensional data. A daily record in this table amounts to approximately 68 kilobytes, reflecting the extensive scope of analytics extraction performed by the CAWAL framework. This level of detailed data storage underscores CAWAL's commitment to delivering an efficient and accurate analytics solution, catering to the nuanced needs of web analytics in enterprise environments.

3.3. Data collection

In the CAWAL framework, data collection centers around metrics and dimensional data, gathered from four primary sources: HTTP requests, network-level, application-level, and external data (Canay & Kocabicak, 2023). HTTP request data, including details from HTTP headers, is captured from request messages. Network-level data, while related to HTTP requests, is not part of the request itself but essential for transmission, such as the client's IP address. Application-level data, transmitted alongside HTTP requests, encompasses session and referral information managed by server-side web programming languages. External data, typically stored in the web application's database, includes elements like country identification from IP addresses and user profiles. These data sources are processed and transformed into meaningful metrics and other valuable data. The CAWAL framework's data collection involves user tracking and data fusion, combining data from these varied sources. It employs a server-side, application-based data collection API to acquire, interpret, format, and store this data. As shown in Fig. 4, CAWAL's interaction with the server for data collection is significantly leaner compared to 3rd-party tools that send extra requests for user tracking.

The data collection process involves gathering both qualitative and quantitative information. Qualitative dimensions, such as the client's browser and operating system, IP address, preceding webpages, channel identification, geolocation, and user profiles, are derived from various sources, including HTTP requests and network protocols. Quantitative metrics, like pageviews and application-specific data, are also obtained from these sources. Integrating these qualitative and quantitative elements into CAWAL entails data cleansing, refinement, and analysis, enabling a comprehensive understanding of user interactions and behaviors. This integrated approach significantly enhances the precision of analytics within the CAWAL framework.

3.4. Data processing and storage

The CAWAL framework systematically aggregates transaction logs during routine operations for effective data processing and storage. Daily log processing generates valuable statistics, stored in a single database record. The framework serves multiple purposes: calculating server load and user density hourly, evaluating user navigation details (like page visits and dwell times), optimizing slow-loading pages, and detecting unauthorized access attempts. It also logs important messages, captures data from cookies and sessions, and processes HTTP methods such as POST and GET. These collected data establishes the groundwork not only for generating comprehensive reports or performing analysis but also for enhancing system performance and security measures. The subsequent sections outline the steps involved in handling these transaction logs.

Data backlog. Especially in large-scale systems, the number of records generated in daily transactions can exceed millions. The sustainability of a database table of this scale in an OLTP structure, its usability for analysis purposes, and the manageability of backup processes will not be efficient. To solve this problem, the CAWAL model employs a comprehensive ETL process. Daily log data is transferred to temporary transaction tables at the beginning of the analytics extraction process after midnight, and OLTP tables are emptied and optimized. After the analytics extraction process on the temporary tables, the processed data is transferred to the data warehouse for long-term storage and future use. Thus, the problem of data backlog and slowdown in OLTP tables is eliminated.

Analytics extraction process. Analytics extraction within the CAWAL framework is an automated process carried out nightly after midnight. This process begins with database maintenance operations, followed by session and pageview information processing from the preceding day. Analytical extraction steps are then executed to create daily web analytics data. This comprehensive procedure performs highly complex and sophisticated queries involving various SQL commands categorized as DDL, DML, and DQL (Brahmia et al., 2022). These queries, the most critical component of the analytics extraction phase, result in a significant amount of single or multi-dimensional data. Multi-dimensional data in the form of arrays are serialized and stored in the database. Optimizing database indexes and extensive queries is critical for data collection and analytics performance in systems that process millions of records daily. To achieve this, considerable enhancements have been made to the database and the algorithm, resulting in a highly efficient script. Critical process information is timestamped and stored in a text-based log file, enabling reliable and database-independent monitoring of script operations over the web.

ETL process and data warehouse. CAWAL incorporates a robust ETL process to structure captured web usage data into a specialized data warehouse. This process starts with analytics extraction, aggregating data into a central data store. Subsequently, tailored transformations are executed to convert this raw data into an analytical schema conducive to reporting and mining. Post-transformation, the operational data is routed to the respective monthly data mart within the data warehouse, utilizing specialized procedures to manage transitions between months and years. This data warehouse strengthens CAWAL's analytical capabilities by providing an environment conducive to multi-dimensional analysis such as OLAP (Keskin & Yazıcı, 2022). It allows the association of various metrics—ranging from sessions and users to pages, timestamps, and traffic sources—thereby enabling the generation of detailed long-term reports, uncovering trends and patterns, performing segmentation analyses, and applying diverse data mining algorithms.

3.5. Features and functionality

The CAWAL framework has been developed to meet the increasing need for robust, adaptive, and responsive web analytics tools in large-scale and complex web environments. It offers a range of features and functionalities tailored to enhance efficiency, reliability, and user experience in modern web systems. Specific features and functionalities contributing to the framework's effectiveness are detailed below.

Multi-server environment adaptation. CAWAL is specifically designed for compatibility with large-scale, multi-server environments, commonly known as web farms. This architecture supports distributed computing across server networks, providing load balancing, scalability, and reliability. In web farms, CAWAL's distributed structure allows for rapid real-time processing of tracking data. The framework efficiently manages computational tasks and load demand, potentially enhancing system performance. This capability reflects CAWAL's strategic alignment with current technological demands and optimization principles.

Cross-domain user tracking. Engineered for environments with multiple Apache servers, CAWAL effectively manages user sessions across different domains and subdomains. It utilizes a common session directory on a NAS server to maintain data consistency for users navigating various web services. The framework includes database fields like "server number" and "service" to identify the server processing each request and the corresponding application service. This approach improves session data management and expands the analytical capabilities, enabling multi-dimensional analysis of user interactions across servers and services.

Session timeout notification. A key feature of CAWAL is the session timeout notification method, enhancing user engagement and maintaining session continuity. The framework monitors user inactivity and alerts users with a notification as their session nears expiration. This method reduces data loss and improves user experience by minimizing unintended session terminations. Particularly useful in applications handling sensitive data, such as Internet banking, it involves users in session management and ensures a secure environment. Unlike other tools that default to a 30-minute session assumption, CAWAL provides a more precise definition of session duration.

Midnight boundary problem solution. CAWAL effectively addresses the midnight boundary issue, crucial for maintaining accurate web analytics. Unlike many tools that extend the session timeout threshold or treat the transition to a new day as a

different session, CAWAL focuses on the session start time. Its analytics extraction script, running in the early morning hours, treats sessions that begin before midnight and continue into the next day as continuous, counting them in the previous day's statistics. This approach provides a more accurate assessment of user interactions, avoiding misclassifications that can distort metrics and impact strategic decisions.

Dynamic IP compliance. In CAWAL's session tracking methodology, sessions initiated by the web server for different devices using the same IP address are recognized as distinct. Operating based on server session variables, CAWAL avoids the confusion of mixing sessions with identical IPs, a common issue with web server logs. Sessions are terminated either upon user logout or following a period of inactivity signaled by a warning window. This precision in session tracking, even amidst dynamic IP allocation, is a key advantage of CAWAL, ensuring accurate user tracking.

Data ownership and governance. Data ownership and governance are pivotal in today's landscape of data security and privacy. As an on-premises solution, CAWAL empowers organizations with complete control over their analytics data assets. The data remains within the institutional infrastructure, adhering to internal policies for privacy and security. This setup makes compliance with local data regulations more manageable. CAWAL's data governance features, including access controls, audit logs, and encryption, enhance both trust and transparency. Institutions can customize data retention strategies, enjoying full control over analytics processes. CAWAL thus promotes organizational autonomy in web intelligence, aligning with ethical standards and values while maintaining data ownership and enabling comprehensive analytics.

Management interface. The CAWAL framework's management interface is a central hub equipped with advanced functionalities for overseeing various aspects of the web analytics process. The user-friendly dashboard grants administrators easy access to essential functions, including user management, security settings, data configuration, and reporting feature customization. Its modular design enables streamlined navigation and efficient adjustments to system parameters. Responsive to organizational needs, this panel simplifies the management and analysis of web analytics data, making it a versatile tool for administrators.

