Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

DOI: 10.37943/BTTB5517

Kassimova Botakoz
Candidate of Technical Sciences, Associate Professor, Department of System Analysis and Management Education
b.kassimova@enu.kz, orcid.org/0000-0001-8589-4645
L.N. Gumilyov Eurasian National University, Kazakhstan
Aliya Zhetpisbayeva
Master's Degree, Senior-Lecturer, Researcher
A.Zhetpisbayeva@astanait.edu.kz, orcid.org/0009-0004-9542-5817
CyberTech Research and Innovation Center, Kazakhstan
Leila Rzayeva
PhD, Director
L.Rzayeva@astanait.edu.kz, orcid.org/0000-0002-3382-4685
CyberTech Research and Innovation Center
Murat Zhakenov
Candidate of Technical Sciences, High Researcher
muratzhakenv@outlook.com, orcid.org/ 0009-0005-9672-4365
Digital Heritage of Eurasia LLP, Kazakhstan
Kurmangali Elzhas
Master of Technical Science, Researcher
k.elzhas@outlook.com, orcid.org/0009-0002-6703-7935
CyberTech Research and Innovation Center
Gul Dzhussupova
Candidate of Technical Sciences, Director, Department of Information Security
G.Jussupova@sci.gov.kz, orcid.org/0009-0009-7205-3666
Ministry of Science and Higher Education of the Republic of Kazakhstan, Kazakhstan

IDENTIFYING USER BEHAVIORAL PATTERNS IN WEB BROWSER ACTIVITY USING THE K-NEAREST NEIGHBORS ALGORITHM

Abstract: The rapid expansion of internet usage has increased the volume of digital traces created by users, making web browsers a valuable source of evidence in digital forensics. Web browsers store user information, including browsing history, search queries, visited domains, cookies, and cache files, which reveal behavioral patterns and decision-making tendencies. However, existing forensic tools focus on manually extracting and visualizing browser artifacts, offering limited support for automated behavioral analysis. This study addresses this issue by proposing an automated method for identifying and analyzing user behavioral patterns based on web browser activity using machine learning techniques. The proposed approach involves extracting and structuring browser history data from personal computers, followed by similarity-based classification using the k-nearest neighbor algorithm. Behavioral characteristics are extracted from recurring search queries, domain frequencies, and content categories of visited websites, enabling the creation of a structured behavioral dataset. To support model selection, the k-nearest neighbors classifier was evaluated alongside logistic regression and a Gaussian naive Bayes classifier using performance metrics, including accuracy, precision, recall, F1-score, and area under the curve. Experimental results show that the k-nearest neighbors model achieves the highest recall and F1-score, demonstrating performance in detecting clustered and nonlinear patterns in user behavior compared to the baseline and probabilistic models. The proposed method enables the transformation of browser data into interpretable user behavior profiles integrating interests, behavioral indicators, and visit frequency statistics. This improves the applicability of

Received: 25.02.2026 Accepted: 17.07.2026 Published: 30.06.2026

DOI: 10.37943/BTTB5517

web browser analysis by supporting structured behavioral assessment rather than the interpretation of raw data. Overall, the study demonstrates the feasibility and effectiveness of integrating machine learning-based behavioral analysis into web browser forensics workflows and lays the foundation for future research that incorporates temporal features and larger datasets.

Keywords: web browser forensics; user behavior analysis; digital forensics; machine learning; k-nearest neighbors; cybersecurity.

Introduction

In recent years, the increased use of the internet by users has led to a significant increase in potential crimes related to digital forensics. One of the primary and most popular ways to access the internet is through the web browser. Consequently, browsers store a large amount of user data, known as browser artifacts, which include cache, cookies, bookmarks, and browsing history. This data plays a key role in web forensics and is therefore crucial for investigations.

Attackers also use the internet to commit illegal activities, thereby leaving digital traces. Such data is crucial for security, especially in digital investigations, where digital data can serve as important evidence [1].

The relevance of the research lies in the development of a new method based on a machine learning algorithm. Unlike existing solutions, which rely on manual data processing, this proposed method uses machine learning to more accurately identify patterns of user behavior online. Since attackers disguise themselves as regular users, analyzing their digital traces can help identify data. Existing research demonstrates that external and internal factors can influence human behavior. Specifically, external factors include patterns such as financial instability, economic crises, and unemployment. Consequently, these economic risks can drive individuals to commit illegal acts in pursuit of quick gains [2]. Internal factors include aspects such as a person's psychological patterns. Taken together, external and internal factors can influence a person's online actions. Accordingly, this study aims to identify and analyze behavioral patterns based on a user's browser history, including repeated search queries and visits to potentially suspicious websites.

Existing programs, such as NirSoft and Oxygen Forensics, provide information about visited websites but lack automated analysis capabilities [3]. The proposed method automatically analyzes web browsing history and identifies patterns in online behavior, including browsing interests and number of visits, to determine user behavior patterns. The feasibility of the method is implemented in Python. Compared to existing solutions, the proposed method supports a wider range of data formats and has increased accuracy in identifying behavior patterns.

The scientific novelty of the study lies in the application of a method based on the k-NN machine learning algorithm to web browser forensics.

Literature review

Overview of Web Browser

A web browser is software that provides users with interfaces for performing actions. A web browser's functionality consists of sending requests from the browser to servers, thereby securely exchanging information using protocols such as HTTPS. The web browser then displays the data to the end user. Different web browsers run on digital devices and display web pages on those devices. When a user requests a web page, the browser internally parses the HTML code into the appropriate structure to display an image or text, along with JavaScript [4].

