IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 31, No. 10, October, 2026, pp. 3519-3529
Journal homepage: http://iieta.org/journals/isi

Pattern Aware Synthetic Generation for Network Intrusion Detection: An Adaptive Approach to Imbalanced Data

Mohammad M. Rasheed[1], Mustafa Muwafak Alobaedy[2*]

[1] College of Engineering, University of Information Technology and Communications, Baghdad 10013, Iraq
[2] Centre for Image and Vision Computing, Multimedia University, Cyberjaya 63100, Malaysia

Corresponding Author Email: mustafa.alobaedy@mmu.edu.my

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/isi.311023

Received: 18 March 2026
Revised: 22 July 2026
Accepted: 14 August 2026
Available online: 31 October 2026

Keywords:
class imbalance, network intrusion detection, adaptive resampling, oversampling selection, CIC-IDS2018, oracle headroom

ABSTRACT

Class imbalance is a persistent obstacle in network intrusion detection: the attack flows that matter most are the ones the training data contains least of. Oversampling is the usual remedy, but which method works best depends on the data, and that dependence has been documented far more often than acted upon. We present Pattern Aware Synthetic Generation (PASG), a framework that measures the geometry of the minority class and rules out resamplers the data does not support. It commits to one only when the evidence separating it from the alternatives is strong enough to trust. A prior question drives the study: how much is there to gain from such a choice at all? We answer it with the oracle headroom, which bounds what any selector could achieve, and we track it while varying the class ratio from 1:2.34 to 1:100 on seven attack subsets of CIC-IDS2018. The answer turns out to depend on the regime. At moderate imbalance the methods are statistically indistinguishable, the headroom is negligible, and leaving the data untouched is competitive. Beyond roughly 1:50 the headroom widens by an order of magnitude, the methods separate, and PASG records the highest mean minority-class F1-score of all seven strategies while recovering most of what is available. The contribution is an operating regime for adaptive oversampling, together with a cheap diagnostic that tells a practitioner in advance whether the effort is worth spending.

1. INTRODUCTION

Network intrusion detection systems (IDS) carry a structural awkwardness, in that the class mattering most for security, attack traffic, is the one the training data represents least. Benign flows dominate normal operation, while attacks are infrequent, varied in form, and uneven in how cleanly they separate from ordinary traffic [1, 2]. Classifiers respond to this asymmetry in a predictable way, optimizing aggregate accuracy and neglecting the minority class [3, 4]. Oversampling methods such as the Synthetic Minority Oversampling Technique (SMOTE) [5] and Adaptive Synthetic Sampling (ADASYN) [6], along with hybrids like SMOTE combined with Edited Nearest Neighbours (SMOTE-ENN), are the standard correction. They are simple, widely available, and often effective, but they are not uniform in their behavior, since their usefulness depends on the geometry of the minority partition, including how strongly features correlate, how attack samples cluster, how much noise sits on the boundary, and how far the classes overlap [4, 7, 8]. A method that handles volumetric distributed denial-of-service (DDoS) traffic well may be poorly suited to low-rate brute-force attacks, and one that is dependable on slow-rate denial-of-service (DoS) traffic can fail outright on infiltration.

This observation is not new, and surveys by Santos et al. [7], Carvalho et al. [9], and Chen et al. [10] have all recorded the data-dependent character of resampling. What has been missing is a way to act on the observation at training time, not merely to explain results after the fact. Pattern Aware Synthetic Generation (PASG) closes that gap by wrapping a reliability-aware selection mechanism around established oversampling techniques. The framework estimates a small set of minority-class meta-features, builds a filtered candidate pool, evaluates the candidates by inner validation, and gates the final commitment on the margin between the leading two. The design is conservative on purpose, so that when the evidence is ambiguous, PASG falls back to a safe default and declines a choice it cannot justify.

There is a question lurking behind all this, and the current study tackles it head-on: when, if ever, can a selection mechanism be of any assistance? No selector, no matter how well constructed, can make a significant difference if the candidates are almost the same on a particular set of data. We use a diagnostic we call the oracle headroom to measure the available margin, and define it in Section 4.5, rather than assuming it. We then systematically change the class ratio to allow the imbalance severity to be correlated with the margin available, rather than guessed.

The evaluation uses the CSE-CIC-IDS2018 benchmark of the Canadian Institute for Cybersecurity, hereafter CIC-IDS2018, from which we derive seven attack subsets covering infiltration, web attacks, DDoS, and slow-rate DoS. Each

subset pairs one attack family with benign traffic, so the seven share a capture environment and a feature extractor and differ only in the attack they contain. The protocol uses three classifier families and five seeds, a candidate pool of five resampling methods, and a Friedman omnibus test with a Nemenyi post-hoc procedure that controls the family-wise error rate across all pairwise comparisons. Four contributions follow from this design, and we set them out in the order the paper takes them up. We formalize PASG as an adaptive selector whose feasibility criteria and fallback rule are stated as testable conditions, so that the shortlist and the final choice can be reproduced from the manuscript alone. We introduce the oracle headroom as a diagnostic that bounds the attainable benefit of any selection procedure on given data. We identify the imbalance regime in which adaptive selection becomes worthwhile, which on this benchmark begins near a ratio of 1:50. And we separate saturated cases from difficult ones throughout, which lowers the headline numbers but produces a more reproducible account of where adaptive oversampling actually earns its cost.

2. RELATED WORK

2.1 Foundations of imbalanced learning

Class imbalance has been known as a long-standing challenge for over 20 years [3, 4]. Thabtah et al. [3] highlighted the systematic bias of traditional classifiers towards the majority classes, particularly in security applications where the minority class is the one of operational interest. Later studies showed that aggregate accuracy is a poor indicator when the data is imbalanced and that special learning schemes are needed [4, 11]. Recent work trends are in a unified trajectory. Carvalho et al. [9] considered the resampling problem from the data perspective and propose a trend towards adaptivity, where the method adapts to measurable properties of the data rather than being applied in a uniform way. The survey is wider in coverage by Chen et al. [10] who suggested that the choice of method should depend on the characteristics of the dataset, leaving no room for trial and error. Elreedy et al. [12] studied the distribution SMOTE creates and provide a mathematical justification for an intuition that many practitioners have: Interpolation is not always safe.

2.2 Oversampling techniques and their variants

The reference framework is still SMOTE [5], which creates synthetic minority instances by interpolating between the nearest neighbors. There have been dozens of variants since, each correcting some flaw in the original, and three of these are pertinent here. Borderline-SMOTE (B-SMOTE) only generates samples at the decision boundary areas, where samples are most likely to be misclassified [13]. The algorithm ADASYN focuses on the minority samples that have fewer neighbors, giving more weight to those minority samples surrounded by majority neighbors [6]. Hybrids like SMOTE-ENN combine oversampling with noise reduction by data cleaning to reduce the noise increase caused by interpolation [14]. Adaptive strategies have come in more recent times and there are four that are applicable here. The idea of selecting oversampling regions from minority density is similar to that of FLEX-SMOTE proposed by Bunkhumpornpat et al. [15]. Rana et al. [16] compared the performance of SMOTE, B-SMOTE and ADASYN and also find that their relative strengths vary depending on the noise level, boundary complexity and class geometry. Yi et al. [17] suggested adaptive synthesizer selection (ASN-SMOTE) and Bej et al. [18] introduced localized random affine shadow sampling (LoRAS) for imbalanced noisy data. These papers make it clear that there is no single oversampling method that is superior in all scenarios, but surprisingly few papers have developed working oversampling selection procedures that make use of the fact at training time.