Real-time monitoring. Real-time monitoring is crucial for web and system administrators to promptly detect and address issues. CAWAL's framework offers real-time analysis capabilities, facilitating proactive responses to prevent potential escalations in system and user experience issues. Administrators can monitor the system's current status, including server, service, and module activity, visitor numbers, user types, and IP address information. The admin panel also allows for immediate administrative actions like banning or unbanning users, enhancing operational control within the system.

Reporting and visualization. CAWAL excels in presenting complex web analytics data through advanced reporting and visualization tools. These components are designed to extract and display diverse metrics, such as user engagement, traffic sources, and behavioral patterns. The reporting tool is customizable to meet specific business objectives, while the visualization feature translates raw data into interactive formats like charts, graphs, or heatmaps. This combination not only facilitates intuitive understanding and trend identification but also aids in making informed tactical and strategic decisions based on real-time insights.

4. Implementation and evaluation

CAWAL was successfully implemented in the Campus Automation Web Information System (CAWIS) at Sakarya University (Canay et al., 2011), a critical step in validating the multi-dimensional hypothesis of the framework. Its deployment has significantly enhanced the monitoring of the entire system, including server and web service densities, and has deepened the understanding of user behavior and usage practices. This implementation has led to substantial improvements in the web application's quality and security, confirming CAWAL's advantages over existing tools in performance and data governance.

4.1. Web application integration

Within the CAWIS, CAWAL is intricately embedded into the application code, enabling the recording of events like exceptions and user flows that are typically invisible to external trackers. The framework's APIs facilitate the insertion of custom, application-specific data, such as form fields, into the tracking logs. Additionally, by monitoring application servers, CAWAL enriches session data, offering a comprehensive view of user interactions. Its data collection API, integrated at the start of the application code, is invoked with every page request, ensuring seamless alignment with the application's systematic design. This integration abstracts the complexities of logging, allowing developers to concentrate on core functionalities. The API operates within the layered framework throughout the application's lifecycle, discreetly gathering data. It updates the page log table with details like page loading times, database query delays, and errors before the web application code concludes. This level of integration provides unique insights into the application's operations, contributing to continuous improvements and optimization.

4.2. Web farm integration

The implementation of CAWAL across CAWIS's extensive network, encompassing ten web servers and multiple subdomains, showcases its sophisticated approach to managing heavy web traffic and analytics. The framework efficiently operates alongside a well-configured load-balancing mechanism, tracking transactions across different servers and maintaining robust performance during high user activity periods. The integration with a NAS server, serving as a centralized repository for session and settings data from the web servers, adds a layer of consistency to CAWAL's implementation. This setup ensures uniform session data management across the web farm, supporting scalability, uninterrupted service, and a flexible solution for the complex web environment of CAWIS.

Table 2
Comparison of diverse methods and tools in web analytics based on features and capabilities.

	Web server logs (Apache)	Google analytics (Cloud)	OWA/Matomo (On-premises)	CAWAL (On-premises)
Platform independence	No	Yes	No	No
Application independence	Yes	Yes	Yes	No
Intensive local resource utilization	No	No	Yes	Yes
Setup requirements and coding effort	Low	Low	Medium	High
User tracking (sided)	Server-side	Client-side	Client-side	Server-side
User tracking (based)	Web server-based	JavaScript-based	JavaScript-based	Application-based
Single HTTP request to web (log) server	Yes	Yes	No	Yes
Additional load to the client browser	No	Yes	Yes	No
Additional load to the web server	No	No	Yes	Yes
Cross-(sub)domain tracking	Limited	Limited	Limited	Yes
Session timeout notification	No	No	No	Yes
Session boundary detection	Configurable	Improved	Improved	Adaptive
Midnight boundary handling	Baseline	Partial	Partial	Full
SPA and PWA integration	Natural support	With Effort	With Effort	Natural support
Mobile and IoT compatibility	Natural support	With Effort	Partially	Natural support
Application-specific (custom) data	No	Limited	Limited	Unlimited
Local data warehouse infrastructure	No	No	No	Yes
Data accumulation method	Local (Service)	Cloud (SaaS)	Local (API)	Local (API)
Free access to raw data	Yes	No	Yes	Yes
Data sovereignty and ownership	Yes	No	Yes	Yes
Type of collected data	Semi-structured	Structured	Structured	Structured
Complexity of data transformation	High	Medium	Low	Medium
Suitability of data for WA	Low	High	High	High
Suitability of data for WUM	Medium	No	Medium	High

4.3. Time complexity of the framework

The CAWAL framework's server-side application-based web analytics process begins with log collection. In this stage, each user request to the server, totaling n requests, generates a log entry, leading to a linear time complexity of $O(n)$. The subsequent phase, data transformation, involves performing k transformations on each log entry. Examples of such transformations include converting IP addresses into geographic locations. This step results in a time complexity of $O(kn)$. Database storage of the processed information introduces additional complexity. This stage's complexity depends on factors like the database's structure and indexing, contributing to the overall computational load. As data is further processed, particularly during normalization where it is grouped by sessions or users, the time complexity initially appears to be quadratic, $O(n^2)$. However, the use of optimized algorithms can significantly reduce this to $O(n\log n)$ or maintain it at linear time. Finally, the aggregation and summary operations in the framework, which produce data summaries, vary in complexity. Simple aggregation operations, such as counting pageviews or sessions, have a linear time complexity of $O(n)$. In contrast, more complex aggregation tasks, which may involve multi-dimensional data analysis, necessitate greater computational resources and exhibit higher time complexity.

4.4. Feature comparison

Comparing the methods and tools used for web analytics can be approached both qualitatively and quantitatively. An empirical evaluation can be conducted by examining selected methods and tools according to various criteria in terms of qualitative characteristics. A feature comparison of web server logs, Google Analytics (freemium), open-source web analytics tools (OWA and Matomo), and the CAWAL framework is given in Table 2.

Although a comparison with free solutions has been made here, it is worth noting that some of the restrictions in the freemium version of Google Analytics are not available in the paid version, GA360. These methods should also be considered regarding capabilities such as performing real-time analysis, operating under heavy load, obtaining non-standard and application-based data, privacy, and data security. The proposed model is quite successful compared to traditional methods in terms of its advantages, such as data accuracy, completeness, timeliness, privacy, scalability, and ability to adapt to future data needs and insight research. CAWAL, which integrates with other tools and systems the organization uses, is designed in a structure that can handle increasing amounts of data and user traffic over time.

4.5. Performance comparison

Due to their inherent differences, comparing client-side and server-side web analytics tools and measuring their effectiveness against each other presents complex challenges and requires a combination of sophisticated approaches. In addition to the qualitative approach, the differences in the page loading times on the client side to be examined within the framework of searching for common ground between the methods can be a quantitative performance indicator. In this section, the performance of the proposed CAWAL

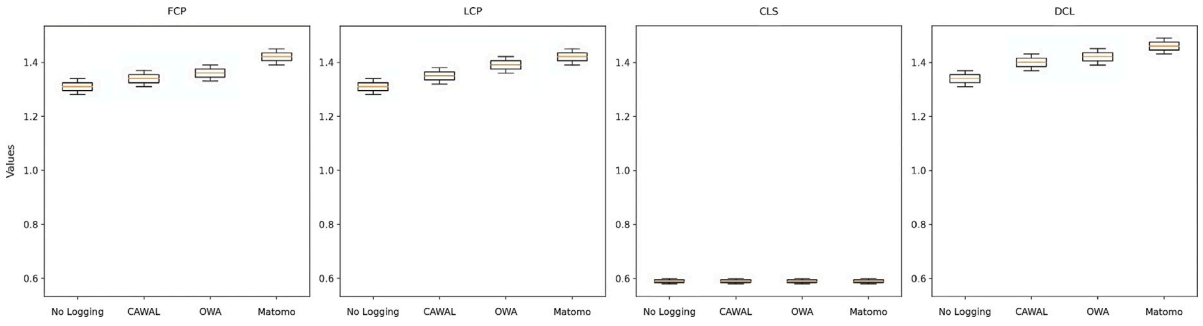

Fig. 5. Comparison of equivalent on-premises web analytics tools based on client performance metrics.

framework is compared with the leading open-source analytics tools, OWA and Matomo. However, it is imperative to highlight the inherent limitations restricting the direct comparison of these systems with SaaS web analytics tools like Google Analytics and Adobe Analytics. To create the experimental environment, CAWAL, OWA, and Matomo frameworks have been made to run locally with the sample corporate web application on a workstation with an Intel i7 processor and 16 GB of RAM. All performance measurements, including server and client roles, were performed on a Windows 11 OS workstation. For web applications to work correctly, all hostnames, including subdomains, are redirected to localhost via the "hosts" file of the Windows OS. The logging framework selection during the request has been provided with a GET parameter added to the main page of the experimental website. According to the incoming parameter value, the JavaScript snippet of the relevant framework has been included in the HTML content and sent to the client.

