Enhancing Malware Detection with Advanced Balancing Techniques: Comparing HBBB and Zebin Method

Camila Vieira
Federal University of Pernambuco
Recife, Brazil
cbv2@cin.ufpe.br

Dayane Lira
Federal University of Pernambuco
Recife, Brazil
dls6@cin.ufpe.br

José Vinicius de S Souza
Federal University of Pernambuco
Recife, Brazil
jvss2@cin.ufpe.br

Abstract

As cyber threats evolve in complexity and volume, traditional manual detection methods are no longer sufficient. Machine learning (ML) techniques have become essential for malware detection, but challenges such as class imbalance can hinder their effectiveness. This study explores the impact of class imbalance on malware classification and evaluates advanced data-balancing strategies using two datasets: Drebin, for Android malware detection, and CIC-DoHBrw-2020, for DNS over HTTPS (DoH)-based attacks. We investigate two advanced balancing techniques: the Zebin method, which divides the majority class into subgroups and replicates the minority class before balancing, and the HBBB method, which balances each bootstrap in bagging individually, increasing data variability and classifier robustness. Additionally, we assess model explainability via SHAP and instance hardness analysis. Our results demonstrate that the HBBB method outperforms the Zebin approach in improving malware detection performance. Specifically, the HBBB technique achieves superior metrics, particularly in recall and MCC, for both the CIRA-CIC-DoHBrw-2020 and Drebin datasets. The HBBB method effectively addresses class imbalance while maintaining high performance across both the minority and majority classes, leading to a more robust and interpretable detection system.

I. Introduction

Check Point Software recently published the Global Threat Index for February 2025, highlighting the rise of AsyncRAT, a remote access Trojan (RAT) that continues to evolve as a significant threat in the global cybersecurity landscape [3]. This growing sophistication of cyberattacks underscores the urgent need to develop effective detection and mitigation strategies.

The volume and complexity of modern cyber threats have increased to a level where manual detection methods are no longer viable. Traditional static rule-based approaches struggle to keep up with the rapid evolution of malware, as new variants can easily evade signature-based defenses. As a result, machine learning (ML) has become an essential tool for malware detection, offering the ability to adapt to new threats by identifying patterns and anomalies in vast amounts of data. However, the effectiveness of ML-based systems depends on careful model design to avoid high false negative rates, which allow threats to slip through, and high false positive rates, which generate excessive alerts and reduce practical usability.

A key challenge in deploying ML for malware detection is the class imbalance inherent in cybersecurity datasets. Regardless of the type of attack, benign traffic overwhelmingly dominates most datasets, leading to biased models that favor the majority class. This imbalance significantly impacts a model's ability to effectively identify real threats, necessitating the adoption of advanced data-balancing strategies.

To evaluate the effectiveness of various strategies in addressing class imbalance and enhancing malware detection, we conduct experiments using two datasets: Drebin, which focuses on Android malware detection, and CIC-DoHBrw-2020, which captures DNS over HTTPS (DoH)-based attacks. These datasets represent two distinct yet critical cybersecurity threats. The Android ecosystem presents unique security challenges due to the diversity of applications, system flexibility, and hardware limitations. Meanwhile, DoH-based attacks obfuscate malicious communications within seemingly legitimate encrypted traffic, reducing the effectiveness of traditional detection mechanisms.

Our study investigates two advanced balancing techniques. The first, proposed by [9], involves dividing the majority class into subgroups while replicating the minority class before balancing, thereby reducing the reliance on synthetic samples. Our approach, HBBB method [**?**], balances each bootstrap in bagging individually, increasing data variability and, consequently, enhancing classifier robustness.

We carefully consider evaluation metrics to ensure they are robust to class imbalance. Furthermore, we explore model explainability using SHAP and instance hardness analysis, ensuring that predictions remain interpretable.

By integrating advanced balancing techniques with diverse classifier paradigms and explainability methods, this study aims to enhance malware detection systems, ensuring both robust performance and interpretability.

