Journal of Advanced Computer Science & Technology, 13 (1) (2026) 1-10

Journal of Advanced Computer Science & Technology

Website: www.sciencepubco.com/index.php/JACST
https://doi.org/10.14419/786rgw50
Research paper

Cybersecurity-Driven Machine Learning Approaches for The Web Browser Digital Forensics: A Comparative Analysis Of Classification Performances on Browser Artifact Data

Richard Nkrumah [1] *, Edwin Mends-Brew [1], Asante Michael [2], Yeboah Andrews Murphy [1], Osei Antwi [1] , Angela Nkrumah [3], Henry Amoako Mante [4]

[1] Department of Applied Mathematics and Statistics, Accra Technical University
[2] Department of Computer Science, Kwame Nkrumah University of Science and Technology
[3] Ghana Health Service
[4] Accra Institute of Technology
**Corresponding author E-mail: rnkrumah@atu.edu.gh*

Received: November 19, 2025, Accepted: February 9, 2026, Published: February 18, 2026

Abstract

Selecting a machine learning algorithm with optimal precision, accuracy, and recall has been a major challenge in cybersecurity and digital forensic analysis. Common challenges include difficulty in impact visualization, deterioration in efficiency when datasets are large, the discretized nature of datasets, complex relationships among variables, linearity assumptions, overfitting, and other related issues. In an attempt to mitigate these challenges in practice, this study aims to compare the classification performance of machine learning algorithms applied to web browser extracts in digital forensics. Consequently, the most efficient algorithm is proposed for forensic analysis. The study utilized data from 20 computers, each installed with web browsers including Microsoft Internet Explorer, Microsoft Edge, Google Chrome, Mozilla Firefox, and Opera. Browser extracts were obtained using the Web Browser Forensic Analyzer (WEFA) tool (version 1.2). Browser artefacts were extracted and categorized into history, cache, cookies, typed URLs, sessions, most visited sites, screenshots, downloaded files, favorites, bookmarks, and thumbnails. The dataset consisted of counts of artefacts extracted from the browsers. Data collection was further supported by Firefox Forensic Analyzer and Google Chrome Analyzer tools. The Python programming language was used as the primary tool for implementing and evaluating the performance of the machine learning algorithms. During the implementation process, the study assessed the performance of the Linear Discriminant Algorithm against five competing classification algorithms: Logistic Regression, Decision Tree Classifier, K-Nearest Neighbors, Naive Bayes Classifier, and Support Vector Classifier. The findings revealed that the Linear Discriminant Algorithm outperformed the competing algorithms in terms of accuracy, precision, recall, and F1-score. The study therefore concludes that the Linear Discriminant Algorithm is an enhanced and effective approach for classifying browser extracts (artefacts) in digital forensic investigations.

Keywords: *Cybersecurity; Machine Learning; Linear Discriminant Analysis; Digital Forensics; Browser Artefacts.*

1. Introduction

Internet browsers are used to access the internet. People use web browsers to access social networking sites, perform internet banking, send and receive emails, among other online activities. During browser usage, malicious users may attempt to steal confidential and sensitive information such as email addresses, social security numbers, user address books, and banking credentials. Some attackers go further to hack into systems using retrieved credentials to gain unauthorized access to organizational assets or to blackmail individuals. Due to the complexity of browser activities, digital forensic examiners need to understand browser usage patterns and internet dynamics. Forensic examiners can acquire skills in collecting and analyzing browser artefacts to provide valuable evidence in cases related to cybercrime. Web browsers are computer programs primarily used to communicate with web servers and are also referred to as web clients. They communicate with servers by sending user requests and receiving responses from the servers [1].
The following sections review some popular web browsers and their significant properties.
Microsoft Internet Explorer: Microsoft Internet Explorer was first introduced in the 1990s and officially launched in 1995. It became one of the most widely used browsers for internet access [2]. Internet Explorer version 1.0 was released in July 1995, and Microsoft developed a total of eleven versions between 1995 and 2013.