4.5.1. Client-side performance analysis

In client-side web analytics tools, data such as JavaScript-generated session and user ID, referrer, and screen resolution are transmitted via GET or POST type API requests. For this study, the entire experimental setup was on a single computer to measure the load of selected web analytics tools on both client and server sides. Unlike other tools, CAWAL does not involve client-side processes, whereas other tools make two third-party requests (one script and one API). The transmission of a JavaScript file over 50 kilobytes to track the user introduces some latency on the client side. Despite typically running asynchronously and therefore having a relatively low impact on users' time-to-feel interaction (TTI), these scripts still load both the server and the client's processor (Hossfeld et al., 2018).

Page load times were measured using Chrome DevTools' performance insights tool (Bielak et al., 2022), under fast 3G conditions with no CPU throttling. Average times were calculated from N = 10 consecutive repetitions. Prior studies suggest that the median of five consecutive runs offers a reliable measure while keeping testing time reasonable (Chan-Jong-Chu et al., 2020; Heričko et al., 2021; Yu & Benson, 2021). Performance varied across devices, but comparing tools in the same environment and averaging repeated measures provided meaningful insights. A range of performance metrics like FCP, LCP, and DCL assesses page load speed and responsiveness, crucial for user experience (Armaini et al., 2022; Dewi & Nurdin, 2023; Saif et al., 2021). Fig. 5 shows the performance measures for the considered analytics platforms.

The performance results indicate that the web analytics tools exhibit similar client-side performances with minor differences. As expected, the quickest page load time on the client side was observed when the web application was run without any analysis tool. CAWAL, with its streamlined structure, enabled faster page loading than other tools, outperforming both OWA and Matomo. Matomo typically showed the highest values in the FCP (First Contentful Paint), LCP (Largest Contentful Paint), and DCL (DOM Content Loaded) metrics, indicating a slower performance in these areas. In contrast, CAWAL's performance met expectations for more rapid page loading. The CLS (Cumulative Layout Shift) metric demonstrated minimal variation between the tools, suggesting consistent layout stability. The data gathered under these controlled conditions offer valuable insights for choosing the most suitable framework, tailored to specific performance requirements and optimization objectives.

4.5.2. Server-side performance analysis

This comprehensive server-side performance analysis aims to reveal the resource utilization and response times of selected methods or tools. The measurements were performed using Apache JMeter (Junmei & Jihong, 2019) version 5.5, the leading open-source software in this field. JMeter operates at the protocol level and emulates browser-like behavior, resembling the actions of one or more browsers (Vasylyshyn et al., 2023). However, it is important to highlight that JMeter's functionalities do not encompass the full spectrum of browser capabilities. JMeter does not execute JavaScript code embedded within HTML pages, nor does it render HTML pages in the manner characteristic of conventional browsers. Nevertheless, an adequately designed JMeter test may closely mirror the interactions of genuine users employing browsers. To perform the server-side performance analysis, a custom test script has been created in this study that is appropriately configured by considering client-side web analytics tools' JavaScript and API requests. First, the non-analytics version of the web application and then the tools to be compared were tested individually using a different value given to the GET parameter. For each test, 150 threads (requests) were executed in 15 s, and these results were then combined to produce the graphs shown in Fig. 6.

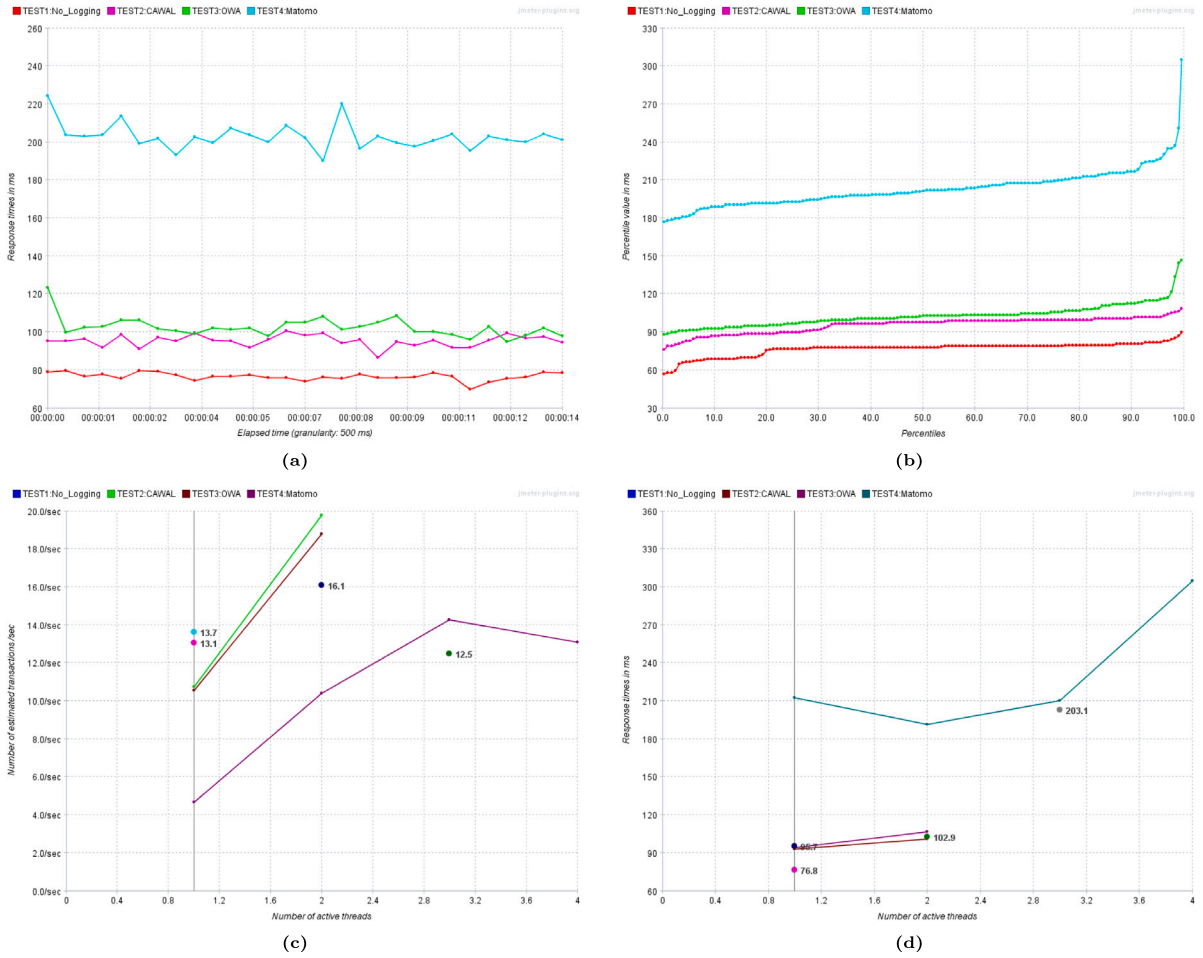

Fig. 6. Load test response performance comparison of equivalent on-premises web analytics tools.

