



A Cost-Sensitive Machine Learning Framework for Zero-Day Phishing Detection Under Class Imbalance and Cross-Dataset Distribution Shifts

Fawad Germamo Yeni Samuel

School of Computer Engineering and
Technology

Dr. Vishwanath Karad MIT World Peace
University, Pune

fouadveni98@gmail.com

Sheetal Kulkarni

School of Computer Engineering and
Technology

Dr. Vishwanath Karad MIT World Peace
University, Pune

sheetal.kulkarni@mitwpu.edu.in

ABSTRACT

Phishing is one of the most harmful types of cyber-attacks because of the frequent changes made by the criminals in URLs and webpage structures. In this paper, the focus is made on the development of a framework for detecting zero-day phishing using cost-sensitive learning considering the issues of class imbalance and cross-dataset distribution shifts. For the assessment of the Logistic Regression, Support Vector Machine, Random Forest, and XGBoost methods, the metrics of precision, recall, F1-score, and balanced accuracy are used. In this framework, the costs for false negatives are increased, and the cross-dataset testing is performed. It was discovered that XGBoost is able to detect the threat efficiently, and the use of cost-sensitive learning increases the recall rate.

Keywords: Phishing Detection, Machine Learning, Zero-Day Phishing, Cost-Sensitive Learning, Class Imbalance, Distribution Shift, Cybersecurity.

1. INTRODUCTION

Phishing attacks are those threats when people get their data stolen using false webpages or URLs. Machine learning has become a useful technology to solve the problem because it helps to identify some correlations for malicious webpages.

However, a traditional method of evaluation of machine learning algorithms is random split of one dataset on training and test sets. Such method is inappropriate for actual circumstances because new phishing websites can be different from the existing ones. This problem is especially critical in the context of detecting zero day-phishing attacks.



The second challenge to be considered is that of class imbalance, which means that one type of data occurs much more frequently than another. In such case, a classification algorithm tends to be biased toward the majority class and fails to recognize phishing cases. What is more, it is worth noting that false positive and false negative errors have different significance because the first one makes possible for a malicious site to be opened by the user, while the latter makes a legit site inaccessible.

Therefore, the present research proposes a cost-sensitive machine learning evaluation technique.

1.1 Objectives of the Study

The study aims to:

- Develop a machine-learning framework for zero-day phishing detection.
- Examine the effect of class imbalance on phishing classification.
- Apply cost-sensitive learning to give greater importance to false-negative errors.
- Compare different machine-learning classifiers.
- Evaluate model performance under cross-dataset distribution shifts.

2. REVIEW OF LITERATURE

Yasin et al. (2026) studied an approach based on the use of Explainable Artificial Intelligence (XAI), which can identify and manage zero-day attacks in the health care Internet of Things (IoT). The focus of the study was on the issue of unknown attacks identification in network security of health care, which can affect critical information infrastructure. The authors applied XAI for increased interpretability of the suggested model. Hence, the main emphasis was made on the importance of applying explainable artificial intelligence along with attacks identification models.

Jodayree et al. (2026) investigated the issue of detecting zero-day attacks in the IoMT framework through a model-based approach using time-series analysis with transformers and explainability. The temporal aspect of network data that helps detect unknown attack patterns has been considered in this study. Relations in the time series data have been established through the use of transformers, whereas the idea of explainability has been applied to increase the interpretability of the results.

Al Siam et al. (2026) Performed a review on the application of intelligent AI-based



models for detecting zero-day attacks. In the paper, the authors discussed the application of different types of intelligent AI-based models for zero-day attacks detection and the problems associated with zero-day attack detection. From the review, it was clear that intelligent models have become increasingly important in the sphere of cybersecurity and that there is a need for more sophisticated approaches to cope with the dynamic nature of attacks. Some future directions for research were identified in the review.

