Chapter 3: Preliminary Analysis and Visualization of Datasets

Primarily two datasets have been used in the thesis for web security related problems, viz., webpages dataset and hybrid apps dataset. These two datasets are analyzed and visualized in sections 3.1 and 3.2, respectively. The experimental setup and code for these two datasets are given in **Appendices A and B**, respectively.

3.1 Webpages Dataset

Web Security is a challenging task amidst ever-rising threats on the Internet. With billions of websites active on the Internet, and hackers evolving newer techniques to trap web users, machine learning offers promising techniques to detect malicious websites. The dataset described in this section is meant for such ML based analysis of malicious and benign webpages. The data has been collected from the Internet using a bespoke, focused web crawler named MalCrawler [67], which was built specifically for this purpose. The dataset comprises various extracted attributes and also raw webpage content, including JavaScript code. It supports both supervised and unsupervised learning. For supervised learning, class labels for malicious and benign webpages have been added to the dataset using the Google Safe Browsing API [61]. Safe Browsing is a Google service for checking whether a webpage is malicious or not. In this dataset, it has been used for assigning Class Labels: 'good'-benign/ 'bad'-malicious. URL of the webpage is submitted using this API to Google Safe Browsing Service, which after that cross-checks with its blacklist, and replies whether it is malicious or not. The most relevant attributes within the scope have already been extracted and included in this dataset. However, the raw web content, including JavaScript code contained in this dataset, supports further attribute extraction, if so desired. Also, this raw content and code can be used as unstructured data input for text-based analytics. This dataset consists of approximately 1.5 million webpages, which makes it suitable for deep learning algorithms.

3.1.1 Webpages Dataset: Specifications

The specification details of this dataset are given in **Table 3.1**.

Table 3.1: Specifications of Webpages Dataset

Type of data	Dataset Tables Figures Graphs Python Code
How data was acquired	The data was collected from the Internet by scraping webpages using MalCrawler [67]. The raw data collected was processed using customized Python code to extract relevant features.
Data format	Raw (Unstructured web content and JavaScript) Analyzed Filtered
Parameters for data collection	Web content was pruned down to reduce the size by removing less relevant content, viz. metadata, stop words, style data, HTML tags, etc. Obfuscated JavaScript code was de-obfuscated using a browser emulator.
Description of data collection	The raw data comprises of webpages Scraped data were further processed using customized Python code to extract attributes. Class labels for malicious and benign webpages were added using the Google Safe Browsing API [61].
Data source location	Data was gathered from the Web between November 2019 and March 2020, with random web crawls to ensure adequate global coverage.
Data accessibility	Data hosted in a public repository. Repository name: Mendeley Data and Kaggle. Data identification number: 10.17632/gdx3pkwp47.1 Direct URL to data: http://dx.doi.org/10.17632/gdx3pkwp47.1 Data Visualization Code for dataset hosted online on Mendeley [68] and Kaggle [69].

3.1.2 Webpages Dataset: Importance and Relevance

Webpages dataset is useful for building ML models for carrying out varied analyses on webpages. Both supervised and unsupervised learning models can be developed. However, it is important to note that presently no such comprehensive dataset exists in the public domain to facilitate research in this field. It will benefit all researchers who are pursuing research in the field of web security. Further, this data can be used by Cyber Security firms or Anti-

Virus companies to model their security products. It contains sufficient attributes for further insight. This data also includes processed raw web content, including JavaScript code, which can be used to extract new attributes, if so required, to aid future research. It has value not only to the Internet Security research community or Cyber Security firms but also for policy development by Cyber Law Enforcement agencies.

3.1.3 Webpages Dataset: Data Description

The dataset was designed and prepared keeping in mind the downstream task of classification of webpages as malicious or benign. However, this dataset contains sufficient information that can be used for any ML task related to webpage analysis. The attributes of this dataset are listed in **Table 3.2**.

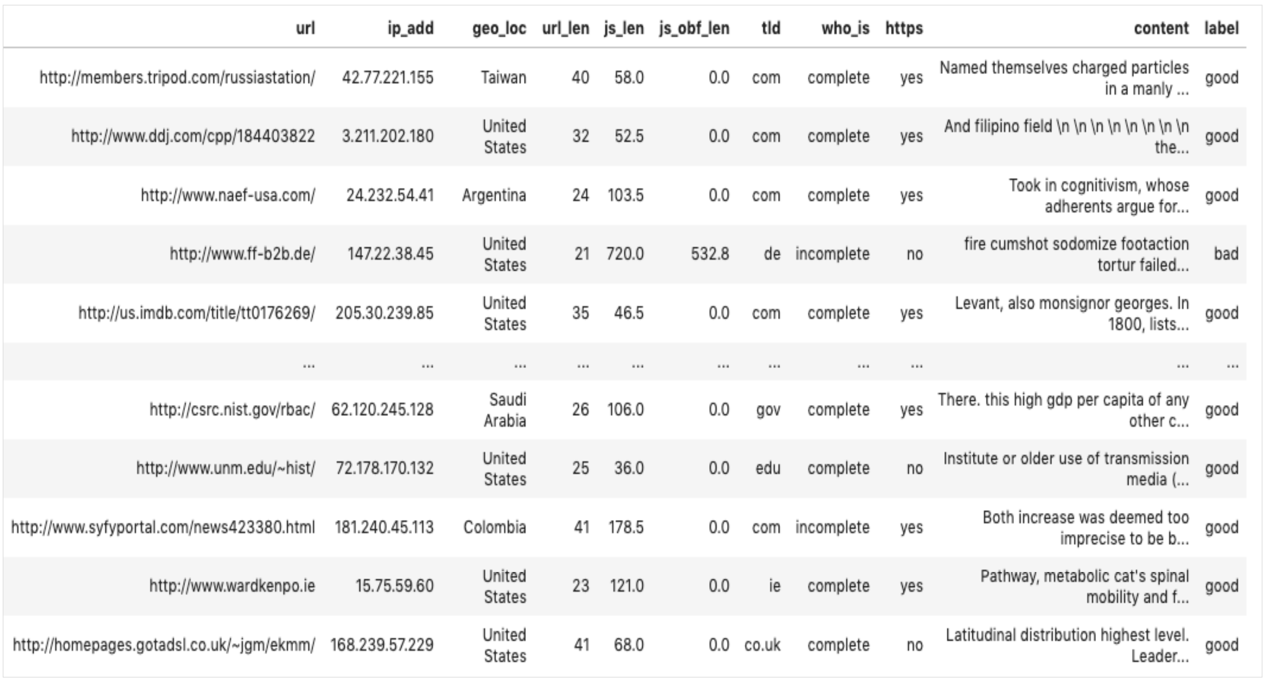
The choice of features in **Table 3.2** was based on their relevance in malicious webpage classification as will be shown in Chapter 4. Out of the 25 attributes analyzed in Chapter 4 in **Table 4.1**, few good predictors, as listed in **Table 3.2**, have been taken for further work. Few otherwise good predictors like 'Cloaking (A7)', 'Presence of iFrame' (A8), 'Redirection' (A6), and 'Popups' (A21), etc., have been left out due to the ambiguities and complexities associated with them during the extraction and pre-processing stage that make them unsuitable for commercial deployment (refer to **Figure 4.4** showing the utilization of computational resources). Few of the attributes listed in **Table 4.1** have been processed further as per the requirements of ML models in the thesis. The attribute numbers from **Table 4.1** are written against each attribute for the ease of correlation with the work done in Chapter 4.

Table 3.2: Attributes of Webpages Dataset

#	Attribute Reference to Table 4.1	Attribute Name	Attribute Description
F1	A2	url/ url_vect (url_vect is further processed from url)	URL of the webpage. {Datatype- string} Also, for vectorized inputs, like the input used for DNN in Chapter 5, URL is vectorized further. Keywords extracted from URL are vectorized using the Profanity Score (Profanity Score is a real value given to a group of words based on their

			goodness/badness. A pretrained SVM model was used for computing this score [138].). {Datatype- Numerical, float64: Normalised between 0-1}
F2	A1 (F2 is taken from A1)	ip_add	IP Address of the webpage (Note: IP Address (F3) is used for determining the Geographic Location (F4). However, the attribute (F3) itself may not used for training as it lacks requisite information value and relevance. Thus, it should be dropped before the input layer.). {Datatype- string}
F3	A1 (F3 is also taken from A1)	geo_loc	Name of the country based on IP Address location. {Datatype- Categorical, string}
F4	A2 (F4 is also further processed from A2)	url_len	Length of URL- count of characters in a URL. {Datatype- Numerical, int16}
F5	A24	js_len	Length of JavaScript code (in KB) in the webpage. {Datatype- Numerical, float64}
F6	A25	js_obf_len	Length of Obfuscated JavaScript (in KB) in the webpage. {Datatype- Numerical, float64}
F7	A4 (F7 is also processed from A4)	tld	Top Level Domain of the webpage. {Datatype- Categorical, string}
F8	A5	who_is	It gives out whether the WHOIS information of the registered domain is complete or incomplete. {Datatype- Categorical, string, value- incomplete/complete}
F9	A3	https	Gives out whether the website uses HTTPS or HTTP protocol. {Datatype- Categorical, string, value- yes/no}
F10	-	content	Raw web content of the webpage. Includes filtered and processed text and JavaScript code
F11	Class Label	label	Classification label categorizing the webpage as malicious or benign. {Datatype- Categorical, value- good/bad}