2.3 Intrusion detection and class imbalance

CIC-IDS2018 has been used as a standard platform for intrusion detection research, providing a number of attack families and features extracted from dense flow-based real traffic [19]. It can be easily separated and tested in hard boundary cases, which is suitable for testing the generalization ability of oversampling. Uwazie et al. [20] used undersampling with SMOTE on CIC-IDS2018 and achieved an F1-score greater than 0.98 for the major attack categories and less than 0.9 for the minor subcategories. Abdelkhalek and Mashaly [21] combined ADASYN with deep neural networks and achieved F1-scores ranging from 0.82 to 0.89 for various datasets, with four times the number of operations of the baseline. SMOTE-ENN was found to be effective for fraud detection by Bounab et al. [22] which has a similar imbalance structure as our domain. Allagi et al. [23] used a conditional tabular generative adversarial network (CTGAN), which combines convolutional networks with their data and observed that generative models are able to better capture non-linear distributions than interpolation, but come at a high cost. Gu et al. [24] solved the problem of high-dimensional imbalanced traffic by learning-based protection, while Tabassum et al. [25] proposed a federated system based on generative adversarial networks for privacy-preserving deployment.

We are interested in oversampling on the CIC-IDS family because of four recent studies that are relevant to the family in particular and bear directly on our results. Talukder et al. [26] have applied SMOTE to extra-tree feature selection on CIC-IDS2017 and CIC-IDS2018, and found that the benefit was limited to the rarest attack categories, which is what we report here is an anticipation of. Alzughaibi and El Khediri [27] tested deep architectures on CSE-CIC-IDS2018 and found that the reported improvements decreased as soon as saturated attack families were taken out of the averages. Rodríguez et al. [28] have compared the resampling strategies using various classifiers on CIC-IDS2017 and concluded that the ranking of the methods is dependent on the type of classifiers, which is the reason why we chose a multi-classifier protocol. Ahmad et al. [29] found that hybrid sampling improved minority recall at the CIC-IDS2018, but decreased precision, which is exactly what our reliability gate is designed to regulate.

2.4 Toward adaptive method selection

The notion of method selection based on data is rooted in the algorithm-selection and meta-learning literature. Santos et al. [7] found that intrinsic data properties have an impact on the effectiveness of balancing techniques and Barella et al. [8] showed in a large empirical study that performance is very sensitive to data conditions, but neither of these studies developed a deployable system based on the findings.

2.5 Synthesis and research gap

The above work is all directed towards the same mechanism, but is not always explicitly mentioned. Interpolation-based oversampling is able to reconstruct the minority distribution from the geometry of the local neighborhood, and thus is reliable only to the extent that the geometry of the local neighborhood provides information. This dependence gives rise to three failure conditions: all of which have been reported in the literature reviewed above. If there is a strong correlation between features, then linear interpolation will result in implausible instances, as formalized by Elreedy et al. [12]. When the minority partition is sparse compared to its dimensionality, nearest-neighbor distances stop being informative and density-weighted schemes like ADASYN fail, which is why Rana et al. [16] report coverage failures. If there is a lot of overlap in the classes at the class boundary, creating classes there will increase the label noise rather than the support, which is what SMOTE-ENN's cleaning step is intended to address [14].

What the literature does not do is explain these conditions after a method is selected, rather than as quantities to be measured prior to the selection of a method. That is the gap we address, since if we can estimate the conditions under which reliability can be achieved from the training partition, then we can rule out methods that are likely to be unsuitable and can decide whether or not to leave a safe default. This is made possible by the reliability gate, which transforms a comparison of validation utilities into a statement of uncertainty that is essential for any component that is part of a security system.

3. PROPOSED METHOD

3.1 Framework overview

PASG is a training time selector that is wrapped around a normal oversampling pipeline. It takes a training fold, describes the minority class, filters and assesses candidates on a held-out fold, and then picks candidates within a reliability gate. The philosophy is being conservative; we want to avoid a switch that will give us a marginal benefit, but we do not want to accept the instability that switching on noise will bring.

PASG is not a learning algorithm, but an engineering component. What it controls is not the performance of the detection; it is the risk of having a configuration that is not supported by the evidence, and that changes on a small fluctuation of the validation, and that is difficult to audit, and even more difficult to explain to an operator. The reliability gate specifies the conditions for a switch; the feasibility rules specify the exclusion of a method as a function of measured properties of the data.

The framework is pattern-aware at the level of the generation policy and not the interpolation rule. The seven meta-features are used to describe the pattern of the minority partition, and the description of the pattern determines which generator will be admissible and which will be eventually used, thus two training partitions with different minority geometry will get different synthetic data from the same framework. Changing the policy, rather than the arithmetic, is a conscious decision, since that ensures that all candidates are a well-known published procedure, which is important for a component that needs to be audited in a security pipeline.

3.2 Meta-feature computation

Seven descriptors are computed from the minority class partition of the training data for each training fold, prior to the beginning of the inner cross-validation, and thus each candidate is evaluated with the same characterization. Each is a different aspect of the geometry that determines the effectiveness of oversampling.

Correlation Complexity (CC) is the percentage of feature pairs with a correlation value greater than 0.7 (in absolute value). With high CC, it is easy to get synthetic samples that do not respect the dependency structure by interpolation. Clustering Tendency (CT) is used to determine if the minority class is clustered, and is estimated by the Hopkins statistic, with values close to 1.0 being more cluster-aware and values close to 0.5 being more uniform. Variance Uniformity (VU) is a measure of the uniformity of variance, defined as 1—the coefficient of variation of the feature standard deviations. Low uniformity implies that some features have more influence on the distance computation, leading to bias of nearest neighbor methods.

Data Density (DD): It is the proportion of the minority sample size to the dimensionality. High values indicate that the settings are sparse, where distance-based methods are not reliable; Imbalance Ratio (IR) indicates the ratio of the majority to the minority setting; Class Overlap is an estimate of intermixing of the boundary of the classes, based on a cross-class analysis of the nearest neighbors. Minority Sparsity flags cases where there are few minority instances in the neighborhood; thus the local interpolation is delicate.

Algorithm 1. PASG training-time selector

Input: training set D_{train}, candidate pool C,
inner folds $F = 3$, margin threshold τ
Output: selected resampling method
M ← ComputeMetaFeatures(D_{train}) // seven descriptors
S ← BuildShortlist(M, C) // Algorithm 2, rules F1–F4
if $S = \emptyset$ then return SMOTE // no feasible candidate
partition D_{train} into F stratified folds
for each fold (D_{tr}, D_{val}) do
  for each method $m \in S$ do
    D'_{tr} ← Resample(D_{tr}, m) // on failure, F5 drops m
    clf ← Fit(D'_{tr})
    $J_f(m)$ ← MinorityF1(clf, D_{val})
$J(m)$ ← mean of $J_f(m)$ over the F folds
if $|S| = 1$ then return that method
m^*, m_2 ← the two best methods by J
if $J(m^*) - J(m_2) < \tau$ then
  return SafeFallback(S, M) // S1–S3, Table 2
else
  return m^*
Refit on full D_{train} with the selection

3.3 Feasibility criteria

The feasibility check that is called by Algorithm 1 is defined in the following explicit rules, allowing us to exclude a candidate and trace back to a measured property of the data. If any of the conditions in Table 1 are met, the candidate is declared infeasible, where $n_{\min}$ is the number of minorities in

the training partition and k is the size of the neighborhood. The rules F1–F4 are checked once at the construction of the shortlist and if a method is not excluded by these rules, it will not be fitted. The runtime guard (Rule F5) is used to remove a method from the shortlist if it fails the resampling test, but passes the first four tests.