Fig. 6(a) presents the baseline response time of the web application without any logging, which stands at an average of 76 ms. Upon the integration of CAWAL, this figure sees a slight increase to 95 ms. In comparison, OWA's inclusion results in a response time of 101 ms. The deviation becomes most pronounced with Matomo, where the average response time escalates to 204 ms. Compared to the others, CAWAL offers a response time of about 24% faster than OWA and 85.16% faster than Matomo. Fig. 6(b) sheds light on the cumulative percentiles. Although they are close to each other, CAWAL outperforms OWA, while Matomo has a significant negative divergence. Progressing to Fig. 6(c), which focuses on the number of estimated transactions per second, the no-logging scenario demonstrates the highest throughput, as expected again. However, this figure diminishes incrementally with the inclusion of CAWAL, OWA, and Matomo, most prominently. The ability of a system to address a more considerable number of requests per second inherently translates to heightened performance. Lastly, Fig. 6(d) shows that the response time concerning the number of active threads is close to each other in CAWAL and OWA, while it is almost twice as high in Matomo. Their underlying structure and complexity can explain the differentiation of results on the various platforms.

Besides load testing, examining hardware resource utilization is essential for a holistic understanding of system performance. Fig. 7 shows the server's resource utilization, such as CPU, memory, disk, and TCP, measured via JMeter's PerfMon module during load testing. CAWAL's performance in terms of CPU consumption, as seen in Fig. 7(a), is remarkably similar to the no-logging benchmark, emphasizing efficient compute flow. In contrast, Matomo exhibits increased utilization due to computational complexities or less optimized algorithms. Memory usage insights in Fig. 7(b) suggest Matomo might retain more data, potentially pointing to caching strategies needing improvement, while CAWAL and OWA exhibit close efficiencies despite architectural differences. Disk performance in Fig. 7(c) indicates CAWAL's balanced memory and disk operational dependency.

Network utilization in Fig. 7(d) highlights Matomo's higher resource consumption, pointing to its complex design or less streamlined communication protocols. These findings confirm expectations based on the distinct internal designs of CAWAL, OWA, and Matomo. At this point, CAWAL exhibits a systematic separation of data logging and processing mechanisms, which sets it apart from other evaluated tools. Tracking data is processed instantly, while analytic inference occurs post-midnight, during times of lower server load, ensuring optimum resource utilization. This approach is further amplified by a distributed structure maintained

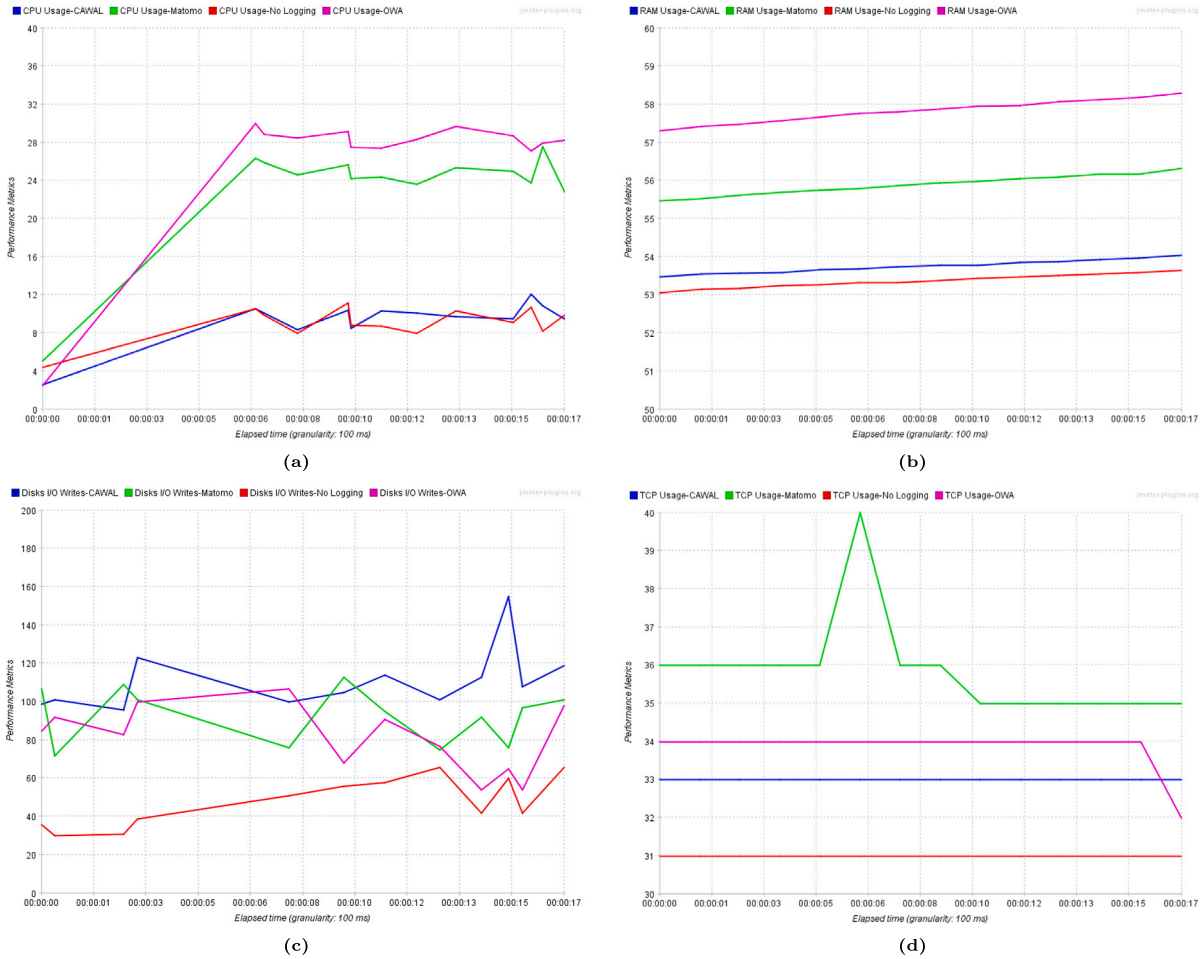

Fig. 7. Load test resource usage comparison of equivalent on-premises web analytics tools.

over a web farm, enabling rapid calculations when real-time data processing is required. Such strategic management of resources and efficient allocation of processing tasks exemplifies the holistic understanding of system performance and demands, reflecting a well-orchestrated approach to analytics.

5. Results

This section presents the main results from a comprehensive evaluation of the CAWAL framework, as applied in CAWIS, a corporate web application. The analysis is based on data collected over 24 h on a specific weekday in 2018, encompassing 22,104 sessions and 161,672 pageviews. The record size of a tuple in the tables where the data is stored is around 0.25 KB for the session table and 1 KB for the pageview table. Multiple factors, such as user behavior patterns, geospatial trends, and the efficiency of tracking users across varying domains within a multi-server architecture, were examined in the study. The insights gained through robust data collection lay the groundwork for an in-depth analysis of CAWAL's potential applications, implications, and future research directions in web analytics and web usage mining. The dataset generated encompasses a wide array of metrics and features, including but not limited to site engagement, demographic profiles, device types, network routing, server configurations, and different modes of access. This expansive dataset allows for various analytical pursuits, ranging from time-based trends to comparative assessments, and has potential utility in areas like anomaly detection and system load forecasting. The reliability of the methodology was confirmed by the evaluation of the collected data, which yielded consistent results and suggested new directions for future research. Additional analyses will be discussed in subsequent sections, elaborating on the accuracy and significance of the data obtained. The breadth of the collected data substantiates the viability of even more elaborate analyses than those presently reported.

