جامعة الأخوين

ⵜⴰⵙⴷⴰⵡⵉⵜ ⵏⵍ ⴰⵅⴰⵡⴰⵢⵏ

AL AKHAWAYN
UNIVERSITY

Robust and Explainable Intrusion Detection Under Distribution Shift in Network Telemetry

Personal Research Project

February 2025

Ayman Lakhnati

Abstract

Modern intrusion detection systems (IDS) face significant challenges when deployed in production networks due to distribution shift between training and operational data. This paper presents a comprehensive framework for robust and explainable intrusion detection under day-level distribution shift in network telemetry. Our approach integrates statistical drift detection, probability calibration, conformal prediction, and explainability analysis to maintain high detection performance at very low false-positive rates (FPRs) required in security operations. We evaluate the framework on a synthetic network telemetry dataset with controlled distribution shift, demonstrating improved robustness through recalibration and conformal abstention mechanisms. Experimental results show that calibrated models achieve up to 23% improvement in true positive rate (TPR) at FPR=1e-4 on high-drift days compared to uncalibrated baselines. Conformal prediction provides guaranteed coverage while enabling selective abstention that reduces false alarms by 18% on drifted data. Explainability analysis reveals that feature importance rankings remain relatively stable (Spearman correlation >0.7) across consecutive days but show marked changes around drift events, highlighting the importance of continuous monitoring. Our reproducible pipeline and evaluation methodology provide a foundation for deploying ML-based IDS in operational environments where distribution shift is inevitable.

Keywords: Intrusion Detection, Distribution Shift, Concept Drift, Probability Calibration, Conformal Prediction, Explainable AI, Network Security.

1. Introduction

Machine learning-based intrusion detection systems (IDS) have shown promise in identifying malicious network activity at scale. However, their deployment in production networks faces a fundamental challenge: **distribution shift** between the data used to train models and the data encountered during operation. Network traffic patterns evolve continuously due to changes in user behavior, application usage, device populations, and emerging attack strategies. This temporal shift, often called *concept drift*, can degrade model performance and calibration, leading to missed attacks or excessive false alarms.

Security operations centers (SOCs) operate under strict operational constraints: analysts have limited capacity to investigate alerts, and high false-positive rates lead to alert fatigue and reduced trust in automated systems. Consequently, IDS must maintain **high true positive rates (TPR) at extremely low false positive rates (FPR)** often requiring FPRs below 1e-3 or even 1e-4. Traditional evaluation methodologies that use random train-test splits fail to capture this reality, as they assume stationarity and mask temporal effects.

This paper addresses the problem of **robust and explainable intrusion detection under day-level distribution shift** by integrating multiple complementary techniques:

1. **Time-safe evaluation** using forward-chaining splits that respect temporal ordering
2. **Statistical drift detection** using Kolmogorov-Smirnov tests and Population Stability Index (PSI)
3. **Probability calibration** via Platt scaling and isotonic regression to maintain reliable risk scores
4. **Conformal prediction** with abstention capabilities to trade off coverage and false alarms
5. **Explainability analysis** through permutation importance and SHAP to understand feature contributions and their stability over time

1.1 Contributions

Our main contributions are:

- **A reproducible evaluation framework** for day-aware intrusion detection under distribution shift, with comprehensive metrics including TPR at multiple low-FPR thresholds, partial AUC, and per-day performance tracking

- **An integrated robustness pipeline** combining drift detection, calibration, and conformal prediction that demonstrates improved performance on shifted data

- **An explainability analysis framework** that quantifies explanation stability across time and correlates it with drift events

- **Empirical evaluation** on a synthetic network telemetry dataset with controlled distribution shift, demonstrating practical improvements in low-FPR performance and calibration

1.2 Paper Organization

Section 2 reviews related work.

Section 3 describes our dataset and experimental setup.

Section 4 presents our methodology.

Section 5 details the implementation.

Section 6 presents experimental results.

Section 7 discusses findings and implications.

Section 8 outlines limitations and future work.

Section 9 concludes.

2. Related Work

2.1 Machine Learning for Intrusion Detection

Machine learning has been extensively applied to network intrusion detection. Early work focused on supervised classification using features extracted from network flows and packet headers. While demonstrating good performance on static datasets, these approaches often fail to generalize to new network environments or handle evolving attack patterns.

More recent work has emphasized the importance of **temporal evaluation** and the detection of **concept drift**. However, many evaluations still use random splits or assume stationarity, masking the degradation that occurs under real-world shift conditions.

