Int. J. of Data Mining, Modelling, and Management , Vol. 12, No. 3, 255-276

Phish Webpage Classification Using Hybrid Algorithm of Machine Learning and Statistical Induction Ratios

Hiba Zuhair[1, a]
[1]Department of Systems Engineering,
College of Information Engineering,
Al-Nahrain University,
Baghdad, Iraq.
E-mail1: hiba.zuhir@coie-nahrain.edu.iq;
E-mail2: hiba.zuhair.pcs2013@gmail.com.
[a] Corresponding author.

Ali Selamat[2,3,4]
[2] School of Computing, Faculty of Engineering, UTM & Media and Games Center of Excellence (MagicX), Universiti Teknologi Malaysia (UTM), Johor, Malaysia.
[3] Malaysia Japan International Institute of Technology (MJIIT), Universiti Teknologi
Malaysia, Jalan Sultan Yahya Petra, Kuala Lumpur, Malaysia
[4] Center for Basic and Applied Research, Faculty of Informatics and Management, University of Hradec Kralove, Rokitanskeho 62, 500 03 Hradec Kralove, Czech Republic.
E-mail:aselamat@utm.my.

Abstract: Although the conventional machine learning-based anti-phishing techniques outperform their competitors in phishing detection, they are still targeted by zero-hour phish webpages due to their constraints of phishing induction. Therefore, phishing induction must be boosted up with the extraction of new features, the selection of robust subsets of decisive features, the active learning of classifiers on a big webpage stream. In this paper, we propose a hybrid feature-based classification algorithm (HFBC) for decisive phish webpage classification. HFBC hybridizes two statistical criteria Optimized Feature Occurrence (OFC) and Phishing Induction Ratio (PIR) with the induction settings of the most salient machine learning algorithms, Naïve Bays and Decision Tree. Additionally, we propose two constituent algorithms of features extraction and features selection for holistic phish webpage characterization. The superiority of our proposed approach is justified and proven throughout chronological, real-time, and comparative analyses against existing machines learning-based anti-phishing techniques.

Reference to this paper should be made as follows: Zuhair, H., and Selamat, A., 'Phish Webpage Classification Using Hybrid Algorithm of Machine Learning and Statistical Induction Ratios', *Int. J. Data Mining, Modelling, and Management*.

Biographical notes:
Hiba Zuhair currently works as senior lecturer and researcher in Dept. of Systems Engineering at College of Information Engineering, Al-Nahrain University, Baghdad, Iraq. Recently, she is member of the Journal Editorial Boards: MALTESAS Multi-disciplinary Research Journal, Journal of Software Engineering and Intelligent Systems, and International Journal of Artificial Intelligence Research. Prior to this, she is awarded her PhD from Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia with a high distinction as the "Best Postgraduate Student Award". Before, she received her MSc. and BSc. with a high distinction from Dept. of Computer Science at College of Science, Al-Nahrain University, Baghdad, Iraq. Her recent research interests and publications are of cyber-crimes, intrusion detection systems, ethical hacking, digital forensics, big data science and analytics, machine learning and deep learning as well as other fields in computer networks and security.

Ali Selamat is currently a Dean Office of Malaysia Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia (UTM), Kuala Lumpur, Malaysia. He is also a visiting professor at University of Hradec Kralove, Czech Republic - EU. Furthermore, he is also a professor in Software Engineering Department at Faculty of Computing, UTM, Johor, Malaysia. He is nominated as the Chair of IEEE Computer Society Malaysia since 2014. He is the Editor-in-Chief of International Journal of Digital Enterprise Technology, Inderscience Publications. In addition, he is a Co-Editor in Chief of International Journal of Software Engineering and Technology (IJSET), and a member of the Journal Editorial Boards: Knowledge Based Systems, Elsevier, International Journal of Information and Database Systems, Inderscience Publications, and Vietnam Journal of Computer Science, Springer Verlag. His research interests and publications include software engineering, software process improvement, software agents, web engineering, information retrievals, pattern recognition, genetic algorithms, neural networks and soft computing, computational collective intelligence.

This paper is a revised and expanded version of a paper entitled [Phishing Hybrid Feature-Based Classifier by Using Recursive Features Subset Selection and Machine Learning Algorithms] presented at [*The 3rd International Conference of Reliable Information and Communication Technology 2018, Putrajaya Kuala Lumpur, Malaysia, 23-24 July 2018*].

Keywords: phish webpage, machine learning, optimized feature occurrence**,** phishing induction ratio, hybrid feature-based classifier.

1 Introduction

Although the Web provides a huge communication channel and many services to both users and enterprises, it causes digital identity theft and monetary loss annually due to phishers' activities. Phishers usually evolve their variants of phish webpages that impersonate legitimate webpages to deceive users [1, 2]. To thwart phishers' activities and mitigate their consequences on both web security and economy, researchers develop different anti-phishing techniques. Amongst them, are phishing machine learning-based classifiers [1-3] that assisted by various features and baseline machine learning algorithms looking forward effective and efficient phishing classification [3]. Constructed classifiers rely on various features to promote their discriminating power by characterizing phish exploits. Also, they apply algorithms of Naïve Bayes (*NB*), Support Vector Machine (*SVM*), Decision Tree (*DT*), and Logic Regression (*LR*) etc. to classify phish webpages decisively. Therefore, machine learning-based anti-phishing techniques outperform their competitors [2-4]. However, they are evaded by phishers who evolving zero-hour phish webpages continually due to their deficiency of inductive factors. Indeed, lack of potential inductive factors leads to partial phishing characterization and inefficient phishing classification in realistic mode [3-5].

In this context, this paper proposes a *Hybrid Feature-Based Classifier (HFBC)* which is a hybrid of features subset selection algorithm and two of the most salient machine learning algorithms with statistical phishing induction criteria. The outperformance of *HFBC* is validated throughout three analyses: a chronological analysis across three benchmarking datasets, 30 days interval of real-time analysis on an evolving webpage stream, and a comparative analysis against the state-of-the-art machine learning algorithms. Next sections give the bird's eye on *HFBC* as it follows: Section 2 surveys and synthesizes the related works to address what facets need to boost. Whereas, Section 3 depicted the workflow of *HFBC* including its constituent algorithms and supportive induction criteria. Then, Sections 4 exhibits *HFBC* performance validation and discusses the outcomes. In Section 5, conclusions are drawn and future insights are inferred.

2 Related Work

Among the prominent phish webpage, classifiers were CANTINA and its upgraded version CANTINA+ that developed as ensemble feature-based classifiers by the researchers at Carnegie Mellon [5]. Both learned multiple machine learning algorithms including Naïve Bayes (NB), Support Vector Machine (SVM), and Logic Regression (LR) etc. Around 15 features were derived from the webpage content and URL to accurately classify phishing on redirecting webpage, login forms, and webpages of English language. Although CANTINA+ achieved a True Positive Rate (TPR) of (92%) and a False Positive Rate (FPR) of (1.4%). However, it encountered a trade-off in classifying phishing on other webpage exploits involved in up-to-date webpage streams due to the use of textural features that were limited to English text. Then, a single feature-based classifier of Support Vector Machine (SVM) was devoted to tackling phishing on login form webpages [6, 7]. The developed SVM classifier could identify 17 phishing features in login forms' contents. It achieved a rationale performance of (99.6%) as TPR and (0.42%) as the FPR. However, it was computationally intensive and time-consuming due to its pre-defined whitelist of the topmost legitimate webpages that was integrated as an external resource. Meanwhile, the authors in [8] attempted to identify phishing deceptions on Chinese e-business webpages via their single feature-based classifier. For their motive, the best-ranked subset of features was obtained from 15 URL features by using Chi-Squared (χ^2) for further learning with four machine learning algorithms including Sequential Minimum Optimization (SMO), Logic Regression (LR), Naïve Bayes (NB), and Random Forests (RF). Experimentally, their developed classifier performed (95.83%) of detection accuracy on a life-like Chinese webpage flow. However, it was applicable to Chinese e-business webpages rather than other webpage exploits due to the exclusive phishing features and datasets that it used.