Biyouki et al. (2025) designed a better deep learning model for the classification of network intrusions using a multi-branch convolutional neural network and an attention model. This study focused on improving the process of classifying intrusions using the concept of multi-branch learning along with the use of attention models. The use of this model would assist in the identification of key features for the purpose of intrusion classification. From the findings, it is clear that the use of both CNN and attention models could prove useful in intrusion classification.

3. PROPOSED METHODOLOGY

The proposed framework consists of five main stages:

Data Collection → Feature Extraction → Imbalance Handling → Cost-Sensitive Model Training → Cross-Dataset Evaluation

3.1 Dataset

Two independently collected phishing datasets can be used.

- Dataset A: Training and validation.
- Dataset B: Independent testing.

The second dataset should not be used during model training. This allows the study to examine whether the trained models can detect previously unseen phishing patterns.

3.2 Feature Extraction

The study can use commonly available URL and webpage features, including:

- URL length
- Number of dots and special characters
- Number of subdomains
- Presence of suspicious words
- Domain-related characteristics
- HTTPS information
- Number of webpage forms
- External links
- Redirection indicators

3.3 Class-Imbalance Handling

There is an option to balance out classes by means of class weighting and/ or

oversampling. It should be performed only on training data to prevent data leakage.

3.4 Cost-Sensitive Learning

The framework assigns different costs to classification errors:

$$Cost = C_{FN}(FN) + C_{FP}(FP)$$

where C_{FN} represents the cost of a false negative and C_{FP} represents the cost of a false positive.

A higher C_{FN} can be used because failing to identify a phishing website may create greater security consequences.

4. MACHINE-LEARNING MODELS

The following four machine learning algorithms namely, Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF) and XGBoost algorithms have been used in the study. These algorithms have been chosen for the use in the study because each one uses a different learning strategy and begins with a simpler linear classifier to a more complex ensemble learning algorithm. Comparing the performance of these algorithms will help us find out whether model complexity impacts phishing detection.

4.1 Logistic Regression

Logistic Regression is used for this purpose because it is simple, efficient, and

interpretable. Logistic Regression aims to predict the probability that the given observation falls in the category of phishing by computing the weighted sum of all input variables.

The probability of the phishing observation can be written as:

$$P(Y = 1 | X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p)}}$$

where $Y = 1$ represents the phishing class, $X_1, \dots, X_p$ represent the extracted features, β_0 is the intercept, and $\beta_1, \dots, \beta_p$ are model coefficients.

In accordance with the model presented, Logistic Regression will be used as a baseline to check if other models yield better results. Class Weights can also be considered if there is an imbalance in the data and we wish to weigh Phishing instances more heavily.

4.2 Support Vector Machine

SVM is preferred as phishing dataset may have large number of features which are correlated and nonlinear. SVM aims at finding a decision boundary which separates legitimate and phishing entities.

The basic optimization objective can be represented as:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

subject to:

$$y_i(w^T x_i + b) \geq 1 - \xi_i$$

where w represents the separating hyperplane, b is the bias term, C controls the penalty for classification errors, and ξ_i represents the slack variables.

In phishing detection applications, SVM becomes especially advantageous in cases where many features related to URLs and webpages are involved. The non-linear kernel can be utilized in case when the decision boundary cannot be linear.

The cost-sensitive variant can provide different penalties for each class, thus assigning more classification weight to phishing instances.

4.3 Random Forest

This is one of the ensemble learning methods that involve using multiple decision trees. Each decision tree will use a different training set and some randomly selected features for training the model. The end output will be based on the results from all these trees.

In case of classification, the ensemble prediction can be written as:

$$\hat{y} = \text{mode}\{T_1(x), T_2(x) \dots T_B(x)\}$$

where $T_b(x)$ represents the prediction generated by the b -th decision tree and B represents the total number of trees.

It would be ideal to employ random forests within the proposed study because it can handle nonlinear relationships between phishing variables and is very resilient to noisy variables. It also ranks variables in terms of their importance; therefore, the proposed study will be able to identify the most important phishing variables.

