International Journal on Advanced Science Engineering Information Technology

Vol.14 (2024) No. 6
ISSN: 2088-5334

A Study on Browser Fingerprinting Uniqueness Using Clustering Methods and Entropy Validation

Vicki Wei Qi Lee [a], Shih Yin Ooi [a,*], Ying Han Pang [a], Kiu Nai Pau [a]

[a] *Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Bukit Beruang, Melaka, Malaysia*
Corresponding author: [*]syooi@mmu.edu.my

***Abstract*—Browser fingerprint is often linked to privacy as it is a method to gather data about the browser's configuration to identify the user. The browser's configurations, which are also known as attributes, are the keys to make the user to be identified. Web browsers explicitly disclose information about the host system to websites by making it available to them, such as attributes like the screen resolution, local time, or operating system (OS) version. Since each of the browsers has different attributes that make each unique, it is essential to understand the attributes well. This research paper emphasizes the method of collecting data for browser fingerprinting and ensuring the acquisition of fingerprint data without compromising personal information. One of the research motivations is to transform this data into an easily accessible raw dataset for the industry's utilization in future research projects. Additionally, the study explores the potential use of Shannon Entropy to unveil distinctive attributes in browser fingerprinting, revealing that higher entropy values correlate with more distinct and recognizable fingerprints. The other purpose is to discover which attribute produces the highest unique value using the clustering algorithm. Experiment results showed that if the attribute is unique, it will be hard to cluster into groups. This can be proved by using a clustering algorithm where the unique attributes will have a high value in the incorrectly clustered instances because it is harder to be clustered.**

***Keywords*— Browser fingerprints; attributes; data collection; JavaScript; Shannon entropy; clustering algorithm.**

Manuscript received 5 Dec. 2023; revised 20 Apr. 2024; accepted 25 Sep. 2024. Date of publication 31 Dec. 2024.

I. INTRODUCTION

Browser fingerprinting is the process by which a website on the World Wide Web extracts a user's browser fingerprint. Browser fingerprint functions like human fingerprints. Browser fingerprints are the hints a web user leaves behind that identify their presence on a website, much like human fingerprints are the hints that a person leaves behind identifying their presence at a location. Web browsers voluntarily divulge information about the host system to websites by making data like screen resolution, local time, or OS version available to them. The information collected from the user's browser is called the attributes. User agent, screen resolution, canvas fingerprint, date, time, fonts, and many other attributes are examples of the types of information present in each browser fingerprint. Even if some of the data acquired is not distinctive, the browser fingerprinting will only become unique when all the attributes have been compiled and combined. To do browser fingerprinting, the web server must instruct the web client to remove the browser fingerprint from its device and then send the fingerprint back to the web server via a network request [1]. The specifications of the browser the user uses to access the Internet, the system accessing the Internet, and many other characteristics of the machine/user tend to be the types of information typically collected.

Using browser fingerprints does not necessitate storing any data on the client side, and users cannot invalidate browser fingerprints by removing cookies or privacy options because browser fingerprint technology is stateless [2]. The information collected from a user's browsers is subject to distinguish between one user and another, which involves privacy implications. The information can track users, which has become a serious problem for the technology industry. It is no secret that fingerprints can identify certain users or devices, even fully or partially, when cookies are disabled. Even though many people erroneously believe that cookies are browser fingerprints, they are not. The distinction between the two is that while browser fingerprinting can track users throughout the internet, cookies can only be accessed by the website from which they were collected and cannot be transferred between websites. Also, cookies can be easily deleted by implementing privacy options that affect cookie

tracking. Therefore, cookies are not much of a threat compared to browser fingerprints.

However, restricting fingerprinting is significantly harder because it is impossible to identify when it is being used, and most systems do not yet have reliable, predictable means of preventing it. Large-scale research using browser fingerprint collecting has been tested in recent years. The Panopticlick [3], AmIUnique [4], and Hiding in the Crowd [5] are three well-known modern studies. The studies mentioned above had a significant impact and advanced the study of browser fingerprinting.

This paper discusses the collection of browser fingerprinting and clustering of browser fingerprint attributes while being validated via Shannon entropy. This study aims to provide a comprehensive, all-inclusive fingerprinting test suite segmented into the necessary portions and includes the pertinent data for the various groups. The data collection of browser fingerprinting is the first section to be discussed, followed by clustering the attributes of browser fingerprint instead of using Shannon entropy. In summary, this paper provides the following contributions:

- The flow of a built website to collect browser fingerprint data is discussed.
- Brief explanation of Shannon entropy used by other researchers to find the uniqueness of browser fingerprint.
- Clustering algorithm instead of Shannon entropy is used to find the most prominent attributes.

A. Browser Fingerprinting vs Cookies

Within web-based applications, cookies serve an essential function by maintaining the state of user interactions within the inherently stateless HTTP protocol, thereby facilitating a smoother and more personalized browsing experience. Servers can enrich their responses by embedding a "cookie" within the Set-Cookie header, transmitting data to the client's browser. This uninformed data might be anything the server requires, such as the user's identity, a database key, or other information [6], [7]. The request header and the new request header are two examples of cookie headers that clients return. In addition to its responses, the server can send a new cookie that overwrite the previous one. This establishes a reciprocal agreement between the server and the client. It forms a symbiotic relationship where the server entrusts the client with storing its state and expects them to return this information during ensuing visits. Through this mutual understanding, the server can maintain continuity in user interactions and deliver personalized experiences across multiple sessions[6], [7].

The browser selectively stores cookies from previously visited servers, often accessing servers without the user's awareness. Subsequently, the browser saves these cookies on the user's computer. As a result, the server and client exchange data are known as a "cookie." Additionally, Cookies may be employed to store "login" information, so there is no need to enter a name and password when visiting a website that offers customized access [8]. Cookies are used on a website to keep track of the pages that have been visited. Using cookies and text files to record personal data to a user and their website use has already been surpassed by browser fingerprinting. A key difference between cookies and browser fingerprinting is that cookies are exclusively available to the website that initially issued them and cannot be transferred between websites [9].

Conversely, browser fingerprinting can monitor users across the entire Internet. Utilizing privacy features that affect cookie tracking simplifies removing cookies from websites. In contrast, restricting fingerprinting presents a significantly greater challenge due to the difficulty in identifying its usage and the absence of clear, reliable methods in most systems to limit it effectively [9].

