SynthBGP: Synthetic BGP Traffic Generation for Enhanced Cybersecurity Anomaly Detection

Shadi Motaali, Jorge E. López de Vergara, Luis de Pedro and Iván González
Escuela Politécnica Superior, Universidad Autónoma de Madrid (UAM), Spain
{shadi.motaali, jorge.lopez_vergara, luis.depedro, ivan.gonzalez}@uam.es

***Abstract*—Border Gateway Protocol (BGP) anomalies cause large-scale Internet disruptions, yet the extreme scarcity of labeled training data constrains machine learning-based detection methods. Existing approaches either ignore protocol semantics or require prohibitive expert effort. This paper systematically evaluates synthetic BGP traffic generation for anomaly detection. We propose a consensus-based multi-level labeling pipeline combining five unsupervised detectors to create high-confidence labels from RIPE collector streams, and benchmark thirteen generators from five families (rule-based, deep generative, oversampling, hybrid, statistical) using a protocol-aware, 16-metric fidelity framework. SMOTE_kmeans achieves 97.5% in-distribution fidelity and preserves detector performance (F1 $>$ 0.99), but cross-collector evaluation reveals significant limitations: F1 scores degrade to 0.79, with false negative rates of 35–41% on unseen anomalies. GAN neural generators degrade further under a distribution shift.**

***Index Terms*—BGP security, routing anomaly detection, synthetic data generation, generative models, class imbalance, time-series synthesis, SMOTE, GAN, Copula, fidelity evaluation.**

I. Introduction

Border Gateway Protocol (BGP) is the de facto routing protocol governing traffic exchange between autonomous systems across the Internet. However, BGP's trust-based design and lack of cryptographic validation in many deployments make it vulnerable to routing anomalies that cause widespread outages and misdirection. Notable incidents [1] such as the 2008 YouTube hijack, the 2014 Indosat incident, and recurring route leaks, demonstrate that BGP security remains a critical operational challenge. Machine learning has emerged as a promising approach for automatic anomaly detection [2]–[4], with recent works showing that supervised classifiers and unsupervised methods can achieve high accuracy on labeled datasets. However, practical deployment remains constrained by a fundamental bottleneck: the extreme scarcity of labeled BGP training data.

Public BGP collectors such as RIPE NCC RIS [5] and RouteViews [6] provide raw UPDATE streams at massive scale. However, assigning reliable labels—distinguishing normal behavior from anomalies like prefix hijacks, route leaks, and path manipulations—remains largely manual, expensive, and ambiguous. Unlike intrusion detection, where public datasets (e.g. CIC-IDS2017 [7]) contain thousands of labeled samples, BGP anomalies are documented sporadically through incident reports and expert review, creating severe class imbalance [8]. Existing augmentation approaches fall short: rule-based generation requires substantial domain expertise and does not discover temporal dependencies from data [9], while generic synthetic methods (GANs, SMOTE, Copula) successfully applied to intrusion detection [10], [11] do not enforce BGP-specific protocol constraints, raising questions about realism and downstream utility for control-plane anomaly detection.

To address this gap, we conduct a systematic evaluation of synthetic BGP traffic generators for anomaly detection, focusing on how different synthesis approaches generalize across collectors and incident types. In contrast to prior studies that either evaluate synthetic data on generic intrusion-detection benchmarks [10] or assess BGP anomaly detection without a broad comparison of synthesis methods, our work explicitly unifies both problems: it benchmarks thirteen generators on real BGP UPDATE streams using a protocol-aware evaluation framework and tests whether the resulting synthetic data remains useful for downstream detection under distribution shift. The results reveal both the promise and limitations of current synthesis methods for scaling BGP anomaly detection datasets.

This work makes three contributions that distinguish it from prior synthetic-data and BGP anomaly-detection studies: (1) a consensus-based labeling pipeline combining five unsupervised detectors on raw RIPE streams, validated via clustering (silhouette score 0.965) and distributional separation; (2) a benchmark of thirteen generators from five families using a 16-metric, feature-weighted evaluation; and (3) cross-collector and cross-incident analysis quantifying synthesis degradation under distribution shift.

The remainder of this paper is organized as follows. Section II reviews background and related work. Next, Section III details the dataset, preprocessing, and synthetic generation methods. After this, Section IV presents the experimental results. Finally, Section V concludes the paper and identifies future research lines.

II. Background and Related Work

This section reviews BGP security challenges and surveys synthetic data generation approaches relevant to our work.

A. BGP Security and Labeled Data Scarcity

The Border Gateway Protocol (BGP) remains vulnerable to routing anomalies due to its trust-based design [12]. BGP anomalies manifest in several forms: *prefix hijacks*, where an Autonomous System (AS) falsely announces IP prefixes belonging to another organization; *route leaks*, where an AS improperly propagates announcements beyond their intended

scope; *path manipulations*, where ASes modify AS-PATH attributes to influence routing decisions; and *denial-of-service (DoS) attacks*, including UPDATE flooding and convergence exploitation [1], [13]. These incidents have repeatedly caused large-scale outages affecting millions of users over the years.

Cryptographic defenses such as Resource Public Key Infrastructure (RPKI) and BGPsec provide origin and path validation [14], [15]. However, RPKI deployment remains incomplete—covering approximately 60% of announced IPv4 prefix-origin pairs as of January 2025 [16]—while BGPsec adoption is negligible due to computational overhead. Consequently, anomaly detection through traffic analysis remains essential for operational security.

Machine learning has been widely explored for BGP anomaly detection. However, progress is limited by the availability of labeled data. While public collectors such as RIPE NCC RIS [5] and RouteViews [6] provide BGP UPDATE streams at Internet scale, ground-truth labeling remains incident-driven and expert-dependent, resulting in sparse and highly imbalanced datasets [3], [13].

B. Synthetic Data for Cybersecurity

Previous work addressed data scarcity through several generator families. **Rule-based approaches** such as Scapy [17] and network simulators (ns-3) [18] enable explicit protocol modeling with guaranteed compliance. Recent work extends Scapy with an RFC-compliant BGP framework, reporting 96.88% detection accuracy on synthetic traces for BGP anomaly detection [9]. However, these evaluations are conducted solely on Scapy-generated traffic and do not assess how closely the resulting feature distributions resemble those extracted from real BGP UPDATE streams.