Later, other classifiers were developed such as that adopted in [9] to learn 17 URL features with an ensemble platform of machine learning algorithms including Support Vector Machine (SVM), and Random Forest (RF). The developed classifier achieved (94.91%) and (1.44%) of classification accuracy and faults respectively by using three different mechanisms of features selection such as Information Gain (IG), Correlation-Based Feature Selection (*CFS*) and Chi-Squared (χ^2). Although, it could learn relatively big datasets; it was inefficient to handle data imbalanced issue. On the other hand, a Neural Network (*k-NN*) feature-based classifier was presented in [10-12] to detect different phish exploit on a big webpage stream (96,018 webpages) with an accuracy of (96.71%). Due to the substantial rate of misclassification, the classifier was optimized into an ensemble design by integrating four machine learning algorithms including Support Vector Machine (*SVM*), Random Forest (*RF*), *C4.5*, and *JRip*. Thus, it learned actively with 212 typical features to increase the detection accuracy and decrease the detection faults.

As time progresses, a case based reasoning classifier *CBR-PDS* with K-NN machine learning algorithm was proposed in [13]. Experimentally, *CBR-PDS* could predict phish webpages over scalable and different datasets with a range of accuracy rates from 95.62% up to 98.07%. *CBR-PDS* pursued phishing detection in both offline and online modes to easily predict webpages. However, it deployed typical URL features to distinguish phish webpages on small sets of data. The set of typical URL features was not distinctive and decisive enough to adapt the advanced exploits and the new features that encompassed in zero-hour phish webpages during real-time practice. Later, in [14] a *Cognitive Framework* using typical domain knowledge features and semantic text as well as layout features, was adopted to detect phish webpages. *Cognitive Framework* deployed a deep learning algorithm of Bidirectional LSTM RNN for phishing classification along with *Convolution Networks* (*CNN*) for features extraction actively. However, not all necessary performance validations were involved, and particular phishing cases on business websites were studied to demonstrate its effectiveness and its efficiency as a web safe service through browser extensions or API calls.

The aforesaid achievements encountered the problems of: (i) using typical and uninformative features rather than new and distinctive features for phishing characterization; (ii) the sets of features were not decisive enough because they were minimally relevant to phishing classes and maximally redundant in phishing exploitations [1-4, 15-19]; (iii) heavyweight webpage crawling and processing along with an imprecise phishing induction against variable datasets [1-4, 15, 16]; (iv) the learning datasets were unreflective to the web data that of scalable size, variable webpage exploits, and imbalanced phish and/or not-phish class [3, 4, 17, 18], (v) inactive learning strategy pursued by the developed classifiers so that their default phishing induction settings were inaccurate and inadaptable to classify zero-hour phish webpages at any given time [4, 3,19-20], (vi) almost developed classifiers produced significant cost of operating errors and misclassification in real-time practice due to their divergent induction settings and pruning parameters, and then (vii) the use of limited number and typical features rather than the new features that might not be crafted in zero-hour phish webpages more and this in turn yields a partial phishing characterization with high rates of misclassifications.

3 Materials and Method

This section exhibits the constituent steps of the proposed approach "*Hybrid Feature-Based Classifier* (*HFBC*)" alongside their relevant computational algorithms, experimental datasets, performance measures, and experimental design, as it follows:

3.1 Hybrid Feature-Based Classifier

The proposed approach "*Hybrid Feature-Based Classifier* (*HFBC*)" was pursued through three steps: *features extraction*, *features selection*, and *phishing classification*. In *features extraction step*, three different sub-vectors of features including 24 embedded objects features *(*F_E^{24}*)*, 24 cross Site Scripting (*XSS*) features (F_X^{24}), and ten language independent features (F_L^{10}); were extracted from three parts of the webpage such as *Hypertext Markup Language* (*HTML*) part, *JavaScript* part, and *URL* part [15, 16] by implementing *Feature Extraction Algorithm* (*FEA*)**.** Extracted features were formulated into a single feature vector$(F_v = \{\bigcup_1^{58}(F_E^{24}, F_X^{24}, F_L^{10}\ )\})$, as illustrated in **Figure 1**. Each phish or legitimate webpage in the training webpage stream was characterized into a feature vector along with its actual class $C_m$as either a phish or a legitimate (see **Figure 1)**. All extracted feature vectors were formulated into a multidimensional matrix named as feature space (F_{space}). Thus, $F_{space} = \{F_{v1}, F_{v2}, \dots, F_{vj}, \dots, F_{vm}\}$) in which each row represented a feature vector (F_{vj}) with its corresponding values (v_j) and its relative class label (C_j) where ($j = 1,2\ ,3\ , \dots , m$) and ($C_j \in \{-1, 0, 1\}$). Whenever, ($C_j = -1$) then (F_{vj}) was a phish; or ($C_j = 1$) then (F_{vj}) was a legitimate. Otherwise, ($C_j = 0$) denoting that (F_{vj}) was a suspicious webpage (neither valid phish nor valid legitimate).

In *features subset selection step*, the *Recursive Features Subset Selection Algorithm* (*RFSSA*) including its supportive sub-algorithm *Features Selection Algorithm* (*FSA*); was proposed to select the most distinctive features and the best subset of decisive features, respectively. *FSA* pursued *mRMR* which could boost both the mutual information of the targeting class and the mutual dependencies among features in the same set of features. Then, the output set of distinctive features were fed to *RFSSA* which split it into *N* subsets to prioritize the best features subset according to validating ratios of goodness (*Good Ratio*), stability (*Stab Ratio*), and similarity (*SimRatio*) [4, 17 and 18] as per Equations (2), (3), and (4) [4, 17 and 18]. The *Good Ratio* and *Stab Ratio* demonstrated subset's distinction and robustness versus the evolving data flow over a period of time; whereas, *SimRatio* proved the subset's potentiality amongst its competitors versus the evolving data [4, 17 and 18].

$$max\ \Phi(D, R), \Phi = D(S, c) - R(S). \qquad (1)$$

Where, D is the set of features S having the maximal dependency to the target class c, and R is the set of features having the highest dependency among other features in S.

$$GoodRatio(S_i) = \frac{1}{Y}\sum_{i=1}^{Y}\frac{N_i^{tp}}{N_i} \qquad (2)$$

Where Y, N_i^{tp} and N_i are the number of classes in the dataset, the number of true positive of each class and the total number of instances for class i respectively.

$$StabRatio(S) = \sum_{f_i \in X} \frac{F_{f_i}}{N} \times \frac{F_{f_i} - 1}{|D| - 1} \tag{3}$$

Where $f_i \in X$ and $\frac{F_{f_i}}{N}$ are all features in a collection dataset S and the relative frequency of each feature in a subset. If all subsets are identical then *Stab(S)* closes to 1; otherwise it closes to 0.

$$SimRatio(t_1, t_2) = 1 - \frac{1}{2} \sum \left| \frac{F_{f_i}^{t_1}}{N^{t_1}} - \frac{F_{f_i}^{t_2}}{N^{t_2}} \right| \tag{4}$$

Where $F_{f_i}^{t_1}$ and $F_{f_i}^{t_2}$ denoting the number of frequencies of feature f_i in two candidate feature selection methods t_1 and t_2 respectively. Similarity takes values within [0,1].