B. Motivation

Newly emerging fingerprinting technologies and heightened security measures are contending for attention within the constantly evolving realm of online privacy. It is essential to broaden the discussion surrounding online tracking to encompass browser fingerprinting. The issue has gained wide attention from the popular browser such as Mozilla, where it embedded the mechanism to prevent fingerprint collection as well as the frequent usage of ad blockers [10].

Undoubtedly, browser fingerprinting is an important field for research. At the same time, it serves as a foundation to raise awareness and educate users, developers, policymakers, and law enforcement about the visibility of browser fingerprints and their potential for profile linking. Furthermore, there is no foolproof method to detect or completely prevent it, significantly highlighting the research gaps that can be filled here.

Thus, this study postulated that understanding how browser fingerprinting works will significantly impact the user's awareness when using the Internet. It also provides insight into this technology, especially to those who are concerning their digital footprints on the Internet and privacy concerns. It also assists in determining the best way to reduce the privacy risk [11]. As technology progresses, comprehending and addressing the implications of browser fingerprinting will be crucial for shaping the future landscape of online privacy and security.

C. Background and Related Work

When a user accesses the specified website, a process known as "browser fingerprinting" occurs. This collects data about the user's system and browser configurations. Hardware, operating system, browser, and configuration data combine to form a browser fingerprint [5]. Over the past few years, there have been a few large-scale browser fingerprint data collection studies that have had a significant influence on the research of browser fingerprints. The most well-known research was Panopticlick [3], AmIUnique [4], and Hiding in the Crowd [5]. Although all 3 of the studies had a significant impact, there are still some limitations. Thus, the self-collection of the browser fingerprinting website needs to be built. The reasons are that most of their datasets involve violating users' privacy and that datasets are biased. For example, both the AmIUnique [4] and Panopticlick [3] datasets were considered biased because both websites were devoted to browser fingerprinting, and their visitors were interested in internet monitoring, significantly impacting the accuracy of results. Although the datasets in Hiding in the Crowd [5] are not biased, their datasets raised privacy issues.

This is because they acquired data collection through 15 French websites, a political website, and a weather forecast website, which were the identified websites [5]. Their results were noteworthy because 33.6% of the fingerprints in the collected data were distinct; nevertheless, they did not have permission to gather those fingerprints, which was regarded as a violation of the users' privacy. Other than the above reasons, another was that all three websites had fewer collected attributes, approximately 17; instead, we collected around 52. Table I below shows the summary of the benchmark studies.

TABLE I
SUMMARY OF THE BENCHMARK STUDIES

	Panoptilick [3]	**AmIUnique** [4]	**Hiding in the Crowd** [5]
Source	Free	Free	Free
Total Fingerprint collected	470,161	118,934	2,067,942
The way to collect data	Own website	Own website	15 French websites
Number of attributes collected	10	17	17

In past research, they determined the uniqueness of each attribute in the browser fingerprint by utilizing the conventional mathematical approach via Shannon entropy. This is because the amount of distinguishable information in a fingerprint can be determined using entropy. A fingerprint's entropy value increases with how distinct and recognizable it is [10]). In the past, Laperdix et al. [4] used Shannon entropy to identify the most distinct values for specific attributes. They found that deleting browser plugins and utilizing generic HTTP headers significantly reduced fingerprints' uniqueness on desktops by 36% [4]On the other hand, this paper discusses how to handle unlabeled data to find the most prevalent attributes with high identifiable value. This research employed the clustering approach to locate the identifiable attribute. A clustering method is utilized while being validated via Shannon entropy to determine how characteristics with a high degree of dissimilarity are clustered.

D. Proposed Website

This section discusses the formation of the browser fingerprint website, created in 2022 to collect fingerprints. The primary goal of this proposed website is to avoid collecting any privacy-related information, including IP addresses, to ensure that the research remains free from any behavioral biases. Thus, this approach minimizes the privacy concerns in the three benchmark studies. The website can be viewed at this link: https://fpting.com/. Fig. 1 below shows a snapshot of the website's front page.

The website's client-side software, primarily created in JavaScript, was inspired by the TorZillaPrint (Arkenfox) project. Once the user has visited the site by clicking the link, the client's web browser data is collected. However, user consent is crucial for ethical reasons; as a result, no personal data was gathered. When a user connects to the page containing the fingerprinting script, the server begins gathering HTTP headers [12].

Fig. 1 Snapshot of the first page of the designated website

Around 1500 fingerprints were collected in 2 years for this proposed website. The finalized data are shown in Table II below.

TABLE III
SUMMARY OF THE DATASET COLLECTED

Total Number of Datasets	**Number of Attributes Collected**	**Year of Proposed Website**	**The Way to Collect the Data**	**Datasets**
1500	52	2022	Own website	Public datasets

If the user has not deactivated JavaScript, the browser will run the script that collects most of the fingerprint information. Each fingerprint has various information depending on the browser being used, how it is configured, and the hardware and software environment it is running in. Every time a user accesses a website, their browser sends a GET request to the server to retrieve a page, and the server then responds with a response containing the content of the requested page. JavaScript files in the form of fingerprinting scripts are included in the delivered HTML [1]. These scripts may be used to track users between other websites by the site being visited as first-party scripts or through any third-party sites. After it has completed executing, the JavaScript script used to perform the fingerprinting must submit the data it has collected to a server. Some fingerprinting scripts transmit the complete attributes, while others merely compute and transmit a hash. Fig. 2 below delivers a summary of the browser fingerprinting procedure.

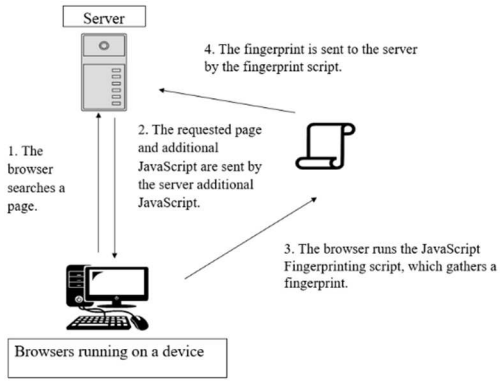

Fig. 2 Summary of the browser fingerprinting procedure

E. Data Collected from the Proposed Website

Data collection involves gathering attributes that make up the browser fingerprint. These attributes serve as the key elements distinguishing each browser fingerprint's uniqueness. Fig. 3 below provides a comprehensive view of the designed

webpage, visually representing the attributes and information encapsulated within the platform.