Deep generative models learn sequential structure from data. LSTM-GAN captures temporal dependencies [19], TimeGAN [20] jointly optimizes supervised and adversarial objectives, and DoppelGANger [21] handles mixed continuous-discrete features. However, these models require substantial training data, careful tuning, and provide no protocol-correctness guarantees [10].

Oversampling methods operate in feature space. SMOTE variants (Borderline-SMOTE, ADASYN, k-means-SMOTE) generate synthetic samples through interpolation [11], [22], while **statistical methods** like Copula [23] model multivariate distributions. These methods are computationally efficient but do not capture temporal evolution or protocol semantics.

Ammara et al. [10] compared twelve generators on NSL-KDD and CIC-IDS2017, showing the value of synthetic-data benchmarking for generic intrusion detection. Our work differs in four substantive ways for the BGP setting: (1) it evaluates generators on real BGP UPDATE streams rather than conventional host/network intrusion datasets; (2) it uses a consensus-based multi-level labeling pipeline combining five unsupervised detectors, validated through distributional analysis; (3) it explicitly tests cross-collector and cross-incident generalization under distribution shift; and (4) it assesses whether synthetic data preserves downstream detector calibration and utility, not only feature-level fidelity.

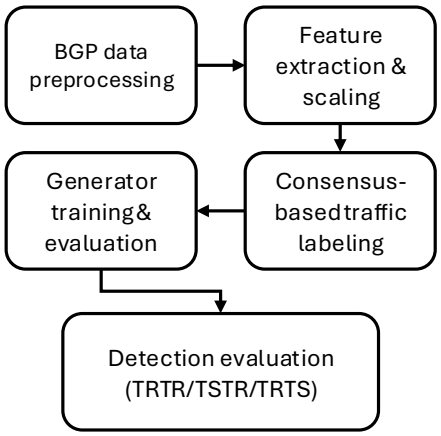

Fig. 1. End-to-end workflow of the proposed methodology.

III. Methodology

This section introduces the data sources, feature representation, labeling strategy, synthetic BGP traffic generation methods, and detection models used in this study. Figure 1 summarizes the following phases.

A. Dataset, Features, and Labeling

We use BGP UPDATE streams from multiple RIPE [5] collectors over selected background and incident periods, focusing on IPv4 unicast updates and excluding malformed or incomplete messages. UPDATEs are aggregated into non-overlapping 1-second sliding windows (chosen to balance temporal resolution with sufficient BGP events per window for statistical feature extraction). For each window, we extract 27 statistical features [8] across four groups: (i) *volume features* (counts of announcements, withdrawals, NLRI entries); (ii) *path features* (AS_PATH length statistics, number of unique paths, edit distances between consecutive paths); (iii) *stability features* (origin AS changes, route flaps, implicit withdrawals); and (iv) *prefix/rarity features* (duplicate announcements, presence of rare ASes, prefix-level statistics). All features are transformed with a RobustScaler [24] fitted exclusively on benign traffic to mitigate heavy tails and outliers, and the same scaling parameters are reused across all stages.

To obtain training labels, we follow a two-stage consensus-based pipeline for normal and anomalous traffic. For normal traffic, we first collect a full day of BGP UPDATE messages from a RIPE RIS collector on a day with no documented routing incidents, and aggregate them into 1-second windows. An ensemble of five unsupervised detectors (Isolation Forest [25], Local Outlier Factor [26], robust Z-score with threshold 3.0 and interquartile range(IQR) rules with multiplier 1.5 [24], Elliptic Envelope based on Minimum Covariance Determinant [27], and HDBSCAN outlier scores with `min_samples`=5 [28]) is then applied to these windows. Each detector outputs a binary anomaly flag, and

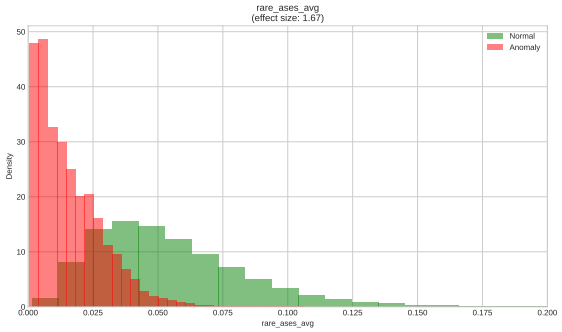

Fig. 2. Feature distribution comparison for `rare_ases_avg`. Anomalous traffic (red) shows strong concentration in the interval [0, 0.05] (peak value ≈50 near zero), while normal traffic (green) exhibits a broader distribution extending to 0.15, demonstrating statistical separation (Cohen's $d = 1.67$, large effect).

windows classified as benign by at least four out of five methods are retained as *high-confidence normal*, forming the benign baseline used in subsequent steps.

For anomalous traffic, we curate incident windows using publicly reported incidents that specify the start and end time, affected IP prefixes, and involved ASes for prefix hijacks, path manipulations, and denial-of-service events. This provides partial external grounding for the anomaly labels at the incident level. All BGP UPDATE windows overlapping these intervals and matching the reported prefixes/ASes are initially marked as candidate anomalies. We then refine these labels by scoring the candidate windows against the high-confidence normal baseline using multiple complementary measures: Mahalanobis distance in the feature space defined by the normal mean and covariance, One-Class SVM [29], reconstruction error of an autoencoder trained only on normal windows [30], aggregated feature-wise Z-scores, and LOF-based density contrast relative to normal traffic. These scores are further checked for incident-level temporal coherence (e.g., sustained deviation during the reported attack interval) to avoid isolated false positives. Finally, all signals are fused into a single confidence score that yields multi-level *validated anomaly* labels (very-high, high, medium, low, needs-review), and only windows above a chosen confidence threshold are included in the anomaly dataset.