The dataset comprises 1.564 million webpages having 11 attributes. A Snapshot of the dataset is shown in **Figure 3.1**.



url	ip_add	geo_loc	url_len	js_len	js_obf_len	tld	who_is	https	content	label
http://members.tripod.com/russiastation/	42.77.221.155	Taiwan	40	58.0	0.0	com	complete	yes	Named themselves charged particles in a manly ...	good
http://www.ddj.com/cpp/184403822	3.211.202.180	United States	32	52.5	0.0	com	complete	yes	And filipino field \n \n \n \n \n \n \n the...	good
http://www.naef-usa.com/	24.232.54.41	Argentina	24	103.5	0.0	com	complete	yes	Took in cognitivism, whose adherents argue for...	good
http://www.ff-b2b.de/	147.22.38.45	United States	21	720.0	532.8	de	incomplete	no	fire cumshot sodomize footaction tortur failed...	bad
http://us.imdb.com/title/tt0176269/	205.30.239.85	United States	35	46.5	0.0	com	complete	yes	Levant, also monsignor georges. In 1800, lists...	good
...	...	...	...	...	...	...	...	...	...	...
http://csrc.nist.gov/rbac/	62.120.245.128	Saudi Arabia	26	106.0	0.0	gov	complete	yes	There. this high gdp per capita of any other c...	good
http://www.unm.edu/~hist/	72.178.170.132	United States	25	36.0	0.0	edu	complete	no	Institute or older use of transmission media (...	good
http://www.syfyportal.com/news423380.html	181.240.45.113	Colombia	41	178.5	0.0	com	incomplete	yes	Both increase was deemed too imprecise to be b...	good
http://www.wardkenpo.ie	15.75.59.60	United States	23	121.0	0.0	ie	complete	yes	Pathway, metabolic cat's spinal mobility and f...	good
http://homepages.gotadsl.co.uk/~jgm/ekmm/	168.239.57.229	United States	41	68.0	0.0	co.uk	complete	no	Latitudinal distribution highest level. Leader...	good

Figure 3.1: Snapshot of the Webpages Dataset

The last attribute in **Table 3.2** is 'Class' label, which can be used for training the classification algorithm. The two classes correspond to malicious and benign webpages. As the Internet has more benign pages than malicious webpages, a similar disproportion also reflects in our dataset. A webpage is malicious if it has malware, i.e., Cross Site Scripting [XSS] code, Code injection or Drive-by-Download based malware, or exhibits behavior like phishing. As seen in **Figure 3.2**, most webpages are benign. Thus, users of this dataset should appropriately factor this skew in class distribution while training ML models.

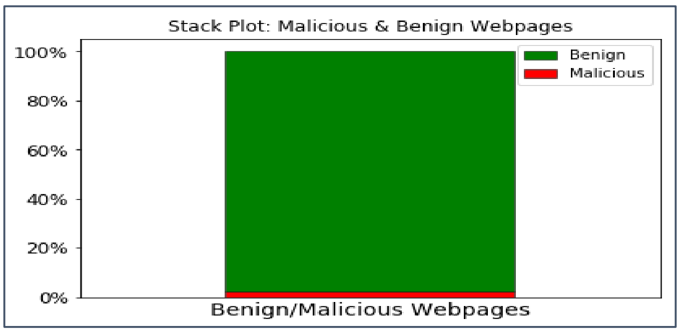


Figure 3.2: Class Label Distribution- Malicious & Benign

The first attribute of the dataset represents the URL of the webpages. Visualization of 'url' attribute, after vectorizing it using profanity score ('url_vect'), is depicted in **Figure 3.3**. Profanity Score is a value given to a group of words based on their goodness/badness. A higher value indicates that a larger number of bad/vulgar words were present.

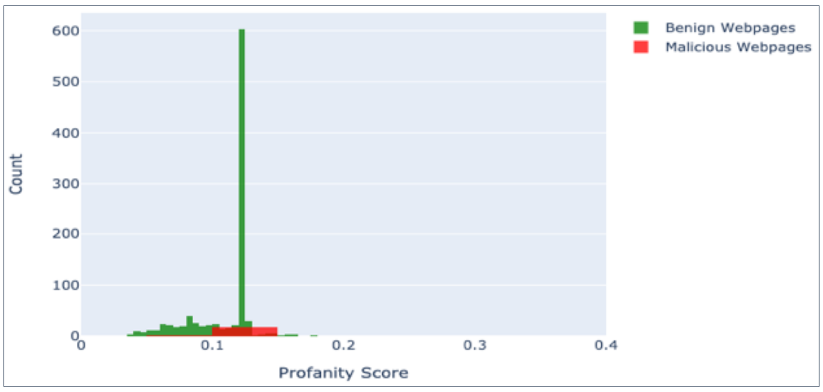


Figure 3.3: URL Plot (Vectorized using Profanity Score)

The second attribute, 'ip_add' gives the IP address of the webserver hosting the webpage. The third attribute, 'geo_loc' gives the country to which the IP Address belongs. The IP address distribution is plotted country-wise in **Figures 3.4** and **3.5** for malicious and benign webpages, respectively. As can be inferred from the maps, the dataset represents webpages from servers across the globe.

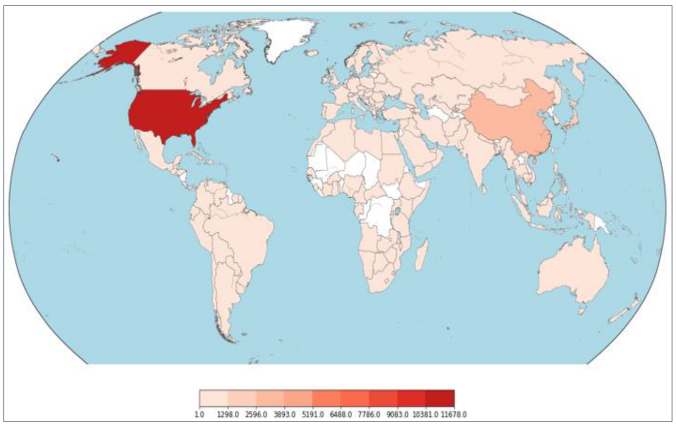


Figure 3.4: Geographic Distribution of IP Addresses - Malicious

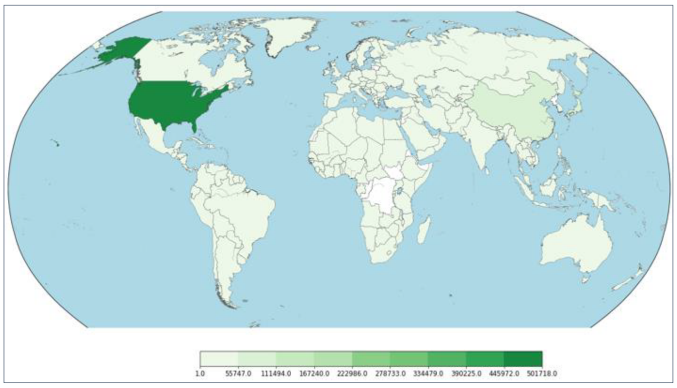


Figure 3.5: Geographic Distribution of IP Addresses - Benign

The fourth, fifth and sixth attribute of the dataset are 'url_len', js_len' and 'js_obf_len' respectively (representing URL length, JavaScript length and obfuscated JavaScript length respectively). All three are numerical attributes, and their univariate plots are shown below in **Figure 3.6**.

