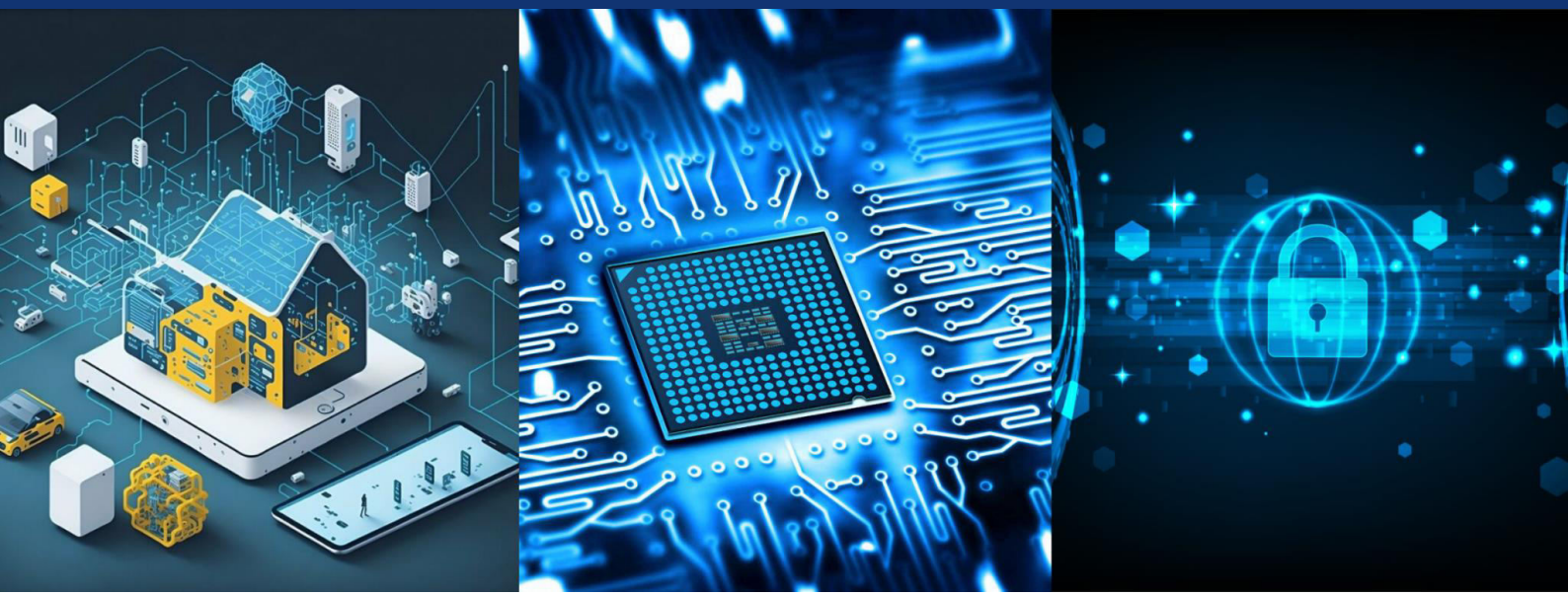
IJIRCCE

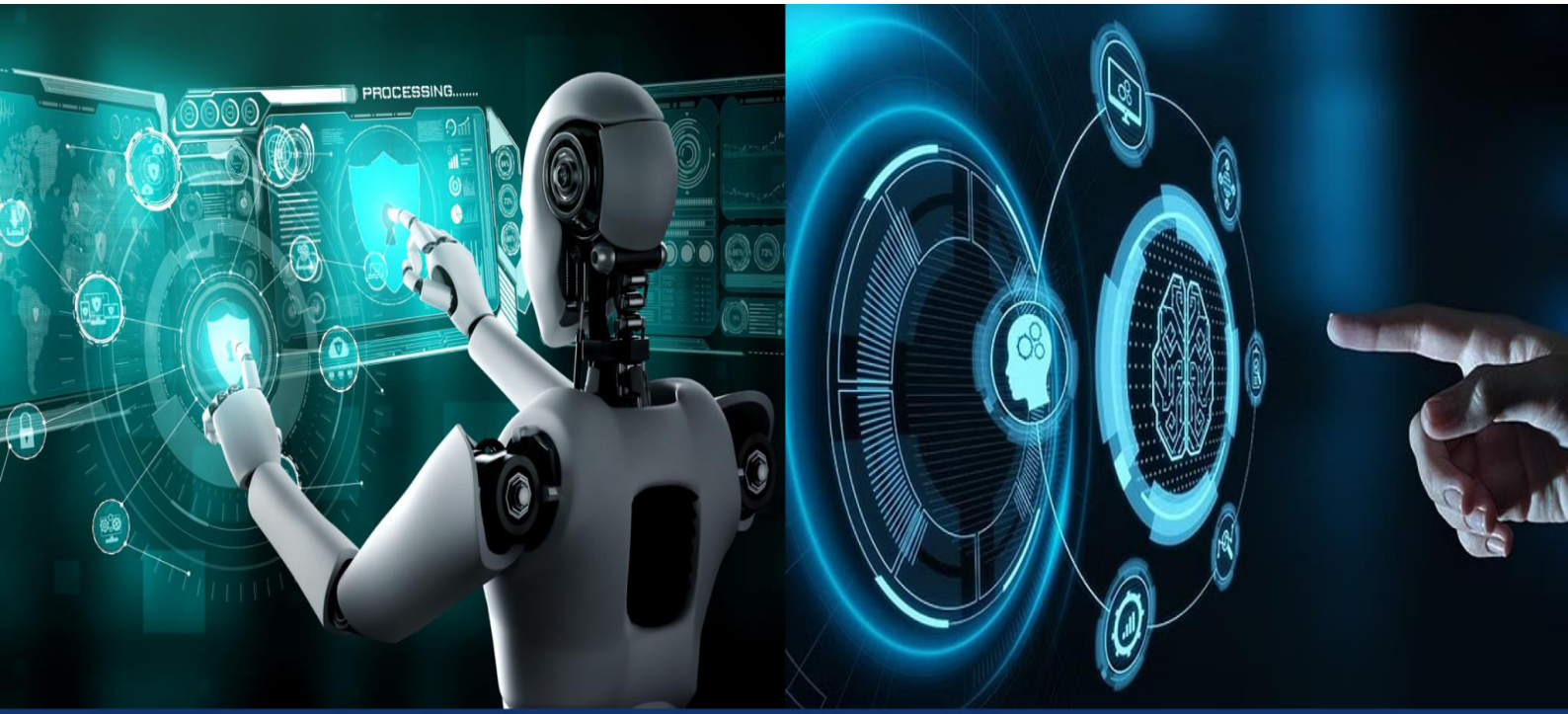
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 14, Issue 4, April 2026

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Detecting and Quantifying Data Drift in Network Intrusion Detection Systems: A Statistical Monitoring System for Drift Detection and Dashboard Visualization

E Bharath, B V Dasaradha Rami Reddy, B Samba, D Sarath

Department of Information Technology, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

ABSTRACT: Machine learning-based Network Intrusion Detection Systems (NIDS) suffer from silent performance degradation when network traffic distributions evolve over time—a phenomenon known as data drift. This paper presents a drift monitoring system that combines XGBoost-based intrusion detection with a statistical drift monitoring system, evaluated on the CICIDS2018 benchmark dataset (16.1 million network flows across 10 days). We propose a three-stage feature selection pipeline that reduces 78 raw CICFlowMeter features to 20 high-information features, followed by a dual drift detector using the Kolmogorov-Smirnov (KS) test and the Population Stability Index (PSI). Experiments on a strictly chronological 60/20/20 split demonstrate that the trained model achieves 95.19% validation accuracy but its attack recall collapses to just 0.66% on the test set—a failure directly attributable to distributional shift. All 20 monitored features exhibit statistically significant KS drift ($p < 0.05$), and the PSI analysis identifies 7 significantly drifted features on the validation set (avg PSI = 0.162) and 1 significantly drifted feature on the test set (avg PSI = 0.101), providing an actionable retraining signal without requiring ground-truth labels at inference time. A FastAPI backend and React dashboard operationalize drift monitoring through a web dashboard; live stream processing for true real-time inference remains as future work.

KEYWORDS: network intrusion detection, data drift, concept drift, XGBoost, Kolmogorov-Smirnov test, Population Stability Index, CICIDS2018, machine learning, anomaly detection.

I. INTRODUCTION

Network Intrusion Detection Systems (NIDS) form a critical layer of defense in modern cybersecurity architectures. Machine learning-based NIDS, trained on labeled network flow data, can achieve high accuracy on held-out test sets drawn from the same distribution as their training data. In production, however, network environments are not static: attack toolkits evolve, normal traffic patterns shift with changing user behavior, and entirely new campaigns emerge with signatures the model has never encountered.

This degradation has two related but distinct causes. Concept drift occurs when the relationship between features and labels changes. Data drift (also called covariate shift) occurs when the marginal distribution of input features changes even if the decision boundary remains nominally valid. Both phenomena cause a model trained on historical data to become unreliable on current traffic without triggering any observable error signal, as predictions continue to be produced but are increasingly wrong.

The core problem motivating this work is that standard deployment pipelines have no mechanism to detect this degradation. A NIDS that achieves 99% accuracy by classifying all traffic as benign is operationally worthless—yet without ground-truth labels at inference time, such silent failure goes undetected for days or weeks.