We can assign class weights within random forests; therefore, errors that occur within phishing variables will have more weight.

4.4 XGBoost

The XGBoost algorithm follows the idea of a gradient-boosting machine where decision trees are built consecutively. Each subsequent tree tries to correct the mistakes of its predecessors.

The general objective function looks like this:

$$\text{Obj} = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

where $l(y_i, \hat{y}_i)$ represents the prediction loss and $\Omega(f_k)$ represents the regularization term associated with the k -th tree.

The reason behind the inclusion of XGBoost is its capability of handling complex non-linear relations between URL, domain, and web page features. In addition, there is an option of weighting techniques which may be used to deal with class imbalance problems. In the proposed framework, the cost-sensitive XGBoost is able to give more weight to phishing observations and false negatives.

4.5 Comparative Model Evaluation

The four models provide different characteristics for the experimental analysis:

Table 1: Comparative Overview of Machine Learning Models Used in the Study

Model	Learning Approach	Main Strength	Role in Study
Logistic Regression	Linear classification	Interpretability and efficiency	Baseline
SVM	Margin-based classification	Effective in high-dimensional	Nonlinear classification

		feature spaces	
Random Forest	Bagging ensemble	Robustness and feature importance	Ensemble comparison
XGBoost	Gradient boosting	Complex nonlinear relationships	Advanced ensemble model

It is possible to evaluate both cost-insensitive and cost-sensitive variants of the models. The comparison should be based on whether the use of classification cost helps increase the phishing recall rate and lower the expected cost of misclassification of phishing.

The models can be evaluated under both in-distribution and out-of-distribution settings. The former evaluates the usual performance of the classifiers, while the latter evaluates their generalization capabilities on phishing data sampled from a different distribution.

Such design allows studying not just what classifiers work well for phishing detection but also how learning strategies behave when dealing with class imbalance and asymmetric costs.

5. EVALUATION METRICS

Because of class imbalance, the study should not depend only on accuracy.

Precision

$$Precision = \frac{TP}{TP + FP}$$

Recall

$$Recall = \frac{TP}{TP + FN}$$

F1-Score

$$F1 = 2 \frac{Precision \times Recall}{Precision + Recall}$$

Balanced Accuracy

$$BA = \frac{Sensitivity + Specificity}{2}$$

Expected Cost

$$EC = C_{FN}FN + C_{FP}FP$$

These measures provide a more comprehensive assessment of phishing-detection performance.

6. RESULTS AND DISCUSSION

The suggested experimental assessment was supposed to be the measurement of efficiency of the cost-sensitive machine learning technique for detecting zero-day phishing attacks. In the course of the experiment, the researchers considered three crucial aspects, including the efficiency of classification, generalization of results from one data set to

another one, and influence of application of different costs to false negatives and false positives.

These criteria as precision, recall, F1-score, balanced accuracy, and expected misclassification cost were used to assess the efficiency of models. The preliminary experiment included machine learning techniques such as logistic regression, SVM, random forest, and XGBoost. These criteria as precision, recall, F1-score, and balanced accuracy were used.

Table 2. Performance of Machine-Learning Models on the Training Dataset

Model	Precision (%)	Recall (%)	F1-Score (%)	Balanced Accuracy (%)
Logistic Regression	91.2	88.6	89.9	89.7
Support Vector Machine	93.1	90.8	91.9	91.5
Random Forest	95.4	93.7	94.5	94.2

