1. DATA PREPARATION

January 2, 2026

0.0.1 Phishing URL Website detection using Machine Learning

Problem Statement The problem is to detect whether a URL is phishing or legitimate using Machine Learning. The phishing URLs looks like legitimate but redirects to a malicious website. So, our goal is to build a detection system that detects any type of URL using Machine Learning with low latency.

URLs are taken from different data sources and combined together. By carefully observing those URLs, the common patterns I found are - Some of the Phishing URLs have HTTP protocol. This may be due to the ease of setup and less secure. But most of the Phishing Websites have HTTPS protocol overcoming the security. - Rarely FTP protocol is used to download malicious content by clicking on the URL. - Many of the Phishing URLs use IP address as the domain name. This sometimes make the users harder to identify the site as fradulent. - Phishing URLs length is very larger. Many characters are present in Path, Domain and Query. - Phishing URLs embed hexadecimal characters to escape detection by security.

[2]:
```
import pandas as pd
import numpy as np
import warnings

warnings.filterwarnings('ignore')
```

[3]:
```
data = pd.read_csv('data/raw/raw_data.csv')

data.head()
```

[3]:

	url	label
0	https://adoaeco.cn/Loggin	phishing
1	https://gageparkhighschool.com/QTeuUe	phishing
2	https://wwnox.miraltek.cfd/qzxn3	phishing
3	https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn	phishing
4	https://yqcjl.miraltek.cfd/plis0	phishing

[4]:
```
print(f'The dataset consists of {data.shape[0]} rows and {data.shape[1]}␣
↪columns')
```

```
The dataset consists of 253098 rows and 2 columns
```

[5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253098 entries, 0 to 253097
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype 
---  ------  --------------   ----- 
 0   url     253098 non-null  object
 1   label   253098 non-null  object
dtypes: object(2)
memory usage: 3.9+ MB
```

[6]:
```
data.describe()
```

[6]:

	url	label
count	253098	253098
unique	253098	2
top	https://adoaeco.cn/Loggin	legitimate
freq	1	129420

Cleaning the URLs

[7]:
```
import re

def clean_url(url):

    # Remove zero-width characters
    zero_width = r'[\u200B\u200C\u200D\u2060\uFEFF]'
    url = re.sub(zero_width, '', url)

    # Remove all ASCII control characters (0-31 and 127)
    url = re.sub(r'[\x00-\x1F\x7F]', '', url)

    # Remove non-printable Unicode control symbols
    unicode_controls = r'[\u202A-\u202E\u2066-\u2069]'
    url = re.sub(unicode_controls, '', url)

    # Remove soft hyphens & BOM issues
    url = url.replace('\u00AD', '').replace('\ufeff', '')

    # Normalize accidental triple slashes
    url = re.sub(r':\/\/\/+', '://', url)

    # Fix missing colon after https (caused by invisible chars earlier)
    url = re.sub(r'^(https?)(\/\/)', r'\1://', url)

    # Remove spaces around colon in scheme (malformed URLs)
    url = re.sub(r'^(https?)\s*:\s*\/\/', r'\1://', url)

    # Remove trailing slashes
```

```
    url = url.rstrip('/')

    return url

data['url'] = data['url'].apply(clean_url)
```

Checking null & duplicate values

[8]:
```
data.isnull().sum()
```

[8]:
```
url      0
label    0
dtype: int64
```

[9]:
```
data.duplicated().sum()
```

[9]:
```
np.int64(0)
```

Validating URLs

[10]:
```
url_pattern = r"^(([^:/?#]+):)?(//([^/?#]*))?([^?#]*)(\?([^#]*))?(#(.*))?"