The extensive datasets meticulously collected as part of this study are positioned to be made openly accessible to the public, underscoring our commitment to transparency and furthering scholarly exploration in browser fingerprinting. Our foresight anticipates that this move will not only enhance the scientific discourse but will also serve as a catalyst for an increased focus on browser fingerprinting studies.

Fig. 3 Snapshot of the website built

A deliberate choice has been made to maintain the data in its raw, unaltered form, a strategic decision aimed at preserving the authenticity of every fingerprint. Inviting interested researchers and practitioners to request copies of the collected datasets is welcome. This process is streamlined by submitting a request via a user-friendly Google Form, thoughtfully embedded in our website. Fig. 4 below shows the screenshot of the Google form for anyone interested in getting the datasets.

Table III presents examples of browser fingerprints, including several key attributes along with their corresponding sources and sample values.

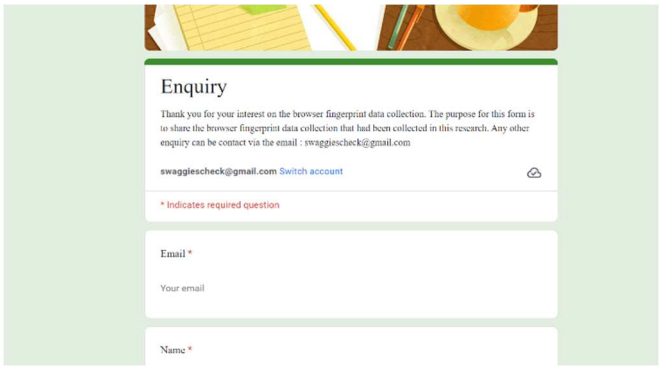

Fig. 4 Snapshot of Google form inquiry

Next, all the collected data was manually entered into an Excel file. This manual data entry phase is critical to pre-processing the data for further analysis through a machine learning algorithm. Since we are also sharing this dataset (upon request), the data is intentionally processed into a CSV (Comma-Separated Values) format, the most general and widely used format for almost all machine learning platforms, so that any researchers can directly use this dataset regardless of the machine learning platforms they use.

Fig. 5 below shows the screenshot of the datasets in CSV form. It is important to note that the dataset shared is in its raw form, encapsulated within text files. This deliberate choice underscores our dedication to maintaining the authenticity of the data, ensuring that researchers receive an unaltered version for their analyses.

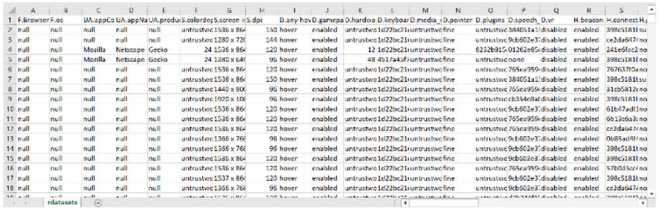

Fig. 5 Snapshots of datasets in CSV form

This raw format is the cornerstone for a wide range of research possibilities, offering versatility in application and interpretation. Other than that, by making these datasets freely available, we aim to contribute to the collective advancement of understanding browser fingerprints and promote a culture of open access in research.

F. Attributes Collected as Datasets

When compiling each browser's attributes, their uniqueness may be determined. Examples of attributes that were gathered for this study are shown in Table III below. Table IV lists the categories of commonly used attributes that need to be gathered, the source from which the data should be retrieved, and some examples of the attributes' values. The collection of attributes made up one whole browser fingerprint.

TABLE III
EXAMPLE OF BROWSER FINGERPRINTS

Attribute	Source	Value Examples
User Agent	HTTP header	Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.5005.63 Safari/537.36
App CodeName		
Screen	JavaScript	1280 x 670
window inner		
Canvas. Get Image Data	JavaScript	Cwm fjordbank glyphs vext quiz, 😃 Cwm fjordbank glyphs vext quiz, 😃
Header Connection	HTTP header	710341f5d2d27854f2c6b8322c88228d011fb06b
Timezone	JavaScript	-180
Lists of fonts	Flash or JavaScript	Abyssinica SIL, Aharoni CLM, AR PL UMing CN, AR PL UMing HK, AR PL UMing TW, etc.
Media session	JavaScript	yes

The impetus behind creating a self-collection dataset of browser fingerprints stems from a notable scarcity in the availability of such datasets. A significant portion of existing datasets in this domain often involve breaches of user privacy, raising ethical concerns. Moreover, the datasets commonly employed within the industry exhibit a scarcity of attributes. A poignant example lies in comparing widely recognized datasets like AmIUnique [4] and Hiding in the Crowd [5], comprising 17 attributes (refer to Table IV).

In contrast, the self-proposed website boasts a rich dataset encompassing 52 attributes (refer to Table V). Through the meticulous operation of the website, a total of 1500 fingerprints, each associated with approximately 52 attributes, were systematically gathered. The discrepancy in the number of collected fingerprints compared to the benchmark studies

stems from our discovery that the quantity of browser fingerprinting does not directly correlate with the outcomes of browser fingerprint analysis. Our findings revealed that the collected number of browser fingerprints does not function as samples but represents browser values. Therefore, the volume of browser fingerprint collection is inconsequential. This abundance enriches the dataset and amplifies the depth and diversity of information available for rigorous analysis and evaluation within browser fingerprinting.

Referring to the previous review done above, this research postulated that there is no need for the browser user's personal identification data, such as the IP address of the browser, to be collected for the attributes in this study. Given that the objective is to assess browser behaviors rather than intrude upon user privacy, utmost consideration is given to safeguarding the privacy of the user's browser. All the attributes collected for this study do not directly violate the user's privacy, making them all non-privacy invasion attributes. Although each gathered attribute may not individually exhibit uniqueness, the distinctiveness of browser fingerprinting is only achieved when all collected attributes are amalgamated into a singular entity.

