SMOTEEN Hybrid Sampling Based Improved Phishing Website Detection

SUBMISSION DATE / POSTED DATE

01-07-2022 / 05-07-2022

CITATION

Srivastava, Jaya; Sharan, Aditi (2022): SMOTEEN Hybrid Sampling Based Improved Phishing Website Detection. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.20207765.v1

DOI

10.36227/techrxiv.20207765.v1

SMOTEEN Hybrid Sampling Based Improved Phishing Website Detection

Jaya Srivastava [ID1], and Aditi Sharan [ID2]

***Abstract*—In several real-world tasks such as Cancer Patient Detection, Phishing Website Detection, etc. the datasets suffer from skewed class distribution. However, the traditional Machine Learning (ML) classification algorithms are based on two basic assumptions (i) balanced class distribution, and (ii) equal misclassification costs, both of which get violated when faced with class imbalanced datasets, thereby leading to inaccurate predictive ML models. Thus, appropriately handling of the class imbalance plays a crucial role in developing high performing and accurate predictive ML models. In this study we propose four novel Phishing Website Classification models namely, SMOTEENN-XGB, SMOTEENN-LR, SMOTEENN-RF, and SMOTEENN-SVM by combining SMOTEENN (SMOTE + ENN) hybrid sampling technique with eXtreme Gradient Boosting (XGB), Logistic Regression (LR), Random Forest (RF), and Support Vector Machine (SVM) classifiers. We propose the use of SMOTEENN hybrid sampling as the novel approach to address the problem of class imbalance in Phishing Website datasets prior to building classification models. In Section IV RESULT ANALYSIS AND DISCUSSION, the comparative evaluation of the performance of our novel proposed four models as against the (i) Original (ii) SMOTE and (iii) ENN approaches and with other related studies, successfully demonstrates that the two of our four novel proposed models i.e., the SMOTEENN-XGB and SMOTEENN-RF outperforms the others. To the best of our knowledge and belief our novel proposed four models SMOTEENN-XGB, SMOTEENN-RF, SMOTEEEN-LR, and SMOTEENN-SVM for Phishing Website Detection based on SMOTEENN hybrid sampling approach have not been published in the existing studies as of now.**

***Index Terms*—Class Imbalance, Cyber Security, ENN, Phishing, SMOTE, SMOTEENN.**

I. INTRODUCTION

In real-world scenarios such as Fraud Detection, Rare Disease Detection, Electricity Pilferage, Phishing Website Detection etc. where anomaly detection is crucial, there exists severe class imbalance in the training datasets. Class imbalance is said to occur in a dataset when the class distribution of the classes present in the dataset are approximately not equal. The class having relatively huge number of instances, say, 99% of the total sample space is called the Majority class, and the class having relatively very less number of instances is called the Minority class. For example, consider the case of Credit Card Fraud detection problem, where, let's say the fraudulent transactions are just 1% of the total dataset. This Credit Card Transaction dataset is said to have class imbalance of the order of 100:1. Most often in such real-world scenarios of anomaly detection, it is the minority class which is 'important' or 'interesting' rather than the majority class which being 'unimportant' from the investigation point of view is also termed as 'uninteresting' under the given context.

In real-world scenarios, the misclassification costs are also not equal. For example, in a binary classification task such as Phishing Website Detection, the objective is to correctly identify the Phishing Website (minority class) rather than the Legitimate Website (majority class). And therefore, a Phishing Website when predicted as a Legitimate Website (False Negative) has much more serious consequences than when a Legitimate Website predicted as a Phishing Website (False Positive). Clearly the misclassification costs in the class imbalanced real-world datasets are also not equal as assumed by the traditional machine learning algorithms [1] - [3]. According to Provost [4] another the assumption that is built into ML models is to maximize accuracy which also falls flat in case of class imbalanced datasets. According to [1] - [4] accuracy as performance evaluation measure is misleading for the class imbalanced datasets.

The main objectives of this study are:

- to explore and propose better ways of handling class imbalance towards producing unbiased and more accurate predictive Phishing Website Detection Machine Learning models having high performance measures with low false positives, and low false negatives.
- to adopt appropriate performance evaluation measures such as Area under Receiver Operating Characteristics (ROC) Curve, Area under Precision-Recall (PR) Curve, F1, GMean and Accuracy and as suggested by [1], [5] - [6].

In line with the laid down objectives, this study deals with handling the between-class imbalance by using and comparing three data-level balancing approaches (i) Synthetic Minority Oversampling Technique (SMOTE) [7] (ii) Edited Nearest Neighbors (ENN) under sampling [8] and (iii) SMOTEEN hybrid sampling [9] approaches, where SMOTEENN (SMOTE + ENN) employs the SMOTE oversampling and ENN and under sampling techniques as the underlying data sampling approaches.

Corresponding author: J. Srivastava.

J. Srivastava is with Indian Institute of Technology Delhi, AD-147, Computer Services Centre, Hauz Khas, New Delhi, INDIA. (E-mail: jaya@iitd.ac.in).

A. Sharan is with School of Computer and Systems Sciences (SC & SS), Jawaharlal Nehru University, New Delhi, INDIA. (E-mail: aditisharan@mail.jnu.ac.in).

The four major contributions of this study are:

- Proposing the use of SMOTEENN hybrid sampling [9] to balance out the Phishing Website dataset(s) adequately and effectively, first by oversampling using SMOTE [7] oversampling technique and then performing data-cleaning to remove noise , overlapping etc. using ENN under sampling [8] technique.
- Proposing four models, namely, SMOTEENN-XGB, SMOTEENN-RF, SMOTEENN-LR and SMOTEENN-SVM based on four classifiers i.e., XGB, RF, LR and SVM respectively.
- Comparing our four novel proposed models with the existing data-balancing approaches i.e. (a) Original, (b) SMOTE oversampling, and (c) ENN under sampling as provided in Figs. 4 thru 8 and TABLE VI, and
- Comparative evaluation and validation of the performance of our four novel proposed models with the existing studies as provided in TABLE VII.

The rest of this study is structured as follows: In Section II. BACKGROUND AND RELATED WORKS, we present a background of the class imbalance problem and state-of-the-art solutions as provided by various researchers. We also provide a brief review of the related works. In Section III. METHODOLOGY, we discuss out our research methodology which includes the (a) dataset, (b) feature set, (c) class imbalance handling methods, (d) classification algorithms, (e) performance evaluation measures, (f) proposed framework, and (g) the experimental setup used by us in this study. In Section IV. RESULTS AND DISCUSSION, we present, analyze and discuss out the experimental results of our four novel proposed models, i.e., SMOTEENN-RF, SMOTENN-XGB, SMOTEENN-LR and SMOTEENN-SVM. We also present comparative evaluation of our four novel proposed models among themselves as well as with the existing related studies that have used the same dataset [10] as used by us in this study, as well as some related studies [21], [28], and [30] that have used different Phishing Website datasets [34] – [36]. In Section V. CONCLUSION, we conclude our findings and briefly mention about the limitations of our proposed approach and future work plans.

II. Background And Related Works

In this section we present a brief review of the literature which includes (1) the Class Imbalance problem, (2) the State-of-the-Art solutions for Class Imbalance problem, (3) Performance Evaluation measures as identified and proposed by various researchers, and (4) a review of the Related Works [22] - [27], and [29] that have used the same dataset [10] as used by us in this study, as well as some other related studies [3], [21], [28], and [30] that have used different UCI ML Repository [34]-[35], and Mendeley [36] Phishing Website datasets.

1) The Class Imbalance Problem

Fig. 1 depicts the inescapable class imbalance problem in the real-world actual datasets which adversely impacts the performance of the Machine Learning (ML) Models. As depicted in Fig. 1, the traditional classification algorithms are based on three assumptions (i) the distribution of classes in the dataset is approximately equal or balanced, (ii) the misclassification costs are equal for all the classes present in the dataset, and (iii) the goal of ML Models is to maximize Accuracy [1] - [3]. When the traditional machine learners are provided such class imbalanced datasets, they ignore the minority class thereby yielding biased and inaccurate ML models. Such biased and inaccurate ML Models are not of much use to us for predictive modelling purposes as they produce unsatisfactory and unreliable results. For example, ML Model based on severely class imbalanced training dataset having 99% Majority class samples and 1% Minority class samples would achieve 99% Accuracy with any classification algorithm that makes Majority class predictions for everything [4].