Microsoft Edge: Microsoft Edge is a modern browser developed by Microsoft to replace Internet Explorer. It was launched in 2015 and is a cross-platform browser compatible with most operating systems. Initially, Edge was available only on Windows 10 and Xbox, but versions for other platforms were released in 2017, 2019, and 2020, respectively. Edge is a lightweight and powerful browser and serves as the default browser for Windows 10 and Xbox devices. Its features include a combined search and address bar, forward and backward navigation buttons, bookmarks, favorites, and settings tools. Edge offers several advantages, including lower memory usage compared to Chrome, text-to-speech functionality, PDF reading capabilities, a privacy mode, and support for Chrome extensions [3].
Opera Browser: The Opera browser project began in 1994 at Telenor, a telecommunications conglomerate in Norway. In 1995, the project became independent and was named Opera Software ASA. The first public version of Opera was released in 1996 and was known as MultiTorg Opera. In 2013, Opera discontinued its proprietary rendering engine, Presto, and the browser was fully rewritten in subsequent versions.
Mozilla Firefox: Mozilla Firefox is an open-source web browser developed by the Mozilla Foundation. It was released on September 23, 2002. The browser was developed by Mozilla, Mozilla Corporation, Mozilla Foundation, Netscape, StatCounter, Blake Ross, and Ben Goodger. Firefox is cross-platform and compatible with all major operating systems [4].
Google Chrome: Google Chrome is a cross-platform browser initially developed for Windows. Later versions were released for Linux, macOS, Android, and IOS. Chrome is widely known for its dominance in the browser market, accounting for approximately 70% of global market share as of November 2020 [5]. Key features of Chrome include bookmarks, enhanced security and privacy, speed, stability, and extensive extension support.
Web browser extracts, such as browsing history, cookies, bookmarks, and cache files, constitute major sources of digital evidence in forensic investigations. Web activities are stored in specific locations on computer systems, enabling forensic investigators to analyze these artefacts and draw conclusions related to digital crimes [5]. Several studies, including [2] and [6], have focused on using limited information sources for classifying and analyzing browser extracts. However, the use of minimal information sources does not adequately represent the classification process. Furthermore, many existing algorithms fail to account for data variability and extreme values, resulting in poor classification performance [5], [7]. Another limitation in forensic analysis is the reliance on traditional, non-classification Gaussian models, which often overlook discriminant effects essential for accurate classification [1], [7].
Additionally, forensic investigations frequently prioritize determining whether a crime has occurred, rather than identifying and linking classifiable digital traces. The objectives of this study include the collection of digital browser extracts using the Web Browser Forensic Analyzer (WEFA), evaluation of algorithm performance in classifying computers with similar artefacts, assessment of algorithm effectiveness in classifying browsers with similar artefacts, and analysis of linked Internet Protocols, as well as booted and unbooted computers before confiscation.
Machine Learning Algorithms: This study examines several machine learning algorithms commonly applied in classification tasks, with particular attention to their suitability for browser artefact classification in digital forensic investigations. The discussion focuses on algorithmic behavior under conditions typical of forensic data, including limited sample size, high-dimensional feature spaces, and the need for interpretability.
Distance-based methods such as K-Nearest Neighbor (KNN) offer simplicity and flexibility, making them useful for exploratory forensic analysis. However, their performance is sensitive to the choice of distance metric and the number of neighbors, and they scale poorly with increasing dimensionality and dataset size. These limitations reduce their robustness in practical forensic deployments.
Logistic Regression provides interpretability and computational efficiency, which are desirable in forensic contexts. Nevertheless, its reliance on linear decision boundaries restricts its ability to model complex relationships among browser artefacts. This constraint becomes more pronounced when the number of features is large relative to the number of observations.
The Linear Discriminant Algorithm (LDA) demonstrates strong theoretical and empirical suitability for forensic classification tasks [7]. By maximizing between-class variance while minimizing within-class variance, LDA effectively exploits discriminative structure within the artefact features. Although LDA assumes approximate normality and is sensitive to outliers, controlled experimental conditions and preprocessing mitigate these effects. The strong performance of LDA observed in this study aligns with its documented success in other high-dimensional classification domains.
Decision Tree Classifiers offer transparency and ease of interpretation, which are critical for forensic evidence presentation. However, they are highly susceptible to overfitting in small datasets and can exhibit instability due to minor data variations. Without ensemble enhancement, their predictive reliability remains limited [7].
Support Vector Machine Classifiers (SVMC) are well-suited for high-dimensional data and can model non-linear decision boundaries through kernel functions. Despite their strong theoretical performance, SVMCs present challenges related to kernel selection, computational cost, and limited interpretability, which constrain their applicability in forensic environments where explainability is essential.
Naïve Bayes Classifiers provide computational efficiency and resistance to overfitting but rely on a strong independence assumption among features. Given the inherent dependencies among browser artefacts, this assumption is often violated, limiting classification effectiveness despite the model's simplicity [7].
Modern ensemble methods such as Random Forest and Gradient Boosting have demonstrated strong predictive performance in many machine learning applications due to their ability to capture non-linear feature interactions and reduce variance. However, these methods require larger datasets to avoid overfitting and produce complex models that are less interpretable. In forensic contexts, this reduced transparency may limit evidential admissibility and practical adoption [7].
Similarly, Neural Networks (NN) offer powerful representation learning capabilities but demand substantial data and computational resources. Their black-box nature further limits interpretability, making them less suitable for controlled forensic investigations where traceability and explanation of decisions are critical [10].
Overall, while advanced models such as Random Forests (RF), Gradient Boosting (GB), and Neural Networks (NN) offer strong predictive potential, their data demands and interpretability limitations justify their exclusion from the present study. In contrast, LDA provides a balanced combination of discriminative power, computational efficiency, and interpretability, making it particularly appropriate for browser artefact classification under realistic forensic constraints.

2. Materials and Methods

2.1. Design framework

The study adopts an experimental research design, supported by a quantitative approach. The experimental design was selected because of its scientific rigor and its suitability for hypothesis testing and controlled representation of subjects. Additionally, the quantitative design was employed because some of the elements under investigation are measurable and countable in nature. The conceptual framework in Figure 1 explains the experimental processes. In the conceptual framework proposed, there is an evidence computer and a master computer. This is to ensure that the digital investigations do not interfere with the evidence:

1) The evidence computer houses the web browsers and the extracts.
2) The evidence computer passes the extracts(artefacts) onto the master computer
3) The master computer contains the machine learning algorithm tools
4) The passed-on extracts(artefacts) on the master computer are subjected to analysis based on the machine learning algorithms (LDA, LRC, SVMC, NBC, KNN, DTC, RF, GB, NN)
5) The machine learning algorithms are evaluated, and the classification performance results (Accuracy, Precision, Recall, and F1, Root Mean Square Error (RMSE), Statistical Significance) are reported.

