IRJCS: International Research Journal of Computer Science
Volume 13, Issue 05, May 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1305.01

Density-Aware Cost-Sensitive Learning for Enhanced Detection of Imbalanced Web-Based Intrusions

Amrendra Kumar Sharma*
Department of Computer Application,
Chhatrapati Shahu Ji Maharaj University, Kanpur, India
aks.malviyan@gmail.com
https://orcid.org/0009-0000-5616-5179
Mamta Tiwari
Department of Computer Application,
Chhatrapati Shahu Ji Maharaj University, Kanpur, India
mamtatiwari@csjmu.ac.in
https://orcid.org/0000-0002-5217-4841

Publication History
Manuscript Reference No: IRJCS/RS/Vol.13/Issue05/MYCS10080
Research Article | Open Access | Double-Blind Peer-Reviewed| Article ID: IRJCS/RS/Vol.13/Issue05/CSMY26.MYCS10080
Received:02,April 2025, Revised: 18, April 2026, Accepted: 30,April 2026,Published Online: 10, May 2026.
https://www.irjcs.com/volumes/Vol13/iss-05/01.MYCS10080.pdf
ArticleCitation:Amrendra,Mamta(2026),Density-Aware Cost-SensitiveLearningforEnhanced Detection of Imbalanced Web-Based, IRJCS: International Research Journal of Computer Science, Volume 13, Issue 04 of 2026 pages 562-569
Doi:> https://doi.org/10.26562/irjcs.2026.v1305.01
BibTeX Key: Amrendra@2026Density-Aware
IRJCS papers should be cited as IRJCS (International Research Journal of Computer Science, AM Publications, India 2026, ISSN 2393-9842, https://doi.org/10.26562/irjcs.2026.v1305.01 The journal's official abbreviation is IRJCS.
Orcid: https://orcid.org/0009-0004-9398-7488

Abstract: In contemporary network environments, intrusion detection faces significant challenges due to highly skewed traffic distributions, where benign instances predominate while certain attack types occur infrequently. This imbalance often leads to biased learning, resulting in poor detection performance for rare but crucial attack types. To address this challenge, this paper presents a density-aware cost-sensitive learning framework for web-based intrusion detection. Initially, logarithmic class weights are employed to mitigate majority class dominance. Subsequently, a k-nearest neighbor (k-NN)-based density estimation is utilized to capture data sparsity enabling density-based adaptive sample weighting that emphasizes both minority and hard-to-learn instances. Furthermore, a hard sample amplification mechanism is introduced to amplify misclassified samples to enhance the model's ability to learn complex patterns. Experimental results demonstrate that the proposed approach significantly improves minority class detection, achieving a G-Mean of 91.95% with Histogram Gradient Boosting Model, while maintaining an overall accuracy of 99.81%. Additionally, an average minority gain of 3.8% for HGB and 3.5% for Random Forest highlights its effectiveness in handling severely underrepresented classes.

Keywords: Intrusion Detection, Class Imbalance, Cost-Sensitive Learning, Minority Class Detection, Density-Based Weighting

I. INTRODUCTION

With the growth of communication channels, networking devices and IoT applications the web attacks are also increasing tremendously, making web servers and applications more vulnerable [1][2]. Attackers continuously seek exploitable vulnerability, use obfuscation techniques to intrude into the system so that they can steal sensitive data, get personal benefits and perform unethical activity [3]. Moreover, widespread installation of third-party plugins and potentially compromised web extensions, often installed with minimal security verification, significantly increases the attack surface of web applications. Consequently, web-based attacks like cross cite scripting (XSS), SQL injection, DDoS and request forgery attacks have become increasingly prevalent across web services. As these attacks are often more destructive, fast-evolving and distributed in nature, traditional security mechanism becomes inadequate and ineffective in detecting and preventing them. Therefore, researchers have shifted towards machine-learning (ML) based security systems [4]. In this direction, Intrusion Detection System (IDS) [29][30] integrated with ML techniques have emerged as effective solutions in encountering web-based attacks. By leveraging statistical modeling and learning from historical data, ML methods are capable of identifying underlying patterns in network traffic, leading to higher accuracy in intrusion detection [5]. The development of IDS using ML techniques typically involves processing vast amount of heterogenous data collected from various sources. Although this heterogeneity enhances comprehensive learning of attack patterns, it simultaneously introduces the challenge of class imbalance where certain classes are significantly underrepresented compared to others with large number of instances [6]. This imbalance often leads to biased learning, resulting in poor detection, especially for rare attack classes like XSS and web-based attacks. To handle the class imbalance issue, several methods have been explored in the existing literature [7][8][9][10][31]. These methods attempt to balance the class distribution by altering the natural proportion of samples and therefore tend to overlook the intrinsic structure of the data, potentially causing to the inclusion of redundant samples [11].