Since the web browser is the gateway to the Internet, supply and demand have led to the development of various web browsers on the global market. Furthermore, recent research shows that users' dependence on the internet is growing daily, along with the increasing number of online services. As internet usage grows, more and more users strive to fully utilize all its features. Currently, one of the primary ways for various users to access the World Wide Web is through a web browser, which has led to the development of various web browsers to meet demand [5].

According to official browser usage statistics, Google Chrome dominates the web browser market in the CIS countries, including Kazakhstan and Ukraine. Safari and Yandex Browser are

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

also significantly popular among users. The distribution shown in the figure corresponds to publicly available data [6].

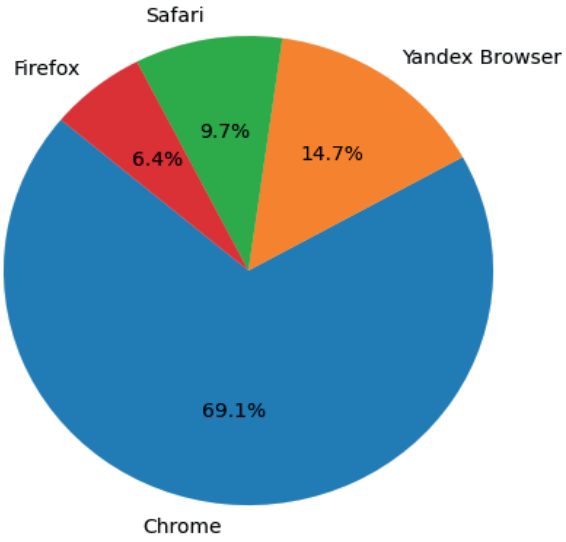

Figure 1. Distribution of web browser usage in the CIS countries.

Despite the existence of different web browsers, their internal architecture of modern web browsers is largely the same. Modern web browsers are primarily built on three rendering engines: Blink, WebKit, and Gecko. Blink is the rendering engine used in Chromium-based browsers such as Google Chrome, Microsoft Edge, Opera, and Brave. WebKit is used in Safari, and Gecko is the primary rendering engine in Mozilla Firefox [7]. While these browsers differ in implementation details, their internal data storage structures remain largely comparable for forensic analysis purposes. In browser forensics, Chromium-based browsers (Blink) are considered the most suitable for analysis because they store browser artifacts, i.e., data, in SQLite databases [8].

Browser data is stored in standard data structures such as databases and configuration files, which contain the history of visited pages, cookies, cache data and bookmarks in Figure 2. Therefore, the similarity of the architecture allows the use of a forensic method regardless of the type of browser [9].

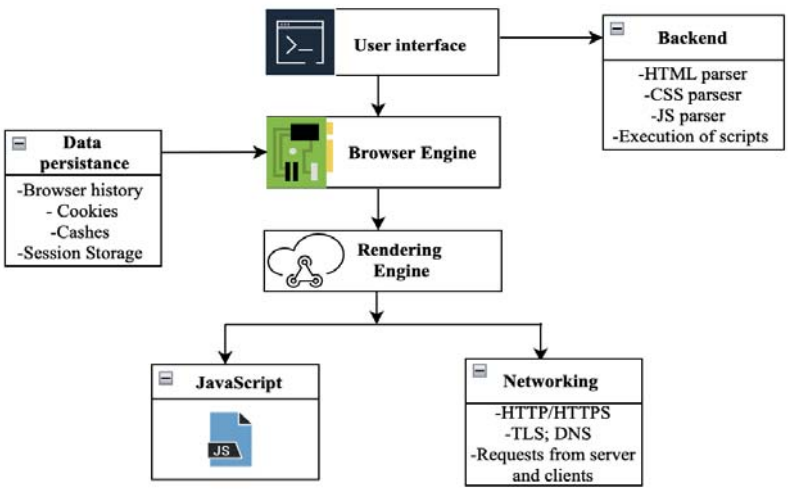

Figure 2. Block-diagram of Browser architecture

DOI: 10.37943/BTTB5517

Figure 2 shows the general architecture of a modern web browser. Although web browsers may differ in implementation details, the internal architecture of their components is similar.

The user interface (UI) acts as the interface between the user and the browser, providing input/output of content. User actions are processed by the browser engine, which is the primary component between the browser's subsystems. The browser engine interacts with the rendering engine, which is responsible for displaying web content by interpreting HTML, CSS, and other components [10].

The UI backend compiles JavaScript and executes scripts. JavaScript modules play a key role in client-side logic and interaction with web pages.

Network communication is managed by the UI backend via protocols such as HTTP and HTTPS, along with mechanisms such as TLS and DNS. This ensures data exchange between the client and servers.

A key component for digital forensic analysis is the storage of data such as browsing history, cookies, cache files, and session storage [11]. These stored data are important digital traces as they are the primary source of data for analyzing behavioral patterns.

Analysis of User Behavioral Patterns in Web Activity

User behavioral analysis is necessary for information and communication technologies and for identifying internal threats.

Overall, user behavior analysis has become an important area of research in cybersecurity and data analytics. User behavior analysis aims to identify patterns based on digital traces online. Specifically, digital data in a web browser includes browsing history, search queries, and interaction frequency. Collectively, this reflects users' core habits and decision-making patterns.

Existing research has shown that user behavior can be influenced by both external and internal factors. External factors include socioeconomic conditions, such as financial instability.

Internal factors relate to psychological and emotional states, which can be indirectly determined by recurring search queries, browsing interests, and activity time [12].