Table 1. Feasibility rules applied before inner validation

Rule	Condition	Applies to	Rationale
F1	$n_{min} < k + 1$	All SMOTE variants	Interpolation needs k neighbors to be defined
F2	$n_{min} < 10$	ADASYN	Density weighting is undefined on very small minorities
F3	sparsity > 1.5 and DD > 0.5	ADASYN	Degenerate points in sparse high-dimensional regions
F4	$n_{min} < 40$	SMOTE-ENN, SMOTE-Tomek	Cleaning removes more support than it repairs
F5	Exception during resampling	All	Runtime guard against undetected degeneracy

Note: SMOTE = Synthetic Minority Oversampling Technique; ADASYN = Adaptive Synthetic Sampling.

3.4 The SafeFallback rule

SafeFallback is not simply a synonym for SMOTE, although SMOTE is its most frequent outcome. The rule is deterministic and, by design, does not reconsult the validation utilities that triggered it, since those utilities are the evidence, we have just judged insufficient. Table 2 gives the three rules, applied in order to the surviving shortlist S and the meta-feature vector *M*.

S1 reflects a simple observation. When overlap is high, boundary-focused generation is safer, because uniform interpolation in that condition amplifies boundary noise. S2 and S3 impose a total order over the pool so that the outcome is fully determined whatever the shortlist contains. The rule is reached only when at least two candidates survive filtering and their validation utilities fall within the margin, since a single survivor leaves nothing to arbitrate.

Table 2. SafeFallback rules, applied in order

Rule	Condition	Returned Method
S1	Overlap > 0.35 and B-SMOTE in S	Borderline-SMOTE
S2	Otherwise, SMOTE in S	SMOTE
S3	Otherwise	First feasible member of SMOTE, B-SMOTE, SMOTE-Tomek, SMOTE-ENN, ADASYN

Note: SMOTE = Synthetic Minority Oversampling Technique; ADASYN = Adaptive Synthetic Sampling.

3.5 Shortlist construction and selection logic

The candidate pool comprises SMOTE, B-SMOTE, ADASYN, SMOTE-ENN, and SMOTE combined with Tomek-link removal (SMOTE-Tomek). The admission thresholds of Algorithm 2 are fixed in advance and reported with the algorithm so that the shortlist is reproducible from the manuscript alone. We exclude the no-resampling Baseline from the candidate set and keep it as an external comparator.

Algorithm 2. Shortlist generation (BuildShortlist)

Input: meta-feature vector M, candidate pool C = {SMOTE, B-SMOTE, ADASYN, SMOTE-ENN, SMOTE-Tomek}
Output: shortlist S
$S \leftarrow$ {SMOTE, B-SMOTE} // safe defaults
if sparsity ≤ 1.5 and overlap ≤ 0.45 then
  $S \leftarrow S \cup$ {ADASYN} // density weighting valid
if overlap > 0.25 and $n_{min} \geq 40$ then
  $S \leftarrow S \cup$ {SMOTE-ENN, SMOTE-Tomek} // cleaning warranted
drop from S any method failing F1–F4 (Table 1)
return S

Two principles drive the selector, one applied before evaluation and one after. Filtering discipline removes methods the descriptors do not support before any held-out evaluation takes place, so that rules F1 to F4 of Table 1 are applied once inside Algorithm 2, so that infeasible candidates never enter the cross-validation loop, while rule F5 acts at runtime and drops a method that fails during resampling. Reliability-aware control handles the subtler case, which is what to do when the two leading candidates score almost identically on validation. Committing there means acting on a difference that may be nothing but noise, so PASG applies the rules of Table 2 instead. Where filtering leaves a single survivor, the gate has nothing to compare, and that method is returned directly.

The inner validation uses three-fold stratified cross-validation in place of a single holdout, because a single-holdout estimate proves too noisy to separate closely performing candidates, a point we quantify in Section 6.

3.6 Observed selector behavior

Across the 105 selector invocations of the extended protocol, SMOTE was chosen 90 times, B-SMOTE 14 times, and SMOTE-Tomek once. It was the reliability gate, not the utility comparison, that decided most of these: rule S2 fired 86 times, rule S1 nine times, and the margin was exceeded, permitting a direct commitment in ten cases. We do not present this as a strength, and Section 6 examines what it means.

4. EXPERIMENTAL DESIGN

4.1 Datasets

All experiments used the comma-separated value files distributed with CIC-IDS2018 [19], with no simulated or synthetic traffic involved. The seven attack subsets, summarized in Table 3, span infiltration, web attacks, DDoS, and slow-rate DoS. Each was reduced to a binary Benign versus Attack task, for which we removed duplicate header rows, dropped non-numeric fields, and cleaned numeric features of invalid values. The row counts in Table 3 are those of the source files; the imbalance-regime experiments of Section 4.4 resample the minority partition of each subset to the target ratio.

Table 3. Attack subsets and the per-seed evaluation cap

Attack Subset	Rows	Features	Benign	Attack	Cap/Seed
Thursday-01-03	331,100	69	238,037	93,063	15,000
Distributed denial-of-service (DDoS)-LOIC-UDP	5,784	69	4,054	1,730	5,784
BruteForce-Web	2,073	67	1,462	611	2,073
BruteForce-XSS	734	69	504	230	734
Denial-of-service (DoS)-Slowloris	36,754	68	25,764	10,990	15,000
DoS-SlowHTTPTest	469,631	68	329,741	139,890	15,000
DoS-GoldenEye	139,922	68	98,414	41,508	15,000

Table 4. Divergence between full and capped distributions

Attack Subset	Rows (Full)	KL Mean	KL Max	JS Mean	JS Max	Attack Rate Full / Capped
Thursday-01-03	310,830	0.00542	0.01955	0.01749	0.02475	0.2994 / 0.2994
DoS-SlowHTTPTest	467,233	0.00332	0.02899	0.01014	0.02354	0.2994 / 0.2994
DoS-GoldenEye	138,637	0.00484	0.02386	0.01656	0.02336	0.2994 / 0.2994
DoS-Slowloris	36,707	0.00299	0.01635	0.01316	0.02012	0.2994 / 0.2994

Note: KL = Kullback-Leibler; JS = Jensen-Shannon; DoS = denial-of-service.

Reproduction depends on more than the seeds. The five capture files used here, dated 15, 16, 21 and 22 February and 1 March 2018, were taken from the ids-intrusion-csv distribution of CIC-IDS2018 on Kaggle, which is one of several redistributions that differ in how they clean the original release. Every stochastic step is seeded from the run seed: the benign draw, the ratio construction, the train-test split, the inner folds, each resampler, and each classifier. Running the notebook twice in the same environment reproduces every reported value to fifteen decimal places, but the values are tied to the library versions, since the behavior of the resamplers and of the gradient boosting implementation has changed between releases. Hyperparameters are given in Section 4.3; every classifier is otherwise left at its library defaults.

Large datasets were capped at 15,000 rows per seed so that the protocol would run under limited compute. Smaller files were used in full, and Section 4.2 gives the evidence that this cap does not distort the results.