Consequently, they fail to distinguish between actual and redundant samples within the same class. To overcome this, researchers have adopted class-weighted learning [11][12][13], where minority classes are assigned more weights than majority classes to improve their impact during training. However, the primary drawback of this approach, despite its effectiveness in handling minority classes, lies in assigning uniform importance to all samples within a class, which fails to capture granularity of local data distribution. In practice, both conventional and class-weighted methods tend to focus more on density populated areas due to the higher number of samples. As a result, rare but informative instances located in low-density regions may not be adequately learned, leading to suboptimal decision boundaries and poor detection of minority attack classes. Motivated by above limitations, this study proposes a density-aware cost-sensitive learning framework that integrates local density of data with class imbalance information.

The leading contribution of the proposed work are as follows-

- A novel density-aware weighting mechanism that incorporates local data distribution with class level imbalance into the leaning process to effectively address extreme class imbalance.
- An adaptive weighting strategy that combines density information with logarithmic class weights to emphasizes misclassified instances and improve the learning of difficult samples.
- A cost-sensitive learning paradigm in which samples are assigned adaptive weights based on class imbalance and local density, enabling improved intrusion detection performance under severely imbalanced conditions.
- A comprehensive evaluation on web-based attacks from CICDS2017 dataset is conducted using class-wise performance metrics, including minority class detection rate, False Positive Rate (FPR) and G-Mean, with comparisons against baseline and class-weighted approaches.

II. LITERATURE REVIEW

To handle class imbalance issue in intrusion dataset extensive work have been done and among them three common approaches have been implemented including data-driven, algorithm-level and hybrid methods. Data-driven methods are the resampling approaches where the instance proportion is altered within the dataset to make the class distribution more balanced. This is done through undersampling and oversampling applied during data pre-processing phase, making them independent of the ML classifiers [9][14][15]. For example, Synthetic Minority Oversampling Technique (SMOTE) [16] was applied using Principal Component Analysis (PCA) to achieve the highest accuracy of 95.1% on UNSW-NB15 dataset. Similarly, authors in [17] demonstrated the effectiveness of SMOTE by conducting experiments using five ML classifiers on CIDDS-001 dataset under both balanced and imbalanced conditions. Several variations of SMOTE, including FF-SMOTE [17], Borderline-SMOTE [18], Cure SMOTE [19] and FW-SMOTE [20] have developed to improve performance on imbalanced datasets. Abdelkhalek and Maggie [21] effectively employed both oversampling and undersampling techniques to address the class imbalance problem. Specifically, they utilized ADASYN and Tomek Links with deep learning methods, achieving a classification accuracy of 99.8%. Their approach also outperformed in multi-classification on NSL-KDD dataset. Likewise, the study in [22] both oversampling and undersampling simultaneously incorporated, where SMOTE and ADASYN, as oversampling, were introduced with Random Undersampler along with Instance Hardness Threshold, as undersampling methods in a unified framework to demonstrate that minority attacks can be effectively detected.

Seo an Kim [10] proposed SVR-SMOTE, a ML surrogate approach to optimize oversampling ratios for minority classes (R2L, U2R and Probe) in the highly imbalanced KDD Cup dataset, achieving strong detection gains with minimal computations. In [23], SMOTE was effectively utilized to oversample minority classes of web-based attacks, resulting in improved values. Some authors have merged sampling techniques with clustering to overcome the limitations of the sampling. For example, Tsai et al. [15] introduced Cluster-based Instance Selection (CBIS) technique, a novel undersampling cluster-based approach in which redundant majority class have been eliminated from imbalanced dataset and then subset of majority class is merged with minority class using instance selection. Their result has shown the improved discriminability. In contrast to data-driven approaches, algorithm-level methods handle class imbalance by modifying the learning process rather than altering the original dataset which can be implemented through, cost-sensitive and class-weighted learning, where misclassification cost (penalty) or higher importance is assigned to minority class instances during the model training. In this context, Telikani et al. [24] proposed a cost-sensitive stacked autoencoder (CSSAE) method, in which class-specific costs were assigned to KDD and NSL-KDD datasets (DoS, R2L, U2R and Probe). Subsequently, a stacked autoencoder was employed to learn the discriminative features that effectively distinguish between both binary and multi-classification. Similarly, a focal loss function was implemented in [13] to resolve the class imbalance issue by introducing a novel CBF-IDS framework. The proposed technique utilizes both CNN and BiLSTM networks to capture discriminative features. Additionally, focal loss down-weights the impact of majority samples to focus on hard minority samples on NSL-KDD, UNSW-NB15 and CIC-IDS2017 datasets, demonstrating high accuracy, precision and recall. The authors in [11] assigned dynamic weighted balanced loss to overcome the limitation of static weighting, where weights are assigned based on class frequency and the predicted probabilities. This self-adaptive mechanism enables the models to focus more effectively on difficult samples and was validated on CICIDS2017. Chkirbene et al. [12] proposed a weighted ML framework that dynamically optimizes class weights using Confusion Score Matix. The approach was experimented on NSL-KDD and UNSWNB15 to mitigate severe class imbalance in attack classification. Zhou et al. [9] introduced an imbalance-aware learning approach that integrates F1 score maximization with an improved boosting algorithm (NIBoost). The method adaptively updates samples weights to emphasize minority and hard instances to improve classification performance of imbalance datasets.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 05, May 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1305.01