Existing research has shown that both external and internal factors can influence user behavior. External factors include socioeconomic conditions such as financial instability, economic pressure, and unemployment, which can influence decision-making processes and increase the likelihood of risky online behavior. Internal factors relate to psychological and emotional states, which can be indirectly determined by repeated search queries, browsing interests, frequency of interaction, and activity time. Together, these factors shape individual online behavior patterns and can manifest as repeat visits to certain types of websites, changes in browsing intensity, and engagement with potentially suspicious content. Thus, browser activity analysis provides valuable insights into user behavior and helps identify behavioral patterns relevant to cybersecurity and digital forensics [13].

To address these issues, recent research has increasingly utilized machine learning algorithms to analyze and predict user behavior. Distance-based classifiers, probabilistic models, and clustering methods have shown promising results in identifying similarities and deviations in behavior. However, the application of such models to web analytics data for behavior prediction remains limited, highlighting the need for further research in this area [14].

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

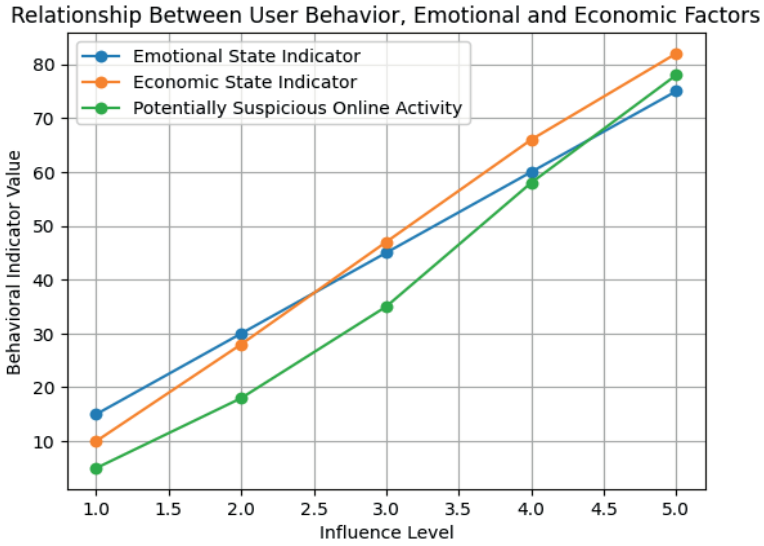

Figure 3. Conceptual relationship between emotional factors, economic factors, online user behavior and detection of suspicious activity.

Figure 3 presents a conceptual relationship showing how emotional and economic factors can influence user behavior online, which in turn can contribute to the detection of suspicious activity when analyzing browser artifacts [15].

k-Nearest Neighbors Algorithm in Behavioral Analysis

The k-nearest neighbors (k-NN) algorithm is a widely used supervised machine learning method for classification and pattern recognition problems. The basic principle of k-NN is based on measuring the similarity between data instances in a high-dimensional feature space. An unlabeled data point is classified according to the majority class among its k nearest neighbors, which are determined using a distance metric, most commonly Euclidean distance [16].

Due to its nonparametric nature, the k-NN algorithm does not require an explicit training step or assumptions about the underlying data distribution. This property makes k-NN particularly suitable for behavioral analysis, where user activity data is often heterogeneous and does not fit predefined statistical models [17].

Previous studies have demonstrated the effectiveness of k-NN in tasks of behavior recognition, anomaly detection, and user profiling. The algorithm has been successfully applied in areas such as intrusion detection systems, fraud detection, and user interaction pattern analysis. Its ability to classify behavior based on similarities allows for intuitive interpretation of results, which is especially important in digital forensics [18].

The k-nearest neighbors (k-NN) algorithm is a supervised machine learning method used for classification and recognition [19]. The basic principle of k-NN is based on similarity: data points are classified according to the classes of their nearest neighbors in the feature space. Instead of building an explicit model during the training phase, the algorithm stores all labeled data and performs classification at query time [20].

Method

Data source and description

The proposed method is based on the analysis of web browser activity records obtained from a personal computer or laptop. The primary data source includes browsing history records such as visited URLs, domain names, search queries, timestamps, and visit-frequency statistics. The collected history covered more than 500 unique visited websites over a one-year observation period. Because the same website could be visited repeatedly, the number of unique websites was smaller than the number of individual activity records. After cleaning, deduplication, and feature extraction, the time-stamped records were converted into behavioral observations representing

DOI: 10.37943/BTTB5517

browsing activity through features such as domain frequency, URL category, search-query category, and visit intensity. For the comparative machine-learning evaluation, 1,500 labeled behavioral observations derived from this browser history were used. Thus, the value of more than 500 refers to unique websites, whereas n = 1,500 refers to the behavioral observations evaluated by the classifiers.

Data preprocessing and quality control

To ensure data completeness and consistency before analysis, the collected browser history data was pre-screened. Records related to duplicate browser visits were removed from the dataset. Since some of the original data was stored in JSON format, an additional parsing procedure was performed to extract relevant fields, such as visited URLs, domain names, timestamps, page titles, and visit frequency metrics. The extracted records were then converted into a unified tabular format suitable for further analytical processing. Numerical variables were standardized to ensure comparability of features across different scales, while categorical browser attributes were coded according to their analytical role. In cases where some records contained missing or incomplete values, these records were excluded during the preprocessing step to maintain the reliability of the dataset. The preprocessing steps improved data consistency, reduced noise, and eliminated artifacts that could negatively impact feature extraction, similarity calculations, and subsequent classification results.