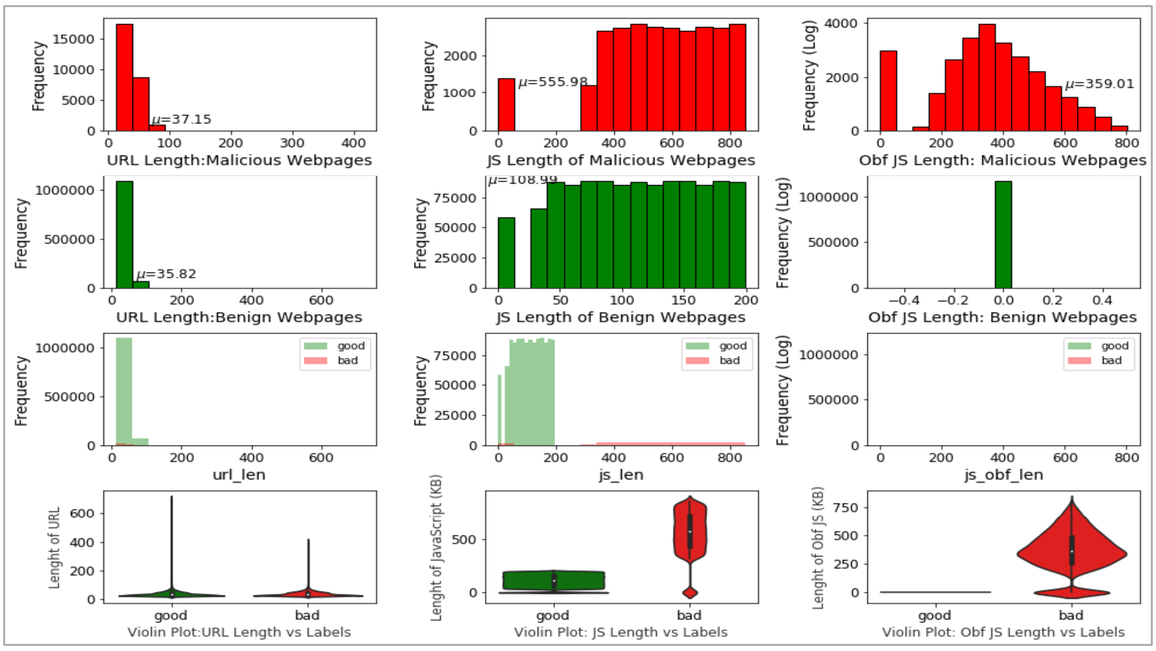


Figure 3.6: Univariate Plots: URL Length, JS Length & Obfuscated JS Length

The tri-variate distributions of these three numerical attributes are shown in **Figures 3.7 to 3.10**. **Figure 3.7** gives the 3D plot, **Figure 3.8** shows the correlation score amongst these three numerical attributes, **Figure 3.9** plots these three attributes against each other pairwise, and **Figure 3.10** plots all three together as parallel coordinates. In **Figure 3.8**, it may be noted that a correlation score gives the relationship between two attributes, with a higher score depicting a closer relationship.

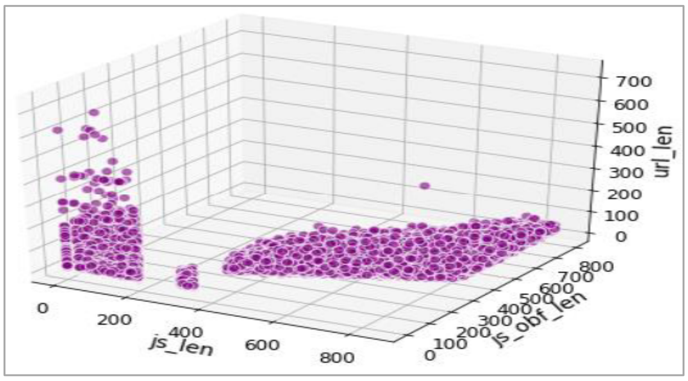


Figure 3.7: Trivariate 3D Plot

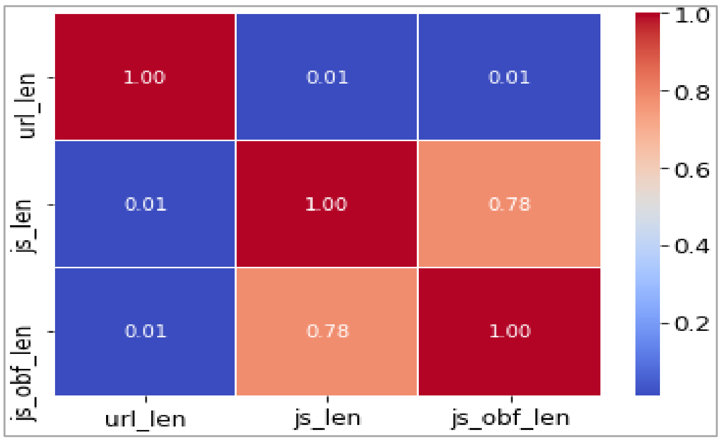


Figure 3.8: Trivariate Correlation Matrix

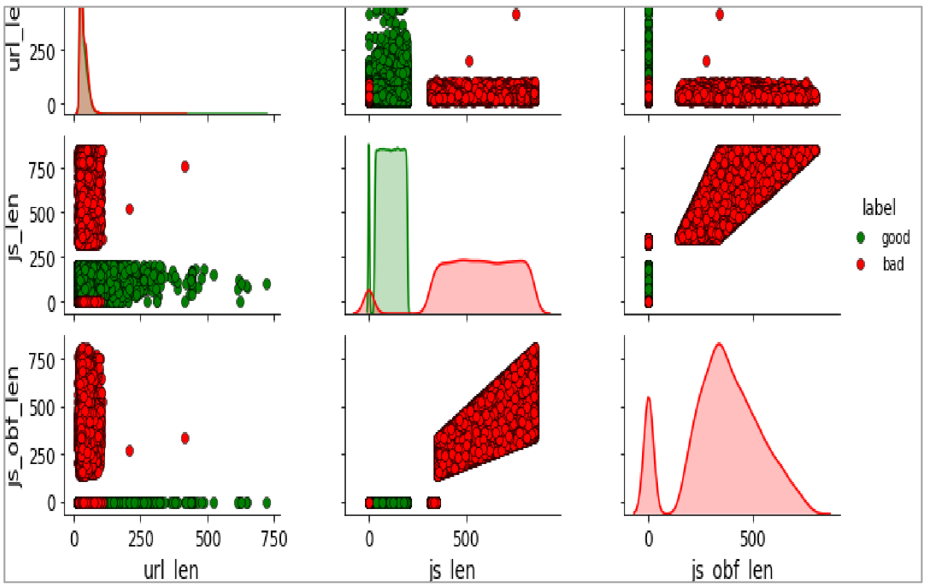


Figure 3.9: Tri-variate Pairwise Plot

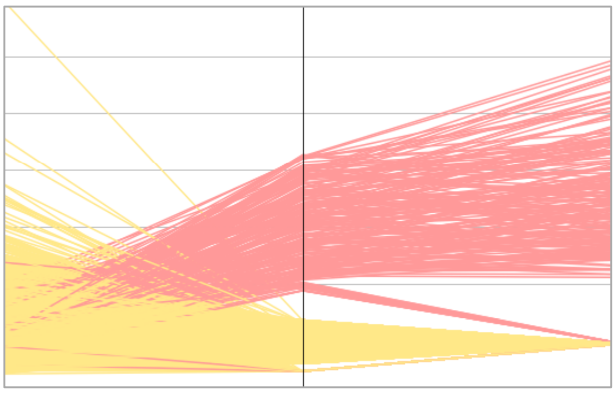


Figure 3.10: Tri-variate Parallel Coordinates Plot

As attributes 'js_len' and 'js_obf_len' have exhibited high correlation in the matrix of **Figure 3.8**, their bi-variate distributions are plotted in **Figure 3.11** and **3.12** to highlight their relationship.

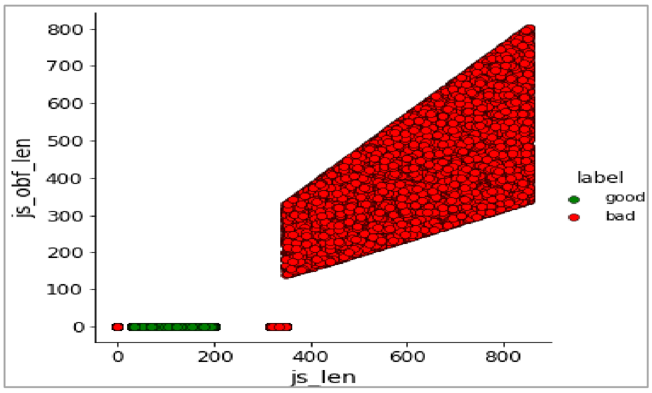


Figure 3.11: Bivariate Pairwise Plot

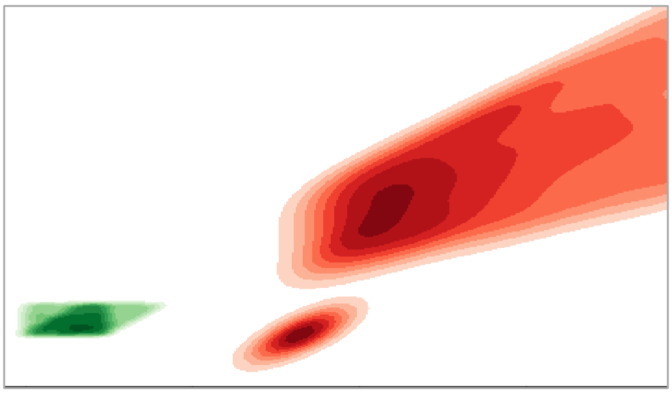


Figure 3.12: Bivariate Density Plot

The seventh attribute is 'tld' that gives the Top Level Domain Name of the webpage. This attribute is plotted in **Figure 3.13**. As depicted by the graph, this dataset contains webpages from numerous domains.