This paper addresses the problem by augmenting an XGBoost-based NIDS with a statistical drift detection layer. Our contributions are: (1) a chronological evaluation framework on 10 days of CICIDS2018 traffic demonstrating severe recall collapse due to distributional shift; (2) a baseline comparison showing that recall collapse affects all classifiers (LR, RF, XGBoost), proving drift is a dataset-level problem; (3) a dual drift detection methodology combining KS test (detection) with PSI (severity quantification), with an ablation study justifying the combined approach; (4) a three-stage

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

feature selection pipeline reducing 78 raw features to 20; and (5) a deployable drift-monitoring system with FastAPI backend and React dashboard; closed-loop retraining is deferred to future work.

Fig. 1. End-to-End System Architecture

Fig. 1. End-to-end system pipeline (6 phases).

II. RELATED WORK

A. Machine Learning for NIDS

The application of machine learning to NIDS has a long history, evolving from decision trees and naive Bayes [5] through early benchmark datasets [9] to the modern CICIDS series [17], which were generated under realistic conditions using updated attack toolkits. Gradient boosting methods, in particular XGBoost [18], have emerged as strong performers on tabular flow-based NIDS data [10], owing to native class-imbalance handling, fast training, and built-in feature importance.

B. Data Drift and Concept Drift

Data drift and concept drift in deployed ML systems have been studied extensively in recent surveys [1, 4, 11]. For network security, Pendlebury et al. [2] demonstrated severe temporal decay in malware classifiers, showing accuracy losses of 20–40% within 12 months. Apruzzese et al. [6] provide a systematic empirical assessment of ML-based NIDS in operational threat scenarios, identifying persistent gaps between benchmark performance and real-world effectiveness. Andresini et al. [15] directly address concept-drift robustness in NIDS through INSOMNIA, proposing proactive retraining strategies triggered by distribution shift across evolving threat campaigns. Yang et al. [16] present CADE, a framework for detecting and explaining concept drift samples in security applications, demonstrating the importance of interpretable drift evidence for analyst workflows.

Our work provides a quantitative drift monitoring framework that operationalizes this body of critique. Unlike INSOMNIA [15] and CADE [16], which require ground-truth labels to trigger retraining or explain drift, our KS+PSI framework operates entirely without labels at inference time—a prerequisite for real NIDS deployment where attack labels are unavailable. Unlike error-stream detectors (CUSUM, ADWIN) [3, 8], our approach monitors raw feature distributions, enabling drift detection before any prediction feedback is available.

C. Statistical Drift Detection Methods

Rabanser et al. [12] provide a comprehensive empirical study of shift detection methods, establishing the KS test as a strong univariate detector that is sensitive to any distributional difference but prone to false positives at large sample sizes. The Population Stability Index (PSI), implemented in production ML monitoring toolkits including Alibi Detect [13], provides complementary magnitude quantification with established thresholds (< 0.1 Stable, 0.1–0.2 Moderate, $\geq$ 0.2 Significant) increasingly adopted in feature-level monitoring [7]. Note that these thresholds were originally developed for financial model monitoring and have not been formally validated for network security distributions; domain-specific recalibration is recommended prior to production deployment.

Hinder et al. [3] and Wang et al. [8] survey concept drift detectors including CUSUM, ADWIN, DDM, and EDDM alongside their statistical assumptions and computational trade-offs. These error-stream detectors require labeled data at inference time, which is unavailable in our deployment scenario, motivating our feature-distribution approach. A natural question is whether a multivariate drift detector—such as Maximum Mean Discrepancy (MMD) or the Least-Squares Density Difference (LSDD) method—would better capture feature interactions than the univariate KS test applied per feature. We adopt the univariate approach for three reasons: (1) high-dimensional NIDS feature sets impose prohibitive

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

quadratic cost for kernel-based multivariate methods at monitoring time; (2) the per-feature PSI breakdown provides directly interpretable, actionable signals that multivariate tests cannot provide; and (3) the PSI severity scores already capture the practical impact of feature-level shift. The trade-off is that feature interactions are not captured; we acknowledge this as a limitation and note that multivariate detectors remain a valuable direction for offline drift root-cause analysis.

III. DATASET AND EXPERIMENTAL SETUP

A. CICIDS2018 Dataset

We use the CICIDS2018 dataset [17], consisting of 10 daily network capture CSV files (February 14 to March 2, 2018) totaling approximately 16.1 million flow records. Each flow is described by 78 CICFlowMeter [14] features. Labels include Benign and multiple named attack types: Brute Force (FTP/SSH), DoS (GoldenEye, Hulk, Slowloris, SlowHTTPTest), DDoS (LOIC-HTTP, LOIC-UDP), Infiltration, Bot, and Web attacks (XSS, SQL Injection). All attack labels are binarized to Attack (1); Benign remains 0.

