Electronic Commerce Research (2026) 26:2467–2502
https://doi.org/10.1007/s10660-025-10029-9

Fraud detection in e-commerce: a comparative analysis of features to enhance machine learning models

Manuel Sánchez-Paniagua[1] · Eduardo Fidalgo[2,3] · Enrique Alegre[2,3] · Francisco Jáñez-Martino[2,3]

Accepted: 22 July 2025 / Published online: 9 September 2025

Abstract

In recent years, e-commerce has experienced growth in sales, brands and customers. Unfortunately, cybercriminals have taken advantage of this by creating fraudulent websites to scam customers. The large amount of new e-commerce websites outnumbers the manual reporting capabilities, exposing users to these attacks. In this work, we used machine learning techniques to identify possible fraudulent online stores. To achieve this, we created ELFW-2031 (E-commerce Legitimate Fraudulent Websites), an updated dataset of manually verified legitimate and fraudulent e-commerce websites and a comprehensive set of resources for researchers to compare their methods. We released this dataset for public use to overcome the lack of a comprehensive corpus of this type of websites. We also designed a novel set of 50 features using six different resources obtained from the website content and external services. We used these new features to train and test two models: (i) a model with all available resources focused on improving accuracy and (ii) a model focused on scalability independent of external services. The proposed models achieve F1 scores of **96.88%** and **96.53%** respectively using XGBoost. Finally, we evaluated the performance of the proposed features, showing that novel features from social media and the technology analysis were the most valuable ones.

Keywords Fraud detection · E-commerce · Machine learning · Feature engineering

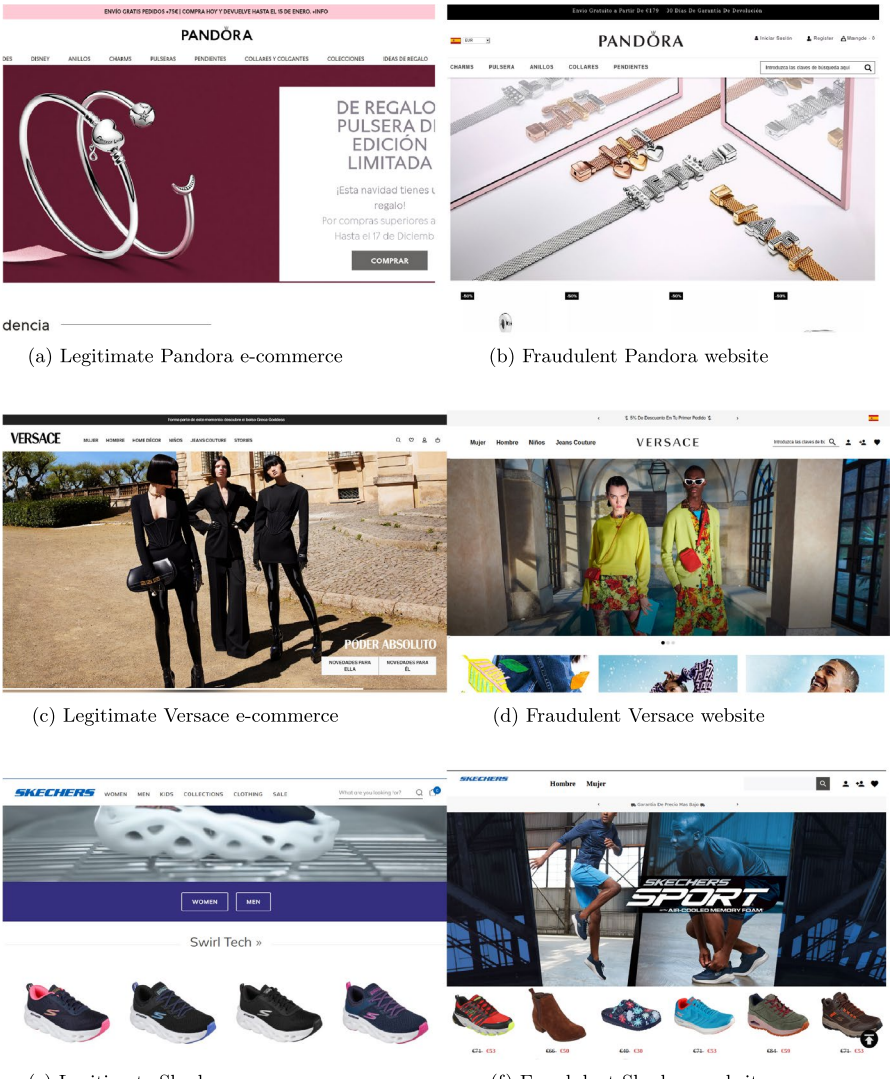
1 Introduction

Over the last few years, e-commerce has grown in the number of users and sales. Companies are developing websites to display their products and services, reaching out to more clients and allowing them to buy at any time of the day [1]. Statista

Extended author information available on the last page of the article

reported 6.630$ billion in e-commerce retail sales worldwide in 2024, and it is expected to reach 8.034$ billions by 2027 [2].

However, the increase in the number of users and websites has paved the way for new attacks where fraudsters aim to steal money or sell counterfeit products. These attacks can lead to other types, such as phishing or account takeover [3]. Fraudsters create websites that use well-known brands and offer great deals on expensive products. The look and feel of most fraudulent websites is designed to deceive users, on Fig. 1 we show the similarity between a legitimate and a fraudulent website. For this



(a) Legitimate Pandora e-commerce (b) Fraudulent Pandora website

(c) Legitimate Versace e-commerce (d) Fraudulent Versace website

(e) Legitimate Skechers e-commerce (f) Fraudulent Skechers website

Fig. 1 Fraudulent websites using trademark theft (right panels) and mimicking the styles of their legitimate counterparts (left panels). Figures **a** and **b** correspond to Pandora, **c** and **d** to Versace, and **e** and **f** to Skechers

task, fraudulent sites are created by using famous brands and offering great deals on expensive products.

The European Commission surveyed online frauds, which found that 55% of European consumers have been scammed while shopping online and 26% have had money stolen or received counterfeit products [4]. Not only do consumers suffer from online frauds, but, according to Juniper, businesses have lost over 38$ billion due to fraudulent activities in 2023 and claimed losses from online payment fraud to exceed $362 billion globally over the next five years [5].

Current defences to protect users are built-in features in browsers or antivirus software. The most common technique is to create a blocklist, which reported fraudulent and phishing websites are stacked. Each time a user visits a website, the browser checks whether the domain is on the list or not. One of the main drawbacks is maintenance, as reports need to be verified and added to the list. In addition, they rely on anonymous reports, which is a concern for detection performance, as users who reach the site before it is reported will enter it unaware.

Additionally, there are tools where users can provide the domain name and obtain a risk factor based on a set of fixed rules, such as ScamAdviser[1] or Scanmer.[2] These rules are grounded on resources such as the SSL certificate, WHOIS information, the age of the domain, or reviewing and ranking platforms like Trustpilot and Alexa. The main disadvantage of these tools is the lack of flexibility due to the rules and the variables considered. For example, a commonly used detection rule is the age of the domain, assuming that a long-lived domain is most likely to be legitimate. Therefore, a legitimate website with a short life span might have a high-risk factor in the evaluation process. Meanwhile, a fraudulent domain registered with an SSL certificate a few months ago might obtain a low-risk factor, encouraging the user to buy from the fraudster's site.

Researchers have developed artificial intelligence techniques to overcome the issues of blocklists and rules [6–8]. These studies use resources such as the URL or the HTML alongside ranking services. However, state-of-the-art features rely on external services such as WHOIS, Google Search or Alexa. However, they depend on the response time from external servers, if available, which can generate delays in response.

Another problem in detecting fraudulent websites is data availability. As a result, researchers have trained their models on small-size datasets that are not publicly available. This prevents the development and fair comparison of newly proposed models.

In this paper, we have focused on developing a highly accurate system that can be integrated into security tools, such as browser extensions or antivirus. In this way, users have access to advice about potentially fraudulent websites.

To achieve this, we first built an updated and manually verified dataset, namely ELFW-2031 (E-commerce Legitimate Fraudulent Websites). This objective is fundamental as the dataset needs to represent the real-world environment to be reliable when working with current fraudulent websites. It is worth noting that we limit the

[1] https://www.scamadviser.com/ Retrieved June 2025.

[2] https://www.scamner.com/ Retrieved June 2025.

scope to e-commerce websites, and other scam techniques are not included in the dataset. This dataset covers raw data extracted from the websites, so other researchers can use it to compare their methods regardless of the input data.

Then, we defined six different types of features depending on the resource used for their extraction. Four of these resources have been used previously for this and similar tasks, such as URLs, HTML, SSL certificates, and HTTP headers. In addition, we proposed two new resources, website technology analysis and social media impact, to improve the result of the algorithm. To prove this, we arranged a series of experiments to determine the best machine-learning algorithm. We also proposed two different models, a complete one focused on performance and a standalone version that is independent of third parties and provides accurate real-time classification. Finally, we compared the performance of individual features and the impact of features depending on the associated resource.

In summary, the contributions presented in this paper are as follows:

- We present a framework for fraudulent e-commerce website detection based on machine learning that can operate with third-party resources to optimize detection without them to achieve endpoint scalability.
- We provide a publicly available ELFW-2031[3] dataset with 2,031 manually verified samples. Each sample contains a wide range of raw resources, allowing researchers to develop and compare their work without restrictions on the method used.
- We induce a new set of 50 features and resources to improve classification results and endurance against bypass methods. The proposed system and related research took place under the needs of the strategic project LUCIA (Fighting Against Cybercrime using Artificial Intelligence) with the Spanish National Cybersecurity Institute (INCIBE). It would be adapted and prepared for integration into tools and services that could be useful for INCIBE and its Computer Emergency Response Team (CERT), allowing the detection of real fraudulent websites to take them down and protect citizens from these attacks.

The rest of the paper is structured as follows. Section 2 explains related works about fraudulent website detection. Section 3 presents the dataset collection and specifications. The designed features and groups are detailed in Sect. 4. Section 5 contains the experiments and obtained results. Finally, conclusions and future work are covered in Sect. 6.

2 Related work

Fraudulent website detection refers to the process of identifying websites that are designed with malicious intent. This includes various types of scams, such as the sale of counterfeit products [9], fraudulent pet sellers, bogus charity websites, and schemes related to cryptocurrency, phishing attacks [10] or stock markets [11].

[3] Dataset available under request on https://gvis.unileon.es.

Cybercriminals currently used social engineering technique to mislead users. Phishing became one of the most common tools to create fake webpages, often with URLs that mimic those of legitimate sites, in order to steal confidential information such as passwords or banking data. Moreover, these websites usually offer counterfeit products that appear to be genuine but are either of poor quality or do not exist at all [9], leading to financial losses for both consumers and businesses. In general, these sites are characterized by poor design, spelling mistakes, invalid or missing security certificates, and even alarming messages intended to pressure the user into acting quickly. Detecting and preventing access to such sites is crucial to protect both personal security and the integrity of brands and organizations.

Researchers have developed machine learning techniques to detect fraudulent websites, either by focusing on specific scam types or by building more generalized detection models [12]. These approaches aim to address the limitations of traditional rule-based and blacklist-based systems, which often struggle to keep up with the evolving nature of online threats. As scammers usually mimic legitimate websites, previous studies have focused on identifying those subtle clues within the website content and structure to differentiate fraudulent sites from legitimate ones.

Given the significant impact of e-commerce scams, which represent one of the most prominent forms of fraudulent websites, it is essential to examine other types of online scams as well. Identifying commonalities and differences across these various scam types can reveal shared tactics used by cybercriminals. This, in turn, can support the development of more robust and generalised detection models capable of addressing the broader spectrum of cyber threats. This work mainly focuses on detecting websites with potentially fraudulent content, techniques used to identify other types of scams can contribute to addressing the wide range of tactics employed by cybercriminals. The relevance and significance of the previous studies are summarized in Table 1.