The key objectives of this work are:

- To evaluate the impact of different balancing techniques on malware classification performance.
- To compare the effectiveness of distinct classifiers
- To investigate SHAP and instance hardness to better understand the classification task, analysing particularly challenging samples.

II. Related Work

The detection of cyberattacks has been extensively studied, particularly with the application of machine learning (ML) techniques. However, the lack of interpretability in these models can hinder the understanding of their decisions, limiting their adoption in critical environments. In this context, Explainable Artificial Intelligence (XAI) emerges as a promising approach to enhance the transparency of Intrusion Detection Systems (IDS).

Alsaheel et al. (2023) proposed an explainable AI-based IDS to detect DNS over HTTPS (DoH) attacks. The authors utilized SHAP (SHapley Additive exPlanations) to interpret the decision-making process of the ML model, enhancing trust in the detection results. Their findings demonstrated that incorporating XAI improves the model's ability to detect DoH-based threats while maintaining interpretability.

Lopes et al. (2022) explored the application of ML in cyberattack detection, emphasizing the need for explainability techniques to understand model learning and assess effectiveness in real-world scenarios. The study highlighted the importance of analyzing false positive rates, a critical aspect of IDS efficiency.

Andreossi (2022) proposed a detection system for DDoS attacks combining XAI and ensemble learning techniques. By utilizing information entropy to detect anomalies, the system categorized network traffic using ensemble classifiers. The integration of XAI techniques provided deeper insights into the internal workings of the model, facilitating result interpretation.

In the context of smart electrical substations, Oliveira (2024) developed an enhanced IDS with explainability capabilities using XAI. The proposed system integrated temporal enrichment techniques and robust preprocessing, resulting in more accurate detection of complex attacks, such as Masquerade attacks. The approach increased the system's transparency, making it more interpretable for security operators.

Gimenes (2024) evaluated Wi-Fi networks (802.11) characteristics for the detection of impersonation attacks, applying XAI to understand the contribution of individual features. Using SHAP and the XGBoost classifier, the study identified key features essential for both normal traffic and specific attacks, promoting greater transparency and trust in IDS decisions.

These studies highlight the growing importance of integrating XAI techniques into IDS, aiming to enhance interpretability and effectiveness in identifying cyber threats across various environments.

III. Proposed Framework

In this section, we present the proposed framework for malware detection. To address the inherent imbalance in the datasets, we introduce the Hybrid Bootstrap-Based Balance (HBBB) technique, a bootstrapping-based approach that incorporates random selection of balancing strategies, and evaluate it alongside the method proposed by [9].

A. HBBB

The goal of HBBB is to improve the generalization ability of the model pool in highly imbalanced scenarios, particularly when integrated with Dynamic Selection (DS) classifier methods, building on the work of Roy et al. [7]. This framework helps train classifiers on a more diverse set of data, leading to improved robustness and adaptability.

The framework, as described in Figure I, begins by splitting the dataset Γ into two sets: the training set Γ_{train} and the test set Γ_{test}. The training set Γ_{train} is then divided into multiple bootstrap subsets $\Gamma_1, \Gamma_2, ..., \Gamma_N$, sampled with replacement. These subsets ensure diversity in the data used for model training.

Each bootstrap subset is balanced using a randomly selected balancing technique chosen from a predefined set. The degree of balancing applied to each subset is randomly chosen between 0.5 and 1.0, allowing partial or complete balancing, which ensures variation across the subsets and promotes diverse model training.

Once each subset is balanced, classifiers are trained on the modified datasets and added to the classifier pool.