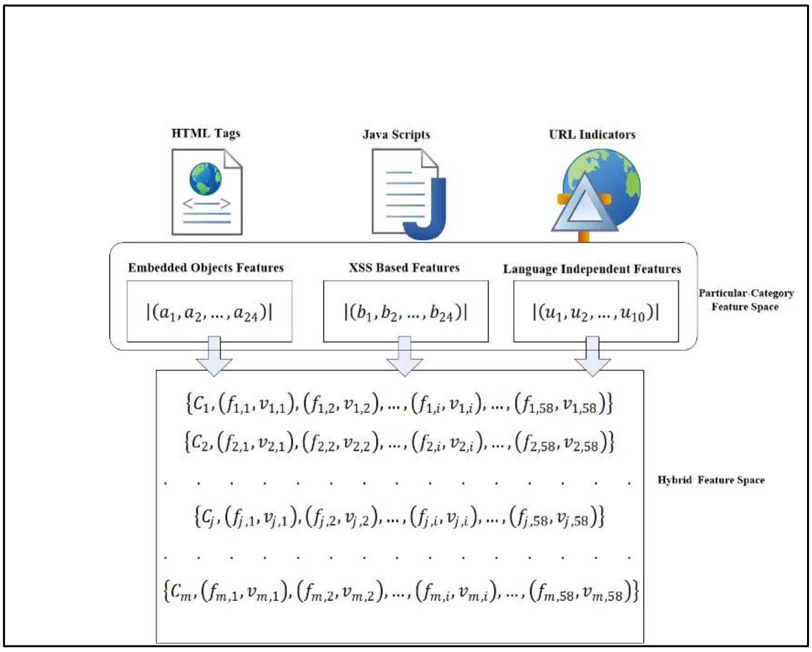

HTML Tags
Java Scripts
URL Indicators
Embedded Objects Features
$|(a_1, a_2, \dots, a_{24})|$
XSS Based Features
$|(b_1, b_2, \dots, b_{24})|$
Language Independent Features
$|(u_1, u_2, \dots, u_{10})|$
Particular-Category Feature Space
$\{C_1, (f_{1,1}, v_{1,1}), (f_{1,2}, v_{1,2}), \dots, (f_{1,i}, v_{1,i}), \dots, (f_{1,58}, v_{1,58})\}$
$\{C_2, (f_{2,1}, v_{2,1}), (f_{2,2}, v_{2,2}), \dots, (f_{2,i}, v_{2,i}), \dots, (f_{2,58}, v_{2,58})\}$
$\{C_j, (f_{j,1}, v_{j,1}), (f_{j,2}, v_{j,2}), \dots, (f_{j,i}, v_{j,i}), \dots, (f_{j,58}, v_{j,58})\}$
$\{C_m, (f_{m,1}, v_{m,1}), (f_{m,2}, v_{m,2}), \dots, (f_{m,i}, v_{m,i}), \dots, (f_{m,58}, v_{m,58})\}$
Hybrid Feature Space

Figure 1. Webpage analysis and features extraction through *FEA*

```
FEA // Features Extraction Algorithm
Input: W, N // W is the input dataset and N is the number of its instances
Output: Feature Space
Begin
1.    Define $F_v$// feature vector, and  Tree   // DOM Tree
      Set Feature Space as $m \times n$ matrix
2.    For I=1 To N  {
      a.        Construct a graph Tree for the input webpage W(I)
```

```
        b.    Remove any uninformative nodes from Tree
        c.    Collapse Tree
        d.    Extractor ($\boldsymbol{Tree}$, $\boldsymbol{F_v}$)
        e.    Store ($I$, $\boldsymbol{F_v}$, Feature Space) }
3.      Feature Space=Mapping (Feature Space) //adjust heterogeneous
        feature values
End of FEA

Extractor
Input   : Tree // DOM tree
Output: $\boldsymbol{F_v}$ // feature vector
Begin
i.    Set     Raw as a raw of Tree
              $F_L$ as vector of URL features
              $F_X$ as vector of XSS-based or JavaScript features
              $F_E$ as vector of HTML features or embedded objects
ii.   If (Raw is text leaf) OR(Raw is the last child)  Skip
      Else If (Raw = = head)       Parser("LINK", $F_L$)
          Else If (Raw= = BODY) { Parser(Raw, $F_X$); Parser(Raw, $F_E$);
      Parser(Raw, $F_L$); }
iii.  $\boldsymbol{F_v} = (F_L \cup F_X \cup F_E)$
End of Extractor

Parser
Input   : Raw // A raw of Tree
Output: $\boldsymbol{F_{sub}}$ // a sub-vector of parsed features
Begin
i.          Set      Node as a node in Raw
                     Count_Start_Node as counter Node 's opening tag
                     Count_End_Node as counter Node 's closing tag
                     StartPos as pointer to the position of Node 's
                     opening tag
                     EndPos as pointer to the position of Node 's closing
                     tag
ii.   EndPose=Count_End_Node
iii.  While (StartPos≠EndPose) Do
      {Content=extract string between StartPos and EndPos
      Count_Start_Node=frequency of open given Node
      Count_ End_Node= frequency of closing given Node
       $\boldsymbol{F_{sub}}$ [StartPos] =Content }
End of Parser
```

```
FSA // Features Selection Algorithm
Input   : FSet// the original set of features such that $FSet = \{FSet_i\}_{i \in |FSet|}$
Output : OutputSet // the set of most relevant and least redundant features
Define     Red_Rev       // the set of maximal relevant and minimal redundant
           OutputSet     // the set of output features such that OutputSet={}
Begin
1     Compute maximal relevancy and minimal redundancy in $\boldsymbol{FSet}$
```

```
        For i=1 To |FSet|  Do the following {
        a.      Compute relevance for FSet
        b.      Compute redundancy for FSet
        } // end of for loop
2       Initialize k
        While (FSet ≠{}) AND (k<=|Red_Rel|) do the following {
        Compute Φ =max(Red_Rel) for FSet as per Equation (1) }
3       Exclude Φ from FSet
        Add Φ to OutputSet
        Project the remaining of FSet onto the feature space to return OutputSet
End of FSA
```

```
RFSSA // Recursive Feature Subsets Selection Algorithm
Input   : FSet // the original set of features
Output: SubSet_Best // the best generated subset of features
Define    SubSet={}     // list of chosen subsets
          RatioSet= {} // list of goodness scores
          SubSet_i      // a generated subset of size $N^{SubSet}$
          N             // the number of generated subsets
          OutputSet // // the set of most relevant and least redundant features
Begin
1       FSA (FSet, OutputSet)
2        Generate N SubSet from OutputSet such that each subset has its size $N^{SubSet}$
        REPEAT  Compute SimRatio  across N SubSet as per Equation (4)
         UNTIL (SimRatio>= ThreshValue)AND(OutputSet ={})
        For i=1 to N {
        a.    Compute $Goodness_{SubSet_i}$  as per Equation (2)
        b.    Compute $Stability_{SubSet_i}$  as per Equation (3)
        c.    Set $RatioSet_{SubSet_i}=\frac{Goodness_{SubSet_i}}{Stability_{SubSet_i}}$
        } // end of for loop
        Sort RatioSet in descending order
        Exclude $SubSet_i$ of max($RatioSet_{SubSet_i}$) from SubSet
3       Repeat 2 Until (OutputSet ={})
4       SubSet_Best ← $SubSet_i$
End of RFSSA
```

In *phishing classification step*, the training webpage stream could be learned actively based on the best selected features subset via the proposed *hybrid features-based classifier* (*HFBC*)**.** *HFBC* leveraged the cutting back decision tree of *DT* to split the training feature space into sub-training spaces that would be pruned by *NB's* induction function recursively. Thus, the training feature space ($W = \{W_1, \dots, W_m\}$) was given such that ($W_j = \{W_{j,i}\}_{j\in m, i\in|W_{i,j}|}$) with the predictive classes ($P_{class} = \{C_1, C_2\}: C_1 = 1, and\ C_2 = -1$). Each feature vector was represented as ($W_j = \{C_k, W_{j,i}\}_{i\in|W_{i,j}|, k\in|C_k|}$), as illustrated in **Figure**