2.1 Fraudulent e-commerce websites

A common strategy employed by cybercriminals is the creation of fake online stores. The primary difference between phishing sites and fake e-commerce websites lies in their intent. Phishing sites typically aim to directly steal user credentials or credit card information in a straightforward manner [21]. In contrast, fake e-commerce sites often feature well-designed, convincing interfaces that lure users with attractive discounts and offers. When users place orders, they unknowingly submit extensive personal information to the attackers–often including more detailed data than what is usually targeted in phishing attacks [22].

Due to the nature of this category, detection systems commonly analyze HTML and content resources, such as prices, discounts, currencies, and other key indicators, to identify fraudulent activity [16]. When users place orders on these fake sites, they unknowingly submit extensive personal information to attackers, often including more detailed data than what is usually targeted in phishing attacks [22].

Nevertheless, other resources have also been incorporated. For example, Carpineto and Romano [13] proposed a framework comprising three modules: a Selenium

Table 1 Comparison of techniques for detecting different types of fraudulent websites, highlighting their strengths, limitations, relevance to e-commerce fraud detection, and the specific scam category addressed

Reference	Techniques	Advantages	Disadvantages	E-commerce connection	Fraud type
Carpineto & Romano [13]	URL, HTML, WHOIS, Alexa, SEO; SVM classifier	Good accuracy (88%); uses multiple data sources	Small dataset; dependent on external metadata	Features like WHOIS and SEO applicable across scam types	FE
Gopal et al. [14]	HTTP traffic and interaction with third-party services (CDN, ads, analytics)	Behavioral-based; captures real-time interactions	Time-sensitive; requires active monitoring	Useful for e-commerce as fraudulent shops misuse ad/tracking services	FE
Audroné et al. [15]	URL, content, third-party features; Random Forest classifier	High accuracy (96.93%); highlights features like young domain & refund policy	Public but limited dataset; fixed features	Relevant to e-commerce; refund policies useful in phishing detection	FE
Kotzias et al. [16]	Network, custom crawler, HTTP headers, content, domain data	Novel features; combines behavioral and structural signals	Custom crawler; dataset not public	Overlaps with counterfeit and phishing; Facebook links and domain age cross-cutting	FE
Xie et al. [17]	CNN on browser, IP, signup/purchase timestamps	Deep learning; learns complex patterns	Lower accuracy (83.6%); less interpretable	Purchase behavior and device data relevant for fake e-commerce	FE
Maktabar et al. [8]	Text preprocessing, BoW, POS tagging; Naive Bayes, LR	High accuracy (99.41%); text-focused	Language dependent; may not generalize well	Persuasive language, fake reviews common in e-commerce scams	P
Beltzung et al. [18]	HTML, CSS, DOM, JS + TF-IDF; XGBoost	Strong performance (96.7%); code-based semantic patterns	Dataset in German; limited generalizability	Code and text patterns overlap with phishing and e-commerce fraud	P
Khoo et al. [19]	BoW; metadata; image screenshots and XGBoost	Combines visual and textual; 98.9% accuracy	Screenshot analysis costly	Visual analysis helps detect fake branding in e-commerce fraud	P
Mostard et al. [6]	HTML features + CNN on screenshots for logos/payment detection	High performance (F1=0.980); visual fraud indicators	Requires image processing; resource-heavy	Detects fake logos/payment icons relevant to counterfeit and e-commerce	FE CP
Wadleigh et al. [20]	FQDN, content (brands, prices), WHOIS; SVM	Focus on counterfeit; 86.4% accuracy	Older study; narrow domain	Shared traits with e-commerce fraud: pricing, brand misuse, domain data	CP
Wu et al. [7]	URL, content, structure, WHOIS; LR, DT, SVM	Social media links highly discriminative (7.75% vs. 80.18%)	Dataset limited to China	Social media presence relevant in fraudulent e-commerce and phishing	CP

FE Fraudulent E-commerce, *P* Phishing, *CP* Counterfeit products

WebDriver[4] feature extractor, a binary classifier to distinguish e-commerce websites, and a second binary classifier to identify fraudulent ones. The final stage uses 33 features derived from the URL, HTML, WHOIS data, Alexa ranking, and Search Engine Optimization (SEO) information, which are fed into an SVM classifier, achieving 88% accuracy on a balanced dataset of 500 samples. Gopal et al. [14] presented a third-party baseline that uses the interaction of the actual websites with external services such as CDN (Content Delivery Networks), analytics services, advertisement domains and others. They defined 10 out of 20 independent variables regarding the HTTP traffic generated in a 15 second time frame. They compared a broad set of traditional models using two subsets of 205 legitimate pages and the same 93 fraudulent websites across both subsets.

Audroné et al. [15] also proposed a set of 18 features divided into three categories, URLs, content and third-party, and evaluated their combination using machine learning models as classifiers. From the combination of seven features to fifteen, the Random Forest obtained the best accuracy (96.93%). is achieved using thirteen and fifteen features. Indication of the young domain and the presence of money-back payments were the most relevant features. They highlighted the lack of publicly available datasets in similar studies and decided to make their dataset public.[5] However, this dataset includes only their specific features, limiting its use to compare their work directly.

In ther work, Kotzias et al [16] developed a novel set of features, building on previous works, which includes data from the network, search engine crawlers (as defined by robots.txt), HTTP headers, content, and domain information. Link to the external review system, domain age and Facebook account were the most relevant features. They used a custom dataset and evaluated SVM and RF classifiers.

Finally, Xie et al. [17] evaluated a Convolutional Neural Network (CNN) against traditional machine learning models on a Kaggle dataset containing the website source, purchase value, browser employed, sign up time, purchase time and IP address. The CNN model outperformed the traditional ones with 83.60% accuracy.

Building on these approaches, some studies have moved beyond exclusive feature extraction to focus on identifying patterns in language through text analysis [8], structural examination and visualization via webpage screenshots [6, 19], or by combining these diverse inputs with traditional features using multimodal models [23].

Maktabar et al. [8] used a web crawler and a preprocessed step based on deleting HTML tags, omitting empty words and punctuation marks, converting to lowercase and removing prefixes and suffixes through stemming to obtain the text from the websites. Then, they applied Bag of Words (BoW) [24] and Part-of-Speech [25] for feature extraction and machine learning models as a classifier. Naïve Bayes Multinomial, 99.41%, and LR, 98.83%, achieved the highest accuracy on their custom and private dataset.

Leveraging textual features, Beltzung et al. [18] utilized HTML, CSS, Document Object Model (DOM), and JavaScript files, combined with the Term Frequency-

[4] https://www.selenium.dev Retrieved June 2025.

[5] https://data.mendeley.com/datasets/m7xtkx7g5m/1/files/33f913ef-0f84-4f36-b2de-3c4e281b24b1 Retrieved June 2025.

Inverse Document Frequency (TF-IDF) technique. The authors achieved 96.70% accuracy using an XGBoost classifier on their dataset of German web pages, which included 3,801 fraudulent websites and 2,838 legitimate ones.

Khoo et al. [19] proposed a system adding HTML metadata and image screenshots to the feature vector created by BoW. Among models evaluated on their dataset of 258 legitimate and 239 fraudulent samples, eXtreme Gradient Boosting (XGBoost) with only BoW obtained 98.90% of accuracy. Whereas Mostard et al. [6] collected a larger dataset of 1876 fraudulent domains from a Dutch consumer association and local law enforcement agencies and 1499 legitimate domains from the Dutch e-commerce. They evaluated the combination of two sets on this dataset: (a) using only contextual 17 features from HTML, highlighting the number of web pages, social media links, copyright, open ports in the server and email or phone numbers in the website and (b) analyzing the screenshots to detect payment methods and social media using a Convolutional Neural Network (CNN) for logo detection. Finally, Random Forest (RF) outperformed the rest of the models with 0.9800 F1-Score.

Multimodal approaches achieved high performance in evaluations; however, they have a significant drawback when applied to real-time tasks or those requiring immediate detection: their complexity and runtime. These methods demand substantial computational resources and are slower compared to feature-based alternatives.

2.2 Phishing websites

Although phishing detection is commonly considered a subset of fraudulent website detection, it has emerged as a distinct research area due to the high prevalence and rapidly evolving nature of phishing attacks, making it the most extensively studied type of fraudulent website in the literature.

These attacks are not only widespread but have also extended beyond traditional websites to platforms such as social media, SMS [26], and spam emails [27]. Given the similarities between phishing and other forms of online fraud [28], it is important to examine the specific deceptive techniques phishers use to trick users. These include typo-squatting, use of long or complex subdomains, creating a sense of urgency, employing social engineering tactics, and cloning legitimate websites, among other strategies.

Phishing websites typically impersonate well-known companies to mislead users and steal sensitive credentials or valuable information. To combat this threat, researchers have developed a variety of detection methods that leverage different resources, such as analyzing URLs, inspecting HTML content, and even evaluating screenshots of the webpages. For example, Sahingoz et al. [29] proposed two feature sets regarding the URL: The first set is composed of 39 NLP (Natural Language Processing) features, and the second one combines 102 word features. Using a Random Forest (RF) classifier and a NLP set of features, they reached 97.98% accuracy on a 73, 575 phishing URL dataset collected in their previous work [10].

Li et al. [30] focused only on URL and HTML features, combining three methods–GBDT, XGBoost, and LightGBM–to construct a stacking model utilizing 20 distinct features. They built a model with 97.11% accuracy using their 50, 000 samples dataset. Additionally, they implemented a small CNN (Convolutional Neural Network)

based on visual features extracted from website screenshots, which improved their model accuracy to 98.60%.

Rao et al. [31] proposed CatchPhish, a phishing detection framework based only on URLs. They combined TF-IDF and 35 handcrafted features extracted from the URL. First, they proposed a set of keywords with high frequency on the phishing subset. Then, they used it to count the keywords within the hostname and the entire URL. Using an RF classifier, they obtained 94.26% accuracy on their dataset, 98.25% on the URL dataset of Sahingoz et al. [29] and a 97.49% on a URL dataset of Marchal et al. [32].

In our previous research [33], we proposed a phishing detection model based on novel resources and features, including URL, HTML and website technologies. Furthermore, we used login pages to train the model, improving performance in real environments. We obtained 97.95% accuracy on a 134, 000 samples dataset, where features based on the technology used improved the algorithm performance.

Finally, Majgave and Gavankar [34] evaluated the integration of transformer models with a deep belief network, proposing a novel phishing detection system that takes advantage of One Hot Encoding for URL representation, transfer learning-based feature extraction, and a hybrid Transformer-based Deep Belief Network (TB-DBN). Their model achieves high accuracy (99.4%) and strong performance metrics, demonstrating advanced capabilities in the detection and prevention of early phishing websites.

2.3 Counterfeit products detection

Although less frequently addressed in the literature, another significant type of website-related scam involves the sale of counterfeit products. This differs from fake e-commerce sites in that counterfeit products are genuine items that are illicitly designed to resemble authentic goods. In contrast, fake e-commerce websites aim to replicate legitimate sites to mislead users but typically do not offer genuine products. Reviewing related works allows us to identify potential common features and gaps in detecting fake e-commerce.