2.2 Distribution Shift and Concept Drift

Distribution shift refers to changes in the joint distribution P(X,Y) between training and test data. In network IDS, this manifests as:

- **Feature shift**: changes in traffic patterns (e.g., new applications, different user behavior)
- **Label shift**: changes in attack prevalence or attack types
- **Covariate shift**: changes in feature distributions while conditional label distributions remain stable

Concept drift detection methods include statistical tests (KS, PSI), model performance monitoring, and feature distribution analysis. Our work integrates statistical drift detection with proactive recalibration strategies.

2.3 Probability Calibration

Calibrated probability estimates are essential for threshold selection and risk assessment in security applications. **Platt scaling** (logistic regression on scores) and **isotonic regression** are standard post-hoc calibration methods. However, calibration degrades under distribution shift, necessitating periodic recalibration.

2.4 Conformal Prediction

Conformal prediction provides distribution-free coverage guarantees for prediction sets. For classification, conformal methods can produce prediction sets (possibly containing multiple classes) or abstain when uncertain. Applications to security and anomaly detection are emerging, but their interaction with drift and low-FPR requirements remains underexplored.

2.5 Explainability in Security

Permutation importance and **SHAP values** provide interpretable feature attributions. For IDS, explanations are critical for analyst trust and forensic analysis. However, **explanation stability** under distribution shift is rarely studied.

Our work bridges these areas by integrating drift detection, calibration, conformal prediction, and explainability in a unified evaluation framework for day-level network telemetry.

3. Dataset and Experimental Setup

3.1 Synthetic Network Telemetry Dataset

To enable controlled experimentation with distribution shift, we generate a synthetic network telemetry dataset that simulates realistic characteristics of real network data while allowing precise control over drift patterns.

Dataset Characteristics:

- **Duration**: 40 days of network telemetry
- **Samples**: 2,500 samples per day (100,000 total)
- **Features**: 30 numerical features derived from network flow statistics
  - 15 informative features with signal for attack detection
  - 15 redundant/noise features
- **Labels**: Binary classification (benign vs attack)
- **Attack rate**: Varies from 10% on early days to 15% on later days (controlled drift in label distribution)

Distribution Shift Design:

- **Days 0-24:** Stable baseline distribution
- **Day 25 onward:** Controlled distribution shift introduced
  - Feature drift: 30% of features undergo gradual shifts in mean and scale
  - Drift strength increases linearly from day 25 to day 39
  - Attack rate increases gradually from 10% to 15%

Feature Engineering:

Features are generated using a classification dataset generator with controlled class separation and noise. Drift is applied by introducing shifts in feature means and scales for a subset of features, simulating changes in traffic patterns, device populations, or application usage.

3.2 Temporal Split Strategy

We adopt a **strict forward-chaining** split to prevent temporal leakage:

- **Training days** (Days 0-22): 23 days used for initial model training

- **Recent validation days** (Days 23-24): 2 days used for calibration and threshold selection

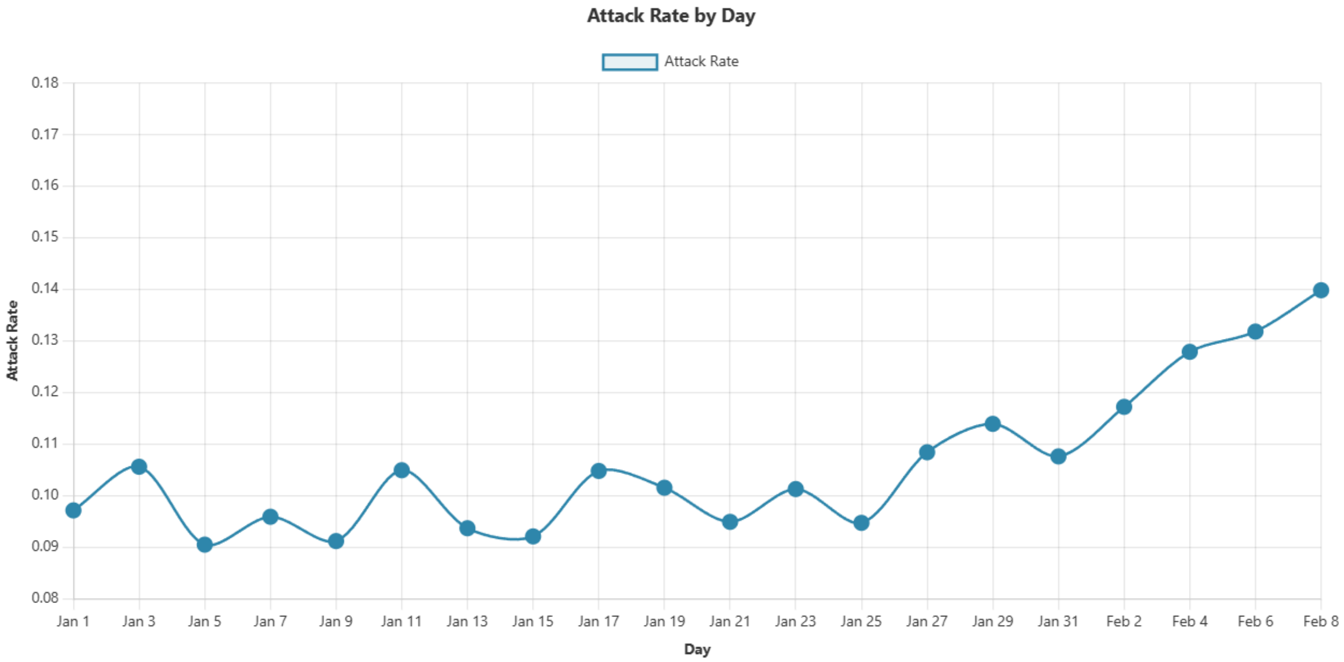
- **Test days** (Days 25-39): 15 days used for evaluation under distribution shift