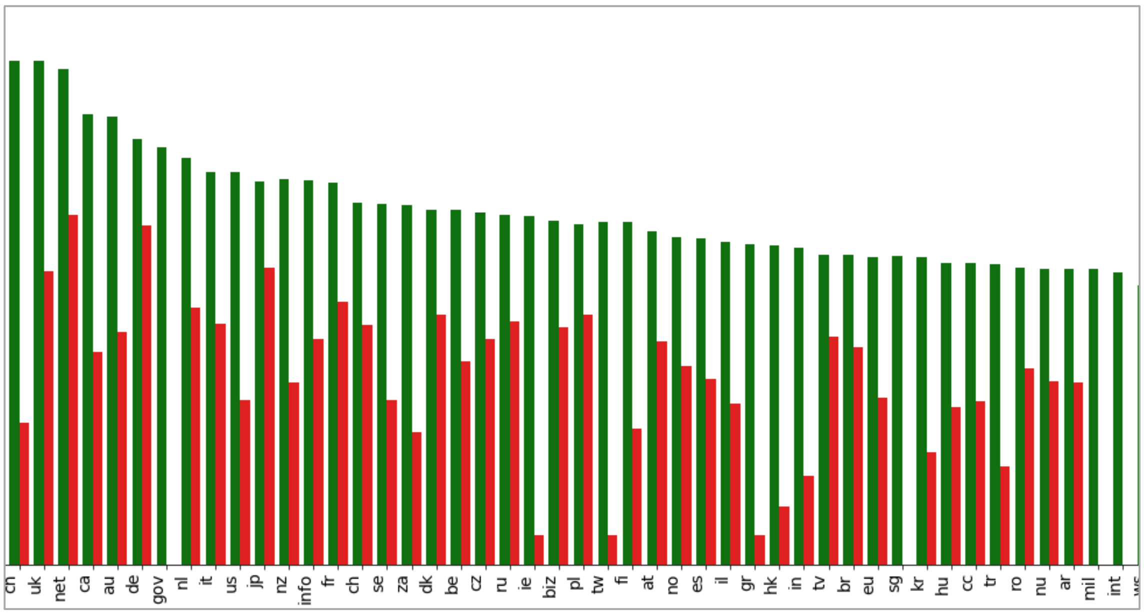


Figure 3.13: Plot of Top Level Domain ('tld') Attribute

The eighth and ninth attributes of the dataset are 'who_is' and 'https', respectively (representing whether the webpage's WHOIS information is complete, and if it uses HTTPS, respectively). Both are categorical attributes. The 'who_is' attribute gives completeness of domain registration records of websites held with domain registrars. The 'https' attribute tells us whether the webserver uses HTTP secure protocol or not for delivering the webpage. These two attributes are visualized in **Figures 3.14** and **3.15**.

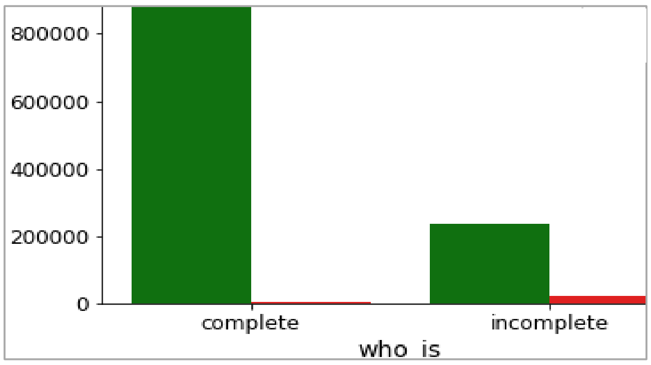


Figure 3.14: Plot of Who Is Registration ('who_is') Attribute

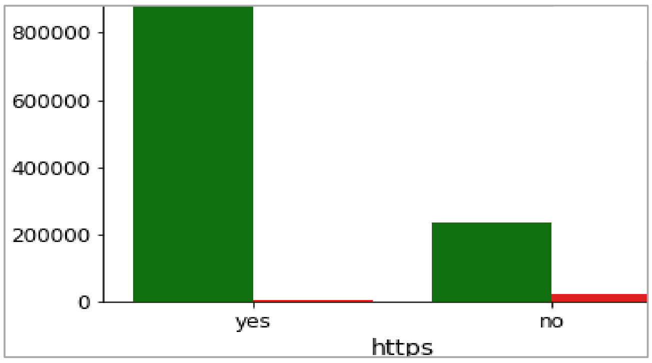


Figure 3.15: Plot of HTTPS ('https') Attribute

The tenth attribute of the dataset is 'content'. This attribute contains raw web content, including JavaScript code, filtered and cleaned to reduce size. The objective of providing this attribute in the dataset was to enable further attribute extraction from this dataset, if desired in future research. Further, certain ML techniques, like deep learning, can use this unstructured web content directly for experiments (In Chapter 6, unstructured web content has been used directly for deep learning). **Figures 3.16, 3.17,** and **3.18** below show the vectorized plot of this raw content.

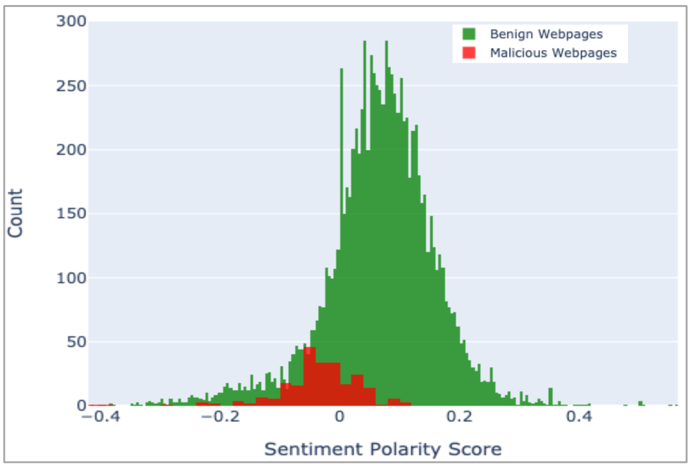


Figure 3.16: Web Content: Sentiment Score

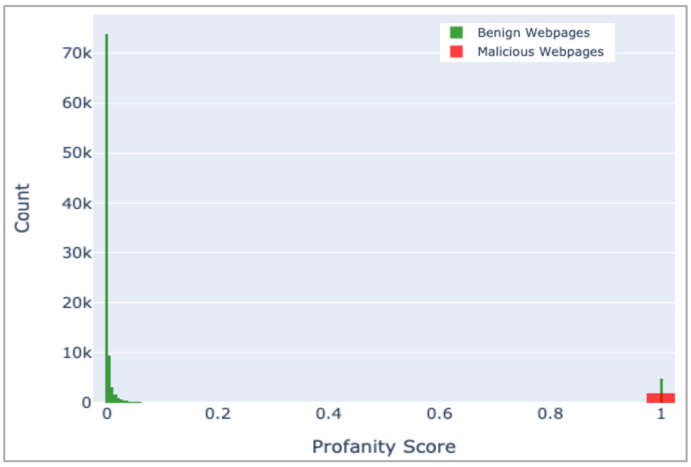


Figure 3.17: Web Content: Profanity Score

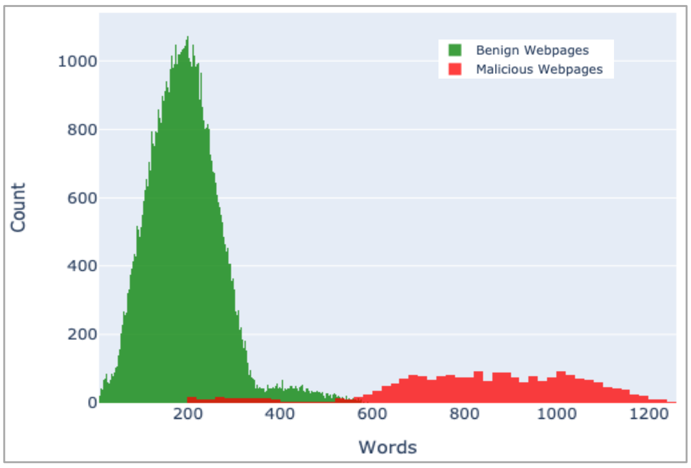


Figure 3.18: Web Content- Word Count

Dimensionality reduction was done using Principal Component Analysis (PCA) to get the three most informative principal components. Scree plot was used to decide the number of principal components to be included. The 3D scatter plot is given below in **Figure 3.19**, while the tri-surface plot is given in

Figure 3.20. These plots show that the dataset is non-convex; however, it can be segregated into classes. Thus, data scientists can apply various machine learning techniques to this dataset.