Wadleigh et al. [20] went over Google queries to find counterfeit products and proposed a system focused on detecting counterfeit websites using the URL, the HTML and third-party information. They generated a dataset with 234 websites selling fake products and 368 legitimate pages. Then, a 13 features set was designed and divided into three groups. The first one is related to the Fully Qualified Domain Name (FQDN), and the second one covers information about website content, including features about discounts, brands and prices. Finally, the third group involves external services, such as WHOIS and Alexa, and defines descriptors about the registration country, domain age, and whether the domain is in Alexa's top 100, 000. SVM classifier outperformed the rest and obtained 86.40% of accuracy. Furthermore, this study stated that the number of currencies detected, the use of large iframes and the age of the domain were the most valuable features. Wu et al. [7] trained three machine learning models: Logistic Regression (LR), Decision Trees (DT) and Support Vector Machines (SVM), to detect websites selling counterfeit products. The authors built a balanced dataset with 800 websites registered in China. By analyzing the samples,

they proposed a set of 17 features extracted from the URL, the content, the structure of the website and WHOIS information. They obtained 90.88% accuracy using LR and proved the importance of the social media links (Facebook, Instagram and Line) in the e-commerce websites since only 7.75% of fraudulent websites had social media links against the 80.18% of the legitimate ones. Removing features related to the social media reduced the accuracy to 84.63%.

2.4 Converging strategies for fraudulent detection

Given the overlapping characteristics and detection challenges across phishing, fake e-commerce, and counterfeit product scams, this work advocates for the development of integrated detection systems that unify these fraud types under a comprehensive framework. By leveraging converging strategies, such systems can enhance fraud mitigation through the combined analysis of multiple signals, including URL behavior, website design patterns, user interaction data, and product authenticity indicators. This holistic approach may aim to improve detection accuracy while minimizing false positives, effectively countering the increasingly sophisticated techniques employed by fraudsters in online marketplaces.

Moreover, although many of the aforementioned methods demonstrate high performance, their reliance on limited datasets, often constrained by annotation methodologies, language, or source diversity, undermines confidence in their generalizability to real-world scenarios. Additionally, the majority of these datasets are either not publicly available, require complex access procedures, or include only the specific features proposed by their original studies, which restricts their usability primarily to replicating existing work rather than fostering broader research and development.

Another one relevant limitation for real-world scenarios is the reliance on third-party or external services, such as Alexa or WHOIS. This reliance makes solutions incorporating these features less scalable and, consequently, imposes time and cost constraints when implemented in end-user applications. Finally, content features can be easily bypassed by cloned websites with attributes identical to those of the original site.

Building on the analysis of these challenges, we propose a novel set of features inspired by the conclusions, limitations, and recommendations of prior studies. Furthermore, we considered mandatory to build and introduce a comprehensive custom dataset that includes all files extracted from the websites, empowering the scientific community not only to replicate our work but also to explore new feature engineering approaches and develop innovative detection models.

3 Dataset ELFW-2031

In this chapter, we explain the dataset creation process along with its content and structure, focusing on its usability across different methods to allow other authors to use it for comparisons and research.

3.1 Creation

One of the main obstacles to our goal is data availability. Researchers create their own datasets only with the resources needed to fit their method, and usually, they are not publicly available. Using the same data is crucial to compare different approaches objectively. To the best of our knowledge, currently publicly available datasets are limited in terms of resources and features and, therefore, cannot be used to develop or compare techniques that depend on other uncollected resources. For this reason, we have focused on building a standard dataset suitable for different methods using the comprehensive set of resources described below. The ELFW-2031 (E-commerce Legitimate Fraudulent Websites) dataset is publicly available and can be requested by email.[6]

The dataset is composed of data collected from the actual websites. This task is becoming difficult due to the new web technologies that limit the amount of information that can be extracted [8]. These measures are based on CAPTCHA to avoid bots and automatic web crawlers. To overcome this difficulty, we use the Selenium Web Driver and a custom application to visit and extract information from websites by simulating user actions in the browser.

In order to create the dataset, we obtained a list of legitimate domains from the Spanish trust mark"Confianza Online".[7] This trust mark verifies the legitimacy of the company behind a registered domain. For fraudulent websites, periodic reports were obtained from INCIBE (Spanish National Cybersecurity Institute), and a list of fraudulent domains was manually verified. From November 2020 to November 2021, we collected 1292 legitimate e-commerce domains and 739 fraudulent websites.

3.2 Collected information

To enable the use of our dataset across different methods, we collected a set of resources to meet most of the state-of-the-art data requirements, including URLs, HTML content and screenshots, which were widely used in previous works. In addition, we extended these resources to extract new features and improve classification performance. The resources have been collected as they appear to end-users, i.e., no preprocessing or feature extraction has been applied. They are presented in a raw file format for researchers to extract their custom features.

After reviewing the state-of-the-art, we observed that the main resources used by other authors are the URL, the HTML and WHOIS information. The final resources collected are listed below:

URL: It represents the unique identifier of the website and it was used in other works [7, 13] to search for keywords such as"offer","replica"or"cheap". In addition, e-commerce websites typically place their brand or company name in the URL domain. We extract the final complete URL of the visited domain.

HTML: This source code comprises most of the information from a website and was used in most of the state-of-the-art works [6, 7]. Data such as product names,

[6] https://gvis.unileon.es/ After the review process, we provide a link to the dataset.

[7] https://www.confianzaonline.es/ Retrieved June 2025.

prices, and discounts can be found in this resource. We retrieved the complete HTML code, including the CSS and JavaScript code used on the main page of the domain.

Screenshots: Previous works [6, 19] used the screenshots to detect social media or payment logos using different techniques, such as Regions of Interest and Image Moments. In addition, screenshots were used in similar areas, such as phishing detection to carry out logo detection [35] and product detection [36] with the potential to identify the affected brands. To provide a valuable image resource, we collected two high-resolution screenshots (1848px x 911px) from both the top and the bottom side of the website, using Selenium Web Driver.

HTTP headers: are commonly used to establish the first layer of website security, as they define directives that change the behaviour of the client. There are a wide variety of headers that improve website security; for example, HSTS (HTTP Strict Transport Security) ensures that the client uses HTTPS to access the website, and X-Frame-Options defines the allowed origins to load the page in frames. Developers combine different HTTP headers to set an appropriate configuration to minimize the attack surface [37]. We retrieve the HTTP response headers to explore their use in this classification task.

Web technologies: E-commerce and other companies are developing their relationship with customers by providing the best shopping experience on their websites. To achieve this, companies invest in SEO technologies and frameworks to make the website more accessible and attractive to customers [38]. Also, implementing advanced web technologies allows the website to scale horizontally while smoothing its maintenance. Our premise is that fraudulent websites are clones created with toolkits or using the raw source code copied from another website. Therefore, the technologies used in the process are minimal. To detect them, we used Wappalyzer,[8] a tool based on fingerprinting that analyzes the HTML and files provided by the server. Using this tool, we collected a JSON file with all the technologies and categories used in each website.

SSL certificate: The SSL certificate is the main security measure for websites to ensure data encryption with a strong protocol and provide confidence to the customers [39]. In addition, encryption is key for e-commerce websites, where users enter their personal and financial details [40]. E-commerce websites without an SSL certificate are relegated to the bottom results in search engines and, therefore, fail to attract customers. We exported the information from the SSL certificate and checked whether it was valid or not.

Social media: has become a significant factor in brand marketing campaigns. Their growth has encouraged companies to connect with their customers to strengthen their relationships by having an active profile on these platforms [41]. E-commerce websites develop their image by uploading posts, reviews and photos of their stores and products [42]. In addition, social media has been used discreetly in previous works [7] by checking if the website has social media logos. As they found this to be a crucial feature, in this work, we obtained information on the actual profiles, such as the number of followers, posts or likes. For this task, we used three of the most used

[8] https://www.wappalyzer.com/ Retrieved June 2025.

social media platforms: Facebook, Instagram and X (former Twitter[9]), and looked for the profile name in the announced URL and queried it on the corresponding API to obtain this information. We also retrieved the data from the review platform Trustpilot, such as the overall score and the total number of votes.

Text: E-commerce websites typically present pages with terms and conditions or shipping information. To the best of our knowledge, this resource has never been used in previous works, and it can be valuable for Natural Language Processing (NLP) tasks because it is redacted text. We reviewed the pages on different terms and conditions and retrieved the corresponding HTML.

Offline copy: To collect as much information as possible from the website, we obtain an offline copy of the website using the WGET command to download all the resources needed to render the website in offline mode. This data includes images, external styles and all the necessary files. As with the text resource, to the best of our knowledge, no other research is using this resource, and it could be useful for detecting phishing kits [43] or automated tools for creating these fraudulent websites.

Although other relevant information, such as WHOIS, was an influential resource in previous works, we did not collect it. Due to the General Data Protection Regulation (GDPR), most of the WHOIS information available for Spanish domains is redacted or hidden for privacy reasons, biasing the overall recollection of this resource. On top of that, WHOIS information availability is under consideration in many countries [44], and it could be censored in the coming years. Finally, state-of-the-art works that rely on this information will remain inconsistent in their predictions.

We proposed a directory structure to store all samples and the data. Figure 2 depicts the proposed organization implemented in every collected sample.

3.3 Recollection

To automatically recollect and organize all the samples and resources, we developed a Python3 application to visit the target domains from"Confianza Online"[10] and the periodic reports from INCIBE. Legitimate samples were collected from December 2021 to May 2022, while fraudulent websites were collected between November 2021 and November 2022. We used a set of libraries, tools and APIs to extract the information and complete the process. Figure 3 depicts the complete procedure with the following steps:

Firstly, we obtained the list of domains from the services as mentioned earlier. They were divided into two different lists according to their class. For the fraudulent websites, the lists were obtained every 15 days; therefore, we executed the recollection system as soon as we received the report to avoid visiting an empty or seized domain. Due to the small number of sites in these lists, the collection process for the fraudulent class was extended up to one year.

[9] To maintain consistency with the dates in our dataset, we will refer to X as Twitter throughout the remainder of this paper.

[10] https://www.confianzaonline.es/ Retrieved June 2025.

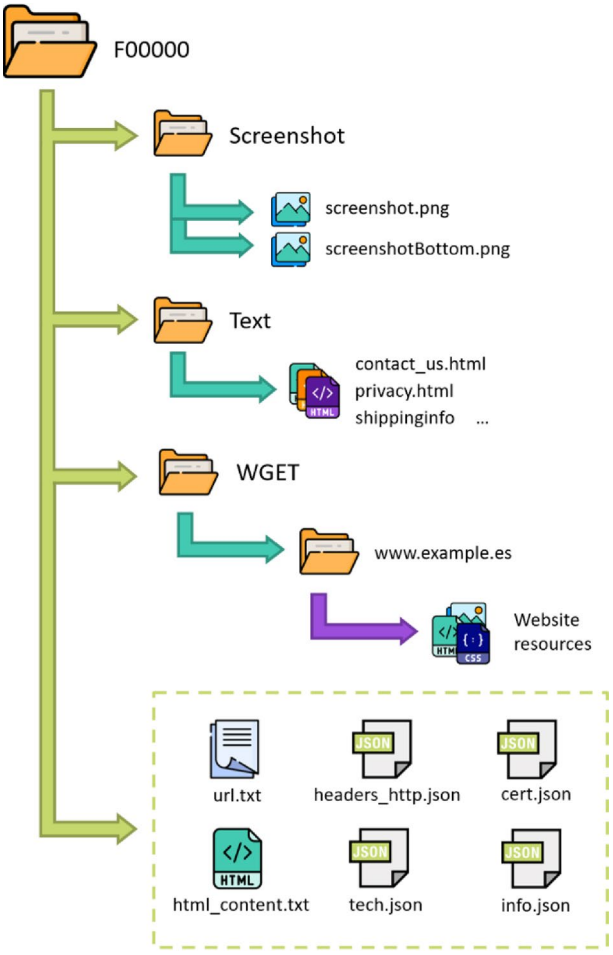


Fig. 2 Proposed organization implemented for every collected sample after fetching aforementioned resources

Secondly, we visited each domain, loading its content into a controlled browser environment. Using Selenium Web Drive capabilities, we extracted the URL, the HTML and the rest of the terms and conditions of web pages.