Garcia *et al.* [1], and Guan *et al.* [3] etc. discussed in detail about the class imbalance problem and the state-of-the-art solutions with supportive examples and well explained diagrams. According to [1] - [3] the class-imbalance can be of two types (i) between-class imbalance and (ii) within-class imbalance. The between-class imbalance may happen naturally as well as artificially e.g. it may happen (i) in a natural way because of the type of the data or transactions that gets collected or accumulated over a period of time such as fewer fraudulent credit card transactions as compared to the huge number of valid credit card transactions, and/or (ii) it may happen in an artificial way when a class balanced data gets collected in a faulty manner to actually make it class imbalanced with samples of some particular class(es) inadvertently got missed due to network glitches etc., resulting into class distribution being split into majority versus minority class(es). The with-in class imbalance occurs when several sub concepts of the main concept(s) i.e., the majority and /or the minority class (es) are also present in the dataset. The degree of between-class imbalance in real-world datasets can be of the order of 100:1, 1000:1, 10,000:1 or even more. They also differentiated between (i) intrinsic class imbalance which arises due to nature of dataset for example, 1% of fraudulent financial transactions versus 99% of legitimate financial transactions and (ii) extrinsic class imbalance which may arise due to external factors such as time and storage, for example, when a continuous stream of audio or video data is collected with some jitter or sporadic transmissions due to network glitches, delays, outages etc. They also mentioned about relative class imbalance that occurs in real-world datasets when the minority class instances remain relatively very less, say in thousands, (e.g., fraudulent financial Transactions) even after oversampling them thousand times more, for instance, from 1000 to 10000 instances, still they fail to match up with the relatively huge number, say millions or billions of the majority class instances (e.g., legitimate financial transactions). They also noted that the data complexity due to class overlapping, the existence of small disjuncts or smaller sub concepts of the main concept(s), lack of the availability of the representative data etc., are the primary factors which deteriorates the classification performance. The complex manner in which various concepts and their sub-concepts represent themselves in the dataset together with the relative between-class imbalance significantly impacts the classification performance of the machine learners.

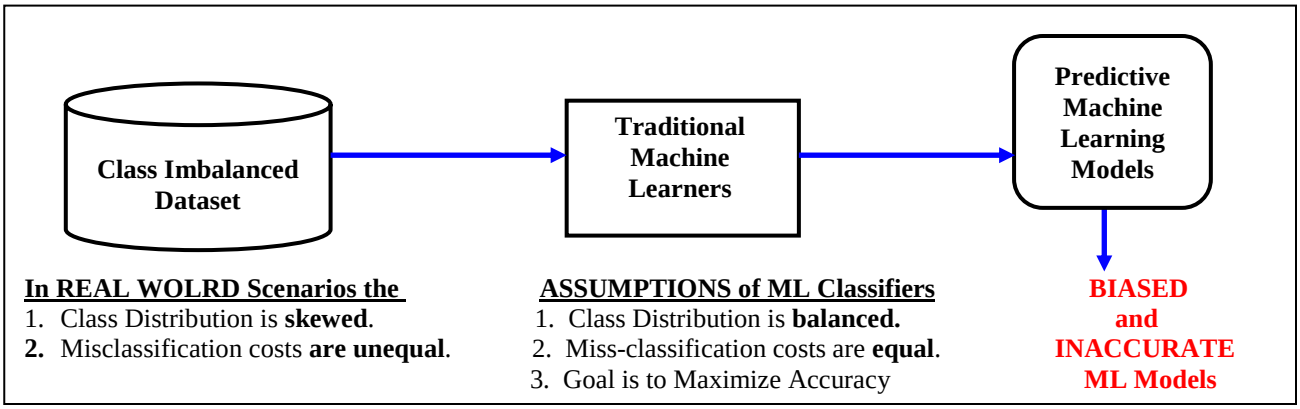

Fig. 1. The Class Imbalance Problem.

2) The State-of-the-Art Techniques for Handling the Class Imbalance

Fig. 2 depicts the State-of-the-Art techniques for handling the class imbalance problem. These may be broadly grouped into (1) the Dataset-balancing and (2) the Algorithm-balancing techniques [1] – [9], and [11-13]. Dataset-balancing solutions aim at reducing the class imbalance by modifying the class distribution using (a) Oversampling e.g. SMOTE [7] (b) Under sampling e.g. ENN[8] (c) Hybrid Sampling e.g. SMOTEENN[9], SMOTETomek[9]. Hybrid sampling techniques combine oversampling and under-sampling techniques. The Algorithm-balancing solutions modify the behavior of the underlying classification algorithms by using a cost or weight function etc. e.g. (a) Cost-Sensitive Learning [33] (b) One Class Learning (c) Kernel-based learning (d) Active Learning, and (e) Feature Selection Algorithm [1] - [2], and [17] - [18]. Both of these solutions aim at straightening out the bias towards the majority class making the prediction results more reliable and accurate [14]. However, each of such approaches have their advantages and disadvantages. Oversampling techniques may lead to overfitting and high computational and memory cost [1] – [2], and [5]. Under sampling techniques may lead to under fitting and prospective loss of some valuable information [1] - [2], and [5]. The detailed discussion of each these techniques is beyond the scope of this study.

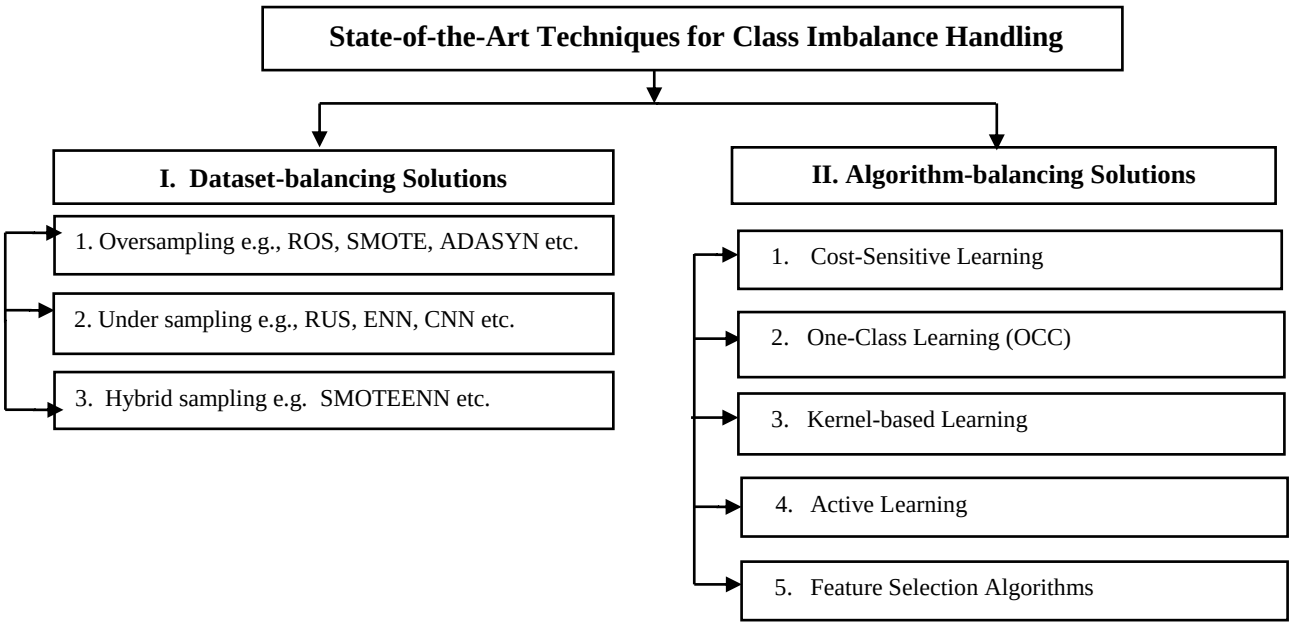

Fig. 2. The State-of-the-Art Techniques for Class Imbalance Handling.

3) Performance Evaluation Measures

According to [1], [6], and [19] – [20] the class imbalance severely impacts the predictive accuracy of machine learners, and therefore Area under Receiver Operating Characteristics (ROC) Curves, Precision, Recall, F1, GMean, Area under Precision-Recall (PR) Curves, and Cost Curves (CC) should be used.

4) Related Works

In this section, we briefly present a review of some of the related studies. Through the review of the related literature, we observed that most of the studies have ignored the prevalent class imbalance that exists in the Phishing website dataset [10] that is also used by us. Further, most of the studies have used only Accuracy as the performance evaluation measure without adequately handling the class imbalance prevalent in the Phishing Website training dataset [10]. Guan *et al*. [3] proposed SMOTE-WENN method which combines SMOTE[7] , followed by Weighted ENN. They used 'Weighted' distance function instead of the 'Euclidean' distance in the traditional kNN algorithm. Sagnik *et al*. [21] used the dataset [35] and employed four Swarm Intelligence (SI) optimization algorithms, namely Bat, Whale Optimization (WO), Grey Wolf Optimization (GWO) and Firefly, to find the optimum values of the hyper parameters of the RBF Kernel of the SVM classifier. Alsariera *et al*. [22] proposed four novel Meta-Learner Phishing Website Detection models, using Extra-tree base classifier combined with Adaboost.M1 (ABET), Bagging (BET), Rotation Forest (RoFET) and LogitBoost (LBET). Bikku *et al*. [23] proposed an Optimized Random Forest Algorithm by hyper tuning of the Random Forest classifier. Sindhu *et al*. [24] employed lexical feature analysis on the dataset [10] and developed three ML models using RF, SVM and Neural Network with Backpropagation, and implemented their best SVM (Accuracy 97.451%) model as the Google Chrome browser extension. Aminu *et al*. [25] selected top 24 features of dataset [10] using RF Feature Importance algorithm, followed by XGBoost for classification. Tabassum *et al.* [26] used (i) Feature Importance using Random Forest and XGBoost, and (ii) Correlation Matrix with Heatmap, and selected top 23 features and proposed a hybrid classifier based on SVM + DT + RF and XGBoost. Prayago *et al.* [27] employed SMOTE [7], and Information Gain and Correlation Heat Map to select top 12 features from the dataset [10] and used kNN as classifier. Prince *et al.* [28] used five Feature Selection techniques, namely, Chi-Square, Information Gain (IG), Gain Ratio (GR), Pearson Correlation Coefficient (PCC), and Principal Component Analysis (PCA). Their RF model with top 32 features yielded the best accuracy (98.36%). Ahmed *et al.* [29] selected top 20 features using (i) Wrapper Feature Selection (WFS), and (ii) Correlation-based Feature Selection (CFS) using Decision Tree classifier. Pristianto *et al.* [30] used two different UCI ML Phishing Website datasets [34] – [35] and employed hybrid sampling using OSS [15], followed by SMOTE [7] . They used SVM, kNN, DT and ANN classifiers.