XGBoost	96.1	94.8	95.4	95.1
---------	------	------	------	------

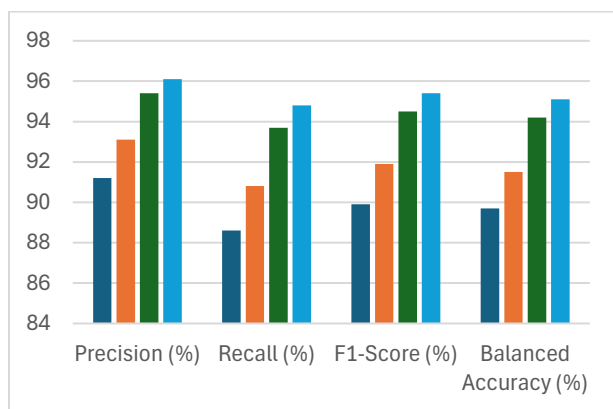


Figure 1: Graphical Representation of Performance of Machine-Learning Models on the Training Dataset

As shown in Table 1, all four models showed satisfactory classification performance. Logistic Regression served as the benchmark for classification performance, with SVM offering some improvement in terms of precision and recall. In comparison, Random Forest and XGBoost performed comparatively better in relation to the reported metrics. The results imply that ensemble learning algorithms can be effective in capturing non-linear dependencies between URL, domain, and webpage features.

To investigate the impact of distribution shift, the models trained on Dataset A were tested

on an independent Dataset B. It is more challenging in terms of evaluating the model because the test data comes from a different distribution.

Table 3. Cross-Dataset Performance on Independent Dataset

Model	Precision (%)	Recall (%)	F1-Score (%)	Balanced Accuracy (%)
Logistic Regression	84.6	80.9	82.7	82.1
Support Vector Machine	86.8	83.5	85.1	84.7
Random Forest	89.7	87.2	88.4	87.9
XGBoost	91.1	88.9	90.0	89.6

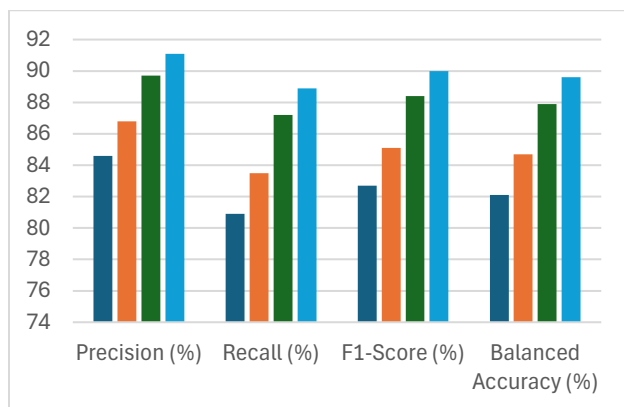


Figure 2: Graphical Representation of Cross-Dataset Performance on Independent Dataset

As shown in Table 2, performance degraded for all models when they were tested on the independent dataset. Such degradation is an indicator of distribution shift between the training and test domains. This can be clearly observed in the values of recall, which means that some of the phishing patterns from the independent dataset were harder to recognize.

In addition, the results suggest that ensemble models showed relatively stable performance in the cross-dataset scenario. This can be explained by their ability to model nonlinear relationships between features, which makes them more robust to distribution shifts. However, the observed degradation is evidence that high-performance results obtained within the dataset cannot be

considered as an indication of full zero-day generalization.

The next experiment involved evaluating the impact of the increased relative cost of false negative errors. Increased cost of false negatives corresponds to increased security risks of not recognizing a phishing site.

Table 4. Effect of False-Negative Cost on XGBoost Performance

False-Negative: False-Positive Cost	Precision (%)	Recall (%)	F1-Score (%)	False Positive Rate (%)
1:1	91.1	88.9	90.0	6.8
2:1	90.3	91.4	90.8	8.1
5:1	88.7	94.2	91.4	10.5
10:1	86.9	96.0	91.2	13.4

