Research on a Calibration Model for ML-Based Obfuscated Malware Detection

Timur Jamgharyan
National Polytechnic University of Armenia
Yerevan, Armenia
e-mail: t.jamharyan@polytechnic.am

***Abstract*—This paper provides a challenge of calibrating probabilistic predictions in machine learning models used for detecting obfuscated malware. A non-parametric post-processing technique - isotonic regression, is proposed to improve the reliability of output probabilities generated by nonlinear classifiers, particularly gradient boosting (XGBoost). The research is conducted in a virtualized environment using real-world and synthetically obfuscated malware samples. Evaluation metrics such as ROC AUC, PR AUC, Brier Score, Log Loss, and Expected Calibration Error (ECE) demonstrate that isotonic regression significantly enhances the calibration of probabilistic outputs without compromising classification performance. The results confirm the suitability of isotonic regression for highly imbalanced and noisy datasets, typical in obfuscated malware detection tasks.**

***Keywords*—Obfuscated malware, isotonic regression, gradient boosting, probability calibration, classification accuracy, ROC Curve.**

I. Introduction

Malware detection remains one of the key challenges in the field of information security. Malware developers frequently employ obfuscation (*obfuscation is the reduction of the source text or executable code of a program to a form that preserves its functionality, but complicates analysis, understanding of operating algorithms and modification during decompilation* [1]) techniques to conceal the presence of malware [2, 3]. Most deterministic obfuscators are based on two fundamental algorithms: *Kolberg algorithm* and the *algorithm proposed by Chenxi Wang* [4, 5]. These algorithms have served as a foundation for numerous methods and implementations of deterministic obfuscators. However, when machine learning (ML) methods are employed as obfuscation tools, detecting malware becomes significantly more difficult. Unlike deterministic obfuscators (*Dotfuscator CE, Net Reactor, ProGuard,* etc.), ML-based obfuscators incorporate a stochastic component, which complicates the analysis process. Malware classification also presents considerable challenges, particularly for models designed for static or pseudo-static analysis. Classical ML algorithms have demonstrated high effectiveness in the binary classification of malware and benign executable files [6-9]. An essential requirement for such ML-based malware detection models is the calibration of probabilistic predictions: incident response systems relying on model outputs must be able to trust the predicted probabilities.

To address this challenge, researchers in infrastructure security systems have explored a range of methods and solutions [10-12]. However, the use of isotonic regression as a technique for calibrating probabilistic outputs of classification models has not yet been thoroughly investigated. The motivation for employing this method lies in its ability to improve the calibration of output probabilities-especially in scenarios involving class imbalance and non-standard data distributions, which are typical in obfuscated malware detection. Unlike parametric approaches, isotonic regression does not impose assumptions on the shape of the calibration function [13], making it particularly suitable for tasks characterized by high uncertainty and noisy feature spaces.

The scientific novelty of this research lies in the formalization of isotonic regression as a post-processing technique for calibrating probabilistic estimates produced by nonlinear classifiers. To quantitatively assess the discrepancy between the empirical distribution of predicted probabilities and the true class distribution, the Kolmogorov-Smirnov test was employed.

II. Terms and Definitions

❖ **Gradient Boosting (XGBoost)** – an ensemble method based on decision trees that utilizes gradient approximation to minimize a predefined loss function. In binary classification tasks, the logarithmic loss function is commonly used:

$$L_{\log} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \qquad (1)$$

where, N - the number of observations, $y_i \in \{0,1\}$ -the true mark, p_i - predicted probability.

❖ **Isotonic regression** is a monotonic approximation method based on minimizing the squared deviation between predicted and true data.

$$\min_{f \in X} \sum_{i=1}^{N} \left(f(x_i) - y_i \right)^2 \qquad (2)$$

where X - the set of isotonic functions, y_i - the true values of the objective function, N - the number of observations [14-16].