Ensemble learning has also been explored under algorithm-level approaches, where methods like bagging and boosting combine multiple classifiers to improve better results on imbalanced data. A detailed study of it has been reviewed in [25]. In this context, Bedi et al. [14] proposed I-SiamIDS, a two-layer ensemble approach using b-XGBoost, Siamese Neural Network and DNN for multi-class attack detection, achieving improved performance on NSL-KDD and CIDDS-001. Hybrid methods integrate both data-driven and algorithm-level techniques to efficiently improve the classification performance of minority classes in imbalanced datasets. In this context, Salehi and Khedmati [26] proposed a novel cluster-based SMOTE Both-sampling (CSBBoost) algorithm where K-means is employed in cluster formation of minority and majority classes rather than artificially generating the data samples directly from SMOTE. The result outperformed over 20 benchmarked datasets in terms of precision and recall. Chawla et al. [27] proposed SMOTEBoost which combined SMOTE with boosting to improve minority class learning. Similarly, Wu et al. [28] proposed hybrid approach that combined SMOTE with K-means to obtain the balanced NSL-KDD dataset which improve the minority class representation. An enhanced Random Forest model was subsequently employed to result in better detection.

III. PROPOSED METHOD

The proposed work aims to effectively handle the class imbalance in web-based attack classes within CICIDS2017 dataset in order to improve overall accuracy and enhancing detection rate of minority classes. The overall workflow of the proposed work is as follows-

A. DATASETACQUISITION AND DESCRIPTION

CICIDS2017 dataset, released by Canadian Institute for Cyber security is used in this study. The dataset comprises contemporary cyber-attacks collected over 5 days and consists of 8 files. Since the proposed work focuses on web-based attacks, the "Thursday-Working Hours-Morning-Web Attacks.pcap_ISCX.csv" file is utilized for analysis. The data distribution of the dataset has been shown in the Fig. 1, where a significant class imbalance can be easily seen. The benign class has valid 168186 instances whereas a total of 2180 web attacks correspond to 3 attacks, namely Brute-Force, XSS and SQL injection with 1507, 652 and 21 instances, respectively. This extreme imbalance, particularly for SQL injection samples, poses a significant challenge for traditional ML models.

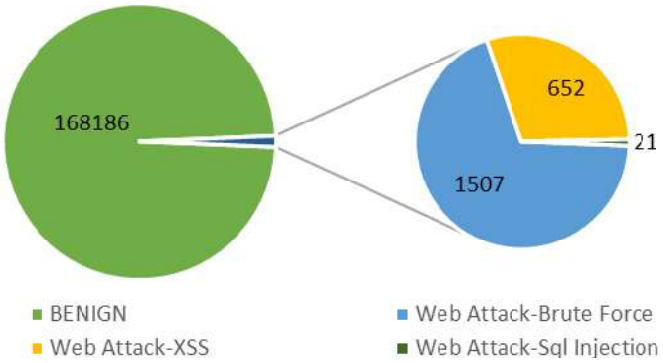

Fig.1 Data distribution of the dataset

B. PRE-PROCESSING

To ensure effective model training, it is essential to transform the raw data into structured and interpretable format that can be efficiently processed by ML algorithms. To do so data with string values have been converted into numeric values followed by deleting the redundant column and NaN values. This process makes the dataset with 84 columns with a total of 170366 instances. To prevent the dominance of features with larger magnitudes, data scaling is done using standard scalar. Since the dataset consists of 3 attack categories along with one benign class, label encoder is utilized to covert each into numerical form and treat them independently.