4.2 Justification of the evaluation cap

This type of cap needs to be justified because if the sample was to alter the feature distributions, then all subsequent comparisons would be incorrect. There are two types of evidence: one distributional and one empirical. The Kullback-Leibler (KL) divergence and Jensen-Shannon (JS) distance between the full empirical distribution and the stratified capped sample were computed for each feature, with 50 histogram bins over the central 99% of the range of each feature. Second, the entire protocol was repeated without any limit on the number of datasets (where it was feasible).

A JS distance of less than 0.05 is considered to be near-identical distributions and a per-feature KL divergence under 0.10 is typically negligible. All values in Table 4 are within both bounds and the proportion of the attack is retained to four decimal places. The same was true for the uncapped runs, which had a mean absolute difference in minority-class F1 of 0.00040 and a maximum difference of 0.00130, both of which were below the seed-to-seed standard deviation for all methods tested. We thus retain the cap and do not go so far as to say that it poses a threat to generalizability.

4.3 Evaluation protocol

Evaluation used repeated stratified holdout with a 70:30 split. Conclusions resting on a single classifier are fragile, because the ranking of resampling methods is known to shift with the learner [28], so the protocol uses five seeds, three classifier families of differing inductive bias, and a five-method candidate pool. The classifiers are a random forest with 30 trees and maximum depth 10, a histogram-based gradient boosting classifier with 100 iterations, maximum depth 6 and a learning rate of 0.1, and logistic regression with a 1,000-iteration limit. Scaling and every resampler were fitted on the training partition only, and the compared methods are Baseline, SMOTE, ADASYN, SMOTE-ENN, B-SMOTE, SMOTE-Tomek, and PASG.

4.4 Imbalance regime protocol

A single class ratio cannot reveal how the margin between methods behaves. Since the margin available between resampling methods is itself a function of imbalance severity, we treated the ratio as an experimental variable. At each target ratio, we held the majority partition fixed and drew the minority partition down to size, so the manipulation touches only the class ratio and leaves the majority distribution alone. The total stayed capped at 15,000 rows, consistent with the rest of the study. A level was skipped whenever the subset could not be built at it: when the attack family holds fewer flows than the ratio demands within the 15,000-row total, when the minority count would fall below 30, or when fewer than 100 majority flows remain, a condition no subset in this study triggered. BruteForce-Web, with 611 attack flows, therefore enters only at 1:100, and BruteForce-XSS, with 230, is absent from the sweep entirely. Five subsets are present at every level, and the balanced-panel check of Section 5.2 uses those five. The full sweep ran on the random forest, and we evaluated the other two classifier families at 1:50 to check that the pattern is not an artifact of one classifier.

4.5 Metrics and statistical testing

The main metric was minority-class F1, since in deployment a missed attack costs more than a small loss of overall accuracy. We report it along with weighted F1, precision, recall and runtime. We use a Friedman omnibus test over complete blocks and then Nemenyi, a multiple-comparison test that controls the family-wise error rate and provides a critical difference for making rank separations. For clarity, the Bonferroni-adjusted Wilcoxon results are retained. There are two new diagnostics. The headroom of the oracle is the distance between the best candidate found in hindsight on the test partition and the default, and is a limit on what any

selector could do. The within-block spread refers to the difference between the minority class F1 within a single block and indicates whether the methods are distinguishable or not.

5. RESULTS

5.1 How much improvement is available

For each block, we calculated the headroom for the oracle and the spread within the block, as described in Section 4.5. Table 5 presents both at each class ratio, together with the rank PASG achieved among the seven strategies and the Friedman *p*-value. The last column shows the percentage of the headroom that the framework recovered; it is blank at the two ratios where PASG scored lower than the default, as a negative percentage of a near-zero headroom is not meaningful. This is a topic that will be revisited in Section 5.2. The spread increases by a factor of eleven between 1:10 and 1:100 and the headroom is approximately 3 times as large, so the methods are on average 0.0013 apart at 1:10, and 0.0146 apart at 1:100, which is less than the variation in any one method from seed to seed. The result is illustrated in Figure 1: The Baseline decreases as the resampling methods become further apart than about 1:50.

Table 5. Achievable margin and separability by class ratio

Ratio	Minority (Mean)	Oracle Headroom	Within-Block Spread	PASG Rank	Friedman *p*	Headroom Captured
1:2.34	varies	0.0061	0.0158	3 of 7	0.010	—
1:10	955	0.0013	0.0101	6 of 7	0.293	—
1:25	404	0.0031	0.0082	2 of 7	0.018	74.6%
1:50	206	0.0115	0.0159	1 of 7	0.030	92.1%
1:100	104	0.0146	0.0317	1 of 7	0.002	83.5%

Note: PASG = Pattern Aware Synthetic Generation.

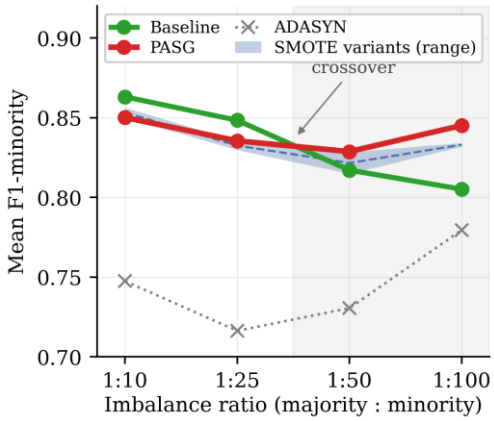


Figure 1. Mean minority-class F1 against class ratio

5.2 Method performance across imbalance regimes

Headroom indicates a gain opportunity; the question then is who will get it? At both 1:10 (0.8630) and 1:25 (0.8483), the highest mean minority-class F1 is recorded in the Baseline and the Friedman test does not reject the null hypothesis at all (p = 0.293). At 1:50, PASG is the most successful strategy for minority-class F1, with a mean score of 0.8287, and at 1:100, PASG outperforms the other strategies with a mean score of 0.8451, while the Baseline strategy scores the lowest. The percentage of the available headroom recovered by the framework increases as the margin offered increases: 74.6% at 1:25, 92.1% at 1:50 and 83.5% at 1:100, where it is the first. The negative value, which is large in magnitude, is at 1:10, since the framework scores below the default with a headroom of just 0.0013; a small absolute error is a large relative error. That is the idea, after all, when the margin is so close, it is more expensive to be wrong than to be right. The crossover is between 1:25 and 1:50, and is clearly visible in Figure 1.

This result has been tested with two checks. The panel is not uniform at the various levels because some of the lower ratios require larger minority samples than some of the subsets: the sweep is based on six subsets at 1:100 and five at other ratios. The trend is maintained when the analysis is repeated on the balanced panel of five subsets present at each level: headroom 0.0013, 0.0031, 0.0115, and 0.0156; PASG is in 6th place, then 2nd place, then 1st place at both severe levels. It is also not a classifier-specific pattern, as at 1:50 PASG is the top performer with the random forest and is the second with the logistic regression, and second with the gradient boosting (B-SMOTE is 0.0011 ahead).

5.3 Results at the native class ratio

Table 6 gives the aggregated results at the native class ratio under the extended protocol. PASG does not achieve the highest mean minority-class F1 at this ratio. B-SMOTE is higher on all three classifier families: 0.9221 against 0.9178 with the random forest, 0.9245 against 0.9233 with gradient boosting, and 0.8967 against 0.8955 with logistic regression. The differences are small relative to the standard deviations and none is significant, but they run the same way on every classifier family, so we do not claim superiority at this ratio. Table 5 explains why: at 1:2.34 the headroom is 0.0061, so there was little to win in the first place.