Analytical procedure

After preprocessing steps with dataset, let assume that the user's device have multiple browsers.

$$B = \{b_1 b_2 \dots b_n\}, \tag{1}$$

where each browser b_n stores browsing history in one or more data sources:

$$S_i = \{s_{i1} s_{i2} \dots s_{im}\} b_n\}, \tag{2}$$

The source s_{im} may be:
- Local SQLite database;
- WAL/SHM files;
- Cached files;
- Exported logs;
- Backups;
- Cloud-synced history

Then the complete set of all sources is:

$$S = \bigcup_{i-1}^{n} S_i , \tag{3}$$

The goal of the process is to extract, normalize, merge, and sort browser history entries into a single, organized collection:

$$H^* = \{h_1, h_2, \dots, h_k\}, \tag{4}$$

Each browser history represent entry as a feature vector:

$$h_k = \langle u_k, t_k, \tau_k, v_k, d_k, b_k, s_k, q_k \rangle, \tag{5}$$

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

where:
u_k – URL;
t_k – page title;
τ_k – visit timestamp;
v_k – number of visits or frequency weight;
d_k – session duration or activity time;
$b_k \in B$- browser identifier;
$s_k \in S$- source of extraction;
q_k – indicator of the reliability/integrity of the record.
Then the set of retrieved records is:

$$H = \bigcup_{s \in S} E(s) \tag{6}$$

where E(s) is the operator for extracting data from source s.
Data extraction stage
For each S source, an extraction function is defined:

$$E: S \to 2^H, \tag{7}$$

where 2^H is the set of all subsets of history records.
If the source is a SQLite database, then the extraction can be described as:

$$E(s) = \{h_k \mid h_k \text{ obtained from SQL} - \text{request k s}\}.$$

For example, if the history table has the fields url, title, visit_time, visit_count, then for each row r the following display is generated:

$$\emptyset(r) = \langle u(r), t(r), \tau(r), v(r), d(r), b(r), s, q(r) \rangle, \tag{8}$$

Then:

$$E(s) = \{\emptyset(r_1), \emptyset(r_2), \dots, \emptyset(r_p)\}, \tag{9}$$

Since different browsers use different time formats, fields, and storage structures, a normalization operator is introduced:

$$N: H \to \tilde{H}, \tag{10}$$

where $\tilde{H}$ is a set of records converted to a uniform format.
For each record:

$$\tilde{h} = N(h_k), \tag{11}$$

Normalization of time.
Let $\tau_k^{(raw)}$ – the original timestamp in browser format. Then the normalized time:

$$\tau_k = f_t\left(\tau_k^{(raw)}\right), \tag{12}$$

DOI: 10.37943/BTTB5517

where f_t is the function for converting time into a uniform format, for example Unix Time:

$$f_t = T_{raw} \to \mathbb{R}, \tag{13}$$

Cleaning and filtering records.
After extraction and normalization, filtering is performed:

$$F: \tilde{H} \to \hat{H}, \tag{14}$$

This removes:
• incomplete records;
• corrupted records;
• system URLs;
• duplicates;
• records with invalid timestamps.

$$\hat{H} = \{\tilde{h}_k \in \tilde{H} \mid P(\tilde{h}_k) = 1\}, \tag{15}$$

Where $P(\tilde{h}_k)$— predicate of correctness of the record.
Example:

$$P(\tilde{h}_k) = \begin{cases} 1, \text{if } u_k \neq \underline{\quad} \in [\tau_{min}\tau_{max}] \\ \wedge\tau_k \\ 0, \text{else} \end{cases} \tag{16}$$

Record deduplication.
Since the same record can be retrieved from multiple sources, an equivalence relation is introduced:

$$h_a \sim h_b \Leftrightarrow p(h_a, h_b) \geq \theta, \tag{17}$$

Where $p(h_a, h_b) -$ measure of similarity between two records;
$\theta -$ equivalence threshold.
The simplest measure:

$$p(h_a, h_b) = \begin{cases} 1, \text{if } u_a = u_b \wedge |\tau_a - \tau_b| < \Delta\tau \\ 0, \text{else} \end{cases}, \tag{18}$$

After deduplication, the result is a set:

$$H' = \frac{\hat{H}}{\sim}, \tag{19}$$

or, equivalently, one representative is selected from each equivalence class.
Sorting history
The basic sorting operation is defined as a mapping:

$$\boldsymbol{\Pi}: H' \to H^*, \tag{18}$$

where H* is a linearly ordered sequence of records.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

Chronological sorting
If sorting is done by time of visit:

$$H^* = \text{sort}(H', \tau, \uparrow) \tag{20}$$

or

$$H^* = \text{sort}(H', \tau, \downarrow) \tag{21}$$

where ↑ is in ascending order of time, ↓ is in descending order.
Algorithmic interpretation
In algorithmic form:
1. Define a set of browsers B and sources S.
2. For each s∈S, extract records E(s).
3. Convert records to a uniform format N.
4. Remove incorrect and irrelevant records F.
5. Perform deduplication D.
6. Sort records by the selected criterion Π.
7. If necessary, group into sessions Σ.

$$H^* = \Pi\left(D(F)(N)\left(\bigcup_{s\in S} E(s)\right)\right), \tag{22}$$

The process of extracting and sorting a user's browser history is formalized as a sequence of operators over a set of digital device data sources. Let S be the set of file, log, and database sources containing browsing history artifacts. Then the resulting set of history records H* is defined as the result of a composition of extraction, normalization, filtering, deduplication, and sorting operators:

$$H^* = \Pi\left(D(F)(N)\left(\bigcup_{s\in S} E(s)\right)\right), \tag{23}$$

Each history entry is described by a tuple of attributes ⟨u,t,τ,v,d,b,s,q⟩, where u is the URL, t is the page title, τ is the timestamp, v is the number of visits, d is the activity duration, b is the browser identifier, s is the entry's source, and q is the confidence coefficient. History sorting is performed by timestamp or a multi-criteria ranking function.

