PHISHING WEBSITE DETECTION

Data Mining MSC 105,
Department of Computer Science,
University of Delhi

Jagriti Mittal
Khushi Jain
Yashi Sharma

REFERENCE 1

An efective detection approach for phishing websites using URL and HTML features

Ali Aljofey, Qingshan Jiang, Abdur Rasool, Hui Chen, Wenyin Liu , Qiang Qu & Yang Wang

OVERVIEW

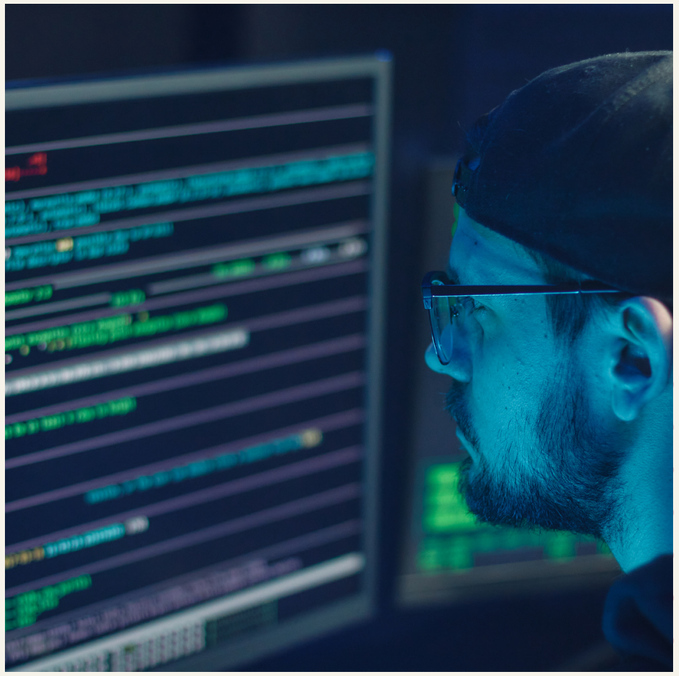
- Problem
- Introduction
- Related Work
- Dataset
- Approach
- Feature Extraction
- Classification
- Result Analysis
- Limitations
- Conclusion
- References

PROBLEM

Today’s growing phishing websites pose signifcant threats due to their extremely undetectable risk. They anticipate internet users to mistake them as genuine ones in order to reveal user information and privacy, such as login ids, pass-words, credit card numbers, etc. without notice. This paper proposes a new approach to solve the anti-phishing problem. The new features of this approach can be represented by URL character sequence without phishing prior knowledge, various hyperlink information, and textual content of the webpage.

INTRODUCTION

This paper provides an effcient solution for phishing detection that extracts the features from **website's URL and HTML source code**. Specifcally, we proposed a hybrid feature set including URL character sequence features , various hyperlink information, plaintext and noisy HTML data-based features within the HTML source code. Tese features are then used to create feature vector required for training the proposed approach by **XGBoost classifer.** Extensive experiments show that the proposed anti-phishing approach has attained competitive performance on real dataset in terms of diferent evaluation statistics.



RELATED WORK

Phishing methods are divided into two categories; expanding the user awareness to distinguish the characteristics of phishing and benign webpages, and using some extra sofware.

Software-based techniques are further categorized into :

1. **List-based detection**
2. **Machine learning-based detection.**

1

List-based phishing detection methods use either whitelist or blacklist-based technique. A blacklist contains a list of suspicious domains, URLs, and IP addresses, which are used to validate if a URL is fraudulent. Simultaneously, the whitelist is a list of legitimate domains, URLs, and IP addresses used to validate a suspected URL.

2

Machine Learning - based Detection : Due to the recent development of phishing detection methods, various machine learning-based techniques have also been employed to investigate the legality of websites. The efectiveness of these methods relies on feature collection, training data, and classifcation algorithm.

DATASET

Our dataset consists of **60,252 webpages** and their HTML source codes, wherein **27,280 ones are phishing** and **32,972 ones are benign**. We have divided the dataset into two groups where D1 is our dataset, and D2 is dataset used in existing works .The data sets were randomly split in 80:20 ratios for training and testing.

APPROACH

Our approach extracts and analyzes diferent features of suspected webpages for effective identifcation of large-scale phishing ofenses. The main contribution of this paper is the combined uses of these feature set. For improving the detection accuracy of phishing webpages, the authors have proposed eight new features. The proposed features determine the relationship between the URL of the webpage and the webpage content.

University of Delhi| 2023

FEATURE EXTRACTION

The authors have introduced eleven hyperlink fea- tures (F3–F13), two login form features (F14 and F15), character level TF-IDF features (F2), and URL character sequence features (F1).