Label Validation. We validate labeling through cluster and distributional analysis. K-means clustering achieves a silhouette score of 0.965 at $k = 2$, confirming class separation. This separation is further supported by agreement across multiple detectors, temporal coherence within reported incident windows, and alignment with incident metadata from public reports.Figure 2 illustrates the distribution of `rare_ases_avg`: normal windows concentrate near zero, while anomalous windows exhibit heavy-tailed distributions (effect size $d = 1.67$). Uncertain windows (23.8%) are excluded from training.

B. Synthetic Traffic Generation

All synthetic generators operate on the same normalized 27-dimensional feature space described in Section III-A, trained on different subsets: a high-confidence normal pool from RIPE route collector RRC04 background streams and a validated anomaly set from 30 documented BGP incidents (prefix hijacks, path manipulations, denial-of-service attacks). For evaluation, models are tested on held-out RRC04 windows (in-distribution) and on an independent collector RRC05 with 40 non-overlapping incidents (out-of-distribution).

We compare thirteen generation methods grouped into five categories:

(1) Rule-based: A Scapy-based script generator [9] produces protocol-compliant BGP UPDATE sequences for normal and attack scenarios without learning from data.

(2) Deep generative models: We evaluate LSTM-GAN, TimeGAN [31], and DoppelGANger [32] in both default literature-based and enhanced tuned configurations. The enhanced variants use longer training, larger latent representations, and additional distribution-matching losses to better preserve sparse and correlated BGP feature patterns. Full architecture, optimizer, and training settings are provided in the public repository[1].

(3) Oversampling: Four SMOTE variants (SMOTE_normal, SMOTE_borderline, SMOTE_kmeans with k=5 neighbors, SMOTE_adasyn) [11] perform statistical oversampling in feature space.

(4) Hybrid: SMOTE–GAN applies SMOTE_kmeans (k=5) followed by the best GAN training on oversampled data.

(5) Statistical: Gaussian Copula [23] fitted separately to normal and anomaly pools with empirical marginals.

1) Post-Processing for Protocol Compliance: Generators do not inherently distinguish between integer-valued and continuous features. Before training, we explicitly mark routing-count dimensions (25 integer features: announcements, withdrawals, flap counters, AS/prefix counts, edit distance bins) as integer-valued and 2 continuous features (editdistanceavg, rareasesavg) as real-valued. Additionally, 9 sparse features (editdistancedict3-6, impwdspath, uniqueaspathmax, origin-changes, flaps, nadas) undergo log1p transformation before normalization to handle heavy-tailed distributions. After sampling, we apply post-processing that (1) inverse-transforms sparse features via expm1, (2) denormalizes all features using the fitted RobustScaler, (3) clips values to valid ranges (≥ 0), and (4) rounds integer features to nearest admissible values, ensuring generated feature vectors correspond to plausible BGP UPDATE windows.

Each method produces synthetic normal and anomaly datasets evaluated using a 16-metric fidelity framework (Section III-E) in in-distribution (RRC04, base incidents) and out-of-distribution (RRC05, extended incidents) settings. Based on these metrics (Table I), SMOTE_kmeans and Copula are selected as the two best generators: SMOTE_kmeans achieves the highest in-distribution fidelity (KS-all=0.1083, KS-sparse=0.0515, overall score=0.0697) and decision-boundary preservation, while Copula offers superior robustness under cross-collector and cross-incident shifts. These two generators are used in final detection experiments.

C. Detection Models and Training Regimes

To study the effect of synthetic data on BGP anomaly detection, we evaluate three standard tabular classifiers: Random Forest, LightGBM, and XGBoost [33]–[35]. These models are widely used strong baselines for imbalanced network-security classification tasks [8], [36].

For each classifier and regime, hyperparameters were tuned using Bayesian optimization with a Tree-structured Parzen Estimator (TPE) sampler and stratified 5-fold cross-validation, maximizing mean F1 score. The best configurations obtained across regimes were: *Random Forest* with $n_{\text{estimators}} = 274$, `max_depth` = 16–19, `max_features` = `sqrt`, and `class_weight` = `balanced`; *LightGBM* with $n_{\text{estimators}} =$ 180–210, `max_depth` = 5–8, and `learning_rate` = 0.14–0.20; and *XGBoost* with $n_{\text{estimators}} =$ 230–259, `max_depth` = 6–8, `learning_rate` = 0.12–0.18, `reg_alpha` = 0.1, and `reg_lambda` = 1.0. Complete per-regime hyperparameter settings and code are provided in the public repository for reproducibility[1].

Detection performance is evaluated under three regimes: TRTR (train-on-real/test-on-real), TSTR (train-on-synthetic/test-on-real), and TRTS (train-on-real/test-on-synthetic). We report both in-distribution results on RRC04 (30 incidents) and out-of-distribution results on RRC05 (40 incidents) to quantify the effect of distribution shift.

For all evaluations, we consider two scenarios: (i) in-distribution (RRC04, base 30 incidents), and (ii) out-of-distribution (RRC05, extended 40 incidents). This dual evaluation reveals how generator quality and detection performance change under distributional shift.

D. Computing Environment

Deep generative models (LSTM-GAN, TimeGAN, DoppelGANger) and the hybrid SMOTE–GAN were trained on an NVIDIA A100 GPU (40 GiB HBM2) with dual AMD EPYC 7F72 processors and 1.0 TiB RAM. Training times ranged from 30 minutes to 1 hour for default configurations to 1–2 hours for enhanced configurations (250 epochs). SMOTE variants, Copula, Scapy, and all classifiers were executed on CPU (Intel Xeon, 128 GiB RAM), completing in under 30 minutes per generator.