III. METHODOLOGY

In this section we discuss out our research methodology which includes the description of the dataset, proposed fraemwork, the experimental set up and performance evaluation measures adopted by us.

1) Dataset Description

In this study we have used publically available benchmark Phishing Website dataset from UCI ML Repository Mohammad *et al.* [10]. The dataset consists of a total of 11,055 samples and 31 attributes consisting of 30 features and one class label [31] – [32]. TABLE I provides a description of the dataset used by us. The Class Imbalance Ratio is given by the equation (1).

TABLE I
PHISHING WEBSITE DATASET DESCRIPTION

Total Samples	**Total Features**	**CLASS DISTRIBUTION**		**Class Imbalance Ratio (IR)***
		Phishing Websites	**Legitimate Websites**	
11055	30	4898	6137	IR= 1.25296

$$* \text{ Class Imbalance Ratio} = \frac{(\text{Size of Majority Class})}{(\text{Size of Minority Class})} \quad (1)$$

2) Class Imbalance Handling

We have explored three data-balancing approaches: (i) Synthetic Minority Oversampling Technique (SMOTE) [7] (ii) Edited Nearest Neighbor (ENN) [8] under sampling, and (iii) the SMOTEENN [9] Hybrid Sampling for class imbalance handling. In SMOTEEN, we first employ SMOTE [7] oversampling method to oversample the minority class samples. The problem with SMOTE is that it assumes that between any two minority class samples the entire space belongs to the minority class [33] which may not be correct. We have used ENN [8] under sampling technique as the data cleaning approach to remove the noise that may get introduced when the new synthetic minority samples invade into the majority sample space. In TABLE II, we enlist some parameters of (i) SMOTE [7], (ii) ENN [8] and SMOTEENN [9] used by us. In Table III, we provide the Class Distribution of the Original dataset as well as the dataset obtained after performing (i) SMOTE [7] (ii) ENN[8], and (iii) SMOTENN [9] (proposed).

3) Classification Algorithms Used

We have used the four classifiers i.e., Random Forest (RF), eXtreme Gradient Boosting (XGB), Logistic Regression (LR), and Support Vector Machine (SVM), with their hyper parameters enlisted in TABLE IV. In TABLE V we present the Performance Evaluation Measures adopted by us.

TABLE II
PARAMETERS USED IN THE DATA SAMPLING ALGORITHMS

DATA SAMPLING METHOD	PARAMETER	VALUE	DESCRIPTION
1. SMOTE [7]	sampling_strategy	auto'	Resample Minority Class.
	random_state	42	Seed used by random number generator
	k_neighbors	5	Consider 5 Nearest Neighbors to construct a new synthetic sample.
	n_jobs	None	No. of CPU Cores used during cross validation.
	n_features	30	No. of features in the input data.
2. ENN [8]	sampling_strategy	auto	Resample Majority class.
	n_neighbors	3	Neighborhood size =3 to compute the Nearest Neighbors.
	kind_sel	mode'	Use the Majority Vote of the neighbors to exclude a sample.
	n_jobs	None	
3. SMOTEENN [9]	sampling_strategy	auto'	Resample the Majority class.
	random_state	42	Seed used by random number generator
	smote	None	Use SMOTE object with default values of the parameter.
	enn	None	Use ENN object with kind_sel='all' and default values of the parameter.
	n_jobs	None	No. of CPU Cores used during cross validation.

TABLE III
CLASS DISTRIBUTION

	ORIGINAL		SMOTE [7]		ENN		SMOTEENN (Proposed)	
Class	**# Samples**	**Distribution (%)**	**# Samples**	**Distribution (%)**	**# Samples**	**Distribution (%)**	**# Samples**	**Distribution (%)**
Legitimate	6157	55.69	6157	54.46	6027	55.17	5649	50.02
Phishing	4898	44.31	5149	45.54	4898	44.83	5644	49.98

4) Proposed Framework

Based on our findings, we propose the use of SMOTEENN Hybrid Sampling as the data-balancing technique to handle the class imbalance in the Phishing Website Detection dataset. The Fig. 3 depicts our four novel proposed models 'SMOTEENN-RF', 'SMOTEENN-XGB', 'SMOTEENN-LR', and 'SMOTEENN-SVM'. As is evident from Fig. 3, we first perform SMOTE [7] oversampling on the original dataset to generate new synthetic minority class samples, followed by ENN [8] under sampling as a data cleaning approach. Then we spilt the SMOTEENN-balanced dataset into 80% Training dataset and 20% Testing dataset using stratified sampling. The stratified sampling maintains the class proportions in the training and testing datasets. After that we perform classification using RF, XGB, LR and SVM to obtain four Machine Learning Models, namely, SMOTEENN-RF, SMOTEENN-XGB, SMOTEENN-LR and SMOTEENN-SVM. For comparative evaluation purposes we also perform classification on the (i) Original dataset and the datasets obtained after (ii) SMOTE oversampling, and (iii) ENN under sampling.

5) Experimental Setup

We have performed all our experiments on the Intel® Core™ i7-4770S CPU @ 3.1 GHZ with 8 GB RAM and 64-bit Windows OS. The Anaconda Navigator (64 bit) version 4.10.3 with jupyter Notebook version 6.0.3, and Python version 3.6.12 Machine Learning Libraries such as sklearn, imblearn, matplotlib, numy etc. formed our experimental test bed.

TABLE IV
HYPER PARAMETERS USED IN THE CLASSIFICATION ALGORITHMS

CLASSIFIER	HYPERPARAMETER	VALUE	DESCRIPTION
RF	n_estimators	1000	No. of trees in the Random Forest = 1000
	max_features	sqrt(30)=5	Maximum no. of features (=5) to consider at every split in a node. By default, max_features=sqrt(
	criterion	'gini'	Node split criteria. By default, criterion = 'gini'
	bootstrap	TRUE	With bootstrap=True (default) the samples are drawn with replacement.
XGBoost	booster	'gbtree'	Type of learner is 'gbtree' i.e. model will consist of an ensemble of trees.
	n_estimators	1000	No. of Decision Trees in the XGBoost = 1000
	objective	binary:logistic	XGBoost loss function for binary classification.
	learning_rate	0.5	Learning Rate of Model = 0.5. It signifies how fast a model learns.
SVM	kernel	linear	Type of Kernel = 'linear'. By default, kernel='rbf'.
	C	10	Regularization Parameter. By default, C=1.0
	degree	3	Degree of polynomial kernel. Ignored by all. By default, C=3.
	tol	0.001	Tolerance for stopping criteria. By default, tol=0.001
LR	C	1	Inverse of regularization strength. By default, C=1.
	penalty	l2	Add penalty. By default, penalty=l2.
	solver	lbfgs	Algorithm to use in optimization. By default, solver='lbfgs'.
	tol	0.0001	Tolerance for stopping criteria. By default, tol=0.0001 .

TABLE V
PERFORMANCE EVALUATION MEASURES

PERFORMANCE EVALUATION MEASURE	FORMULA
Accuracy	(TP+TN)/(TP + TN + FP + FN)
Precision or Positive Predictive Value (PPV)	TP/(TP+FP)
Recall or Sensitivity or TPR	TP/(TP+FN)
F1	(2*Precision*Recall) / (Precision + Recall)
Specificity or TNR	TN/(TN+FP)
GMean	GMean= √ (Sensitivity ∗Specificity)

IV. RESULT ANALYSIS AND DISCUSSION

As is evident from TABLE III, we achieved almost balanced class distribution with our proposed SMOTEEN Hybrid Resampling [9], followed by ENN [8], then SMOTE [7] and lastly Original (Imbalanced) dataset. In TABLE VI, we compare our novel proposed four models namely, SMOTEENN-RF, SMOTEEN-XGB, SMOTEENN-LR, and SMOTEEN-SVM with the (i) Original dataset and datasets obtained after (ii) SMOTE oversampling, and (iii) ENN under sampling. In TABLE VII, we present the comparative evaluation of our proposed approach with the existing studies to validate our claim.

1) Comparative Evaluation of our Models

We present below comparative evaluation of each of our novel four models, namely SMOTEENN-RF, SMOTEEN-XGB, SMOTEENN-LR, and SMOTEEN-SVM with respect to (i) Original (ii) ENN balanced, and (iii) SMOTE balanced datasets using the traditional Random Forest (RF), eXtreme Gradient Boosting (XGBoost), and Support Vector Machine (SVM) classification algorithms among themselves. The highest values of all the performance metrics i.e., ROC_AUC, PR_AUC, F1, GMean and Accuracy are notably achieved in all of our four novel proposed models, and are indicated in bold face in TABLE VI. Empirical results indicate that our novel proposed four models based on SMOTEENN hybrid resampling technique outperforms the

SMOTE oversampling and ENN under sampling approaches**.**

a) Performance Evaluation of SMOTENN-XGB Model

The bar plot in Fig. 4a visually depicts the comparative evaluation of performance of the XGBoost (XGB) Classifier on the (i) Original Dataset (Orig-XGB) and the datasets obtained after (ii) ENN under sampling (ENN-XGB), (iii) SMOTE oversampling (SMOTE-XGB), and (iv) SMOTEEN Hybrid resampling (SMOTEENN-XGB) using F1, GMean and Accuracy as the performance measures. The highest performance of our novel proposed SMOTEENN-XGB model using F1, GMean and Accuracy is distinctively visible in Fig 4a.