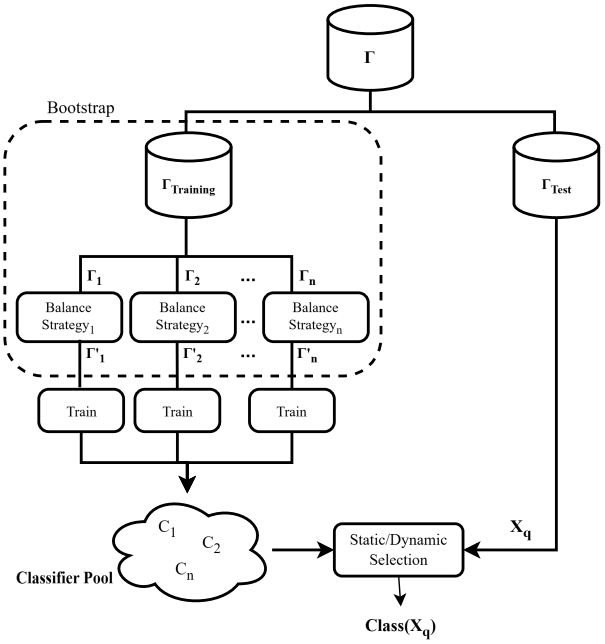

FIGURE I: SCHEMATIC REPRESENTATION OF THE HYBRID BOOTSTRAP-BASED BALANCE (HBBB)

During the testing phase, predictions are aggregated from this pool using static or dynamic selection strategies. Static methods apply the same fusion rule to all instances, whereas Dynamic Selection (DS) techniques adaptively choose the most appropriate classifier for each individual query. This dynamic approach enhances model flexibility, boosting performance, especially in scenarios with highly imbalanced datasets.

B. Zebin Method

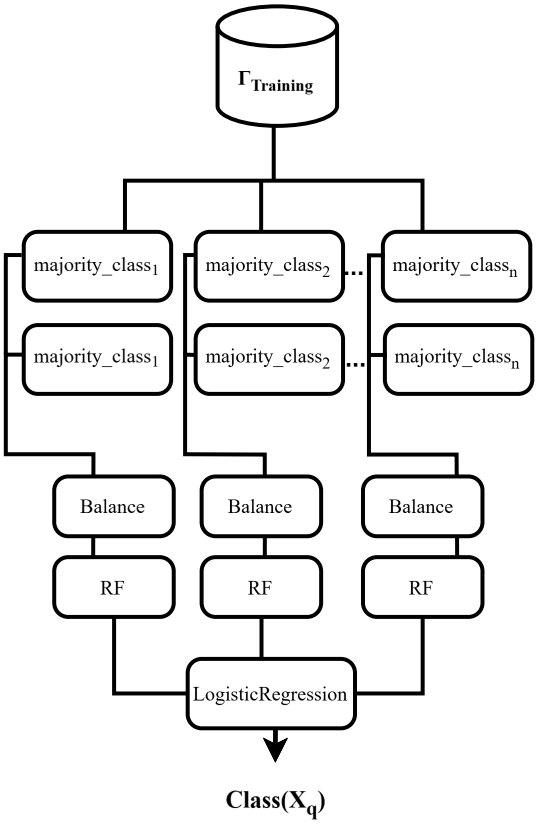

FIGURE II: SCHEMATIC REPRESENTATION OF THE METHOD PROPOSED BY [9]

The method proposed by [9] addresses class imbalance by dividing the majority class into subgroups, replicating the minority class, and applying balancing techniques to each subgroup individually, as shown in Figure **??**. This method enhances the diversity and adaptability of the model, particularly in imbalanced datasets.

First, the majority class is first divided into subgroups. This division helps to reduce the overall dominance of the majority class and ensures that each subgroup has a more balanced distribution of the minority and majority classes. To further balance the dataset, the minority class is replicated across all subgroups. This step ensures that the minority class is adequately represented in each subgroup, preventing the model from being biased toward the majority class.

For each subgroup, a balancing technique, SMOTE, is applied to equalize the number of instances of the majority and minority classes. After balancing, individual classifiers are trained on each of the subgroups. These classifiers are trained to learn patterns from each subgroup, focusing on maintaining a balance between the classes within each subset.

Finally, the models trained on each subgroup are combined using logistic regression. This method aggregates the outputs of the individual models, effectively merging the knowledge learned from each subgroup into a single,

robust classification model.

IV. Methodology

A. Dataset

1. CIRA-CIC-DoHBrw-2020

The CIRA-CIC-DoHBrw-2020 dataset is a benchmark dataset designed for the detection and classification of DNS over HTTPS (DoH) traffic. It was developed by the Canadian Institute for Cybersecurity (CIC) in collaboration with the Communications Security Establishment's Canadian Centre for Cyber Security (CIRA), aiming to facilitate research on identifying DoH-based cyber threats.