print(sum(data['url'].str.match(url_pattern)))
```

```
253098
```

[11]:
```
data.to_csv('data/transformed/cleaned_raw_data.csv',index=False)
```

On carefully observing the data, I found that dataset contains, many shorten URLs. Some shorten URLs may work and some may not. The working shortened URLs are expanded by using a cache of shortening sub-domains and async functions.

[12]:
```
# dataset consists of unshorten URLs
df = pd.read_csv('data/raw/expanded_urls.csv')
df.head()
```

[12]:

	url	label	expanded_url
0	https://adoaeco.cn/Loggin	phishing	https://adoaeco.cn/Loggin
1	https://gageparkhighschool.com/QTeuUe	phishing	https://gageparkhighschool.com/QTeuUe
2	https://wwnox.miraltek.cfd/qzxn3	phishing	https://wwnox.miraltek.cfd/qzxn3
3	https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn	phishing	https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn
4	https://yqcjl.miraltek.cfd/plis0	phishing	https://yqcjl.miraltek.cfd/plis0

[13]:
```
sum(df['url'] != df['expanded_url'])
```

[13]:
```
128
```

As many of the shortened URLs are not working properly, only some shortened URLs are expanded.

[14]:
```
def replace_shortened_url(url,expanded_url):
    if url == expanded_url:
        return url
    return expanded_url

data['url'] = df.apply(lambda row:␣
 ↪replace_shortened_url(row['url'],row['expanded_url']),axis=1)
```

[15]:
```
data.head()
```

[15]:

	url	label
0	https://adoaeco.cn/Loggin	phishing
1	https://gageparkhighschool.com/QTeuUe	phishing
2	https://wwnox.miraltek.cfd/qzxn3	phishing
3	https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn	phishing
4	https://yqcjl.miraltek.cfd/plis0	phishing

[16]:
```
data.isnull().sum()
```

[16]:
```
url      0
label    0
dtype: int64
```

[17]:
```
data.duplicated(subset=['url']).sum()
```

[17]:
```
np.int64(47)
```

[18]:
```
data[data.duplicated(subset=['url'])]
```

[18]:

	url	label
4138	https://scanned.page/p/3jY5UN?id=mt8YdgnKiZ	phishing
4139	https://scanned.page/p/3OhzCy?id=nkuyv79qJ7	phishing
4216	https://scanned.page/p/3jY5UN?id=O5rlEUgEjg	phishing
4244	https://scanned.page/p/3OhzCy?id=vNPx5qXIDe	phishing
4889	https://parkingtreeco.com/wp-content/autopas/1…	phishing
5129	https://did.li/secure/	phishing
5325	https://did.li/secure/	phishing
7040	https://did.li/secure/	phishing
7813	https://did.li/secure/	phishing
8131	https://did.li/secure/	phishing
8438	https://did.li/secure/	phishing
8791	https://did.li/secure/	phishing

```
12568                              https://kutt.it/banned    phishing
12668                              https://kutt.it/banned    phishing
12787                              https://kutt.it/banned    phishing
13130                              https://kutt.it/banned    phishing
19683                                      https://hm.ru/    phishing
19723                                      https://hm.ru/    phishing
19789                                      https://hm.ru/    phishing
19829                                      https://hm.ru/    phishing
25207          https://www.hootsuite.com/error/lost-owly    phishing
25209          https://www.hootsuite.com/error/lost-owly    phishing
25210          https://www.hootsuite.com/error/lost-owly    phishing
25211          https://www.hootsuite.com/error/lost-owly    phishing
25330          https://www.hootsuite.com/error/lost-owly    phishing
25340          https://www.hootsuite.com/error/lost-owly    phishing
25506          https://www.hootsuite.com/error/lost-owly    phishing
28165          https://www.hootsuite.com/error/lost-owly    phishing
28414          https://www.hootsuite.com/error/lost-owly    phishing
48351                              https://www.linkedin.com    phishing
49820   https://onedrive.live.com/survey?id=B4DB281230…    phishing
50116          https://www.hootsuite.com/error/lost-owly    phishing
50448          https://www.hootsuite.com/error/lost-owly    phishing
51651                              https://www.linkedin.com    phishing
57954                                      https://hm.ru/    phishing
70923                              https://www.linkedin.com    phishing
77061               https://www.hootsuite.com/pages/owly    phishing
86668                                      https://gg.gg/    phishing
88918                                      https://gg.gg/    phishing
89507                                      https://gg.gg/    phishing
122155                                     https://gg.gg/    phishing
152461           https://publisher.linkvertise.com/adfly  legitimate
155368                                  https://kutt.it/404    phishing
185114  https://aairport-festival-merchandise.myshopif…    phishing
190157       https://href.li/?http://turtletours.org?DBX    phishing
216680               https://www.hootsuite.com/pages/owly    phishing
250652                             https://www.linkedin.com  legitimate
```

[19]:
```
data.drop_duplicates(subset=['url'],inplace=True,ignore_index=True)
```

[20]:
```
data.duplicated().sum()
```

[20]:
```
np.int64(0)
```

[21]:
```
data.shape
```

[21]:
```
(253051, 2)
```

Extracting URL Components : These elements represent the structural building blocks of a URL and are useful for detailed analysis & feature engineering. - Protocol/Scheme of the URL -

Domain of the URL - Path of the URL - Query of the URL - TLD of the URL - SLD of the URL - Path of the URL - Query of the URL

Consider the below URL as an example: https://www.google.com/search/name=phishing_url

- Scheme/Protocol: `https`
- Sub-domain: `www`
- Domain name: `google.com`
- Second-Level Domain: `google`
- Top-Level Domain: `com`
- Path: `search`
- Query: `name=phishing_url`

[22]:
```
url_components_df = data[['url','label']]
```

[23]:
```
# Extracting protocol, domain, path, query, sld, tld & subdomain
from urllib.parse import urlsplit
import tldextract