Similarly, the ROC plots in Fig. 4b visually depicts the comparative evaluation of the of performance of the XGB Classifier on the (i) Original Dataset (Orig-XGB) and the dataset obtained after (ii) ENN under sampling (ENN-XGB), (iii) SMOTE oversampling (SMOTE-XGB), and (iv) SMOTEEN Hybrid resampling (SMOTEENN-XGB) with Area under Receiver Operating Characteristics (ROC) Curve. The highest ROC_AUC of our proposed SMOTEENN-XGB model is clearly visible in Fig. 4b.

As is evident from TABLE VI, the bar plot in Fig. 4a and the ROC plots in Fig. 4b , our novel proposed approach of performing the SMOTEENN Hybrid Sampling data-balancing approach prior to XGBoost classification, significantly enhanced the performance of the XGBoost (XGB) classifier on the Original dataset from **96.98%(ROC_AUC)** to **99.82% (ROC_AUC),** from **97.62% (PR_AUC)** to **99.91% (PR_AUC),** from **96.71% (F1)** to **99.82% (F1)** , from **96.98% (GMean) to 99.82% (GMean),** and from **97.11% (Accuracy) to 99.82% (Accuracy)),** where the former % value refers to the Original dataset and the latter % value refers to that obtained after SMOTEENN Hybrid Sampling (proposed).

Similarly, as is evident from TABLE VI, the bar plot in Fig. 4a and the ROC plots in Fig. 4b, the performance of the XGBoost (XGB) Classifier with SMOTENN resampled dataset was far superior than with the ENN resampled dataset as well as the SMOTE resampled dataset, which are being employed in the in existing studies [27], [30] etc. on Phishing Website Detection

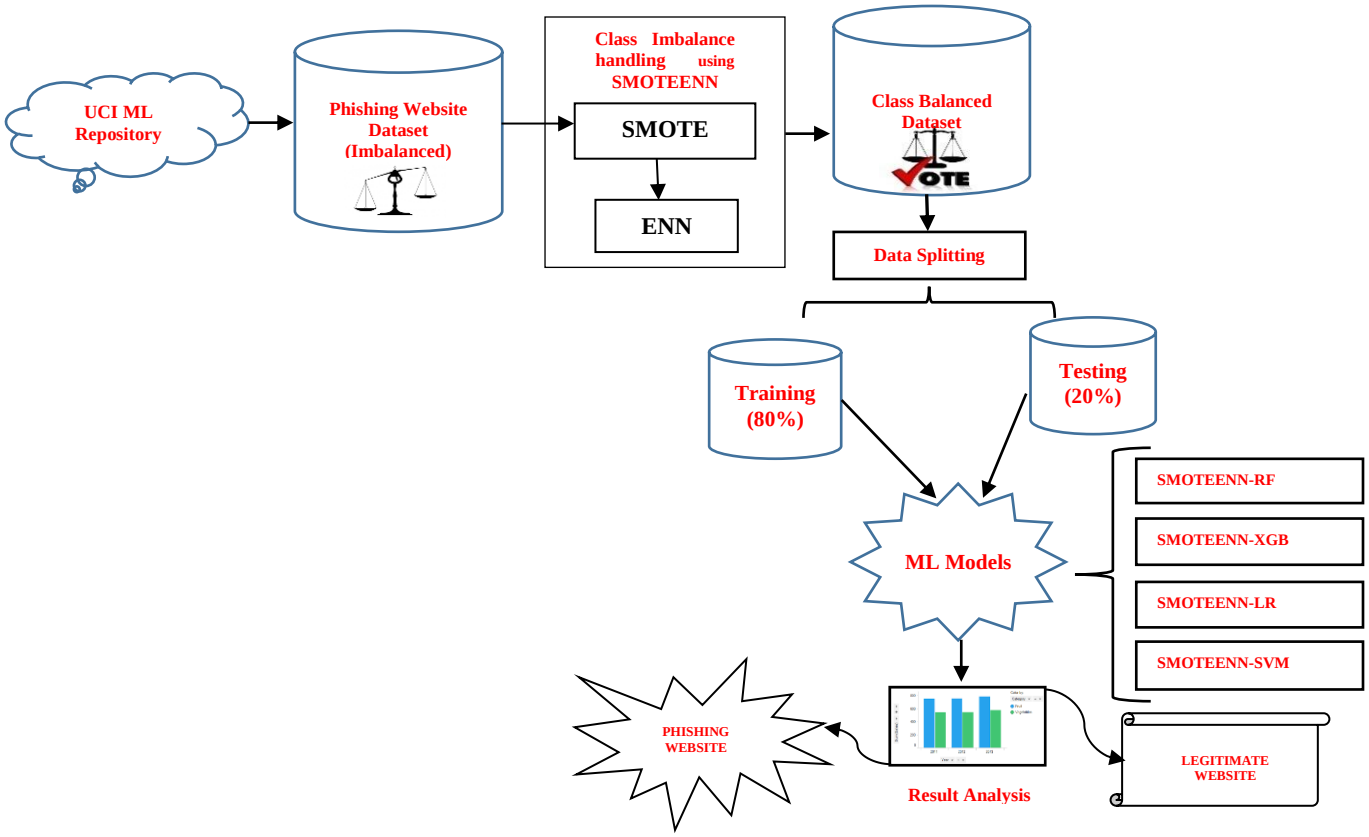

Fig. 3. Proposed Framework based on SMOTEEN Hybrid Sampling approach.

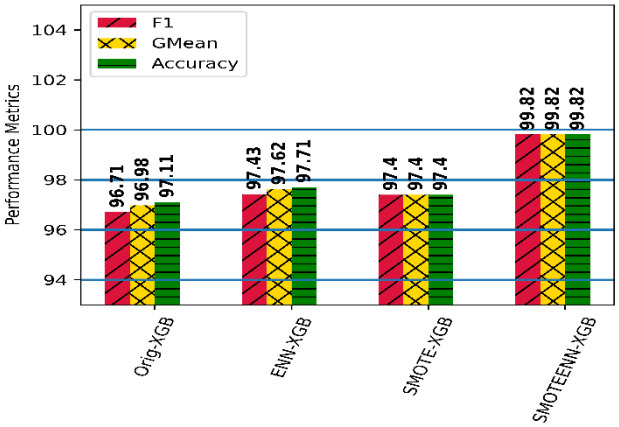

Fig. 4a. Performance of XGB Classifier.

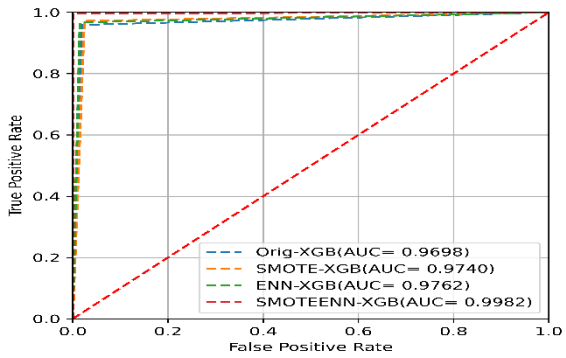

Fig. 4b. Comparison of ROC Curves of XGB Classifier.

b) Performance Evaluation of SMOTENN-RF Model

The bar plot in Fig. 5a. visually depicts the comparative evaluation of performance of the RF Classifier on the (i) Original Dataset (Orig-RF) and the datasets obtained after (ii) ENN under sampling (ENN-RF), (iii) SMOTE oversampling (SMOTE-RF), and (iv) SMOTEEN Hybrid resampling (SMOTEENN-RF) using F1, GMean and Accuracy as performance measures. The highest values of F1, GMean and Accuracy of our novel proposed SMOTEENN-RF model is distinctively visible in Fig 5a.

The ROC plots in Fig. 5b. visually depicts the comparative evaluation of the of performance of the RF Classifier on the (i) Original Dataset (Orig-RF) and the dataset obtained after (ii) ENN under sampling (ENN-RF), (iii) SMOTE oversampling (SMOTE-RF), and (iv) SMOTEEN Hybrid resampling (SMOTEENN-RF) using the Area under Receiver Operating Characteristics (ROC) Curve. The highest ROC_AUC of our proposed SMOTEENN-RF model is clearly visible in Fig 5b. The same holds true for the rest of our novel proposed models, namely, SMOTEEN-XGB, SMOTEENN-LR and SMOTEENN-SVM as explained below.

As is evident from TABLE VI, the bar plot in Fig. 5a, and the ROC plots in Fig. 5b, our novel proposed approach of performing data-balancing using the SMOTEENN Hybrid resampling prior to Random Forest classification significantly enhanced the performance of the Random Forest (RF) classifier on the Original dataset from **96.93% (ROC_AUC)** to **99.47% (ROC_AUC),** from **97.46% (PR_AUC)** to **99.65% (PR_AUC),** from **96.62% (F1)** to **99.47% (F1)** , from **96.93% (GMean) to 99.47% (GMean),** and from **97.01% (Accuracy) to 99.47% (Accuracy),** where the former % value refers to the Original dataset and the latter % value refers to that obtained after SMOTEENN Hybrid Sampling (proposed).