C. DENSITY-AWARE COST SENSITIVE LEARNING

To address the class-imbalance more effectively than relying on the tradition methods, a density-aware cost-sensitive learning is proposed, which consists of density estimation of data distribution followed by cost-sensitive weighting. Initially, the dataset is portioned into training and testing subsets using stratified sampling to preserve the original class distribution. After following preprocessing steps, k-nearest neighbour (k-NN) is applied on the training instance x_i for density estimation data samples as follows-

$$\bar{d}_i = \frac{1}{k}\sum_{j=1}^{k} d_{ij} \qquad \#(1)$$

where $d_{ij}$is the average distance between sample $x_i$and itsj^{th}nearest neighbour. The local density $\rho_i$of each sample is then estimated by taking the inverse of this average distance. The density values are subsequently normalized to [0,1] to ensure the consistency allowing the model to differentiate between dense (majority) and sparse (minority) regions. Next, to handle the class-imbalance, class-level weights are computed. If N_{train} denote the total number of training samples and n_c represents the number of samples belonging to class c, the ratioN_{train}/n_c, reflects the inverse level of each class

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 05, May 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1305.01

which assign higher values to minority classes and lower values to majority classes. A logarithmic scaling is applied to this ratio to prevent excessive large weights for extremely rare classes and accordingly, the class weight is defined as-

$$w_c = log\left(1 + \frac{N_{train}}{n_c}\right) \qquad \#(2)$$

The density information and class weights are then combined to assign adaptive sample weights to each training instance x_i as-.

$$w_i = (1 - \rho_i^{norm}) \cdot \left(w_{y_i}\right)^{\beta} \qquad \#(3)$$

where $\beta$is a tuning parameter controlling the influence of class imbalance, $\rho_i^{norm}$is the normalized density value of $\rho_i$while w_{y_i} is the class weight of class label y_i. The obtained weights from Eq. (3) are further normalized to ensure the numerical stability. Using these weights, an initial model is trained and predictions are made to identify the hard samples. To emphasize difficult instances, the weights of misclassifies samples are increased by γw_i, where $\gamma$is the hard sample amplification factor. Finally, the model is retained with these updated weights to obtain final model and evaluated on the test dataset for its performance. The overall workflow of the proposed framework has been illustrated in Fig. 2.

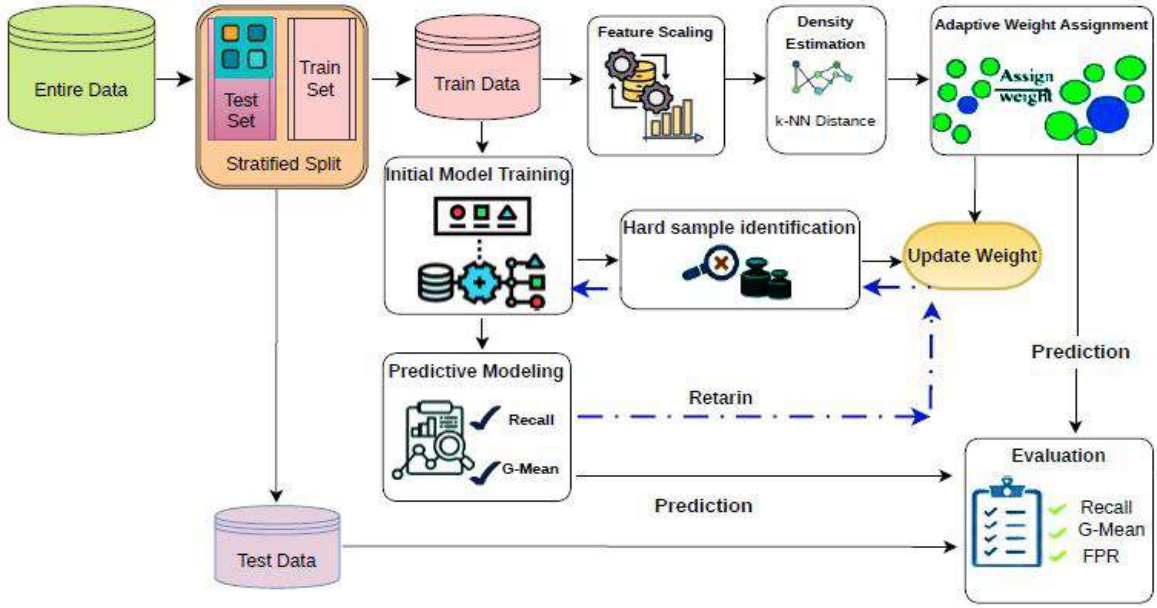

Fig.2 Work flow of the proposed framework

IV. EXPERIMENTS AND RESULTS

All the experiments in the proposed work were conducted on Windows 11 equipped with 12th Gen Intel Core i5 processor. Random Forest (RF) [32] and Histogram Gradient Boosting (HGB) models were successfully implemented and executed using Jupyter notebook with the help of scikit-learn library. The dataset was split into training and testing sets in a 70:30 ratio.

A. EVALUATIONMETRICS

The performance of ML models is typically evaluated using confusion matrix, consisting of true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Here, TP refer to correctly identified attack instances, TN represents correctly identified benign instances, FP denote benign instances incorrectly classified as attacks and FN correspond to attack instances that are incorrectly classified as benign.