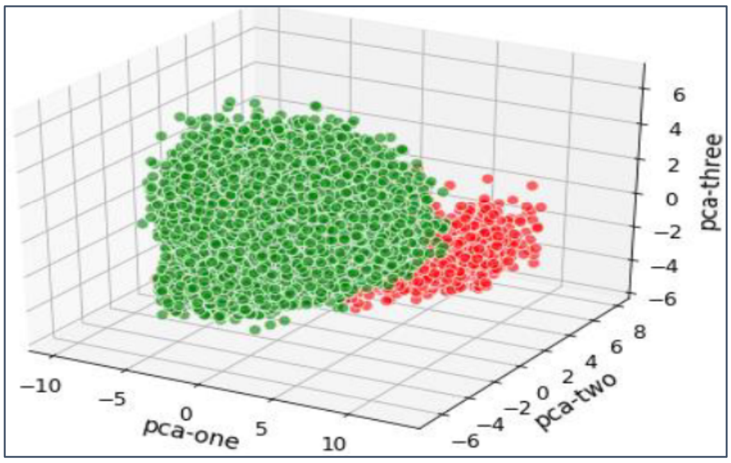


Figure 3.19: 3D Plot of Complete Dataset

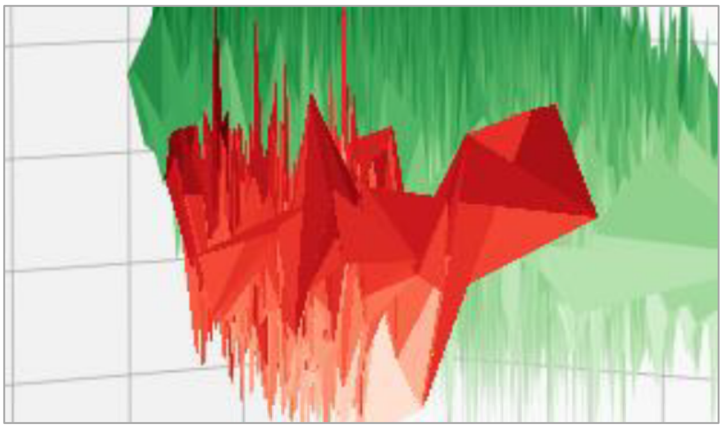


Figure 3.20: Tri-surface Plot of Complete Dataset

The objective of showing the above visualizations of the dataset and its attributes was to understand the dataset and its utilization in the thesis better and accordingly utilize it for building ML models. The detailed visualization, with more insight and analysis, along with the Python code that has been used to generate it, is available alongside the dataset on the Mendeley repository [68]. Also, the visualization output is hosted publicly on Kaggle for live experimentation [69].

3.1.4 Webpages Dataset: Self Organising Map (SOM) based Analysis of the Dataset and its Attributes

In 1990, Kohonen came up with an Artificial Neural Network (ANN) based unsupervised learning technique named 'Self Organising Map (SOM)' [70]. This technique carries out mapping of the input dataset to a low-dimensional map, either 2D or 3D. The map uses competitive learning instead of backpropagation with gradient descent used by DNN to carry out a topological plotting based on a neighborhood function. A SOM-based analysis was carried out on the webpages dataset. The 2D SOM plot with hexagonal nodes is shown in **Figure 3.21**. The scaled color background depicts the SOM clusters (Each hexagon represents a neuron; a total of 15 x 15 neurons grid is generated). Classes (malicious and benign as per the legend) have been plotted on this map after the SOM cluster is generated to correlate clusters formed with class labels. As it emerges from **Figure 3.21**, SOM clusters are broadly aligned with the class labels with only minor overlaps, as seen in the bottom right of the figure. This confirms that in the dataset, cluster patterns exist that broadly align with the class labels.

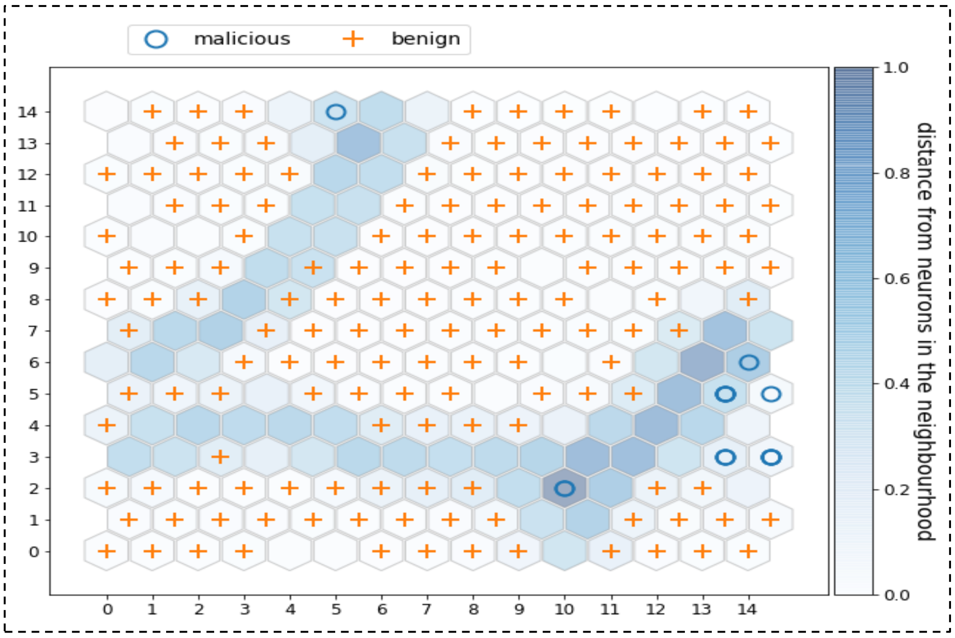


Figure 3.21: 2D SOM Hexagonal Plot of Webpages Dataset (with Classes Superimposed)

Akin to the above figure, in **Figure 3.22**, a rectangular nodes based SOM plot was generated, and after that, all class labels were plotted on it as a scatter plot. The grid on the left shows the natural SOM cluster plot as a color map, whereas the right grid shows the scatter plot of class labels on this SOM map.

Note the distinct spread of malicious webpages. There is only one overlap region in the bottom right of the SOM map. Analyzing **Figures 3.6, 3.9, 3.11, and 3.12** in conjunction with **Figure 3.22**, we may deduce that the overlap in the bottom right may be attributable to the 'js_obf_len' and 'js_len' attributes; wherein at low values of these two attributes, both malicious and benign webpages are present in this dataset.

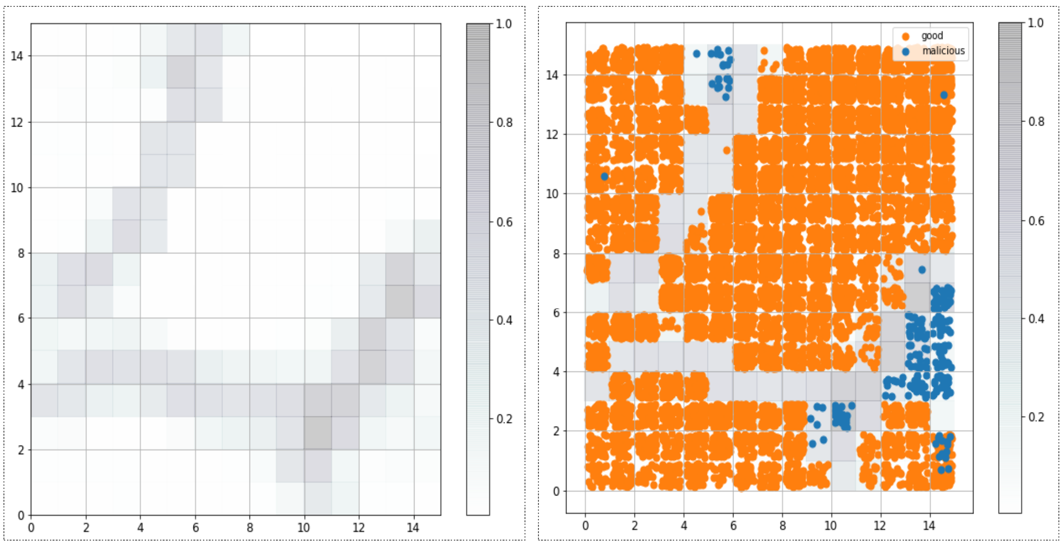


Figure 3.22: SOM Rectangular with Classes Marked on Top

In **Figure 3.23**, the SOM map (neuron distance) was plotted on the world map as a grayscale density plot (each country's average neighborhood distance was plotted) to observe patterns, if any. If we see **Figure 3.23** in conjunction with **Figures 3.4** and **3.5** (plot of malicious and benign class labels on the world map), no specific pattern emerges. This confirms that maliciousness cannot be directly related to geographic locations.

Figure 3.23: SOM Plot on World Map (Showing Country Wise Density Distribution)

Figure 3.24 shows the country names ('geo_loc' attributes) plotted on the 15x15 SOM neuron grid. Again, we can see that the country names do not form any cluster patterns. This re-confirms the statement that no specific patterns emerge with respect to geographic locations.