B. Chronological Split

Since the project's central hypothesis is that data drift occurs over time, we use a strictly chronological split rather than random sampling. Days 1–6 (Feb 14–22) form the training set; Days 7–8 (Feb 23, Feb 28) the validation set; Days 9–10 (Mar 1–2) the test set. Note that Days 7 and 8 correspond to February 23 and February 28 respectively; the intermediate dates (February 24–27) are absent from the CICIDS2018 dataset as no network captures were conducted on those days. The test set crosses a month boundary, providing a strong temporal challenge. Attack prevalence varies substantially across splits (training: 17.5%, validation: 28.0%, test: 27.6%), reflecting different attack campaigns—a form of label drift that compounds the covariate shift measured by our framework.

TABLE I. DATASET SPLIT STATISTICS

Split	Days	Date Range	Total Rows
Train	1–6	Feb 14 – Feb 22	13,114,708
Validation	7–8	Feb 23, Feb 28	1,649,769
Test	9–10	Mar 1 – Mar 2	1,372,706
Total	**10**	**Feb 14 – Mar 2**	**16,137,183**

IV. METHODOLOGY

A. Preprocessing Pipeline

Each raw CICIDS2018 file undergoes eight preprocessing steps: (1) column name whitespace stripping; (2) removal of embedded duplicate header rows (present in Days 3, 8, and 9); (3) identifier column removal (Flow ID, IP addresses, timestamps); (4) numeric coercion using pd.to_numeric with error coercion; (5) replacement of infinite values with NaN; (6) row-level NaN dropping (< 1.1% of rows affected); (7) binary label encoding; and (8) index reset. Files exceeding 500 MB are processed in 200,000-row chunks to respect the 8 GB memory constraint on the development machine. After preprocessing, each split contains 79 columns (78 numeric features + Label).

B. Three-Stage Feature Selection

Feature selection is computed on a 300,000-row stratified training sample to eliminate redundancy and improve both training efficiency and drift monitoring interpretability. Stage 1 (zero-variance filter) removes 10 features that are universally zero in CICIDS2018, including all six bulk-transfer statistics and two TCP flag fields. Stage 2 (high-correlation pruning, $|r| > 0.95$) removes a further 23 features, including exact duplicates such as Subflow Fwd Pkts ($r = 1.000$ with Tot Fwd Pkts) and Pkt Size Avg ($r = 1.000$ with Pkt Len Mean). Stage 3 (mutual information ranking) selects the top 20 features by MI score from the remaining 45 post-correlation features. The cutoff of 20 was chosen at the MI score elbow: the 21st-ranked feature (Bwd Pkts/s, MI ≈ 0.089) yields an 11% reduction in MI contribution relative to the 20th feature (Bwd IAT Mean, MI ≈ 0.100), and features ranked 21–45 contribute cumulatively less than 18% of the total MI mass, indicating diminishing discriminative returns beyond rank 20. The top two features by MI are Init Fwd Win

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Byts (MI = 0.302) and Dst Port (MI = 0.250), reflecting the strong fingerprinting power of TCP window size and destination port for attack identification.

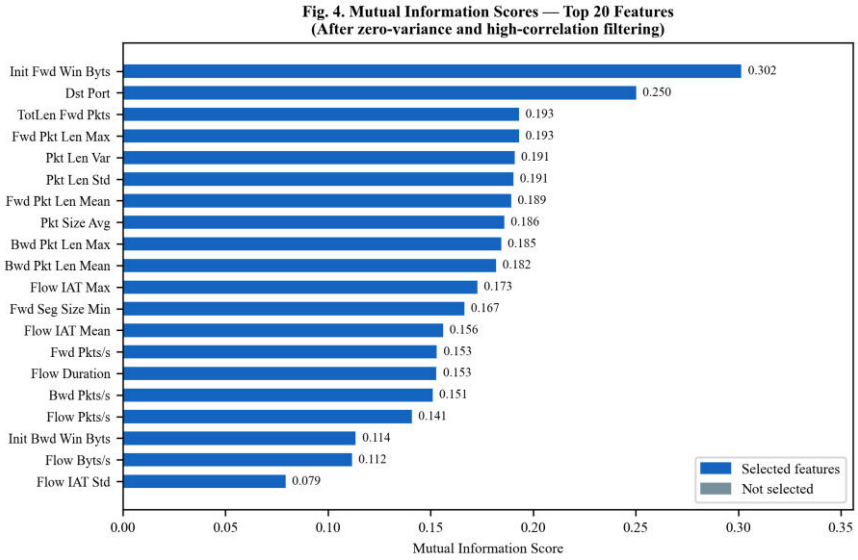


Fig. 2. Mutual information scores (top 20 features after filtering).