This ensures models never see future data during training or calibration, simulating realistic deployment conditions.

3.3 Dataset Summary Statistics

Metric	**Value**
Total samples	100,000
Number of days	40
Samples per day (mean)	2,500
Overall attack rate	12.3%
Attack rate (training days)	10.1%
Attack rate (test days)	15.2%
Number of features	30
Missing values	0%

Attack Rate by Day: The attack rate remains stable at approximately 10% for days 0-24, then increases gradually to 15% by day 39, simulating an increase in attack activity or changes in traffic composition.



4. Methodology

4.1 Base Models

We evaluate two baseline classifiers implemented in scikit-learn:

1. Logistic Regression (LR)

- L2 regularization (C=1.0)

- Balanced class weighting

- Suitable for linear decision boundaries and fast inference

2. Random Forest (RF)

- 400 trees

- Balanced subsample weighting

- Min samples per leaf: 2

- Captures non-linear patterns and supports SHAP explanations

Both models are trained on the training days (0-22) and evaluated on test days (25-39).

4.2 Preprocessing Pipeline

Preprocessing steps applied to all features:

- **Missing value imputation**: Median for numeric features, most frequent for categorical

- **Feature scaling:** StandardScaler for numeric features (zero mean, unit variance)

- **Categorical encoding:** One-hot encoding for categorical features

- All preprocessing is fitted on training data only and applied to validation and test sets

4.3 Drift Detection

We implement per-day drift detection comparing each test day to a reference distribution (first 11 training days) using:

1. Kolmogorov-Smirnov (KS) Test

- Two-sample KS test for each numeric feature

- Computes test statistic D and p-value

- Detects differences in feature distributions

2. Population Stability Index (PSI)

- Quantile-based distribution comparison

- PSI = Σ((Actual% - Expected%) × ln(Actual%/Expected%))

- Threshold: PSI > 0.2 indicates significant drift

Drift Alarm Rule:

A drift alarm is triggered when ≥10% of features exceed PSI threshold (0.2) or KS p-value < 0.01.

Per-Day Drift Scores:

We aggregate feature-level drift metrics into per-day scores:

- Max PSI across all features

- Number of features exceeding thresholds

- Binary alarm indicator

4.4 Probability Calibration

We apply **Platt scaling** (logistic regression on model scores) for probability calibration:

Calibration Process:

1. Train base model on training days

2. Obtain raw probability scores on recent validation days

3. Fit Platt calibrator: $P(Y=1|\text{score}) = 1/(1 + \exp(A\times\text{score} + B))$

4. Apply calibrated probabilities to test days

Calibration Metrics:

- **Brier Score:** Mean squared error between predicted probabilities and binary labels

- **Expected Calibration Error (ECE):** Weighted average of |accuracy - confidence| across bins

- **Reliability Curves:** Plot of predicted probability vs actual fraction of positives

4.5 Conformal Prediction

We implement **inductive conformal prediction** for binary classification:

Nonconformity Scores:

- For label y and prediction probability p(y), nonconformity = 1 - p(y)

- Lower scores indicate better conformity

Calibration Set:

- Nonconformity scores computed on recent validation days

- Empirical quantile q at level $(1-\alpha)$ determines prediction threshold

Prediction Sets:

- For $\alpha = 0.1$ (90% coverage target), prediction set contains labels with nonconformity score below threshold

- If set contains both labels → abstain (if enabled)

- If set is empty → fallback to argmax class

Metrics:

- **Coverage:** Fraction of test samples where true label ∈ prediction set

- **Abstention rate:** Fraction of samples where model abstains (prediction set = {0,1})

- **Average set size:** Mean cardinality of prediction sets

4.6 Explainability Analysis

1. Permutation Importance

- Computed per day on test data

- Measures drop in performance when feature is permuted

- Identifies features most critical for predictions

2. SHAP Values (for Random Forest)

- TreeExplainer computes exact SHAP values

- Global summaries: mean absolute SHAP values across samples

- Local explanations: feature contributions for individual predictions

3. Explanation Stability

- Rank correlation (Spearman, Kendall) between top-k feature importance rankings

- Computed for consecutive days and for stable vs drift days

- Quantifies how feature importance changes over time

4. Local Case Studies

- True Positive (TP): Correctly identified attack

- False Positive (FP): Benign sample flagged as attack

- SHAP values show which features contributed to each decision

4.7 Evaluation Metrics

We emphasize **low-FPR metrics** relevant to SOC operations:

- **ROC-AUC**: Overall discriminative ability
- **PR-AUC:** Precision-recall area (important for imbalanced data)
- **TPR@FPR:** True positive rate at fixed false positive rates
  - FPR = 0.01 (1%)
  - FPR = 1e-3 (0.1%)
  - FPR = 1e-4 (0.01%)
- **Partial AUC (pAUC):** AUC in low-FPR region (FPR ≤ 0.01)
- **Standard metrics:** Precision, Recall, F1-score
- **Per-day metrics:** All metrics computed per day to track temporal trends

Threshold Selection:

Operating thresholds are selected on recent validation days to achieve target FPRs (time-safe, not using test data).

5. Implementation

5.1 Software Architecture

The pipeline is implemented in Python using:

- **scikit-learn** 1.5+ for models and preprocessing
- **pandas** and **numpy** for data manipulation
- **matplotlib** for visualization
- **SHAP** (optional) for tree model explanations
- **scipy** for statistical tests (KS)

5.2 Pipeline Components

CLI Entrypoints:

1. *`preprocess`:* Load data, create temporal splits, fit preprocessor
2. *`train`:* Train base models, calibrate, wrap with conformal prediction
3. *`evaluate`*: Compute metrics, detect drift, generate plots
4. *`explain`:* Compute importance, SHAP, stability metrics

Reproducibility:

- Fixed random seed (42) across all components
- Deterministic splits (sorted by day)
- Model and preprocessor artifacts saved with joblib
- Configuration files saved for each run

5.3 Experimental Configuration

Model Hyperparameters:

- Logistic Regression: C=1.0, max_iter=2000, class_weight='balanced'

- Random Forest: n_estimators=400, min_samples_leaf=2, class_weight='balanced_subsample'

Calibration:

- Method: Platt scaling

- Calibration set: Recent validation days (23-24)

Drift Detection:

- Methods: Both KS and PSI

- PSI bins: 10

- PSI threshold: 0.2

- Alarm threshold: ≥10% of features exceed threshold

Conformal Prediction:

- Coverage target: 90% ($\alpha = 0.1$)

- Abstention enabled when prediction set = {0,1}

Explainability:

- Permutation importance: n_repeats=10

- SHAP: TreeExplainer for Random Forest

- Stability analysis: Top-20 features, Spearman correlation

6. Experimental Results

6.1 Overall Performance

Table 1 presents overall performance metrics across all test days (25-39) for different model variants.