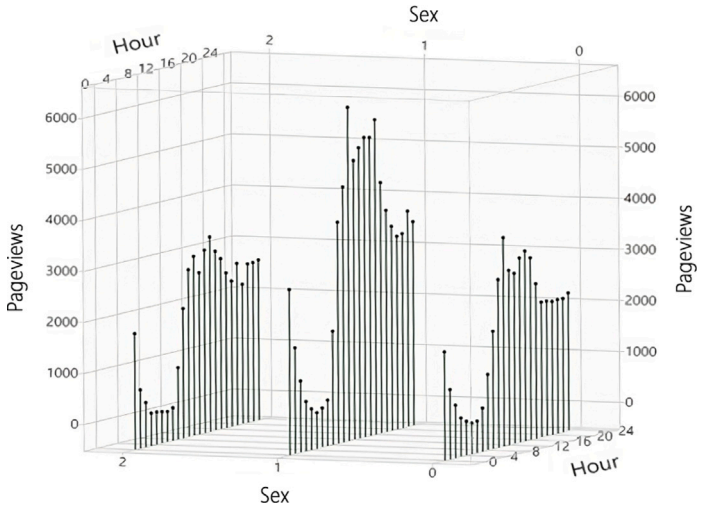

Fig. 8. Analysis of daily pageviews by the dimensions of hour and sex.

Table 3
Number of sessions and pageviews by in-house, in-country, and out-country.

Origin of incoming IP	Number of users	Number of sessions	Number of pageviews	Pageviews per user	Sessions per user	Pageviews per session
In-country	8,706	11,197	76,069	8.74	1.29	6.79
In-house	5,773	8,701	69,443	12.03	1.51	7.98
Out-country	1,702	2,206	16,160	9.49	1.30	7.33
Overall total/avg.	16,181	22,104	161,672	9.99	1.37	7.31

5.1. Time distribution analysis

Understanding the usage of the system through specific metrics and proportional values is paramount for evaluating its efficiency and accessibility. Demonstrating CAWAL's ability to analyze data in multiple dimensions, Fig. 8 provides a visualization illustrating the time distribution of daily pageviews by hour and sex.

Each line, marked by distinct points, represents the number of pageviews for three categories: N/A (0), Male (1), and Female (2). This graphical representation effectively depicts the interaction patterns of users of different sex with the website throughout the day. The N/A category, encompassing unit/department accounts and visitors who are not logged in, exhibits a distinct trend. It shows variations in browsing behavior over a 24-hour period. Males, for instance, show a peak in midday hours, while the pattern for females is more dispersed. A closer examination of the N/A category could lead to further segmentation, isolating the behavioral characteristics of unit accounts from casual visitors. These observations contribute to a deeper understanding of user engagement and offer directions for future research in web analytics, focusing on personalized user experiences.

5.2. Location-based analysis

An essential aspect of web analytics involves understanding users' geographical distribution and engagement patterns. Table 3 highlights the division of user interactions based on the origin of incoming IP addresses, categorizing the data into three distinctive groups: in-house, in-country, and out-country. This categorization facilitates understanding the system's geographical reach and internal utilization, guiding user engagement and localization strategies.

The data shows that in-house users have higher pageviews per session, and user reflects increased organizational engagement. The predominance of in-country users highlights the site's significance within the local context, suggesting opportunities for increased localization and user engagement. Additionally, the differentiation between in-house and other categories provides valuable insights into internal operations and how users interact with the system based on location. This information guides managers in making informed decisions on resource allocation, user experience enhancement, and strategic planning tailored to geographic trends and specific organizational needs.

5.3. Interaction analysis

Examining user interactions with the web system requires a detailed analysis of various elements influencing user navigation and behavior. A comprehensive view of user interactions is presented in Fig. 9 across four aspects: referrer types, top referring sites,

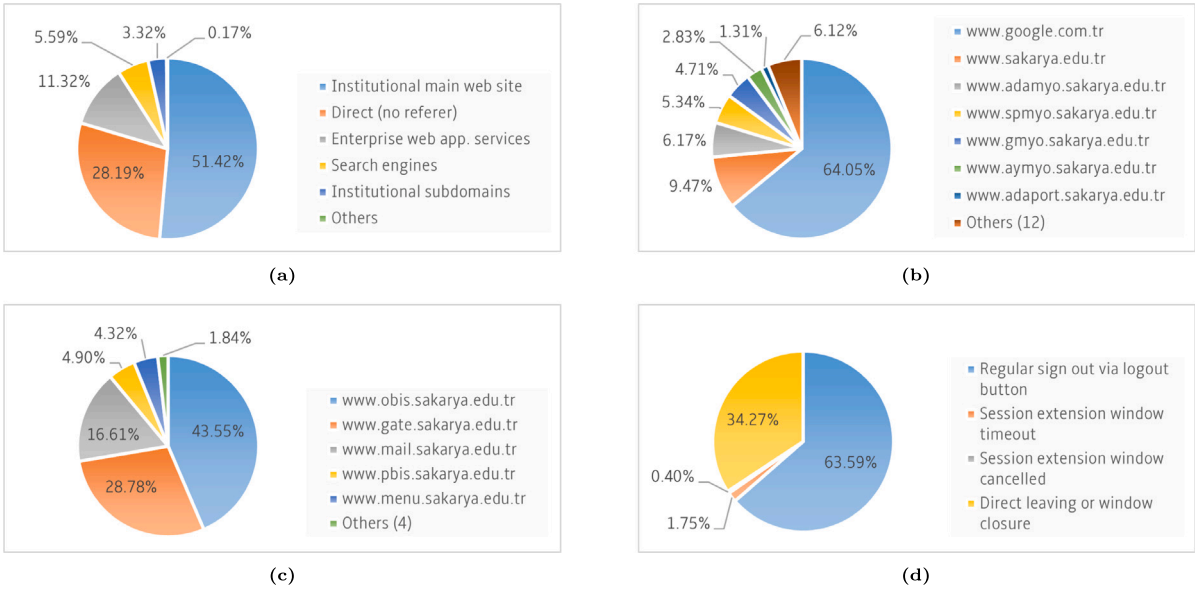

Fig. 9. Frequency of referrer type, referrer sites, landing sub-domains, and logout types by number of sessions.

landing sub-domains, and logout types by the number of sessions. The frequency distribution of the six distinct types of referrers is shown in Fig. 9(a). As expected, the highest percentage belongs to the corporate main website, followed by direct access without a router.

Fig. 9(b) shows that access to the site is predominantly through the Google search engine. The corporate website and other various corporate sub-domains follow this. Fig. 9(c) presents landing sub-domains for arrival to the application. Considering that it is a university web portal, the Student Information System (OBIS) has the highest rate, as expected. Fig. 9(d) shows that while most prefer to log out regularly, many users leave the site by closing the window directly. This information is crucial for understanding user behavior at exit points and optimizing the exit process.

5.4. Server-based analysis

Illustrating CAWAL's capabilities in a web farm environment, Fig. 10 shows the number of pageviews, sessions, and unique server users. In the analyzed period, the distribution of sessions to servers was between 13.3 and 15.3 percent, and the distribution of pageviews to servers was between 13.3 and 15.2 percent. These rates indicate a balanced distribution of request load to servers.

During testing on this web farm system, the first two servers had been temporarily out of service while the 10th server was acting as a test, deployment, and NAS server but not providing direct web services to users. When the pageviews per session (PpS) rates are analyzed, the distribution between the servers is quite close to each other, ranging between 3.62 and 3.86. These results, which show that the requests are correctly distributed to the servers and that the system provides a balanced service with all servers, are incredibly precious in demonstrating the adaptability of CAWAL to the web farm architecture.

6. Discussion

This research contributes significantly to the ongoing discourse on how web analytics and usage mining can be harnessed to create more meaningful and responsive online environments. Implementing the CAWAL framework to a genuine corporate web application has provided a holistic view of user behavior, location-specific trends, and other metrics across different domains within a multi-server setting. The empirical results support the multidimensional hypothesis by confirming that CAWAL excels in several critical areas, especially data accuracy, data governance, performance efficiency, and multi-server compatibility. The number of analyses, of which we can only present some of them here, can be increased in line with the information desired to be obtained. A thorough comprehension of these tables and graphs is critical for site organization, content development, and interface design. These detailed analyses, conducted on a rich and diverse dataset, have also illuminated trends and patterns that carry substantial implications for future strategies and research. For instance, the sex analysis revealed intricate interaction structures, with male users dominating in numbers but female users exhibiting longer average page durations. This trend may guide content and design strategies tailored to different user experiences. Further, location-based analysis has unveiled vital insights into how geographical location influences user engagement. Higher pageviews per session among in-house users, for instance, reflect increased organizational engagement, providing a pathway for further localized content development.