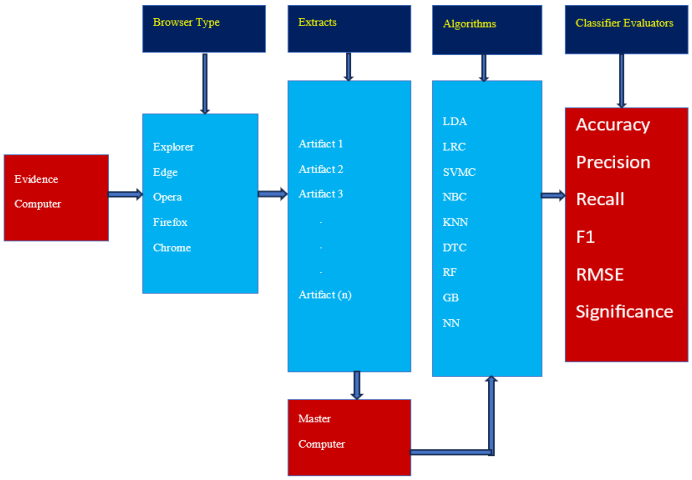

Fig. 1: Proposed Conceptual Framework.

2.2. Computers used and source

Twenty computer devices (20) were used; all 20 devices were HP-brand computers with identical configurations and were selected to ensure consistency and control across experimental conditions. Each computer had the same set of five web browsers installed: Google Chrome, Mozilla Firefox, Opera, Microsoft Internet Explorer, and Microsoft Edge, which allowed for a rigorous and standardized comparative analysis. The primary objective of the study is methodological comparison under realistic forensic conditions, rather than large-scale predictive modeling. Given the legal, ethical, and operational constraints associated with digital forensic investigations, access to additional devices was not possible.

To address dataset limitations, the study explicitly discusses these constraints and incorporates robust mitigation strategies, including cross-validation techniques and comprehensive statistical evaluation, to ensure the reliability and validity of the findings.

2.3. Extraction of artefacts

The Browser artefacts (Figure 2) were extracted using Web Extracts Browser Analyzer (WEBA version 1.2) for downloaded files (pdf, Word, codes), Cache(videos), cache(images), Cache(others), Cookies, Typed URLs, sessions, Most visited sites, Screenshots, and Bookmarks.

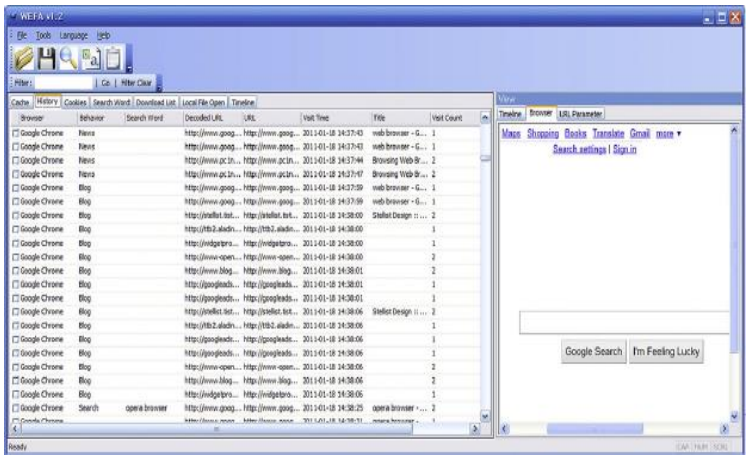

Fig. 2: Web Extracts Browser Analyzer (WEBA).

Evidence data was obtained from each of the hard drives C of the HP computers. The paths to the main artefacts extracted based on each browser are shown in Table 1.

Table 1: Paths to the Browser Artefacts Extracted

Browser	Main Artefacts	Path
Microsoft explorer	History, Downloads, and Cookies :	C:\Users\HP\AppData\Local\Microsoft\Internet Explorer
	Cache	C:\Users\HP\AppData\Local\Microsoft\Internet Explorer\CacheStorage
Microsoft Edge	History, Downloads, and Cookies	C:\Users\HP\AppData\Local\MicrosoftEdge\User\Default
	Cache	C:\Users\HP\AppData\Local\Microsoft\Edge\User Data\Default\Cache
Opera Mini	History, Downloads, and Cookies	C:\Users\HP\AppData\Local\Opera Software\Opera Stable
	Cache	C:\Users\HP\AppData\Local\Opera Software\Opera Stable\Cache
Firefox	History, Downloads, and Cookies	C:\Users\HP\AppData\Local\Mozilla\Firefox\Profiles\4pcuku3l.default
	Cache	C:\Users\HP\AppData\Local\Mozilla\Firefox\Profiles\4pcuku3l.default\cache2
Chrome	History, Downloads, and Cookies	C:\Users\HP\AppData\Local\Google\Chrome\User Data\Default\Download Service
	Cache	C:\Users\HP\AppData\Local\Google\Chrome\User Data\Default\Cache

Artificial Intelligence and Cyber Security Institute (AICSI) Ghana has Initially Compiled browser activities on their computers [17]. This study considered the compiled browser activities and therefore extracted data on the browser activities for the 20 computers used. The details of the browser activities have been presented in Table 2 based on the activity grid in Table 3.3. Each of the activities has been assigned a number under the grid.

Table 2: Activity Grid

Activity	Grid
Facebook browsing	1
Twitter browsing	2
Playing a movie online	3
Accessing a bank account	4
Search for child pornography	5
Accessing Whatsapp	6
Online chat	7
Emailing	8
Downloadings	9
Search for: robbery and fraud	10

In Table 3, the browser activities (on grid) against each computer have been presented. The computers have been coded and represented in capital letters. The grid that applies to each computer has been checked or otherwise unchecked. The summary of the counts of browser activities is shown in Table 4.