The extracted browser features and classification results were further organized into comparative tables and visualized using charts and graphs generated in Python. Plots were used to illustrate the distribution of behavioral categories, comparative model performance, and relationships based on the features identified during the analysis. All analytical procedures were performed in a separate environment using the core libraries pandas, NumPy, matplotlib, and scikit-learn. Browser history data was processed in anonymized form, and any personally identifiable information, including user IDs, account-related data, and sensitive browsing metadata, was excluded from publication to ensure confidentiality and adherence to ethical data processing principles.

The proposed method shows in Figure 5 is based on analyzing web browsing history using data extracted from a personal computer or laptop. Initially, web browser data is collected and analyzed to obtain structured data. The extracted data is then divided into key components such as search queries, visited domains, and URLs.

DOI: 10.37943/BTTB5517

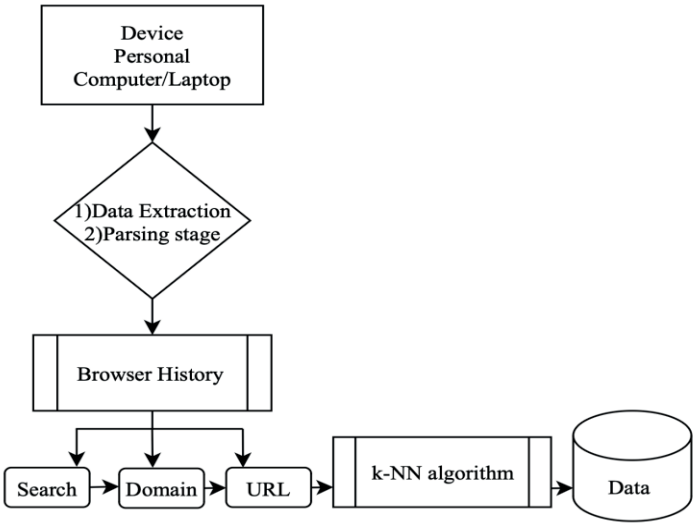

Figure 5. Proposed method for analyzing user behavior based on web browser history using the k-Nearest Neighbors (k-NN) algorithm.

These components are used to form a behavioral dataset, which is subsequently processed using the k-nearest neighbors (k-NN) algorithm. The k-NN classifier identifies similarities between user browsing patterns and previously analyzed data, enabling the detection and classification of user behavior patterns. The result of the method is a structured data that can be used for further forensic analysis. Machine learning algorithm such as K-NN are widely used for internet traffic classification.

The k-nearest neighbors (K-NN) method is used in behavioral pattern recognition tasks because it can classify data based on similarity. Similarity-based classification allows for the identification of browsing patterns by comparing new activity with behavioral data from browser history. Therefore, in this study, the K-NN algorithm is used as the primary analytical approach for further digital forensic analysis.

Results

These mathematical expressions formalize the feature extraction, normalization, classification, and evaluation stages of the proposed browser forensics framework and make the methodology more rigorous for scientific presentation.

To support the selection of a classification model, several machine learning algorithms were evaluated. Logistic regression was used as the linear baseline model, and k-nearest neighbors ($k = 5$, distance-weighted) was used as the similarity-based classifier. A Gaussian Naive Bayes classifier was considered as the probabilistic classifier. This comparative evaluation provided an objective basis for selecting the most suitable algorithm for analyzing user behavior patterns. A total of 1,500 labeled behavioral observations derived from the collected browser history were used as the common evaluation dataset for all three classifiers. The observations were represented by the same standardized browser-activity features, and the original class proportions were preserved during evaluation. Model performance was assessed using accuracy, precision, recall, F1-score, and ROC-AUC. The results are presented in Table 1.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

Table 1. Model Comparison on the Browser Behavior Dataset (Evaluation Set, n = 1500)

Algorithm	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.9513	0.9448	0.9587	0.9517	0.9868
K-Nearest Neighbors (k = 5)	0.9533	0.9416	0.9667	0.9539	0.9823
Gaussian Naive Bayes	0.8460	0.9361	0.7427	0.8283	0.9450

As shown in Table 1, all evaluated models demonstrate strong classification performance on the browser behavior dataset. Logistic regression provides strong baseline results, indicating that part of the selected feature space is linearly separable. The k-nearest neighbors model (k = 5) achieves the highest F1-score (0.9539) and recall (0.9667), indicating its effectiveness in identifying the predefined user behavior patterns. Although the Gaussian Naive Bayes classifier maintains relatively high precision, its lower recall and F1-score indicate limitations associated with the assumption of independence between behavioral features. Based on these results, the k-nearest neighbors algorithm was selected because of its balanced performance and suitability for similarity-based behavior classification [16].

The experimental implementation was performed in Python using the scikit-learn machine learning library. A k-nearest neighbors classifier was implemented with k = 5 neighbors using distance-based weighting to improve classification sensitivity to the local feature distribution. To ensure evaluation consistency, logistic regression and Gaussian naive Bayes classifiers were implemented using the same library. Experiments were conducted in Jupyter Notebook on a standard personal computer with an Intel Core processor and 16 GB of RAM. Feature vectors were generated based on browser history attributes, including visited domains, URL frequency, and search query categories, and standardized before model evaluation.

Figure 6 presents the correlation matrix of the extracted browser behavior features. The heatmap illustrates the strength of pairwise relationships between the selected quantitative indicators derived from browser history records. Stronger positive correlations indicate that some behavioral features tend to increase together, while weaker correlations suggest a lower level of dependence between them.