Similarly, as is evident from TABLE VI, the bar plot in Fig. 5a, and the ROC plots in Fig. 5b, the performance of the Random Forest (RF) Classifier on SMOTENN resampled dataset is far superior than on the ENN resampled dataset as well as the SMOTE resampled dataset.

c) Performance Evaluation of SMOTENN-LR Model

The bar plot in Fig. 6a visually depicts the comparative evaluation of performance of the Logistic Regression (LR) Classifier on the (i) Original Dataset (Orig-LR) and the datasets obtained after (ii) ENN under sampling (ENN-LR), (iii) SMOTE oversampling (SMOTE-LR), and (iv) SMOTEEN Hybrid resampling (SMOTEENN-LR) using F1, GMean and Accuracy as performance measures. The highest performance of our novel proposed SMOTEENN-LR model using F1, GMean and Accuracy is distinctively visible in Fig. 6a.

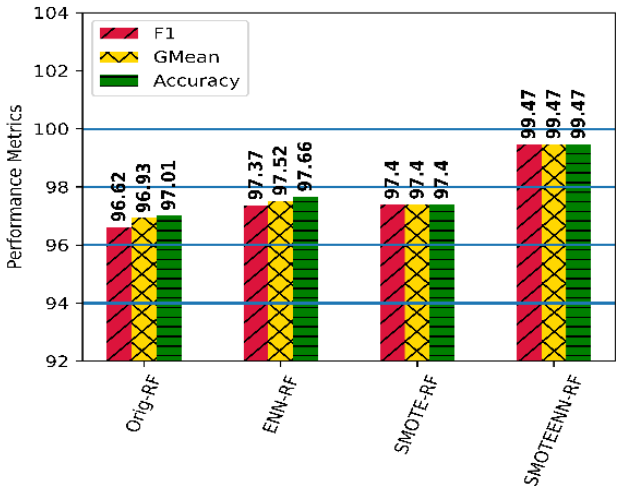

Fig. 5a. Performance of Random Forest (RF) Classifier.

Similarly, the **ROC plots in Fig. 6b** visually depicts the comparative evaluation of the of performance of the Logistic Regression (LR) Classifier on the (i) Original Dataset (Imbalanced-LR) and the dataset obtained after (ii) ENN under sampling (ENN-LR), (iii) SMOTE oversampling (SMOTE-LR), and (iv) SMOTEEN Hybrid resampling (SMOTEENN-LR) using the Area under Receiver Operating Characteristics (ROC) Curve. The highest ROC_AUC of our proposed SMOTEENN-LR model is visible in **Fig. 6b**.

As is evident from TABLE VI, the bar plot in Fig. 6a, and the ROC plots in Fig. 6b, our novel proposed approach of combining the SMOTEENN Hybrid resampling prior to Logistic Regression (LR) classification, significantly enhanced the performance of the LR classifier on the Original dataset from **90.98% (ROC_AUC) to 94.52% (ROC_AUC)**, from **92.37% (PR_AUC) to 95.90% (PR_AUC),** from **89.96% (F1) to 94.55% (F1)** , from **90.96 % (GMean) to 94.52% (GMean)**, and from **91.18% (Accuracy) to 94.52% (Accuracy)**, where the former % value refers to the Original dataset and the latter % value refers to that obtained after SMOTEENN Hybrid Sampling (proposed).

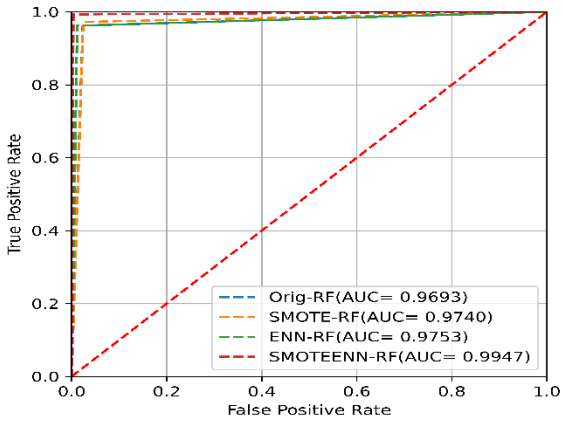

Fig. 5b. Comparison of ROC Curves of RF Classifier.

Similarly, as is evident from TABLE VI, the bar plot in Fig. 6a, and the ROC plots in Fig. 6b, the performance of the Logistic Regression (LR) Classifier with SMOTENN resampled dataset was far superior than with the ENN resampled dataset as well as the SMOTE resampled dataset.

d) Performance Evaluation of SMOTENN-SVM Model

Fig. 7a visually depicts the comparative evaluation of performance of the SVM Classifier on the (i) Original Dataset (Orig-SVM) and the datasets obtained after (ii) ENN under sampling (ENN-SVM), (iii) SMOTE oversampling (SMOTE-SVM), and (iv) SMOTEEN Hybrid resampling (SMOTEENN-SVM) using F1, GMean and Accuracy as performance evaluation measures. The highest performance of our novel proposed SMOTEENN-SVM model using F1, GMean and Accuracy is distinctively visible in Fig 7a.

Similarly, the Fig. 7b visually depicts the comparative evaluation of the of performance of the SVM Classifier on the (i) Original Dataset (Orig-SVM) and the dataset obtained after (ii) ENN under sampling (ENN-SVM), (iii) SMOTE oversampling (SMOTE-SVM), and (iv) SMOTEEN Hybrid resampling (SMOTEENN-SVM) using Area under Receiver Operating Characteristics (ROC) Curve as the performance evaluation measure. The highest ROC_AUC of our proposed SMOTEENN-SVM model is clearly visible in Fig 7b.

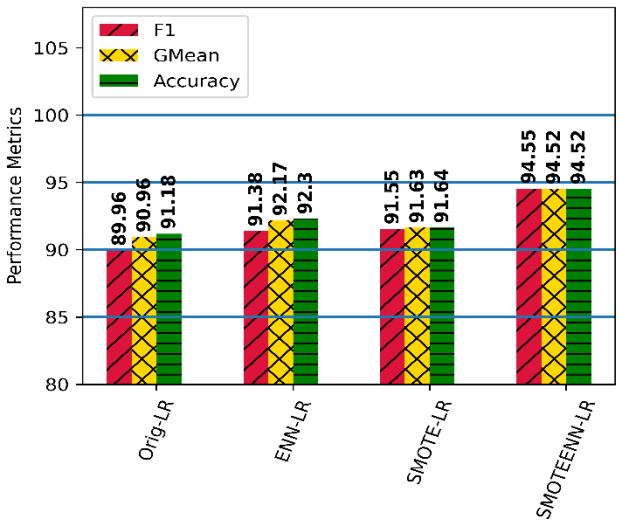

Fig. 6a. Performance of Logistic Regression (LR) Classifier.

As is evident from TABLE VI, the bar plot in Fig. 7a and the ROC plots in Fig. 7b, our novel proposed approach of combining the SMOTEENN Hybrid resampling prior to SVM classification, significantly enhanced the performance of the SVM classifier on the Original dataset from **90.15% (ROC_AUC) to 93.71% (ROC_AUC),** from **92.19% (PR_AUC) to 95.57% (PR_AUC),** from **89.04 % (F1) to 93.66 % (F1)** , from **90.06% (GMean) to 93.70% (GMean),** and from **90.59% (Accuracy) to 93.70 % (Accuracy)**), where the former % value refers to the Original dataset and the latter % value refers to that obtained after SMOTEENN Hybrid Sampling (proposed).

Similarly, as is evident from TABLE VI, the bar plot in Fig. 7a and the ROC plots in Fig. 7b, the performance of the SVM Classifier with SMOTENN resampled dataset was far superior than with the ENN resampled dataset as well as the SMOTE resampled dataset.

1) Comparative Evaluation of Our Four Models

As is evident from TABLE VI, and the bar plot in Fig. 8 the **SMOTEENN-XGB** model outperformed the rest three models i.e., SMOTEENN-RF, SMOTEENN-LR and SMOTEEN-SVM with **99.82% (AUROC), 99.91% (PR_AUC), 99.82% (F1) , 99.82% (GMean) and 99.82% (Accuracy)** , followed **by SMOTEENN-RF** with 99.47% (AUROC), 99.65% (PR_AUC), 99.47% (F1) , 99.47% (GMean) and 99.47% (Accuracy), then **SMOTEENN-LR** with 94.52% (AUROC), 95.90% (PR_AUC), 94.55% (F1) , 94.52% (GMean) and 94.52% (Accuracy), and **SMOTEENN_SVM** with 93.71% (AUROC), 95.57% (PR_AUC), 93.66% (F1) , 93.70% (GMean) and 93.70% (**A**ccuracy) the last.