Category	No	Name
URL based features	F1	Character sequences vectors
Textual content features	F2	TF-IDF vector N-gram chars
Hyperlink information	F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, and F13	Script_files, CSS_files, img_files, a_files, a_Null_hyperlinks, Null_hyperlinks, Total_hyperlinks, Internal_hyperlinks, External_hyperlinks, External/Internal_hyperlinks, and Error_hyperlinks
Login form information	F14 and F15	Total_forms and Suspicious_form

URL CHARACTER SEQUENCE FEATURES (FI)

- Extracting character sequence features from URL.
- Character level URL processing is a solution to the out of vocabulary problem.
- Character level sequences identify substantial information from specific groups of characters that appear together which could be a symptom of phishing.
- Preparing the character vocabulary,
- Creating a tokenizer object using Keras preprocessing package to process URLs in char level.
- Add a “UNK” token to the vocabulary after the max value of chars dictionary,
- Transforming text of URLs to sequence of tokens
- Padding the sequence of URLs to ensure equal length vectors

URL CHARACTER SEQUENCE FEATURES (F1)

Algorithm 1: Extract the URL based features

Input: URL of suspicious website U

Output: Character sequences vector $F_U = < c_1, c_2, c_3, \dots, c_{200} > \in F1$

Start

1. Initialize Tokenizer (char_level=True, oov_token='UNK'),
   Alphabet="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUV
   WXYZ0123456789,;.!?:'\"/\\|_@#$%^&*~`+-=<>()[]{}"
   char_dict = { }
2. **for** *i, char* in enumerate(*Alphabet*) **do**
3. char_dict [char] = i + 1
4. Tokenizer.word_index = char_dict
5. **end for**
   Add 'UNK' to the vocabulary
6. Tokenizer.word_index [oov_token]= max_value_of_char_dict + 1
7. U_character_sequences = Tokenizer.texts_to_sequences (U)
8. **If** len (U_character_sequences) < 200 **then**
   the remaining part will be filled as 0
9. $F_U = < c_1, c_2, c_3, \dots, 0, 0, 0, c_{200} >$
10. **else**
    the part longer will be truncated
11. $F_U = < c_1, c_2, c_3, \dots, c_{200} >$
12. **end if**
13. Return F_U

End

TEXTUAL CONTENT BASED FEATURES (F2)

- TF-IDF character level technique is applied with max features as 25,000.
- To obtain valid textual information, extra portions of the webpage are removed through regular expressions
- Including Natural Language Processing packages such as sentence segmentation, word tokenization, text lemmatization and stemming

TEXTUAL CONTENT BASED FEATURES (F2)

Algorithm 2: Extract the textual content features of a webpage

Input: A HTML document *doc*

Output: TF-IDF vector N-gram chars $F_T = < t_1, t_2, t_3, \dots, t_D > \in F2$

Start

1. P_T = getPlaintext (doc)
2. N_T =getTagAttributesValues (doc)
   (DIV, IMG, Body, Footer, a, Link, Article, Label,
   H1…H5, Template...etc.)
3. $T_1 = P_T \cup N_T$
4. Text cleaning and preprocessing
   - $T_2=\emptyset$,$T_3=\emptyset$
   - T_2 = Text_cleaner (T_1)
     (Remove punctuations symbols, numbers, spaces,
     newline, character that are not English)
   - **for** *token* in T_2 **do**
   - **if** *token* not in STOPWORDS and len (*token*) > 3 **then**
   - $T_3 = T_3 \cup$ lemmatize_stemming (*token*)
   - **end if**
   - **end for**
5. F_T =TF-IDF_Ngram_chars_Transform (T_3)
6. Return F_T

End

F3 - FI5

- Script, CSS, img, and anchor files (F3, F4, F5, and F6)

  FScript_files, FCSS_files, FImg_files, Fa_files are the numbers of Javascript, CSS, image, anchor files existing in a webpage, and FTotal is the total hyperlinks available in a webpage

- Empty hyperlinks (F7 and F8)