The dataset was generated by simulating real-world DoH and traditional DNS traffic in a controlled environment. It consists of network traffic traces collected from both benign and malicious DoH activities, ensuring a balanced representation of normal and attack scenarios. The data was captured using various web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge, as well as different DoH resolvers to improve diversity and realism. The class imbalance ratio is 45:1:12, with the majority class being benign DoH traffic, and the minority classes consisting of malicious DoH and benign-DoH.

The CIRA-CIC-DoHBrw-2020 dataset provides a robust foundation for training and evaluating machine learning models aimed at distinguishing between encrypted DoH traffic and conventional DNS requests. Its labeled nature enables the development of explainable systems, offering insights while maintaining user privacy.

2. Drebin

The DREBIN dataset, designed for malware detection research, contains 129,013 instances sourced from various markets and applications. Among these, 5,560 are classified as malware, with the remaining instances being benign. The dataset exhibits a significant class imbalance, with an imbalance ratio (IR) of 22.20, posing a challenge for model training.

DREBIN extracts a range of features from both the code and manifest files of Android applications, capturing different behavioral and structural aspects. These features include hardware components, requested and used permissions, application components, filtered intents, restricted API calls, suspicious API calls, and network addresses. Each instance in the dataset is represented by a feature vector, which records the frequency of features in each category.

This dataset provides a comprehensive foundation for developing machine learning models for Android malware detection, but the imbalanced nature of the data highlights the need for advanced methods to address class imbalance.

B. Balancing Tecniques

To address class imbalance in the training data, various balancing strategies were applied. These techniques aim to enhance model performance by providing a more balanced representation of classes, especially the minority class, during training.

1. Zebin Method

In [9] method, the training dataset from the CIRA-CIC-DoHBrw-2020 dataset exhibited significant class imbalance, a 45:1:12 class imbalance ratio. To mitigate this, the majority class (Non-DoH) was divided into three distinct subgroups, while the two remaining classes (Benign-DoH and Malicious-DoH) were replicated across all subgroups. The minority class (Benign-DoH) was then balanced relative to the intermediate class (Malicious-DoH) using SMOTE (*Synthetic Minority Over-sampling Technique*) [1]. This approach generated synthetic samples by interpolating neighboring instances in the feature space of the minority class. As a result, each subset had a final balanced distribution of 15:12:12, minimizing the need for excessive synthetic data and allowing for efficient parallel training of models.

In contrast, the Drebin dataset presents a different set of challenges. The imbalance ratio in Drebin was more than 22 benign samples for every malicious sample. In this case, the majority class (Benign) was divided into three distinct subgroups, and the minority class (Malicious) was replicated across all subgroups. SMOTE was applied to balance the minority class relative to the majority class, resulting in a equitable distribution across the classes within each subset.

2. Hybrid Bootstrap-Based Balancing (HBBB)

Hybrid Bootstrap-Based Balancing (HBBB) method performs balancing after the generation of subsets via bootstrap, leveraging sample variability to enhance the diversity of the classifier ensemble. HBBB extends the *Bootstrap-Based Balancing* (BBB) [8] by integrating multiple balancing strategies and different resampling percentages.

In HBBB, each subset generated by bootstrap undergoes a randomly selected balancing technique from a predefined set, including:

- ADASYN (*Adaptive Synthetic Sampling Approach for Imbalanced Learning*) [6]: generates synthetic instances of the minority class, prioritizing the most difficult ones to classify.
- *Random Over-Sampling* [1]: randomly replicates minority instances without creating synthetic data.
- *Random Under-Sampling* [5]: randomly reduces the number of majority class samples.

- SMOTE (*Synthetic Minority Over-sampling Technique*) [1]: generates synthetic instances by interpolating real examples.