❖ **The Kolmogorov-Smirnov** distance is a statistical measure used to quantify the difference between two probability distributions.

$$D_n = \sup_{t\in[0,1]} |F_n(t) - G_n(t)| \qquad (3)$$

where $F_n(t)$ - the empirical distribution function of predicted (calibrated) probabilities, $f(\hat{p}_i)$, $G_n(t)$ - the empirical distribution function of true class labels y_i ,sup – th supremum, that is, the greatest distance between two functions on a segment [0, 1].

III. Description of the Problem

Let the classifier output a probability estimate $\hat{p}(x) = P(y=1|x)$, that might not be properly calibrated. Isotonic regression is used to find a monotonic function $f:[0,1]\to[0,1]$ minimizing the loss functional.

$$\min_{f\in X} \sum_{i=1}^{n} (f(\hat{p}_i) - y_i)^2 \qquad (4)$$

where X – the set of isotonic (monotonically non-decreasing) functions, $\hat{p}_i$ - the predicted (calibrated) probability, $y_i \in \{0,1\}$ - the true class label. This approximation does not require parameterization and is well suited for highly noisy and non-stationary feature spaces typical of obfuscated code. To research the calibration model of probabilistic predictions of classifiers using isotonic regression as a nonparametric approach.

Boundary conditions

- Signs of obfuscated files may overlap with legitimate ones,
- FPR (False Positive Rate) value ≤5%,
- The model was trained on an unbalanced dataset (65%-benign, 35% - malware).

IV. Experiment Description

The Hyper-V role was enabled in a virtualized environment based on Windows Server 2019.Within a software-defined network, multiple operating systems were deployed, including Windows 10, Kali Linux, and Ubuntu 22.04 LTS (Fig. 1). On Ubuntu 22.04 LTS, the Snort intrusion detection system (IDS) was installed, enhanced with an ML plugin.

The following tools were used on Kali Linux:

- Metasploit Framework,
- Veil-Evasion and MSFvenom (for generating obfuscated payloads),
- Invoke-Obfuscation (for obfuscating PowerShell scripts).

As test malware samples, the following families were employed: *engrat*, *surtr*, *stasi*, *otario*, *dm*, *v-sign*, *tequila*, *flip*, *grum*, *mimikatz*, and others obtained from sources [17-19]. Obfuscation was performed using methods proposed in studies [20, 21]. For benchmarking purposes, the EMBER 2018 dataset [22, 23] was used, supplemented with 200 manually crafted obfuscated samples.

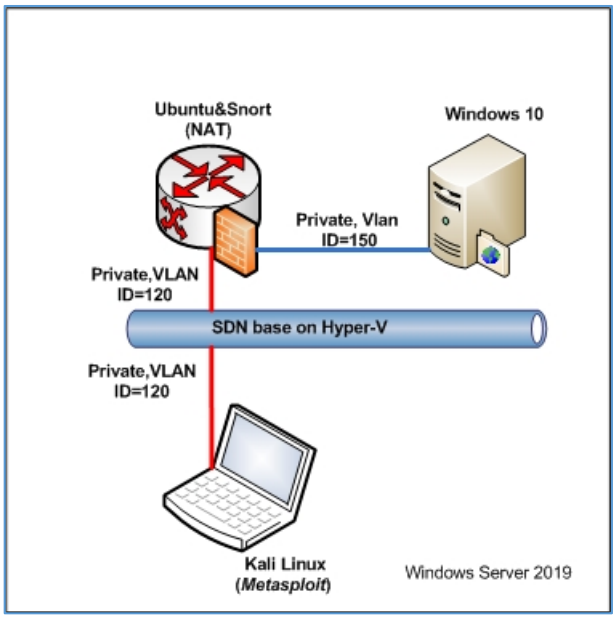

Fig. 1. Diagram of a Software-Defined Network (SDN)

Evaluation Metrics**:**

The following metrics were used to assess the model performance:

- AUC-ROC (Area Under the Receiver Operating Characteristic Curve) and AUC-PR (Area Under the Precision-Recall Curve) - to evaluate the overall ability to distinguish between malicious and benign software.
- Brier Score - to assess the calibration quality of probabilistic predictions.
- Expected Calibration Error (ECE) - to quantify the difference between predicted and actual probabilities;
- Log Loss - to evaluate the reliability of predictions.
- IR (Isotonic Regression) - for probability calibration.
- GB (Gradient Boosting) - to construct a strong classifier from an ensemble of weak learners.