C. XGBoost Model Training

We train an XGBoost binary classifier [18] on 12,066,133 labeled training flows from the six-day training split, preserving the class distribution via scale_pos_weight. Key hyperparameters are: n_estimators=300, max_depth=6, learning_rate=0.05, and scale_pos_weight=6.49 (ratio of benign to attack samples) to compensate for class imbalance. The histogram-based tree method (tree_method='hist') is used for efficiency on this large dataset. Model evaluation on validation and test sets uses 200,000-row chunks to avoid loading the full evaluation files into memory, with predictions accumulated and metrics computed on the complete aggregated prediction set.

D. Drift Detection: KS Test and PSI

The drift detection module compares feature distributions of any uploaded dataset against a 100,000-row random sample of the training reference (seeded RS=42 for reproducibility). For each feature, the two-sample Kolmogorov-Smirnov test is computed using scipy.stats.ks_2samp; features are flagged as drifted if $p < 0.05$. The PSI is computed using percentile-based binning (10 bins defined by training data percentiles, epsilon = 10^{-4} to avoid log(0)):

$$PSI = \Sigma (P_{actual} - P_{expected}) \times \ln(P_{actual} / P_{expected})$$

PSI thresholds follow established practice: < 0.1 Stable, 0.1–0.2 Moderate Drift, ≥ 0.2 Significant Drift. These thresholds originate from financial model monitoring [13] and should be treated as indicative guidance for network security contexts pending domain-specific recalibration. The combined report integrates both measures per feature and issues a retraining recommendation when average PSI exceeds 0.1 (moderate) or 0.2 (significant). The critical advantage of this approach is that it operates entirely without ground-truth labels, making it deployable in operational NIDS environments where attack labels are unavailable at inference time.

V. EXPERIMENTAL RESULTS

A. Classification Performance

Table II presents classification metrics on the validation and test sets (see also Fig. 3 for confusion matrices). The model achieves 95.19% accuracy on validation but only 70.05% on test. More critically, attack recall collapses from 91.12% on validation to 0.66% on test—functionally zero for a security system. The dramatic decline in recall between validation

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

and test is the primary evidence of data drift. The ROC-AUC also degrades from 0.940 to 0.423, falling below random chance on the test set, indicating that the model's probability estimates are no longer calibrated for the test distribution.

TABLE II. CLASSIFICATION PERFORMANCE

Metric	Validation (Days 7–8)	Test (Days 9–10)
Accuracy	95.19%	70.05%
Precision	91.68%	6.69%
Recall	**91.12%**	**0.66%**
F1-Score	91.40%	1.21%
ROC-AUC	94.04%	42.28%

Fig. 2. Confusion Matrices — Validation and Test Sets

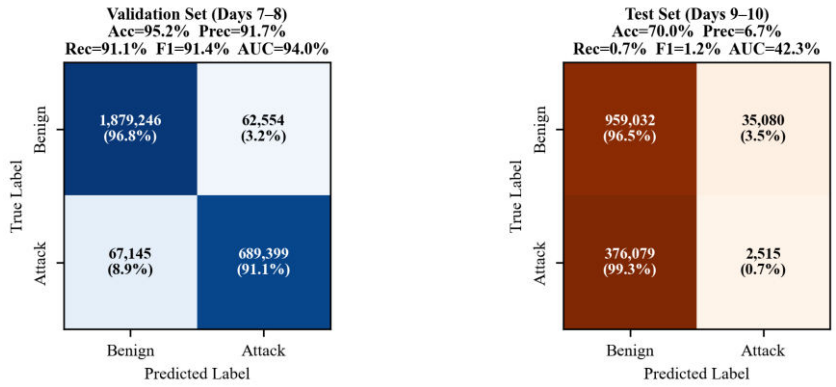


Fig. 3. Confusion matrices for validation (left) and test (right) sets.

The accuracy-recall paradox illustrated in Table II and Fig. 3 is important to emphasize: 95.19% validation accuracy appears strong, but it is dominated by the majority benign class. A model that classifies all traffic as benign would achieve a similar accuracy figure while missing all attacks. The collapse from 91.12% to 0.66% recall between validation and test demonstrates that the model has learned attack signatures specific to the February 14–22 training period and fails completely on the March attack campaigns.

B. Drift Detection Results

Table III and Fig. 4 present the drift summary for both evaluation sets. All 20 monitored features exhibit statistically significant KS drift in both sets. On the validation set, 7 features (35%) exceed the PSI significant drift threshold (≥ 0.2), with an average PSI of 0.162 indicating moderate-to-significant overall shift. On the test set, 1 feature (5%) shows significant drift and the average PSI is 0.101, yet the recall collapse is more severe than on the validation set. This apparent paradox is resolved by distinguishing aggregate PSI from per-feature PSI for the highest-information features: Dst Port (PSI = 0.272, Significant) is the single most discriminative feature for attack classification, and its severe drift alone is sufficient to destroy model utility even when the aggregate average remains moderate.