By dynamically adjusting the resampling percentage for each subset, HBBB prevents excessive insertion of synthetic samples and reduces the risk of overfitting, especially in highly imbalanced scenarios. This process results in distinct data distributions for each subset, promoting greater diversity within the ensemble and enhancing the model's robustness and generalization capabilities.

C. Metrics

In this study, we evaluate the model's performance using six key metrics: accuracy, precision, recall, F1-Score, G-Mean, and MCC. Since the dataset is naturally imbalanced, relying on a single metric can provide a misleading assessment of the model's effectiveness. Each metric captures a different perspective of classification errors, helping to mitigate biases introduced by class imbalance [4].

- **Accuracy** measures the overall correctness of the model, but it can be misleading in imbalanced datasets. A model biased toward the majority class can achieve high accuracy while failing to effectively identify instances of the minority class, which is critical in scenarios like malware detection.
- **Precision** evaluates how many of the predicted malicious samples are actually malicious, helping assess the risk of false positives. In imbalanced datasets, it is important to ensure that precision is not compromised, as a model that incorrectly classifies many instances as malicious can be harmful.
- **Recall** assesses how well the model identifies malicious instances, which is especially important in cybersecurity applications, where detecting rare but critical threats is essential. A low recall rate can mean that many threats go undetected.
- **F1-Score** balances precision and recall, providing a more reliable measure in imbalanced scenarios. This metric is useful for evaluating models where detecting all instances of the minority class is crucial.
- **G-Mean** evaluates the balance between the true positive rate and false negative rate, which is important for checking if the model is detecting the minority class without compromising the majority class.
- **MCC (Matthews Correlation Coefficient)** is a robust metric that takes into account all four combinations of outcomes (true positives, false positives, true negatives, false negatives) and is particularly useful for imbalanced class problems.

V. Results and Discussion

In this section, we present and discuss the results obtained from applying the Zebin Method and Hybrid Bootstrap-Based Balancing (HBBB) methods to two datasets: CIRA-CIC-DoHBrw-2020 and Drebin. The performance metrics, including Accuracy, Precision, Recall, F1-Score, G-Mean, and MCC, are compared for each method on both datasets, as shown in Table I. The goal is to evaluate how each method performs under different balancing strategies and their overall impact on classification tasks.

A. Performance Comparision

From the results shown in Table I, we can observe that both methods, Zebin and HBBB, exhibit strong performance across the metrics

The results suggest that the HBBB method outperforms Zebin on both datasets, especially in terms of the metrics that reflect model adaptability and robustness, such as G-Mean and MCC. This can be attributed to the dynamic nature of the balancing strategies used in HBBB, which allows for more flexibility in addressing the class imbalance compared to Zebin's approach. Furthermore, the slight improvements in accuracy, precision, and recall on both datasets indicate that HBBB can generate more reliable and diverse classifiers, leading to better overall performance.

However, the results demonstrate that both methods are highly effective in detecting malware, but HBBB seems to offer a more refined approach, particularly when handling datasets with higher levels of imbalance or when different classifiers are needed to adapt to diverse data distributions.

Overall, the comparison highlights the strengths of both methods, with HBBB being particularly suitable for scenarios where a higher level of model adaptability and diversity is required, as seen in its superior performance across the datasets compared.

TABLE I: PERFORMANCE METRICS COMPARISON FOR ZEBIN AND HBBB ON DIFFERENT DATASETS

Zebin

Metric	CIRA-CIC-DoHBrw-2020	Drebin
Accuracy	99.49	98.22
Precision	99.56	98.41
Recall	99.49	98.22
F1-Score	99.51	98.29
G-Mean	99.59	93.18
MCC	98.61	80.44

HBBB

Metric	CIRA-CIC-DoHBrw-2020	Drebin
Accuracy	99.67	98.39
Precision	99.70	78.61
Recall	99.67	86.23
F1-Score	99.68	82.23
G-Mean	99.71	92.59
MCC	99.11	92.37

B. Instance Hardness Analysis

Instance Hardness (IH) is a metric used to assess the difficulty of classifying individual instances in a dataset. It helps quantify how challenging it is for a classifier to correctly categorize an instance. The objective is to identify which instances are particularly difficult for the model, shedding light on areas where the model's performance may be weaker. To calculate IH, we use the concept of k-Disagreeing Neighbors (kDN), which is defined as the proportion of the nearest neighbors of a given instance that disagree with its class.