  Fa_null and Fnull are the numbers of anchor tags without href attribute, and null hyperlinks in a webpage

$$F3 = \begin{cases} \frac{F_{Script\_files}}{F_{Total}} & if\ F_{Total} > 0 \\ 0 & if\ F_{Total} = 0 \end{cases}$$

$$F4 = \begin{cases} \frac{F_{CSS\_files}}{F_{Total}} & if\ F_{Total} > 0 \\ 0 & if\ F_{Total} = 0 \end{cases}$$

$$F5 = \begin{cases} \frac{F_{Img\_files}}{F_{Total}} & if\ F_{Total} > 0 \\ 0 & if\ F_{Total} = 0 \end{cases}$$

$$F6 = \begin{cases} \frac{F_{a\_files}}{F_{Total}} & if\ F_{Total} > 0 \\ 0 & if\ F_{Total} = 0 \end{cases}$$

$$F7 = \begin{cases} \frac{F_{a\_null}}{F_{Total}} & if\ F_{Total} > 0 \\ 0 & if\ F_{Total} = 0 \end{cases}$$

$$F8 = \begin{cases} \frac{F_{Null}}{F_{Total}} & if\ F_{Total} > 0 \\ 0 & if\ F_{Total} = 0 \end{cases}$$

- Total hyperlinks feature (F9)

  The number of hyperlinks in a webpage by extracting the hyperlinks from an anchor, link, script, and img tags in the HTML source code.
  F9 = Total of hyperlinks present in a webpage

- Internal and external hyperlinks (F10, F11, and F12)

  F_internal, F_External, and F_Total, are the number of external, internal, and total hyperlinks in a website

$$F10 = \begin{cases} \frac{F_{Internal}}{F_{Total}} & if\ F_{Total} > 0 \\ 0 & if\ F_{Total} = 0 \end{cases}$$

$$F11 = \begin{cases} \frac{F_{External}}{F_{Total}} & if\ F_{Total} > 0 \\ 0 & if\ F_{Total} = 0 \end{cases}$$

$$F12 = \begin{cases} \frac{F_{External}}{F_{Internal}} & if\ F_{Internal} > 0 \\ 0 & if\ F_{Internal} = 0 \end{cases}$$

- Error in hyperlinks (F13)

  Phishers sometimes add some hyperlinks in the fake website which are dead or broken links. FError is the total invalid hyperlinks.

$$F13 = \begin{cases} \frac{F_{Error}}{F_{Total}} & if\ F_{Total} > 0 \\ 0 & if\ F_{Total} = 0 \end{cases}$$

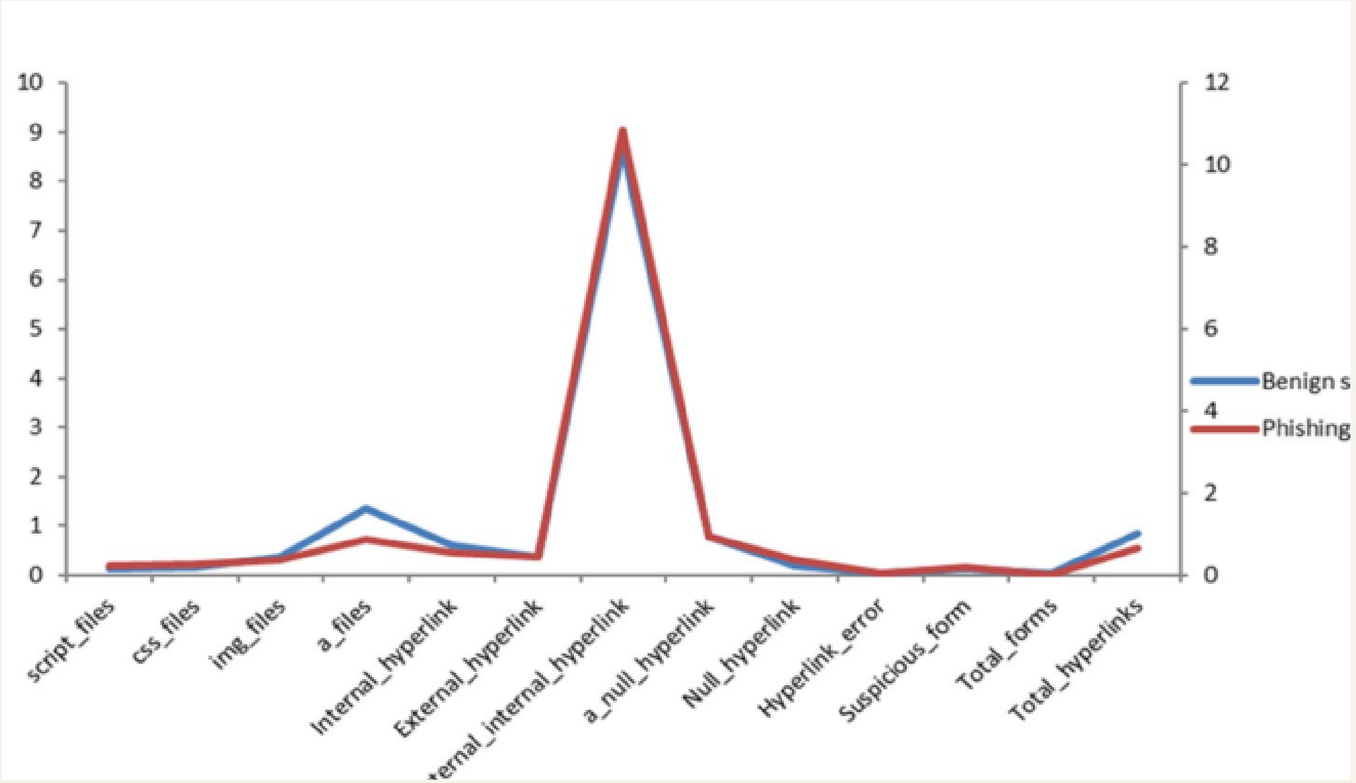
- Login form features (F14 and F15)