E. Evaluation Metrics

To compare generators in a model-agnostic way, we adopt a 16-metric evaluation framework capturing univariate and multivariate similarity between real and synthetic BGP windows. **Distributional fidelity:** mean Kolmogorov–Smirnov (KS) statistics, mean and feature-importance-weighted (FIW) Wasserstein distances, counts of features with KS statistics below 0.05 (excellent) and 0.1 (good); statistical significance is assessed via FDR-corrected (Benjamini–Hochberg) KS p-values at $\alpha = 0.05$. **Correlation structure:** Pearson and Spearman correlation-matrix similarities computed as $1 - \|\mathbf{C}_{\text{real}} - \mathbf{C}_{\text{synth}}\|_F / \|\mathbf{C}_{\text{real}}\|_F$. **Multivariate geometry:** PCA-centroid Euclidean distance between real and synthetic data in the first 10 principal components. **Effect sizes:** Cohen's d per feature (pooled standard deviation with numerical capping), categorized as negligible ($|d| < 0.2$), small ($0.2 \leq |d| < 0.5$), medium ($0.5 \leq |d| < 0.8$), or large ($|d| \geq 0.8$). These components are aggregated—each weighted by operational importance of BGP features (e.g., announcements / withdrawals / flaps carry 1.3–1.5× weight)—into normalized 0–100 scores for distribution, correlation, effect size, Wasserstein, PCA, and an overall FIW similarity score reported in Section IV.

[1]https://github.com/shadimotaali/BGP_Traffic_Generation_Comparision

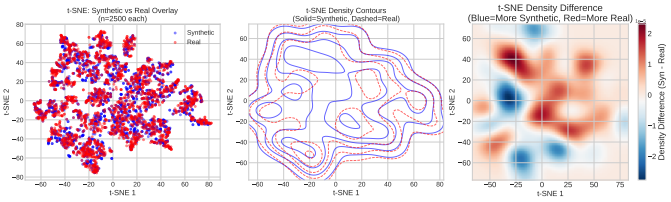

Fig. 3. t-SNE embedding of real and SMOTE_kmeans-generated anomalous BGP windows from the same dataset.

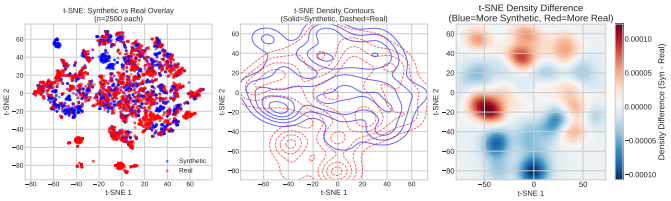

Fig. 4. t-SNE embedding of real and SMOTE_kmeans-generated anomalous BGP windows from the cross dataset.

Also, t-SNE visualizations (figures 3 and 4) complement quantitative metrics by showing how well synthetic samples overlay real BGP clusters in low-dimensional space.

For detection performance, we report F1 score, AUROC, AUPRC, accuracy, precision, recall, Brier score (mean squared error of predicted probabilities), and expected calibration error (ECE, maximum deviation of predicted vs. observed anomaly frequency across 10 equal-probability bins). Confusion matrices enable detailed analysis of false-positive and false-negative trade-offs under TRTR, TSTR, and TRTS regimes for both SMOTE_kmeans and Copula.

IV. Results

This section reports how well each generator reproduces the statistical structure of real BGP traffic. We first analyze benign (normal) windows and then BGP anomalies, using the 16-metric FIW evaluation framework described in Section III.

A. Fidelity for Normal BGP Traffic

Table I summarizes the main fidelity metrics for synthetic benign traffic on both the RRC04 same-dataset and RRC05 cross-dataset settings. SMOTE_kmeans attains the highest overall FIW similarity score (97.5/100 on RRC04), combining near-perfect Pearson correlation (0.996), minimal PCA-centroid distance (0.15), and the lowest mean KS statistic (0.026). Copula forms a strong second tier (76.4/100), preserving correlation structure (Pearson 0.825) while showing slightly higher distributional divergence (mean KS 0.076). The cross-collector evaluation (RRC05) reveals that Copula

maintains the most consistent performance under distribution shift (42.4/100), outperforming SMOTE_kmeans (33.6/100), whose Pearson correlation degrades from 0.996 to 0.790 and PCA distance increases from 0.15 to 4.25.

B. Fidelity for BGP Anomalies

Table I reports the same metrics for synthetic anomalies evaluated on the 30 training incidents (same-incident setting). SMOTE_kmeans achieves the highest overall score (96.6/100), combining low mean KS (0.026) and FIW Wasserstein distance (0.136) with near-perfect correlation (Pearson 0.970) and minimal PCA-centroid shift (0.136). Effect-size analysis confirms that 20/27 features exhibit negligible Cohen's d ($|d| < 0.2$), with only one medium-effect feature (`rareasesavg`, $d = -0.596$). Copula and SMOTE_adasyn follow closely (94.5 and 92.7/100), preserving multivariate structure with slightly worse univariate matching.

Across both benign and anomalous settings, all generators degrade under temporal and collector shift (cross-collector / cross-incident), but Copula and SMOTE_kmeans preserve the largest fraction of their same-dataset scores. Copula shows the most consistent cross-dataset behavior on anomalies (68.7/100 cross-incident vs. 94.5/100 same-incident), likely because its copula-based dependency modeling captures the fundamental rank-order relationships among BGP features that remain stable across different routing incidents [37], [38], independent of the specific AS paths or prefix distributions observed in the training set.

In contrast, SMOTE_kmeans exhibits sharper degradation (68.7/100 cross-incident vs. 96.6/100 same-incident) as its synthetic neighbors inherit the specific feature combinations and local geometry of the training incidents, which may not generalize to anomalies with different topological signatures or update patterns. Neural generators (LSTM, TimeGAN, DoppelGANGER) struggle even more to generalize, with overall scores dropping to 20–52/100 under distribution shift, reflecting their tendency to overfit temporal patterns and feature correlations specific to the training collector routing table dynamics [39], peering relationships [40], and policy configurations [41]—all of which vary substantially across RIPE RIS vantage points and evolve as network operators adjust route filters, implement prefix aggregation, or respond to operational events [42].