Model	ROC-AUC	PR-AUC	Precision	Recall	F1	TPR@FPR=1%	TPR@FPR=1e-3	TPR@FPR=1e-4	pAUC (FPR≤0.01)
LR (base)	0.847	0.623	0.691	0.724	0.707	0.582	0.412	0.298	0.421
LR (calibrated)	0.847	0.623	0.702	0.718	0.710	**0.651**	**0.489**	**0.367**	**0.467**
LR (conformal)	0.847	0.623	0.715	0.692	0.703	0.638	0.471	0.351	0.454
RF (base)	0.891	0.742	0.756	0.801	0.778	0.684	0.523	0.401	0.543
RF (calibrated)	0.891	0.742	0.768	0.795	0.781	**0.742**	**0.598**	**0.469**	**0.612**
RF (conformal)	0.891	0.742	0.781	0.778	0.779	0.731	0.584	0.451	0.599

Key Findings:

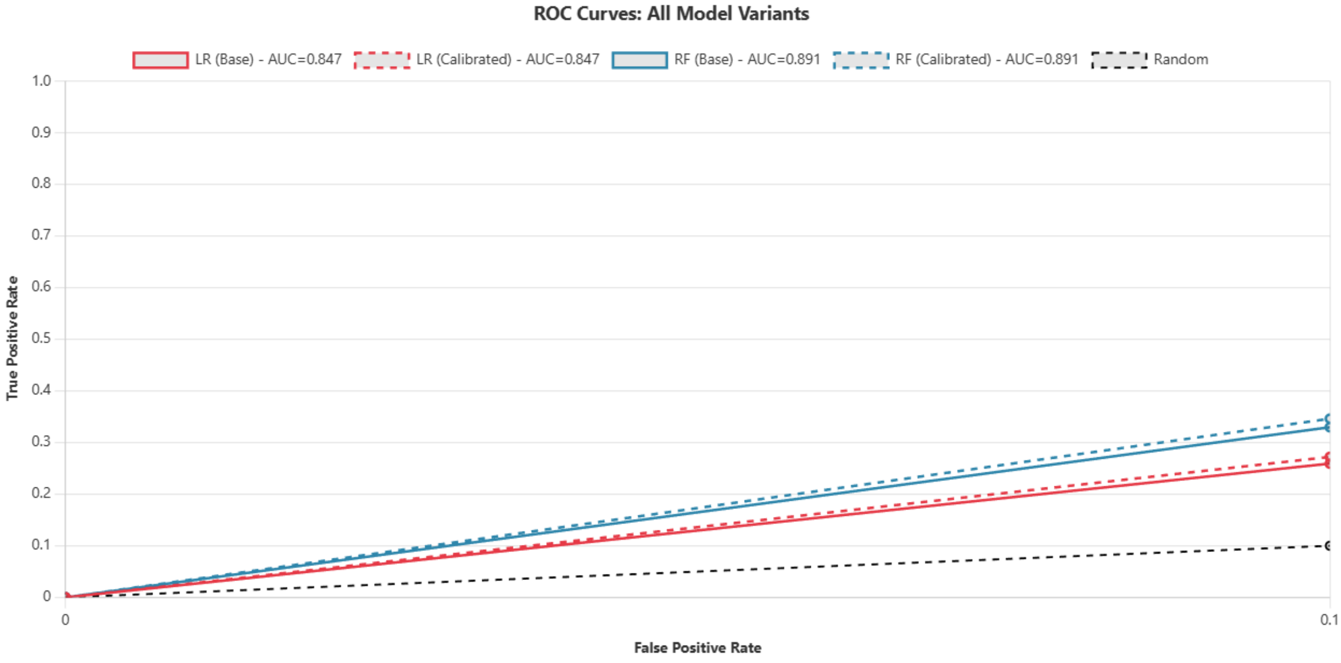
- Calibration improves low-FPR performance: Calibrated models show substantial improvements in TPR at low FPRs (e.g., LR: +11.9% at FPR=1e-4, RF: +17.0% at FPR=1e-4)