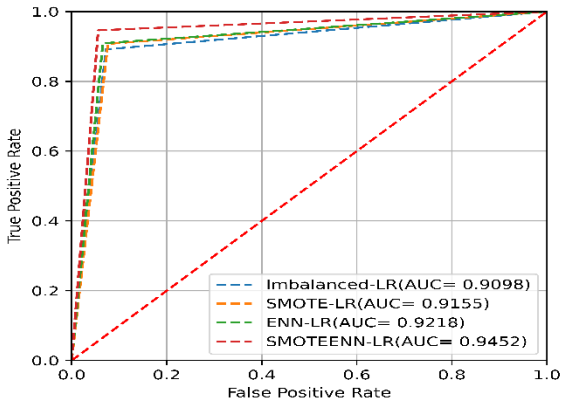

Fig. 6b. Comparison of ROC Curves of LR Classifier.

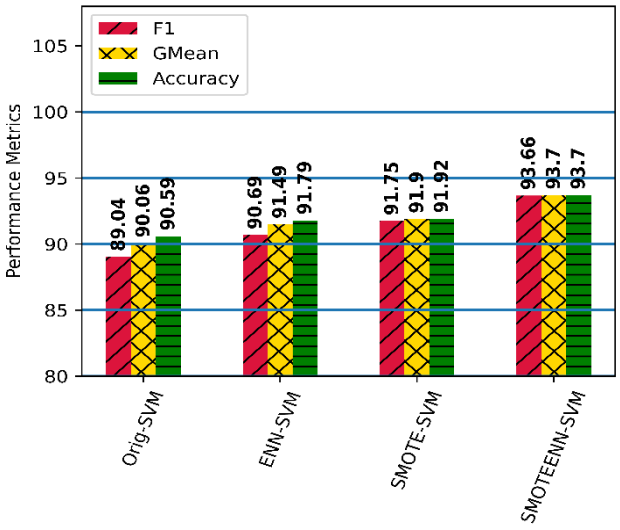

Fig. 7a. Performance of SVM Classifier.

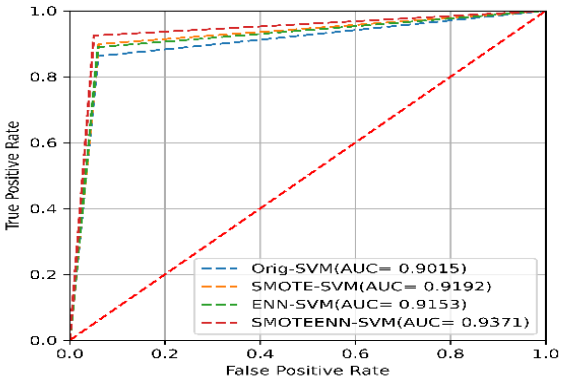

Fig. 7b. Comparison of ROC Curves of SVM Classifier.

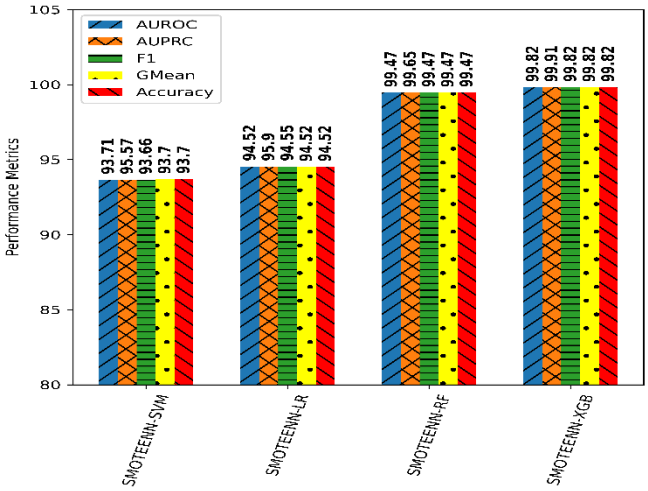

Fig. 8. Classifiers performance on SMOTEENN (Proposed).

1) Comparison and Validation of Our Proposed Approach With The Existing Studies

In this section we present comparative evaluation of our proposed approach with the exiting related studies as presented in TABLE VII**.**

When comparing our proposed models with Sagnik *et al.* [21] and as is evident from TABLE VII, we found that the performance of all of our four models, namely SMOTEENN-XGB (Accuracy 99.82% & F1 0.9982), and SMOTEENN-RF (Accuracy 99.47% & F1 0.9947), SMOTEENN-LR (Accuracy 94.52 % , and F1 0.9455), and SMOTENN-SVM (Accuracy 93.71%, and F1 0.9366) outperformed all of their four models i.e. BA-SVM (Accuracy 92.99%, and F1 0.9183), GWO (Accuracy 92.99%, and F1 0.9182), FA (Accuracy 92.69%, and F1 0.9150), and WOA (Accuracy 92.62 %, and F1 0.9142) .

When compared with Alseria *et al.* [22] from TABLE VII, all their four models, i.e. ABET, RoFET, BET, and have low accuracy and F1 as compared with the two of our best models i.e. SMOTEENN-XGB (Accuracy 99.82%, F1 0.9982) and SMOTEENN-RF (Accuracy 99.47%, F1 0.9947).

On comparing our results with Bikku *et al.* [23] from TABLE VII, the Accuracy (97.62%) and AUROC (61%) of their proposed Optimized RF Model are significantly less than our two best novel proposed models SMOTEENN-XGB (Accuracy 99.82%, AUROC 0.9982) and SMOTEENN-RF (Accuracy 99.47%, AUROC 09947).

When we compare our results with Sindhu *et al.* [24] from TABLE VII, we find that our novel proposed models SMOTEENN-XGB (Accuracy 99.82%, F1 0.9982 and SMOTEENN-RF (Accuracy 99.47%, F1 0.9947), performed significantly better than their RF (Accuracy 97.37%), SVM (Accuracy 97.451%), and NN with Back Propagation (Accuracy 97.260%).

Aminu *et al.* [25] employed Random Forest Feature Importance algorithm to select top 24 features from the same dataset [10] as used by us in this study. Then they used XGBoost algorithm to detect Phishing Websites. As is

evident from TABLE VII, our novel proposed model SMOTEENN-XGB (Accuracy 99.82%, F1 0.9982, Precision 1.0, & Recall 0.9965) outperformed their RF-XGBoost model (Accuracy 97.26 %, F1 0.9721, Precision 0.9726 & Recall 0.9789).

Similarly, while performing comparative evaluation of Tabassum *et al*. [26] from TABLE VII, it was found that all of our four novel models outperformed their respective traditional classifier models i.e. XGB, RF, LR and SVM with 30 features on the same UCI ML Repository benchmark dataset [10] as by us. Their best performing Hybrid Classifier model combining SVM, DT, RF and XGBoost with Top 23 features yielded (Accuracy 98.28%, F1 0.98, Precision 0.98, Recall 0.98), which are low as compared to our models.

TABLE VI
COMPARATIVE EVALUATION OF CLASSIFIERS USING DIFFERENT SAMPLING TECHNIQUES

Classifier	Performance Metrics	Original Dataset Imbalanced) (%)	Data Resampling Technique		
			ENN (%)	SMOTE (%)	SMOTEENN (%)
XGB	**ROC_AUC**	96.98	97.62	97.40	**99.82**
	PR_AUC	97.62	98.17	98.20	**99.91**
	F1	96.71	97.43	97.47	**99.82**
	GMean	96.98	97.62	97.48	**99.82**
	Accuracy	97.11	97.71	97.48	**99.82**
RF	**ROC_AUC**	96.93	97.53	97.40	**99.47**
	PR_AUC	97.46	98.23	97.91	**99.65**
	F1	96.62	97.37	97.40	**99.47**
	GMean	96.93	97.52	97.40	**99.47**
	Accuracy	97.01	97.66	97.40	**99.47**
LR	**ROC_AUC**	90.98	92.18	91.55	**94.52**
	PR_AUC	92.37	93.43	94.08	**95.90**
	F1	89.96	91.38	91.55	**94.55**
	GMean	90.96	92.17	91.63	**94.52**
	Accuracy	91.18	92.30	91.64	**94.52**
SVM	**ROC_AUC**	90.15	91.53	91.92	**93.71**
	PR_AUC	92.19	93.2	94.49	**95.57**
	F1	89.04	90.69	91.75	**93.66**
	GMean	90.06	91.49	91.90	**93.70**
	Accuracy	90.59	91.79	91.92	**93.70**

Prayago *et al.* [27], used the dataset [10], employed Information Gain and Correlation for Feature selection to select top 12 features, used SMOTE for data-level balancing and kNN (k=1) classifer. As is evident from TABLE VII our novel proposed models SMOTEENN-XGB (Accuracy 99.82%, F1 0.9982 F1 0.9982, Precision 1.0, & Recall 0.9965) and SMOTEENN-RF (Accuracy 99.47%, F1 0.9947 Precision 0.9965 & Recall 0.9991) outperformed their kNN with SMOTE model (Accuracy 97.47%, F1 0.975 Precision 0.975 & Recall 0.975).

Comparing our results with Prince *et al.* [28] from TABLE VII we find that our SMOTENN-RF model (Accuracy 99.47%) yielded much higher accuracy as compared to their best RF model with top 32 features (Accuracy 98.36%). Their Accuracy of 98.24% their proposed PDCSF model is much less compared to our two proposed models SMOTEENN-XGB (Accuracy 99.82%) and SMOTEENN-RF (Accuracy 99.47%). However, their SVM model with 44 features (Accuracy 94.1%) performed slightly better than our SMOTEENN-SVM model (Accuracy 93.7%).