The common metric, accuracy (ACC) is defined as follows-

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad \#(4)$$

When the dataset is imbalanced, accuracy may not be the more reliable performance measure, as it is biased toward the majority class. Therefore, additional metrics like precision, recall, F-score, false positive rate (FPR) and G-mean are also evaluated to obtain a more accurate performance of the model, particularly for minority class prediction.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \#(5)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \#(6)$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad \#(7)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \qquad \#(8)$$

$$\text{G} - \text{mean} = \sqrt{\text{Recall} \times \text{Speci icity}} \qquad \#(9)$$

G-mean is used to evaluate the balance between classification performance on both minority and majority classes and defines as the geometric mean of Recall and Specificity where $Specificity = 1 - FPR$.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 05, May 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1305.01

B. PARAMETER SETTING

There are several parameters that influence the performance of the model, as listed in the Table 1. In particular, the parameter β is empirically tuned in the range [1.0,1.5] and 1.2 was selected as it provides the best trade-off between minority recall and overall performance. However, hard amplification factor γ is model specific and its value is selected based on the stability and learning behaviour of the classifier.

Table 1 Parameter configuration of the model

Parameter	Description	Value
k	Default value of KNN	5
β	Class weight exponent	1.2
γ (RF)	amplification factor	1.5
γ (HGB)	amplification factor	1

C. RESULT

The class distribution of the dataset along with the computed class weights is illustrated in Fig. 3. It can be observed that the dataset exhibits severe imbalance, with the BENIGN class (168,186 samples) significantly dominating the attack categories, while minority classes such as SQL Injection contain only 21 instances. The proposed logarithmic weighting mechanism, as defined in Eq. (2), assigns higher importance to these rare classes, ensuring their increased contribution during model training without altering the original data distribution.

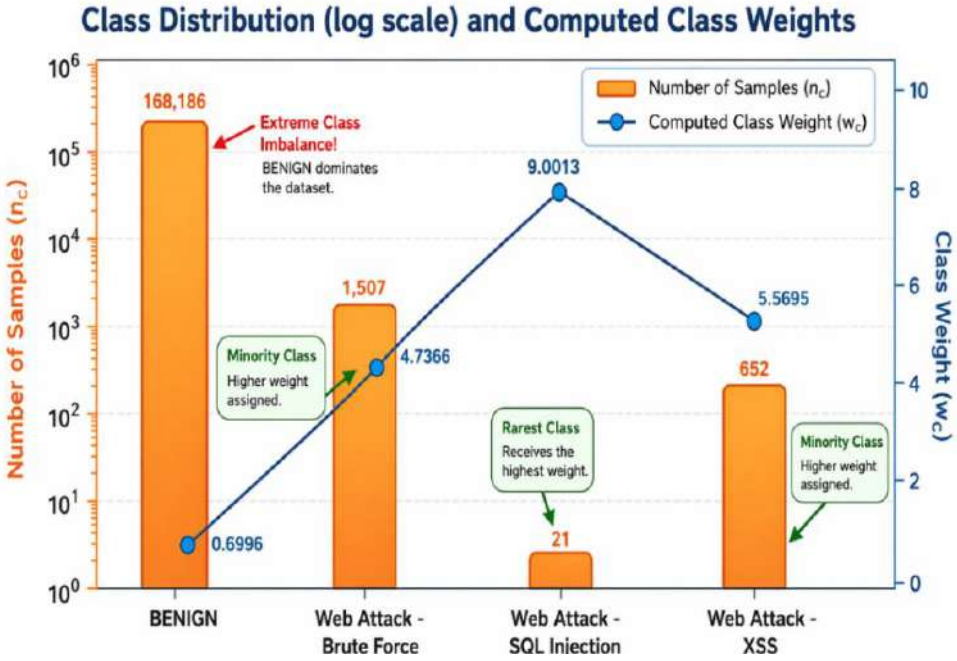

Fig. 3 Class distribution and corresponding class weights (log scale)

To further evaluate the effectiveness of the proposed model, the results are compared with the baseline models and class-balanced variants of RF and HGB. Table 2 represents the comparative performance of HGB for all configurations. While all model achieves similar accuracy due to class imbalance, the HGB classifier in proposed approach significantly improve recall from 87.01% (baseline) to 92.28% and f1 score from 88.88% (baseline) to 91.62%. However, precision of the class weight model of HGB is slightly more than the proposed work. Furthermore, the G-mean increases from 86.58 (baseline) to 91.95, demonstrating substantial enhancement in detection of minority classes.