Table 3: Browser Activity Presence Matrix

	Grid									
Computer	1	2	3	4	5	6	7	8	9	10
A	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓
B	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
C	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
E	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓
G	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
H	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓
I	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
J	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
K	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
L	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
M	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
N	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
O	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Q	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
R	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
T	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓

Table 4: Summary of Browser Activities Presence

Browser activity	Count
Present	88
not present	12
Total	100

3. Dataset

The browser activities were grouped into browser artefacts such as: Downloads, Cache, Cookies, Typed URLs, sessions, Most visited sites, Screenshots, Form values, Favorites, and Bookmarks. The main datasets used in the analysis are the counts of browser artefacts from the browsers. Different classes of browser artefacts from 20 computers were obtained. The dataset can be found in Table 5a.

Table 5: a) Browser and Artefacts Counts

Browser Artefacts	Microsoft explorer	Microsoft Edge	Opera	Firefox	Chrome
History	23	16	177	299	313
Cache	45	38	463	585	611
Cookies	35	28	333	455	469
Typed URLs	23	16	177	299	313
sessions	122	115	1464	1586	1701
Most visited sites	25	18	203	325	339
Screenshots	23	16	177	299	313
Downloaded files	14	7	60	182	196
Favorites	13	6	47	169	183
Bookmarks	15	8	73	195	209
thumbnails	15	8	73	195	209

Table 5: B) Descriptive Analysis of Browser-Artefacts Counts

Statistic	Internet Explorer	Microsoft Edge	Opera	Firefox	Chrome
Count (n)	11	11	11	11	11
Mean	32.09	25.09	295.18	417.18	441.45
Standard Deviation	31.36	31.36	407.65	407.65	437.16
Minimum	13	6	47	169	183
1st Quartile (Q1)	15	8	73	195	209
Median (Q2)	23	16	177	299	313
3rd Quartile (Q3)	30	23	268	390	404
Maximum	122	115	1464	1586	1701
95% Confidence Interval for Mean	[11.02, 53.16]	[4.02, 46.16]	[21.32, 569.04]	[143.32, 691.04]	[147.77, 735.14]

Descriptive statistical analysis in Table 5b revealed that Chrome and Firefox browsers retained significantly higher volumes of forensic artifacts compared to Internet Explorer and Microsoft Edge. Chrome recorded the highest mean artifact count ($\bar{x}$ = 441.45), while Internet Explorer recorded the lowest ($\bar{x}$ = 32.09), indicating substantial differences in artifact persistence across browsers.

Data Analytical Tool and Analyses

Python programming language (Version 3.9) was the main tool used for the dataset analyses. The steps involved are;

STEP 1: Selection of algorithms for the dataset

STEP 2: Loading Python Libraries

STEP 3: Creating arrays for test and training datasets

STEP 4: Configuration for model testing and validations

STEP 5: Performance Evaluation of Algorithms

STEP 6: Graphical presentation of algorithm performance

The models were evaluated based on four metrics: Accuracy, Precision, Recall, and F1 performance. The performance of the Linear Discriminant Analysis was checked against the competitive algorithms. Performances were computed with the metrics in equations (1), equations (2), equation (3), and equation (4). The metrics are defined where TP is true positive, TN is True negative, and FP is false negative. The normal behavior that was wrongly predicted was indicated by FN, and finally, the normal performance that was indeed predicted as correct was indicated by TP. In the case of this study, the interest is in predicting the fact that computers used in forensic investigations have the possibility of containing certain evidence information that is common to others, and vice versa. The criteria and measures are also used by [7].

- Criteria for performance checking

The criteria based on the measures are scored between 0% and 100%. When the measure rate falls between 0% and 49%, it is considered a worse classifier; when it falls between 50% and 75%, it is considered an optimal classifier, and when it falls above 75%, the measure is considered a very good classifier. However, in a competitive analysis, the higher the percentage over other algorithms, the better the algorithm in classification. In the comparative analyses of the algorithms, the algorithm that has the highest score is chosen as the best algorithm [14]. In the measurement, the true positive (TP), True negative (TN), False positive (FP), and false negative (FN) are computed as part of the Python code. The structure of the performance measurement [7] is in Table 6.

Table 6: Structure of the Performance Measurement

Item		Classified (P)	Non-classified (N)
Artefacts expected to be classified	True (T)	TP	TN
Artefacts are not expected be classified.	False (F)	FP	FN

Based on Table 6, the accuracy measure in equation (1) aims at measuring the total classified and the total non-classified of the expected classifiable artefacts over the total counts of artefacts under the algorithm [7].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

Based on Table 6, the precision in equation (2) aims at measuring the total expected classified artefacts over the total counts of both expected to be classified and the non-expected to be classified under the algorithm [7].

$$\text{Precision}(P) = \frac{TP}{TP + FP} \tag{2}$$

Based on Table 6, the Recall in equation (3) aims at measuring the total expected classified over the total counts of artefacts that were not expected to be classified by the algorithm [7]

$$\text{Recall}(R) = \frac{TP}{TP + FN} \tag{3}$$

The F1 in equation (4) aims at measuring the rate of Recall (R) and Precision (P) per their sum. The significance of this measure is that it tells the rate at which the artefacts that were not expected to be classified were independently classified and also failed under classification by the algorithm [7], [18].

$$F1 = \frac{2PR}{P + R} \tag{4}$$

The performances are shown in Table 7, Table 8, Table 9, and Table 10.