Then, we interacted directly with the server to obtain the HTTP headers using the requests library, the local instance of Wappalyzer captured the result from the domain, and finally, the SSL socket retrieved the certificate information.

Next, the domain name, was used to obtain the score and number of reviews from Trustpilot. For the social media information, we fetched the links from the HTML and gathered the username to call the corresponding API and obtain the metrics. Links with an invalid user format, different from that provided by the services, were discarded.

Finally, all information was arranged using the structure defined in Fig. 2.

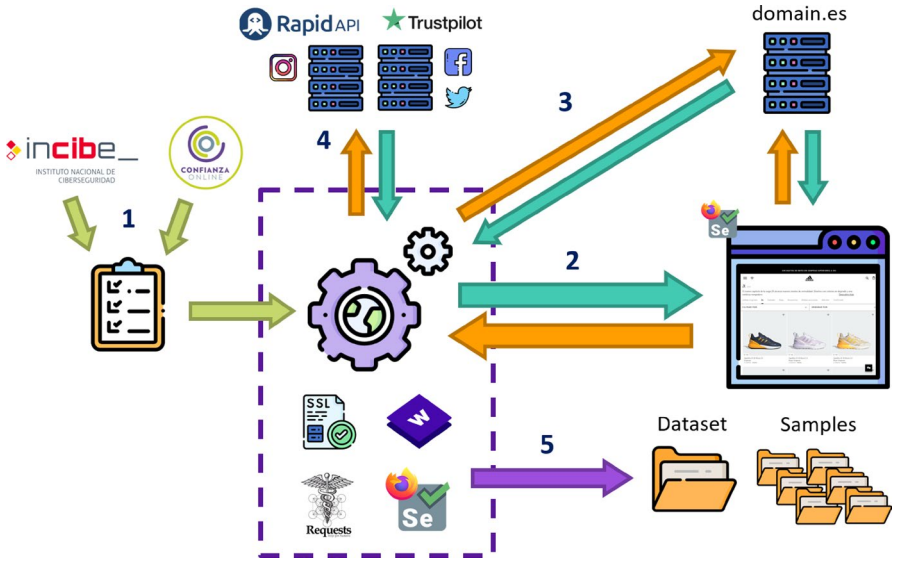


Fig. 3 The recollection process starts by obtaining the domain names (1), then the websites were visited (2) and the information was collected (3). Using the links in the website, external APIs were used to obtain the remaining resources (4). Finally, all the data was stored locally (5)

3.4 Filtering and final samples

The dataset used is an essential part of this study as it must accurately represent the samples. Firstly, the fraudulent samples were collected from manually labelled domains identified as fraudulent by cybersecurity experts, thus defining a high-quality representative set. Secondly, in the legitimate class, domains were retrieved from the entity"Confianza Online,"and there was a set of out-of-scope samples, such as those that were not e-commerce websites. To solve this issue, collected legitimate samples were manually inspected to determine whether they were online shops.

We provided all the collected samples and created a metadata attribute *banned* in the sample metadata file to indicate whether the samples were omitted from the training dataset. Out of the 1668 legitimate samples, we found that 19 were repeated and 357 were not e-commerce websites. Therefore, those samples were banned and were not taken into account in any of the experiments. Fraudulent websites were also manually inspected to ensure that all the websites matched the reported domains. The final dataset comprises 1292 legitimate e-commerce websites and 739 fraudulent ones. Although our analysis is language-independent, we observed that among legitimate e-commerce websites, approximately 92.34% are in Spanish, 4.49% in English, and 3.17% in other languages. While fraudulent e-commerce websites are predominantly in Spanish (79.29%), followed by English (15.97%) and other languages (4.74%).

The categories identified in the analysis are defined as follows: the **Automotive** category includes products and services related to vehicles, such as parts, accessories, and maintenance tools. **Education** covers goods and services intended for learning

purposes, including books, online courses, and educational materials. **Entertainment** refers to items and services designed for leisure and amusement, such as streaming platforms, games, and event tickets. **Fashion** involves apparel, footwear, accessories, and related fashion products. **Food** includes edible products and beverages offered for human consumption. Health represents goods and services related to personal care, wellness, and medical products. **Home** comprises household items, furniture, and home improvement products. **Marketplace** involves online platforms and services that facilitate the buying and selling of various products and shopping. **Office and Industrial Material** refers to equipment, tools, and supplies used in office environments or industrial production. **Pets** includes products intended for pet care, such as food, accessories, and grooming items. **Sport** consists of sports equipment, apparel, and services related to fitness and recreational activities. Technology represents electronic devices, software, and digital services. **Toys** includes playthings and entertainment products designed primarily for children. The distribution of these categories in fraud and legitimate websites is shown in Table 2.

4 Methodology

In this study, we have used six different resources for feature extraction: URL, HTML, technology analysis, social media impact, SSL certificate and HTTP headers. The objective of the features and their design is to reflect the differences between the two classes. To maximize performance, we combined the best features from the state-of-the-art and proposed a novel set to improve the performance of the algorithm on current websites.

Each resource was introduced in a Python3 module that was responsible for extracting the associated features. Once all the features are defined, we run a Python3 script over all valid samples (not banned). Each sample went through the six developed modules (one per group), where a JSON structure was built using the name of the feature as the key and the result of the extraction as its value.

Table 2 Comparison of category count and percentage between the legitimate and fraudulent websites

Category	Fraud (%)	Legit (%)
Automotive	5 (0.68%)	45 (3.48%)
Education	0 (0.00%)	75 (5.80%)
Entertainment	3 (0.41%)	62 (4.80%)
Fashion	310 (41.95%)	179 (13.86%)
Food	2 (0.27%)	100 (7.74%)
Health	11 (1.49%)	142 (10.99%)
Home	19 (2.57%)	208 (16.10%)
Marketplace	231 (31.27%)	142 (10.99%)
Office and industrial material	6 (0.81%)	56 (4.33%)
Pets	1 (0.14%)	19 (1.47%)
Sport	105 (14.21%)	65 (5.03%)
Technology	17 (2.30%)	176 (13.62%)
Toys	29 (3.92%)	23 (1.78%)
Total	**739**	**1292**

Values representing the total are shown in bold

Then, an n-dimensional feature vector is generated for each resource; for instance, $U_v = [U_1, U_2, ..., NU_{8.1}]$, $H_v = [H_{1.1}, H_{1.2}, ..., H_3]$, $Y_v = [Y_1, NY_2, ..., NY_9]$ and $T_v = [NT_1, NT_2, ..., NT_{11}]$. Finally, the six vectors are concatenated to obtain the complete feature vector for a specific sample: $F_v = [U_v, H_v, Y_v, T_v]$. After all the samples had been processed, they were arranged into an X matrix with *MxN* dimension, where *M* corresponds to the number of features and *N* is the number of total samples of the experiment.

We extracted 50 features distributed across the resources indicated below.

4.1 URL

URLs have been used in previous studies [7, 13], and it is one of the principal resources for similar tasks like phishing detection [29, 30]. To obtain the different features, we parsed the URL into parts, as shown in Fig. 4.

Each of the following sections represents the topic covered and the related features, indicated by a number after the topic itself. For example, *NLP features (5)* denotes five features within that topic. The following features have been designed using the URL as input data.

Number of digits in the domain [29]: Fraudulent URLs usually contain more digits than legitimate websites [45]. We count the number of digits in the domain.

Domain and subdomain length (2) [29]: Companies register short domain names with their brand to promote it to customers. In addition, companies allocate their subdomain names to create a reasonable structure for their site, with names such as"shop","store", or"www". On the other hand, fraudulent websites tend to use subdomains to deceive customers by placing the forged domain of legitimate brands in the subdomain. We obtained the length (in characters) of both URL parts: the domain and the subdomain.

NLP features (5) [29]: We obtained five different metrics from the list of words within the URL. To obtain those words, ASCII symbols were used to split the URL and its parts. The final NLP features extracted are the number of words, the average of their length, the standard deviation and the length of the shortest and longest words in the list.

4.2 HTML

HTML code represents the content shown to the user in the browser and it allocates key information. We use it to extract the following features:

HTML text length [6, 7]: Fraudulent websites tend to be simpler, and their text is usually shorter since the provided details for products and menus are poor. Therefore,

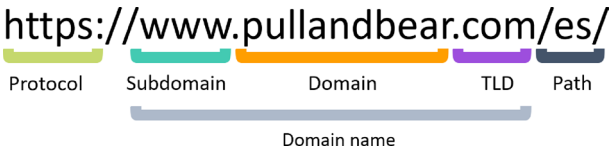


Fig. 4 Parts of a complete URL

we used the number of characters in the HTML text by removing all HTML tags, CSS and JavaScript using *get_text()* function from BeautifulSoup4.[11]

Domain in the title: One of the most suspicious points for fraudulent websites is the mismatch between the forged brand and the domain name [46]. Legitimate sites place their brand name in the title and the domain name, while fraudulent websites are not allowed since they are already registered. Due to this restriction, fraudsters can set the forged brand name in the title but not the domain name. In this feature, we check whether the domain appears in the website title or not.

Domain in the HTML: The domain usually contains the brand name; therefore, if the shop sells products from that brand, the domain is expected to appear multiple times in the HTML code (usually in product names). In Fig. 5, the fraudulent domain name is"salomonzapatos", which is an impersonation of the brand"salomon". When searching in the website text, no words matched the domain, while the name of the brand,"salomon", appeared multiple times on the legitimate website. To extract these features, we used the *get_text()* function from the BeautifulSoup4 library to strip out all tags and links in the HTML, then convert the text to lowercase and count the number of times the domain appears in the text.

Base64 resources: During the sample analysis, we came across many legitimate websites that use base64 encoding to ensure that certain resources are loaded into the website even if the Content Delivery Network (CDN) is unreachable. Most of these resources are icons or logos. This feature checks if the website uses Base64 resources embedded in the HTML.

HREF attributes (5): The HREF attribute is used mainly in two HTML tags: anchor (<a>) and links (<link>), which determine the location of specific resources used by the website. These can be different depending on their location, according to the analyzed website. In Table 3 we show the five types of links studied in this work. External resources are used by fraudulent websites to match the forged domain resources [6]. Internal links point to the same domain, which is the ordinary behav-

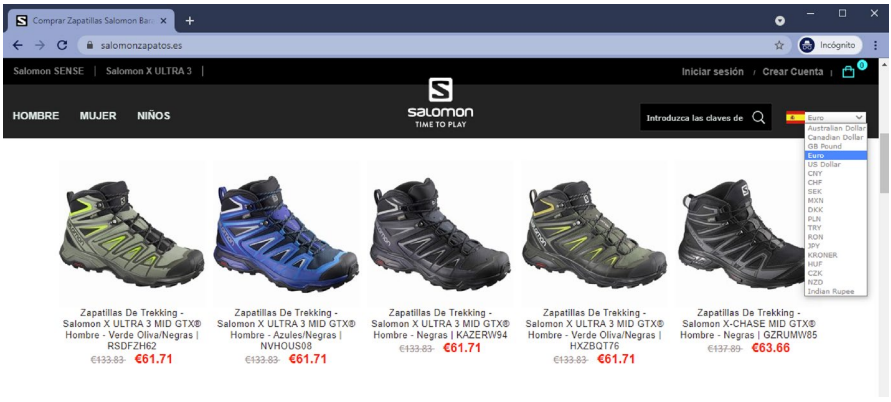


Fig. 5 Fraudulent example impersonated the brand namely"salomon"that contains feature such as repeated prices, currency selector and domain in HTML

[11] https://pypi.org/project/beautifulsoup4/ Retrieved June 2025.

Table 3 Types of links depending on their origin or destination of the href tag

Type of link	Example
External	
Internal	
	
No content	
Empty	
No reference	<a>

iour of legitimate websites, which uses their resources and links to pages within the domain [30]. Other types of links are the empty and null ones, which are used by fraudulent websites to render a link and deceive users with a legitimate look [47]. Furthermore, these links keep users away from leaving the page. Finally, links with no reference were used to cover all the possibilities of the link tags. In these features, we count the number of links for each of the five types described above.