def parse_url(url):
    try:
        result = urlsplit(url)
        protocol = result.scheme
        domain   = result.netloc
        path     = result.path
        query    = result.query
        ext = tldextract.extract(domain)
        subdomain = ext.subdomain
        sld = ext.domain
        tld = ext.suffix
    except Exception:   # Handles IPv6 Address
        protocol = ""
        domain = ""
        path = ""
        query = ""
        sld = ""
        tld = ""
        subdomain = ""

    return protocol, domain, subdomain, tld, sld, path, query

url_components_df[['protocol','domain','subdomain','tld','sld','path','query']]␣
 ↪= data['url'].apply(parse_url).apply(pd.Series)
```

[24]:
```
# Handling IP-based URLs
url_components_df[url_components_df['domain'].str.fullmatch(r'\d{1,3}(\.
 ↪\d{1,3}){3}')]
```

[24]:

	url	label	protocol	\
2994	https://91.92.241.186	phishing	https	
3667	https://140.99.164.68/x0	phishing	https	
6300	https://31.172.87.101/x0	phishing	https	
14536	https://43.153.99.18	phishing	https	
16733	https://185.187.56.126	phishing	https	
…	…	…	…	
252640	http://67.23.255.34/~comentar/Face/Facebook_Co…	phishing	http	
252716	http://191.101.7.221/fire/aasdqwe	phishing	http	
252961	http://178.217.186.224/panel/etc/info/toke/cp…	phishing	http	
252990	http://185.75.46.73/information.cgi	phishing	http	
252997	http://38.118.40.209/CFIDE/debug/serveur.html?…	phishing	http	

	domain	subdomain	tld	sld	\
2994	91.92.241.186			91.92.241.186	
3667	140.99.164.68			140.99.164.68	
6300	31.172.87.101			31.172.87.101	
14536	43.153.99.18			43.153.99.18	
16733	185.187.56.126			185.187.56.126	
…	…	…	..	…	
252640	67.23.255.34			67.23.255.34	
252716	191.101.7.221			191.101.7.221	
252961	178.217.186.224			178.217.186.224	
252990	185.75.46.73			185.75.46.73	
252997	38.118.40.209			38.118.40.209	

	path	\
2994		
3667	/x0	
6300	/x0	
14536		
16733		
…	…	
252640	/~comentar/Face/Facebook_Comentario.exe	
252716	/fire/aasdqwe	
252961	/panel/etc/info/toke/cp.php	
252990	/information.cgi	
252997	/CFIDE/debug/serveur.html	

	query
2994	
3667	
6300	
14536	
16733	
…	…
252640	

```
252716
252961                                        m=login
252990
252997  c=CPmn-fHTvaye2wEQiLHW9djz6sRC&amp;hl=fr