TABLE IV
AMIUNIQUE AND HIDING IN THE CROWD'S 17 ATTRIBUTES

User-agent	Accept	Content-Encoding	Content Language
List of plugins	Cookies enabled	Use of local/session storage	Time zone
Screen resolution and color depth	List of fonts	List of HTTP headers	Platform
Do Not Track	Canvas	WebGL Vendor	WebGL Renderer
Use of an ad blocker			

TABLE V
PROPOSED WEBSITE 52 ATTRIBUTES

Feature Browser	Device. Gamepads	Header. Connection	Media. Can Play Audio
Feature OS	Device Hardware Concurrency	Header. global privacy Control	Media.can Play video
User Agent.app Code Name	Device. keyboard	Header. Online	Media.Type Supported audio
User Agent. appName	Device.media devices	Storage.app Cache	Media.isTypeS upported video
User Agent. product	Device. pointer	Storage Notification	Media. Capability
Screen. Color depth	Device. plugins	Storage. push	Media. Session
Screen. inner	Device. speech engines	Fonts. documents	CSS. colors css4
Screen.dpi	Device.vr	Fonts.Glypho ffset	Canvas. ToDataURL
Device. any-hover	Header. beacon	Fonts. proportional	CSS.colors system
MISC. Navigator keys	MISC.perf navigation	Canvas. toBlob	Canvas.get Image Data
.Language	Timezone	Geolocation	Audio
Datetime	Elements. height	Elements. keys	Canvas. PointInPath
Canvas.is PointInStroke	DomRect. ElementBoundi ng. ClientRect	DomRect. Element.getC lientRects	DomRect. getClient. Rects

II. MATERIALS AND METHOD

This section portrays the overall method of this research, including how the clustering algorithm find the uniqueness of browser fingerprints' attribute values while validating with Shannon Entropy. Fig.6 below shows the overall method.

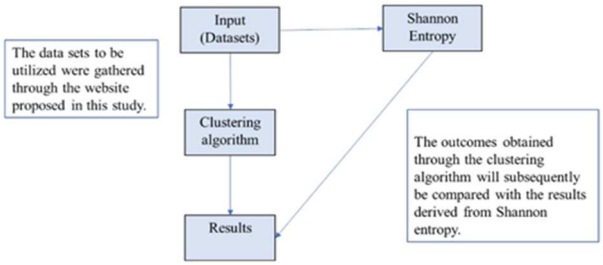

Fig. 6 Shannon Entropy's Compatibility with Proposed Clustering methods for Identifying Unique Attributes in Browser Fingerprinting

A. Theoretical Foundation of Shannon Entropy

This section portrays how Shannon Entropy is used to identify the unique attributes in browser fingerprinting. Most previous studies have used entropy to determine the distinct values for specific attributes. This is evident, for instance, in investigations by Laperdix et al. [10]. For example, using generic HTTP headers and removing browser plugins reduced the uniqueness of fingerprints on desktops by a significant 36%. The Shannon Entropy was implemented via Python because it is a well-liked programming language for scientific computing and data analysis, and it has many libraries and packages created. Python, in particular, includes several packages for dealing with CSV files, such as pandas, which offers a robust DataFrame data structure for working with and analyzing tabular data. Algorithm 1 showing the calculation of Shannon Entropy using Python.

Algorithm 1. Shannon Entropy

Input: Browser fingerprint's attribute values
Output: Shannon Entropy of the attribute values
Procedure:
1. Use the self-collected datasets for the input data.
2. Set total_count to the sum of all values in the data.
3. Initialize entropy to 0.0.
4. For each value count in data:
5. If the count is greater than 0:
6. Calculate the probability of the event: probability = count / total_count.
7. entropy = probability * log2(probability).
8. Return the result

Shannon Entropy has been widely used over the decades to find the uniqueness of each attribute. This is because entropy has been proven reliable in representing the identifiable browser fingerprint. A higher entropy value denotes a highly distinguishable browser fingerprint [13]. The concept of entropy has been used in various scientific fields and applications [14]. Entropy is crucial to many areas of science and engineering, such as thermodynamics, information theory, and statistical mechanics [15]. It is a way to gauge how chaotic or unpredictable a system is, and it has significant consequences for how we comprehend natural and biological systems. Shannon [15] [16] first introduced the idea of entropy as part of the primary communication theory,

defining that a source of data, a communication channel, and a receiver mainly form a data communication system.

Throughout the year, many variations of entropies have been customized and generated to fit into different physical and mathematical systems [16]. Each type of entropy has a distinct definition, as well as different applications and implications. Among them. thermodynamic entropy is one of the notable ones used to measure the disorder or randomness of a system at the macroscopic level [15].

Since then, Shannon entropy has also been used to calculate the uniqueness of browser fingerprint attributes [4]. By applying Shannon entropy to the values of each attribute, it is possible to estimate the uniqueness level of a particular browser configuration.

The formula of Shannon entropy can be described as below:

$$H(X) = -\sum_i p\ (x_i) \log_b p\,(x_i) \qquad (1)$$

The formula for Shannon entropy [14], denoted by $H(X)$, calculates the information content or uncertainty of a set of data, where x is a random variable, and $p(x)$ is the probability distribution of x. The formula captures the degree of uncertainty or randomness in the message by computing the average amount of information in bits required to transmit a message from a source with a probability distribution $p(x)$. A higher entropy value indicates greater uncertainty, meaning more bits are necessary for accurate transmission.

Researchers have frequently utilized Shannon Entropy to gauge the uniqueness of browser fingerprinting in their experiments. Their studies demonstrate that Shannon Entropy effectively discerns the values of attributes collected through browser fingerprinting. By applying Shannon entropy in browser fingerprinting, researchers can evaluate the diversity and distinctiveness of collected attributes among users. Higher entropy values signify greater diversity and a heightened potential for unique identification. This assertion finds support from prominent researchers in the field, such as AmIUnique [4], Hiding in the Crowd [5], and Panoptilick [3]. These studies have consistently yielded significant findings in identifying the most distinctive attributes.

Table VI below shows lists of attributes with their Shannon's entropy value for the AmIUnique [4], Panoptilick [3], and Hiding in the Crowd [5]. The Table shows the highest entropy of four attributes derived from past research.