TABLE III. DRIFT DETECTION SUMMARY

Metric	Validation	Test
Overall Drift	**YES**	**YES**
KS Drifted Features	20 / 20	20 / 20
PSI Stable (< 0.1)	5	12

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

PSI Moderate (0.1–0.2)	8	7
PSI Significant (≥ 0.2)	7	1
Average PSI	0.1622	0.1006

TABLE IV. TOP DRIFTED FEATURES — VALIDATION SET

Feature	KS Stat	PSI	Status
Fwd Seg Size Min	0.1129	0.3534	Significant
Pkt Len Max	0.1131	0.3177	Significant
Flow Duration	0.1896	0.2861	Significant
Flow Pkts/s	0.1655	0.2188	Significant
Fwd Pkts/s	0.1631	0.2170	Significant
Dst Port	0.1705	0.2162	Significant
Flow IAT Mean	0.1639	0.2057	Significant

TABLE IVB. TOP DRIFTED FEATURES — TEST SET

Feature	KS Stat	PSI	Status
Dst Port	0.1128	0.2721	Significant
Fwd Seg Size Min	0.0971	0.1609	Moderate
Bwd Pkts/s	0.1724	0.1599	Moderate
Flow Duration	0.1009	0.1486	Moderate
Bwd Pkt Len Max	0.1035	0.1414	Moderate
Flow IAT Mean	0.1025	0.1339	Moderate
Flow Pkts/s	0.1040	0.1264	Moderate

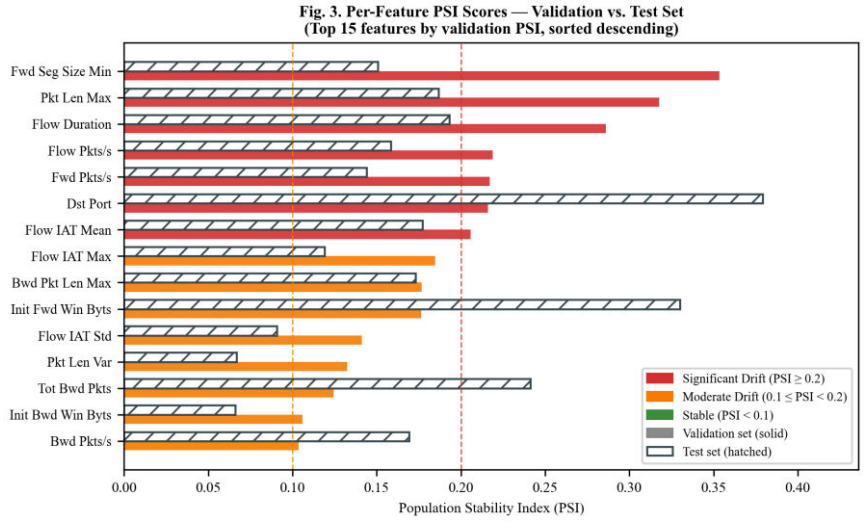


Fig. 4. Per-feature PSI scores: validation vs. test set.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

C. Baseline Model Comparison

To validate the choice of XGBoost and to demonstrate that recall collapse is a dataset-level phenomenon (not an artifact of a single classifier), we evaluate two standard baselines on the same training sample: Logistic Regression (LR, class_weight='balanced', lbfgs solver, C=1.0) and Random Forest (RF, 100 trees, max_depth=15, class_weight='balanced_subsample'). Table V presents the full comparison.

TABLE V. BASELINE MODEL COMPARISON

Model	Val Acc	Val Rec	Val F1	Test Acc	Test Rec	Test F1
Log. Regression	89.08%	69.25%	78.05%	70.43%	2.15%	3.85%
Random Forest	72.01%	0.23%	0.46%	72.41%	0.03%	0.06%
XGBoost (Ours)	**95.19%**	**91.12%**	**91.40%**	**70.05%**	**0.66%**	**1.21%**

XGBoost substantially outperforms both baselines on validation: it achieves 91.12% recall vs. 69.25% for Logistic Regression and 0.23% for Random Forest. Note that the RF hyperparameters (max_depth=15, class_weight='balanced_subsample') were not cross-validated and may not represent the best achievable RF performance; the comparison should be interpreted as illustrative of classifier behavior under drift rather than as a definitive RF benchmark. The near-zero RF validation recall is explained by the interaction between max_depth=15 and class_weight='balanced_subsample': deep trees with per-tree balanced sampling overfit attack-class boundaries to the specific training distribution, producing high false-negative rates even on the moderately shifted validation set. On the test set, all three models collapse to near-zero recall (XGBoost 0.66%, LR 2.15%, RF 0.03%). This universal collapse across architecturally different classifiers is the strongest possible evidence that the performance degradation is caused by data drift, not a model-specific weakness.