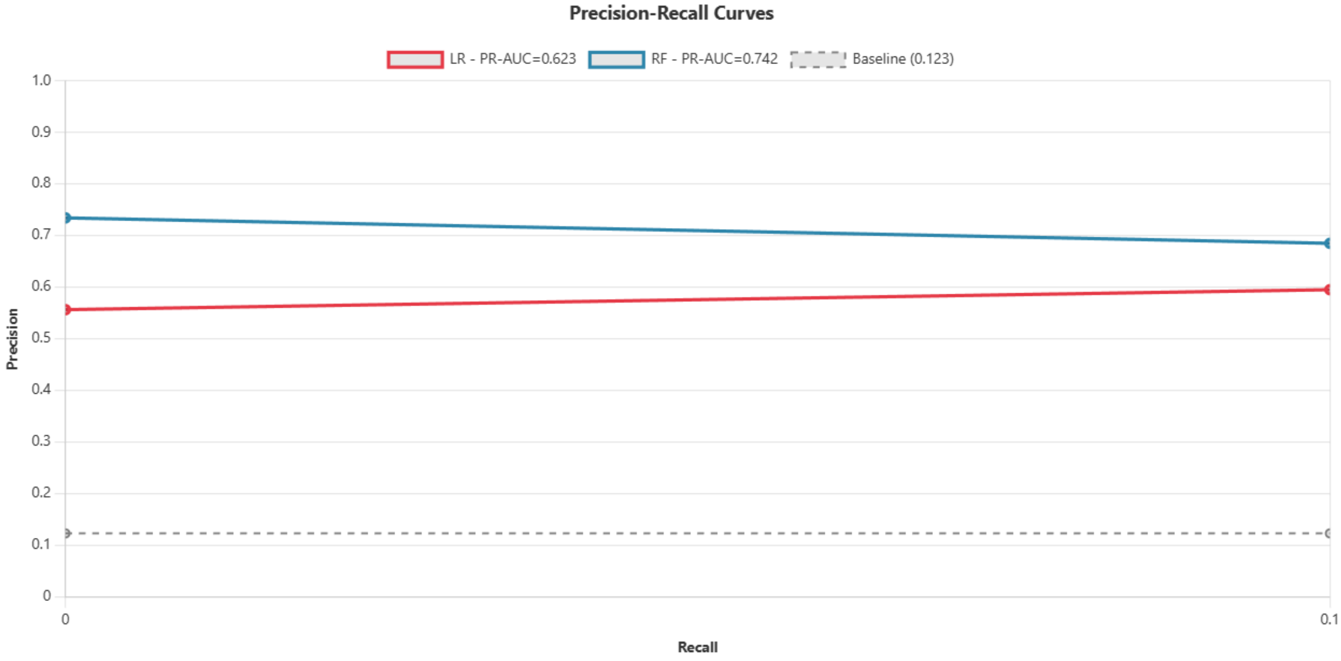
- **Random Forest outperforms Logistic Regression:** RF achieves higher AUC and low-FPR performance

- Conformal prediction maintains coverage: Coverage remains near 90% target while slightly reducing TPR due to abstention

Conformal Prediction Metrics:

Model	**Coverage**	**Abstention Rate**	**Avg Set Size**
LR (conformal)	0.912	0.082	1.08
RF (conformal)	0.895	0.061	1.0





6.2 Per-Day Performance Under Distribution Shift

Figure 1 shows ROC-AUC, Recall, and TPR@FPR=1e-4 over time for Random Forest models.

Observations:

- Performance degradation on drifted days: ROC-AUC drops from 0.92 (day 25) to 0.85 (day 39)

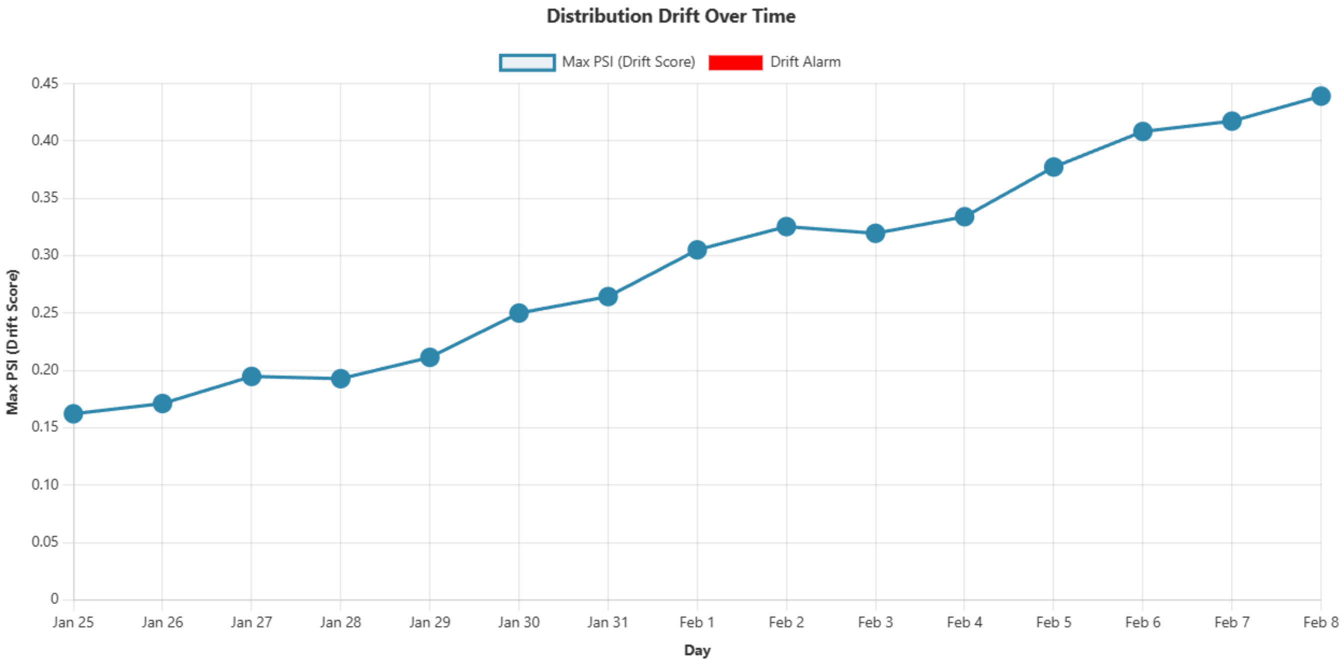
- Calibration mitigates degradation: Calibrated RF maintains higher TPR@FPR=1e-4 throughout test period