2. The prior probability $P(C_k)$ was computed as per Equation (5) to predict how often each class occurs over (W) relatively to the feature vector (W_j). Whilst, the conditional probability of (W_j) was computed by Equations (6) to predict the relevance between the predicted class (C_k) and its corresponding feature ($W_{j,i}$) as it was indicated by ($P\left(W_{j,i}|C_k\right)$). Then, the predictive class of each examined feature vector (W_j) as well as any miss-examined feature vector (W_j) were prioritized by *Optimized Feature Co-occurrence (OFC)* and *Phishing Induction Ratio (PIR)* as per Equations (7), (8) and (9) respectively [17, 18]. It is noteworthy to mention that *OFC* was optimized from the formerly used criterion (*Co-occurrence*) in [17] to update the predictive class labels of features. Unlike *Co-occurrence*, OFC computed the features' frequencies with respect to all labeled phish and not-phish feature vectors (W_j) across the training feature space (W).

$$\mathrm{P}\left(\mathrm{W_j} \lceil \mathrm{C_k}\right) = \mathrm{P}(\mathrm{C_k}) \prod_{\mathrm{e}=1\rightarrow \mathrm{p}} \left(\mathrm{W_{j,i}} | \mathrm{C_k}\right) \tag{5}$$

$$\mathrm{C_k} = \mathrm{C_j} \rightarrow \mathrm{P_{ms}}\left(\mathrm{W_j}, \mathrm{C_k}\right) \tag{6}$$

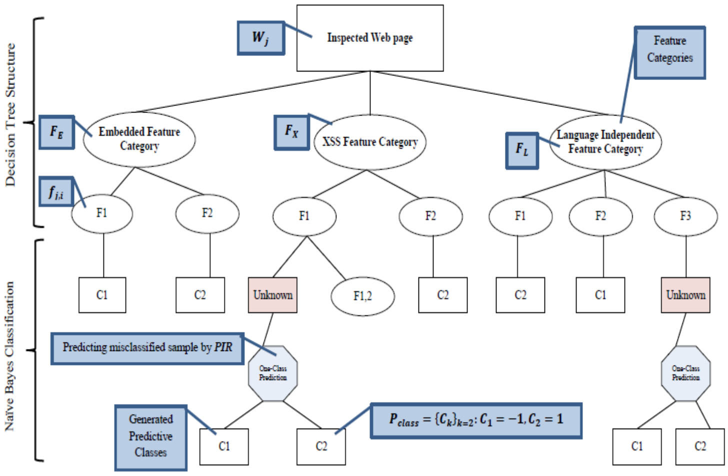
$$C_f = \frac{C_{f,D} - C_{f,D'}}{C_{f,D} + C_{f,D'}} \tag{7}$$

Where$d \in D, d' \in D'$ *and* $f \in F$, *D* is the set of phish websites, and D' is the set of non-phish websites. Then, C_f is the occurrence of each feature *f* belongs to feature vector *F* in all instances that included in *D* and *D'*. Whereas, $C_{f,\,D}$ is the occurrence of feature *f* with respect to all instances in *D*, and $C_{f,\,D'}$ is the occurrence of feature *f* with respect to all instances in D'.

$$P_r(P \mid f_i) = \frac{N_{f_i \rightarrow P}}{N_{f_i \rightarrow P} + N_{f_i \rightarrow L}} \tag{8}$$

$$Phishing\ Induction\ Ratio, PIR\left(W_j\right) = \frac{\sum_{i=1}^{|W_{j,i}|} \mathrm{Pr}(P|f_i)}{|W_{j,i}|} \tag{9}$$

Where W_j is the examined webpage, *Phish Ratio* (W_j) is the prediction of phishing susceptibility, f_i is the feature in W_j, $N_{f_i \rightarrow P}$ is the number of occurrences of f_i in phish instance, $N_{f_i \rightarrow L}$ is the number of occurrences for f_i in legitimate instance, and $\left|W_{j,i}\right|$ is the number of features in W_j.


Decision Tree Structure
Naïve Bayes Classification
W_j
Inspected Web page
Feature Categories
F_E
Embedded Feature Category
F_X
XSS Feature Category
F_L
Language Independent Feature Category
$f_{j,i}$
F1
F2
F1
F2
F1
F2
F3
C1
C2
Unknown
F1,2
C2
C2
C1
Unknown
Predicting misclassified sample by *PIR*
One-Class Prediction
One-Class Prediction
Generated Predictive Classes
C1
C2
$P_{class} = \{C_k\}_{k=2}: C_1 = -1, C_2 = 1$
C1
C2

Note:

W_j is the examined webpage or feature vector;
P_{Class} represents the predictive class such that $P_{class} = \{C_1, C_2\}$;
$C_1 = 1$ denotes phish class;
$C_2 = -1$ denotes not-phish class;
$f_{j,i}$ is a given feature in the examined feature vector;
F_E refers to embedded objects features category;
F_X refers to XSS based features category;
F_L refers to Language independent feature category.

Figure 2. Illustration of phishing classification step throughout **HFBC**

```
HFBC // Hybrid Feature-Based Classifier
Input:   $W$   // Webpage stream such that $W = \{w_m\}_{m \in |W|}$
         $S$   // the set of the proposed novel features
Output:  $Phish_{Data}, Non, Phish_{Data}$
Define   $F_{space}$   // the generated feature space where $F_{space} = \{F_i\}_{i \in |F|}$
         $F_i$         // a feature vector included in $F_{space}$
         $TreeNode$    // constructed decision tree for $F_{space}$
         $F_{sub}$     // a sub-space of $F_{space}$ such that $F_{space} = \{F_{sub}\}_{sub \in |sub|}$
         $S^{\sim}$    // the best chosen feature subset generated by RFSSA
         $C_j$         // a targeting class in $F_i$, where $C = \{C_j\}_{j \in N}$, and $N$ is the number of classes
Begin
    Apply FEA ($W$) to generate $F_{space}$
    For each $F_i$, parse $F_{space}$ for replica and keep only a single copy
```

```
3   REPEAT // pursue steps from (3) to (8)
    a.  Create $TreeNode$
    b.  IF (all feature vectors $\{F\ _i\}_{i\in|F|}$ in $F$ have the same class $C_j$) THEN
        $TreeNode \leftarrow LeafNode$
    c.  IF $F = \{\ \ \}$ THEN attach $TreeNode$ with the majority class $C_j$
4   $S^{\sim} \leftarrow$ RFSSA($S$) // apply RFSSA with FSA to give the best feature subset
    $S^{\sim}$ from original set $S$
5   Exclude $S^{\sim}$ from $S$ such as $S \leftarrow S - S^{\sim}$
6   With $S^{\sim}$, for each $F_i$ in $F_{space}$ do the following steps Classify $F_i$ over $F_{space}$
    a.  Calculate the prior probability of a phish class $C_j$ over $F_{space}$ such that
        $P(C_j|F_{space})$ as per Equation (5)
    b.  Calculate the conditional probability of each feature $f_{i,j}$ in regard to $C_j$
        over $F_{space}$ $P(f_{i,j}|C_j)$ as per Equation (6)
    c.  Update $F_i$ in $F_{space}$ regarding to its class $C_j$ with the maximal
        $P(f_i|C_j)$ $such\ that$ $P(C_j|f_i)$; $C_j \rightarrow P_{ml}(C_j|f_i)$ as per Equation (7)
7   IF any misclassified $F_i$ THEN
    a.  Calculate PIR as per Equations (8) and (9)
    b.  Choose maximal PIR
    c.  Partition $F_{space}$ into sub-spaces such that $F_{space} = \{F_{sub}\}_{sub\in|F|}$ and
        $F_{space} \leftarrow \{F_{sub}\}_{sub\in|F|}$
8   Repeat (6) Until ($F_{space} \neq \{\ \ \}$) AND ($S^{\sim} \neq \{\ \ \}$)
9   Keep all computed probabilities in $\boldsymbol{Phish_{Data}}$ and $\boldsymbol{Non,Phish_{Data}}$
    to classify unknown webpages in the future in
End of HFBC
```

3.2 Benchmarking Datasets

Through experiments, three different benchmarking datasets were utilized to attain the motives of *HFBC*'s performance analysis as they are presented in **Table 1**. The benchmarking datasets differed in their abundance of both phish and legitimate samples, size, and webpage exploits such as homepages, login forms, e-business webpages, end-up, and pharming webpages. Furthermore, they encompassed different hosting languages like English, Chinese, French, Italian, German, Spanish, etc. Each benchmarking dataset was split up into $\frac{2^{nd}}{3}$ and $\frac{1^{rd}}{3}$ splits for both training and testing tasks, respectively. As such, *HFBC* and its competitors were tested and evaluated across the splits of every benchmarking dataset individually. Experimental results were averaged to estimate the overall performance outcomes and overall performance overhead of the tested classifiers.