Number of currencies [7, 13, 20]: Most legitimate e-commerce websites focus on a specific region where they sell their products. Therefore, they use only one currency to display the prices. On the contrary, fraudulent websites allow currency selectors for users to choose their currency and distribute website products to most countries, increasing their target number [20]. These selectors are in the top right-hand corner of these websites, as shown in Fig. 5. We count the number of different currencies found in the HTML text from a list of 12 currencies: Dollar, Euro, Rupee, Yuan, Yen, Pound, Ruble, Won, Turkish Lira, Philippine Pound, Czech Koruna and Zloty.

Prices (4) [7, 13, 20]: Fraudulent websites offer bargains to swindle customers [48]. Samples analysis revealed that fraudulent websites tend to repeat prices for most of their products, as stated in [20] and shown in Fig. 5. We implemented a reliable method to design and extract features from the prices. First, we used regular expressions on the HTML text to extract the prices with float and integer values. We use the HTML text instead of the source code because the code contains numbers and CSS properties which could introduce noise into the list of matches due to the use of the $ symbol. Before taking any further action, we analyzed the list to remove values regardless of prices using a regular expression with the aforementioned list of 12 currency symbols. From the price list, we separated the normal prices and the discounts. We used the CSS and HTML code to identify strikethrough prices, specifically using the CSS property called “text-decoration: line-through” and the HTML tag “<del>”, as these were the most common implementations. Finally, we extracted four features from this data: (1) the total number of prices found, (2) the number of repetitions of the mode price [7, 20], (3) the average repetitions for each price and (4) the average discount percentage offered on the page.

Social media links in the HTML (3): Wu et al. [7] stated the importance of the social media links in this task and verified that only 7.75% of the fraudulent websites had at least one link, while the legitimate websites that were the 80.18% had at least one link. In this work, we analyzed social media links and profiles to better understand this matter. We used three of the most used social media platforms in e-commerce activities [49]: Facebook, Instagram and Twitter as Wu et al. [7] and reproduced their analysis on our dataset. Results showed that 21.65% of fraudulent websites had at least one link to social media, significantly higher than the study

mentioned above. We inspected the social media URLs used by these pages. We found that most of them were generic links with no username in the URL or share links, as shown in Fig. 6, which are predefined messages to post on the corresponding platforms with the link of the fraudulent website. After removing these findings, only 2.68% of the fraudulent websites had at least one valid link to a social media profile, compared to the 83.78% of the legitimate websites.

Based on this finding, we proposed three features related to HTML and social media links. Firstly, the number of correctly formed links to social media profiles, i.e., where the shared links were not considered. Secondly, we identified which platforms (Facebook and Twitter) were used to share the website. If the website contains a link to post a tweet, the binary feature for the Twitter sharer is set to one in the case of a shared link or zero in the case of an empty link or a link to the actual user profile. The same behaviour is reproduced with the Facebook sharer link.

4.3 Technologies

Developers use web technologies to improve the performance, scalability and appearance of websites. To detect these technologies, we used Wappalyzer,[12] which can identify up to 1950 different technologies within 71 categories by using the fingerprint generated in the HTML, CSS or JavaScript code. This resource can improve the performance of the algorithms and increase the resistance to bypassing mechanisms

Fig. 6 Example of the tweet and share links found on a fraudulent website for Twitter (on the top) and Facebook (on the bottom)

[12] https://www.wappalyzer.com/ Retrieved June 2025.

since changing the HTML may be effortless for the attacker. Still, the implementation of web technologies might not be. From the Wappalyzer report, we extracted the following features:

Number of technologies: Legitimate e-commerce websites are created by using modern technologies and frameworks to ensure security and ease of maintenance. However, attackers craft fraudulent websites effortlessly and with minimal resources to run the site. To do this, fraudsters use HTML templates and toolkits or copy the HTML code from another legitimate website. Therefore, the number of technologies is limited to the web server and JavaScript libraries. In this feature, we counted the number of technologies detected by Wappalyzer with a 100% confidence.

Technologies categories (5): All technologies are classified according to their purpose, e.g., databases, security, CDN or e-commerce, among others. In this study, we present below the five relevant categories for e-commerce developments.

- E-commerce: Online shops can be built in a number of ways, the easiest being e-commerce frameworks or tools that allow developers and owners to easily manage the platform. Some of the technologies within this category are Shopify, WooComerce, PrestaShop, or OpenCart.
- Live-chat: Many e-commerce websites offer live chat for customers to ask questions to online agents. Fraudulent websites are not interested in the customer experience. Consequently, they do not employ these services. Usually, they are implemented using tools and plugins that run over the website. Some of the technologies in this category are Zendesk, Tawk.io or Intercom.
- Cookie-Compliance: Current websites have to announce and confirm to the user that they are using cookies to obtain information about the actual user. For this task, tools have been developed to pop a banner with the information and obtain the response of the user. Quantcast, Osano and OneTrust are the most common tools for this task.
- Analytics: Legitimate e-commerce websites are interested in increasing sales and customers. To achieve this goal, they analyze the statistics to know about the users who visit the website, such as what they buy or where they live. In this way, marketing campaigns can be targeted to specific sectors and users to display custom advertisements for specific users based on their activities. To implement this functionality, technologies such as Google Analytics or Facebook Pixel help to collect and manage the information and ads. Fraudulent websites are not interested in these services as they require company information to create accounts on these services.
- Payment methods: Payment platforms allow customers to perform payments in safe environments. Some of these platforms are Apple Pay, Visa, American Express, PayPal or Google Pay. A usual practice on fraudulent websites is to display a custom form with the same look and feel as other payment platforms. This way, fraudsters collect and steal user data without processing the payment, and they can use that data to perform illegal transactions for their benefit.**Specific technologies (3)**: Apart from the aforementioned categories, we found that Google Analytics, Google Analytics Enhanced for E-commerce and reCaptcha are technologies implemented in most legitimate websites, while on the fraud class, almost

no website enforces any of these technologies. We defined three binary features whose value is one if the technology is used and zero if it is not.

4.4 SSL Certificate

SSL certificates, which are one of the basic security measures for websites, provide end-to-end encryption and keep all exchanged data private. It was used to improve malicious website detection [31], but in the last few years, it has become a standard among malicious websites. The APWG (Anti-Phishing Working Group) states that 82% of the phishing websites count on an SSL certificate [50] since, currently, attackers can obtain free SSL certificates from entities like Let's Encrypt. Even though certificates are no longer a distinguishing element, websites without them are suspicious to be deceitful; thereby, we extracted two features from the certificate:

Valid SSL certificate: A binary feature where one represents a valid certificate and zero if not.

Number of registered names in the certificate: One single SSL certificate can be used for multiple websites or subdomains. Brands usually certify all their websites with the same certificate. We count the number of domains or subdomains registered under the SSL certificate by examining the"alternative names"field in the certificate.

4.5 HTTP Headers

HTTP headers are essential for web security. Their proper configuration can reduce the attack surface and create a secure exchange between users and the final server. Iv et al. [51] analyzed HTTP headers for identifying malicious websites with 91.05% accuracy, demonstrating the effectiveness of this resource. In this work, we combined four HTTP headers related to the security (Content-Security-Policy, Strict-Transport-Security, X-Content-Type-Options and X-Frame-Options) and the relevant headers used by Iv et al. [51] (Cache-Control and Expect-CT) to identify websites that have an optimal configuration.

4.6 Social media information and reviews (external)

In this work, we arranged a deep analysis of social media platforms by evaluating the metrics and impact of companies' social media accounts. The main advantage of our approach is to differentiate between legitimate accounts with a great number of followers or posts and those created to spoof a brand, which usually lacks followers and activity. We focused on Facebook, Instagram and Twitter, the most used social media in Spain [52] and combined their metrics to obtain an overall media impact. If a domain misses one social media platform, its correspondent metrics are zero. We defined six features to represent the impact of the profile on its social media accounts:

Total followers: We obtained the total amount of followers by summing up the metrics from each social media account found.

Total following: Represents the number of users the profile follows on Instagram and Twitter.

Total posts: We added all the posts published on Instagram and the total tweets posted on the Twitter account.

Facebook likes: The total likes of the company's Facebook profile.

Facebook visits: Recorded visits to the company's Facebook profile in the last 24 h.

Twitter account age: We calculated the months from creating the Twitter account until the current date.

Moreover, we introduced Trustpilot, an online platform to rate and review the experience of customers on different e-commerce websites. These kinds of platforms have a high impact on the reputation of e-commerce websites and are crucial for final users [53]. They also have a bias on the reputation since brands boost their score with fake positive reviews [54], award users for positive reviews [55] or post fake negative reviews on competitors profiles [56]. However, Trustpilot uses algorithms and methods to mitigate bots and deceitful reviews to avoid the previous cases. We tested its effectiveness by including two features collected in our dataset:

Trustpilot score: Trustpilot calculates the average score for a given domain from all the user scores. The score is rated between zero and five.

Trustpilot reviews: To provide a comprehensive analysis of the Trustpilot platform, we counted the total number of reviews for a given domain. This approach establishes the relation between the score and the website reputation since a shop with an average score of 3.0 supported by 2, 000 reviews has more reputation than another store with only one review and an average score of 5.0. Polarized scores cause this since users tend to vote on the extreme cases, either a satisfactory experience rated with five stars or disgraceful incidents with one star [57] (Table 4).

5 Experimentation and results

5.1 Experimental setup

We tested an Intel Core i3 9100F at 3.6 GHz and 16 GB of DDR4 RAM. We used scikit-learn[13] and Python 3 for the implementation of the different experiments and the creation of the machine learning models.

We have used nine different classification algorithms used in the state-of-the-art [6, 18, 19] to test and compare their performance on the design features, including eXtreme Gradient Boosting (XGBoost) and Gradient Boosting Classifier (GBC), Random Forest (RF), k-Nearest Neighbour (kNN), Support Vector Machines (SVM), Logistic Regression (LR), Naïve Bayes (NB) and Adaboost (ADA). Classifiers were trained using their best-found hyper-parameters determined from a 5-fold cross-validated grid search. Table 5 displays the selected hyper-parameters for the proposed models.

We scaled the features vector using scikit-learn's *StandardScaler* over the complete set of training samples and features, then applied the obtained scaler to the test samples.

[13] https://scikit-learn.org/stable/ Retrieved June 2025.