Comparing our result with Ahmed *et al.* [29] from TABLE VII, it is clear that their best DT model with Wrapper Feature Selection (WFS) (Accuracy 98.80%, F1 0.9793, Precision 0.9804 and Recall 0.9783) is significantly lower than our two best performing models i.e. SMOTEENN-XGB (Accuracy 99.82%, F1 0.9982, Precision 1.0, Recall 0.9965) and SMOTEENN-RF (Accuracy 99.47%, F1 0.9947, Precision 0.9965, Recall 0.9991).

Pristyanto *et al*. [30] used two datasets DS1 [34] and DS2 [35]. Again, as is evident from TABLE VII, our two best performing models i.e. SMOTEENN-XGB (Accuracy 99.82%, F1 0.9982, Precision 1.0, Recall 0.9965) and SMOTEENN-RF (Accuracy 99.47%, F1 0.9947, Precision 0.9965, Recall 0.9991) outperformed their all four models based on DS1 and DS2.

V. CONCLUSION

The class imbalance handling plays a crucial and significant role in developing unbiased, more accurate and high performing machine learning models. The important contributions of this study can be summarized into four. Firstly, we have successfully demonstrated that our novel proposed SMOTEENN Hybrid Resampling method for data-balancing outperformed the other three approaches, namely, (i) Original (Imbalanced) dataset which serves as a baseline, (ii) SMOTE oversampling, and (iii) ENN under sampling. The highest values of ROC_AUC, PR_AUC, F1, GMean and Accuracy obtained from the proposed approach are highlighted in bold in TABLE VI. Secondly, our results indicate that out of the four models the SMOTEENN-XGB model outperformed the rest three models i.e., SMOTEENN-RF, SMOTEENN-LR and SMOTEENN- SVM consistently in all the performance measures such as ROC_AUC, PR_AUC, F1, GMean and Accuracy, followed by SMOTEEN-RF, then SMOTEENN-LR and SMOTEENN-SVM the last. Thirdly, we have also compared our four novel

proposed models with the existing related studies to validate our claim that our novel proposed four models based on SMOTEEN Hybrid sampling data-level balancing approach performed significantly better than the existing related studies [22] – [27], [29] which have used the same dataset [10] as ours. Fourthly, we have also compared our results with other related studies such as [21], [28], and [30] who have used different publically available benchmark Phishing Website Datasets [34] – [36], using same classifiers as ours and/or some different classifiers, and different approaches such as (OSS+SMOTE) [30], it can be said that our novel proposed approach gave significantly better results as compared to these studies. Through our comparative evaluation with related studies using different datasets [34] – [36] than ours [10], we validate the claim of other studies that the dimensionality reduction through optimal feature selection, though a challenging task, plays a crucial and significant role in the performance of the Machine Learning models. Moderate choice of feature set selection may yield moderate performing models than the available full feature set. The Limitations of our proposed model revolve around the limitations of the SMOTE Oversampling and ENN Under sampling techniques that we have used in this study as the underlying methods, and as explained in the Section II RELATED WORKS. The algorithm-level balancing approaches is beyond the scope of this study. In our future work we would like to explore and propose new methods for class imbalance handling both at the data-level and/or at the algorithm-level.

TABLE VII
COMPARISON AND VALIDATION OF OUR PROPOSED APPROACH WITH THE EXISTING STUDIES

S.No	Authors [Reference]	Classification Algorithms	Accuracy	AUCROC	AUPRC	F1	GMean	Precision	Recall
1.	**Jaya *et al.* [Proposed]**	**SMOTEENN-XGB**	**99.820%**	**0.9982**	**0.9991**	**0.9982**	**0.9982**	**100.0000**	**0.9965**
		SMOTEENN-RF	**99.470%**	**0.9947**	**0.9965**	**0.9947**	**0.9947**	**0.9965**	**0.9991**
		SMOTEENN-LR	**94.520%**	**0.9452**	**0.9590**	**0.9455**	**0.9452**	**0.9447**	**0.9464**
		SMOTEENN-SVM	**93.710%**	**0.9370**	**0.9557**	**0.9366**	**0.9370**	**0.9488**	**0.9247**
2.	**Sagnik *et al.* [21]**	FA-SVM	92.690%	-	-	0.9150	-	-	-
		WOA-SVM	92.620%	-	-	0.9142	-	-	-
		GWO-SVM	92.990%	-	-	0.9182	-	-	-
		Optimized RF	91.370%	-	-	0.9021	-	-	-
3.	**Alsariera *et al.*[22]**	ABET	97.485%	-	-	0.9750	-	-	-
		RoFET	97.449%	-	-	0.9740	-	-	-
		BET	97.404%	-	-	0.9740	-	-	-
		LBET	97.576%	-	-	0.9760	-	-	-
4.	**Bikku *et al.* [23]**	XGBoost	94.690%	0.9900	-	-	-	0.9600	0.9300
		RF	96.830%	1.0000	-	-	-	0.9600	0.9500
		LR	92.650%	0.9300	-	-	-	0.9400	0.9200
		DT	93.470%	0.9700	-	-	-	0.9300	0.9400
		Optimized RF	97.620%	0.6100	-	-	-	0.9800	0.9700
5.	**Sindhu *et al.* [24]**	RF	97.369%	-	-	-	-	-	-
		SVM	97.451%	-	-	-	-	-	-
		NN with Back Prop.	97.259%	-	-	-	-	-	-
6.	**Aminu *et al.* [25]**	RF	95.650%	-	-	0.9611	-	0.9442	0.9785
		XGBoost	97.130%	-	-	0.9709	-	0.9713	0.9773
		PNN	96.790%	-	-	0.9714	-	0.9640	0.9789
		RF-XGBoost	97.260%	-	-	0.9721	-	0.9726	0.9789
7.	**Tabassum *et al.* [26]**	XGBoost	96.850%	-	-	0.9700	-	0.9700	0.9700
		RF	97.10%	-	-	0.9700	-	0.9700	0.9700
		LR	92.66%	-	-	0.9300	-	0.9200	0.9200
		SVM	92.73%	-	-	0.9300	-	0.9300	0.9300
		SVM, DT, RF & XGBoost (Top 23 F)	98.280%	-	-	0.9800	-	0.9800	0.9800
8.	**Prayogo *et al.*[27] (SMOTE)**	kNN (SMOTE)	97.469%	-	-	0.9750	-	0.9750	0.9750
		kNN (without SMOTE)	97.178%	-	-	0.9720	-	0.9720	0.9720
9.	**Prince *et al.* [28]**	RF (Top 32 Feat.)	98.360%	-	-	-	-	-	-
		SVM (Top 44 Feat.)	94.01%	-	-	-	-	-	-
		PDCFS (48 Feat.)	98.240%	-	-		-	-	-
10.	**Ahmad *et al.* [29]**	DT	95.760%	-	-	0.9577	-	0.9546	0.9609
		DT with WFS	98.800%		-	0.9793	-	0.9804	0.9783
		DT with CFS	97.280%	-	-	0.9729	-	0.9698	0.9761
11.	**Pristyanto *et al.*[30] (OSS+SMOTE)**	SVM [DS1]	95.360%-	-	-	-	0.9536	-	-
		DT [DS1]	95.810%	-	-	-	0.9581	-	-
		ANN [DS1]	96.260%	-	-	-	0.9626	-	-
		kNN [DS1]	93.100%	-	-	-	0.9310	-	-
		SVM [DS2]	91.080%-	-	-	-	0.9328	-	-
		DT [DS2]	92.870%	-	-	-	0.9464	-	-
		ANN [DS2]	90.220%	-	-	-	0.9263	-	-
		kNN [DS2]	90.300%	-	-	-	0.9269	-	-

REFERENCES

[1] H. He, and E. A. Garcia, "Learning from Imbalanced Data," IEEE Transactions On Knowledge And Data Engineering, vol. 21, no. 9, Sep. 2009. [Online]. Available: 10.1109/TKDE.2008.239

[2] S. M. A. Elrahman, and A. Abraham, "A Review of Class Imbalance Problem," Journal of Network and Innovative Computing, vol. 1 , pp. 332-340, ISSN 2160-2174, 2013. [Online]. Available: www.mirlabs.net/jnic/index.html

[3] H. Guan, Y. Zhang, M. Xian, H. D. Cheng, and Xianglong Tang, "SMOTE-WENN: Solving class imbalance and small sample problems by oversampling and distance scaling," Applied Intelligence, Springer, 51, 1394–1409, 2021. [Online]. Available: https://doi.org/10.1007/s10489-020-01852-8

[4] F. Provost, "Machine Learning from Imbalanced Data Sets 101," AAAI Technical Report, WS-00-05, 2000. [Online]. Available: https://www.aaai.org/Papers/Workshops/2000/WS-00-05/WS00-05-001.pdf

[5] I. Letteri, A. D. Cecco, A. Dyoub, and G. D. Penna "A Novel Resampling Technique for Imbalanced Dataset Optimization," Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, Italy, 30 Dec 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2012.15231

[6] N. V. Chawla, N. Japkowicz, and A. Kolcz "Editorial: Special Issue on Learning from Imbalanced Data Sets," ACM SIGKDD Explorations Newsletter, Volume 6, Issue 1, pp 1–6, June 2004. [Online]. Available: https://doi.org/10.1145/1007730.1007733

[7] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002. [Online]. Available: https://doi.org/10.1613/jair.953

[8] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-2, No. 3, pp. 431– 433, July 1972. [Online]. Available:https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4309137