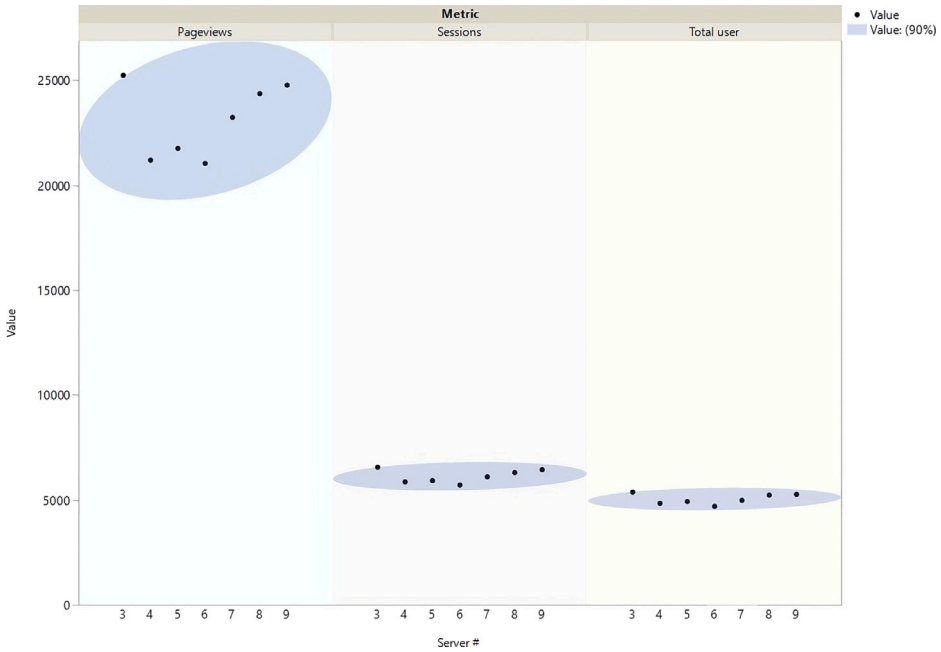

Fig. 10. Distribution of pageviews, sessions, and unique users by servers.

The insights gained from understanding user navigation and behavior patterns, such as the preferred site access and exit methods, offer valuable information for optimizing user experience. Besides this, the web farm environment analysis demonstrated the system's robustness and efficiency by showing how CAWAL's adaptability can be realized on different servers and how session continuity is managed at the domain and subdomain levels. Moreover, the distribution of sessions and pageviews across servers with varying pageviews per session proves the robustness of the load-balancing system implemented. This aspect of the study is significant for organizations seeking to enhance their server deployment strategies. Visualization of daily pageviews and hourly variations and understanding the unique trajectories of various user categories opens new opportunities for content delivery and engagement strategies. In synthesizing the critical insights from this extensive review, the study underlines the adaptability, precision, and comprehensiveness of the CAWAL framework in web analytics. Robust data collection and analysis will have enabled a deeper understanding of user engagement, fostering improved strategies and user-centric content within web analytics. By integrating these insights, institutions and organizations can craft personalized, efficient, and strategically aligned web experiences that respond dynamically to user needs and preferences.

7. Conclusions

In traditional web analytics, client-side data collection often misses critical insights, including application processes, bot interactions, and mobile app activities. This practice, coupled with storing corporate data externally on cloud platforms, raises data privacy, security, and sovereignty concerns. Existing tools frequently show gaps in comprehensive multi-domain and distributed server tracking. The CAWAL model introduces an innovative on-premises solution for in-depth enterprise web analytics. Unlike traditional methods, the model employs a server-side strategy, enhancing its integration with enterprise infrastructures. This approach broadens the capture range, offering more accurate data without taxing the client. A distinctive feature of CAWAL is its combined focus on application monitoring and web analytics, fostering early detection of issues. By emphasizing improved application security and software quality and incorporating advanced analytical techniques, CAWAL emerges as a formidable solution in web analytics. The framework's adaptability and emphasis on data ownership make it vital for organizations prioritizing data sovereignty. Its design caters to various data storage systems, serving organizations of different scales and sectors. CAWAL offers a tailored on-premises web analytics solution that emphasizes flexibility, precision, and data sovereignty.

The current study relies on testing from a single corporate web application, limiting the generalizability of results. For a broader evaluation of its applicability and impact, deploying the framework across multiple domains is essential. CAWAL encounters challenges that other on-premises solutions face, such as computational efficiency and real-time responsiveness, which fluctuate depending on infrastructure and software configurations. Further challenges include the lack of device-specific data, like display resolutions, commonly available in client-side tracking solutions. However, these challenges can be overcome with additional solutions if needed. Moreover, CAWAL's on-premises design demands significant in-house technical expertise for seamless implementation and adaptation, potentially posing challenges for entities with constrained technical resources. A thorough cost analysis of initial and recurring expenses is imperative when adopting on-premises versus cloud-based options.

Future work should focus on expanding CAWAL's deployment to new domains and use cases for enhanced evaluation of its applicability and impact. Optimizing infrastructure reliance is recommended for improved real-time responsiveness. Examining the data warehouse architecture configuration to store accumulated data could reveal strategic insights via multidimensional analysis. Furthermore, exploring the rich application and usage data the framework collects through data mining can provide valuable insights. Investigating mobile, IoT, and smart device integration expands user tracking capabilities across platforms. Moreover, assessing the potential of the framework to facilitate more dynamic security responses via adaptive learning also holds potential. Addressing these future directions promises substantial progress in web analytics, web usage mining, and cybersecurity.

CRediT authorship contribution statement

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

Data availability

The data that has been used is confidential.

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

During the preparation of this work the authors used DeepL, ChatGPT, Claude, 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.

Acknowledgment

This research received administrative backing as part of the doctoral thesis project (2010-50-02-024) from the Sakarya University Scientific Research Projects Coordinatorship.

References

Ahadi, A., Singh, A., Bower, M., & Garrett, M. (2022). Text mining in education—A bibliometrics-based systematic review. *Education Sciences*, *12*(3), 210. http://dx.doi.org/10.3390/educsci12030210.

Al-Otaibi, S., Alnassar, A., Alshahrani, A., Al-Mubarak, A., Albugami, S., Almutiri, N., & Albugami, A. (2018). Customer satisfaction measurement using sentiment analysis. *International Journal of Advanced Computer Science and Applications*, *9*(2), http://dx.doi.org/10.14569/ijacsa.2018.090216.

Al-Rahman, S. Q. A., Hasan, E. H., & Sagheer, A. M. (2023). Design and implementation of the web (extract, transform, load) process in data warehouse application. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *12*(2), 765. http://dx.doi.org/10.11591/ijai.v12.i2.pp765-775.

Alby, T. (2023). Popular, but hardly used: Has google analytics been to the detriment of web analytics? In *Proceedings of the 15th ACM web science conference 2023* (pp. 304—311). ACM, http://dx.doi.org/10.1145/3578503.3583601.

Alhlou, F., Asif, S., & Fettman, E. (2016). *vol. 1*, *Google analytics breakthrough*. Wiley, http://dx.doi.org/10.1002/9781119266365.

Almatrafi, A. M., & Alharbi, Z. H. (2023). The impact of web analytics tools on the performance of small and medium enterprises. *Engineering, Technology and Applied Science Research*, *13*(5), 11753–11762. http://dx.doi.org/10.48084/etasr.6261.

Alya, R., & Ikhwan, A. (2022). Mobile-based design of information system for vocational internship activities for vocational students. *Sinkron*, *7*(4), 2361–2368. http://dx.doi.org/10.33395/sinkron.v7i4.11764.

Amalina, F., Hashem, I. A. T., Azizul, Z. H., Fong, A. T., Firdaus, A., Imran, M., & Anuar, N. B. (2020). Blending big data analytics: Review on challenges and a recent study. *IEEE Access*, *8*, 3629–3645. http://dx.doi.org/10.1109/access.2019.2923270.