Table 1. Merits of benchmarking datasets

Merits	***Dataset1***	***Dataset2***	***Dataset3***
Dataset Size	52	2878	96,018

Phish Webpages	36	1382	48009
Legitimate Webpages	16	1496	48009
Dataset Archive	PhishTank /Alexa	Chinese E-Business	PhishTank /DMOZ
Related work	[20]	[8]	[10-12]
Aggregation Time	25-31/7/2010	2014	2012-2015
Webpage Exploits	Login Forms/ Pharming/ e-Business/ End-Up/ Homepages English/ French/ German	e-Business Chinese	e-Business/ Homepages/ Login Forms/ Social networking/ Pharming English/French/ German/Italian/ Spanish etc.

3.3 Performance Measurements

To distinguish a webpage as phish or legitimate, its actual and predictive states can be set in a form of *Confusion Matrix* whose rows contain the actual states and columns cover the predictive states. As such, a correct prediction can be depicted by the diagonal cells whereas actual classifications and misclassifications can be depicted by the other cells. On the basis of *Confusion Matrix*, typical performance evaluation measurements were calculated to evaluate *HFBC* and the comparable machine learning classifiers throughout experiments. The derived measurements included: *True Positive Rate* (*TPR*) that indicated the rate of correctly classified phish samples, and *False Positive Rate* (*FPR*) referred to mistakenly classified legitimate samples as phishes, whereas; *False Negative Rate* (*FNR*) referred to mistakenly labeled phish samples as legitimate ones which implied misclassification cost [23]. Furthermore, *Elapsed Time* was used to compute the amount of time spent by the tested classification model from its start-up to its ending-up. The *Elapsed Time* quantified how long the tested classifiers took to detect phishing on a batch of webpage stream in practice [23]. In addition, *Detection Accuracy Rate* [23-27] was used in to validate the effectiveness of the proposed *HFBC* at detecting zero-hour phish webpages adaptably on the flow of webpages during the real-time practice.

3.4 Experimental Design

As illustrated in **Figure 3,** the empirical workflow was pursued via three analyses: chronological analysis, real-time practice, and comparative analysis across the benchmarking datasets. Accordingly, each dataset was formulated into feature vectors to be manipulated by the tested classification model. It is worthy to mention that 27 computerized simulations for the comparable classification models and benchmarking datasets throughout a highly used tool for data mining that is “WEKA 3.5.7-Waikato Environment for Knowledge Analysis” which is developed by some researchers at the University of Waikato. (27 repetitions of the conducted experiment across three benchmarking datasets).

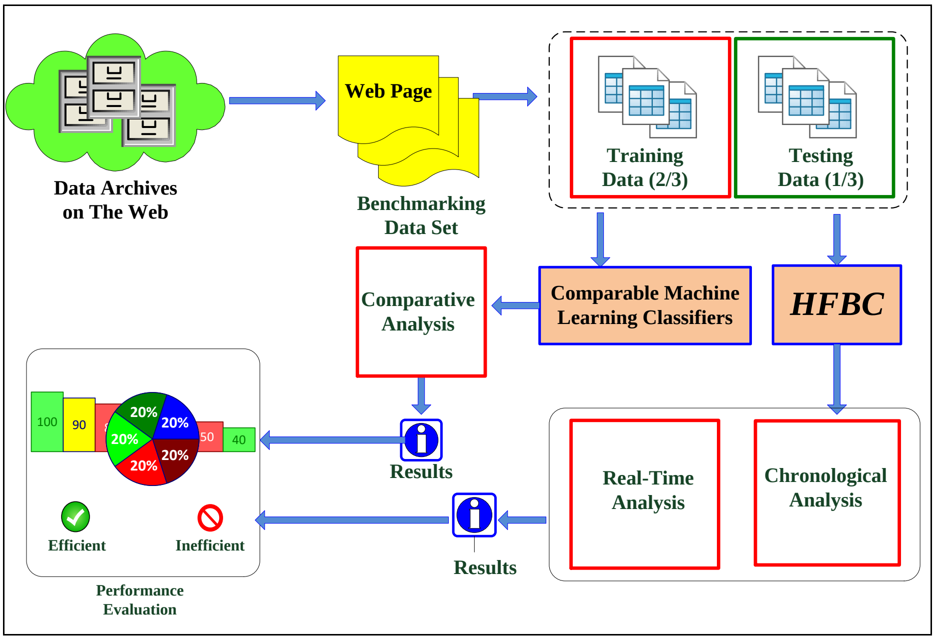


Figure 3: Experimental design

4 Results and Discussion

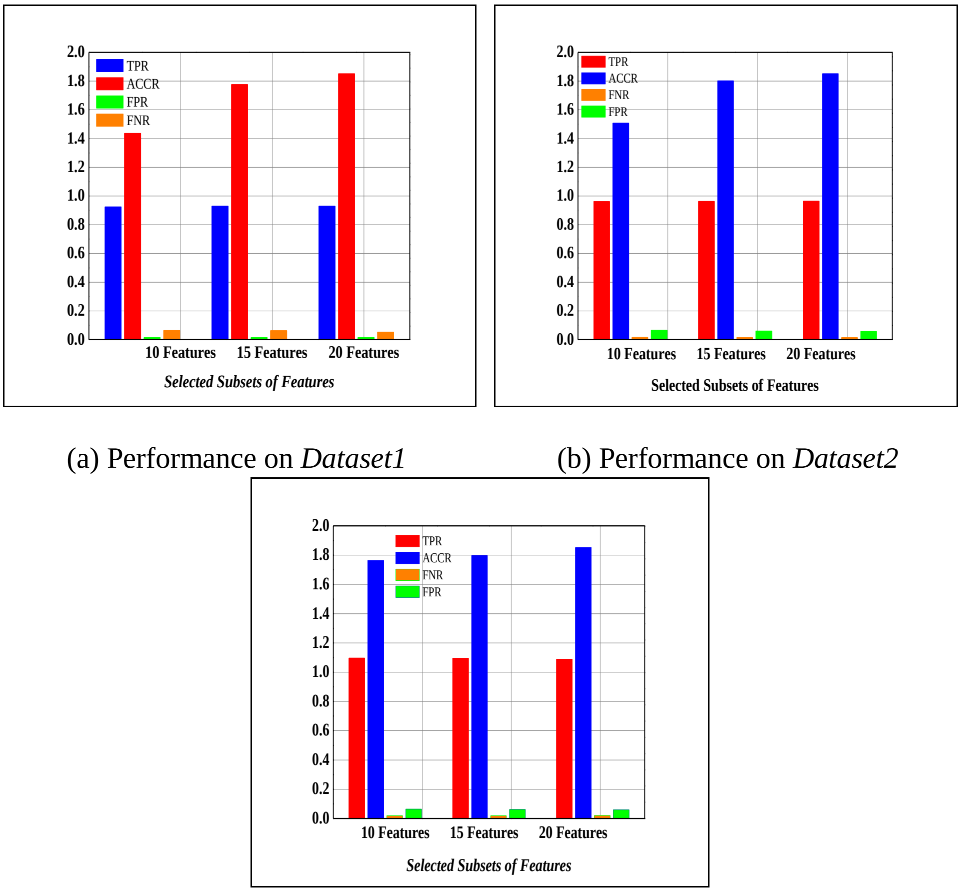
4.1 Chronological Analysis

This chronological analysis was conducted to validate the classification performance of the proposed *HFBC* across different benchmarking datasets and the best selected subsets of features, as shown in **Figure 4.** Throughout a ten-fold cross-validation strategy, the results were merged and evaluated in terms of *TPR*, *FPR*, and *FNR* with respect to the benchmarking datasets. Charts in **Figure 4** showed that *HFBC* could achieve the best rates of *TPR* (from 0.984 to 0.989), *FPR* (from 0.051 to 0.066), and *FNR* (from 0.014 to 0.0156). Overall, the evaluated rates disclosed the following issues:

i. High *TPRs* and low *FPRs* and *FNRs* implied that the chosen subsets of features were decisive to characterize typical and zero-hour phish webpages holistically. Furthermore, *HFBC* could classify almost phish exploits on balanced and imbalanced datasets besides leveraging dataset's scalable size. That was due to the robustness of the selected feature subsets in terms of their goodness and stability against the chronological evolving datasets. Although, the selected subsets of features had several features in common and varied in their settings, they characterized phish webpages similarly versus the scale of datasets. That was attributed to the maximal relevancies and minimal redundancies of all-inclusive features to different phish exploits over all datasets. Hence, they had potential effects on the *HFBC's* performance despite of their distinctions in compactness and settings.