C. Detection Utility with Real and Synthetic Training

To assess how well synthetic data preserves the decision boundary between benign and anomalous BGP behavior, we compare the two best generators (SMOTE_kmeans and Copula) in combination with three classifiers (Random Forest, LightGBM, XGBoost) under three training–testing regimes: train-on-real/test-on-real (TRTR), train-on-synthetic/test-on-real (TSTR), and train-on-real/test-on-synthetic (TRTS). Although Copula attains higher cross-dataset fidelity in the purely statistical evaluation, SMOTE_kmeans consistently yields better downstream detection performance across all three classifiers, and is therefore selected as the primary generator for detailed analysis. These findings were further corroborated using information-theoretic metrics (Jensen–Shannon and Kullback–Leibler divergence), which yielded consistent feature rankings with our statistical evaluation. The main results are summarized in Table II, which reports F1, AUROC, AUPRC, accuracy, precision, recall, and calibration metrics (Brier score, Expected Calibration Error) for all regimes, generators, and models.

For the TRTR baseline (train and test on real data), all three classifiers achieve very high accuracy (> 0.995) in the seen data setting and strong performance (≈ 0.986–0.990) on unseen data, with false-positive rates < 0.003 (seen) and < 0.008 (unseen), indicating that distribution shifts across collectors and incidents primarily manifest as a modest increase in missed anomalies and, to a lesser extent, in spurious alerts (Figure 5).

For the TSTR regime (train on SMOTE_kmeans, test on real), Figure 5 summarizes the result of confusion matrices, showing that synthetic training behaves very differently on seen and unseen data. On the *seen* dataset, all three classifiers produce near-diagonal matrices: false-positive rates remain < 0.005, false-negative rates < 0.007, and accuracies ≈ 0.995–0.996, indicating that SMOTE_kmeans can effectively support detector training when the real traffic closely matches the distribution used to generate the synthetic data. In contrast, on the *unseen* dataset, the dominant error type becomes false negatives: Random Forest, LightGBM, and XGBoost misclassify 35–41% of anomalies as benign (FNR ≈ 0.35–0.41), while false-positive rates remain very small (< 0.006), leading to accuracies in the 0.79–0.83 range. This pattern confirms that SMOTE_kmeans-based training yields very conservative detectors under distribution shift—rarely raising spurious alarms on benign traffic but missing a substantial fraction of previously unseen anomalies.

For the TRTS regime (train on real, test on SMOTE_kmeans-generated data), results indicate that detectors calibrated on real incidents treat most synthetic anomalies as genuinely suspicious. On the *seen* dataset, all three classifiers yield almost perfectly diagonal matrices: false-positive rates stay at < 0.001, false-negative rates < 0.005, and accuracies ≈ 0.997–0.998, meaning that both benign and synthetic anomalous windows are classified correctly in nearly all cases. On the *unseen* dataset, recall remains very high, but precision degrades: false-negative rates increase to approximately 0.07–0.15 while false-positive rates remain very small (< 0.003), and accuracies fall into the 0.93–0.96 range, showing that most synthetic anomalies still fall inside the decision regions induced by real incidents, but some harder anomalies are now missed. Together with the TSTR results, these TRTS confusion matrices (summarized in Figure 5) confirm that SMOTE_kmeans generates traffic that is generally recognized as anomalous by real-trained detectors, while highlighting that distribution shifts primarily manifest as a trade-off between a small increase in missed synthetic anomalies and a moderate loss of overall accuracy on unseen conditions.

TABLE I
SUMMARY OF SYNTHETIC TRAFFIC FIDELITY FOR NORMAL TRAFFIC (RRC04/RRC05) AND ANOMALY TRAFFIC (SAME/CROSS INCIDENT)

Method	Fidelity Evaluation															
	Normal Traffic								Anomaly Traffic							
	Mean KS		Pearson Corr.		PCA Centroid Distance		Overall		Mean KS		Pearson Corr.		PCA Centroid Distance		Overall	
	04	05	04	05	04	05	04	05	same	cross	same	cross	same	cross	same	cross
Copula	0.0757	0.4903	0.8248	0.7771	1.2840	2.6858	76.4	42.4	0.0093	0.1787	0.9699	0.7451	0.1891	0.9045	94.5	68.7
SMOTE_kmeans	0.0257	0.5602	0.9960	0.7895	0.1507	4.2495	97.5	33.6	0.0257	0.2022	0.9698	0.7168	0.1361	0.9209	96.6	68.7
SMOTE_normal	0.0490	0.5751	0.9794	0.7819	0.4188	4.3691	93.8	33.2	0.0446	0.1994	0.8359	0.5962	0.2823	0.9871	92.7	65.8
SMOTE_adasyn	0.0490	0.5770	0.9816	0.7821	0.4408	4.3710	93.7	32.3	0.0445	0.1986	0.8304	0.5866	0.2695	0.9706	92.7	65.6
SMOTE_borderline	0.0866	0.5964	0.8677	0.7300	0.6994	4.5003	80.8	31.3	0.1260	0.1886	0.6014	0.6520	1.1245	0.6278	60.6	62.1
LSTM_default	0.5509	0.7010	0.2145	0.2494	3.9555	4.3492	20.9	22.8	0.3407	0.3573	0.2036	0.3108	1.0488	0.8474	48.5	50.2
TIMEGAN_default	0.1091	0.5449	0.8524	0.6658	0.2413	4.3027	79.4	29.7	0.4282	0.4582	0.4548	0.3557	0.2483	0.9428	63.5	51.6
DoppelGanger_default	0.1372	0.6225	0.9051	0.7333	1.0383	4.4453	71.1	31.0	0.2146	0.2865	0.3816	0.2419	0.4538	0.8731	66.7	57.5
LSTM_tunned	0.1084	0.5327	0.9173	0.7236	0.3392	4.1466	82.3	33.5	0.3557	0.3724	0.6255	0.4339	2.4776	2.3466	50.6	45.7
TIMEGAN_tunned	0.2964	0.7019	0.5761	0.5222	1.4794	4.8903	46.3	20.8	0.3697	0.4565	0.7126	0.5700	1.3421	1.0367	54.3	53.5
DoppelGanger_tunned	0.1003	0.5671	0.9671	0.7376	0.4413	4.2529	88.8	34.4	0.1491	0.2351	0.9051	0.6816	0.6253	1.0303	74.9	65.6
Hybrid	0.1809	0.4503	0.3710	0.3635	1.9006	3.0302	52.9	34.7	0.1177	0.2121	0.4244	0.3754	0.2832	0.9362	80.2	60.8
Scapy	0.7598	-	0.4844	-	5.2907	-	19.9	-	0.5855	-	0.2117	-	2.5155	-	28.6	-

TABLE II
TRTR, TSTR, AND TRTS RESULTS FOR SMOTE_KMEANS ACROSS THREE CLASSIFIERS. METRICS ARE COMPUTED ON REAL TEST DATA (UNSEEN) AND SMOTE_KMEANS-GENERATED SYNTHETIC TEST DATA (SEEN) FOR EACH REGIME.