The impact of balancing on instance hardness (IH) varies depending on dataset characteristics and instance complexity. Figure III presents the cumulative IH distributions for the Drebin dataset and Figure IV for the CIRA-CIC-DoHBrw-2020 dataset, before and after balancing.

In the Drebin dataset, where the imbalance ratio is 22.20, balancing significantly reduced the IH of malicious samples, from 0.27 to 0.04, making them much easier to classify. Conversely, the IH of benign samples increased from 0.01 to 0.04, slightly raising their classification difficulty. Despite this shift, balancing ultimately contributed to a more even distribution of hardness across classes, improving classification performance.

For the CIRA-CIC-DoHBrw-2020 dataset, which consists of three classes with different imbalance levels. The smallest class experienced a significant improvement, with its hardest instances becoming much easier to classify. Meanwhile, the two larger classes saw only a slight increase in IH, indicating that their classification complexity was only marginally affected. This suggests that balancing successfully mitigated the difficulties faced by the minority class without drastically increasing the challenge for the majority classes.

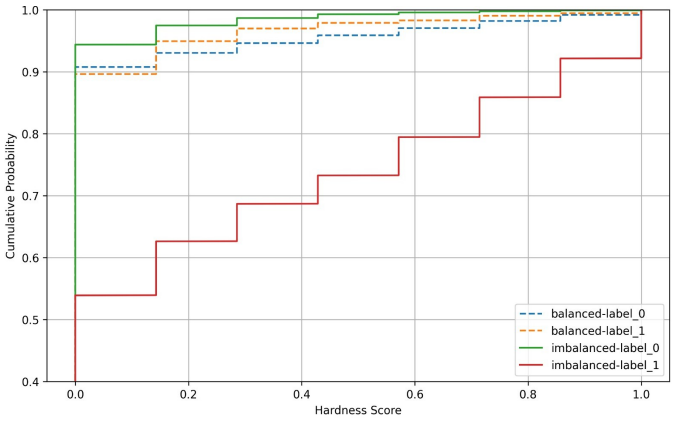

FIGURE III: CUMULATIVE DISTRIBUTION OF THE KDN SCORE FOR THE DREBIN DATASET

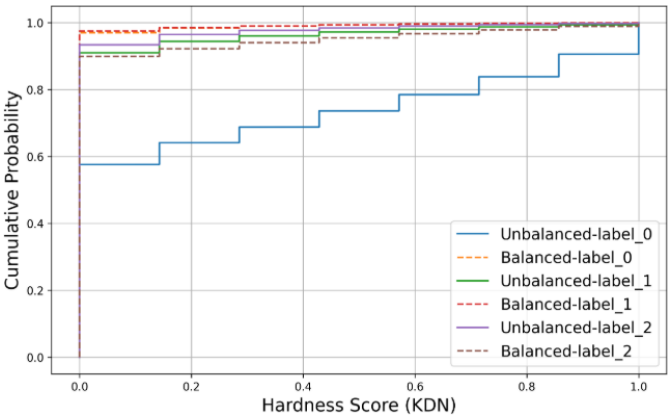

FIGURE IV: CUMULATIVE DISTRIBUTION OF THE KDN SCORE FOR THE CIRA-CIC-DOHBRW-2020 DATASET

C. SHAP Analysis

SHAP (SHapley Additive exPlanations) is a powerful technique used to provide interpretable explanations for predictive models, particularly complex ones like neural networks and random forests. Based on Shapley values from game theory, SHAP assigns a quantitative importance to each feature, indicating its impact on a specific instance's prediction. The Shapley value is essentially the average contribution of a feature across all possible combinations of features, ensuring a fair calculation of each feature's impact on the model's output. This allows SHAP to provide insights into the contribution of individual features while accounting for interactions between them.