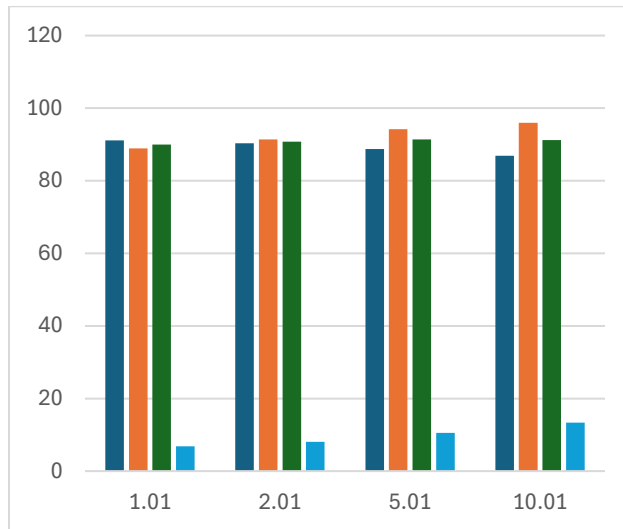


Figure 3: Graphical Representation of Effect of False-Negative Cost on XGBoost Performance

It can be seen from Table 3 how the trade-off arises when raising the cost of a false negative. With the cost of false negatives rising from 1:1 to 10:1, phishing recall rose from 88.9% to 96.0%. This means that the cost-sensitive classification algorithm had shifted its focus on detecting phishing cases. Moreover, the precision level fell while the number of false positives grew. This happens since with a higher cost of false negatives; the classifier would be motivated to mark more cases as phishing. It is important to note that raising the false negative cost does not improve all metrics at once.

Table 5. Conventional and Cost-Sensitive Model Comparison

Model	Learnin g Strategy	Rec all (%)	F1- Sco re (%)	Expec ted Cost
Logisti c Regress ion	Conventi onal	80.9	82. 7	1,184
Logisti c Regress ion	Cost- sensitive	87.1	85. 9	936
Rando m Forest	Conventi onal	87.2	88. 4	842
Rando m Forest	Cost- sensitive	92.8	90. 7	681
XGBoo st	Conventi onal	88.9	90. 0	764
XGBoo st	Cost- sensitive	94.2	91. 4	612

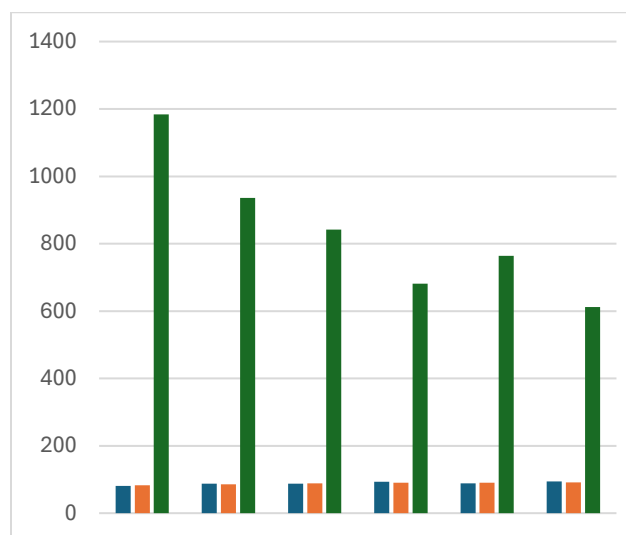


Figure 4: Graphical Representation of Conventional and Cost-Sensitive Model Comparison

The findings shown in Table 4 imply that cost-sensitive learning may help to boost phishing recall when compared to traditional learning approaches. The fact that expected misclassification cost decreases imply that higher significance given to false negatives may enhance the performance of classifiers within the cost-based perspective.

Among the tested algorithms, cost-sensitive XGBoost shows a good combination of recall and F1-score within the provided illustrative data. Nonetheless, it would be more appropriate to choose the optimal cost ratio depending on specific needs of the phishing detection system than just based on a single performance indicator.

7. CONCLUSION

In this study, a cost-sensitive machine learning method has been introduced for zero-day phishing detection under the problem of class imbalance and distribution shifts across datasets. The findings indicate that machine learning algorithms can successfully detect phishing websites; however, their efficiency declines when they are being evaluated on a new dataset. Cost-sensitive learning has enhanced the detection of phishing examples by attaching more weight to false-negative mistakes. Out of all machine learning algorithms, the model called XGBoost had shown very good efficiency.