[1629 rows x 9 columns]
```

[25]:
```
ip_idx = url_components_df[url_components_df['sld'].str.fullmatch(r'\d{1,3}(\.
 ↪\d{1,3}){3}')].index

url_components_df['is_ip'] = False
url_components_df.iloc[ip_idx,-1] = True
url_components_df.iloc[ip_idx,[3,6]] = ''

url_components_df.iloc[ip_idx]
```

[25]:
```
                                                    url     label protocol  \
2994                                https://91.92.241.186  phishing    https
3667                               https://140.99.164.68/x0  phishing    https
6300                               https://31.172.87.101/x0  phishing    https
14536                                  https://43.153.99.18  phishing    https
16733                                https://185.187.56.126  phishing    https
…                                                       …       …       …
252716                  http://191.101.7.221/fire/aasdqwe  phishing     http
252811                              http://91.239.25.38:6892  phishing     http
252961  http://178.217.186.224/panel/etc/info/toke/cp…  phishing     http
252990                http://185.75.46.73/information.cgi  phishing     http
252997  http://38.118.40.209/CFIDE/debug/serveur.html?…  phishing     http

       domain subdomain tld sld                          path  \
2994
3667                                                      /x0
6300                                                      /x0
14536
16733
…       …        …   ..   ..                             …
252716                                          /fire/aasdqwe
252811
252961                              /panel/etc/info/toke/cp.php
252990                                        /information.cgi
252997                               /CFIDE/debug/serveur.html

                                              query  is_ip
2994                                                  True
3667                                                  True
6300                                                  True
14536                                                 True
```

```
16733                                                  True
…                                                 …    …
252716                                                 True
252811                                                 True
252961                                    m=login      True
252990                                                 True
252997  c=CPmn-fHTvaye2wEQiLHW9djz6sRC&amp;hl=fr       True

[2283 rows x 10 columns]
```

[26]:
```
url_components_df.head()
```

[26]:
```
                                          url     label protocol  \
0                      https://adoaeco.cn/Loggin  phishing    https
1           https://gageparkhighschool.com/QTeuUe  phishing    https
2                https://wwnox.miraltek.cfd/qzxn3  phishing    https
3  https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn  phishing    https
4                https://yqcjl.miraltek.cfd/plis0  phishing    https

                   domain subdomain  tld                 sld     path  \
0              adoaeco.cn             cn              adoaeco  /Loggin
1  gageparkhighschool.com            com  gageparkhighschool  /QTeuUe
2      wwnox.miraltek.cfd     wwnox  cfd            miraltek   /qzxn3
3          halfetitur.com            com          halfetitur        /
4      yqcjl.miraltek.cfd     yqcjl  cfd            miraltek   /plis0

                    query  is_ip
0                          False
1                          False
2                          False
3  token=r2I0IU0FEHfPf5Dn  False
4                          False
```

URL Length Features : These metrics help capture patterns that may be indicative of URL behaviour or intent. - Length of the URL - Length of the Domain - Length of the Path - Length of the Query - URL Depth (no. of '/' segments in path) - No. of Subdomains

[27]:
```
url_len_features_df = data[['url','label']]
```

[28]:
```
# Length of the URL
url_len_features_df['url_len'] = data['url'].str.len()
```

[29]:
```
# Length of the Domain
url_len_features_df['domain_len'] = url_components_df['domain'].str.len()
```

[30]:
```
# Length of the Path
def path_length(path):
    if not isinstance(path,str) or not path.strip():
```

```
        return 0

    if path.startswith('/'):
        path = path[1:]

    segments = [seg for seg in path.split('/') if seg]

    return sum(len(seg) for seg in segments)

url_len_features_df['path_len'] = url_components_df['path'].apply(path_length)
```

[31]:
```
# Length of the Query
url_len_features_df['query_len'] = url_components_df['query'].str.len()
```

[32]:
```
# Depth of the URL
protocol_re = re.compile(r'^[a-zA-Z][a-zA-Z0-9+.-]*://)?')