TABLE VI
LIST OF ATTRIBUTES AND VALUES OF SHANNON ENTROPY AMONG PAST RESEARCH

Attribute	AmIUnique	Panoptilick	Hiding in the Crowd
List of fonts	8.379	13.900	6.904
Screen resolution	4.889	4.830	4.847
List of HTTP headers	4.198	-	1.783
Canvas	8.278	-	8.546

B. Results of Shannon Entropy

Python was used to construct the Shannon entropy since it has many libraries and packages and is a popular programming language for scientific computing and data analysis. An experiment was carried out using datasets collected from https://fpting.com/. Table VII below shows the top 5 highest entropy values: screen window inner, HTTP Connection, and others. According to previous studies, fingerprints become increasingly distinctive and traceable as entropy increases [5], [17].

TABLE VII
TOP 5 MOST UNIQUE ATTRIBUTES WITH THEIR ENTROPY VALUE

Attribute	Entropy
Screen Window Inner	6.4125
Device Speech Engines	5.3226
Fonts	4.9793
HTTP Connection	4.9280
Canvas	4.7300

The result showed us that screen resolution with 6.4125 entropy as a dominant attribute lies in the unique display characteristics of each device. The total number of pixels forms the screen resolution on a user's screen. Usually, it varies according to the device size and resolution. For instance, a 13-inch laptop might have a resolution of 1440x900, while a 27-inch desktop monitor may have a resolution of 2560x1440. These distinct resolutions can be utilized to track and identify users [18].

On the other hand, an HTTP Connection with 4.9280 entropy is another essential attribute. The Connection attribute also represents a header that allows senders to define connection preferences and enables multiple HTTP requests and responses over a single TCP connection. Most of the time, browsers will have unique values for this header. Thus, making it a unique feature to form the browser fingerprint. Altering the "Connection" header, as demonstrated in AmIUnique's study by [6] and [10], can significantly reduce the uniqueness of browser fingerprinting by 36%.

Next, fonts with an entropy of 4.9793 also represent a distinguishable attribute in this study. This is because different operating systems and browsers use different default fonts. Examining the default font from a user's device can reveal information about the browser and operating system [19], [20], [21]. The ranking of font lists can also be examined to analyze and interpret the users' browsers and operating systems further. When combining this analysis with canvas fingerprinting, the accuracy of browser fingerprinting can be even improved.

On top of that, Canvas fingerprinting with an entropy value of 4.7300 also represents the uniqueness of an image usually generated by the HTML5 Canvas element. This method further intricacies that browser fingerprinting can showcase the multifaceted nature of attributes that can be exploited for user profiling. To execute canvas fingerprinting, JavaScript is employed to craft a concealed image on the canvas element, comprising various elements such as text, shapes, and colors. A unique identity is assigned to the user's browser by creating a hash value from the image using a hashing algorithm. This resilience underscores the reliability and persistence of this method in uniquely identifying browsers. Next, we delve into the attributes that lack uniqueness, where most users or devices share identical values. Table VII highlight some of the least unique attributes.

TABLE VIII
LEAST UNIQUE ATTRIBUTES

Attribute	Entropy
Devices Virtual reality	0.0000
Gamepads	0.1593
Beacon Header	0.2569

In browser fingerprinting, higher entropy corresponds to greater uniqueness in attribute values, while lower entropy signifies less distinctiveness. Table VII above reveals that the attribute devices' virtual reality exhibits zero entropy values. This outcome stems from the absence of diversity in the data, leading to an entropy value of 0 for a specific column if it contains only one unique value across all rows. This observation is consistent with our research's collected browser fingerprint data, indicating that virtual reality is unsupported on the devices surveyed.

Additionally, gamepads with 0.1593 entropy value emerge as another category with relatively low uniqueness. The properties associated with gamepads possess only binary values of enabled or disabled, lacking the granularity required for robust user identification. Consequently, gamepad properties do not provide sufficiently distinctive information for effective browser fingerprinting. Similarly, the beacon header is among the least unique attributes due to its potential absence in certain scenarios. Specifically, the Beacon header with a 0.2569 value depends on the user's browser enabling the Beacon API and the website actively utilizing this API for data transmission. When the Beacon API is not enabled or the website is not utilized, the Beacon header will be absent from HTTP requests [22], [23].

Next, Table VIII below shows lists of attributes with their Shannon's entropy value for the AmIUnique [4], Panoptilick [3], Hiding in the crowd [5], and the work in this paper. Table IX below shows the highest entropy of attributes derived from past research and ours.

TABLE IX
SHANNON'S ENTROPY FOR ALL ATTRIBUTES FROM PAST RESEARCH AND OUR DATA

Attribute	AmI-Unique	Panoptilick	Hiding In the Crowd	Fpting (our solution)
List of fonts	8.379	13.900	6.904	4.9793
Screen resolution	4.889	4.830	4.847	6.4125
List of HTTP headers	4.198	-	1.783	4.9280
Canvas	8.278	-	8.546	4.7300

In conclusion, the findings of this research underscore the correlation between attributes with high unique values and elevated entropy and, conversely, attributes with lower unique values and diminished entropy. By applying Shannon entropy theory, this study successfully identifies the most unique attributes within browser fingerprinting. The implications of this investigation are significant, shedding light on the critical role certain attributes play in user identification, thereby accentuating the heightened privacy concerns associated with browser fingerprinting. Recognizing these privacy implications emphasizes the need for continued scrutiny and ethical considerations in the evolving landscape of online users.

III. RESULT AND DISCUSSION

In general, clustering algorithms take an unsupervised method in which the input is unlabelled, and the algorithm learns from a set of practice issues to discover the solution to a problem. Clustering seeks to split a finite unlabelled data collection into a definite and separate set of "natural," hidden data structures rather than precisely describing unseen samples made from the same probability distribution [24], [25] This study employed the clustering approach to pinpoint the attributes with the most obvious unique values. The clustering algorithm was implemented via WEKA [26]. WEKA supports many clustering methods, such as EM, FilteredClusterer, HierarchicalClusterer, SimpleKMeans, etc. This study uses five clustering techniques: FarthestFirst, FilteredClusterer, SimpleK-Means, HierarchicalClusterer, and MakeDensityBasedClusterer.

In this proposed approach, the clustering algorithm was determining the most prominent attributes of the collected browser fingerprinting. We postulated that if the attribute is unique, it was hard to cluster into groups. This can be demonstrated using a clustering algorithm where the unique attributes have a high value in the incorrectly clustered instances because it is harder to cluster them into groups. If the attributes are exceptional, they cannot be adequately clustered into groups, creating a high value in incorrectly clustered instances.