[9] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," ACM Sigkdd Explorations Newsletter, vol. 6 , Issue 1, pp. 20-29, 2004. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/1007730.1007735

[10] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Phishing Websites Features," School of Computing and Engineering, University of Huddersfield, 2015. [Online]. Available: Index of /ml/machine-learning-databases/00327 (uci.edu)

[11] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," Telecommunication Systems, Springer, 76, pp. 139–154 , 2021. [Online]. Available: https://doi.org/10.1007/s11235-020-00733-2

[12] R. Alabdan, "Phishing Attacks Survey: Types, Vectors, and Technical Approaches," Future Internet, 12, 168, 2020. [Online]. Available: https://doi.org/10.3390/fi12100168

[13] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing Attacks: A Recent Comprehensive Study and a New Anatomy," Frontier Computer Science, 30 September 2020. [Online]. Available: https://doi.org/10.3389/fcomp.2021.563060

[14] J. M. Johnson, and T. M. Khoshgoftaar "Survey on deep learning with class Imbalance," Journal of Big Data, vol. 6, no. 27, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0192-5

[15] M. Kubat, S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," In Proc. of 14th International Conference on Machine Learning (ICML), vol. 97, pp. 179-186, 1997. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.4487

[16] C. Seiffert, T. M. Khoshgoftaar, and J. V. Hulse, "Hybrid Sampling for Imbalanced Data," IEEE, July 13-15, 978-1-4244-2660-7/08/$25.00, 2008. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4583030

[17] C. Elkan, "The Foundations of Cost-Sensitive Learning," Proc. Int'l Joint Conf. Artificial Intelligence, pp. 973-978, 2001. [Online]. Available: https://dl.acm.org/doi/10.5555/1642194.1642224

[18] F. Provost, T. Fawcett, "Robust Classification for Imprecise Environments", Machine Learning, 42, 203–231, 2001. https://link.springer.com/content/pdf/10.1023/A:1007601015854.pdf

[19] M. Lv, Y. Ren, and Y. Chen, "Research on imbalanced data based on SMOTE-AdaBoost algorithm," IEEE, 978-1-7281-3584-7/19, 2019. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9094859

[20] P. Branco, L. Torgo, and R. P. Ribeiro, "A Survey of Predictive Modelling under Imbalanced Distributions," arXiv:1505.01658, 2015. [Online]. Available: https://doi.org/10.48550/arXiv.1505.01658

[21] Y A. Sagnik, and A. K. Kar "Phishing website detection using support vector machines and nature-inspired optimization algorithms," Telecommunication Systems, vol. 76, pp. 17–32, 2021. [Online]. Available: https://doi.org/10.1007/s11235-020-00739-w

[22] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun and A. K. Alazzawi, "AI Meta-Learners and Extra-Trees Algorithm for the Detection of Phishing Websites," IEEE Access , vol. 8, Aug. 2020. [Online]. Available: 10.1109/ACCESS.2020.3013699

[23] T. Bikku, M. Nikitha, A. Vajja, and K. Harshitha, "Optimized Machine Learning Algorithm to classify Phishing Websites," International Conference on Electronics and Renewable Systems (ICEARS), IEEE, 2022. [Online]. Available: 10.1109/ICEARS53579.2022.9752223

[24] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman, S. A. N. "Phishing Detection using Random Forest, SVM and Neural Network with Backpropagation," IEEE, 978-1-7281-7213-2/20/$31.00 , 2020. [Online]. Available:https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9277256

[25] A. A. Aminu, A. Abdulkarim, A. Y. Aliyu, M. Aliyu, and A. M. Turaki, "Detection of Phishing Websites Using Random Forest and XGBOOST," International Journal of Pure and Applied Sciences ISSN: 2635-3393, Vol. 2, Issue 3, Sept. 2019. [Online]. Available: http://www.smrpi.com/images/journals/IJPAS/20.pdf

[26] N. Tabassum, F. F. Neha, M. S. Hossain and H. S. Narman, "A Hybrid Machine Learning based Phishing Website Detection Technique through Dimensionality Reduction," IEEE Inter. Black Sea Conference on Communications and Networking (BlackSeaCom), 2021. [Online]. Available: 10.1109/BlackSeaCom52164.2021.9527806

[27] R. D. Prayogo, and S. A. Karimah, "Optimization of Phishing Website Classification Based on Synthetic Minority Oversampling Technique and Feature Selection," IEEE, 978-1-7281-9098-3/20, 2020. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9255562

[28] M. S. M. Prince, A. Hasan, and F. M. Shah, "A New Ensemble Model for Phishing Detection Based on Hybrid Cumulative Feature Selection," 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 987-1-6654-0038-2/21, 2021. [Online]. Available: 10.1109/ISCAIE51753.2021.9431782

[29] D. S. Ahmed, K. Q. Hussein, and H. A. A. A. Allah, "Phishing Websites Detection Model based on Decision Tree Algorithm and Best Feature Selection Method," Turkish Journal of Computer and Mathematics Education vol. 13, no. 01, pp. 100-107, 2022. [Online]. Available: https://www.turcomat.org/index.php/turkbilmat/article/view/11964

[30] Y. Pristyanto, and A. Dahlan, "Hybrid Resampling for Imbalanced Class Handling on Web Phishing Classification Dataset," 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), IEEE, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/9003803

[31] R. M. Mohammad, F. Thabtah, and L. McCluskey "Phishing Websites Features," 2015. [Online]. Available: http://eprints.hud.ac.uk/id/eprint/24330/6/MohammadPhishing14July2015.pdf

[32] R. M. Mohammad, F. Thabtah, and L. McCluskey, "An Assessment of Features Related to Phishing Websites using an Automated Technique," International Conference for Internet Technology and Secured Transactions (ICITST), IEEE, London, UK, pp. 492-497, ISBN 978-1-4673-5325-0, 2012. [Online]. Available: http://eprints.hud.ac.uk/id/eprint/16229/

[33] N. Thai-Nghe, Z. Ganter and L. Schmidt-Thieme, "Cost-Sensitive Learning Methods for Imbalanced Data," IEEE, 978-1-4244-8126-2/10, 2010. [Online]. Available: https://www.ismll.uni-hildesheim.de/pub/pdfs/Nguyen_et_al_IJCNN2010_CSL.pdf

[34] Dua, D. and Graff, C, UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California,

School of Information and Computer Science, 2019 [Online]. Available: UCI Machine Learning Repository: Phishing Websites Data Set.
[35] N. Abdelhamid, UCI Machine Learning Repository: Phishing Website Dataset, 2016. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Website%2BPhishing
[36] C. L. Tan, "Phishing dataset for machine learning: Feature Evaluation," 2018. [Online]. Available: https://data.mendeley.com/datasets/h3cgnj8hft/1R. M. Mohammad, F. Thabtah, and L. McCluskey , 2015. [Online]. Available: UCI Machine Learning Repository: Phishing Websites Data Set.
[37] N. Abdelhamid, UCI Machine Learning Repository: Phishing Website Dataset, 2016. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Website%2BPhishing
[38] C. L. Tan, "Phishing dataset for machine learning: Feature Evaluation," 2018. [Online]. Available: https://data.mendeley.com/datasets/h3cgnj8hft/1

J. SRIVASTAVA was born in Kanpur, INDIA in 1968. She earned her bachelor's degree, B.Sc. Computer Science from DELHI UNIVERSITY, New Delhi, INDIA in 1998. She received her two master's degrees, i.e., Master of Computer Applications (MCA) from BANASTHALI VIDYAPITH, Rajasthan, INDIA in 1991, and M. Tech. Computer Applications from INDIAN INSTITUTE OF TECHNOLOGY DELHI, New Delhi, INDIA in 2002. The author is presently pursuing Ph.D. from the School of Computer & Systems Sciences (SC & SS), JAWAHARLAL NEHRU UNIVERSITY, New Delhi, India under the mentorship of Dr. Aditi Sharan, Associate Professor, Jawaharlal Nehru University, New Delhi, India.

From 1994-2004, she worked in National Informatics Centre (NIC), New Delhi, INDIA and left NIC as Principal Systems Analyst in 2004 to join Indian Institute of Technology Delhi (IIT Delhi), New Delhi, INDIA. Currently the author is working as System Architect in Computer Services Centre, INDIAN INSTITUTE OF TECHNOLOGY DELHI (IIT Delhi), New Delhi, INDIA.

A. Sharan was born in INDIA. She earned her B.Sc. degree from Sukhadia University, Udaipur, INDIA in 1988. She received her master's degree in M.Sc. Computer Science from BANASTHALI VIDYAPITH, Rajasthan, INDIA in 1990. She pursued her Ph. D degree from Jai Narain University, Jodhpur, INDIA in 1996. She joined as Assistant Professor in 2004 Jawaharlal Nehru University (JNU) New Delhi, INDIA. She is currently working as Associate Professor, School of Computer & Systems Sciences (SC & SS), Jawaharlal Nehru University (JNU), New Delhi, INDIA.

She is actively involved in research for last 20 years. Her research interest includes: Machine Learning, Natural Language Processing, Information Retrieval and Extraction, Sentiment Analysis, Ontologies and their applications and other related fields. She has supervised around 20 Ph.D. and more that 30 M. Tech. students. She has several publications in reputed journals and presented papers in various National and International Conferences in INDIA and abroad. She has delivered 'Invited Talks' in many Institutes of repute in INDIA and abroad.