Research on Classifier Calibration Methods in Network Infrastructure

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

Oxana Yarovikova
Military Academy of Communication,
Saint Petersburg, Russia
e-mail: oxanyarovikova@gmail.com

***Abstract*—This paper addresses the problem of calibrating probabilistic predictions produced by machine learning models in intrusion detection systems. Due to inherent bias and overconfidence, such models often yield poorly calibrated probability estimates. We propose Stepwise Dirac Calibration, a piecewise-constant calibration method based on partitioning the [0, 1] interval into non-overlapping subintervals.**

Stepwise Dirac Calibration explicitly accounts for the boundary conditions and targets reduction of calibration errors, including Expected Calibration Error, Brier score, and Log-loss.

Experimental evaluation on the CIC-IDS 2017 dataset demonstrates that Stepwise Dirac Calibration outperforms traditional methods in overconfident and unstable settings, while maintaining robustness and computational efficiency.

***Keywords*—Probability calibration, machine learning, intrusion detection systems, Stepwise Dirac Calibration, calibration error, CIC-IDS 2017, piecewise-constant functions.**

I. Introduction

The growing complexity and volume of network traffic have made traditional rule-based intrusion detection approaches increasingly insufficient. As cyber threats become more sophisticated, there is a pressing need for intelligent systems capable of detecting novel and subtle attack patterns. Artificial Neural Networks (ANN), when integrated into Intrusion Detection Systems (IDS), play a crucial role in enhancing the resilience of network infrastructures against cyberattacks. ANN and the Machine Learning (ML) models built upon them are increasingly applied in domains where both decision accuracy and the reliability of the decision are of paramount importance [1, 2]. Probabilistic classifiers provide such reliability in the form of estimated class probabilities. However, due to the overconfidence bias inherent in most models, these estimates often turn out to be poorly calibrated. For instance, a score of 0.9 does not guarantee that 90% of such cases correspond to the positive class [3, 4]. Many classification models tend to be overconfident in their predictions, which negatively affects decision-making in real-time security systems. Within IDS environments, it is essential not only to classify traffic correctly but also to interpret the confidence level of predictions.

In such high-risk scenarios, the assessment of prediction reliability requires deeper modification of the output probability distribution. Several studies have addressed probability calibration under these constraints [5-10], but the Stepwise Dirac Calibration (SDC) method has not yet been explored in the context of cybersecurity. SDC is particularly suitable under conditions of limited data availability and high noise, as it constructs a discrete, stepwise calibration function rather than a smooth one.

The scientific novelty of the proposed method lies in its ability to strengthen the discontinuities between adjacent intervals, thereby approximating the behavior of an ideal discrete calibrator.

II. Terms and Definitions

❖ **Stepwise Dirac Calibration** is a method belonging to the class of discrete models for calibrating probabilistic predictions. SDC assumes a piecewise-constant approximation of the calibration function over the set of predicted probability values. The idea is that after calibration, each value of $\hat{p}$ receives a point (discrete) estimate g_k. The plot of $g(\hat{p})$ resembles a sum of weighted delta functions and is defined by Expression (1) [3, 4].

$$g(p) \approx \sum_{k=1}^{K} g_k . 1_{B_k(p)} \qquad (1)$$

where: $g(\hat{p})$ is the calibration function - it transforms the predicted probability $\hat{p}$ into the calibrated one, K is the number of bins (intervals) in the calibration model, B_k is the k^{th} bin (an interval of probability values) on the [0, 1] axis, g_k is the local value of the function g, p is the generalized probability variable (often used as the argument of the function), $1_{B_k(p)}$ - is an indicator function for bin membership [11–13].

❖ **Expected Calibration Error (ECE)** is a measure of the discrepancy between the predicted probabilities of a model and the empirical frequency of positive outcomes. ECE is defined based on Expression (2) [4].

$$ECE = \sum_{k=1}^{K} \frac{|B_k|}{N} | g_k - \bar{p}_k | \qquad (2)$$

where: N is the total number of examples in the validation set, $\overline{p}_k$ is the average predicted probability value in bin B_k [4].