Fig. 7 below represents our proposed methodology on how the clustering algorithm can find the most prominent attributes. All attributes from the collected browser fingerprinting dataset are unlabeled in nature. All of them will be fed into a clustering model and let the algorithm cluster them based on similarity. For the outliers or those attributes that cannot be clustered, we deemed these to be the unique attributes that indicate the possibility of using them to profile a user.

In other words, attributes that can be clustered indicate that every browser might use the same attributes, thus unable to uniquely represent a specific user, whereas attributes that cannot be clustered indicate that they are "unique" and, thus, can be further exploited and linked to a specific user.

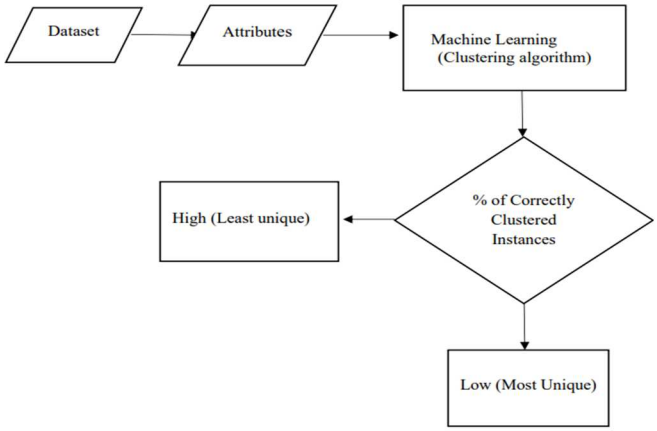

Fig. 7 Flowchart of finding the most prominent attributes via clustering algorithm

A. Related Work

Using clustering algorithms in browser fingerprinting is not a novel concept, given that the data involved in browser fingerprinting is predominantly unlabelled. Several studies within the industry have made notable contributions regarding the utilization of clustering algorithms in managing browser fingerprinting data. Many of these studies have employed clustering algorithms in diverse ways to improve browser fingerprinting configurations. For instance, in 2019, Gomez

et al. [27] observed that most existing countermeasures aimed at mitigating browser fingerprinting fail to provide comprehensive privacy protection, as some may introduce discrepancies or peculiar characteristics that distinguish these users from others. Consequently, they devised an innovative strategy to address these issues, reducing user identification and configuration adjustment requirements. They also employed clustering techniques to identify devices likely to share similar or identical fingerprints, assigning them new non-unique fingerprints. Consequently, web browsers used by devices within the same cluster exhibit uniformity, making differentiating challenging.

Similarly, Zou and Zhai [28] experimented with developing a browser fingerprinting identification method based on numerous readily available implicit identifiers. They implemented an incremental clustering algorithm based on autoencoders to address the challenge of traditional fingerprinting technology's inability to determine whether a user accessing a website twice with different attribute values is using the same browser on each occasion, resulting in poor identification results.

Another study by Ding et al. [29] focused on browser fingerprinting on mobile devices, marking one of the earliest attempts to handle unlabeled mobile device data. Their emphasis was on creating a device fingerprint using configuration-related traits of mobile devices, enabling distinct and reliable device characterization. They proposed an incremental clustering method to group unlabeled device data into clusters based on similarity. The study also proposed a precise authentication system between users and devices by adjusting each user's distance threshold based on how frequently their device settings change. While these studies represent significant strides in browser fingerprinting, it is essential to note that the works described above are limited in optimizing identifying attributes highly identifiable with the value via clustering algorithms.

B. Clustering Algorithm's Results

This section describes how the clustering technique was used to identify the most recognizable attributes in the research. Most previous studies have used entropy to determine the values of various attributes, such as research by Laperdix et al. [10] demonstrates this. They found that utilizing generic HTTP headers and removing browser plugins might significantly reduce the uniqueness of fingerprints on desktops by 36%. However, a novel method for finding the most easily identifiable attributes was employed in this study, incorporating a clustering algorithm. The experiment demonstrated that an attribute cannot be adequately clustered if it appears higher in an incorrectly clustered instance. To put it another way, this paper demonstrates

that the attribute has a more excellent unique value and may effectively identify users. The figures below display the results of a few attributes sorted in the least correctly clustered instances. The attributes shown in the figures are the top 5 most identifiable and unique.

Table X illustrates the attributes of FarthestFirst, FilteredClusterer, HierarchicalClusterer, MakeDensity-BasedClusterer, and SimpleK-Means, their percentages of correctly clustered instances, and their percentages of incorrectly clustered cases. It displays all five clustering algorithms along with their outcomes.

TABLE X
FARTHESTFIRST, FILTEREDCLUSTERER, HIERARCHICALCLUSTERER, MAKEDENSITYBASEDCLUSTERER AND SIMPLEK-MEANS'S ATTRIBUTES, PERCENTAGE OF CORRECTLY CLUSTERED INSTANCES, AND THE PERCENTAGE OF INCORRECTLY CLUSTERED INSTANCES

	FarthestFirst		**FilteredClusterer**		**HierarchicalClusterer**		**MakeDensityClusterer**		**SimpleKMeans**	
Attribute	% of Correctly Clustered Instances	% of Incorrectly Clustered Instances	% of Correctly Clustered Instances	% of Incorrectly Clustered Instances	% of Correctly Clustered Instances	% of Incorrectly Clustered Instances	% of Correctly Clustered Instances	% of Incorrectly Clustered Instances	% of Correctly Clustered Instances	% of Incorrectly Clustered Instances
Screen window inner	19.9702	80.0298	19.8212	80.1788	20.044	79.9553	19.8212	80.1788	19.8212	80.1788
Canvas	17.8092	82.1908	21.9821	78.0179	17.3621	82.6379	21.9821	78.0179	21.9821	78.0179
Device speech engine	20.5663	79.4337	23.5469	76.4531	19.8212	80.1788	23.5469	76.4531	23.5469	76.4531
Fonts	27.1982	72.8018	30.8495	69.1505	26.9001	73.0999	30.8495	69.1505	30.8495	69.1505
Header connection	27.7943	72.2057	35.9165	64.0835	25.4844	74.5156	35.9911	64.0889	35.9165	64.0635

In most clustering algorithms, screen window inner characteristics have the highest percentage of incorrectly grouped instances. This can also be seen in the earlier experiment on Shannon entropy, whereby the screen window inner has the highest entropy values. Although none of the five clustering approaches have the same value, they all fall within the same range. The top attribute that is the most difficult to cluster effectively is the screen window inner because different browsers have different preferences for their screen window widths. Users can alter the settings to suit their preferences. Therefore, this can be one of the defining characteristics used to identify them.