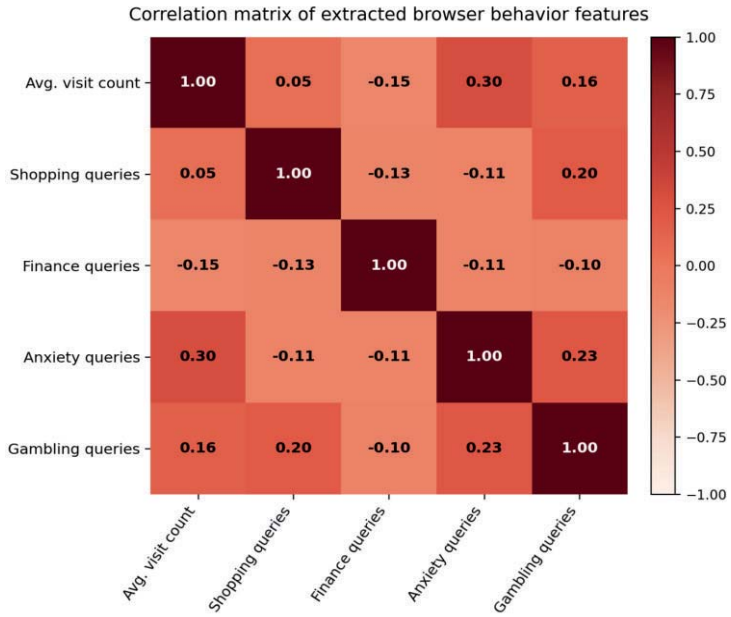

Figure 6. Correlation matrix of extracted browser behavior features

DOI: 10.37943/BTTB5517

This visualization is useful for identifying the most closely related browsing characteristics and for understanding the internal structure of the feature space used for classification. The observed relationships confirm that the extracted browser activity indicators are not random, but form meaningful behavioral patterns suitable for analytical modeling.

Figure 7 illustrates the relationship between anxiety-related queries and total daily search activity based on the extracted browser history data. The scatter plot shows the distribution of observations and the general trend between the selected variables. The fitted trend line indicates that an increase in anxiety-related queries is generally associated with an increase in overall search activity, although the distribution remains heterogeneous and includes several dispersed points.

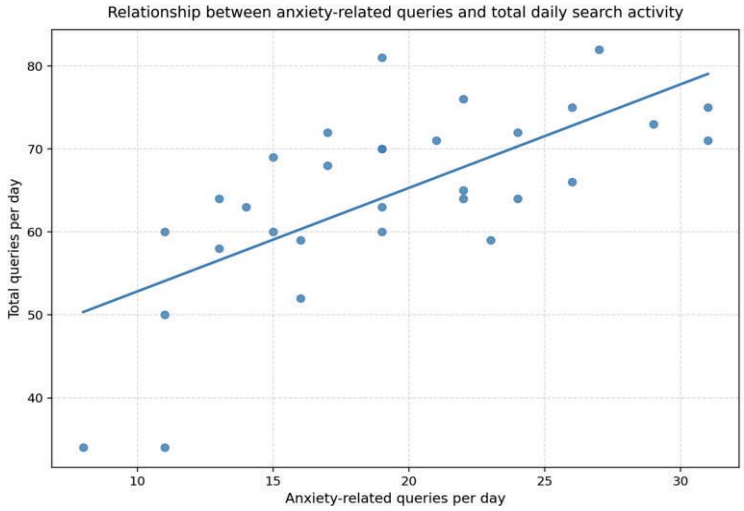

Figure 7. Relationship between anxiety-related queries and total daily search activity.

This result suggests that emotionally significant or behavior-specific search patterns may be reflected in broader browsing activity and can therefore serve as informative indicators for behavioral profiling and classification.

A method for integrating a machine learning algorithm such as k-nearest neighbors (k-NN). This algorithm was used primarily because the k-nearest neighbors' algorithm is suitable for classification problems. The principle of this algorithm is to find the "k" closest points to a given input value and make a prediction based on the classification of the majority. One of the main advantages of this algorithm is its use for classification, which is an important aspect when creating a user behavior profile. Using the k-nearest neighbors (k-NN) algorithm, it becomes possible to observe patterns in user activity in the browser based on visited domains, URLs, and search queries, as shown in the Figure 8. Another justification for using this algorithm in this research is that k-nearest neighbors (k-NN) is a robust machine learning algorithm. This is because it classifies new data by comparing it to an existing training dataset, thereby identifying the most similar patterns.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

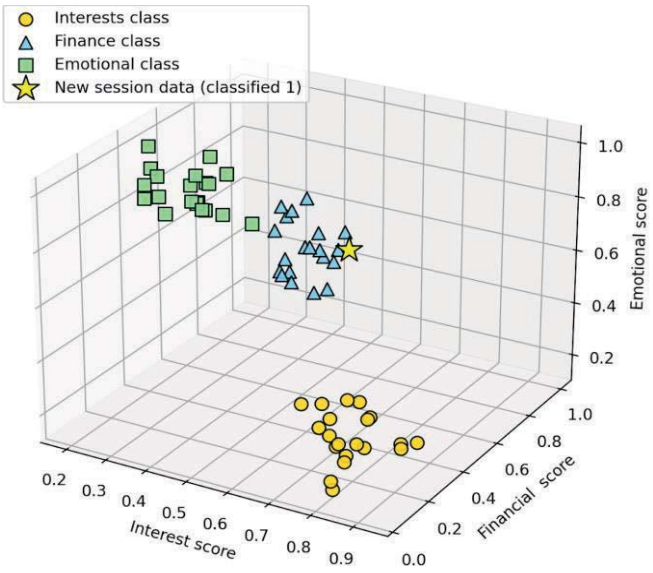

Figure 8. User behavior classification using the k-NN algorithm in the feature space.