One of SHAP's key strengths is its ability to highlight the relevance of variables within a model. By generating visualizations such as dependence plots or feature importance graphs, SHAP reveals which features have the greatest influence on a model's predictions. For instance, in malware classification, SHAP can identify which app permissions or attributes (e.g., camera access or network usage) are most influential in classifying an instance as

malicious or benign. Interpreting SHAP values involves understanding that positive SHAP values increase the likelihood of a positive class (e.g., malicious), while negative SHAP values suggest that a feature pushes the prediction toward the negative class (e.g., benign).

Figure V presents a SHAP value plot, where each point corresponds to an instance in the dataset. The x-axis represents the SHAP value, indicating the impact of this variable on the model's prediction. Higher absolute SHAP values suggest a stronger influence on classification, helping to interpret how this specific feature contributes to the decision-making process.

However, SHAP does come with some limitations. While it is incredibly useful for explaining complex models, it can be computationally expensive for very large datasets or models. Additionally, as the number of features increases, interpreting SHAP values may become more challenging, and the insights may require additional context or visualization techniques. Despite these challenges, SHAP remains a valuable tool for model interpretability, offering consistent and fair explanations that are essential, particularly in fields like cybersecurity where understanding model decisions is crucial.

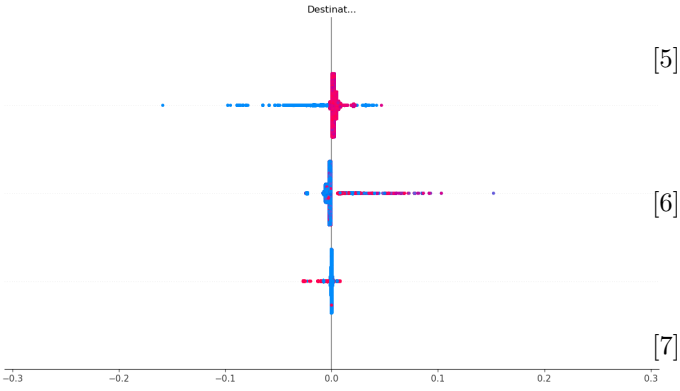

Figure V: SHAP value plot.

VI. Conclusion and Future Work

This study explored the impact of class balancing techniques, specifically Zebin and HBBB, on malware detection performance. Our results demonstrate that these techniques significantly improve detection, particularly for the minority class, without overly increasing complexity for the majority class. HBBB yielded superior results, highlighting its effectiveness in enhancing classifier performance. Overall, balancing contributes to more robust and interpretable malware detection systems.

Future research could focus on testing new classifiers and exploring the advantages and limitations of the methods in various balancing scenarios. Additionally, testing on different datasets with diverse attack types would provide valuable insights into the adaptability of these techniques.

[2]

References

[1] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2002.

[2] Yun-Chun Chen, Yu-Jhe Li, Aragorn Tseng, and Tsungnan Lin. Deep learning for malicious flow detection, 2018.

[3] Minuto da Segurança. Plataformas legítimas disseminam malware, 2025. Acessado em: 31 mar. 2025.

[4] Damien Dablain, Bartosz Krawczyk, and Nitesh Chawla. Towards a holistic view of bias in machine learning: bridging algorithmic fairness and imbalanced learning. *Discov Data*, Apr 2024.

[5] Chris Drummond and Robert C. Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *International Conference on Machine Learning (ICML)*.

[6] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks (IJCNN)*.

[7] Anandarup Roy, Rafael M.O. Cruz, Robert Sabourin, and George D.C. Cavalcanti. A study on combining dynamic selection and data preprocessing for imbalance learning. *Neurocomputing*, 2018.

[8] José Vinicius Souza, Camila Barbosa Vieira, George Cavalcanti, and Rafael Menelau Oliveira e Cruz. Imbalanced malware classification: an approach based on dynamic classifier selection. In *2025 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2025.

[9] Tahmina Zebin, Shahadate Rezvy, and Yuan Luo. An explainable ai-based intrusion detection system for dns over https (doh) attacks. *IEEE Transactions on Information Forensics and Security*, 2022.