C. Comparison of the Attribute's Values

This part discusses how both attributes differ and how unique they are. For example, Fig. 8 and Fig. 9 above show the graph of the header connection and gamepad attributes in the SimpleK-Means clustering algorithm. The graph of the clustering of header connection attribute values is shown in Fig. 8. The graphic shows that certain instances are not adequately clustered. The fact that they cannot be fully clustered indicates that this attribute has high distinct values. The connection attribute of a header is a general type of header that allows the sender or client to provide preferences for that connection. Instead of opening a new TCP connection

for each request or response, the connection allows sending or receiving many HTTP requests and responses using a single TCP connection [30]. It was also proven that HTTP header connection is a unique attribute, according to the study of AmIUnique, where they mentioned that meddling with the HTTP header can reduce the uniqueness of browser fingerprinting by 36% [11]. This is because each browser has a different header connection setting, making it one of the most distinctive attribute values.

The other, Fig. 9, shows the graph of how the gamepad attribute is clustered into groups. These instances are appropriately clustered in the illustration. Gamepad properties can only have an enabled or disabled value [31], which is not distinctive enough for user identification. As a result, attributes with low unique values can be clustered easily because each user is similar, whereas attributes with high unique values are more challenging to classify appropriately. This demonstrates that some attributes used in browser fingerprinting are essential for user identification and present additional privacy concerns.

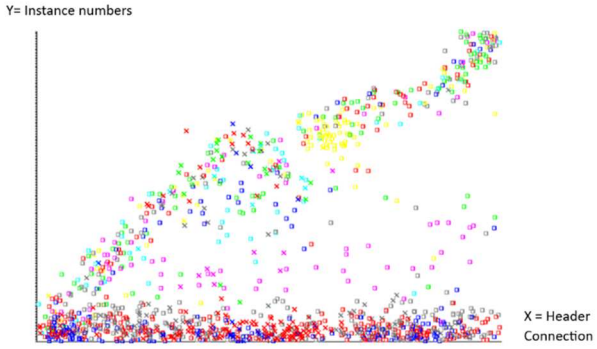

Fig. 8 Graph of header connection attribute

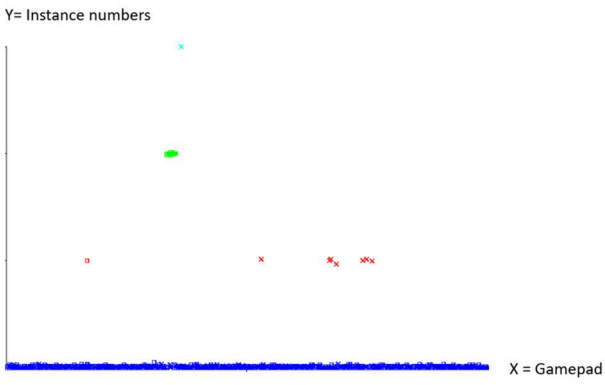

Fig. 9 Graph of gamepad attribute

IV. Conclusion

In conclusion, a browser fingerprinting website has been developed and hosted at https://fpting.com/, with a collection of 1,500 fingerprints with 52 attributes. The research aims to offer a technique for collecting browser fingerprinting data without using any personal data. Next, collecting browser fingerprint databases and making them accessible to the public would be the other goal. Browser fingerprint raises privacy issues in the modern digital era; thus, research should be highly encouraged. It is possible to identify a user by learning information about their browser, such as the user agent, screen resolution, local time, or OS version, and the process is called "browser fingerprinting." Since each browser is unique and has its personality, it is essential to understand the attributes that make up a browser's setup for privacy protection.

In past research, they found that using Shannon entropy is one technique for identifying distinguishing features in browser fingerprinting. This study's findings are relevant because they may improve the accuracy and dependability of browser fingerprinting techniques, which is important in online monitoring and privacy [32]. Thus, this research paper proposes a new method to find the most prominent attributes via a clustering algorithm. The experiment has shown that unique attributes are more difficult to cluster and have higher inaccurate clustering scores. To determine whether the theory is feasible, experiments were fairly conducted. As a result, handling browser fingerprint identification in the future will be simpler once the uniqueness of attributes has been determined.

References

[1] M. A. I. Mohd Aminuddin, Z. F. Zaaba, A. Samsudin, F. Zaki, and N. B. Anuar, "The rise of website fingerprinting on Tor: Analysis on techniques and assumptions," *Journal of Network and Computer Applications*, vol. 212, p. 103582, Mar. 2023, doi:10.1016/j.jnca.2023.103582.

[2] D. Zhang, J. Zhang, Y. Bu, B. Chen, C. Sun, and T. Wang, "A Survey of Browser Fingerprint Research and Application," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–14, Nov. 2022, doi: 10.1155/2022/3363335.

[3] P. Eckersley, "How Unique Is Your Web Browser?," *Privacy Enhancing Technologies*, pp. 1–18, 2010, doi: 10.1007/978-3-642-14527-8_1.

[4] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine, "Browser Fingerprinting," *ACM Transactions on the Web*, vol. 14, no. 2, pp. 1–33, Apr. 2020, doi: 10.1145/3386040.

[5] A. Gómez-Boix, P. Laperdrix, and B. Baudry, "Hiding in the Crowd," *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, pp. 309–318, 2018, doi: 10.1145/3178876.3186097.

[6] B. M. Berens, M. Bohlender, H. Dietmann, C. Krisam, O. Kulyk, and M. Volkamer, "Cookie disclaimers: Dark patterns and lack of transparency," *Computers & Security*, vol. 136, p. 103507, Jan. 2024, doi: 10.1016/j.cose.2023.103507.

[7] R. Pan and A. Ruiz-Martínez, "Evolution of web tracking protection in Chrome," *Journal of Information Security and Applications*, vol. 79, p. 103643, Dec. 2023, doi: 10.1016/j.jisa.2023.103643.