❖ **CIC-IDS 2017** (*Canadian Institute for Cybersecurity Intrusion Detection System 2017 dataset*) is an open dataset developed to simulate real network traffic, including various types of attacks. The traffic includes the following categories of attacks: *DDoS (Distributed Denial of Service), Brute Force, Botnet, Web attacks, Infiltration, PortScan, Heartbleed*, and others. The dataset contains more than 80 features based on flow analysis [15–16].

III. Description of the Problem

Let there be a classification model that produces probabilistic predictions $f(x) = \hat{p} \in [0,1]$], interpreted as an approximation of the conditional probability of belonging to the positive class. Here, $f(x)$ is the output of the uncalibrated model (the predicted probability of belonging to the positive class), $\hat{p}$ -is the estimated probability that object x belongs to the positive class. It is required to construct a calibration function $g:[0,1] \to [0,1]$ such that:

$$P\left(Y = 1 | g\left(\hat{p}\right) = p\right) \approx p$$

where: P is the probability function (probability measure), Y is a random variable, p is a probability value on the interval [0, 1].

Boundary conditions

- The number of intervals $K \in N$ is fixed.
- The value of the calibration function is constant within each interval B_k.
- Separate datasets are used for training and calibration: training is performed on one subset, while calibration and evaluation are performed on another one.

IV. Experiment Description

A High Performance Computing (HPC) cluster was used to install the Windows Server 2019 operating system (OS) [17], with the Hyper-V role activated. A Software-Defined Networking (SDN) environment was configured, within which Windows 10, Kali Linux, and Ubuntu 22.04 LTS OS were deployed (Fig. 1) [18, 19].

- On Ubuntu 22.04 LTS, the *Snort* IDS was deployed alongside ML libraries - *TensorFlow*, *NumPy*, and *Scikit-learn* - to evaluate performance using the metrics *brier_score_loss*, *log_loss* and *accuracy_score* [20].
- On Windows 10, a File Transfer Protocol (FTP) server was deployed and used as the target system for both legitimate and malicious connection attempts.
- Kali Linux was used to generate and analyze network traffic using tools such as *Metasploit* and *tcpdump*. Additionally, malware samples including *engrat*, *surtr*, *stasi*, *otario*, *dm*, *v-sign*, *tequila*, *flip*, *grum*, *mimikatz*, and others - sourced from [21–23] - were injected into the CIC-IDS 2017 dataset via *Metasploit*. Detection was carried out using the methods described in [25-27].

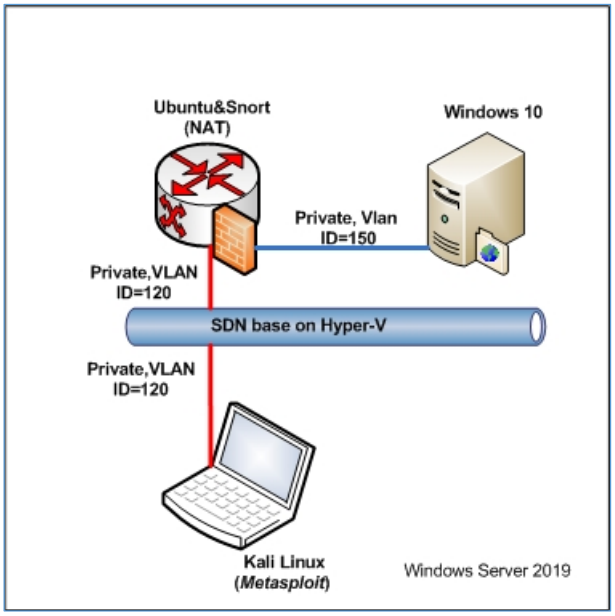

Fig. 1. Diagram of a Software-Defined Networking

After training the classifier on a training set, the output probability values were calibrated using the proposed method. For comparison, an approach based on isotonic regression was used [28, 29].

Calibration quality was evaluated using the metrics: *accuracy, ECE, Brier score, and log-loss.*

V. Research Results

Research results are summarized in Table 1, which shows the average calibration error across probability subranges, and Table 2, which compares calibration methods using key metrics on the CIC-IDS 2017 test set. Visualizations are presented in Figures 2–5 and were generated using the *TensorFlow Probability* and *Matplotlib* libraries.