Table 6. Overall results at the native class ratio

Classifier	Method	Runs	F1-Weighted	F1-Minority	Precision	Recall	Time (s)
Random forest	B-SMOTE	35	0.9500 ± 0.1165	0.9221 ± 0.1804	0.9157	0.9302	0.70
	PASG	35	0.9475 ± 0.1160	0.9178 ± 0.1797	0.9130	0.9244	2.67
	SMOTE	35	0.9542 ± 0.0993	0.9165 ± 0.1829	0.9305	0.9066	0.53
	SMOTE-Tomek	35	0.9539 ± 0.1001	0.9155 ± 0.1855	0.9303	0.9051	1.99
	SMOTE-ENN	35	0.9497 ± 0.1057	0.9143 ± 0.1810	0.9175	0.9120	1.95
	Baseline	35	0.9550 ± 0.0994	0.9060 ± 0.2128	0.9732	0.8857	0.34
	ADASYN	24	0.9341 ± 0.1198	0.8829 ± 0.2142	0.8928	0.8755	0.86

	B-SMOTE	35	0.9511 ± 0.1142	0.9245 ± 0.1756	0.9148	0.9359	0.93
	PASG	35	0.9522 ± 0.1097	0.9233 ± 0.1753	0.9180	0.9305	4.06
	SMOTE-Tomek	35	0.9562 ± 0.0994	0.9233 ± 0.1752	0.9269	0.9211	2.84
Gradient boosting	SMOTE	35	0.9556 ± 0.1009	0.9225 ± 0.1773	0.9255	0.9207	0.93
	SMOTE-ENN	35	0.9522 ± 0.1053	0.9209 ± 0.1743	0.9180	0.9240	2.50
	Baseline	35	0.9580 ± 0.0957	0.9125 ± 0.2034	0.9741	0.8957	0.59
	ADASYN	24	0.9359 ± 0.1209	0.8918 ± 0.2044	0.8893	0.8947	1.17
	B-SMOTE	35	0.9283 ± 0.1343	0.8967 ± 0.1885	0.8684	0.9334	1.14
	PASG	35	0.9276 ± 0.1339	0.8955 ± 0.1878	0.8661	0.9334	4.12
	SMOTE	35	0.9362 ± 0.1129	0.8929 ± 0.1942	0.8766	0.9128	0.91
Logistic regression	SMOTE-Tomek	35	0.9362 ± 0.1128	0.8928 ± 0.1945	0.8767	0.9125	2.45
	SMOTE-ENN	35	0.9322 ± 0.1151	0.8884 ± 0.1928	0.8698	0.9103	2.82
	Baseline	35	0.9414 ± 0.1044	0.8845 ± 0.2208	0.9207	0.8874	0.82
	ADASYN	24	0.9087 ± 0.1263	0.8460 ± 0.2194	0.8237	0.8739	1.14

Note: F1-minority is the F1-score of the attack class; F1-weighted is the support-weighted F1 across both classes; precision and recall refer to the attack class; Runs counts the 35 tasks completed without failure; B-SMOTE denotes Borderline-SMOTE. ADASYN completed 24 because rules F2 and F3 of Table 1 exclude it where the minority partition is sparse, and we keep its results for completeness. Values are mean ± standard deviation over the seven attack subsets and five seeds.

PASG = Pattern-Aware Synthetic Generation; B-SMOTE = Borderline-SMOTE; ADASYN = Adaptive Synthetic Sampling; SMOTE = Synthetic Minority Oversampling Technique; SMOTE-ENN = SMOTE combined with Edited Nearest Neighbours; SMOTE-Tomek = SMOTE combined with Tomek-link removal.

The means in Table 6 are actually very strongly bimodal, with respect to the individual tasks. The scores are concentrated close to 1.0 on the easy sets of items and have a long tail on the hard sets of items, with interquartile ranges from 0.021 to 0.045 and ranges from 0.38 to 1.00. Section 5.4 explores what this pattern means.

5.4 Saturation and per-subset behavior

Meanings of averages over the entire benchmark are obscured as well as clarified. The picture is rather uneven for the oracle headroom for each attack subset at each ratio (Figure 2). Two subsets, DDOS-LOIC-UDP and DoS-SlowHTTPTest, are flat zero, meaning that there was no selection procedure that could have done better than the default for these two subsets anywhere. DoS-GoldenEye is 0 at three of four levels and only 0.0023 at 1:50, while DoS-Slowloris remains in the 0.0012-0.0049 range. The effect reported in Table 5 is overwhelmingly from the subset of infiltration, Thursday-01-03, with a headroom of 0.0729 at 1:100. Remove it and the mean headroom at 1:100 falls from 0.0146 to 0.0029. We say this simply because an average absorbs it, and it defines the limit of usefulness of the framework. If there is no difference between the scores of all candidates (as in Figure 2, the default), then adaptive selection would have been no better than the default. If the subset cannot be built at that ratio due to lack of attack flows, then it will be marked as a blank cell.

Table 7 collects the three attack subsets that were genuinely difficult at the native class ratio, namely Thursday-01-03, BruteForce-Web and BruteForce-XSS; all seven are available there, unlike in the sweep of Section 5.2. PASG does much better than Baseline (0.8087 vs. 0.7811) and does not underperform: B-SMOTE does at 0.8187 and ADASYN at 0.8129. The total separation between the seven methods is 0.0376, which is consistent with Section 5.3, and with the values in Table 5, as these three subsets are at moderate class ratios.

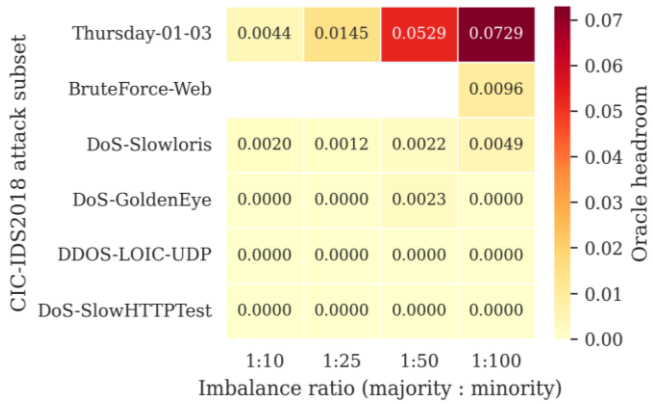


Figure 2. Oracle headroom per attack subset and class ratio

Table 7. Results on the challenging subset

Method	F1-Weighted	F1-Minority	Precision	Recall	Time (s)
B-SMOTE	0.8835	0.8187	0.8035	0.8376	0.68
ADASYN	0.8947	0.8129	0.8286	0.8012	0.68
PASG	0.8777	0.8087	0.7976	0.8239	3.43
SMOTE	0.8934	0.8057	0.8385	0.7824	0.50
SMOTE-Tomek	0.8927	0.8032	0.8379	0.7790	1.15
SMOTE-ENN	0.8830	0.8007	0.8083	0.7950	1.09
Baseline	0.8953	0.7811	0.9378	0.7336	0.37

Note: ADASYN = Adaptive Synthetic Sampling; SMOTE = Synthetic Minority Oversampling Technique; PASG = Pattern Aware Synthetic Generation; SMOTE-ENN = SMOTE combined with Edited Nearest Neighbours; SMOTE-Tomek = SMOTE combined with Tomek-link removal.