Table 4 Summary of the implemented features and their corresponding group

#	Group	Feature	Value	Description
U1	URL	domain_digit_count	D	# digits in the domain name
U2.1	URL	domain_length	D	# characters in the domain name
U2.2	URL	subdomain_length	D	# characters in the subdomain
U3.1	URL	raw_word_count	D	# words in the URL
U3.2	URL	average_word_length	C	Average length of words in the URL
U3.3	URL	longest_word_length	D	Longest word length in the URL
U3.4	URL	shortest_word_length	D	Shortest word length in the URL
U3.5	URL	std_word_length	C	Standard deviation of words length
H1	HTML	text_length	D	# characters in the HTML text
NH2	HTML	domain_title	B	The domain appears in the title
NH3	HTML	domain_in_html	D	# times the domain appears in the HTML text
NH4	HTML	base64	B	Website loads resources in base64
H5.1	HTML	link_int	D	# internal links
H5.2	HTML	link_ext	D	# external links
H5.3	HTML	link_#	D	# empty links
H5.4	HTML	link_emp	D	# null links
H5.5	HTML	link_null	D	# links without href attribute
H6	HTML	currencies	D	# currencies detected on the website
NH7.1	HTML	prices	D	Total prices detected in the website
H7.2	HTML	most_times	D	Repetitions of the mode price
NH7.3	HTML	avg_times	C	Average repetitions of the prices
NH7.4	HTML	avg_discount	C	Average discounts on the prices
NH8.1	HTML	num_social_html	D	# links to social media
NH8.2	HTML	fake_fb	B	Link to share website on Facebook
NH8.3	HTML	fake_tw	B	Link to share website on Twitter
NT1	Tech	n_tech	D	# technologies detected
NT2.1	Tech	e-commerce	D	# E-commerce technologies used
NT2.2	Tech	live-chat	D	# live chats technologies used
NT2.3	Tech	cookie-compliance	D	# cookies technologies used
NT2.4	Tech	analytics	D	# analytics technologies detected
NT2.5	Tech	payment-processors	D	# payment platforms detected
NT3.1	Tech	google-analytics	B	Website uses Google Analytics
NT3.2	Tech	google-analytics-enh	B	Website uses Google Analytics for e-commerce
NT3.3	Tech	recaptcha	B	Website uses reCaptcha for bot avoidance
S1	SSL	has_cert	B	The domain uses a valid SSL
S2	SSL	n_name	D	# domain names registered in the SSL certificate
NP1	HTTP	content-security-policy	B	Website defines CSP header
NP2	HTTP	strict-transport-security	B	HSTS is implemented in the website
NP3	HTTP	x-content-type-options	B	\textit{Nosniff} directive is set
NP4	HTTP	x-frame-options	B	Use \textit{deny} or \textit{sameorigin} directives
P5	HTTP	cache-control	B	Website does not use \textit{post-check} outdated directive
P6	HTTP	expect-ct	B	Header is configured in the website

Table 4 (continued)

#	Group	Feature	Value	Description
NM1	External	total_followers	D	Total followers on social media
NM2	External	total_following	D	Total following on Instagram and Twitter
NM3	External	total_posts	D	Total posts on Instagram and Twitter
NM4	External	fb_likes	D	Likes on Facebook website
NM5	External	fb_visits	D	# visits on Facebook website in last 24 h
NM6	External	tw_age	D	Months since Twitter account registration
NM7	External	trustpilot_score	C	Trustpilot review score
NM8	External	trustpilot_reviews	D	# of reviews in Trustpilot

The value represents the kind of feature: D for discrete, C for continuous and B for binary. In favor of simplicity, the symbol"#"is treated as"number of"

Table 5 Evaluation of machine learning classifiers for the proposed methods

Classifier	Hyper-parameter	Value
XGBoost	eval_metric	error
	n_estimators	120
	objective	binary
	scale_pos_weight	2
GBC	learning_rate	0.1
	max_depth	3
	max_features	sqrt
	n_estimators	242
RF	max_features	auto
	n_estimators	127
kNN	metric	manhattan
	n_neighbors	2
	weights	uniform
SVM	C	100
	gamma	0.001
	kernel	rbf
LR	C	0.1
	penalty	l2
Adaboost	learning_rate	0.1
	n_estimators	43
Naive Bayes	kind	BernoulliNB

Results are indicated in %

Finally, we estimated the classifier performance using the k-fold cross-validation technique, with the following settings: $k = 5$, $shuffle = True$ and $random\_state = 42$.

5.2 Performance metrics

We used the averaged values from the 5-fold cross-validation, reporting the accuracy (Eq. (3)), the precision (Eq. (1)), the recall (Eq. (2)) and the F1-Score (Eq. (4)) [7, 8, 19]. TP denotes the true positives, i.e., how many fraudulent websites were correctly classified. FP refers to the false positives and represents the number of legitimate

samples wrongly classified as fraudulent. TN (i.e., the true negatives) denotes the number of legitimate samples correctly classified. Finally, FN represents the false negatives that represent the number of fraudulent websites misclassified as legitimate ones.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

5.3 Comparing proposed models

The objective of this work is to generate a machine-learning model capable of identifying fraudulent e-commerce websites to advise users about suspicious websites that might lead to fraud.

In this experiment, we evaluated two different approaches. The first one aimed to optimize performance by using all the designed features, including those that were obtained from external services. The second approach focuses on creating a standalone model that works only with features obtained from local resources, i.e., without external features.

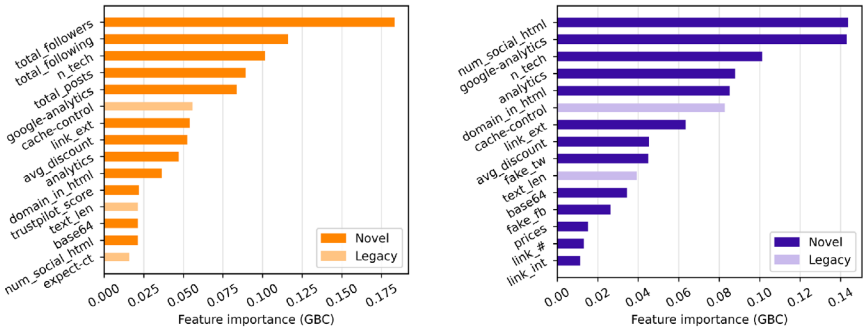
According to the results of Table 6, in the full set of features, XGBoost had the best classification performance in terms of F1-Score (0.9688), followed by GBC

Table 6 Evaluation of machine learning classifiers for the proposed methods

Algorithm	Full set				Standalone			
	Precision	Recall	F1-Score	Accuracy (%)	Precision	Recall	F1-Score	Accuracy (%)
XGBoost	0.9778	0.9601	**0.9688**	97.78	0.9647	0.9660	**0.9653**	97.49
GBC	0.9751	0.9619	0.9684	97.73	0.9590	0.9686	0.9637	97.39
Random Forest	0.9847	0.9483	0.9661	97.59	0.9765	0.9342	0.9546	96.80
SVM	0.9622	0.9602	0.9611	97.19	0.9619	0.9618	0.9618	97.24
LR	0.9566	0.9633	0.9599	97.09	0.9535	0.9576	0.9555	96.80
kNN	0.9564	0.9366	0.9463	96.21	0.9337	0.9481	0.9407	95.72
AdaBoost	0.9373	0.9534	0.9452	96.01	0.9444	0.9454	0.9448	96.01
NB	0.9231	0.9412	0.9320	94.84	0.9286	0.9346	0.9316	94.84

Values representing the highest performance for each F1-Score are shown in bold

Results are indicated in %



(a) Full set model (b) Standalone model

Fig. 7 Feature importance for the presented methods: **a** full set model and **b** standalone model. In light colors, legacy features from previous works; in darker colors, novel features are proposed in this work

(0.9684) and Random Forest (0.9661). These results are interpreted as acceptable for a fraudulent website detection system, which can correctly classify up to 97.78% of the total samples (accuracy), in the case of the XGBoost algorithm. The difference between the top two performers was minimal, but Random Forest showed a different behaviour, increasing the precision by 0.0069 and reducing recall by up to 0.0118. Therefore, it is suggested for systems where higher sensitivity to these attacks is needed.

Comparing the two different approaches, the standalone version performed above expectations. XGBoost obtained the best results with 0.9653 F1-Score, 0.0035 underneath the XGBoost model with the full feature set. Standalone models manifest a lower precision and higher recall against the full set method. This implies a higher awareness when detecting fraudulent websites and a higher number of false positives (legitimate websites classified as fraud). From these results, we can state that external features are effective for high-sensitivity environments, which aim to detect as many fraud websites as possible, accepting a higher chance of finding legitimate websites among them. On the contrary, the standalone version would be optimal in environments where false positives are not penalized, i.e., legitimate websites are classified as fraud and might lose reputation due to the prediction.

5.4 Importance of the designed features

We used the feature importance coefficient from the generated GBC models[14] to visualize in Fig. 7 the twelve most important proposed features in both methods.

On the full set model, it is noticeable that the three external features, which combine information from the three analyzed social media, are among the top most important ones, proving their value for fraudulent website detection tasks. In addition, three features from the novel technological set are within the top-10, including"Number

[14] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html Retrieved June 2025.

of technologies detected","Google Analytics"and the"analytics"category. To complete the most valuable features, average discounts and the number of external links extracted from the HTML showed great performance. It is worth noting that none of the URL features made it to the top, and they were only relevant for detecting the domain name in the HTML.

On the side of the standalone model, the number of social media links replaced the top one, followed by the technological features. As shown in previous results, they hold great results even after removing valuable external features. Furthermore, we consider this set of features to tolerate the strategy of attackers strategies to avoid detection since their mitigation implies workload and exposure due to the data required to implement analytics on the website. Other HTML features entered the top, detecting X (former Twitter) sharer links, the number of prices and other link-related ones. As seen in the full set, none of the URL features reached the top.

Finally, it is worth noting that novel features presented in this work ranked higher than the legacy ones, as depicted in Fig. 7. Therefore, designed features contribute to the fraudulent website detection problem.

5.5 Comparing resources for fraudulent website detection

Once we analyzed individual features, we evaluated the performance of the algorithms with different input resources. The objective of these experiments is to identify the most valuable resource when classifying fraudulent websites.

The first experiment excludes different subsets of features defined in Table 4. Figure 8 displays the results obtained for XGBoost, GBC and Random Forest classifiers. The XGBoost algorithm performed best on the complete set of designed features (0.9687 F1-Score). When training and testing algorithms without the URL group, the results barely decreased. This proves the poor performance of the proposed URL features. One of the main reasons is the syntax of fraudulent website URLs, which looks legitimate since attackers do not use long subdomains, combo-squatting or typo-squatting to deceive users, like phishing detection. Most of the collected fraudu-

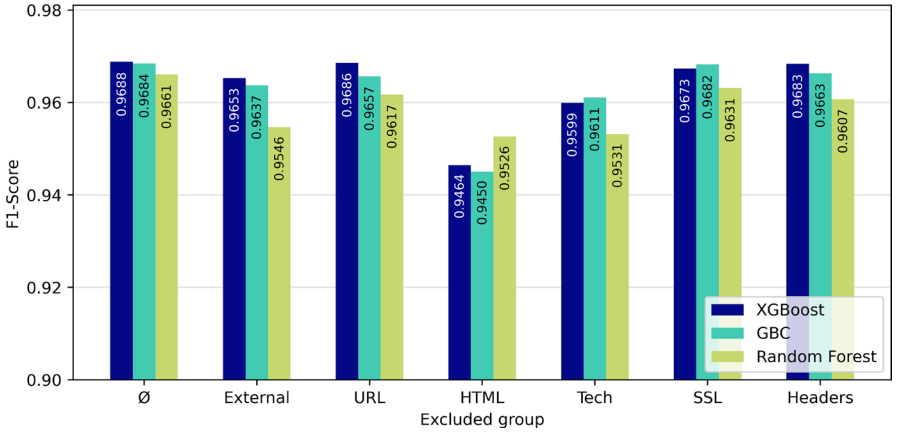


Fig. 8 Results for best performance algorithms when omitting one of the available resources

lent websites use regular domain names for their custom shops, while a minor set of them used the impersonated brand as part of their domain. When we arranged the tests without the HTML group of features, we recorded the worst performance, proving its importance as a resource in this task. While XGBoost dropped 0.0191 F1-Score, Random Forest obtained the best result, 0.9521 F1-Score, only 0.0135 below its result using the complete set. Excluding the technology set of descriptors also decreased performance notably, stating their importance in reaching better performance. Finally, excluding SSL or HTTP Headers did not drastically impact algorithm performance, and they could be ignored in the case where those resources are unavailable.