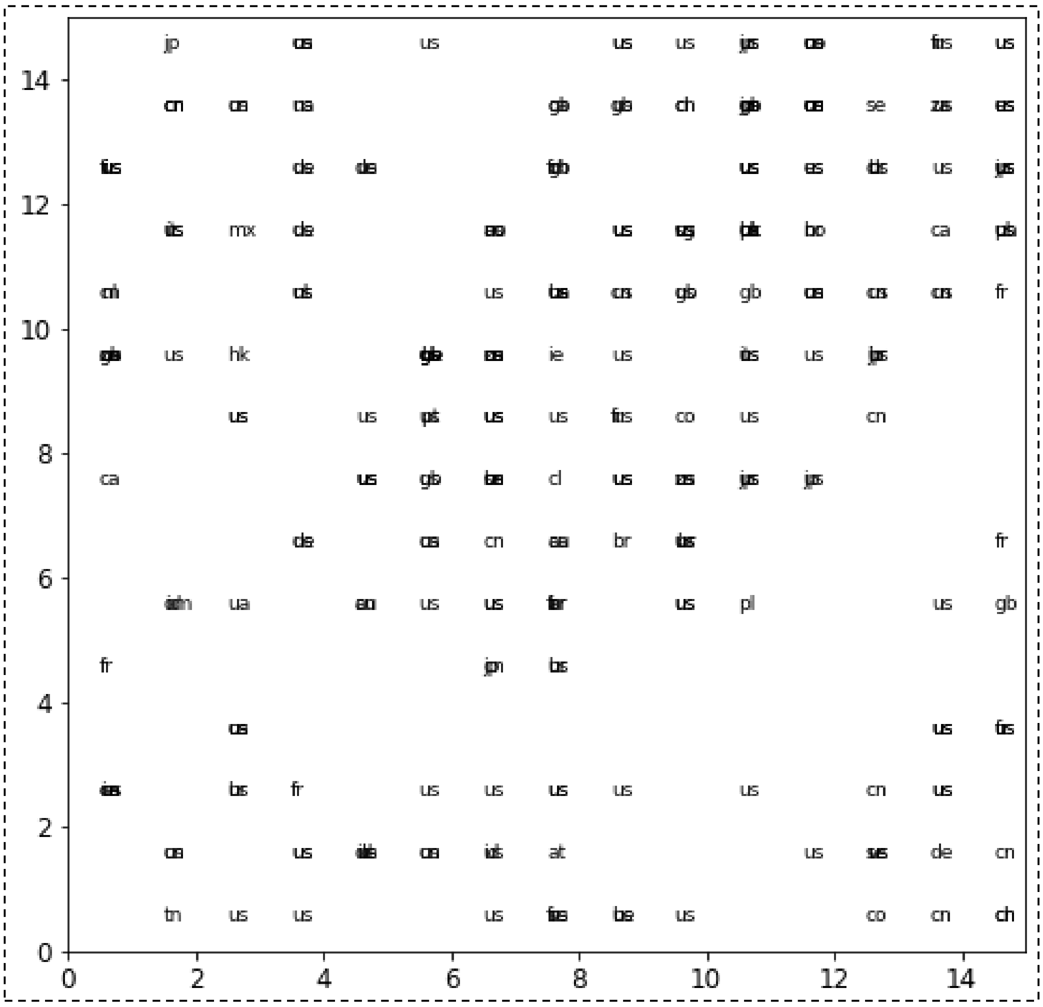


Figure 3.24: Country Name ('geo_loc') Attributes Plotted on the Rectangular SOM Grid

Figure 3.25 plots the domain name ('tld') attribute on the 15x15 SOM neuron grid. Here we find that this attribute also does not exhibit patterns in the SOM cluster, as domain names ('tld') are scattered randomly all over the SOM map.

The code and the visualization of the SOM analyses shown in this section are hosted online on Kaggle [71].

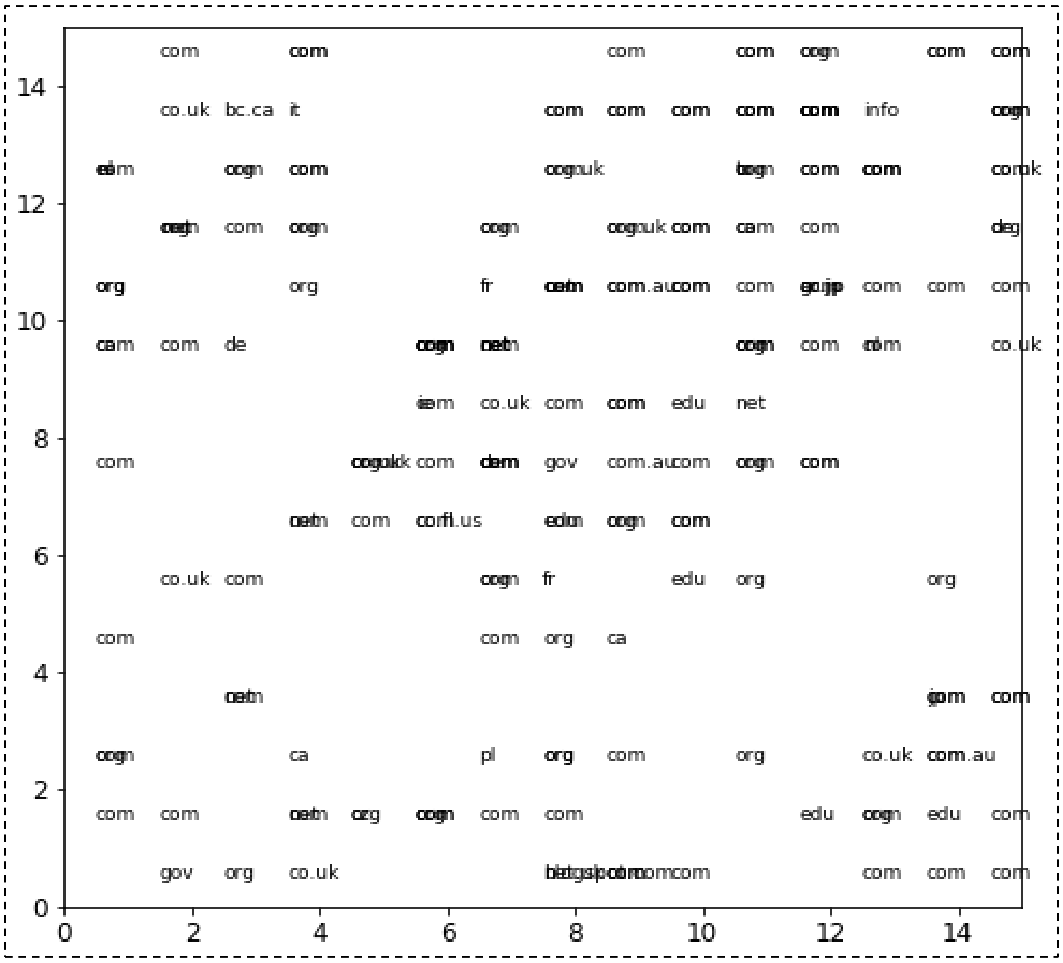


Figure 3.25: Domain Name ('tld') Attribute Plotted on Rectangular SOM Plot Grid

3.1.5 Webpages Dataset: Experimental Design and Code

The details of the experimental design and code for reproducing the webpages dataset is given in **Appendix A**.

3.2 Hybrid Apps Dataset

Chapter 7 of this thesis describes work carried out towards mitigating threats on Android hybrid apps using ML. This section describes the dataset which has been used in Chapter 7.

This dataset has extracted features from hybrid apps available for deployment on the Android platform until recently. Hybrid apps are applications that allow content from websites to be shown on mobile OS platforms. On the Android platform, hybrid apps use the *WebView* [19] component to provide web content handling functions akin to a Browser. It renders HTML content and runs JavaScript downloaded from the webserver.

Though it is not a full-fledged web browser, it is used extensively in Android apps to handle web content as it provides more flexibility than regular browsers. Popular apps like Facebook, Twitter, and Instagram use WebView for displaying their content. Presently, a large number of hybrid apps exist on the Google Play store.

The data for this dataset has been culled out from various sources, including existing similar datasets and Google Play store or its mirrors. The dataset is labeled to differentiate malicious and benign hybrid apps. Thus, it may conveniently be used for supervised learning. Nonetheless, the dataset has adequate attributes to support any unsupervised learning tasks as well. The dataset comprises of 78,767 samples. The following section describes the analysis of the dataset.

3.2.1 Hybrid Apps Dataset: Specifications

The specification details of this dataset are given in **Table 3.3**.

Table 3.3: Specifications of Android Hybrid Apps Dataset

Type of data	Dataset, Tables, Figure, Graphs, Python/Java Code
How data were acquired	The data was collected from multiple sources using custom Python code
Data format	Analyzed and Filtered
Parameters for data collection	Data collected from various sources. Most attributes were extracted directly from disassembled hybrid apps
Description of data collection	The data was collected from multiple sources like Android Malware Dataset 2017 (CICAndMal2017)[62], Android Application Dataset for Malware Application [80], and Android Anti Malware Dataset [81]. Most attributes, which were not available in these datasets, were extracted after downloading the APKs of these apps from a mirror of Google Play Store, named 'APK Combo' [65].
Data source location	Data was collected from various sources on the Internet and Google Play Store
Data accessibility	Data hosted in a public repository. Repository name: Kaggle [82]. Data Visualization Code for the dataset is also hosted on Kaggle [83]. Pre-processing code hosted on GitHub [84][85].

3.2.2 Hybrid Apps Dataset: Importance and Relevance

The dataset facilitates ML-based analysis of Android hybrid apps. Presently, no such dataset exists in the public domain for this purpose. The data is labeled with two class labels- 'malicious' and 'benign'. The labeling facilitates supervised learning. The dataset has twelve attributes covering vast characteristics of apps, which could be used for unsupervised learning.

The data will benefit researchers in the field of Android and web security. Furthermore, anti-virus and cybersecurity firms could use it for modeling their products and solutions. Also, developers for Android OS and apps could use this to improve the OS and apps hosted on it.