5.5 Which method wins where

Table 8 lists the best method for each attack subset under the random forest. The pattern is the argument of the study in miniature: no strategy wins everywhere. B-SMOTE leads both brute-force subsets, ADASYN takes DoS-Slowloris, and B-SMOTE and PASG tie on Thursday-01-03. Three subsets are saturated: DDOS-LOIC-UDP and DoS-SlowHTTPTest tie at 1.0000 across all methods except ADASYN, while DoS-GoldenEye reaches 0.9999 across all methods except ADASYN and B-SMOTE. The one subset carrying real headroom is also one where PASG is among the leaders.

Table 8. Best method by attack subset at the native class ratio

Attack Subset	Best Method(s)	F1-Minority	F1-Weighted	Precision	Recall
BruteForce-Web	B-SMOTE	0.9855	0.9912	0.9792	0.9920
BruteForce-XSS	B-SMOTE	0.9833	0.9900	0.9837	0.9833
DDOS-LOIC-UDP	All except ADASYN (tie)	1.0000	1.0000	1.0000	1.0000
DoS-GoldenEye	All except ADASYN, B-SMOTE (tie)	0.9999	1.0000	1.0000	0.9999
DoS-SlowHTTPTest	All except ADASYN (tie)	1.0000	1.0000	1.0000	1.0000
DoS-Slowloris	ADASYN	0.9992	0.9995	0.9994	0.9990
Thursday-01-03	B-SMOTE and PASG (tie)	0.4873	0.6693	0.4476	0.5373

Note: ADASYN = Adaptive Synthetic Sampling; DoS = denial-of-service; PASG = Pattern Aware Synthetic Generation.

Table 9. Friedman tests and Nemenyi average ranks

Classifier	Blocks	Chi-squared	*p*-Value	Critical Difference	Average Ranks (1 = Best)
Random forest	24	16.898	0.00966	1.839	B-SMOTE 3.08, ADASYN 3.35, PASG 3.85, SMOTE 4.04, SMOTE-ENN 4.52, SMOTE-Tomek 4.54, Baseline 4.60
Gradient boosting	24	27.334	0.00013	1.839	ADASYN 2.88, B-SMOTE 3.52, PASG 3.81, SMOTE-Tomek 4.10, SMOTE 4.23, Baseline 4.46, SMOTE-ENN 5.00
Logistic regression	24	29.872	0.00004	1.839	B-SMOTE 2.92, Baseline 3.54, ADASYN 3.58, PASG 3.73, SMOTE 4.38, SMOTE-Tomek 4.50, SMOTE-ENN 5.35

Note: PASG = Pattern-Aware Synthetic Generation; B-SMOTE = Borderline-SMOTE; ADASYN = Adaptive Synthetic Sampling; SMOTE = Synthetic Minority Oversampling Technique; SMOTE-ENN = SMOTE combined with Edited Nearest Neighbours; SMOTE-Tomek = SMOTE combined with Tomek-link removal.

5.6 Statistical analysis

The Friedman tests and Nemenyi critical differences at the native ratio are reported in Table 9. The effect of the method on the minority class F1 is statistically significant for all three families of classifiers. There is no separation for Nemenyi between any pair of the classifiers: the widest rank difference between PASG and B-SMOTE under the random forest is 0.77, while the critical difference is 1.839. With Bonferroni only PASG against SMOTE-ENN under logistic regression remains and we report it in full since it is a qualification rather than support of the ranking.

At the severe levels the Friedman test rejects the null at 1:50 ($p = 0.030$, 17 blocks) and at 1:100 ($p = 0.002$, 24 blocks), with PASG holding the best average rank at both. Nemenyi again cannot localize the difference: at 1:100 PASG ranks 3.19 against 3.60 for B-SMOTE, a gap of 0.41 where the critical difference is 1.839. What the evidence supports is that the methods differ as a group at severe imbalance and that PASG holds the leading rank. It does not support a pairwise claim: under the Bonferroni-adjusted threshold of 0.0083 no comparison against PASG reaches significance on any classifier, and establishing one would need a considerably larger panel of data.

5.7 Selection behavior

The vocabulary of the selector is small: it selected SMOTE for almost all the cases, and B-SMOTE for the remaining cases, and at no class ratio selected anything else. Most outcomes were determined by the reliability gate, rather than the utility comparison: 86 times at the native ratio, 9 times at the S1 ratio, and 10 times when the margin was exceeded, a focus that is taken up in Section 6.

How can SMOTE and B-SMOTE be the sole two results returned by the selector if it is to outrank both of them at 1:50 and 1:100? Since it is returning them in the proper blocks. At 1:50, the selector selected B-SMOTE in five blocks; in all five blocks, B-SMOTE was the better choice; in the remaining twelve blocks, the selector selected SMOTE, the better choice in eleven of these blocks, and the selection was correct in 16 out of 17 blocks. The same pattern is observed at 1:100, where 5 of the 24 blocks are B-SMOTE and 15 are SMOTE, all of which are correct. Neither fixed method matches that: against SMOTE the framework wins five blocks and loses none at 1:50, and against B-SMOTE it wins four and loses one. A method that is right block by block outperforms both methods uniformly, even if it has just two words in its vocabulary. This is how the ranking in Section 5.2 is determined, and it is also how it resolves the apparent conflict with Section 6: at the native ratio the two candidates are close to interchangeable, so choosing between them cannot help, while at severe imbalance they diverge and the choice begins to matter.

In Figure 3, the methods are plotted in the precision-recall plane at the native ratio, and what they purchase is displayed. The Baseline is at high precision and low recall, as would be expected if the imbalance is left untreated; PASG and B-SMOTE shift towards higher recall at a relatively small cost to precision. That is the reason why it is important to shift in this context and why aggregate accuracy is a bad indicator.

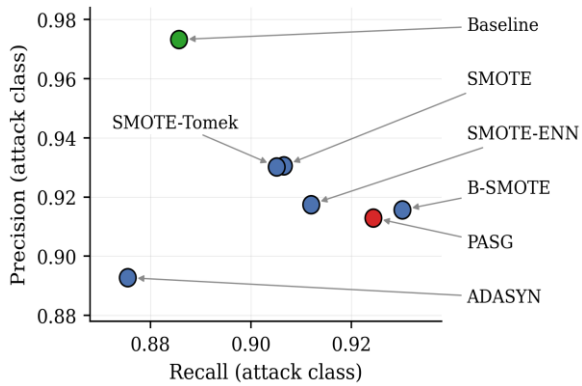


Figure 3. Precision-recall operating points at the native class ratio

6. DISCUSSION

These results have several points that could not be conveyed in the tables.

The primary outcome is a boundary condition, rather than a performance statement. Adaptive selection is only worthwhile if there is a difference in the candidates; the class ratio determines if there is a difference. The headroom for the data below about 1:25 is between 0.0013 and 0.0031; the methods are not distinguishable, and it is better to leave the data alone. The headroom increases by an order of magnitude above about 1:50; the methods branch off, and adaptive selection becomes the top method. The same boundary is used for the practical reading. If a practitioner is working at moderate imbalance, he or she should not anticipate resampling of any sort (adaptive or fixed) to provide much, and the cost of selection will not be recouped. With the rare attack families that are normally in severe imbalance in live traffic, a practitioner has a real margin to play. Headroom is very easy to estimate on a held-out partition, and can be used as a screening test prior to any selection framework being implemented.