  FS and LTotal are the number of suspicious forms and total forms present in a webpage

$$F14 = \text{Total of forms present in a webpage}$$

$$S = \begin{cases} 1\ if\ the\ URL\ of\ action\ field\ is\ Null \\ 1\ if\ the\ URL\ of\ action\ field\ is\ not\ valid \\ 1\ if\ the\ URL\ of\ action\ filed\ is\ external\ link \\ 0\ Otherwise \end{cases}$$

$$F15 = \begin{cases} \frac{F_S}{L_{Total}} & if\ L_{Total} > 0 \\ 0 & if\ L_{Total} = 0 \end{cases}$$



The figure shows a comparison between benign and fishing hyperlink features based on the average occurrence rate per feature within each website in our dataset. From the figure, we noticed that the ratios of the external hyperlinks to the internal hyperlinks, and null hyperlinks in the phishing websites are higher than that in benign websites. Whereas, benign sites contain more anchor fles, internal hyperlinks, and total hyperlinks.

CLASSIFICATION

- To measure the effectiveness of the proposed features, we have used various machine learning classifers such as **eXtreme Gradient Boosting (XGBoost), Random Forest, Logistic Regression, Naïve Bayes** to train our proposed approach.
- The major aim of comparing different classifers is to expose the best classifer fit for our feature set.
- To apply different machine learning classifers, Scikit-learn.org package is used, and Python is employed for feature extraction.
- From the empirical results, we noticed that XGBoost outperformed other classifers. XGBoost algorithm is a type of ensemble classifers, that transform weak learners to robust ones and convenient for our proposed feature set, thus it has high performance.

WHY XGBOOST CLASSIFIER?

First

Can handle missing values in training dataset

Second

Can handle huge size of data that cannot fit into memory

Third

Faster, makes use of multiple cores on CPU.

Fourth

Better accuracy when compared to other classification algorithm

RESULT

Classifier	Textual content features	Pre (%)	Recall (%)	F-Score (%)	AUC (%)	Acc (%)
LR	TF-IDF word level	85.68	88.25	86.95	85.38	85.62
	TF-IDF N-gram level	85.23	85.42	85.33	83.93	84.05
	TF-IDF character level	84.55	87.15	85.83	84.13	84.39
	Count vectors	86.84	79.12	82.80	82.45	82.16
	Word sequences vectors	55.87	83.27	66.87	52.61	55.23
XGBoost	TF-IDF word level	88.44	88.56	88.50	87.41	87.52
	TF-IDF N-gram level	87.77	86.51	87.13	86.10	86.14
	TF-IDF character level	**89.01**	**90.58**	**89.79**	**88.65**	**88.82**
	Word sequences vectors	82.66	85.87	84.23	82.24	82.55
	Count vectors	88.26	87.75	88.00	86.95	87.02
	Character sequences vectors	81.47	87.81	84.52	82.05	82.54
RF	TF-IDF word level	85.94	92.67	89.18	87.34	87.80
	TF-IDF N-gram level	86.77	89.57	88.14	86.68	86.93
	TF-IDF character level	85.44	92.81	88.97	87.02	87.51
	Count vectors	85.81	93.08	89.30	87.41	87.90
	Word sequences vectors	81.56	90.71	85.89	83.19	83.83
	Character sequences vectors	79.51	93.91	86.11	82.60	83.56
NB	TF-IDF word level	84.50	79.12	81.72	80.95	80.79
	TF-IDF N-gram level	82.45	71.16	76.39	76.59	76.13
	TF-IDF character level	76.45	81.89	79.08	75.98	76.49
	Count vectors	82.62	71.63	76.74	76.88	76.43
	Word sequences vectors	62.89	42.66	50.83	56.39	55.22

From the experimental results, it is noticed that **TF-IDF character level features** outperformed other features with signifcant accuracy, precision, F-Score, Recall, and AUC using **XGBoost classifer**. Hence, we implemented TF-IDF character level technique to generate text features (F2) of the webpage.