Figure 8 illustrates the principle of classifying user behavior based on similarity in a multidimensional feature space. Each user session is represented as a feature vector and classified using the k-nearest neighbors algorithm based on its proximity to previously observed behavior patterns.

As shown in Figure 9, the proposed method enables the creation of a structured user profile by integrating browsing history analysis with behavioral indicators. The results demonstrate how raw browser data can be transformed into interpretable user behavior attributes, supporting further forensic or analytical research.

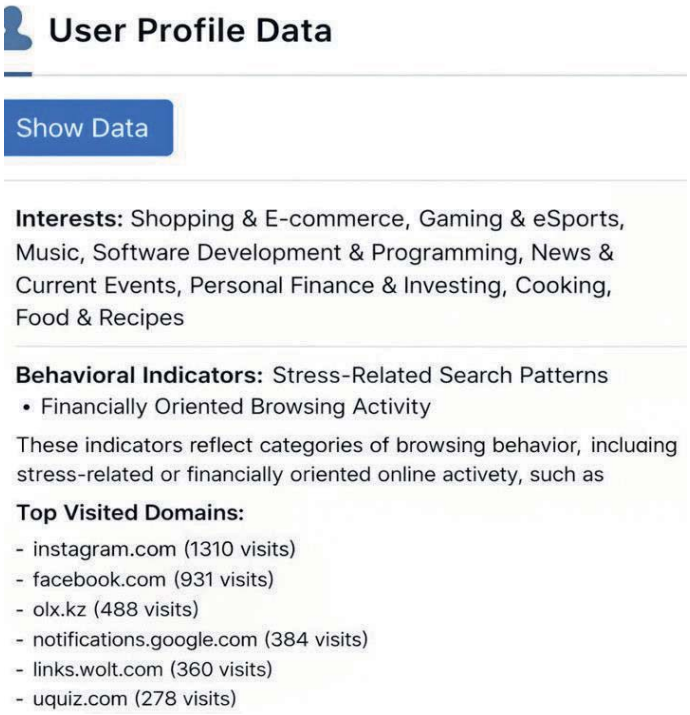

Figure 9. Output of the proposed method in the form of a structured user behavior profile.

DOI: 10.37943/BTTB5517

The proposed approach transforms raw browser history data into an interpretable user behavior profile. The resulting profile combines interest categories, behavioral indicators, and visitor frequency statistics, demonstrating the method's effectiveness for behavioral and forensic analysis.

The profile aggregates web browsing data into thematic interest categories, behavioral indicators, and frequently visited domains. The identified interests are determined based on the content categories of the visited URLs, while the behavioral indicators reflect browsing patterns related to stress and financial orientation, which are inferred from the user's activity. The list of frequently visited domains provides quantitative confirmation of the extracted behavioral characteristics and illustrates the practical applicability of the proposed method.

Discussion

This study examined the effectiveness and accuracy of similarity-based machine learning for analyzing user behavior patterns derived from web browser activity. The results confirm that behavioral characteristics extracted from web browser activity can be successfully used to classify user behavior patterns.

A comparative evaluation shows that logistic regression and k-nearest neighbors demonstrated consistently high results across the reported metrics. Logistic regression provided a strong baseline, indicating partial linear separability of the URL-, domain-, and query-based feature space. However, the k-nearest neighbors model ($k = 5$, distance-weighted) achieved the highest F1-score and recall, demonstrating an improved ability to identify the predefined browser behavior patterns. The superior performance of k-nearest neighbors can be attributed to the clustered nature of browsing activity: similar user actions tend to form localized groups in a multidimensional feature space, making similarity-based classification effective. This observation is consistent with previous research on user behavior modeling and supports the suitability of k-NN for behavioral and forensic analysis tasks. The Gaussian Naive Bayes classifier showed lower recall and F1-score values despite maintaining high precision. This limitation is likely related to its conditional-independence assumption, which is rarely satisfied in real-world browser activity data because behavioral characteristics are often correlated.

In addition to quantitative metrics, the proposed approach enables the creation of interpretable user behavior profiles by aggregating raw browser data into meaningful behavioral indicators. This enhances the practical applicability of the method in digital forensics by facilitating structured analysis of user activity. Future research will focus on incorporating temporal features, evaluating the approach on larger datasets, and exploring hybrid or ensemble models to improve scalability and robustness.

Conclusion

This study addressed the problem of analyzing and identifying user behavior patterns based on web browser activity in the context of digital forensics. Given the increasing role of web browsers as the primary means of internet access and the growing volume of user-generated browser data, the development of an automated method is necessary.

The proposed method demonstrated that web browser data, including browsing history, visited domains, and search queries, can be effectively transformed into structured behavioral characteristics and analyzed using machine learning methods. Using the k-nearest neighbors (k-NN) algorithm, we were able to classify user behavior patterns based on similarity, enabling the identification of recurring interests, behavioral indicators, and potentially suspicious activity. Experimental results confirmed that the k-NN model provides high classification performance and balanced accuracy compared to baseline and probabilistic classifiers.

This expands the practical applicability of browser analysis by supporting structured behavioral assessment rather than raw data analysis. The obtained results confirm that k-NN-based classification is well suited for behavior analysis, particularly in scenarios where user activity exhibits clustered and nonlinear characteristics. More broadly, the obtained results contribute to the field of web browser analysis by demonstrating the feasibility of integrating machine learning

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 26, JUNE 2026

methods into behavior analysis workflows. The proposed methodology can be applied regardless of the specific browser due to the similarity of modern browser architectures, increasing its adaptability to real-world studies.