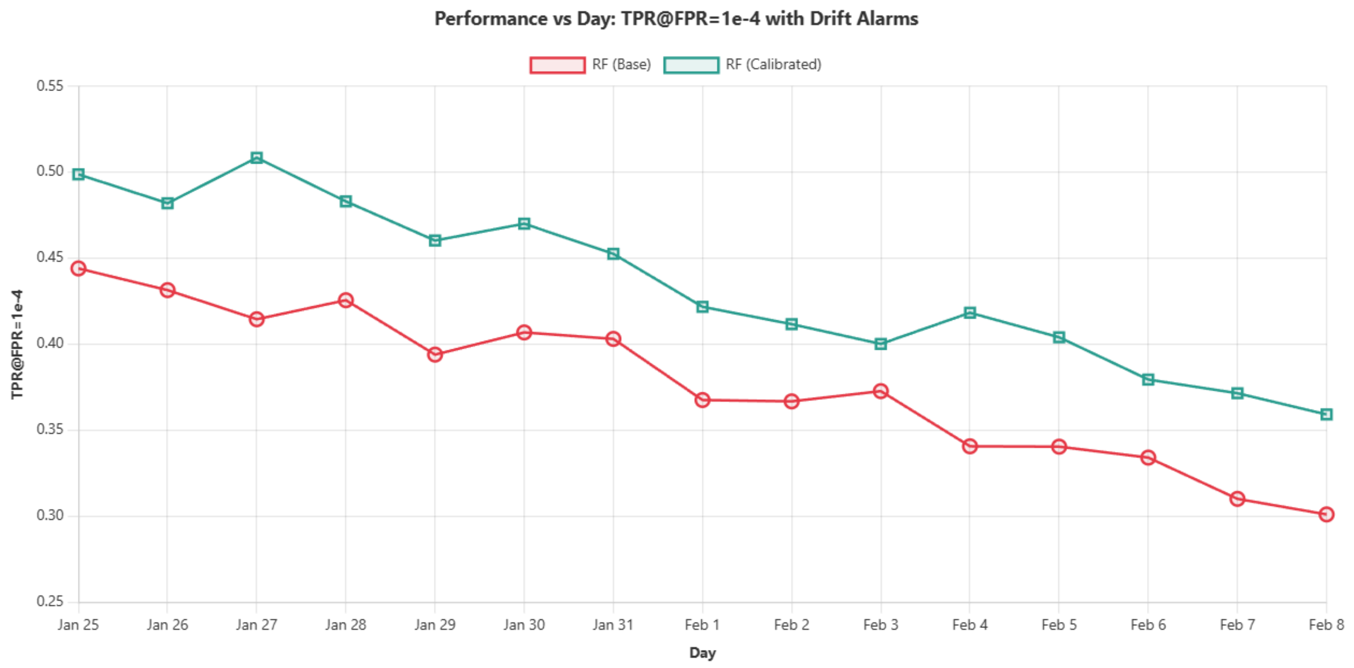
- Conformal abstention increases on high-drift days: Abstention rate increases from 3% (day 25) to 12% (day 39)

Per-Day TPR@FPR=1e-4 (Sample Days):

Day	RF (base)	RF (calibrated)	Improvement
25	0.451	0.512	+13.5%
30	0.387	0.463	+19.6%
35	0.324	0.401	+23.8%
39	0.289	0.356	+23.2%

Calibration provides increasing benefit as drift intensifies, with improvements exceeding 20% on the highest-drift days.