Table 1

Average Calibration Error Across Probability Subranges

Probability Range	SDC (Error)	Isotonic Regression	No Calibration
[0.0 – 0.2]	0.011	0.015	0.038
[0.2 – 0.4]	0.014	0.019	0.041
[0.4 – 0.6]	0.026	0.026	0.047
[0.6 – 0.8]	0.018	0.023	0.044
[0.8 – 1.0]	0.016	0.020	0.039

Table 2

Comparison of Calibration Methods Based on Key Metrics

Calibration Method	ECE ↓	Brier score ↓	log-loss ↓	accuracy ↑
SDC	0.024	0.081	0.193	0.971
Isotonic regression	0.036	0.089	0.214	0.978
No calibration	0.083	0.104	0.271	0.962

The arrows indicate the desired direction of each metric: ECE - a lower value indicates more accurate model calibration, Brier score, log-loss - lower values indicate better performance, and accuracy - a higher value indicates better performance.

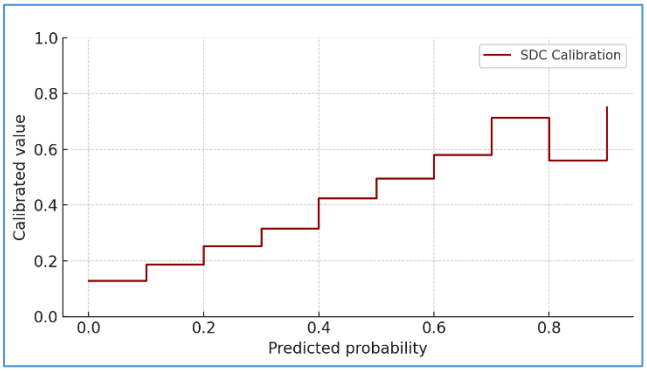

Fig. 2. Discrete Calibration Function

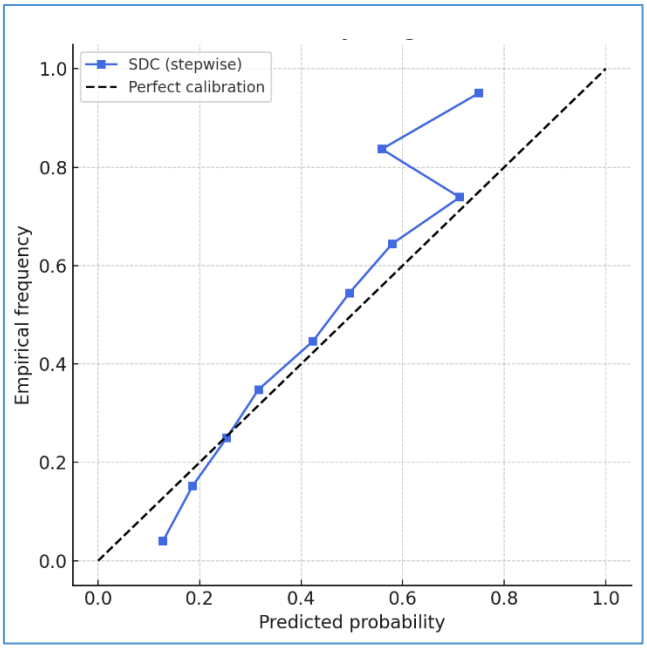

Fig. 3. Comparison of SDC and Perfect Calibration

In the mid-probability range ([0.4–0.6]), the accuracy of SDC is comparable to that of isotonic regression, demonstrating SDC's capacity to adapt to regions of high uncertainty while preserving its discrete structure. Moreover, SDC provides a more precise alignment between predicted probabilities and true class frequencies. This is particularly beneficial for overconfident models where conventional calibration methods tend to overcorrect.

By preserving local probability structure, SDC maintains calibration quality without introducing unnecessary smoothing, making it well-suited for security-critical applications such as IDS.

Additionally, its low computational complexity and interpretability further support its applicability in real-time and resource-constrained environments.