Classifier	Features	Pre (%)	Recall (%)	F-Score (%)	AUC (%)	ACC (%)
LR	F_{URL}	74.67	67.92	71.13	74.25	74.79
	F_{HTML}	83.50	81.98	82.74	84.16	84.35
	$F_{URL+ HTML}$	77.71	68.74	72.95	76.06	76.68
NB	F_{URL}	81.41	22.09	34.76	58.92	62.06
	F_{HTML}	65.67	87.57	75.06	74.49	73.38
	$F_{HTM + URL}$	86.99	62.15	72.51	77.16	78.44
Ensemble	F_{URL}	98.42	92.05	95.13	95.40	95.69
	F_{HTML}	90.22	82.01	85.92	87.25	87.70
	$F_{URL+ HTML}$	93.89	87.85	90.77	91.52	91.83
RF	F_{URL}	98.54	92.14	95.23	95.49	95.78
	F_{HTML}	90.77	81.98	86.16	87.48	87.95
	$F_{URL + HTML}$	93.81	86.79	90.16	90.98	91.34
XGBoost	F_{URL}	99.58	92.27	95.79	95.97	96.29
	F_{HTML}	88.21	87.68	87.94	88.90	89.01
	$F_{URL + HTML}$	**98.28**	**94.56**	**96.38**	**96.58**	**96.76**

It is observed that URL and HTML features are valuable in phishing detection. However, one type of feature is not suitable to identify all kinds of phishing webpages and does not result in high accuracy. Thus, we have combined all features to get more comprehensive features.

LIMITATION

First Problem

Web-page Language dependent(work only on English language)

Second Problem

Unable to identify textual content hidden in embedded objects

Third Problem

Not sufficiently capable of detecting attached malware

Fourth Problem

Large training time (due to high dimensional vector)

CONCLUSION

The given approach detect phising websites based on URl and HTML features. XGBoost Classifier with all type of feature give best result with accuracy of 96.76%.
As blockchain technology technology emerges, it becomes a perfect target for phising attacks.

REFERENCE 2

A new hybrid ensemble feature selection framework for machine learning-based phishing detection system

Kang Leng Chiew , Choon Lin Tan , KokSheik Wong , Kelvin S.C. Yong, Wei King Tiong

OVERVIEW

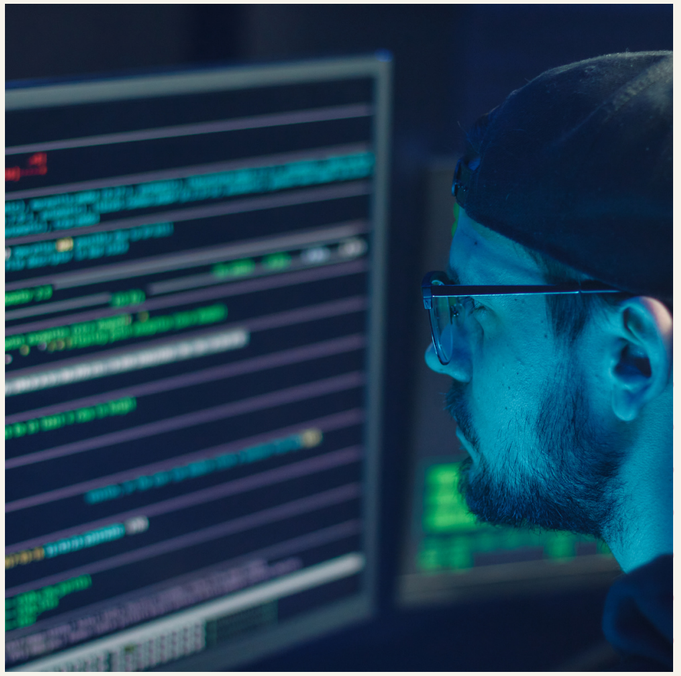
- Problem
- Introduction
- Motivation
- Related Work
- Dataset
- Approach
- Feature Extraction
- Classification
- Result Analysis
- Practical Implications
- Conclusion
- References

PROBLEM

Phishing attacks are a major problem and current anti-phishing solutions are not effective. Feature selection is crucial for machine learning-based phishing detection systems to achieve high accuracy. However, existing feature selection techniques are limited in terms of efficiency and effectiveness.