Table 2 Comparative performance report of models with its variants

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1 score (Macro)	G-Mean	FPR (Benign class)
HGB (baseline)	99.79	91.13	87.01	88.88	86.58	0.0091
HGB (class weight)	99.81	91.36	88.16	89.48	87.88	0.0045
HGB (Proposed)	99.81	91.18	92.28	91.62	91.95	0.0045
RF (baseline)	99.72	89.53	73.27	78.64	69.69	0.0229
RF (class weight)	99.77	90.68	78.09	82.30	75.60	0.018
RF (Proposed)	99.78	90.97	77.26	82.0	75.01	0.019

For RF classifier, the proposed method improves performance compared to baseline model; however, its performance with its class-weighted variant, suggest that the impact of density aware weighting is more pronounced in boosting-based models which the lowest value of FPR in benign class further justifies.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 05, May 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1305.01

Table 3 Class-wise performance of the proposed method

Model	Class	Precision	Recall	F1 score	G-Mean
HGB (Proposed)	BENIGN	100	100	100	99.76
	Brute Force	92.0	87.0	89.0	92.97
	SQL injection	100	100	100	100
	XSS	72.0	83.0	77.0	90.85
RF (Proposed)	BENIGN	100	100	100	98.99
	Brute Force	88.0	89.0	88.0	81.64
	SQL injection	100	50.0	67.0	70.7
	XSS	76.0	70.0	73.0	79.19

Table 3 represents the class-wise performance of the proposed work for both the models. Both HGB and RF models achieved near perfect performance for BENIGN class. Despite being underrepresented, the SQL injection achieved 100% recall and F1 score for HGB model. Similarly, the Brute Force and XSS classes achieve recall values of 87% and 83%, respectively, with the high G-mean values indicating balanced classification performance for the same model. In contrast, RF model maintains strong performance for the majority class but exhibits comparatively lower detection capability for minority classes, particularly SQL injection and XSS, due to their limited representation in the dataset. Overall, the results confirms that the proposed approach significantly improves minority class detection, especially when combined with booting model like HGB.

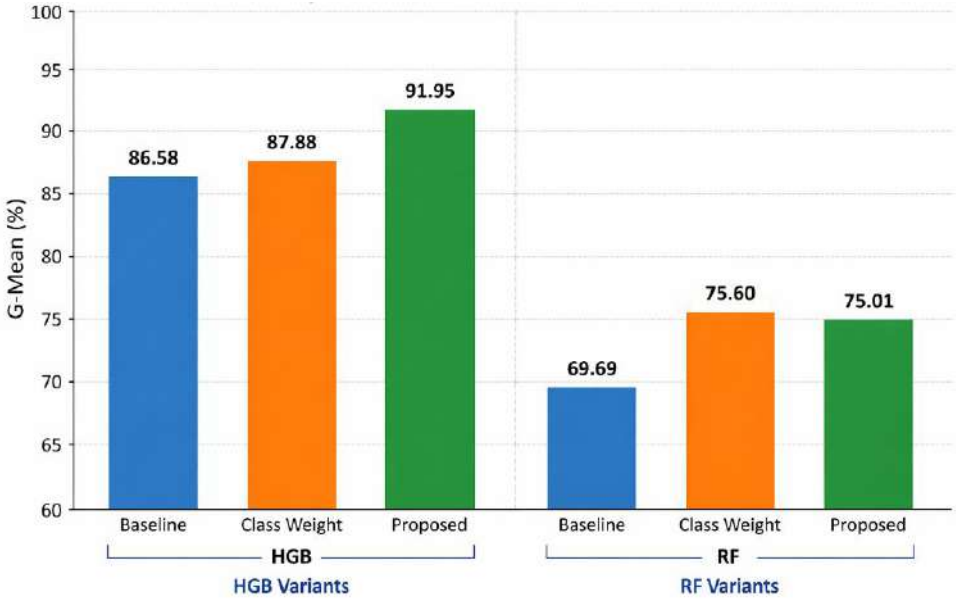

Fig. 4 G-Mean comparison of HGB and RF

Figure 4 illustrates a progressive improvement in G-Mean for HGB model from the baseline to the proposed approach, indicating enhanced balanced classification performance while RF model shows improvement over the baseline and achieves performance comparable to class-weighted variant, with only a marginal difference.