Regime	Train / Test	Model	F1	AUROC	AUPRC	Accuracy	Precision	Recall	Brier	ECE
TRTR	RIPE_route_collector/unseen_dataset	RandomForest	0.9864	0.9999	0.9997	0.9989	0.9987	0.9992	0.0010	0.0010
TRTR	RIPE_route_collector/unseen_dataset	LightGBM	0.9903	1.0000	1.0000	0.9993	0.9992	0.9995	0.0004	0.0006
TRTR	RIPE_route_collector/unseen_dataset	XGBoost	0.9894	0.9997	0.9995	0.9989	0.9984	0.9995	0.0008	0.0006
TRTR	RIPE_route_collector/seen_dataset	RandomForest	0.9959	0.9999	0.9999	0.9959	0.9972	0.9946	0.0031	0.0012
TRTR	RIPE_route_collector/seen_dataset	LightGBM	0.9980	0.99998	0.99998	0.9980	0.9986	0.9974	0.0016	0.0010
TRTR	RIPE_route_collector/seen_dataset	XGBoost	0.9971	0.99997	0.99997	0.9971	0.9977	0.9965	0.0020	0.0009
TSTR	SMOTE_kmeans/unseen_dataset	RandomForest	0.7886	0.9086	0.9120	0.8245	0.9911	0.6548	0.1568	0.1767
TSTR	SMOTE_kmeans/unseen_dataset	LightGBM	0.7390	0.9021	0.9030	0.7929	0.9990	0.5864	0.1810	0.2020
TSTR	SMOTE_kmeans/unseen_dataset	XGBoost	0.7764	0.8974	0.8988	0.8168	0.9954	0.6364	0.1638	0.18417
TSTR	SMOTE_kmeans/seen_dataset	RandomForest	0.9949	0.9992	0.9992	0.9950	0.9956	0.9944	0.0041	0.0027
TSTR	SMOTE_kmeans/seen_dataset	LightGBM	0.9958	0.9986	0.9985	0.9958	0.9977	0.9939	0.0034	0.0024
TSTR	SMOTE_kmeans/seen_dataset	XGBoost	0.9954	0.9990	0.9989	0.9955	0.9974	0.9935	0.0037	0.0030
TRTS	SMOTE_kmeans/unseen_dataset	RandomForest	0.9203	0.9980	0.9971	0.9260	0.9972	0.8545	0.0455	0.08185
TRTS	SMOTE_kmeans/unseen_dataset	LightGBM	0.9635	0.9974	0.9974	0.9647	0.9994	0.9300	0.0245	0.0417
TRTS	SMOTE_kmeans/unseen_dataset	XGBoost	0.9458	0.9940	0.9940	0.9485	0.9982	0.89867	0.0392	0.0537
TRTS	SMOTE_kmeans/seen_dataset	RandomForest	0.9974	0.99998	0.99998	0.9974	0.9990	0.9957	0.0019	0.0015
TRTS	SMOTE_kmeans/seen_dataset	LightGBM	0.9982	0.99999	0.99999	0.9982	0.9992	0.9972	0.0011	0.0010
TRTS	SMOTE_kmeans/seen_dataset	XGBoost	0.9982	0.99999	0.99999	0.9982	0.9992	0.9972	0.0012	0.0009

D. Discussion and Limitations

Our results consistently show that statistical methods (SMOTE variants, Copula) outperform both deep generative models (LSTM-GAN, TimeGAN, DoppelGANGER) and packet-level generators (Scapy) across all evaluation settings.

Scapy achieved the lowest fidelity scores (19.8–29.6/100) despite generating RFC-compliant BGP messages. This fundamental limitation arises because Scapy operates at the packet level, producing individual UPDATE messages without modeling the complex temporal and topological dependencies that create correlated feature patterns in operational networks. When statistical features are extracted from Scapy-generated traffic, correlation similarity reaches only 48% (Pearson) and 85% of features exhibit large effect-size differences compared to real data. Packet-level generators cannot capture how announcements cluster during convergence events, how withdrawal bursts correlate with path instability, or how rare-AS appearances co-occur with origin changes—patterns that emerge from network-wide routing dynamics rather than individual message construction.

Neural generators (LSTM-GAN, TimeGAN, and DoppelGANGER) perform better than Scapy but still underperform statistical methods due to three factors. First, *sample efficiency*: SMOTE and Copula require only pairwise distances or rank correlations, while neural generators need sufficient data to learn high-dimensional temporal dependencies—a challenge given that labeled BGP anomalies remain scarce [43]. Second, *BGP traffic characteristics*: control-plane data exhibits heavy-tailed distributions and sparse events (e.g., rare AS appearances, infrequent origin changes) that statistical methods preserve through explicit modeling, whereas GANs tend to generate smoother, more averaged outputs [44]. Third, *overfitting risk*: neural generators memorize collector-specific patterns (routing policies, peering relationships, update timing)