INTRODUCTION

This research paper proposes a new feature selection framework for machine learning-based phishing detection systems called the Hybrid Ensemble Feature Selection (HEFS). The framework consists of two phases: the first phase utilizes the Cumulative Distribution Function gradient (CDF-g) algorithm to generate primary feature subsets, while the second phase derives baseline features from the secondary feature subsets. Experimental results show that HEFS, when integrated with the Random Forest classifier, performs best, identifying baseline features that correctly distinguish 94.6% of phishing and legitimate websites using only 20.8% of the original features.



MOTIVATION

- There is no existing framework to reliably determine the cut-off ranks (i.e., cut-off point) for the features ranked by filter measures.
- Effectiveness of the existing feature selection technique are dataset dependent (i.e., not robust)
- The question on “which classifier is the most predictive one” remains as a huge subject of debate

RELATED WORK

1. Toolan and Carthy [28] studied the selection of features for spam and phishing detection using 40 techniques. They ranked features across three datasets using Information Gain (IG) and identified 9 consistent features. However, the study is incomplete as it only assessed the performance of a single filter measure, IG, from a general perspective.
2. Khonji et al.'s study compared feature selection methods for phishing email classification, finding that the wrapper method, combined with the best forward searching method, outperforms IG and Relief-F feature subsets, while CFS performs worst. The Random Forest classifier is most predictive, but the wrapper method is computationally expensive.
3. Basnet et al.'s study evaluated two common feature selection techniques: CFS and wrapper method. They tested genetic algorithm and greedy forward selection on webpage and search engine features. Results showed wrapper method achieved higher detection accuracies than CFS, but it was computationally more intensive, making it less feasible for feature selection applications.

4. Qabajeh and Thabtah's study on phishing email detection found that filter measures can reduce feature dimensionality without compromising classification accuracy. They used a cut-off rank between the 20th and 21st feature in IG and Chi-Square ranked features, resulting in a stable detection accuracy. Further tests showed a 0.28% decrease in average accuracy compared to the full feature set.
5. Thabtah and Abdelhamid [27] proposed a threshold-based rule set for identifying cut-off ranks for features ranked by IG and Chi-Square for phishing website detection. The rule sets are not robust and may fail to identify appropriate cut-off ranks for uniform filter measure values. This suggests a need for a systematic approach to identify the optimal cut-off rank in filter measures-based feature selection studies.

DATASET

Authors selected **5000 phishing webpages** based on URLs from PhishTank2 and OpenPhish3, and another **5000 legitimate webpages** based on URLs from Alexa4 and the Common Crawl5 archive.The downloaded dataset were further processed to remove defective webpages which failed to load or “Error 404” pages in both the phishing and legitimate dataset. Duplicate instances of webpages are also removed. After filtering the samples, feature extraction is performed.

APPROACH

The researchers utilized filter measures, such as Information Gain, Relief-F, and Symmetrical Uncertainty, to rank features and generate primary feature subsets. The CDF-g algorithm was then applied to identify the optimal cut-off rank of features. The hybrid ensemble strategy combined data perturbation and function perturbation techniques to produce secondary feature subsets and the final baseline feature set.

FEATURE SELECTION

1. The hybrid ensemble strategy

There are two major types of ensemble techniques, namely **data perturbation** and **function perturbation**. Data perturbation applies the same feature selection technique on multiple subsets of a dataset, while function perturbation applies multiple feature selection techniques on the same set of data. Combining both data perturbation and function perturbation results in a hybrid perturbation ensemble technique