The objective of the second experiment is to illustrate the effectiveness of the different sets of proposed features when used isolated from the rest of the groups. The 17 HTML features obtained the best result, with 0.9541 and 0.9564 F1-Score for XGBoost and Random Forest, respectively. It is worth noting that HTML features require the URL resource since seven out of 17 features are related to the domain name. Thus, it is recommended for systems with a minimum number of resources. The model generated from the eight external features came in second place with 0.8667 and 0.8662 F1-Score for GBC and Random Forest, respectively. These results confirm the value of the eight designed features in this group. The main drawback is their dependence on external parties, therefore it cannot work if any of the services are down. The nine features extracted from the Wappalyzer technology report got 0.8329 and 0.8333 F1-Score for XGBoost and GBC, respectively, which are still acceptable for general-purpose systems. The URL, SSL and HTTP Headers set obtained low results in this experiment for different reasons. First, the URL set demonstrates its lack of information when classifying legitimate and fraudulent websites due to the similarities in the domain name between both classes. It is worth mentioning that we did not use keywords or brand lists to design features since they could generate a language-dependent model. The SSL set was composed of two features; therefore, its bad results are justified due to the lack of information inputted into the model. Finally, the HTTP Header set obtained above-average results for the last three sets (0.7740 F1-Score for the GBC algorithm). Nevertheless, the main problem resides in its high sensitivity and false positive rate (0.9094 recall and 0.6746 precision on GBC) Figure 9.

5.6 Comparing of proposed methods against existing techniques

in this section, we compare the proposed techniques with Wu et al. [7] and Wadleigh et al. [20] works. Comparisons to other works are challenging because none published their data, or it is restricted to fit their method. Additionally, state-of-the-art works lack detail on the feature and methodology implementation, this forced us to design our own extraction, which could be sightly different, but as accurate as possible. Last but not least, these works include third-party features unavailable in Europe due to GDPR restrictions. A list of not implemented features is provided in Table 7

In order to provide a fair comparison, we will compare these methods against our standalone version with no third party data. Also, we have used the same experimen-

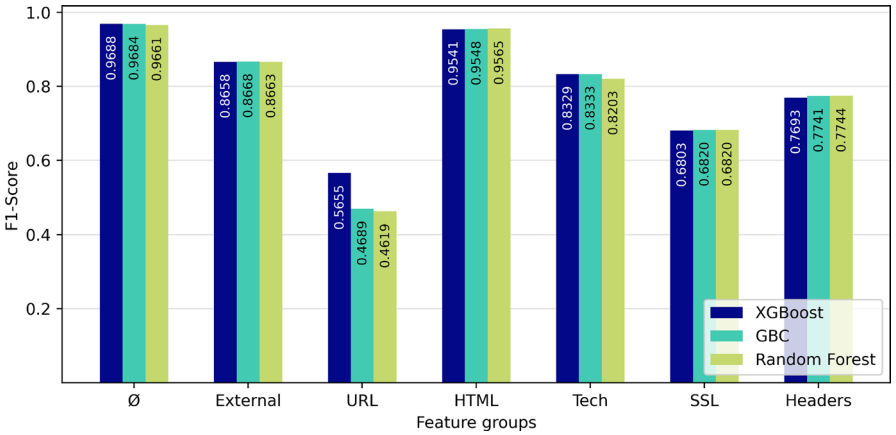


Fig. 9 Results for best performance algorithms when using an isolated group of features corresponding to the resources used in this work

Table 7 Features not implemented in the comparison

Work	Feature	Reason
Wadleigh et al. [20]	Private or China WHOIS WHOIS Registration < 1 Year	No WHOIS data is publicly available for most EU websites
	Website on Takedown Page Website in Alexa Top 100K	Our dataset contains no seized websites, only working ones Costly API.
Wu et al. [7]	in_top_one_million china_registered under_a_year	No WHOIS data is publicly available for most EU websites

Table 8 Comparison between our method and existing works

Work	Classifier	Precision	Recall	F1-Score	Accuracy
Our standalone method	XGBoost	**0.9647**	**0.9660**	**0.9653**	**95.49**
Wu et al. [7]	RF	0.9224	0.8795	0.9003	92.91
Wadleigh et al. [20]	XGBoost	0.6599	0.7715	0.7111	77.25

Values representing the highest performance for each metric are shown in bold

tal setup to find the best hyper-parameters since no details were provided on these works.

As depicted in Table 8, our method outperforms current state-of-the-art works. Furthermore, the proposed features are independent of any external data, which ensures their operation over time in any country since it is also independent of languages and brand lists.

6 Conclusions and future works

In this paper, we present a fraudulent website detection system by designing a novel set of features and using a comprehensive set of resources from websites. This system relies on six different resources, five of which are obtained from actual websites, and the last one is assisted by external resources.

We used legacy and novel features extracted from the URL, the HTML code, the SSL certificate, external resources, and two novel categories proposed in this work: the technologies used by the website and the HTTP header focused on security. In the external resources, we proposed using social media metrics such as followers, profile activity, and reviews, which were proven to increase model performance.

To meet the different requirements for practical use, we have proposed two systems, the main one focused on maximizing accuracy. The second one was designed to work independently from external services, allowing its usage in real and scalable environments to protect users against fraudulent websites. According to experimental results, proposed approaches obtained 0.9688 and 0.9653 F1-Score, respectively, using the XGBoost algorithm.

Additionally, we verified the performance of the novel features proposed among all different groups. The social media analysis and the technologies obtained the highest rank in the feature importance analysis, followed by proposed features related to prices and discounts. It is worth noting that the proposed features are difficult to evade due to the efforts required to simulate a legitimate website in those terms.

Due to the scarcity of publicly available data, we created a manually verified dataset with extensive data collection. This dataset comprises 2031 e-commerce websites from both classes, legitimate and fraudulent. This dataset provides a wide variety of data for researchers to benchmark their approaches independently of the resources they use. This data includes URL, HTML, Screenshots, Technology analysis, HTTP headers, SSL certificate, social media information, text pages, and an offline copy of the website and its resources. The dataset will be publicly available at the request of researchers to develop and compare their approaches.

Although the obtained results accomplished the objective in terms of accuracy, there is room for improvements in future works, specifically in data extracted from URLs. To maximize their potential, other learning techniques, such as deep learning, can be implemented.

Additionally, we left behind valuable resources on our dataset, like screenshots, extensive text files such as policies, and other website resources such as files, source code, and other content. These can improve the robustness of features or can be used to design a new approach with deep learning techniques. Furthermore, this dataset can also be used to detect counterfeit products or brands along with fraud detection, which is of great value for government investigations.

Finally, dataset enlargement is crucial for improving model reliability. Deep learning techniques might find this dataset insufficient due to the number of samples. Active learning or pre-trained models can be proposed to overcome this issue with the current dataset size. Furthermore, as new fraudulent websites appear, they will be included in the dataset to keep models updated with the latest trends and to ensure that obtained results can be transposed to practical environments.

The limitations of this work are mainly related to the likely underutilization of available dataset resources, the limited exploitation of URL-based features, and the dataset size:

1. While the results obtained in this study successfully met the objectives in terms of accuracy, there remains significant room for improvement, particularly in the exploitation of data extracted from URLs. Future work could explore more advanced learning techniques, such as deep learning, to fully leverage these features and uncover more complex patterns.
2. Moreover, several valuable resources available in our dataset were not utilized in this initial study. These include webpage screenshots, full-text documents (e.g., privacy policies), and other embedded website elements such as downloadable files, source code, and multimedia content. These components present an opportunity to enhance the robustness of existing features or to support the development of novel approaches, particularly those involving deep learning and multimodal analysis.
3. The dataset also holds potential for broader applications beyond fraud detection, such as the identification of counterfeit products and brand impersonation. This could be of particular interest to regulatory agencies and law enforcement for investigative purposes.
4. Lastly, expanding the dataset is essential to improving model reliability and generalization. The current dataset size may be insufficient for training deep learning models effectively. To address this, future work may incorporate active learning strategies or leverage pre-trained models to mitigate the limitations posed by data scarcity. Additionally, the continuous integration of newly discovered fraudulent websites will help keep the dataset up to date, ensuring that detection models remain effective against evolving threats and are suitable for deployment in real-world environments.

7 Supplementary information

Pages collected in the dataset can contain malicious files or code in the HTML, data is provided as it is, use it at your own risk. An alternative dataset with no WGET files can be provided to reduce the size and risk.

Acknowledgements This work has been funded by the Recovery, Transformation, and Resilience Plan, financed by the European Union (Next Generation), thanks to the LUCIA project (Fight against Cybercrime by applying Artificial Intelligence) granted by INCIBE to the University of León.

Funding Open access funding provided by FEDER European Funds and the Junta de Castilla y León under the Research and Innovation Strategy for Smart Specialization (RIS3) of Castilla y León 2021-2027.

Data availibility Data availability under request.

Declarations

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

References

1. Weng, H., Li, Z., Ji, S., Chu, C., Lu, H., Du, T., & He, Q. (2018). Online E-commerce Fraud: A Large-scale Detection and Analysis, pp. 1441–1452. https://doi.org/10.1109/ICDE.2018.00162
2. Statista. (2024). Retail e-commerce sales worldwide from 2014 to 2027. Retrieved October 14, 2024, from https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/
3. Ali, M. A., Azad, M. A., Parreno Centeno, M., Hao, F., & Moorsel, A. (2019). Consumer-facing technology fraud: Economics, attack methods and potential solutions. *Future Generation Computer Systems, 100*, 408–427. https://doi.org/10.1016/j.future.2019.03.041
4. Commission, E. (2020). Survey on scam and fraud experienced by consumers. Retrieved October 25, 2024, from https://commission.europa.eu/system/files/2020-01/factsheet_fraud_survey.final_.pdf
5. Juniper. (2024). Online Payment Fraud: Emerging threats, segment analysis and market forecast 2023 - 2028. Retrieved October 14, 2024, from https://www.juniperresearch.com/researchstore/fintech-payments/online-payment-fraud-research-report?utm_source=juniperpr&utm_campaign=pr1_onlinepaymentfraud_financial_fintech_apr21 &utm_medium=email
6. Mostard, W., Zijlema, B., & Wiering, M. (2019). Combining visual and contextual information for fraudulent online store classification, pp. 84–90. https://doi.org/10.1145/3350546.3352504
7. Wu, K., Chou, S., Chen, S., Tsai, C., & Yuan, S. (2018). Application of machine learning to identify counterfeit website, pp. 321–324. https://doi.org/10.1145/3282373.3282407
8. Maktabar, M., Zainal, A., Maarof, M. A., & Kassim, M. N. (2018). Content based fraudulent website detection using supervised machine learning techniques. *Advances in Intelligent Systems and Computing, 734*, 294–304. https://doi.org/10.1007/978-3-319-76351-4_30
9. Wabeke, T., Moura, G. C. M., Franken, N., & Hesselman, C. (2020). Counterfighting counterfeit: Detecting and taking down fraudulent webshops at a cctld. In A. Sperotto, A. Dainotti, & B. Stiller (Eds.), *Passive and Active Measurement* (pp. 158–174). Springer.
10. Buber, E., Diri, B., spsampsps Sahingoz, O. (2018). NLP based phishing attack detection from URLs, pp. 608–618. https://doi.org/10.1007/978-3-319-76348-4_59
11. Bitaab, M., Cho, H., Oest, A., Lyu, Z., Wang, W., Abraham, J., Wang, R., Bao, T., Shoshitaishvili, Y., & Doupé, A. (2023). Beyond phish: Toward detecting fraudulent e-commerce websites at scale, pp. 2566–2583. https://doi.org/10.1109/SP46215.2023.10179461
12. Rodrigues, V. F., Policarpo, L. M., da Silveira, D. E., da Rosa Righi, R., da Costa, C. A., Barbosa, J. L. V., Antunes, R. S., Scorsatto, R., & Arcot, T. (2022). Fraud detection and prevention in e-commerce: A systematic literature review. *Electronic Commerce Research and Applications,56*, 101207. https://doi.org/10.1016/j.elerap.2022.101207
13. Carpineto, C., & Romano, G. (2017). Learning to detect and measure fake ecommerce websites in search-engine results, pp. 403–410. https://doi.org/10.1145/3106426.3106441
14. Gopal, R. D., Hojati, A., & Patterson, R. A. (2022). Analysis of third-party request structures to detect fraudulent websites. *Decision Support Systems, 154*, 113698. https://doi.org/10.1016/j.dss.2021.113698