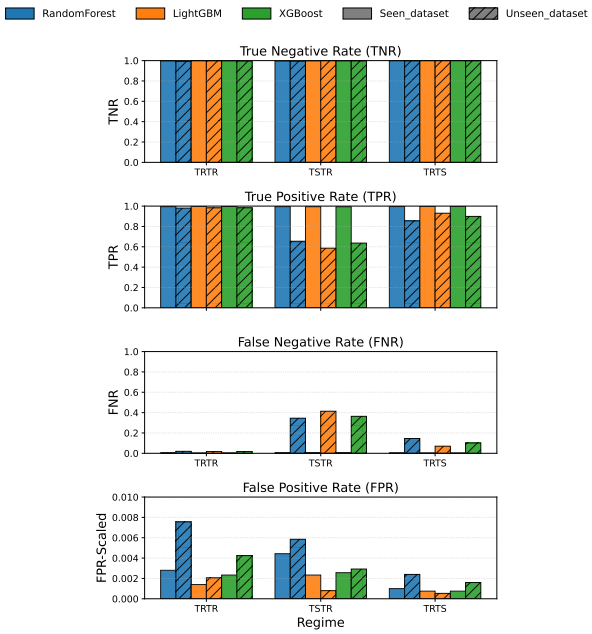

Fig. 5. True negative, true positive, false negative, and false positive rates (TNR, TPR, FNR, and FPR, respectively) for all classifiers under TRTR, TSTR, and TRTS regimes with SMOTE_kmeans, comparing seen (same-collector) and unseen (cross-collector) data. FPR is scaled.

that do not transfer across vantage points [45], while statistical generators focus on aggregate dependencies that are more stable across networks.

In contrast, SMOTE_kmeans and Copula perform well because they better preserve the statistical structure of window-level BGP features. SMOTE_kmeans maintains realistic local neighborhoods, while Copula preserves cross-feature dependencies. Accordingly, SMOTE_kmeans is strongest in-distribution, whereas Copula is more robust under cross-collector shift.

Several limitations should be noted. First, our consensus-based labeling relies on unsupervised detectors and publicly reported incidents, without direct validation from network operators or a formal sensitivity analysis. While the silhouette score (0.965), detector agreement, and feature separability analysis support label quality, some semantic mislabeling may persist. Incorporating operator feedback would strengthen the labeling pipeline and is an important direction for future work. Second, our incident coverage, while spanning 30 events across three anomaly types, may not capture the full diversity of real-world BGP anomalies, especially more heterogeneous or advanced operational scenarios. Third, to isolate the effect of the generation method, all experiments use the same fixed 27-feature set derived from prior BGP anomaly-detection work [8]. A dedicated ablation study to quantify the contribution of individual features or feature families was beyond the scope of this generator-centric evaluation and remains future work. Fourth, the current study is limited to RIPE RIS collectors. Extending the benchmark to additional public BGP sources such as RouteViews [6], as well as private operational feeds, remains an important direction for improving the external generalizability of the results.

V. Conclusion

This work addresses a critical barrier to deploying machine learning for BGP anomaly detection: the scarcity of labeled training data. We present three contributions. First, a consensus-based labeling pipeline that combines five unsupervised detectors to generate high-confidence labels from raw RIPE collector streams, validated through cluster analysis (silhouette score 0.965) and distributional separation. Second, a comprehensive benchmark comparing thirteen generators from five families (rule-based, deep generative, oversampling, hybrid, statistical) using a 16-metric feature-importance-weighted evaluation framework. Third, systematic cross-collector and cross-incident evaluation reveals how synthesis quality degrades under distribution shift.

Our findings yield actionable insights for practitioners. SMOTE_kmeans achieves the highest in-distribution fidelity (97.5/100) and, more importantly, consistently yields the best downstream detection performance across all three classifiers (F1 $>$ 0.99) when training and deployment conditions match. However, cross-dataset evaluation exposes a fundamental limitation: synthetic-trained detectors based on SMOTE_kmeans exhibit conservative behavior on unseen anomalies (FNR = 0.35–0.41), prioritizing low false-positive rates over recall. Accordingly, in live deployments, these models should be periodically retrained or recalibrated using recent collector-specific real data and monitored for distribution drift. Copula attains lower in-distribution fidelity but offers better generalization stability in the purely statistical evaluation (42.4% vs. 33.6% cross-collector), indicating that it better preserves cross-dataset feature similarity, although its downstream detection performance remains weaker than that of SMOTE_kmeans. More broadly, statistical generators appear more practical for deployment than sequence-based neural models because they rely on compact feature vectors and are easier to retrain as new incidents become available. Packet-level generators (Scapy) and neural models fail to capture the correlated feature patterns arising from network-wide routing dynamics, achieving only 19.8–52/100 fidelity.

These results demonstrate that synthetic BGP traffic generation is viable for detector development under matched conditions, but cannot yet replace diverse real-world data for robust cross-network deployment. We release our evaluation framework, labeled datasets, and generator implementations to support reproducible research in control-plane security.

Three directions will extend this work. First, incorporating Routing Information Base (RIB) snapshots would enable topology-aware evaluation through graph-based metrics (e.g., AS-path trees and reachability graphs), complementing the current statistical feature-level fidelity analysis. Second, training generators on multiple collectors simultaneously to improve cross-dataset generalization and reduce the 25–30% accuracy gap observed under distribution shift. Third, integrating BGP protocol invariants (AS-path validity, RPKI

origin validation) directly into the generation process to ensure semantic correctness without post-hoc filtering.

ACKNOWLEDGMENT

This work is partially funded by a grant from the Dept. of Electronics and Communication Technologies at UAM, as well as by the R&D activity program with ref. TEC-2024/COM-504 and the acronym RAMONES-CM, granted by the Comunidad de Madrid, Spain, through the Directorate General for Research and Technological Innovation via Order 5696/2024.

REFERENCES

[1] D. Madory, "A brief history of the internet's biggest BGP incidents," *Kentik Blog*, June 2023. [Online]. Available: https://www.kentik.com/blog/a-brief-history-of-the-internets-biggest-bgp-incidents/

[2] P. Edwards, L. Cheng, and G. Kadam, "Border gateway protocol anomaly detection using machine learning techniques," *SMU Data Science Review*, vol. 2, no. 1, 2019.

[3] K. Hoarau, P. U. Tournoux, and T. Razaindralambo, "Unsupervised representation learning for BGP anomaly detection," *ITU Journal on Future and Evolving Technologies*, March 2024.