Calibration curves and 3D error surfaces were visualized using *TensorFlow*. All computations were performed on a computing cluster consisting of 6 nodes, each equipped with an Intel Core i9 CPU and 32 GB of RAM. Model calibration was conducted using the *sklearn.isotonic.IsotonicRegression* module, and training was performed using *sklearn.ensemble.GradientBoostingClassifier*.

V. Research Results

Experiments were conducted on a dataset containing samples of both malicious and legitimate software. Particular attention was given to malware instances subjected to obfuscation techniques, including maximum code transformation, parameter substitution, and control flow generalization.

Both numerical and graphical results of the research are presented in Figures 2–5 and Table 1.

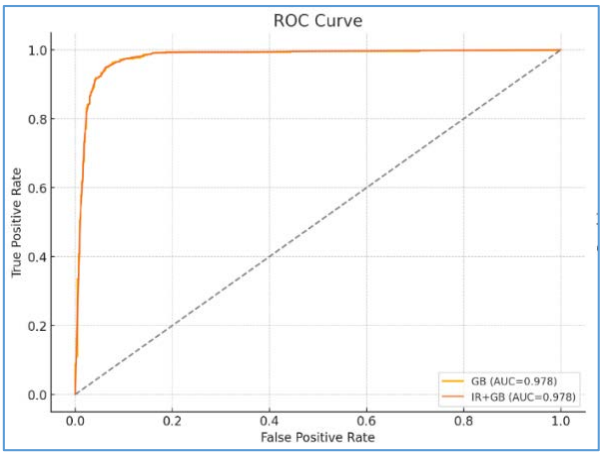

Fig. 2. TPR, FPR and ROC score for obfuscated malware

The ROC curves revealed a slight decrease in sensitivity (True Positive Rate, TPR) after calibration; however, the overall area under the curve (AUC) remained stable.

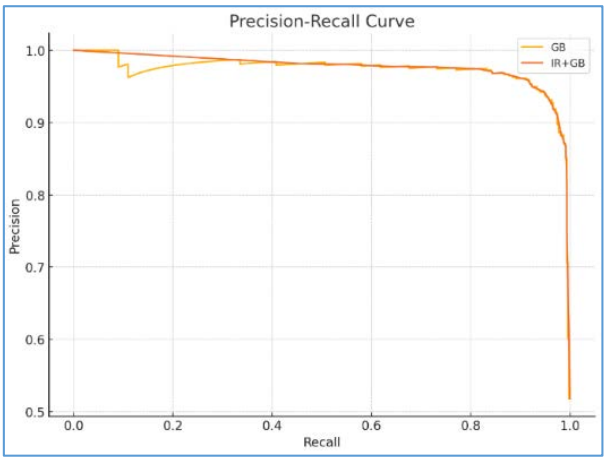

Fig. 3. Precision and recall metrics for obfuscated malware

The Precision-Recall curves improved for the calibrated model, particularly in the range of high recall values.

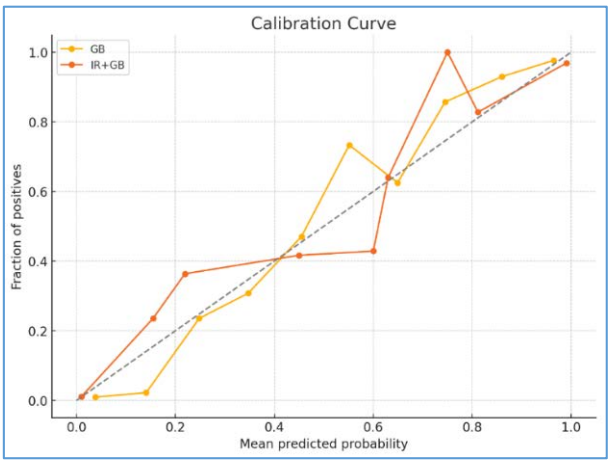

Fig. 4. Calibration curve for obfuscated malware

The calibration curves for XGBoost combined with isotonic regression closely align with the ideal diagonal.