Armaini, I., Dar, M. H., & Bangun, B. (2022). Evaluation of labuhanbatu regency government website based on performance variables. *Sinkron*, *7*(2), 760–766. http://dx.doi.org/10.33395/sinkron.v7i2.11404.

Bielak, K., Borek, B., & Plechawska-Wójcik, M. (2022). Web application performance analysis using angular, react and vue.js frameworks. *Journal of Computer Sciences Institute*, *23*, 77–83. http://dx.doi.org/10.35784/jcsi.2827.

Binns, R. (2017). Data protection impact assessments: a meta-regulatory approach. *International Data Privacy Law*, *7*(1), 22–35. http://dx.doi.org/10.1093/idpl/ipw027.

Biørn-Hansen, A., Majchrzak, T. A., & Grønli, T.-M. (2017). Progressive web apps: The possible web-native unifier for mobile development. In *Proceedings of the 13th international conference on web information systems and technologies* (pp. 344–351). SCITEPRESS - Science and Technology Publications, http://dx.doi.org/10.5220/0006353703440351.

Boufenneche, W., Hebboul, M., & Benabderrahmane, O. (2022). Web analytics tools for e-commerce: An overview and comparative analysis. In *International conference on managing business through web analytics* (pp. 51–72). Springer International Publishing, http://dx.doi.org/10.1007/978-3-031-06971-0_5.

Brahmia, Z., Brahmia, S., Grandi, F., & Bouaziz, R. (2022). JUpdate: A JSON update language. *Electronics*, *11*(4), 508. http://dx.doi.org/10.3390/electronics11040508.

Cahyanto, K. A., Hilmi, M. A. A., & Mustamiin, M. (2022). Pengujian rule-based pada dataset log server menggunakan support vector machine berbasis linear discriminat analysis untuk deteksi malicious activity. *Jurnal Teknologi Informasi dan Ilmu Komputer*, *9*(2), 245. http://dx.doi.org/10.25126/jtiik.2022924107.

Canay, O., & Kocabicak, U. (2023). An innovative data collection method to eliminate the preprocessing phase in web usage mining. *Engineering Science and Technology, an International Journal*, *40*, 1–14. http://dx.doi.org/10.1016/j.jestch.2023.101360.

Canay, O., Meric, S., Evirgen, H., & Varan, M. (2011). Realization of campus automation web information system in context of service unity architecture. In *International symposium on computing in science & engineering (ISCSE)* (pp. 173—179).

Cegan, L., & Filip, P. (2017). Webalyt: Open web analytics platform. In *2017 27th international conference radioelektronika (RADIOELEKTRONIKA)* (pp. 1–5). IEEE, http://dx.doi.org/10.1109/radioelek.2017.7937605.

Chan-Jong-Chu, K., Islam, T., Exposito, M. M., Sheombar, S., Valladares, C., Philippot, O., Grua, E. M., & Malavolta, I. (2020). Investigating the correlation between performance scores and energy consumption of mobile web apps. In *Proceedings of the evaluation and assessment in software engineering* (pp. 190—199). ACM, http://dx.doi.org/10.1145/3383219.3383239.

Cheng, Y.-J., & Chen, K.-H. (2022). Website analytics for government user behavior during COVID-19 pandemic. *Aslib Journal of Information Management*, *75*(1), 90–111. http://dx.doi.org/10.1108/ajim-11-2021-0329.

De-Groot, R., Golumbic, Y. N., Martínez, F. M., Hoppe, H. U., & Reynolds, S. (2022). Developing a framework for investigating citizen science through a combination of web analytics and social science methods—the CS track perspective. *Frontiers in Research Metrics and Analytics*, *7*, http://dx.doi.org/10.3389/frma.2022.988544.

Demirkan, H., & Delen, D. (2013). Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud. *Decision Support Systems*, [ISSN: 0167-9236] *55*(1), 412–421. http://dx.doi.org/10.1016/j.dss.2012.05.048.

Dewi, M. J., & Nurdin, N. (2023). Analisis performa platform sosial media menggunakan perbandingan software automated testing. *Information Management for Educators and Professionals*, *7*(2), 164–173. http://dx.doi.org/10.51211/imbi.v7i2.2343.

Durántez, A. B. (2023). Implications of zero party data on user decision-making in digital advertising. *Behanomics*, *1*, http://dx.doi.org/10.55223/bej.7.

Everest-Phillips, M. (2019). Big data-driven public service in the twenty-first century: The politics of big data. In *Public service excellence in the 21st century* (pp. 275–318). Springer Singapore, http://dx.doi.org/10.1007/978-981-13-3215-9_9.

Fundingsland, E. L. J., Fike, J., Calvano, J., Beach, J., Lai, D., & He, S. (2022). Methodological guidelines for systematic assessments of health care websites using web analytics: Tutorial. *Journal of Medical Internet Research*, *24*(4), Article e28291. http://dx.doi.org/10.2196/28291.

Gamalielsson, J., Lundell, B., Butler, S., Brax, C., Persson, T., Mattsson, A., Gustavsson, T., Feist, J., & Lönroth, E. (2021). Towards open government through open source software for web analytics: The case of matomo. *JeDEM - eJournal of eDemocracy and Open Government*, *13*(2), 133–153. http://dx.doi.org/10.29379/jedem.v13i2.650.

Haaksma, T. R., de Jong, M. D. T., & Karreman, J. (2018). Users' personal conceptions of usability and user experience of electronic and software products. *IEEE Transactions on Professional Communication*, *61*(2), 116–132. http://dx.doi.org/10.1109/tpc.2018.2795398.

Harika, B., & Sudha, T. (2019). Extraction of knowledge from web server logs using web usage mining. *Asian Journal of Computer Science and Technology*, *8*(S3), 12–15. http://dx.doi.org/10.51983/ajcst-2019.8.s3.2113.

Hasan, S. B. (2022). Web analyzer software tool in order to identify an know the behavior of users that visit website. *Academic Journal of Nawroz University*, *11*(4), 75–81. http://dx.doi.org/10.25007/ajnu.v11n4a506.

Heričko, T., Šumak, B., & Brdnik, S. (2021). Towards representative web performance measurements with google lighthouse. In *Proceedings of the 2021 7th student computer science research conference (StuCoSReC)* (pp. 39–42). University of Maribor Press, http://dx.doi.org/10.18690/978-961-286-516-0.

Hossfeld, T., Metzger, F., & Rossi, D. (2018). Speed index: Relating the industrial standard for user perceived web performance to web QoE. In *2018 tenth international conference on quality of multimedia experience (QoMEX)* (pp. 1–6). IEEE, http://dx.doi.org/10.1109/qomex.2018.8463430.

Jain, V., & Kashyap, K. L. (2021). An efficient algorithm for web log data preprocessing. In *Lecture notes in electrical engineering* (pp. 505–514). Springer Singapore, http://dx.doi.org/10.1007/978-981-16-5078-9_41.

Jansen, B. J., gyo Jung, S., & Salminen, J. (2022). Measuring user interactions with websites: A comparison of two industry standard analytics approaches using data of 86 websites. In H. Suleman (Ed.), *PLoS One*, *17*(5), Article e0268212. http://dx.doi.org/10.1371/journal.pone.0268212.

Junmei, W., & Jihong, W. (2019). Research on performance automation testing technology based on jmeter. In *2019 International conference on robots & intelligent system (ICRIS)* (pp. 55–58). IEEE, http://dx.doi.org/10.1109/icris.2019.00023.

Keskin, S., & Yazıcı, A. (2022). Modeling and querying fuzzy SOLAP-based framework. *ISPRS International Journal of Geo-Information*, *11*(3), 191. http://dx.doi.org/10.3390/ijgi11030191.

Kousik, N. V., Sivaram, M., Yuvaraj, N., & Mahaveerakannan, R. (2021). Improved density-based learning to cluster for user web log in data mining. In *Inventive computation and information technologies* (pp. 813–830). Springer Singapore, http://dx.doi.org/10.1007/978-981-33-4305-4_59.