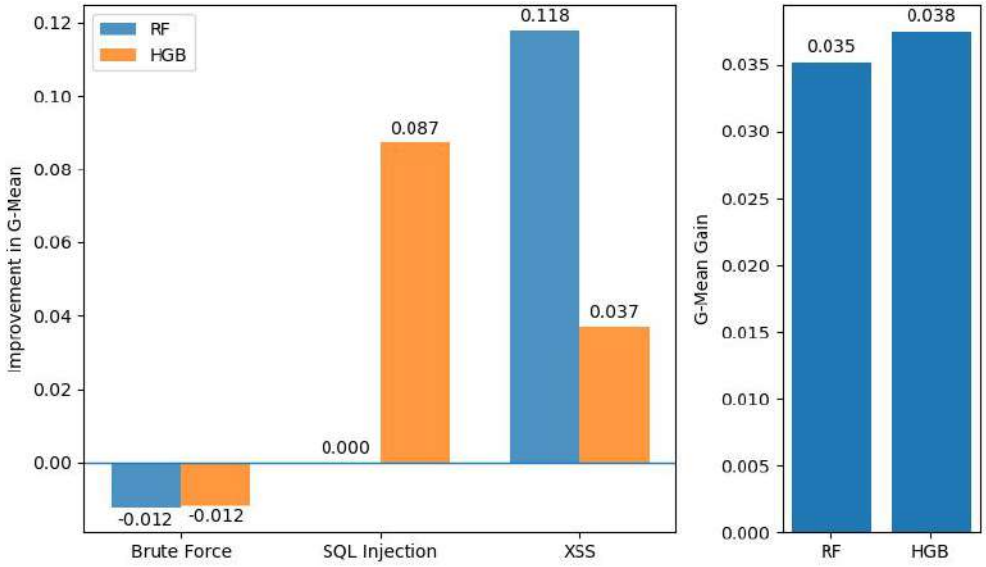

Fig. 5 Class-wise improvement in G-Mean and average minority gain

This indicates that the proposed strategy preserves balanced classification capability. A class-wise change in G-Mean for minority attacks classes along with average minority gain for both models have been illustrated in Fig. 5. It can be observed that both models exhibit notable improvements for highly underrepresented classes, particularly SQL Injection and XSS. The HGB model achieves a substantial gain of 0.087 for SQL Injection, while RF shows the highest improvement of 0.118 for the XSS class. Similarly, HGB also improves XSS detection with a gain of 0.037. A slight decrease in Brute Force class in both models occur which may be due to its comparatively less separable characteristics, especially when the model focuses more on distinct and rarer attack classes. Despite this minor trade-off, the overall minority gain remains positive, with HGB achieving 3.8% and RF achieving 3.5%, indicating consistent improvement in minority class performance.

V. CONCLUSIONS

This paper has presented a density-aware cost sensitive learning framework to address extreme class imbalance in web-based intrusion detection. By integrating logarithmic class weighting with k-NN-based density estimation, the approach captures both global imbalance and local data distribution characteristics. The adaptive weighting mechanism, enhanced through hard sample refinement, enables improved detection of sparse and difficult instances. Experimental results on the CICIDS2017 dataset show significant improvements in detection rate and G-Mean, particularly for highly underrepresented attacks such as SQL Injection and XSS, while maintaining stable accuracy and low FPR. The method is especially effective with boosting-based models, such as Histogram Gradient Boosting, demonstrating strong balanced classification performance. Overall, the findings highlight that incorporating density awareness into cost-sensitive learning provides a robust solution for imbalanced intrusion detection without relying on data resampling. Future work includes investigating the impact of data resampling on model performance, extending the proposed framework toward model explain ability and exploring its integration with deep learning models.

ACKNOWLEDGMENT

The author extends sincere thanks to Dr. Mamta Tiwari, Assistant Professor, Department of Computer Application, Chhatrapati Shahu Ji Maharaj University, Kanpur, for her consistent support throughout this research.

REFERENCES

1. S.Kumar, S.Gupta, and S.Arora, “Research Trends in Network-Based Intrusion Detection Systems: A Review,” IEEE Access, vol. 9, pp. 157761–157779, 2021, https://doi:/10.1109/ACCESS.2021.3129775
2. N.Agarwal, “A Closer Look at Intrusion Detection System for Web Applications,” vol. 2018, 2018, https://doi.org/10.1155/2018/9601357.
3. E.Chang and T.Dillon, “Provenance-based Intrusion Detection Systems: A Survey,” vol. 55, no. 7, 2026, https://doi.org/10.1145/3539605.
4. X.Yu, W.Yu, S.Li, X.Yang, Y.Chen, and H. Lu, “WEB DDoS Attack Detection Method Based on Semisupervised Learning,” vol. 2021, 2021, https://doi.org/10.1155/2021/9534016.
5. S.Promodya, T.Lasitha, J.Lasith, and Y.Pushpika, “Deep Neural Network Based Real Time Intrusion Detection System,” SN Comput. Sci., vol. 3, no. 2, pp. 1–12, 2022, https://doi.org/10.1007/s42979-022-01031-1.
6. A.A.Hagar,“Apache Spark and Deep Learning Models for High-Performance Network Intrusion Detection Using CSE-CIC-IDS2018,” vol. 2022, 2022, https://doi.org/10.1155/2022/3131153.
7. G.Haixiang, L.Yijing, J.Shang, G.Mingyun, and H.Yuanyue, “Learning from class-imbalanced data: Review of methods and applications,” Expert Syst. Appl., vol. 73, pp. 220–239, 2017, https://doi.org/10.1016/j.eswa.2016.12.035.
8. S.Susan, “The balancing trick: Optimized sampling of imbalanced datasets — A brief survey of the recent State of the Art,” no. September 2020, 2021, https://doi.org/10.1002/eng2.12298.
9. Q.Zhou, Y.Qi, H.Tang, and P.Wu, “Machine learning - based processing of unbalanced data sets for computer algorithms,” 2023.
10. J.Seo and Y.Kim, “Machine-Learning Approach to Optimize SMOTE Ratio in Class Imbalance Dataset for Intrusion Detection,” vol. 2018, 2018.
11. K.R. and C.P.T.Fernando, “Dynamically Weighted Balanced Loss: Class Imbalanced Learning & Confidence Calibration of Deep Neural Networks,” 2021.
12. Z.Chkirbene et al., “A Weighted Machine Learning-Based Attacks Classification to Alleviating Class Imbalance,” pp. 1–12, 2020.
13. H.Peng, C.Wu, and Y.Xiao, “CBF-IDS: Addressing Class Imbalance Using CNN-BiLSTM with Focal Loss in Network Intrusion Detection System,” 2023.
14. P.Bedi, N.Gupta, and V.Jindal, “I - SiamIDS : An Improved Siam - IDS for handling class imbalance in Network - based Intrusion Detection Systems”.
15. C.Tsai, W.Lin, Y.Hu, and G.Yao, “Under-Sampling Class Imbalanced Datasets by Combining Clustering Analysis and Instance Selection,” Inf. Sci. (Ny)., 2018, https://doi.org/10.1016/j.ins.2018.10.029.
16. A.Hameed and N.Z.Bawany, “Network intrusion detection using oversampling technique and machine learning algorithms,” 2022, https://doi.org/10.7717/peerj-cs.820.
17. R.Abdulhammed, M.Faezipour, A.Abuzneid, and A.AbuMallouh, “Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic,” IEEE sensors Lett., vol. 3, no. 1, pp. 1–4, 2018.