D. Connecting Drift to Performance Degradation

The drift results directly explain the recall collapse. The training data (Days 1–6) contains FTP/SSH brute-force and DoS attacks from February 2018. The test set (Days 9–10) introduces March campaigns targeting different services with different tools, creating a completely different attack signature distribution. Dst Port is the most severely drifted feature on the test set (PSI = 0.272, Significant Drift), consistent with March attacks targeting different services than February. Seven features show Moderate drift (PSI 0.10–0.20): Fwd Seg Size Min (0.161), Bwd Pkts/s (0.160), Flow Duration (0.149), Bwd Pkt Len Max (0.141), Flow IAT Mean (0.134), Flow Pkts/s (0.126), and Fwd Pkts/s (0.101). Notably, Init Fwd Win Byts—the highest-MI feature (MI = 0.302)—shows Stable drift (PSI = 0.093), indicating the TCP window fingerprint of March attack tools is similar to training. The PSI monitoring system correctly triggers the 'URGENT: Model retraining strongly recommended' alarm on both evaluation sets without requiring any ground-truth labels.

VI. DISCUSSION

A. The Accuracy-Recall Paradox

The most striking result is the coexistence of high accuracy (95.19%) and near-zero recall (0.66% on test) simultaneously. This is not an evaluation artifact—it is a direct consequence of class imbalance combined with distributional shift. When the majority class (benign) dominates and the model has not learned the new attack signatures, accuracy remains high while recall collapses. This reinforces that recall and F1-score must be the primary evaluation criteria for imbalanced security tasks, and that accuracy alone is dangerously misleading in NIDS contexts.

B. PSI vs. KS Test: Complementary Signals

The choice to use both KS and PSI rather than either alone is deliberate. With 100,000 reference samples and millions of evaluation samples, the KS test has enormous statistical power: any distributional difference, however tiny, is detected. The KS result (20/20 features drifted) confirms drift exists but gives no practical guidance on severity. The PSI distinguishes stable from significantly drifted features, providing the actionable signal. A KS-only system would generate constant alarms; a PSI-only system might miss statistically significant but small shifts. The combination provides both detection sensitivity and severity calibration.

© 2026 IJIRCCE | Volume 14, Issue 4, April 2026 DOI: 10.15680/IJIRCCE.2026.1404018

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

C. Ablation Study: KS-Only vs. PSI-Only vs. Combined

To justify the dual-method approach, Table VI compares the three detection strategies on both evaluation sets. KS-only flags all 20 features as drifted in both sets (100% detection rate), but with 100,000 reference samples vs. over a million evaluation samples, the KS test has sufficient power to detect sub-1% distributional differences as statistically significant. This results in a permanent alarm state that provides no severity guidance and no feature prioritization. PSI-only monitoring correctly identifies severity levels and prioritizes the 7 most drifted features on the validation set (PSI ≥ 0.2), providing actionable guidance. However, PSI operates on bin proportions and lacks a formal statistical test. The combined approach uses KS as a computationally cheap first-pass detector and PSI as a severity quantifier, delivering statistical confirmation and practical guidance simultaneously.

TABLE VI. ABLATION STUDY: DRIFT DETECTION STRATEGY COMPARISON

Property	KS-Only	PSI-Only	KS + PSI (Ours)
Drift Detected (Val)	YES (20/20 features)	YES (avg PSI=0.162)	YES
Drift Detected (Test)	YES (20/20 features)	YES (avg PSI=0.101)	YES
Severity Level	None	Yes (3-tier)	Yes (3-tier)
Feature Prioritization	None	Yes (by PSI)	Yes (by PSI)

D. Limitations

This study has several limitations. First, all experiments are conducted on a single controlled laboratory dataset (CICIDS2018); real-world deployments encounter different traffic mixtures. Second, we perform binary classification, collapsing all attack types into one class; per-class PSI monitoring in a multi-class model would provide far more actionable retraining signals, enabling targeted updates to specific attack detectors rather than requiring full model retraining. Third, we demonstrate that drift causes degradation but do not evaluate retraining strategies to recover performance. Fourth, PSI thresholds borrowed from financial modeling require recalibration for network security contexts before production use. Fifth, the near-zero Random Forest validation recall (0.23%) reflects sensitivity to the interaction between tree depth and balanced_subsample weighting; hyperparameter re-tuning via cross-validation may yield materially different baseline comparisons. Sixth, all experiments are offline batch evaluations; the FastAPI dashboard has not been benchmarked under live traffic load, and latency and throughput measurements are required to substantiate any real-time deployment claims.