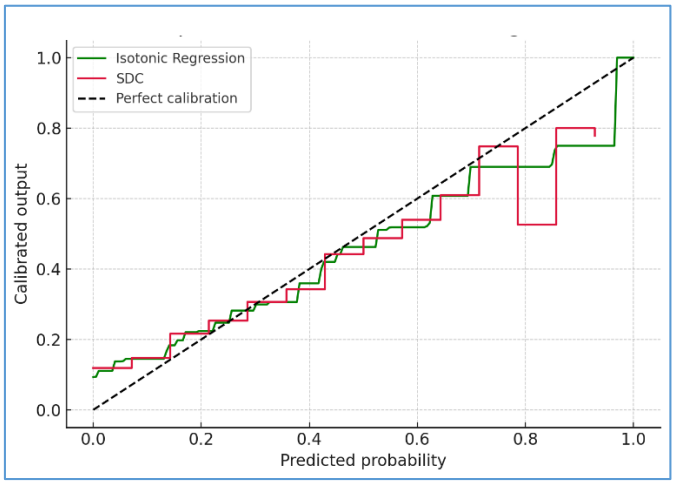

Fig. 4. Comparison of SDC and Isotonic Regression

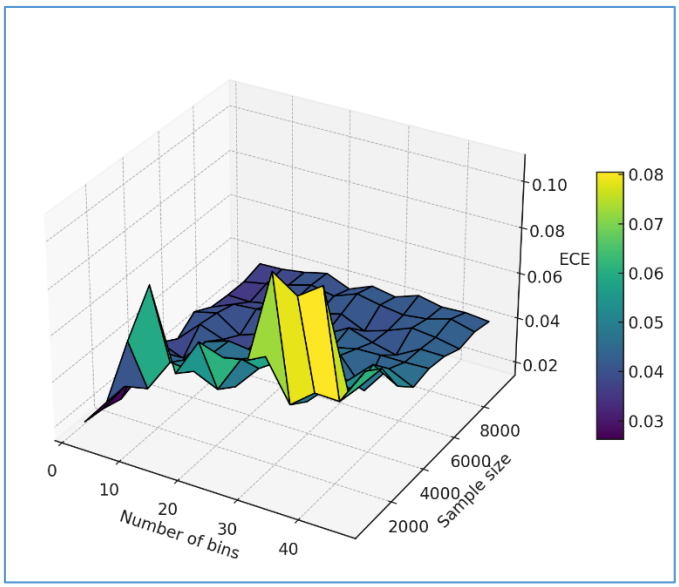

Fig. 5. 3D Error Surface in Feature Space

The spatial 3D error surface reveals that local minima align more precisely with regions of high sample density. In the boundary zones of probability distributions (i.e., at low and high levels of predicted confidence), SDC demonstrates stable and structurally consistent discrete calibration, which is especially important for systems making binary decisions based on threshold values in aggregate. This stability helps prevent critical misclassifications near decision boundaries. As a result, the method maintains reliability even under uncertainty or data sparsity at the extremes of the probability spectrum.

The piecewise-constant structure of SDC further reduces sensitivity to noise, ensuring smoother and more interpretable calibration behavior. Such consistency is especially valuable in safety-critical applications, where overconfident or unstable predictions can lead to severe consequences. Additionally, the alignment of calibration steps with sample-dense regions supports efficient use of available data without requiring aggressive smoothing or interpolation.

VI. Conclusions

The SDC method yields the lowest calibration error metrics, including ECE, Brier score, and log-loss, compared to isotonic regression. SDC maintains high classification accuracy without degrading the accuracy metric, confirming its compatibility with high-performance classifiers (in this case, XGBoost).

The method is effective when there are sufficient observations in each bin. However, in the presence of very rare probability values, overfitting may occur. SDC is particularly suitable for models with a tendency toward overconfidence, especially under class imbalance conditions.

Therefore, SDC can be recommended as an effective and easily implementable method for the discrete calibration of probabilistic predictions in cybersecurity applications, particularly within IDS systems operating under conditions of high dynamism and variability in network traffic.

References

[1] P. Malik, L. Nautial, M. Ram, *Machine Learning for Cyber Security*, Walter de Gruyter GmbH, Berlin/Boston, 2023.