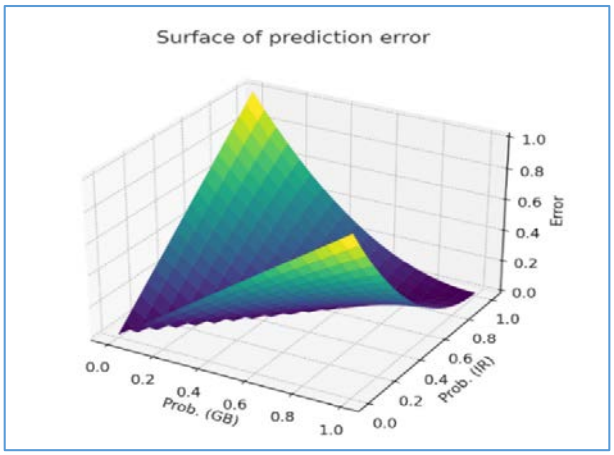

Fig. 5. 3D visualization of the predictor error surface

The 3D calibration error surface reveals local minima in regions of high prediction density, confirming the effectiveness of isotonic regression as a smoothing layer.

Table 1

Model	ROC AUC	PR AUC	Brier Score	Log Loss	ECE
XGBoost (uncalibrated)	0.971	0.944	0.132	0.314	0.072
XGBoost + isotonic regression	0.970	0.946	0.091	0.291	0.029
Isotonic regression (standalone)	0.912	0.885	0.102	0.336	0.047

This research presents a quantitative evaluation of the effectiveness of isotonic regression for calibrating probabilistic predictions in classification models. The results support the following conclusions:

- Isotonic regression significantly improves prediction calibration. A comparison of calibration curves before and after applying isotonic regression, along with a reduction in the Brier score, demonstrates improved alignment between predicted probabilities and the actual frequency of the positive class.
- The most notable improvements occur when using isotonic regression with models prone to overestimating probabilities, such as gradient boosting. In these cases, the method helps correct systematic deviations in the probability scale.
- Isotonic regression is particularly effective in imbalanced data scenarios, where the model's original probability estimates fail to reflect the true prior probability of the positive class.

VI. Conclusions

The results obtained demonstrate the practical value of using isotonic regression for calibrating probabilistic predictions in the task of detecting obfuscated malware. While gradient boosting (XGBoost) achieves high ROC AUC and PR AUC scores, its probability estimates tend to be poorly calibrated: the model overestimates its confidence, particularly around the 0.5 decision threshold. This is

evidenced by a relatively high Brier Score and Expected Calibration Error (ECE). Introducing an isotonic regression layer significantly reduces the calibration error (ECE from 0.072 to 0.029) without a notable decrease in the model's discriminative performance. This improvement is also reflected in the reduced Log Loss, which is critical in scenarios where predicted probabilities are used for decision-making, such as in incident prioritization systems. Isotonic regression is a simple yet effective tool for enhancing the reliability of probabilistic predictions and can be recommended as part of a standard pipeline for training and evaluating classification models.

It is also noteworthy that isotonic regression, even as a standalone model, delivers acceptable performance, particularly in terms of the Brier Score. This highlights its ability to capture generalized structure in probabilistic outputs despite its simplicity.

REFERENCES

[1] J. Aayush, H. Lin, A. Sahai, *"Indistinguishability Obfuscation from Well-Founded Assumptions"*. [Online]. Available: https://eprint.iacr.org/2020/1003

[2] S. Cao, N. He, Y. Guo, H. Wang, *"WASMixer: Binary Obfuscation for WebAssembly"*. [Online]. Available: https://doi.org/10.48550/arXiv.2308.03123

[3] N. Varnovsky, V. Zakharov, N. Kuzyurin, A. Shokurov, *"The current state of research in the field of program obfuscation: determining the resistance of obfuscation"*. [Online]. Available: https://doi.org/10.15514/ISPRAS-2014-26(3)-9

[4] C. Collberg and Jasvir Nagra, *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*, Addison-WesleyProfessional, 2009.