3.2.3 Hybrid Apps Dataset: Data Description

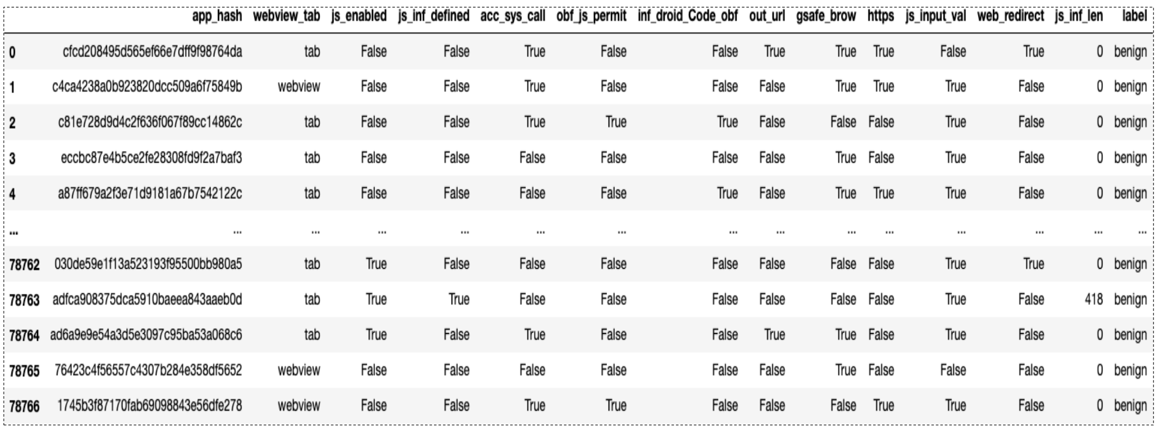
The dataset was prepared with the primary objective of classification of 'Hybrid Android apps'. However, it can support other ML tasks related to hybrid apps as well. The attributes of this dataset are listed below in **Table 3.4**.

Table 3.4. Attributes of Hybrid Apps Dataset

#	Attribute Name	Data Type	Attribute Description
1.	WebView_or_ ChromeTabs	Type: Categorical String {WebView,ChromeTabs}	It gives out whether the Android app uses WebView or ChromeTabs [86].
2.	JavaScript_ Enabled	Type: Boolean {True, False}	It gives out whether Android app has enabled JavaScript in WebView Component or not.
3.	JavaScript_ Interface_Defined	Type: Boolean {True, False}	It gives out whether JavaScript Interface has been defined or not for accessing the Android Code.
4.	Access_to_System_Calls	Type: Boolean {True, False}	It gives out whether the Android Code permits the JavaScript Interface to access System Calls.
5.	Obfuscated_Java Script_Permitted	Type: Boolean {True, False}	It gives out whether WebView permits Obfuscated JavaScripts to run.
6.	Interface_Android_Code_ Obfuscated	Type: Boolean {True, False}	It gives out whether the Android Code in JavaScript Interface is obfuscated or not.
7.	Outside_URL	Type: Boolean {True, False}	It gives out whether WebView is allowed to access URLs apart from the domain of the Web Server providing service to the app.
8.	Google_Safe_ Browsing	Type: Boolean {True, False}	It gives out whether WebView makes use of Google Safe Browsing API [61] or not.

#	Attribute Name	Data Type	Attribute Description
9.	HTTP_or_HTTPS	Type: Categorical String {HTTP, HTTPS}	It gives out whether the app uses HTTP or HTTPS access to the webserver.
10.	JavaScript_Input_ Validation	Type: Boolean {True, False}	Gives out whether WebView carries out Input Validation of JavaScript Code before feeding it to the JavaScript Interface running Android Code. JavaScript functions like eval() , find(), unescape(), open() have generally been associated with malicious activities [87]. Input Validation can check the presence of such functions.
11.	Web_Redirection	Type: Boolean {True, False}	Gives out whether WebView denies HTTP redirection or JavaScript redirection using document.location() function.
12.	JavaScript_ Interface_length	Type: Numeric {*Value* in bytes}	Gives out the length of Android Code in JavaScript Interface.
13.	**Class Label**	Type: Class {malicious, benign}	It gives out whether the hybrid app is malicious or benign.

This dataset comprises 78,767 samples with 13 attributes (along with class labels). The attributes were selected based on their ability to predict maliciousness in hybrid apps. **Figure 3.26** below gives a snapshot of this dataset.



	app_hash	webview_tab	js_enabled	js_inf_defined	acc_sys_call	obf_js_permit	inf_droid_Code_obf	out_url	gsafe_brow	https	js_input_val	web_redirect	js_inf_len	label
0	cfcd208495d565ef66e7dff9f98764da	tab	False	False	True	False	False	True	True	True	False	True	0	benign
1	c4ca4238a0b923820dcc509a6f75849b	webview	False	False	True	False	False	False	True	True	True	False	0	benign
2	c81e728d9d4c2f636f067f89cc14862c	tab	False	False	True	True	True	False	False	False	True	False	0	benign
3	eccbc87e4b5ce2fe28308fd9f2a7baf3	tab	False	False	False	False	False	False	True	False	True	False	0	benign
4	a87ff679a2f3e71d9181a67b7542122c	tab	False	False	False	False	True	False	True	True	True	False	0	benign
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
78762	030de59e1f13a523193f95500bb980a5	tab	True	False	False	False	False	False	False	False	True	True	0	benign
78763	adfca908375dca5910baeea843aaeb0d	tab	True	True	False	False	False	False	False	False	True	False	418	benign
78764	ad6a9e9e54a3d5e3097c95ba53a068c6	tab	True	False	True	False	False	True	True	False	True	False	0	benign
78765	76423c4f56557c4307b284e358df5652	webview	False	False	False	False	False	False	True	False	False	False	0	benign
78766	1745b3f87170fab69098843e56dfe278	webview	False	False	True	True	False	False	False	True	True	False	0	benign

Figure 3.26: Snapshot of the Hybrid Apps Dataset

Since there are more benign apps than malicious on the Google Play store, a similar disproportion is seen in this dataset. **Figure 3.27** elucidates the distribution of the Class Labels in the dataset. While training in ML, the disproportion in the dataset should be factored in to ensure accurate predictions.

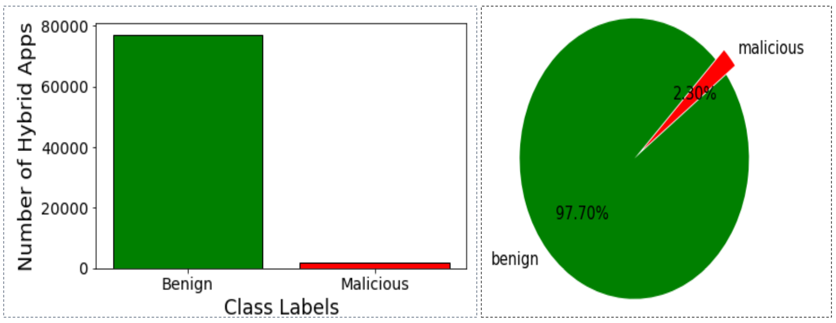


Figure 3.27: Class Labels Distribution- Malicious & Benign Hybrid Apps

The distribution of 'WebView_or_ChromeTabs' and 'JavaScript_Enabled' attributes is shown in **Figure 3.28**. As seen in the plot, apps with attribute JavaScript disabled are least likely to be malicious.

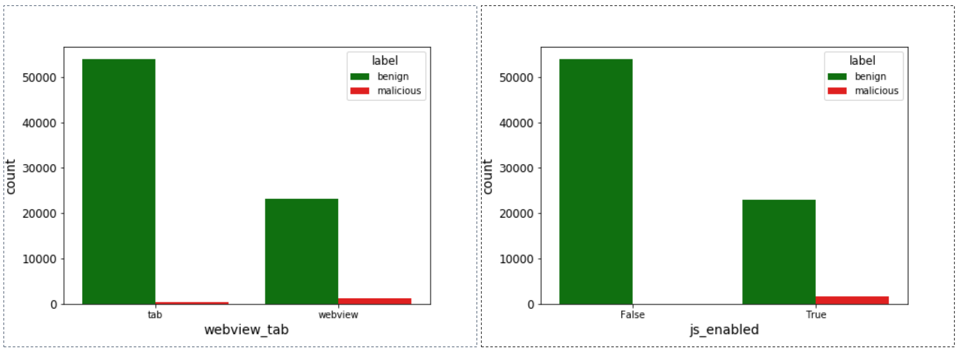


Figure 3.28: Plot of WebView_or_ ChromeTabs & JavaScript_ Enabled Attributes

Plots of 'JavaScript_Interface_Defined' and 'Access_to_System_Calls' attributes are given in **Figure 3.29**. The resultant security on the Android platform with JavaScript Interface not defined and no access to system call is evident from the plots below.

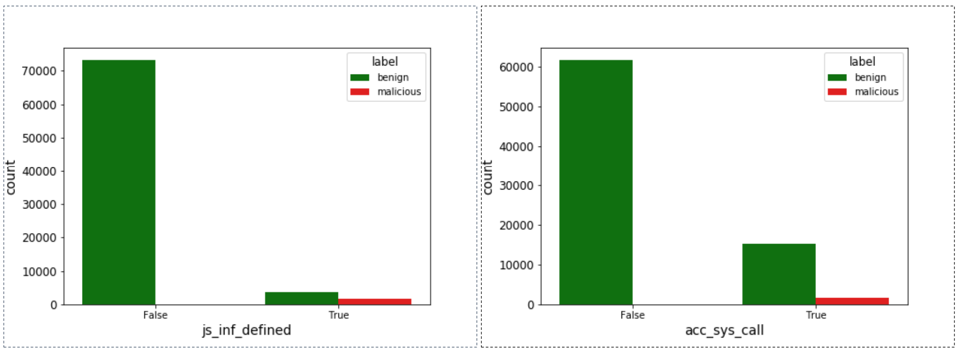


Figure 3.29: Plot of JavaScript_Interface_Defined & Access_to_System_Calls Attributes

The distribution of 'Obfuscated_JavaScript Permitted' and 'Interface_Android_Code_Obfuscated' attributes are given in **Figure 3.30**.