[2] T. Jamgharyan, V. Ispiryan, "Network Infrastructures assessment stability", *AIP Conference Proceedings*, Yerevan, Armenia, vol. 2757, no. 1, 2023. [Online]. Available: https://doi.org/10.1063/5.0136237

[3] Y. Goodfellow, Y. Bengo, A. Courville, *Deep Leaning*, MIT Press, 2017.

[4] Simon J. Prince, Understanding Deep Learning, MIT Press, 2023.

[5] J. H. Shen, E. Vitercik, A. Wikum, *"Algorithms with Calibrated Machine Learning Predictions"*. [Online]. Available: https://arxiv.org/html/2502.02861v3

[6] T. S. Filho, H. Song et all. *"Classifier Calibration: A survey on how to assess and improve predicted class probabilities"*. [Online]. Available: https://arxiv.org/abs/2112.10327

[7] Cheng Wang, *"Calibration in Deep Learning: A Survey of the State-of-the-Art"*. [Online]. Available: https://doi.org/10.48550/arXiv.2308.01222

[8] S. E. Davis et all, *"Detection of calibration drift in clinical prediction models to inform model updating"*. [Online]. Available: https://doi.org/10.1016/j.jbi.2020.103611

[9] S. A. Balanya et all, *"Adaptive Temperature Scaling for Robust Calibration of Deep Neural Networks"*. [Online]. Available: https://doi.org/10.48550/arXiv.2208.00461

[10] J. Xiao et all, *"Restoring Calibration for Aligned Large Language Models: A Calibration-Aware Fine-Tuning Approach"*. [Online]. Available: https://arxiv.org/pdf/2505.01997

[11] K. P. Murphy, *Probabilistic Machine Learning. Advanced Topics*. MIT Press, 2022.

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

[13] P. Conde et all, *"A Theoretical and Practical Framework for Evaluating Uncertainty Calibration in Object Detection"*. [Online]. Available: https://arxiv.org/html/2309.00464v2

[14] A. Hekler, L. Kuhn, F. Buettner, *"Beyond Overconfidence: Foundation Models Redefine Calibration in Deep Neural Networks"*. [Online]. Available: https://www.arxiv.org/abs/2506.09593

[15] (2025) The Canadian Institute for Cybersecurity datasets download homepage. [Online]. Available: https://www.unb.ca/cic/datasets/index.html

[16] T. Breuel, *"The Effects of Hyperparameters on SGD Training of Neural Networks"*. [Online]. Available: https://doi.org/10.48550/arXiv.1508.02788

[17] (2025) Official download page for various versions of the Windows operating system, https://www.microsoft.com/en-us/evalcenter.

[18] Microsoft Learn. Software Defined Networking (SDN) description page https://learn.microsoft.com/en-us/windows-server/networking/sdn/

[19] (2025) Ubuntu operating systems official download page https://ubuntu.com/download/server

[20] (2025) Kali Linux operating systems official download page https://www.kali.org/get-kali/#kali-platforms

[21] (2025) Tensor Flow software official weebsite. https://www.tensorflow.org/

[22] (2025) The malware bazaar database website. [Online]. Available: https://bazaar.abuse.ch/browse/

[23] (2025) The malware database website. [Online]. Available: http://vxvault.net/ViriList.php

[24] (2025) The malware download webpage. [Online]. Available: https://github.com/vxunderground

[25] 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

[26] T. V. Jamgharyan, "Research the Model of Detection Polymorphic Malware by the Convolutional Neural Network", *Bulletin Of High Technology*, Yerevan, Armenia, vol. 3(27), pp.10-17, 2023. [Online]. Available: https://doi.org/10.56243/18294898-2023.3-10

[27] T. V. Jamgharyan, T. N. Shahnazaryan, "A Studu of a Model of Neural Network Application in the Decoy Infrastructure in the Defence Sphere", *Haykakan Banak /Armenian Army*, Yerevan, Armenia, vol. 2(120), pp.71-83, 2024. [Online]. Available: https://doi.org/10.61760/18290108-ehp24.2-71

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

[29] 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