18. H.Han, W.Wang, and B.Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," pp. 878–887, 2005.
19. L.Ma and S.Fan, "CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests," pp. 1–18, 2017, https://doi.org/10.1186/s12859-017-1578-z.
20. A.Qaddos et al., "A novel intrusion detection framework for optimizing IoT security," Sci. Rep., pp. 1–22, 2024, https://doi.org/10.1038/s41598-024-72049-z.
21. A.Abdelkhalek and M. Mashaly, Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning, vol. 79, no. 10. Springer US, 2023. https://doi.org/10.1007/s11227-023-05073-x.
22. Z.Fan, S. Sohail, and F. Sabrina, "Sampling-Based Machine Learning Models for Intrusion Detection in Imbalanced Dataset," pp. 1–19, 2024.
23. G.Karatas, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," IEEE Access, vol. 8, pp. 32150–32162, 2020, https://doi.org/10.1109/ACCESS.2020.2973219.
24. A.Telikani and A.H.Gandomi, "Cost-sensitive stacked auto-encoders for intrusion detection in the Internet of Things," no. xxxx, 2025, https://doi.org/10.1016/j.iot.2019.100122.
25. M.Galar, A.Fern, E.Barrenechea, and H.Bustince, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches," vol. 42, no. 4, pp. 463–484, 2012.
26. A.R.Salehi and M.Khedmati, "A cluster based SMOTE both sampling (CSBBoost) ensemble algorithm for classifying imbalanced data," Sci. Rep., pp. 1–17, 2024, https://doi.org/10.1038/s41598-024-55598-1.
27. N.V.Chawla, A.Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost : Improving Prediction of the Minority Class in Boosting," pp. 1–10, 2003.
28. T.Wu,H.Fan,H.Zhu,C.You,H.Zhou, and X.Huang, "Intrusion detection system combined enhanced random forest with SMOTE algorithm," EURASIP J. Adv. Signal Process., vol. 6, 2022, https://doi.org/10.1186/s13634-022-00871-6
29. S.Kottilingal, "Deep Learning Based Network Intrusion Detection System: A Deep Abstract Networks (DANets) Model Approach," Int. Res. J. Comput. Sci., vol. 11, pp. 539–544, Jul. 2024, https://doi.org/10.26562/irjcs.2024.v1107.01.
30. A.R.Raj and S.Siddarama, "The False Positive Alert Reduction Using Data mining Techniques in Intrusion Detection System," Jan. 2016, https://doi.org/10.6084/M9.FIGSHARE.3490106.V1.
31. S.Gautam and R.Dey, "METHODS FOR CLASSIFICATION OF IMBALANCED DATA: A REVIEW," Int. Res. J. Comput. Sci., vol. 9, pp. 89–95, Apr. 2022, https://doi.org/10.26562/irjcs.2021.v0904.004.
32. P.S.R.,"BOTNET Detection in IoT Environments," Int. Res. J. Comput. Sci., vol. 13, pp. 24–28, Jan. 2026, https://doi.org/10.26562/irjcs.2026.v1301.05.