Table 7: Classification Performance Output of Algorithms on Computers That Have Similar Artefacts

Algorithm	Accuracy	Precision	Recall	F1-Score
LDA	0.8635	0.8931	0.8535	0.8728
LRC	0.8582	0.8486	0.8482	0.8484
SVMC	0.8504	0.8604	0.8304	0.8451
NBC	0.8451	0.8551	0.8351	0.8449
KNN	0.8166	0.8262	0.8066	0.8162
DTC	0.7903	0.8100	0.7803	0.7948
RF	0.8423	0.8519	0.8326	0.8418
GB	0.8139	0.8231	0.8037	0.8134
NN	0.7876	0.8068	0.7775	0.7919

Table 8: Classification Performance Output of Algorithms of Browsers That Have Similar Artefacts

Algorithm	Accuracy	Precision	Recall	F1
LDA	0.8531	0.8505	0.8535	0.8519
LRC	0.8482	0.8402	0.8482	0.8441
SVMC	0.8401	0.8404	0.8404	0.8404
NBC	0.8351	0.8301	0.8351	0.8325
KNN	0.8063	0.8006	0.8066	0.8350
DTC	0.7703	0.7813	0.7803	0.7808
RF	0.8504	0.8479	0.8508	0.8493
GB	0.8456	0.8376	0.8456	0.8416
NN	0.8375	0.8378	0.8378	0.8378

Table 9: Classification Performance Output of Algorithms of Computers That Have Linked IPS

Algorithm	Accuracy	Precision	Recall	F1
LDA	0.7531	0.7535	0.7530	0.7525
LRC	0.7482	0.7482	0.7480	0.7481
SVMC	0.7401	0.7404	0.7401	0.7403
NBC	0.7351	0.7351	0.7350	0.7351
KNN	0.7063	0.7066	0.7067	0.7065
DTC	0.6703	0.6803	0.6800	0.6801
RF	0.7324	0.7323	0.7322	0.7324
GB	0.7036	0.7039	0.704	0.7038
NN	0.6677	0.6776	0.6773	0.6774

Table 10: Classification Performance Results on Artefacts of Algorithms of Booted and Unbooted Computers Artefacts

Algorithm	Accuracy	Precision	Recall	F1
LDA	0.8250	0.8497	0.8593	0.8545
LRC	0.7180	0.8478	0.8424	0.8451
SVMC	0.7100	0.7455	0.7458	0.7457
NBC	0.7050	0.7365	0.7368	0.7367
KNN	0.6960	0.6076	0.7033	0.6519
DTC	0.6405	0.6883	0.6811	0.6847
RF	0.8224	0.8471	0.8567	0.8519
GB	0.7154	0.8452	0.8398	0.8425
NN	0.7074	0.7429	0.7432	0.7431

4. Results

The results given are based on the research objectives. Graphs are used to represent the various results.

4.1. Graphical display of the dataset

The output graph of the dataset in Table 5a is presented in Figure 3. Data was obtained based on counts of artefacts present in the browser type. The investigation revealed that there were browser activities on each browser. The browsers on the computers recorded higher sessions compared with the other artefacts. There were a few bookmarks and favorite artefacts. There were reasonable counts of cache and cookies. Comparatively, Google Chrome (35%) and Firefox (45%) recorded the highest counts of artefacts. The implication is that Google Chrome and Firefox are the most preferred and widely used for browsing the web [16], which is relevant to the study. The significance of the artefacts analysis to the study is that it allows for classification with the algorithms and also sets the platform to enhance classification performance.

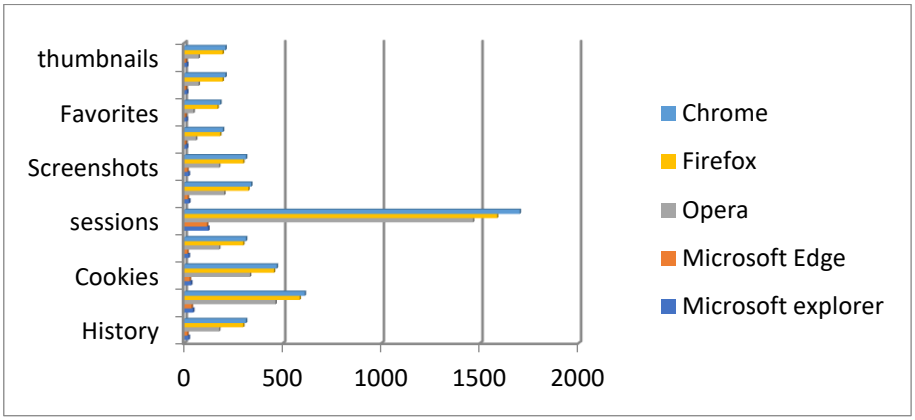

Fig. 3: Dataset on Browser Type and Artefact.

4.2. Performance of the algorithm on classifying computers that have similar artefacts

The dataset in Table 5a was used to classify computers that have similar artefacts. Based on the output Table 7, the results in Figure 4 show the Linear Discriminant algorithm (LDA) to have higher accuracy (86.35%), precision (89.31%), Recall (85.35%), and F1 (87.28%). These results are better than the competitive algorithms, which follow: Logistics Regression classifier (LRC), Decision Tree Classifier (DTC), K-Neighbors Neighbor (KNN), Naive Bayes Classifier (NBC), and Support Vector Classifier (SVMC). In a more significant way, the LDA, as per the results achieved in the study, enhances the classification performance of the existing algorithms. Practically, the LDA can be considered in forensic investigations where the interest is in checking the dependent usage of two computers, more specifically, where online browsing is suspicious. The Graphical presentation of algorithm performance in classifying computers that have similar artefacts are showed in Figure 4. The LDA algorithm here is seen as an enhanced algorithm over the competitive algorithms for browser extracts classification in the digital forensic exercise.