ii. *HFBC* achieved accurate classification versus the scale in size and the variable abundance in phish class of the benchmarking datasets (see **Figure 4 (a)**). This was attributed to building the predictive classes in a structure of *DT* for learning known labeled datasets as well as pruning the classification process of *DT* by the induction function of *NB* for learning unknown instances.

iii. Since *HFBC* utilized *OFC* and *PIR* as its own constraints of the decision to tune its own default induction boundaries (i.e. the hybrid induction settings of *DT* and *NB*). Therefore, *HFBC* could re-examine any misclassified features or feature vectors in the learning datasets, and it could judge their phishing class with minimal false rates (see **Figure 4 (b)** and **Figure 4 (c)**).



(a) Performance on *Dataset1*

(b) Performance on *Dataset2*

(c) Performance on *Dataset3*

Figure 4. Chronological analysis of **HFBC**

Altogether, (i) the hybridity of two machine learning algorithms (*NB* and *DT*) for more effective phishing classification, (ii) the synchronization of two statistical ratios (*OFC* and *PIR*) for more decisive phishing prediction, (iii) the conjunction of *HFBC* with *RFSSA* for robust feature subsets selection, and (iv) the hybrid set of 58 new and different features for holistic phishing characterization; provided potential induction factors to *HFBC* in phishing detection on three different datasets.

4.2 Real-Time Analysis

This section exhibits the real-time analysis of the proposed *HFBC* during one-month sampling interval on evolving webpage streams. During real-time analysis, classification outcomes were reported and evaluated day after day by averaging the *Detection Accuracy Rate (ACCR)* as per Equation (10).

$$Detection\ Accuracy\ Rate\ (ACCR) = \frac{TPR+TNR}{N_{total}} \qquad (10)$$

Where, TPR denotes the number of is correctly testified non-phish webpages, TNR is the number of correctly testified phish webpages, and N_{total} is the total number of all inspected webpages included in the dataset.

Respectively, the plotted values of *ACCRs* in **Figure 5** showed the progressive effectiveness of *HFBC* from the 1st day to the 30th day and they inferred that *HFBC* could manifest its adaptive and effective classification against both the zero-hour phish webpages (new phishes) as well as the prevalent phish webpages. That was attributed to the hybridization of *NB* and *DT* induction margins besides the synchronization of *OFC* and *PIR* which could update *HFBC*'s default induction margins actively on every batch of webpage flow. However, a radical escalating and/or de-escalating of performance trend line was reported at certain days. This was caused by the merits of the daily fetched webpage batches that varied in their size, their webpage exploits, and their rational and irrational distribution of phish class to non-phish class. Moreover, the daily grabbed batch of webpages might have different ratios across webpage functionality and exploitations that might need a long-term crawling and a complex processing. Furthermore, each grabbed batch of webpages might encompass typical and/or new that were either embedded objects, or cross-site scripting (*XSS*), or language independent features, or hybrid features. Thus, *HFBC* yielded a variable performance outcomes during 30 days of real-time practice.

On the other hand, the elapsed time required to examine a batch of 100 webpages per day was varied according to the examined webpages themselves as it can be observed from **Figure 6**. Indeed, an examined webpage might be either a zero-hour phish variant, or a prevalent phish variant, or a valid legitimate webpage. So far, such webpages might exploit login-forms, pharming, homepages, e-business websites, etc. Thus, the elapsed time required for a single webpage or a batch of webpages could vary according to the webpages' exploitations and classes. That, in turn, could increase or decrease the time spent by the steps of features extraction and/or features selection and/or phishing classification during *HFBC*'s implementation.

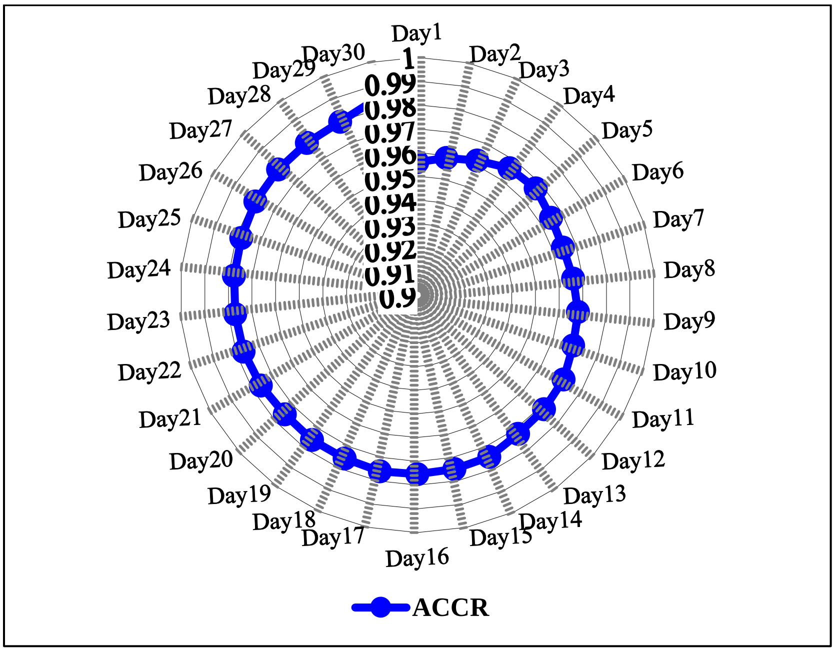


Figure 5. Daily basis real-time analysis of *HFBC* in accuracy rate

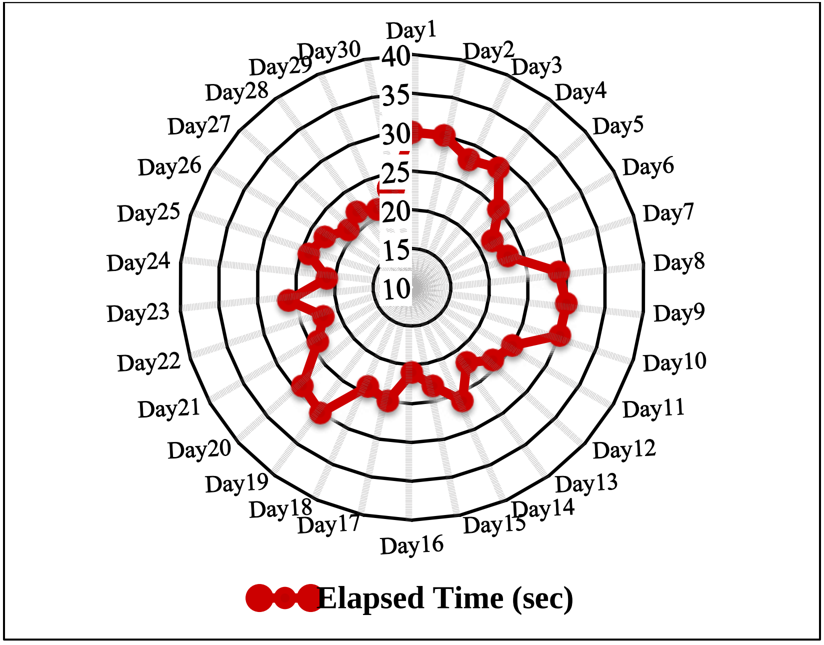


Figure 6. Daily average elapsed time spent by HFBC to tackle phishes on a batch of 100 webpages during real-time analysis.