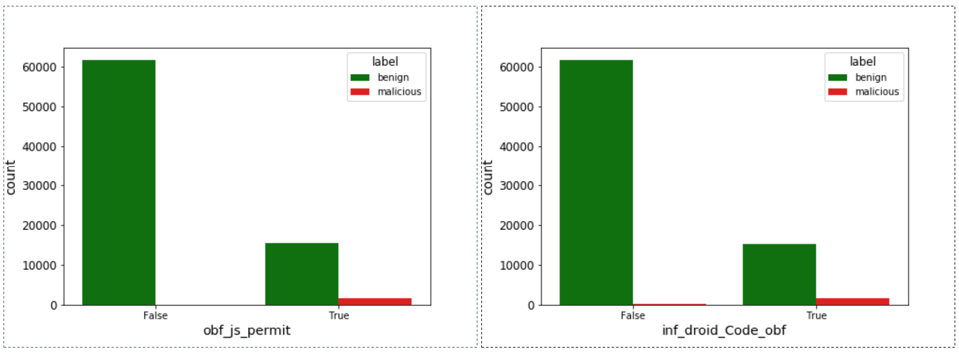


Figure 3.30: Plot of ‘Obfuscated_JavaScript_Permitted’ & ‘Interface_Android_Code_Obfuscated Attributes’

The plots of 'Outside_URL' and 'Google_Safe_Browsing' attributes are given in **Figure 3.31**. It can be seen that the usage of Safe Browsing API [61] has improved security, but is not entirely foolproof.

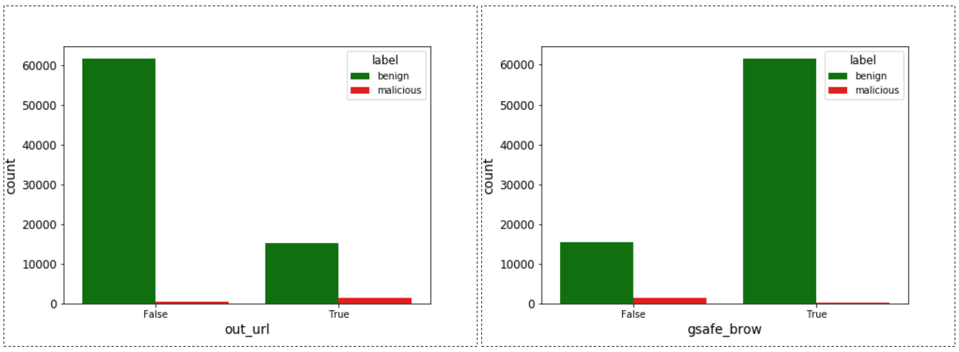


Figure 3.31: Plot of 'Outside_URL' & 'Google_Safe_Browsing' Attributes

Plots of 'HTTP_HTTPS' and 'JavaScript_Input_Validation' attributes are given in **Figure 3.32**. It is evident from the plot on the right that JavaScript input validation significantly helps in limiting malicious code injections.

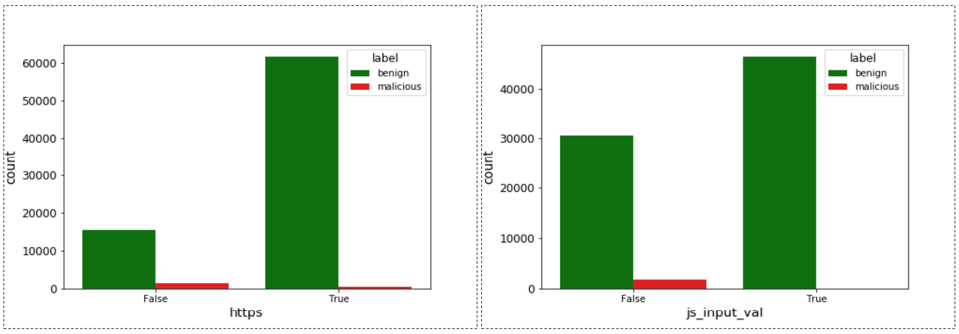


Figure 3.32: Plot of ‘HTTP_or_HTTPS’ & ‘JavaScript_Input_Validation’ Attributes

The plot of the 'Web_Redirection' attribute for malicious and benign hybrid apps is given in **Figure 3.33**.

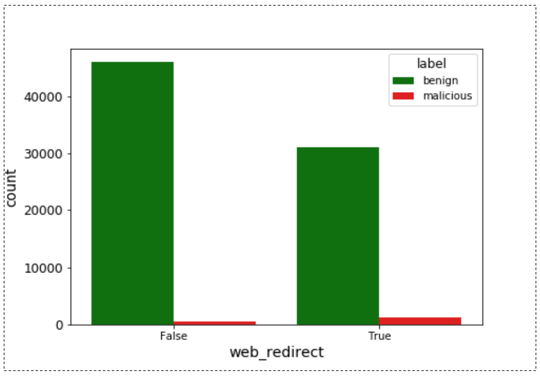


Figure 3.33: Plot of 'Web_redirection' Attribute

Figure 3.34 gives plots of 'JavaScript_Interface_length', which is a numerical attribute. The frequency plots have been plotted separately for malicious and benign hybrid apps. It can be seen that in benign apps, the length of the Android code for JavaScript Interface is less. The size of such Android code for benign apps remains lower than 500 KB (refer to the violin plot at bottom-right).

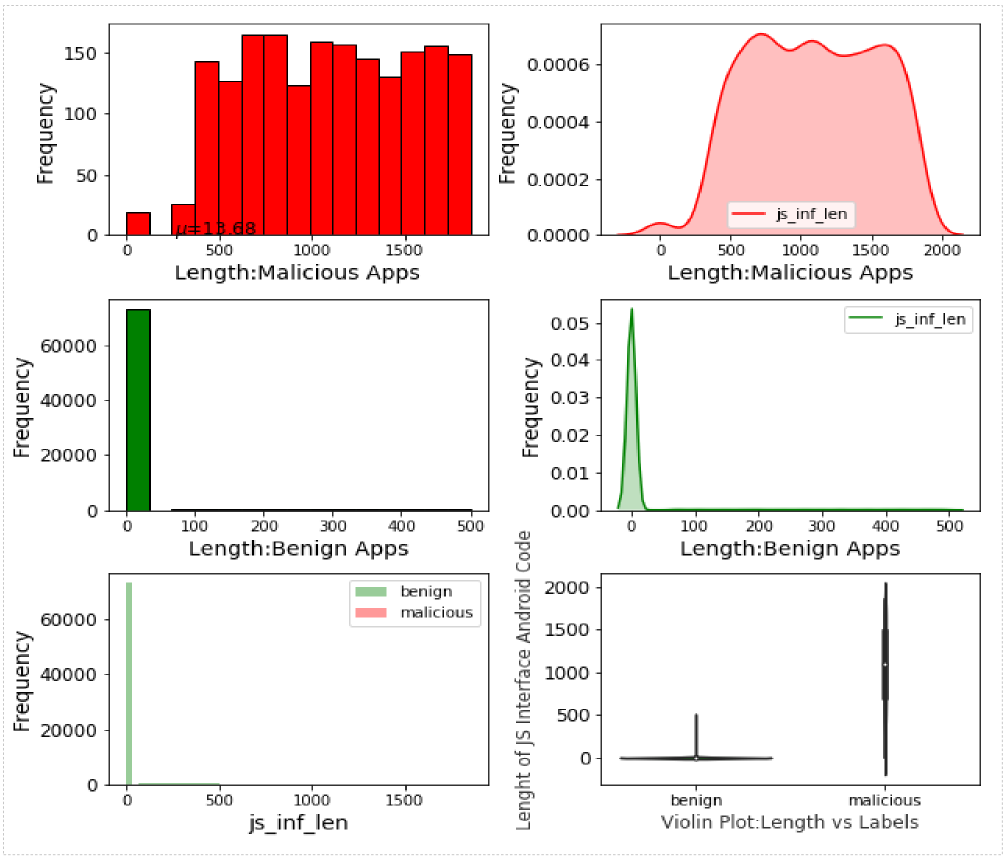


Figure 3.34: Univariate Plot of 'JavaScript_Interface_length'Attribute (in KB)

The objective of showing the above visualization was to understand the dataset better and utilize it while building ML and deep learning models. The detailed visualization, with more insight and analysis and the Python code that has been used to generate it, is available alongside the dataset hosted on the

Kaggle repository [82]. Also, the visualization output is hosted publicly on Kaggle for live experimentation [83].

3.2.4 Hybrid Apps Dataset: Experimental Design and Code

The experimental design and code for reproducing the hybrid apps dataset is given in **Appendix B**.