Future research will focus on extending the method to include temporal and contextual features, evaluating its performance on larger and more diverse datasets, and exploring hybrid or ensemble learning approaches to improve robustness.

Acknowledgments

This study was carried out with the financial support of the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan under Contract 388/PTF24-26 dated 01.10.2024 under the scientific project IRN BR24993232 “Development of innovative technologies for conducting digital forensic investigations using intelligent software-hardware complexes”.

References

[1] Akintola, G. B. (2024). *Performance evaluation of four different forensic tools for web browser analysis.* International Journal of Scientific Research in Multidisciplinary Studies, 10(10), 68–82. Retrieved from International Journal of Scientific Research in Multidisciplinary Studies website:http://isroset.org/journal/IJSRMS/full_paper_view.php?paper_id=3655

[2] Rasool, A., & Jalil, Z. (2020). *A review of web browser forensic analysis tools and techniques.*Researchpedia Journal of Computing, 1(1), 15–21. Retrieved from Researchgate website:https://www.researchgate.net/publication/342338294_A_Review_of_Web_Browser_Forensic_Analysis_Tools_and_Techniques

[3] Majeti, G., Yvl, S. S., Ulichi, S. S., Mohanty, S. N., & S, S., V. (2023). Digital Forensic advanced evidence collection and analysis of web browser activity. ICST Transactions on Scalable Information Systems. https://doi.org/10.4108/eetsis.3357

[4] Alotibi, A. M., Altaleedi, S. Y., Zia, T., & Qazi, E. U. H. (2024). Examining the behavior of web browsers using popular forensic tools. International Journal of Digital Crime and Forensics, 16(1), 1–22. https://doi.org/10.4018/ijdcf.349218

[5] Laperdrix, P., Bielova, N., Baudry, B., & Avoine, G. (2020). Browser fingerprinting. ACM Transactions on the Web, 14(2), 1–33. https://doi.org/10.1145/3386040

[6] StatCounter GlobalStats. (2026). *Browser market share.* Retrieved from StatCounter GlobalStats website:https://gs.statcounter.com/browser-market-share

[7] Chromium Project. (2026). *Multi-process architecture. Chromium Design Documents.* Retrieved from Chromium Project website: https://www.chromium.org/developers/design-documents/multi-process-architecture

[8] Berham, S., & Morris, S. (2022). A CRITICAL COMPARISON OF BRAVE BROWSER AND GOOGLE CHROME FORENSIC ARTEFACTS. □the □Journal of Digital Forensics, Security and Law. https://doi.org/10.15394/jdfsl.2022.1752

[9] Pau, K. N., Lee, V. W. Q., Ooi, S. Y., & Pang, Y. H. (2023). The development of a data collection and browser fingerprinting system. Sensors, 23(6), 3087. https://doi.org/10.3390/s23063087

[10] García, B., Ricca, F., Del Alamo, J. M., & Leotta, M. (2023). Enhancing Web Applications Observability through Instrumented Automated Browsers. Journal of Systems and Software, 203, 111723. https://doi.org/10.1016/j.jss.2023.111723

[11] Chand, R. R., Sharma, N. A., & Kabir, M. A. (2025). Advancing Web Browser Forensics: Critical evaluation of emerging tools and techniques. SN Computer Science, 6(4). https://doi.org/10.1007/s42979-025-03921-6

[12] Nuswantara, D. A., & Maulidi, A. (2020). Psychological factors: self- and circumstances-caused fraud triggers. Journal of Financial Crime, 28(1), 228–243. https://doi.org/10.1108/jfc-05-2020-0086

DOI: 10.37943/BTTB5517

[13] Jawadi, F., Mallick, S. K., Cheffou, A. I., & Augustine, A. (2019). Does higher unemployment lead to greater criminality? Revisiting the debate over the business cycle. Journal of Economic Behavior & Organization, 182, 448–471. https://doi.org/10.1016/j.jebo.2019.03.025

[14] Pang, G., Shen, C., Cao, L., & Van Den Hengel, A. (2021). Deep learning for anomaly detection. ACM Computing Surveys, 54(2), 1–38. https://doi.org/10.1145/3439950

[15] Verma, G., Sarkar, M., & Seth, D. (2022). Visualization of online social dynamics for forensic investigation of user's behavior. In Lecture notes in networks and systems (pp. 173–186). https://doi.org/10.1007/978-981-16-7118-0_16

[16] Syriopoulos, P. K., Kalampalikis, N. G., Kotsiantis, S. B., & Vrahatis, M. N. (2023). kNN Classification: a review. Annals of Mathematics and Artificial Intelligence, 93(1), 43–75. https://doi.org/10.1007/s10472-023-09882-x

[17] Rzayeva, L., Zhetpisbayeva, A., Batkuldin, A., Nyssanov, N., Ryzhova, A., & Saeed, F. (2026). An intelligent browser history forensics method for automated analysis of web activity logs, credentials, and user behavioral profiles. Algorithms, 19(1), 75. https://doi.org/10.3390/a19010075

[18] Yi, J., & Tian, Y. (2024). Insider Threat Detection Model Enhancement Using Hybrid Algorithms between Unsupervised and Supervised Learning. Electronics, 13(5), 973. https://doi.org/10.3390/electronics13050973

[19] Cunningham, P., & Delany, S. J. (2021). K-Nearest Neighbour Classifiers - a tutorial. ACM Computing Surveys, 54(6), 1–25. https://doi.org/10.1145/3459665

[20] Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., & Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. Journal of Big Data, 11(1). https://doi.org/10.1186/s40537-024-00973-y