def calculate_url_depth(url):
    url = protocol_re.sub('',url)
    parts = url.split('/',1)

    if len(parts) == 1:
        return 0
    path = parts[1]

    depth = sum(1 for seg in path.split('/') if seg)

    return depth

url_len_features_df['url_depth'] = data['url'].apply(calculate_url_depth)
```

[33]:
```
# Subdomain count
url_len_features_df['subdomain_count'] = url_components_df['subdomain'].str.
 ↪split('.').apply(lambda x: len(x))
url_len_features_df.loc[ip_idx,'subdomain_count'] = 0   # IP-based URLs
```

[34]:
```
url_len_features_df.head()
```

[34]:

	url	label	url_len	\
0	https://adoaeco.cn/Loggin	phishing	25	
1	https://gageparkhighschool.com/QTeuUe	phishing	37	
2	https://wwnox.miraltek.cfd/qzxn3	phishing	32	
3	https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn	phishing	46	
4	https://yqcjl.miraltek.cfd/plis0	phishing	32	

	domain_len	path_len	query_len	url_depth	subdomain_count
0	10	6	0	1	1

```
1          22          6           0          1                1
2          18          5           0          1                1
3          14          0          22          1                1
4          18          5           0          1                1
```

Domain Structure Features : These features may provide insights into domain structure and potential indicators of unusual or suspicious patterns. - TLD Length - Is the domain a IP address? - Does the URL contains Port number?

[35]:
```
domain_features_df = data[['url','label']]
domain_features_df['tld'] = url_components_df[['tld']]
```

[36]:
```
# TLD Length
domain_features_df['tld_len'] = domain_features_df['tld'].str.len()
```

[37]:
```
# Is the domain a IPv4 address?
domain_features_df['url_has_ipv4'] = url_components_df['is_ip']
```

[38]:
```
# Does the URL contains Port number?
from urllib.parse import urlparse

def url_has_port(url):
    try:
        return urlparse(url).port is not None
    except Exception:
        return False

domain_features_df['url_has_port'] = url_components_df['url'].
 ↪apply(url_has_port)
```

[39]:
```
domain_features_df.head()
```

[39]:
```
                                          url     label  tld  tld_len  \
0                      https://adoaeco.cn/Loggin  phishing   cn        2
1           https://gageparkhighschool.com/QTeuUe  phishing  com        3
2                https://wwnox.miraltek.cfd/qzxn3  phishing  cfd        3
3  https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn  phishing  com        3
4                https://yqcjl.miraltek.cfd/plis0  phishing  cfd        3

   url_has_ipv4  url_has_port
0         False         False
1         False         False
2         False         False
3         False         False
4         False         False
```

SLD-based Features : These features carries meaningful information about the entity represented by the URL. - Length of SLD - SLD has digit? - SLD has hyphen? - SLD Token count

[40]:
```
sld_features_df = data[['url','label']]
sld_features_df['sld'] = url_components_df[['sld']]
```

[41]:
```
# Length of SLD
sld_features_df['sld_len'] = sld_features_df['sld'].str.len()
```

[42]:
```
# SLD has digit?
sld_features_df['sld_has_digit'] = sld_features_df['sld'].str.contains(r'\d')
```

[43]:
```
# SLD has hyphen?
sld_features_df['sld_has_hyphen'] = sld_features_df['sld'].str.contains('-')
```

[44]:
```
# No of tokens in SLD
sld_features_df['sld_token_count'] = (sld_features_df['sld'].str.split('-')).
↪str.len()
```

[45]:
```
sld_features_df.head()
```

[45]:

	url	label	\
0	https://adoaeco.cn/Loggin	phishing	
1	https://gageparkhighschool.com/QTeuUe	phishing	
2	https://wwnox.miraltek.cfd/qzxn3	phishing	
3	https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn	phishing	
4	https://yqcjl.miraltek.cfd/plis0	phishing	

	sld	sld_len	sld_has_digit	sld_has_hyphen	sld_token_count
0	adoaeco	7	False	False	1
1	gageparkhighschool	18	False	False	1
2	miraltek	8	False	False	1
3	halfetitur	10	False	False	1
4	miraltek	8	False	False	1

Character Features : These features may help to reveal obfuscation techniques in phishing URLs. - Dot count in Domain - Hyphen count in domain + path - Underscore count in path + query - Slash count in URL - Digit count in URL - Alphabet count in URL

[46]:
```
char_features_df = data[['url','label']]
```

[47]:
```
# Dot count in Domain
char_features_df['dot_count_domain'] = url_components_df['domain'].str.
↪count(r'\.')
```

[48]:
```
# Hyphen count in domain + path
char_features_df['hyphen_count_domain_path'] = (url_components_df['domain'] +␣
↪url_components_df['path']).str.count('-')
```

[49]:
```
# Underscore count in path + query
```