[8] U. Iqbal, S. Englehardt, and Z. Shafiq, "Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors," *2021 IEEE Symposium on Security and Privacy (SP)*, May 2021, doi: 10.1109/sp40001.2021.00017.

[9] I. Fouad, C. Santos, A. Legout and N. Bielova, "Did I delete my cookies? Cookies respawning with browser fingerprinting", 2021.

[10] P. Laperdrix, W. Rudametkin, and B. Baudry, "Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints," *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 878–894, May 2016, doi: 10.1109/sp.2016.57.

[11] A. Vastel, P. Laperdrix, W. Rudametkin, and R. Rouvoy, "FP-STALKER: Tracking Browser Fingerprint Evolutions," *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 728–741, May 2018, doi: 10.1109/sp.2018.00008.

[12] K. N. Pau, V. W. Q. Lee, S. Y. Ooi, and Y. H. Pang, "The Development of a Data Collection and Browser Fingerprinting System," *Sensors*, vol. 23, no. 6, p. 3087, Mar. 2023, doi: 10.3390/s23063087.

[13] L. Polčák, M. Saloň, G. Maone, R. Hranický, and M. McMahon, "JShelter: Give Me My Browser Back," *Proceedings of the 20th International Conference on Security and Cryptography*, pp. 287–294, 2023, doi: 10.5220/0011965600003555.

[14] A. Hoayek and D. Rullière, "Assessing clustering methods using Shannon's entropy," *Information Sciences*, vol. 689, p. 121510, Jan. 2025, doi: 10.1016/j.ins.2024.121510.

[15] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948, doi: 10.1002/j.1538-7305.1948.tb01338.x.

[16] M. L. Morrison and N. A. Rosenberg, “Mathematical bounds on Shannon entropy given the abundance of the ith most abundant taxon,” *Journal of Mathematical Biology*, vol. 87, no. 5, Oct. 2023, doi: 10.1007/s00285-023-01997-3.

[17] W. A. Kreiner, “First Digits’ Shannon Entropy,” *Entropy*, vol. 24, no. 10, p. 1413, Oct. 2022, doi: 10.3390/e24101413.

[18] C. Thota, C. Mavromoustakis, and G. Mastorakis, “CAP2M: Contingent Anonymity Preserving Privacy Method for the Internet of Things Services,” *Computers and Electrical Engineering*, vol. 107, p. 108640, Apr. 2023, doi: 10.1016/j.compeleceng.2023.108640.

[19] V. W. Q. Lee, S. Y. Ooi, and Y. H. Pang, “Assessing the Importance of Browser Fingerprint Attributes towards User Profiling through Clustering Algorithms,” *2023 IEEE 13th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 326–331, May 2023, doi: 10.1109/iscaie57739.2023.10165492.

[20] R. Zhao, “Toward the flow-centric detection of browser fingerprinting,” *Computers & Security*, vol. 137, p. 103642, Feb. 2024, doi: 10.1016/j.cose.2023.103642.

[21] A. A. Salomatin, A. Yu. Iskhakov, and R. V. Meshcheryakov, “Comparison of the Effectiveness of Countermeasures Against Tracking User Browser Fingerprints,” *IFAC-PapersOnLine*, vol. 55, no. 9, pp. 244–249, 2022, doi: 10.1016/j.ifacol.2022.07.043.

[22] J. Kumuthini et al., “Genomics data sharing,” *Genomic Data Sharing*, pp. 111–135, 2023, doi: 10.1016/b978-0-12-819803-2.00003-1.

[23] S. Jayanthy, A. Arunkumar, J. J. A. Kovilpillai, M. Bhuvardhena, and K. D. Pandian, “Secured Health Data Sharing System using IPFS and Blockchain with Beacon Proxy,” *Procedia Computer Science*, vol. 230, pp. 788–797, 2023, doi: 10.1016/j.procs.2023.12.054.

[24] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, “K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data,” *Information Sciences*, vol. 622, pp. 178–210, Apr. 2023, doi: 10.1016/j.ins.2022.11.139.

[25] J. Redha and J. Redha Mutar, “A Review of Clustering Algorithms,” *International Journal of Computer Science and Mobile Applications*, vol. 10, pp. 44–50, 2022.

[26] I. H. Witten, E. Frank, and M. A. Hall, “Introduction to Weka,” *Data Mining: Practical Machine Learning Tools and Techniques*, pp. 403–406, 2011, doi: 10.1016/b978-0-12-374856-0.00010-9.

[27] A. Gómez-Boix, D. Frey, Y.-D. Bromberg, and B. Baudry, “A Collaborative Strategy for Mitigating Tracking through Browser Fingerprinting,” *Proceedings of the 6th ACM Workshop on Moving Target Defense*, pp. 67–78, Nov. 2019, doi: 10.1145/3338468.3356828.

[28] F. Zou and H. Zhai, “Browser Fingerprinting Identification Using Incremental Clustering Algorithm Based on Autoencoder*,” 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, Dec. 2021, doi: 10.1109/hpcc-dss-smartcity-dependsys53884.2021.00093.

[29] Z. Ding, W. Zhou, and Z. Zhou, “Configuration-Based Fingerprinting of Mobile Device Using Incremental Clustering,” *IEEE Access*, vol. 6, pp. 72402–72414, 2018, doi: 10.1109/access.2018.2880451.

[30] E. Conrad, S. Misenar, and J. Feldman, “Domain 4: Communication and Network Security,” *CISSP® Study Guide*, pp. 225–293, 2023, doi: 10.1016/b978-0-443-18734-6.00003-9.

[31] C. Y. Seek, S. Y. Ooi, Y. H. Pang, S. L. Lew, and X. Y. Heng, “Elderly and Smartphone Apps: Case Study with Lightweight MySejahtera,” *Journal of Informatics and Web Engineering*, vol. 2, no. 1, pp. 13–24, Mar. 2023, doi: 10.33093/jiwe.2023.2.1.2.

[32] Y. H. Tay, S. Y. Ooi, Y. H. Pang, Y. H. Gan, and S. L. Lew, “Ensuring Privacy and Security on Banking Websites in Malaysia: A Cookies Scanner Solution,” *Journal of Informatics and Web Engineering*, vol. 2, no. 2, pp. 153–167, Sep. 2023, doi: 10.33093/jiwe.2023.2.2.12.