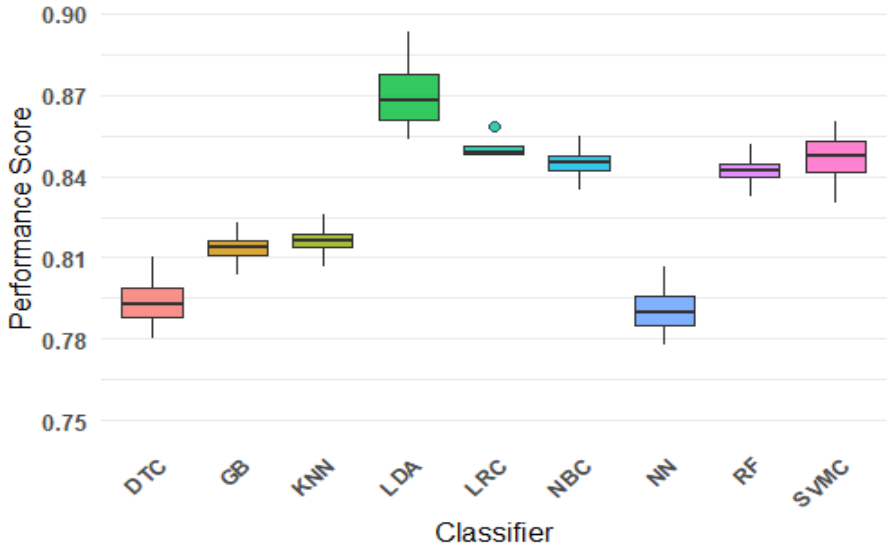

Fig. 4: Box-Plot of Algorithms' Performance.

4.3. Performance of the algorithm in classifying browsers with similar artefacts

The dataset in Table 5a was used to classify browsers that have similar artefacts. The results in Table 8 and as displayed on Figure 5 shows Linear Discriminant Analysis (LDA) to have higher accuracy (85.31%) and better in classifying browsers that have similar artefacts than the competitive classification algorithms such as Logistic Regression classifier (LRC), Decision Tree Classifier (DTC), K-Neighbors Neighbor (KNN), Naive Bayes Classifier (NBC) and Support Vector Classifier (SVMC). This analysis is significant because, in digital forensics, which involves cybercrime, investigators will be interested to know if two browsers on different computers have some common activities. In such a situation, the Linear discriminant algorithm classifier is better to use. The LDA algorithm here is seen as an enhanced algorithm over the competitive algorithms for browser extracts classification in the digital forensic exercise.

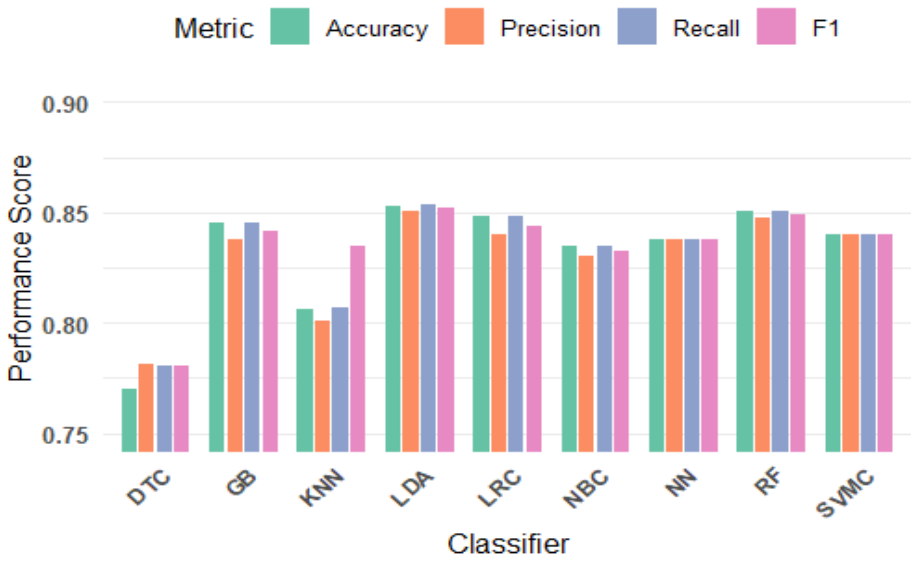

Fig. 5: Performance of Algorithm on Classifying Browsers with Similar Artefacts.

4.4. Performance of the algorithm on classifying computers that have linked IPs

The dataset in Table 5a was used to classify computers that have linked IP addresses based on artefacts. The output in Table 9 is demonstrated in Figure 6. The results show Linear Discriminant Analysis (LDA) to have higher accuracy (75.31%) and better in classifying computers that have linked or shared IPs than the competitive classification algorithms such as Logistic Regression classifier (LRC), Decision Tree Classifier (DTC), K-Neighbors Neighbor (KNN), Naive Bayes Classifier (NBC), and Support Vector Classifier (SVMC). In forensic investigations, it becomes essential to establish if two computers communicate. The best way to do this is to establish a connectivity link on IPs. Nonetheless, the surest way of analyzing this is to employ the linear discriminant algorithm to classify the entities involved. The LDA algorithm here is seen as an enhanced algorithm over the competitive algorithms for browser extracts classification in the digital forensic exercise.