VII. CONCLUSION

This paper has demonstrated that data drift is a primary driver of performance degradation in ML-based NIDS, and that statistical monitoring can reliably detect this degradation without requiring ground-truth labels at inference time.

Using the CICIDS2018 dataset across a 10-day chronological evaluation, we showed that an XGBoost classifier achieving 95.19% validation accuracy experiences near-total recall collapse (0.66%) on the test set. A baseline comparison with Logistic Regression and Random Forest confirmed this collapse is universal across classifier types (recall: LR 2.15%, RF 0.03% on test), proving the degradation is caused by data drift—not a model-specific weakness. The PSI framework correctly triggers retraining recommendations from feature distributions alone, enabling proactive maintenance without ground-truth labels. An ablation study confirmed that the combined KS + PSI approach provides strictly more information than either method alone.

Future work will focus on: (1) evaluating online retraining strategies (sliding window, reservoir sampling) to recover recall in drifting environments; (2) extending to multi-class classification with per-class drift monitoring; (3) investigating active learning for efficient labeling of drifted traffic samples; and (4) validating the framework on real-world enterprise network captures.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

REFERENCES

[1] H. Jmila and M. I. Khedher, "Evolving Cybersecurity Frontiers: A Comprehensive Survey on Concept Drift and Feature Dynamics Aware ML and DL in Intrusion Detection Systems," Engineering Applications of Artificial Intelligence, vol. 137, article 109143, 2024.
[2] Z. Kan, S. McFadden, D. Arp, F. Pendlebury, R. Jordaney, J. Kinder, F. Pierazzi, and L. Cavallaro, "TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time (Extended Version)," arXiv preprint arXiv:2402.01359, 2024.
[3] F. Hinder, V. Vaquet, and B. Hammer, "One or Two Things We Know about Concept Drift—A Survey on Monitoring in Evolving Environments. Part A: Detecting Concept Drift," Frontiers in Artificial Intelligence, vol. 7, article 1330257, 2024.
[4] F. Hinder, V. Vaquet, and B. Hammer, "One or Two Things We Know about Concept Drift—A Survey on Monitoring in Evolving Environments. Part B: Locating and Explaining Concept Drift," Frontiers in Artificial Intelligence, vol. 7, article 1330258, 2024.
[5] A. Thakkar and R. Lohiya, "A Review of the Advancement in Intrusion Detection Dataset," Computers & Security, vol. 132, article 103382, Sep. 2023.
[6] G. Apruzzese, P. Laskov, and J. Schneider, "SoK: Pragmatic Assessment of Machine Learning for Network Intrusion Detection," in Proc. IEEE European Symposium on Security and Privacy (EuroS&P), Delft, 2023.
[7] A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in Deploying Machine Learning: A Survey of Case Studies," ACM Computing Surveys, vol. 55, no. 6, article 114, 2023.
[8] Y. Wang, X. Ma, and Q. He, "Fast Concept Drift Detection Using Raw Data in the Presence of Missing Values," Knowledge-Based Systems, vol. 277, article 110812, 2023.
[9] R. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic," Applied Sciences, vol. 12, no. 2, article 1181, 2022.
[10] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a Standard Feature Set for Network Intrusion Detection System Datasets," Mobile Networks and Applications, vol. 27, pp. 357–370, 2022.
[11] F. Bayram, B. S. Ahmed, and A. Kassler, "From Concept Drift to Model Degradation: An Overview on Performance-Aware Drift Detectors," Knowledge-Based Systems, vol. 245, article 108632, Jun. 2022.
[12] S. Rabanser, S. Günnemann, and Z. C. Lipton, "Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift," Journal of Machine Learning Research, vol. 23, no. 322, pp. 1–57, 2022.
[13] J. Klaise, A. Van Looveren, G. Vacanti, and A. Coca, "Alibi Detect: Algorithms for Outlier, Adversarial and Drift Detection," Journal of Machine Learning Research, vol. 23, no. 147, pp. 1–6, 2022.
[14] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems," in Big Data Technologies and Applications, Lecture Notes ICST, vol. 371, Springer, 2022.
[15] A. Andresini, F. Pendlebury, F. Pierazzi, C. Loglisci, A. Appice, and L. Cavallaro, "INSOMNIA: Towards Concept-Drift Robustness in Network Intrusion Detection," in Proc. 14th ACM Workshop Artif. Intell. Security (AISec '21), 2021.
[16] Q. Yang, J. Xu, and C. Hao, "CADE: Detecting and Explaining Concept Drift Samples for Security Applications," in Proc. 30th USENIX Security Symp., 2021.
[17] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Inf. Syst. Security Privacy (ICISSP), 2018.
[18] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details