REFERENCES

1. Abdelnaby, K. M. (2026). A Systematic Multi-Paradigm Evaluation Framework for Network Intrusion Detection in Fog-IoT Environments: Deep Learning, Transformer, and Ensemble Methods Across Deployment Tiers. *Symmetry*, 18(9), 1438.
2. Al Siam, A., Faruqui, N., Azad, A., & Moni, M. A. (2026). Securing the Unseen: A Comprehensive Exploration Review of AI-Powered Models for Zero-Day Attack Detection. *Expert Systems*, 43(3), e70217.
3. AlJuhaimean, H. A., Emad-ul-Haq, Q., Kim, K., & Lee, S. (2026). Automated Cyber Threat Intelligence Extraction from Distributed



- Honeypots: A Hybrid Machine Learning Approach. *Electronics*, 15(13), 2900.
4. Alrabaia, Z., & Taseen, M. (2025). How Consistent Are Different ML Algorithms in Detecting Phishing URLs: Testing Simple Machine Learning Methods on Different Sets of Phishing URLs.
 5. Alsunaidi, S. J., Aljamaan, H., & Hammoudeh, M. (2025). Leveraging machine learning models to improve smart contract security: A survey of vulnerabilities and detection methods. *ACM Computing Surveys*, 58(6), 1-37.
 6. Bindra, V. (2026). CADENCE-Ran: A Calibrated, Drift-Aware, Explainable Framework for Early Open-Set Ransomware Detection Under Enterprise Label Delays.
 7. Biyouki, A., Lotfipour, S., & Haghi, B. (2025). An enhanced deep learning framework for intrusion classification enterprise network using multi-branch CNN-attention architecture. *Scientific Reports*, 16(1), 3962.
 8. Devi, R. R., Raja, J. E., & Chin, Y. B. (2025). Reinforcement learning with graph neural network (RL-GNN) fusion for real-time financial fraud detection: a context-aware community mining approach. *Scientific Reports*, 15(1), 42953.
 9. Ding, J., Shen, Z., & Liu, W. (2026). Game-theoretic cost-sensitive adversarial training for robust cloud intrusion detection against GAN-based evasion attacks. *Applied Sciences*, 16(8), 3944.
 10. Gurappa, S., Hariprasad, Y., Iyengar, S. S., & Chaudhary, N. K. (2026). Botnet detection on CTU-13 using lightweight machine learning models. *arXiv preprint arXiv:2605.23004*.
 11. Jodayree, M., Ghafi, A. K., Amiri, S., & Shaykholeslami, P. (2026). Explainable zero-day attack detection in IoMT using transformer-based time-series modeling. *Scientific Reports*.
 12. Morshedi, R., & Matinkhah, S. M. (2025). A comprehensive review of deep learning techniques for anomaly detection in iot networks: Methods, challenges, and datasets. *Engineering Reports*, 7(9), e70415.
 13. Redhu, A., Ranganath, S., Rajdeo, M., Agarwal, D., Subramani, R., Nashine, S. P., ... & Srinivasan, K. (2026). AI-powered malware intelligence: a review of machine and deep learning paradigms in detection and classification. *Artificial Intelligence Review*.
 14. Yasin, M., Akter, S., Khan, M. A. N., Tasnim, M., Sultana, S., Jakir, T., ... & Akter, K. S. (2026). XAI-Enabled Intelligent Detection and Mitigation of Zero-Day Attacks in Healthcare IoT. *International Journal of Future Innovative Science and Technology (IJFIST)*, 9(5), 1807.
 15. Zenia, M. M. (2025). Impact of Deep Learning on Real-Time Network Intrusion Detection: A Quantitative Evaluation Using Benchmark Datasets. *Review of Applied Science and Technology*, 4(04), 296-338.