[4] O. Elroy and A. Yosipof, "Border gateway protocol hijacks and anomalies detection: A graph-based deep learning approach," in *Artificial Intelligence Applications and Innovations*, I. Maglogiannis, L. Iliadis, A. Andreou, and A. Papaleonidas, Eds. Cham: Springer Nature Switzerland, 2025, pp. 75–86.

[5] RIPE Network Coordination Centre, "RIPE routing information service (RIS)," https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris.

[6] U. of Oregon, "Routeviews project," http://www.routeviews.org/.

[7] Canadian Institute for Cybersecurity, "Intrusion detection evaluation dataset (cic-ids2017)," https://www.unb.ca/cic/datasets/ids-2017.html.

[8] S. Motaali, J. E. López de Vergara, and L. de Pedro, "Hybrid feature selection and explainable machine learning for bgp anomaly detection," in *Proc. 4th International Conference on Computing, IoT and Data Analytics*, Madrid, Spain, 2025.

[9] S. Motaali, J. E. López de Vergara, L. de Pedro, and I. González, "Generating balanced and realistic BGP traffic for machine learning-based anomaly detection," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2026.

[10] D. A. Ammara, J. Ding, and K. Tutschku, "Synthetic network traffic data generation: A comparative study," *arXiv preprint arXiv:2410.16326*, February 2025.

[11] E. Amachaghi, S. Abdulkareem, C. Foh, and M. Shojafar, "Improving intrusion detection in O-RAN with synthetic data," *SSRN Electronic Journal*, vol. 20, April 2025.

[12] B. A. Scott, M. N. Johnstone, and P. Szewczyk, "A survey of advanced border gateway protocol attack detection techniques," *MDPI*, October 2024.

[13] J. Wang, "BGPWatch — a comprehensive platform for detecting and diagnosing hijacking incidents," *APNIC Blog*, February 2024. [Online]. Available: https://blog.apnic.net/2024/02/07/bgpwatch-a-comprehensive-platform-for-detecting-and-diagnosing-hijacking-incidents/

[14] M. Lepinski and S. Kent, "RFC 6480: An infrastructure to support secure internet routing," IETF RFC 6480, 2012.

[15] M. Lepinski and K. Sriram, "RFC 8205: BGPsec protocol specification," IETF RFC 8205, 2017.

[16] National Institute of Standards and Technology, "NIST RPKI monitor," https://rpki-monitor.antd.nist.gov/, 2025.

[17] Scapy Project, "BGP Contrib Module – Scapy Documentation," https://scapy.readthedocs.io/en/stable/api/scapy.contrib.bgp.html.

[18] T. R. Henderson, M. Lacage, and G. F. Riley, "ns-3: A discrete-event network simulator for internet systems," in *Proceedings of the Workshop on ns-2*, 2008.

[19] T. Whitaker, "LSTM-GAN for enhanced anomaly detection in time series data," *Journal of Chinese Computer Technology and Science*, 2023.

[20] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series generative adversarial networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[21] Z. Lin, A. Jain, C. Wang, G. Fanti, and V. Sekar, "Using GANs for sharing networked time series data: Challenges, initial promise, and open questions," in *ACM Internet Measurement Conference (IMC)*, 2020.

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, 2002.

[23] F. Benali, D. Bodénès, N. Labroche, and C. de Runz, "Synthetic complex data generation using copula," *CEUR Workshop Proceedings*, 2021.

[24] P. J. Huber and E. M. Ronchetti, *Robust Statistics*. John Wiley & Sons, 2009.

[25] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, 2008.

[26] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM SIGMOD Record*, 2000.

[27] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, 1999.

[28] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 10, no. 1, 2015.

[29] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[30] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2015.

[31] T. C. Tavares, L. C. de Almeida, W. R. D. Silva, M. Chiesa, and F. L. Verdi, "TimeGAN as a simulator for reinforcement learning training in programmable data planes," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2024.

[32] Z. Lin, "DoppelGANger: A new tool for sharing time series data with GANs," *APNIC Blog*, December 2020. [Online]. Available: https://blog.apnic.net/2020/12/18/doppelganger-a-new-tool-for-sharing-time-series-data-with-gans/

[33] L. Breiman, "Random forests," *Machine Learning*, 2001.

[34] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[35] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2016.

[36] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "A detailed analysis of the cicids2017 data set," in *Proceedings of the 2018 International Conference on Information Systems Security and Privacy (ICISSP)*, 2018.

[37] C. Testart, P. Richter, A. King, A. Dainotti, and D. Clark, "BGP Communities: Even more Worms in the Routing Can," in *Proceedings of the Internet Measurement Conference*. ACM, 2019.

[38] P. Sermpezis, P. Kotzias, P. Gigis, X. Dimitropoulos, D. Cicalese, A. King, and A. Dainotti, "ARTEMIS: Neutralizing BGP Hijacking Within a Minute," in *IEEE/ACM Transactions on Networking*. IEEE, 2018.

[39] R. Fontugne, A. Shah, and E. Aben, "BGPStream: A Software Framework for Live and Historical BGP Data Analysis," in *Proceedings of the Internet Measurement Conference*. ACM, 2017.

[40] V. Giotsas, M. Luckie, B. Huffaker, and K. Claffy, "Inferring Complex AS Relationships," in *Proceedings of the Internet Measurement Conference*. ACM, 2015.

[41] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and K. Claffy, "AS Relationships, Customer Cones, and Validation," *Proceedings of the Internet Measurement Conference*, 2013.

[42] J. Li, M. Sung, J. Xu, and L. Li, "Characterizing and Modeling Internet Background Radiation," in *Proceedings of the Internet Measurement Conference*. ACM, 2006.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[44] A. Vaslavsky, "The shape of a bgp update," *RIPE Labs*, 2014, https://labs.ripe.net/author/vastur/the-shape-of-a-bgp-update/.

[45] M. Lad, D. Massey, and L. Zhang, "Analysis of bgp update surge during the Slammer worm attack," in *International Workshop on Distributed Computing*. Springer, 2003.