15. Janavičiūtė, A., Liutkevičius, A., Dabužinskas, G., & Morkevičius, N. (2024). Experimental evaluation of possible feature combinations for the detection of fraudulent online shops. *Applied Sciences, 14*(2), 919. https://doi.org/10.3390/app14020919
16. Kotzias, P., Roundy, K., Pachilakis, M., Sanchez-Rola, I., & Bilge, L. (2023). Scamdog millionaire: Detecting e-commerce scams in the wild. In Proceedings of the 39th Annual Computer Security Applications Conference. ACSAC '23, (pp. 29–43). Association for Computing Machinery, New York, USA. https://doi.org/10.1145/3627106.3627184
17. Xie, S., Liu, L., Sun, G., Pan, B., Lang, L., & Guo, P. (2023). Enhanced e-commerce fraud prediction based on a convolutional neural network model. *Computers, Materials & Continua,75*(1), 1107–1117. https://doi.org/10.32604/cmc.2023.034917
18. Beltzung, L., Lindley, A., Dinica, O., Hermann, N., & Lindner, R. (2020). Real-Time detection of fake-shops through machine learning, pp. 2254–2263. https://doi.org/10.1109/BigData50022.2020.9378204
19. Khoo, E., Zainal, A., Ariffin, N., Kassim, M.N., Maarof, M.A., spsampsps Bakhtiari, M. (2021). Fraudulent e-commerce website detection model using HTML, text and image features. Advances in Intelligent Systems and Computing 1182 AISC, pp. 177–186. https://doi.org/10.1007/978-3-030-49345-5_19
20. Wadleigh, J., Drew, J., & Moore, T. (2015). The E-commerce market for "lemons": identification and analysis of websites selling counterfeit goods, pp. 1188–1197. https://doi.org/10.1145/2736277.2741658
21. Zhou, S., Ruan, L., Xu, Q., & Chen, M. (2023). Multimodal fraudulent website identification method based on heterogeneous model ensemble. *China Communications, 20*(5), 263–274. https://doi.org/10.23919/JCC.fa.2022-0234.202305
22. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),1398*, pp. 137–142. https://doi.org/10.1007/s13928716
23. Brill, E. (1992). A simple rule-based part of speech tagger. In Third Conference on Applied Natural Language Processing, Trento, Italy, pp. 152–155. https://doi.org/10.3115/974499.974526. https://www.aclweb.org/anthology/A92-1021
24. Kamble, N., & Mishra, N. (2024). Hybrid optimization enabled squeeze net for phishing attack detection. *Computers & Security,144*, 103901. https://doi.org/10.1016/j.cose.2024.103901
25. Martínez-Mendoza, A., Jáñez-Martino, F., Carofilis, A., Fernández-Robles, L., Alegre, E., & Fidalgo, E. (2024). Towards multi-class smishing detection: A novel feature vector approach and the smishing-4c dataset. In CEUR Workshop Proceedings, vol. 3846, pp. 58–68. CEUR-WS, ??? http://ceur-ws.org/Vol-3846/paper-07.pdf
26. Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., & Alegre, E. (2023). Classifying spam emails using agglomerative hierarchical clustering and atopic-based approach. *Applied Soft Computing*. https://doi.org/10.1016/j.asoc.2023.110226
27. Saraswathi, P., Anchitaalagammai, J. V., & Kavitha, R. (2023). A system review on fraudulent website detection using machine learning technique. *SN Computer Science, 4*(6), 702. https://doi.org/10.1007/s42979-023-02084-6
28. Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications, 117*, 345–357. https://doi.org/10.1016/j.eswa.2018.09.029
29. Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W. (2019). A stacking model using URL and HTML features for phishing webpage detection. *Future Generation Computer Systems, 94*, 27–39. https://doi.org/10.1016/j.future.2018.11.004
30. Rao, R. S., Vaishnavi, T., & Pais, A. R. (2020). Catchphish: Detection of phishing websites by inspecting urls. *Journal of Ambient Intelligence and Humanized Computing,11*(2), 813–825. https://doi.org/10.1007/s12652-019-01311-4
31. Marchal, S., François, J., State, R., & Engel, T. (2014). Phishstorm: Detecting phishing with streaming analytics. *IEEE Transactions on Network and Service Management,11*(4), 458–471. https://doi.org/10.1109/TNSM.2014.2377295
32. Sánchez-Paniagua, M., Fidalgo, E., Alegre, E., & Alaiz-Rodríguez, R. (2022). Phishing websites detection using a novel multipurpose dataset and web technologies features. *Expert Systems with Applications, 207*, 118010. https://doi.org/10.1016/j.eswa.2022.118010

33. Majgave, A. B., & Gavankar, N. L. (2024). Automatic phishing website detection and prevention model using transformer deep belief network. *Computers & Security, 147*, 104071. https://doi.org/10.1016/j.cose.2024.104071
34. Wu, Y., Xu, Y., & Li, J. (2019). Feature construction for fraudulent credit card cash-out detection. *Decision Support Systems, 127*, 113155. https://doi.org/10.1016/j.dss.2019.113155
35. Bozkir, A. S., & Aydos, M. (2020). Logosense: A companion hog based logo detection scheme for phishing web page and e-mail brand recognition. *Computers & Security,95*, 101855. https://doi.org/10.1016/j.cose.2020.101855
36. Zhang, P. (2021). E-commerce products recognition based on a deep learning architecture: Theory and implementation. *Future Generation Computer Systems, 125*, 672–676. https://doi.org/10.1016/j.future.2021.06.058
37. Lavrenovs, A., & Melón, F.J.R. (2018). HTTP security headers analysis of top one million websites, vol. 2018, pp. 345–370. https://doi.org/10.23919/CYCON.2018.8405025
38. Bilgihan, A., Okumus, F., Nusair, K., & Bujisic, M. (2014). Online experiences: Flow theory, measuring online customer experience in e-commerce and managerial implications for the lodging industry. *Information Technology & Tourism,14*(1), 49–71. https://doi.org/10.1007/s40558-013-0003-3
39. Valdez-Juárez, L. E., Gallardo-Vázquez, D., & Ramos-Escobar, E. A. (2021). Online buyers and open innovation: Security, experience, and satisfaction. *Journal of Open Innovation: Technology, Market, and Complexity, 7*(1), 1–24. https://doi.org/10.3390/joitmc7010037
40. Toapanta, S. M. T., Zamora, M. E. C., & Gallegos, L. E. M. (2020). Appropriate security protocols to mitigate the risks in electronic money management. *Smart Innovation, Systems and Technologies, 165*, 65–74. https://doi.org/10.1007/978-981-15-0077-0_7
41. Kian, T. P., Boon, G. H., Fong, S. W. L., & Ai, Y. J. (2017). Factors that influence the consumer purchase intention in social media websites. *International Journal of Supply Chain Management, 6*(4), 208–214.
42. Jeong, S., Lee, J., Park, J., & Kim, C.-K. (2017). The social relation key: A new paradigm for security. *Information Systems, 71*, 68–77. https://doi.org/10.1016/j.is.2017.07.003
43. Castaño, F., Fernañdez, E. F., Alaiz-Rodríguez, R., & Alegre, E. (2023). Phikita: Phishing kit attacks dataset for phishing websites identification. *IEEE Access, 11*, 40779–40789. https://doi.org/10.1109/ACCESS.2023.3268027
44. Layton, R., & Elaluf-Calderwood, S. (2019). *A Social Economic Analysis of the Impact of GDPR on Security and Privacy Practices.* https://doi.org/10.1109/CMI48017.2019.8962288
45. Abbasi, A., Zhang, Z., Zimbra, D., Chen, H., & Nunamaker, J. F., Jr. (2010). Detecting fake websites: The contribution of statistical learning theory. *MIS Quarterly: Management Information Systems, 34*(SPEC. ISSUE 3), 435–461. https://doi.org/10.2307/25750686
46. Ding, Y., Luktarhan, N., Li, K., & Slamu, W. (2019). A keyword-based combination approach for detecting phishing webpages. *Computers & Security,84*, 256–275. https://doi.org/10.1016/j.cose.2019.03.018
47. Gu, X., Wang, H., & Ni, T. (2013). An efficient approach to detecting phishing web. *Journal of Computational Information Systems, 9*(14), 5553–5560. https://doi.org/10.12733/jcis6350
48. Lu, H. Y., Chan, S., Chai, W., Lau, S. M., & Khader, M. (2020). Examining the influence of emotional arousal and scam preventive messaging on susceptibility to scams. *Crime Prevention and Community Safety, 22*(4), 313–330. https://doi.org/10.1057/s41300-020-00098-3
49. Uşaklı, A., Koç, B., & Sönmez, S. (2017). How 'social' are destinations? Examining European DMO social media usage. *Journal of Destination Marketing & Management,6*(2), 136–149. https://doi.org/10.1016/j.jdmm.2017.02.001
50. Group, A.-P.W. (2021). Phishing Activity Trends Report 2Q, Retrieved October 14, 2024, from https://docs.apwg.org/reports/apwg_trends_report_q2_2021.pdf
51. Iv, J.M., Bhansali, D., Gratian, M., & Cukier, M. (2019). A comprehensive evaluation of HTTP header features for detecting malicious websites, pp. 75–82. https://doi.org/10.1109/EDCC.2019.00025
52. Statista. (2021). Social media usage by platform in Spain. Retrieved October 15, 2024, from https://www.statista.com/global-consumer-survey/tool/10/gcs_esp_202103?index=0&absolute=0 &heatmap=0 &missing=0 &rows%5B0%5D=v0443_inte_social &tgeditor=0 &pendo=0
53. Wang, Y., Tariq, S., & Alvi, T. H. (2021). How primary and supplementary reviews affect consumer decision making? Roles of psychological and managerial mechanisms. *Electronic Commerce Research and Applications,46*, 101032. https://doi.org/10.1016/j.elerap.2021.101032
54. Wang, Z., & Chen, Q. (2020). Monitoring online reviews for reputation fraud campaigns. *Knowledge-Based Systems, 195*, 105685. https://doi.org/10.1016/j.knosys.2020.105685

55. Khern-am-nuai, W., Kannan, K., & Ghasemkhani, H. (2018). Extrinsic versus intrinsic rewards for contributing reviews in an online platform. *Information Systems Research, 29*(4), 871–892. https://doi.org/10.1287/ISRE.2017.0750
56. Luca, M., & Zervas, G. (2016). Fake it till you make it: Reputation, competition, and Yelp review fraud. *Management Science,62*(12), 3412–3427. https://doi.org/10.1287/mnsc.2015.2304
57. Lee, S., Lee, S., & Baek, H. (2021). Does the dispersion of online review ratings affect review helpfulness? *Computers in Human Behavior, 117*, 106670. https://doi.org/10.1016/j.chb.2020.106670

Authors and Affiliations

Manuel Sánchez-Paniagua[1] · Eduardo Fidalgo[2,3] · Enrique Alegre[2,3] · Francisco Jáñez-Martino[2,3]

✉ Francisco Jáñez-Martino
francisco.janez@unileon.es

Manuel Sánchez-Paniagua
msancp@unileon.es

Eduardo Fidalgo
eduardo.fidalgo@unileon.es

Enrique Alegre
enrique.alegre@unileon.es

[1] Pentester and Read Team Operator at A2SECURE, Barcelona, Spain

[2] Department of Electrical, Systems and Automatics Engineering, University of León, León, Spain

[3] Researcher at INCIBE (Spanish National Institute of Cybersecurity), León, Spain