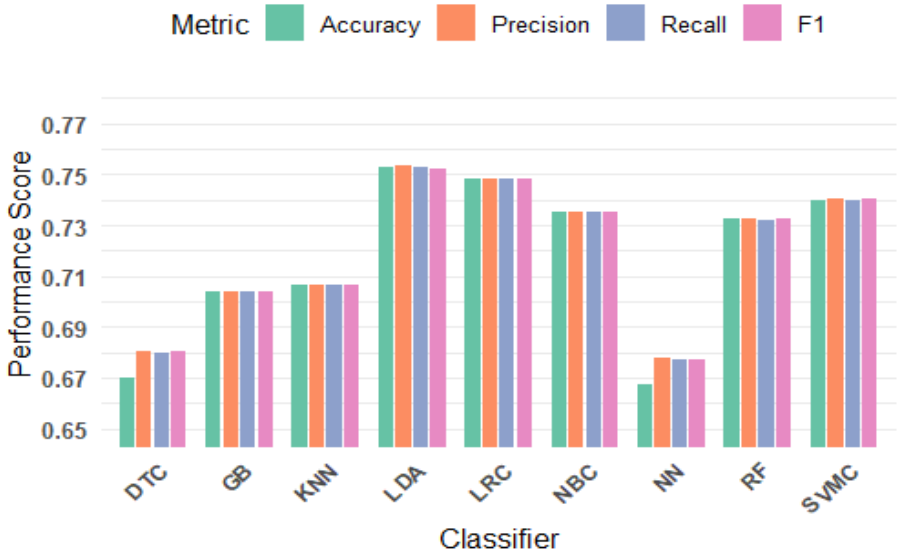

Fig. 6: Performance of the Algorithm on Classifying Computers That Have Linked IPs.

4.5. Performance of the algorithm in classifying artefacts on booted computers and unbooted computers before confiscation

Before assessing the computers, some were booted, and others were not booted. Similarly, in a real-life situation, during an investigation, investigators may come across computers that have been booted and those that have not been booted. In such a situation, it is therefore necessary to classify these computers well so that it gives a better understanding of the crime in perspective. In Table 5a, the dataset was used to classify computers that were booted against those that were not booted at the time of confiscation. The output in Table 10 has been graphically presented in Figure 7. The results indicate that the Linear Discriminant algorithm (LDA) has higher accuracy (82.50%), Precision (84.97%), Recall (85.93%), and F1 (85.45%). The LDA is better at classifying computers that were booted against those that were not booted than the competitive classification algorithms such as Logistic Regression classifier (LRC), Decision Tree Classifier (DTC), K-Neighbors Neighbor (KNN), Naive Bayes Classifier (NBC), and Support Vector Classifier (SVMC). The Linear discriminant algorithm performs better in classifying. This suggests that if an investigator wishes to classify booted computers and unbooted computers at the time of confiscation, it is more appropriate to use the linear discriminant algorithm for such classification. The LDA algorithm in the result is seen as an enhanced algorithm over the competitive algorithms for browser extracts classification in the digital forensic exercise.

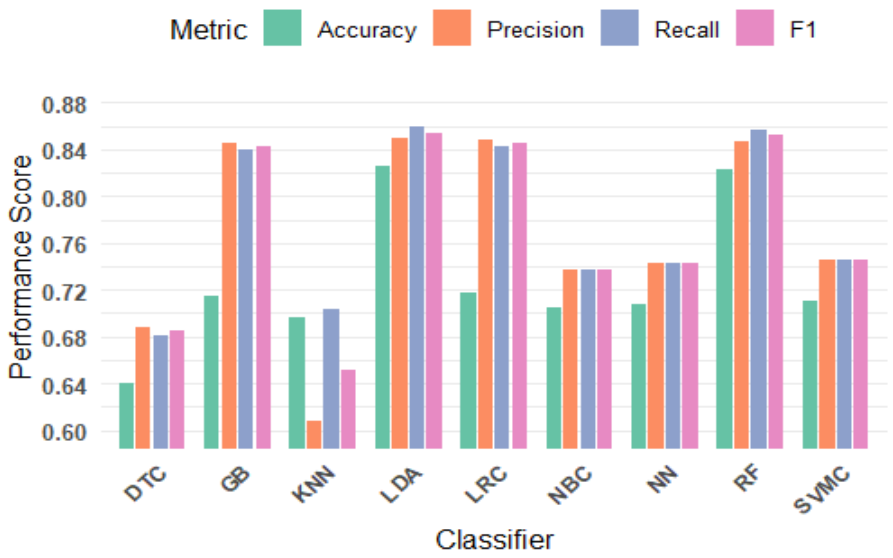

Fig. 7: Performance of Algorithm on Classifying Artefacts on Booted Computers and Unbooted Computers Before Confiscation.

5. Conclusion

The Linear Discriminant Algorithm is better in classifying browser extracts than the decision tree, Naïve Bayes, Support Vector Machine, multi-layer perceptron, and K-Neighbors. Some reach strengths of the LDA are: supporting multiple dependent variables, ensuring the reduction of errors, and easiness of interpretation were demonstrated. The LDA algorithm from the results was seen as an enhanced algorithm over the competitive algorithms for browser extracts classification in the digital forensic exercise.