[5] C. Wang, J. Hill, J. Knight and J. Davidson, *"Software tamper resistance: obstructing static analysis of programs"*, Technical Report. University of Virginia, Charlottes ville, VA, USA., 2000.

[6] A. Niculescu-Mizil, R. A. Caruana, *"Obtaining Calibrated Probabilities from Boosting"*. [Online]. Available: https://doi.org/10.48550/arXiv.1207.1403

[7] E. Berta, F. Bach, M. Jordan, *"Classifier Calibration with ROC-Regularized Isotonic Regression"*. [Online]. Available: https://doi.org/10.48550/arXiv.2311.12436

[8] P. G. Fonseca, H. D. Lopes, *"Calibration of Machine Learning Classifiers for Probability of Default Modelling"*. [Online]. Available: https://doi.org/10.48550/arXiv.1710.08901

[9] M. V. Wüthrich, J. Ziegel, *"Isotonic Recalibration under a Low Signal-to-Noise Ratio"*. [Online]. Available: https://doi.org/10.48550/arXiv.2301.02692

[10] M. P. Naeini, G. F. Cooper, M. Hauskrecht, Binary Classifier Calibration Using a Bayesian Non-Parametric Approach. *Proceedings of the SIAM International Conference on Data Mining. SIAM International Conference on Data Mining*, pp. 208–216, 2015. [Online]. Available: https://doi.org/10.1137/1.9781611974010.24

[11] P. Pernot,*"Stratification of uncertainties recalibrated by isotonic regression and its impact on calibration error statistics"*. [Online]. Available: https://doi.org/10.48550/arXiv.2306.05180

[12] M. P. Naeini, G. F. Cooper, *"Binary Classifier Calibration using an Ensemble of Near Isotonic Regression Models"*. [Online]. Available: https://doi.org/10.48550/arXiv.1511.05191

[13] M. Kull, S. Filho, P. Flach, (2017). "Beyond Sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration", *Electronic Journal of Statistics*, vol.11(2). [Online]. Available: https://doi.org/10.1214/17-EJS1338SI

[14] S. Ravichandiran. *Deep Reinforcement Learning with Python. Master classic RL, deep RL,distributional RL, inverse RL, and more with open AI Gym and TensorFlow, Second Edition*, Packt. Birmingham-Mumbai, 2020.

[15] H. Brink, J. W. Richards, M. Fetherolf, *Real-World Machine Learning*, Manning Publications, pp. 338, 2017.

[16] H. Nelson, *Essential Math for AI. Nest-Level Mathematics for Efficient and Successful AI Systems*, O`Reily, pp. 594, 2024.

[17] Malware Bazaar Database, official download page. [Online]. Available: https://bazaar.abuse.ch/browse/

[18] Malware Bazaar Database, official download page. [Online]. Available: http://vxvault.net/ViriList.php

[19] Official page of the malware checking service. [Online]. Available: https://www.virustotal.com

[20] T. V. Jamgharyan, A. A. Khemchyan, "Malware Obfuscation Model Using Machine Learning", *Bulletin of High Technology*, Yerevan, Armenia, vol. 3 (31), pp. 77--83, 2024. [Online]. Available: https://doi.org/10.56243/18294898-2024.3-77

[21] T. V. Jamgharyan, V. S. Iskandaryan, A. A. Khemchyan, "Obfuscated Malware Detection Model", *Mathematical Problems of Computer Science*, Yerevan, Armenia, vol. 62, pp. 72--81, 2024. [Online]. Available: https://doi.org/10.51408/1963-0122

[22] H. S. Anderson, P. Roth, *"EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models"*. [Online]. Available: https://doi.org/10.48550/arXiv.1804.04637

[23] EMBER dataset download page. [Online]. Available: https://github.com/elastic/ember