4.3 Comparative Analysis

This comparative analysis was conducted to appraise the classification performance of the proposed *HFBC* versus those machine learning-based classifiers adopted by the state of the art of anti-phishing techniques. The chart legends in **Figure 7** pointed out how the *HFBC* and its competitors including *SMO*, *SVM*, *TSVM*, *NB* and *DT*; did perform in the presence of the three different benchmarking datasets. Obtained classification outcomes were exposed in terms of *TPR*, *FPR*, and *FNR*.

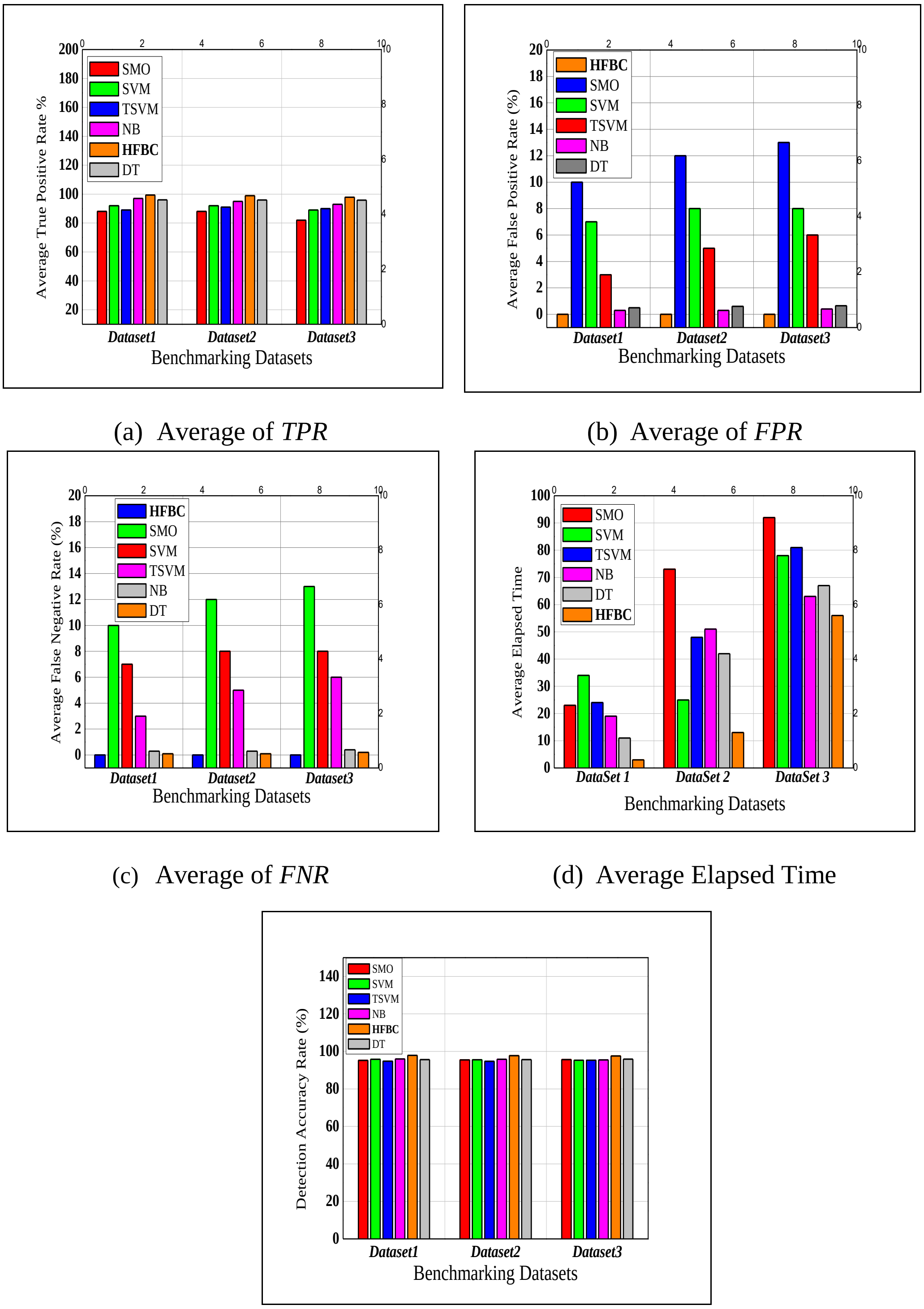
So far, *HFBC* showed its superiority among its competitors as presented in **Figure 7**. This was disclosed to:

i. As shown in **Figure 7** (**a**), the comparable classifiers rendered variations in *TPRs* because they fall short in characterizing new phishing features and various phish webpage exploits. Whilst, *HFBC* achieved the highest *TPRs* which assured its decisive characterization at different phish exploits and its effective classification at zero-hour phish webpages across the datasets.

ii. The active learning of *HFBC* versus the inactive learning of the comparable classifiers attained the minimal false classifications, i.e. *FPRs* of *HFBC* were closest to zero among those of its competitors as shown in **Figure 7** (**b**). Because *HFBC* could adjust its initial induction margins by hybridizing the induction functions of both *NB* and *DT* with an updating criterion like *OFC*. By using *OFC*, *HFBC* could update its default induction margins and then it could adapt various phish webpages and their exploits across benchmarking datasets.

iii. Due to their deficiency at employing new features to characterize novel and leveraging discriminating criteria to classify unknown samples; the comparable classifiers rendered high *FNR*s (see **Figure 7** (**c**)) on the benchmarking datasets. Unlikely, *HFBC* could report very minimal and often negligible *FNR*s (i.e. *FNR*s closed to zero) due to its *PIR* which identifies the phish class of any misclassified sample decisively.

The overall observations restated the important role of inductive factors in machine learning-based phishing classification as well as the distinction of HFBC versus other tested phishing classifiers whose performance outcomes were related to their deficiency of inductive factors partially or wholly. Unlike the revisited phishing classifiers [5-7, 10-12] specifically those adopted *NB* and *DT* [17-21], both the hybridization of *NB* and *DT*, and the synchronization of *OFC* and *PIR* criteria could manifest the induction settings of *HFBC* progressively and converge the overlooked features vectors from the remaining feature space iteratively for more decisive and effective phishing classification as depicted in **Table 2**.