6. Recommendation

- The Linear Discriminant Algorithm should be used for classifying web browser artefacts, more especially when the focus is on computer against artefacts classification and browser artefacts classification.
- Classification of browser artefacts in terms of linked or shared IPs, booted computer, or unbooted computer should be of major focus in digital forensic practice because these tell if two or more suspected criminals communicate among themselves on the same crime intention. It gives a clue to the investigator where to put further concentration in the investigation process.

Future Work

Future studies may extend this work using larger multi-institutional forensic datasets where access permits. Future studies can also look at applying the Linear discriminant algorithm to other areas of forensic like email forensics and car network forensics.

Conflict of Interest

All authors have no conflict of interest.

References

[1] Andrew, M., Ibrahim, B., and Talal A. I. (2012). Portable web browser forensics: A forensic examination of the privacy benefits of portable web browsers, 2012 International Conference on Computer Systems and Industrial Informatics, 18-20 Dec. 2012.
[2] Junghoon, O., Seungbong, L. (2011). Advanced evidence collection and analysis of web browser activity, Elsevier - Digital Investigation, Volume 8, Supplement, August 2011, Pages S62-S70. https://doi.org/10.1016/j.diin.2011.05.008.
[3] Peter., E., (2010). How Unique is Your Web Browser? In Proceedings of the 10th International Conference on Privacy Enhancing Technologies (PETS'10). Springer-Verlag, Berlin, Heidelberg, 1–18. https://doi.org/10.1007/978-3-642-14527-8_1.
[4] Gábor. G., Gulyás, D., Francis S., Nataliia B., and Claude C. (2018). To Extend or not to Extend: on the Uniqueness of Browser Extensions and Web Logins. In 2018 Workshop on Privacy in the Electronic Society (WPES'18). ACM, 14–27. https://doi.org/10.1145/3267323.3268959.
[5] Donny, J., O,. Narasimha and Shashidhar (2013), Do private and portable web browsers leave incriminating evidence?: a forensic analysis of residual artifacts from private and portable web browsing sessions, EURASIP Journal on Information Security, December 2013, 2013:6. https://doi.org/10.1186/1687-417X-2013-6
[6] Huwida Said, Noora Al Mutawa and Ibtesam Al Awadhi, Forensic analysis of private browsing artefacts, 2011 International Conference on Innovations in Information Technology, 25-27 April 2011. https://doi.org/10.1109/INNOVATIONS.2011.5893816.
[7] Rami, M. A., Mohammad M., Alqahtani (2019) A comparison of machine learning techniques for file system forensics analysis. Journal of Information Security and Applications 46 (2019) 53–61. https://doi.org/10.1016/j.jisa.2019.02.009.
[8] Ankit Agarwal, Megha Gupta, Saurabh Gupta & S.C. Gupta. (2011). Systematic Digital Forensic Investigation Model, International Journal of Computer Science and Security (IJCSS). Volume (5). Issue (1).
[9] Faheem M., Kechadi MT., Le-Khac NA. (2016), Toward a new mobile cloud forensic framework 6th IEEE International Conference on Innovative Computing Technology, Ireland, 2016. https://doi.org/10.1109/INTECH.2016.7845142.
[10] Brown, M., Lary, D., Vrieling, A., Stathakis, D., & Mussa, H. (2008). Neural networks as a tool for constructing continuous NDVI time series fromAVHRR and MODIS. International Journal of Remote Sensing, 29(24), 7141–7158. https://doi.org/10.1080/01431160802238435.

[11] Sadilek, A., Kautz, H. and Bigham, J.P. (2012) Finding Your Friends and Following Them to Where You Are. Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, Seattle, 8-12 February 2012, 723-732. https://doi.org/10.1145/2124295.2124380.
[12] Zhang, R. Hu,Z. Pan,G. and Wang,Y. (2016).''Robust discriminative nonnegative matrix factorization,'' Neurocomputing, vol. 173, pp. 552–561. https://doi.org/10.1016/j.neucom.2015.07.032.
[13] Devi Prasad bhukya and S. Ramachandram (2010)" Decision tree induction- An Approach for data classification using AVL –Tree", International journal of computer and electrical engineering, Vol. 2, no. 4. https://doi.org/10.7763/IJCEE.2010.V2.208.
[14] Gupta, N.A. (2017). Literature Survey on Artificial Intelligence. https://www.ijert.org/research/aliterature-survey-on-artificial-intelligence IJERTCONV5IS19015.pdf (accessed on 7 January 2020).).
[15] Raghavan S. and Raghavan S. V. (2013). AssocGEN: Engine for Analyzing Metadata Based Associations in Digital Evidence, In Proceedings of the 2013 8th International Workshop on Systematic Approaches to Digital Forensics Engineering (SADFE), IEEE 978-1-4799-4061-5, Hong Kong, China, Nov 21-22, 2013. https://doi.org/10.1109/SADFE.2013.6911541.
[16] Pierre L., Gildas A, Benoit B., and Nick N. (2019). Morellian Analysis for Browsers: Making Web Authentication Stronger with Canvas Fingerprinting. In Detection of Intrusions and Malware, and Vulnerability Assessment - 16th International Conference, DIMVA 2019, Gothenburg, Sweden, June 19-20, 2019, Proceedings. 43–66. https://doi.org/10.1007/978-3-030-22038-9_3.
[17] Artificial Intelligent and Cyber security Institute (AICSI) Ghana (2020). Department of Forensic.
[18] Kok, S. H., Azween, A., & Jhanjhi, N. Z. (2020). Evaluation metric for crypto-ransomware detection using machine learning. Journal of Information Security and Applications, 55, 102646. https://doi.org/10.1016/j.jisa.2020.102646.