```
char_features_df['underscore_count_path_query'] = (url_components_df['path'] +␣
↪url_components_df['query']).str.count('\_')
```

[50]:
```
# Slash count in URL
char_features_df['slash_count'] = data['url'].str.count('/')
```

[51]:
```
# Digit count in URL
char_features_df['digit_count'] = data['url'].str.count(r'\d')
```

[52]:
```
# Alphabet count in URL
char_features_df['alphabet_count'] = data['url'].str.count(r'[a-zA-Z]')
```

[53]:
```
# Special characters count in URL
char_features_df['spl_char_count'] = data['url'].str.count(r'[^a-zA-Z0-9]')
```

[54]:
```
char_features_df.head()
```

[54]:

	url	label	dot_count_domain	hyphen_count_domain_path	underscore_count_path_query	slash_count	digit_count	alphabet_count	spl_char_count
0	https://adoaeco.cn/Loggin	phishing	1	0	0	3	0	20	5
1	https://gageparkhighschool.com/QTeuUe	phishing	1	0	0	3	0	32	5
2	https://wwnox.miraltek.cfd/qzxn3	phishing	2	0	0	3	1	25	6
3	https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn	phishing	1	0	0	3	4	35	7
4	https://yqcjl.miraltek.cfd/plis0	phishing	2	0	0	3	1	25	6

Entropy features : Higher entropy often indicates obfuscation or automatically generated strings, which may be strong signals in detecting phishing URLs. - Entropy of URL - Entropy of Domain - Entropy of SLD - Entropy of Path

[55]:
```
# Entropy calculation
import math
from collections import Counter

def shannon_entropy(str_):
    counts = Counter(str_)
```

```
    length = len(str_)

    return -sum((count / length) * math.log2(count / length) for count in␣
 ↪counts.values())
```

[56]:
```
entropy_feature_df = data[['url','label']]
```

[57]:
```
# Entropy of URL
entropy_feature_df['url_entropy'] = data['url'].apply(shannon_entropy)
```

[58]:
```
# Entropy of Domain
entropy_feature_df['domain_entropy'] = url_components_df['domain'].
 ↪apply(shannon_entropy)
```

[59]:
```
# Entropy of SLD
entropy_feature_df['sld_entropy'] = url_components_df['sld'].
 ↪apply(shannon_entropy)
```

[60]:
```
# Entropy of Path
entropy_feature_df['path_entropy'] = url_components_df['path'].
 ↪apply(shannon_entropy)
```

[61]:
```
entropy_feature_df.head()
```

[61]:

	url	label	url_entropy	domain_entropy	sld_entropy	path_entropy
0	https://adoaeco.cn/Loggin	phishing	3.863465	2.721928	2.235926	2.521641
1	https://gageparkhighschool.com/QTeuUe	phishing	4.208925	3.629220	3.419382	2.521641
2	https://wwnox.miraltek.cfd/qzxn3	phishing	4.452820	3.947703	3.000000	2.584963
3	https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn	phishing	4.760096	3.664498	3.121928	-0.000000
4	https://yqcjl.miraltek.cfd/plis0	phishing	4.241729	3.836592	3.000000	2.584963

Token based Features : Tokenization helps revealing structural patterns and semantic clues within the URL. - Token count in domain - Token count in path - Token Count in domain + path - Avg Token Length

[62]:
```
token_features_df = data[['url','label']]
```

[63]:
```
# Calculating Token count in path + domain
def count_tokens(domain,path):
    try:
```