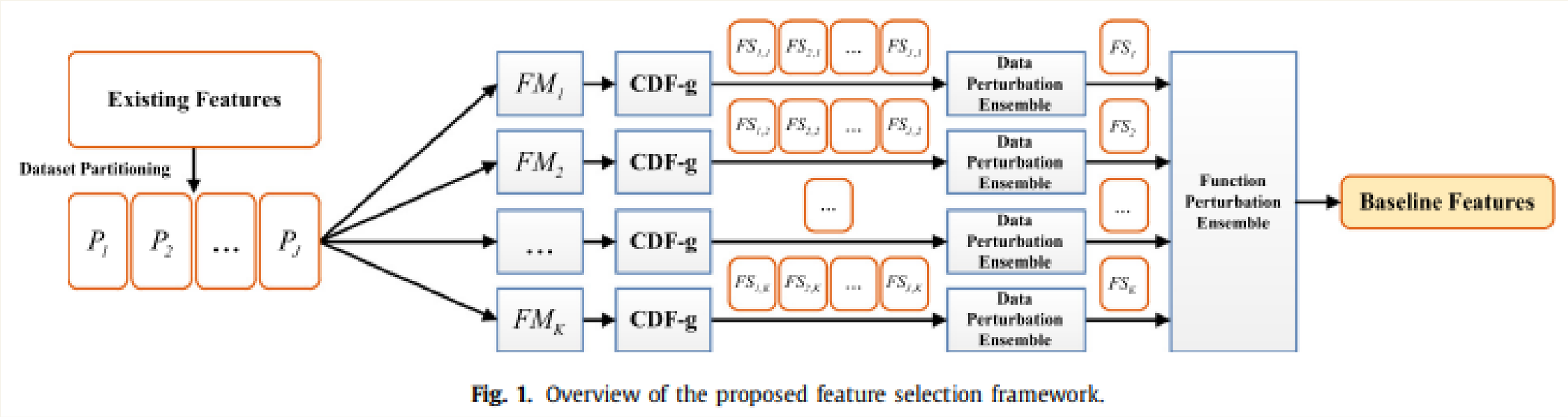


Fig. 1. Overview of the proposed feature selection framework.

A list of feature cut-off ranks {τ1,k, τ2,k,..., τJ,k} is created respectively for each dataset partition by using filter measure FMk. The mean of the feature cut-off ranks, i.e., $\tau^{-}k$, is obtained using Eq 1 and referenced on {P1, P2,..., PJ} to generate an array of primary feature subsets {FS1,k, FS2,k,..., FSJ,k}

The data perturbation ensemble component will apply an intersection operation, as shown in Eq. (2), on the array of primary feature subsets to produce a secondary feature subset FSk

$$\bar{\tau}_k = \frac{1}{J}\sum_{j=1}^{J}\tau_{j,k}\,. \tag{1}$$

$$FS_k = \bigcap_{j=1}^{J} FS_{j,k}\,. \tag{2}$$

The function perturbation cycle is a data perturbation process that runs over K filter measures, producing secondary feature subsets. These subsets are fed into the ensemble component, which aggregates inputs to obtain the best baseline feature set, reducing overfitting.

$$\text{Baseline feature set} := \bigcup_{k=1}^{K} FS_k\,. \tag{3}$$

2. CDF-g algorithm — An automatic feature cut-off rank identifier

The cut-off rank is defined as the specific threshold rank in an ordered list of filter measure values, where any features located beyond the cut-off rank is considered irrelevant and therefore discarded.

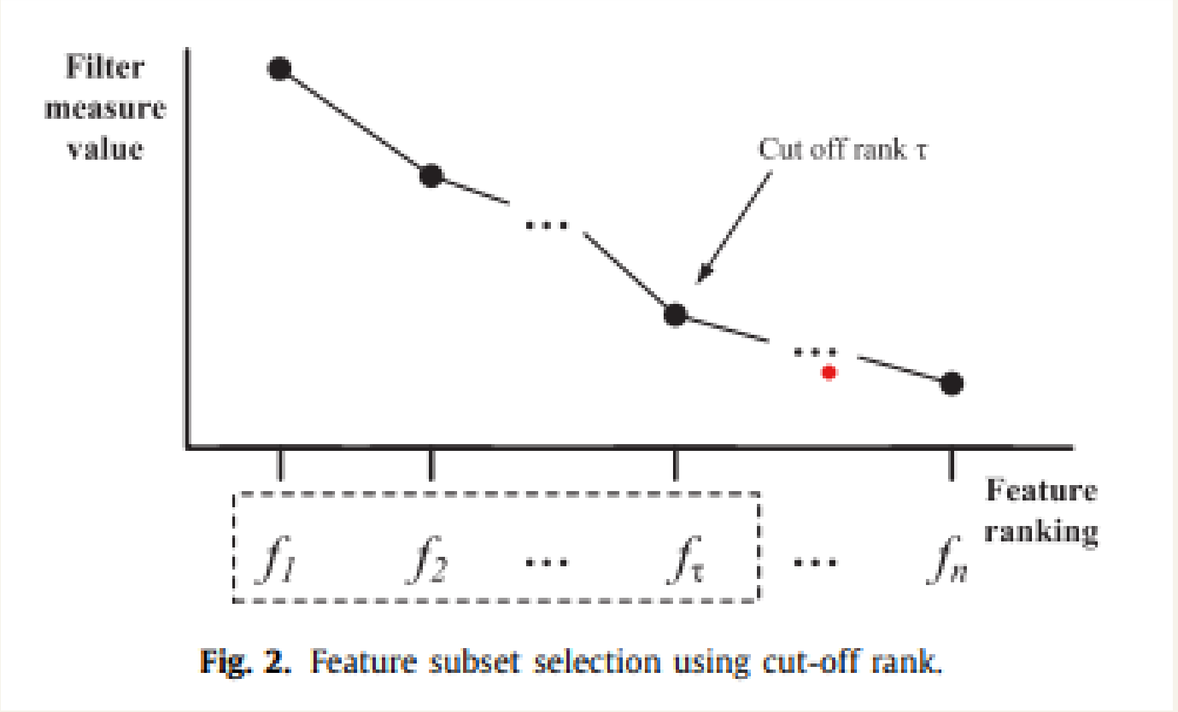


Fig. 2. Feature subset selection using cut-off rank.