The second is about the interpretation of the average scores of the benchmarks. All three attack subsets (SlowHTTPTest, GoldenEye and DDOS-LOIC-UDP) achieve minority-class F1 at or close to 1.0 on most methods, and thus obscure real differences. The imbalance sweep quantifies this: two subsets have no headroom at any ratio tested and two more have no headroom at any ratio tested that exceeds 0.005. Their addition to the overall average raises all methods and narrows the differences between them. The point goes beyond our framework: A significant portion of CIC-IDS2018 cannot distinguish between the resampling methods and averages over the entire benchmark should be interpreted accordingly.

Third, the conservative behavior of the selector is not to be identified by a label, and three findings relate to this. The first is that the performance was found to be nearly flat when performing a sensitivity analysis over the margin threshold, with the random forest having a mean F1 of 0.9180 at the margin threshold of zero, and 0.9178 at our operating value, where the selector misfires in 97% of the cases. The number of regimes should not be the same for two regimes so far apart as these. That they do is not much of a useful piece of information at the native ratio, hence the design's use of three-fold cross-validation and not one holdout. The second is that, while keeping all other parameters the same, replacing SMOTE with B-SMOTE as the default option for the minority class, yields a mean minority-class F1 of 0.0042, which is 10 times the entire threshold effect. The framework is thus running on the default setting and not on an improvement. The third is that at 1:100 the within-block spread is approximately three times greater (0.0317) than at 1:10 and the framework is the top-ranked. A conservative selector remains competitive by not acting on differences that do not exist, and begins to earn its keep when the difference becomes real. The profile is not a strength or weakness, but rather the proper reaction to a signal, which quality is dependent on the operating point.

Fourth, ADASYN shows the trade-off that the framework has to manage, as it was able to complete 24 of 35 tasks, and the failures were not random, but rather occurred on sets where the density weighting of the framework became problematic because the density of the minority was too low. However, at 1:100, ADASYN returns to 0.7794 from 0.7305 at 1:50, suggesting that its aggressive strategy is appropriate in severe imbalance when its conditions are met. If a method is effective when it is used, and it does not work at all when it is not, then it is not a good fit for a security system. The feasibility rules add the applicability condition—making it explicit and machine-checkable.

Fifth, it is best to express the runtime cost in terms of deployment. The overhead is about five times, as inner validation is only performed on the filtered shortlist and not on the entire pool; with the random forest, PASG took 2.67 s per task on average, while SMOTE took 0.53 s per task. It is the cost where it falls that is more important than its size. Selection is done only during training time and inference is not affected as the deployed classifier is identical regardless of which resampler was used during training. The overhead is thus not on detection latency, but rather on the size of the shortlist; it scales with the size of the shortlist, not the volume of traffic. Two additional seconds per model is insignificant for a pipeline that retrains every day, and can be updated less frequently than the model for continuous retraining, as the meta-features of an attack family are stable on short time scales.

Finally, there are two answers to whether or not the cost is warranted, one for each regime. It is not at the native class ratio: B-SMOTE outperforms PASG on all three families of classifiers, with none of the pairwise comparisons being significant, and with no measurable benefit of the additional computation. At this ratio, we make no claim of superiority. At the extreme imbalance, the answer is different, and at 1:50 and 1:100, the mean minority-class F1 is the highest for all seven strategies, while the framework recovers 92.1% and 83.5% of the available headroom, respectively, and has the best average rank for about two additional seconds per training run. There are two caveats that accompany that reading, and neither is insignificant. The ranking is not necessarily significant at the block counts we have, as Nemenyi does not establish pairwise significance. When the effect is not included, the mean headroom at 1:100 is reduced from 0.0146 to 0.0029. The supported conclusion is conditional: adaptive selection is worthwhile when there is significant imbalance, and the data are not saturated, and headroom is the way to identify which datasets are not saturated before investing.

7. LIMITATIONS

Four limitations delimit our findings, the effect at severe imbalance is driven mainly by one attack subset, and all seven subsets come from a single benchmark. the trend holds on a balanced panel and across three classifier families, but a broader collection of non-saturated intrusion detection data, drawn from benchmarks other than this one, is needed before the boundary near 1:50 can be treated as general. Nemenyi distinguishes no pair of methods at any ratio we examined, because the block counts, between 15 and 24, are too small for the observed rank separations to exceed a critical difference of 1.839; our conclusions are therefore framed at the level of the omnibus test and the ranking. The imbalance levels were built by subsampling the minority partition, which isolates the class ratio cleanly but may not capture how a naturally rare attack family differs from a subsampled common one. And the experiments cover binary Benign versus Attack classification only, so that multi-class attack categorization, where overlap and sparsity interact across more than two classes, raises problems this framework does not address.

8. CONCLUSION

This study presented PASG, a reliability-aware adaptive

oversampling framework for intrusion detection under class imbalance. What the evaluation demonstrates is regime-dependent: the value of adaptive selection depends on the imbalance regime. When the imbalance is moderate, the candidates are not statistically distinguishable, doing nothing is competitive, and the attainable margin is very small. At severe imbalance, the methods separate, the margin grows by an order of magnitude, and the framework achieves the top rank among all seven strategies. On this benchmark, that boundary sits near 1:50, although the panel behind the figure is narrow.

The broader takeaway is about "framing" work in imbalanced security learning. Oracle headroom is easy to compute and can be estimated in advance to determine if any selection procedure can assist on specific data. Reporting next to performance would enable the field to differentiate between real methodological progress and progress on "saturated" benchmarks. Our contribution is not a universal 'winner', but an examination of when adaptation is useful, when the problem has reached the saturation point, and how aggressiveness should be balanced against reliability.

Three directions follow from these limits. The first is a wider panel: the boundary near 1:50 was measured on one benchmark, and one attack family carried most of the effect, so replication on non-saturated data from other sources is the immediate priority. The second is a stronger selection signal, since the threshold sensitivity reported in Section 6 shows the inner-validation comparison to be nearly uninformative at moderate imbalance; a signal that degrades more gracefully would let the gate act where it currently defers. The third is a wider pool. Generative candidates such as conditional GANs were left out here because they carry their own training procedure inside each fold, which is not commensurable with single-pass resamplers and would confound the runtime analysis, but a cost-aware gate that admits them is a well-defined extension.

AUTHOR CONTRIBUTIONS

Mohammad M. Rasheed: conceptualization, methodology, software implementation, experimental design, formal analysis, investigation, data curation, visualization, writing of the original draft. Mustafa Muwafak Alobaedy: conceptualization, methodology, validation, critical revision of the manuscript, project administration. Both authors read and approved the final version and agree to be accountable for all aspects of the work.

CONFLICT OF INTEREST

The authors declare no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

DATA AVAILABILITY

The CIC-IDS2018 dataset is publicly available from the Canadian Institute for Cybersecurity [19]; the specific redistribution used here is the ids-intrusion-csv dataset on Kaggle. The experimental code comprises two notebooks, one for the native-ratio protocol of Sections 5.3 to 5.7 and one for the imbalance-regime sweep of Sections 5.1 and 5.2, together with the JSON result files from which every figure and table in this paper was generated. All of these are available from the corresponding author on reasonable request and will be deposited in a public repository on acceptance.