```
        domain_tokens = domain.replace('.',' ').replace('-',' ').replace('_',' ').split()
        path_tokens = path.replace('.',' ').replace('-',' ').replace('_',' ').split()

        total_tokens = len(domain_tokens) + len(path_tokens)

        return len(domain_tokens), len(path_tokens), total_tokens

    except Exception:
        return 0,0,0

token_features_df[['domain_token_count','path_token_count','total_tokens']] = url_components_df.apply(lambda row: count_tokens(row['domain'],row['path']),axis=1).apply(pd.Series)
```

[64]:
```
# Average token length of the URL
protocol_re = re.compile(r'^[a-zA-Z][a-zA-Z0-9+.-]*://',re.IGNORECASE)
split_re = re.compile(r'[./?=\-_&:%]+')

def avg_token_length(url):
    url = protocol_re.sub('',url)
    tokens = split_re.split(url)
    tokens = [t for t in tokens if t]

    if not tokens:
        return 0

    return sum(len(t) for t in tokens) / len(tokens)

token_features_df['avg_token_length'] = data['url'].apply(avg_token_length)
```

[65]:
```
token_features_df.head()
```

[65]:

	url	label	domain_token_count	path_token_count	total_tokens	avg_token_length
0	https://adoaeco.cn/Loggin	phishing	2	1	3	5.00
1	https://gageparkhighschool.com/QTeuUe	phishing	2	1	3	9.00
2	https://wwnox.miraltek.cfd/qzxn3	phishing	3	1	4	5.25
3	https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn	phishing	2	1	3	8.50
4	https://yqcjl.miraltek.cfd/plis0	phishing	3	1	4	5.25

Hexadecimal based Features - URL has Hexadecimal Characters - No. of Hexadecimal Characters - Hexadecimal Ratio

[66]:
```
hex_feature_df = data[['url','label']]
```

[67]:
```
# URL has Hexadecimal Characters?
hex_feature_df['has_hex'] = data['url'].str.contains(r'%[0-9A-Fa-f]{2}')
```

[68]:
```
# No. of hexadecimal characters

def count_hex_chars(url):
    count = 0
    i = 0
    while i < len(url):
        if url[i] == '%' and i+2 < len(url):
            c1, c2 = url[i+1], url[i+2]

            if (c1.isdigit() or c1.lower() in 'abcdef') and (c2.isdigit() or c2.
 ↪lower() in 'abcdef'):
                count += 2
                i += 3
                continue
        i += 1

    return count

hex_feature_df['hex_char_count'] = data['url'].apply(count_hex_chars)
```

[69]:
```
# Hexadecimal Ratio
hex_feature_df['hex_ratio'] = hex_feature_df['hex_char_count'] /␣
 ↪url_len_features_df['url_len']
```

[70]:
```
hex_feature_df.head()
```

[70]:

	url	label	has_hex	\
0	https://adoaeco.cn/Loggin	phishing	False	
1	https://gageparkhighschool.com/QTeuUe	phishing	False	
2	https://wwnox.miraltek.cfd/qzxn3	phishing	False	
3	https://halfetitur.com/?token=r2I0IU0FEHfPf5Dn	phishing	False	
4	https://yqcjl.miraltek.cfd/plis0	phishing	False	

	hex_char_count	hex_ratio
0	0	0.0
1	0	0.0
2	0	0.0
3	0	0.0
4	0	0.0

[71]:
```
url_components_df.drop(columns=['is_ip'],inplace=True)
```

[72]:
```
# Saving all the extracted features in the corresponding CSV files.

data.to_csv('data/transformed/final_raw_data.csv',index=False)
url_components_df.to_csv('data/transformed/1.url_components_data.
 ↪csv',index=False)
url_len_features_df.to_csv('data/transformed/2.component_len_features_data.
 ↪csv',index=False)
domain_features_df.to_csv('data/transformed/3.domain_features_data.
 ↪csv',index=False)
sld_features_df.to_csv('data/transformed/4.sld_features_data.csv',index=False)
char_features_df.to_csv('data/transformed/5.char_features_data.csv',index=False)
entropy_feature_df.to_csv('data/transformed/6.entropy_feature_data.
 ↪csv',index=False)
token_features_df.to_csv('data/transformed/7.token_features_data.
 ↪csv',index=False)
hex_feature_df.to_csv('data/transformed/8.hex_features_data.csv',index=False)
```