(a) Average of *TPR*

(b) Average of *FPR*

(c) Average of *FNR*

(d) Average Elapsed Time

(e) Average of Detection Accuracy Rate

Figure 7. Performance outcomes of **HFBC** during the comparative analysis

Table 2. General comparative analysis of *HFBC* and its competitors

Work **Issues**	**[5]**	**[6-7]**	**[8]**	**[9]**	**[10-11]**	**[12]**	**This work**
Machine Learning Algorithm	SVM, LR, DT	SVM	SMO, LR, RF, NB	SVM, RF, JRip	k-NN	SVM, RF, C4.5, JRip,	*HFBC*
Classifier Type	Ensemble	Single	Single	Single	Single	Ensemble	Hybrid
Statistical Induction	Not	Not	Not	Not	Not	Not	*OFC, PIR*
New Features	3	7	Not	Not	12	12	*58*
Features Selection Criteria	Not	Not	χ^2	CFS, IG, χ^2	Not	Not	*mRMR*
Features Subset Selection Algorithm	Not	Not	Not	Not	Not	Not	*RFSSA*
Big Datasets	13000	2464	2878	1400	96000	96000	100, 000
Web page Exploits	e-Business Login Form Social-Networking Pharming English	e-Business Login Forms Pharming English	e-Business Chinese	e-Business Login Forms Pharming English French	e-Business Pharming English French, German Italian Spanish Portuguese	e-Business Pharming English French, German Italian Spanish Portuguese	e-*Business Pharming Login Forms Social-Networking English French, German Italian Spanish Portuguese*
Active Learning	Active	Not	Not	Not	Not	Active	*Active*
External Resources	Google Trends, Yahoo Clues, Blacklist of Phish Web pages	Google Trends, Yahoo Clues, Blacklist of Phish Web pages	Whitelist of Chinese e-business web pages	_____	Google Trends, Yahoo Clues	Google Trends, Yahoo Clues	*Not*
Real-Time Application	Two-weeks practice	Not	Not	Not	Not	Not	*One Month Practice*
Performance Outcomes	♣ TPR (92%), FPR (1.4%)	♣ TPR (99.6%), FPR (0.42%)	♣ TPR (95.83%) FPR: (2.2)	♣ TPR (94.9%) FPR: (1.44)	♣ TPR (99%), FPR (0.37%)	♣ TPR (96.71%) FPR (0.7%)	♠ *TPR (98.24%), FPR (0.05%) FNR:0.032% Accuracy (0.97) Elapsed Time: 21sec*
Notes	*HFBC*=Hybrid Feature-Based Classifier, SVM =Support Vector Machine; LR=Logistic Regression; BN=Bayesian; DT, C4.5, and JRip are types of Decision Tree Classifier; RF=Random Forest; k-NN=Neural Network; SMO=Sequential Minimal Optimization, NB=Naïve Bayes.						
♣	Overall performance evaluation outcomes as they are presented in the related works.						
♠	Overall performance evaluation outcomes as they are achieved by this work						

5 Conclusions and Future Work

By revisiting the current achievements in machine learning-based anti-phishing domain, it is observed that they affirmed to be computationally effective but in-adaptive to accomplish real-time phishing detection. That is due to their full or partial deficiency of inductive factors such as rich set of features, big web data and its class imbalance, actively learned feature-based classifier, and adaptable modelling. By restating the causality between their limitations and their inductive deficiency throughout an empirical analysis; future outlooks are suggested to promote their induction power. Furthermore, a phishing classification model could be extended in the future via a high level assembly integrating functionally inter-relating and synchronously working modules to adapt zero-hour phish patterns on the evolving webpage stream. Regarding the issues stated in this paper at building any machine learning-based classification model; effectiveness of classification could be elevated along with reducing the misclassification and computational cost. Additionally, this paper with the underlined perspectives are hoped to serve as a navigating taxonomy to the reseachers for their future efforts.

References

1. H. Z., Zeydan, A. Selamat, M. Salleh, Survey of anti-phishing tools with detection capabilities. In 14th Int. Symposium on Biometrics and Security Technologies (ISBAST'2014), 46-54, Kuala Lumpur-Malaysia (August, 2014).
2. H. Z., Zeydan, A. Selamat, M. Salleh, Current state of anti-phishing approaches and revealing competencies. Journal of Theoretical and Applied Information Technology, 70(3), 507-515 (2014).
3. H., Zuhair and A., Selamat, Phishing classification models: Issues and perspectives. In IEEE Conference on Open Systems (ICOS'2017), 26-31, IEEE, Miri-Sarawak, Malaysia (2017).
4. H. Zuhair, A., Selamat, M. Salleh, Feature Selection for phishing detection: a review of research. Int. Journal of Intelligent Systems Technologies and Applications, 15(2), 147-162 (2016).
5. G. Xiang, Towards a phish free world: a cascaded learning framework for phishing detection. Doctoral Dissertation, Carnegie Mellon University, Pittsburgh, PA 15213 (2013).
6. R. Gowtham, and I. Krishnamurthi, A comprehensive and efficacious architecture for detecting phishing webpages. Computers & Security, 40, 23-37 (2014).
7. R. Gowtham, and I. Krishnamurthi, PhishTackle-a web services architecture for anti-phishing. Cluster Computing, 17(3), 1051-1068 (2014).
8. D. Zhang, Z. Yan, H. Jiang, H., and T. Kim, A domain-feature enhanced classification model for the detection of Chinese phishing e-Business websites. Information & Management,51(7), 845-853 (2014).
9. R. M. Mohammad, F. Thabtah, F., and L. McCluskey, Predicting phishing websites based on self-structuring neural network. Neural Computing and Applications, 25(2), 443-458 (2014).
10. S. Marchal, J. François, R. State, and T. Engel, PhishScore: hacking phishers' minds. In 10th International Conference on Network and Service Management (CNSM2014), pp. 46-54, IEEE, Rio de Janeiro (17-21 Nov. 2014).
11. S. Marchal, S., J. François, R. State, and T. Engel, PhishStorm: detecting phishing with streaming analytics. IEEE Transactions on Network and Service Management, 11(4), 458-471 (2014).

12. S. Marchal, DNS and semantic analysis for phishing detection. Doctoral Dissertation. University of Luxembourg (22 June 2015).
13. H. Y. N., Abutair and A., Belgith, Using case based reasoning for phishing detection. Procedia Computer Science 109C, 281–288 (2017).
14. S. Elnagar, and M. Thomas, A cognitive framework for detecting phishing websites. In International Conference on Advances on Applied Cognitive Computing (ACC 2018), 60-61 (2018).
15. H., Zuhair, M. Salleh, and A. Selamat, New hybrid features for phish website prediction. Int. Journal of Advance Softcomputing and Applications, 8(1), 28-43 (2016).
16. H., Zuhair, M. Salleh, and A. Selamat, Hybrid features-based prediction for zero-hour phish website. Jurnal Teknologi, 78(12-3), 95-109 (2016).
17. H. Zuhair, A. Selamat, M. Salleh, Selection of robust feature subsets for phish webpage prediction using maximum relevance and minimum redundancy criterion. Journal of Theoretical and Applied Information Technology, 81(2), 188-205 (2015).
18. H. Zuhair, A. Selamat, M. Salleh, The effect of feature selection on phish website detection: an empirical study on robust feature subset selection for effective classification. Int. Journal of Advanced Computer Science and Applications, 6(10), 221-232 (2016).
19. Y., Pan, and X., Ding, Anomaly Based Web Phishing Page Detection. Proceedings of 22nd Annual Conference on Computer Security Applications (ACSAC'06). Springer, Heidelberg (4186), 381-392 (2006).
20. J. P., Vink, and G., Haan, Comparison of machine learning techniques for target detection. Artificial Intelligence Review, Springer , 43, 125-139 (2015).
21. G., Kumar, K., Kumar, and M., Sachdeva, The use of artificial intelligence based techniques for intrusion detection: a review. Artificial Intelligence Review, Springer , 34(4), 369-387 (2010).
22. H., Shahriar, and M., Zulkernine, Trustworthiness testing of phishing websites: a behavior model-based approach. Future Generation Computer Systems, 8(28), 1258–1271 (2012).
23. M., Khonji, Y., Iraqi, and A., Jones, Phishing detection: a literature survey. Communications Surveys and Tutorials, IEEE, 15(4), 2091-2121 (2013).
24. Y., Li, R., Xiao, J., Feng, and L., Zhao, A semi-supervised learning approach for detection of phishing webpages. Optik-International Journal for Light and Electron Optics, 124(23), 6027-6033 (2013).
25. C. K., Olivo, A. O., Santin, and L. S., Oliveira, Obtaining the threat model for e-mail phishing. Applied Soft Computing, 13(12), 4841-4848 (2013).
26. O., Kwon, and J. M., Sim, Effects of dataset features on the performances of classification algorithms. Expert Systems with Applications, 40(5), 1847-1857 (2013).
27. Miyamoto, D., Hazeyama, H. and Kadobayashi, Y. (2008). November. An evaluation of machine learning-based methods for detection of phishing sites. In International Conference on Neural Information Processing (pp. 539-546). Springer, Berlin, Heidelberg.