REFERENCES

[1] Ferrag, M.A., Maglaras, L., Moschoyiannis, S., Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50: 102419. https://doi.org/10.1016/j.jisa.2019.102419

[2] Kilincer, I.F., Ertam, F., Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks, 188: 107840. https://doi.org/10.1016/j.comnet.2021.107840

[3] Thabtah, F., Hammoud, S., Kamalov, F., Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. Information Sciences, 513: 429-441. https://doi.org/10.1016/j.ins.2019.11.004

[4] Rezvani, S., Wang, X.Z. (2023). A broad review on class imbalance learning techniques. Applied Soft Computing, 143: 110415. https://doi.org/10.1016/j.asoc.2023.110415

[5] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16: 321-357. https://doi.org/10.1613/jair.953

[6] He, H.B., Bai, Y., Garcia, E.A., Li, S.T. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE International Joint Conference on Neural Networks, Hong Kong, China, pp. 1322-1328. https://doi.org/10.1109/IJCNN.2008.4633969

[7] Santos, M.S., Abreu, P.H., Japkowicz, N., et al. (2022). On the joint-effect of class imbalance and overlap: A critical review. Artificial Intelligence Review, 55: 6207-6275. https://doi.org/10.1007/s10462-022-10150-3

[8] Barella, V.H., Garcia, L.P., de Souto, M.C.P., Lorena, A.C., de Carvalho, A.C.P.L.F. (2021). Assessing the data complexity of imbalanced datasets. Information Sciences, 553: 83-109. https://doi.org/10.1016/j.ins.2020.12.006

[9] Carvalho, M., Pinho, A.J., Brás, S. (2025). Resampling approaches to handle class imbalance: A review from a data perspective. Journal of Big Data, 12: 71. https://doi.org/10.1186/s40537-025-01119-4

[10] Chen, W.X., Yang, K.X., Yu, Z.W., Shi, Y.F., Chen, C.L.P. (2024). A survey on imbalanced learning: Latest research, applications and future directions. Artificial Intelligence Review, 57: 137. https://doi.org/10.1007/s10462-024-10759-6

[11] Luque, A., Carrasco, A., Martín, A., de las Heras, A. (2020). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition, 91: 216-231. https://doi.org/10.1016/j.patcog.2019.02.023

[12] Elreedy, D., Atiya, A.F., Kamalov, F. (2024). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. Machine Learning, 113: 4903-4923.

https://doi.org/10.1007/s10994-022-06296-4
[13] Han, H., Wang, W.Y., Mao, B.H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science, vol 3644. Springer, Berlin, Heidelberg, pp. 878-887. https://doi.org/10.1007/11538059_91
[14] Batista, G.E.A.P.A., Prati, R.C., Monard, M.C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 6(1): 20-29. https://doi.org/10.1145/1007730.1007735
[15] Bunkhumpornpat, C., Boonchieng, E., Chouvatut, V., Lipsky, D. (2024). FLEX-SMOTE: Synthetic over-sampling technique that flexibly adjusts to different minority class distributions. Patterns, 5(11): 101073. https://doi.org/10.1016/j.patter.2024.101073
[16] Rana, S., Kanji, R., Jain, S. (2026). Comprehensive analysis of oversampling techniques for addressing class imbalance employing machine learning models. Recent Advances in Computer Science and Communications, 19(2): e26662558347788. https://doi.org/10.2174/0126662558347788241127051934
[17] Yi, X.K., Xu, Y.Y., Hu, Q., Krishnamoorthy, S., Li, W., Tang, Z.Z. (2022). ASN-SMOTE: A synthetic minority oversampling method with adaptive qualified synthesizer selection. Complex & Intelligent Systems, 8: 2247-2272. https://doi.org/10.1007/s40747-021-00638-w
[18] Bej, S., Davtyan, N., Wolfien, M., Nassar, M., Wolkenhauer, O. (2021). LoRAS: An oversampling approach for imbalanced datasets. Machine Learning, 110: 279-301. https://doi.org/10.1007/s10994-020-05913-4
[19] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, Funchal, Portugal, pp. 108-116. https://doi.org/10.5220/0006639801080116
[20] Uwazie, E.C., Obiniyi, A.A., Olalere, M. (2024). Handling class imbalance in intrusion detection dataset using undersampling and oversampling techniques. African Journal of Computing and ICT, 17(2): 24-33.
[21] Abdelkhalek, A., Mashaly, M. (2023). Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning. The Journal of Supercomputing, 79: 10611-10644. https://doi.org/10.1007/s11227-023-05073-x
[22] Bounab, R., Zarour, K., Guelib, B., Khlifa, N. (2024). Enhancing medicare fraud detection through machine learning: Addressing class imbalance with SMOTE-ENN. IEEE Access, 12: 54382-54396. https://doi.org/10.1109/ACCESS.2024.3385781
[23] Allagi, S., Pawan, T., Leong, W.Y. (2025). Enhanced intrusion detection using conditional-tabular-generative-adversarial-network-augmented data and a convolutional neural network: A robust approach to addressing imbalanced cybersecurity datasets. Mathematics, 13(12): 1923. https://doi.org/10.3390/math13121923
[24] Gu, Y.H., Yang, Y., Yan, Y., Shen, F., Gao, M.N. (2023). Learning-based intrusion detection for high-dimensional imbalanced traffic. Computer Communications, 212: 366-376. https://doi.org/10.1016/j.comcom.2023.10.018
[25] Tabassum, A., Erbad, A., Lebda, W., Mohamed, A., Guizani, M. (2022). FEDGAN-IDS: Privacy-preserving IDS using GAN and federated learning. Computer Communications, 192: 299-310. https://doi.org/10.1016/j.comcom.2022.06.015
[26] Talukder, M.A., Hasan, K.F., Islam, M.M., et al. (2023). A dependable hybrid machine learning model for network intrusion detection. Journal of Information Security and Applications, 72: 103405. https://doi.org/10.1016/j.jisa.2022.103405
[27] Alzughaibi, S., El Khediri, S. (2023). A cloud intrusion detection systems based on DNN using backpropagation and PSO on the CSE-CIC-IDS2018 dataset. Applied Sciences, 13(4): 2276. https://doi.org/10.3390/app13042276
[28] Rodríguez, M., Alesanco, Á., Mehavilla, L., García, J. (2022). Evaluation of machine learning techniques for traffic flow-based intrusion detection. Sensors, 22(23): 9326. https://doi.org/10.3390/s22239326
[29] Ahmad, I., Ul Haq, Q.E., Imran, M., Alassafi, M.O., AlGhamdi, R.A. (2022). An efficient network intrusion detection and classification system. Mathematics, 10(3): 530. https://doi.org/10.3390/math10030530

NOMENCLATURE

CC	correlation complexity, proportion of feature pairs with $\|r\| > 0.7$
CT	clustering tendency of the minority partition, Hopkins statistic
VU	variance uniformity across feature dimensions
DD	data density, ratio of dimensionality to minority sample count
IR	imbalance ratio, majority to minority sample count
F	number of stratified folds used in inner cross-validation
$J(m)$	validation utility of candidate method m, minority-class F1
M	meta-feature vector computed on the training partition
$n_{\min}$	number of minority instances in the training partition
S	shortlist of feasible candidate methods

Greek symbols

τ	margin threshold of the reliability gate

Subscripts

min	minority class
tr / val	inner training and validation partitions