The CDF gradient approach, known as CDF-g, analyzes gradient changes in the curve to identify plateau regions along filter measure values, indicating separation between feature subsets with significant predictive power. This method helps pinpoint the cut-off rank of top-ranked features.

To compute the gradient at any point Ri on the CDF curve, we used the central differences for the interior points, and one-side (forward or backwards) differences for the boundaries, as shown below:

$$\text{Gradient, } G(R_i) = \begin{cases} \dfrac{F_X(t_{i+1}) - F_X(t_i)}{h}, & \text{if } i = 1; \\ \dfrac{F_X(t_{i+1}) - F_X(t_{i-1})}{2h}, & \text{if } 1 < i < n; \\ \dfrac{F_X(t_i) - F_X(t_{i-1})}{h}, & \text{if } i = n; \end{cases}$$

3.Features preparation

Algorithm 1 Feature cut-off rank identification using CDF-g.

Input: V, b ▷ V = Feature vector file, b = Bin size
Output: τ ▷ τ = Feature cut-off rank

```
Begin
FM_values = ComputeFilterMeasure(V)
FM_values = Normalise(FM_values)
FM_values = Sort(FM_values)
CDF_values = ComputeCDF(FM_values, b)
for all c ∈ CDF_values do
    if c > 0.5 then
        CDF_gradient = ComputeGradient(c)
        if CDF_gradient = 0.0 then
            τ = MapToFeatureRanking(c)
            break
        end if
    end if
end for
End
```

RESULT

Evaluation I — Performance of various classifiers on different feature subsets derived from filter measures rankings

Comparing the stable states of accuracy among the classifiers, Random Forest appears to consistently outperform all remaining classifiers by scoring above 95% accuracy.

Evaluation II — Fitness of filter measure with selected classifier

Information Gain, Chi-Square, and Symmetrical Uncertainty are the top-3 fittest filter measures to be used in combination with Random Forest classifier

Evaluation III — Performance of baseline features

The final baseline feature set contains 10 features

FrequentDomainNameMismatch	*PctNullSelfRedirectHyperlinks*
PctExtNullSelfRedirectHyperlinks	*PctExtResourceUrlsRT*
NumNumericChars	*ExtMetaScriptLinkRT*
PctExtHyperlinks	*SubmitInfoToEmail*
NumDash	*NumSensitiveWords*

Evaluation IV — Runtime analysis comparison

Baseline feature set produced using HEFS has the lowest testing time

Average runtime per sample for classification phase.

Feature selection technique	Number of features	Classification time (ms)
Chi-Square	37	0.0670
Information Gain	34	0.0673
Symmetrical Uncertainty	46	0.0783
HEFS	10	0.0523

Evaluation V — Benchmarking on other machine learning dataset

The promising results in this evaluation also suggest that the proposed HEFS is robust and thus, can flexibly adapt to different datasets.

Performance benchmarking between the proposed HEFS framework and FACA [11].

Classifier	Feature set	Number of features	Accuracy (%)
FACA	Full	30	92.40
Random Forest	Full	30	94.27
Random Forest	Baseline (derived from HEFS)	5	93.22

PRACTICAL IMPLICATIONS

First

Enhancing the detection accuracy and computational efficiency

Second

Providing a fully automatic, flexible and robust feature selection framework

Third

Prescribing a set of baseline features for future benchmarking

Fourth

Accelerating the development of powerful and compact phishing detection system

CONCLUSION

The HEFS framework is a highly desirable and practical feature selection technique for machine learning-based phishing detection systems. It outperforms existing techniques in terms of accuracy and efficiency, and the derived baseline features are effective in distinguishing between phishing and legitimate websites. The framework can be universally applied to different datasets and provides a benchmark for evaluating future phishing detection techniques.

OTHER REFERENCES

A Hybrid Model to Detect Phishing-Sites using Supervised Learning Algorithms
M. Amaad Ul Haq Tahir, Sohail Asghar, Ayesha Zafar, Saira Gillani

Detection Of Phishing Websites Using Data Mining
Aniket Kote, Sanket Kharche, Pravin Aware, Abhishek pangavhane

Detecting Phishing Websites using Data Mining
Ashna Antony

THANK YOU

University of Delhi | 2023