Krishnan, A., Das, M., Bendre, M., Yang, H., & Sundaram, H. (2020). Transfer learning via contextual invariants for one-to-many cross-domain recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 1081–1090). Association for Computing Machinery, http://dx.doi.org/10.1145/3397271.3401078.

Król, K., & Zdonek, D. (2020). Analytics maturity models: An overview. *Information*, *11*(3), 142. http://dx.doi.org/10.3390/info11030142.

Kumar, V., & Ogunmola, G. A. (2020). Web analytics for knowledge creation: A systematic review of tools, techniques, and practices. *International Journal of Cyber Behavior, Psychology and Learning*, *10*(1), 1–14. http://dx.doi.org/10.4018/ijcbpl.2020010101.

Kumar, B., Roy, S., Sinha, A., Iwendi, C., & Strážovská, Ľ. (2022). E-commerce website usability analysis using the association rule mining and machine learning algorithm. *Mathematics*, *11*(1), 25. http://dx.doi.org/10.3390/math11010025.

Li, J., Lu, Y., Zhang, Y., Wang, Q., Cheng, Z., Huang, K., & Shu, J. (2022). SwitchTx: scalable in-network coordination for distributed transaction processing. *Proceedings of the VLDB Endowment*, *15*(11), 2881–2894. http://dx.doi.org/10.14778/3551793.3551838.

Maslennikova, A., Rotelli, D., & Monreale, A. (2022). Visual analytics for session-based time-windows identification in virtual learning environments. In *2022 26th international conference information visualisation (IV)* (pp. 251–258). IEEE, http://dx.doi.org/10.1109/iv56949.2022.00050.

Mortazavi, S. H., Salehe, M., Balasubramanian, B., de Lara, E., & PuzhavakathNarayanan, S. (2020). SessionStore: A session-aware datastore for the edge. In *2020 IEEE 4th international conference on fog and edge computing (ICFEC)* (pp. 59–68). IEEE, http://dx.doi.org/10.1109/icfec50348.2020.00014.

Nurshuhada, A., Yusop, R. O. M., Azmi, A., Ismail, S. A., Sarkan, H. M., & Kama, N. (2019). Enhancing performance aspect in usability guidelines for mobile web application. In *2019 6th international conference on research and innovation in information systems (ICRIIS)* (pp. 1–6). IEEE, http://dx.doi.org/10.1109/icriis48246.2019.9073617.

Ofoeda, J., Boateng, R., & Effah, J. (2019). Application programming interface (API) research. *International Journal of Enterprise Information Systems*, *15*(3), 76–95. http://dx.doi.org/10.4018/ijeis.2019070105.

Onder, I., & Berbekova, A. (2021). Web analytics: more than website performance evaluation? *International Journal of Tourism Cities*, *8*(3), 603–615. http://dx.doi.org/10.1108/ijtc-03-2021-0039.

Pilz, M., Burke, R., Redmond, E., Sauer, M., & Weichelt, B. (2020). JA:2021-23. Ideal tracking and analytics: Incorporating modern toolsets to enhance analysis of user acquisition and behavior on AgInjuryNews.org. *Journal of Agromedicine*, *25*(3), 248–249. http://dx.doi.org/10.1080/1059924x.2020.1765575.

Pourghassemi, B., Sani, A. A., & Chandramowlishwaran, A. (2019). What-if analysis of page load time in web browsers using causal profiling. In *Abstracts of the 2019 SIGMETRICS/performance joint international conference on measurement and modeling of computer systems* (pp. 1–23). ACM, http://dx.doi.org/10.1145/3309697.3331483.

Quintel, D., & Wilson, R. (2020). Analytics and privacy. *Information Technology and Libraries*, *39*(3), http://dx.doi.org/10.6017/ital.v39i3.12219.

Rafiq, U. (2022). Towards understanding analytics in software startups. In *Proceedings of the 5th international workshop on software-intensive business: towards sustainable software business* (pp. 31–38). ACM, http://dx.doi.org/10.1145/3524614.3528632.

Regueiro, C., Seco, I., de Diego, S., Lage, O., & Etxebarria, L. (2021). Privacy-enhancing distributed protocol for data aggregation based on blockchain and homomorphic encryption. *Information Processing & Management*, [ISSN: 0306-4573] *58*(6), Article 102745. http://dx.doi.org/10.1016/j.ipm.2021.102745.

Saif, D., Lung, C.-H., & Matrawy, A. (2021). An early benchmark of quality of experience between HTTP/2 and HTTP/3 using lighthouse. In *ICC 2021 - IEEE international conference on communications* (pp. 1–6). IEEE, http://dx.doi.org/10.1109/icc42927.2021.9500258.

Sakas, D. P., Giannakopoulos, N. T., Reklitis, D. P., & Dasaklis, T. K. (2021). The effects of cryptocurrency trading websites on airlines' advertisement campaigns. *Journal of Theoretical and Applied Electronic Commerce Research*, *16*(7), 3099–3119. http://dx.doi.org/10.3390/jtaer16070169.

Salminen, J., Jung, S.-G., Chowdhury, S., Sengün, S., & Jansen, B. J. (2020). Personas and analytics: A comparative user study of efficiency and effectiveness for a user identification task. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. ACM, http://dx.doi.org/10.1145/3313831.3376770.

Samarasinghe, N., Adhikari, A., Mannan, M., & Youssef, A. (2022). Et tu, brute? Privacy analysis of government websites and mobile apps. In *Proceedings of the ACM web conference 2022* (pp. 564–575). ACM, http://dx.doi.org/10.1145/3485447.3512223.

Stringam, B., & Gerdes, J. (2019). Service gap in hotel website load performance. *International Hospitality Review*, *33*(1), 16–29. http://dx.doi.org/10.1108/ihr-09-2018-0012.

Szalek, K., & Borzemski, L. (2018). Conversion rate gain with web performance optimization. a case study. In *Advances in intelligent systems and computing* (pp. 312–323). Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-99981-4_29.

Tao, Y., Guo, S., Shi, C., & Chu, D. (2020). User behavior analysis by cross-domain log data fusion. *IEEE Access*, *8*, 400–406. http://dx.doi.org/10.1109/access.2019.2961769.

Utz, C., Amft, S., Degeling, M., Holz, T., Fahl, S., & Schaub, F. (2023). Privacy rarely considered: Exploring considerations in the adoption of third-party services by websites. *Proceedings on Privacy Enhancing Technologies*, *2023*(1), 5–28. http://dx.doi.org/10.56553/popets-2023-0002.

Vamsee, C. S., Rakesh, D., Prathyusha, I., Dinesh, B., & Bharathi, C. (2023). Demographic and psychographic customer segmentation for ecommerce applications. In *2023 2nd international conference on applied artificial intelligence and computing (ICAAIC)* (pp. 615–622). http://dx.doi.org/10.1109/ICAAIC56838.2023.10140861.

Vasylyshyn, S., Susukailo, V., Opirskyy, I., Kurii, Y., & Tyshyk, I. (2023). A model of decoy system based on dynamic attributes for cybercrime investigation. *Eastern-European Journal of Enterprise Technologies*, *1*(9 (121)), 6–20. http://dx.doi.org/10.15587/1729-4061.2023.273363.

Walny, J., Frisson, C., West, M., Kosminsky, D., Knudsen, S., Carpendale, S., & Willett, W. (2020). Data changes everything: Challenges and opportunities in data visualization design handoff. *IEEE Transactions on Visualization and Computer Graphics*, *26*(1), 12–22. http://dx.doi.org/10.1109/tvcg.2019.2934538.

Yang, J., Xiu, P., Sun, L., Ying, L., & Muthu, B. (2022). Social media data analytics for business decision making system to competitive analysis. *Information Processing and Management*, *59*(1), Article 102751. http://dx.doi.org/10.1016/j.ipm.2021.102751.

Yu, A., & Benson, T. A. (2021). Dissecting performance of production QUIC. In *Proceedings of the web conference 2021* (pp. 1157—1168). ACM, http://dx.doi.org/10.1145/3442381.3450103.