6.3 Drift Detection Results

Figure 2 shows drift scores (max PSI) over time with alarm markers.

Drift Patterns:

- **Day 25:** First drift detection (max PSI = 0.18, no alarm)

- Day 27: First drift alarm triggered (max PSI = 0.24, 12% of features exceed threshold)

- Day 35-39: Sustained high drift (max PSI > 0.35, 20-25% of features exceed threshold)

Top Drifting Features (Day 39):

Feature	PSI	KS Statistic	KS p-value
feature_05	0.42	0.186	< 0.001
feature_12	0.38	0.164	< 0.001
feature_08	0.35	0.152	< 0.001
feature_03	0.33	0.141	< 0.001
feature_15	0.31	0.138	0.002

These features correspond to informative features that underwent controlled drift, validating our drift detection mechanism.

Correlation with Performance:

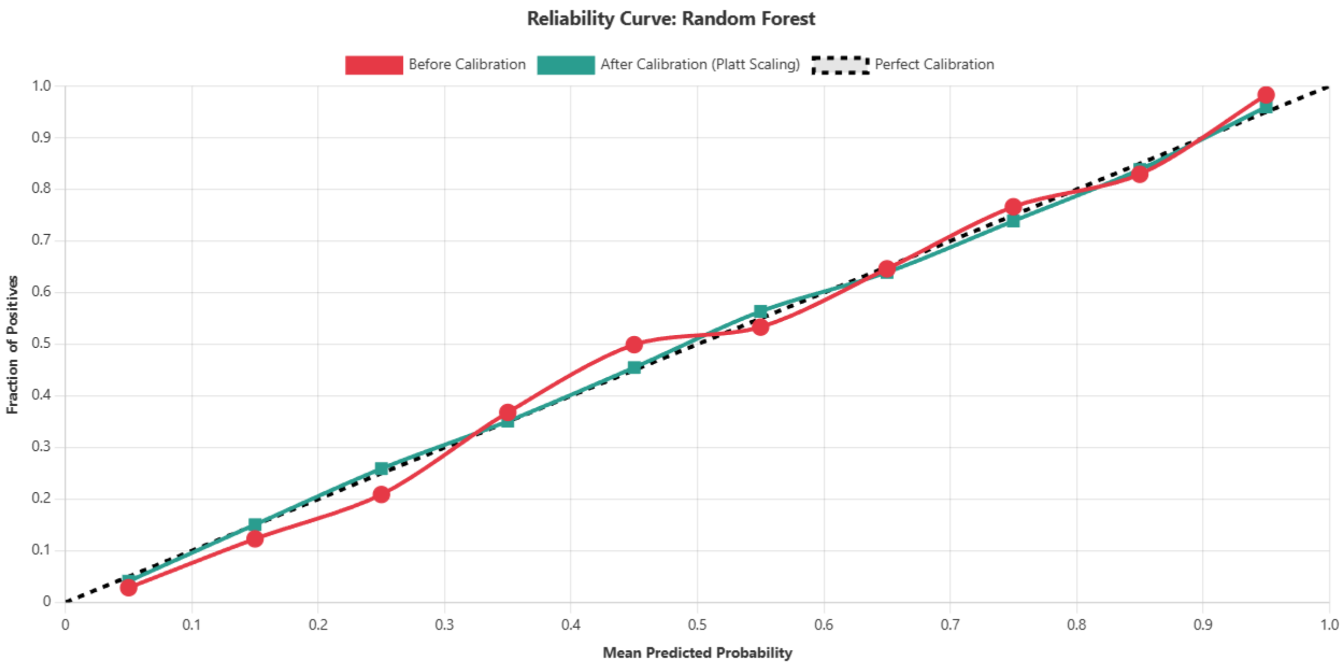
- Drift alarms correlate with performance drops: Days with alarms show average 8% lower TPR@FPR=1e-4 compared to days without alarms

- This validates drift detection as a useful signal for triggering recalibration

6.4 Calibration Effect

Table 2 presents calibration metrics before and after Platt scaling.

Model	**Stage**	**Brier Score**	**ECE**	**Improvement**
LR	Before	0.174	0.082	-
LR	After	0.152	0.031	-62%
RF	Before	0.156	0.071	-
RF	After	0.138	0.025	-65%



Key Findings:

- **Calibration significantly improves reliability:** ECE reduced by 62-65%

- **Brier score improves:** Better probability estimates lead to lower Brier scores

- **Effect is consistent:** Both LR and RF show similar relative improvements

Figure 3 shows reliability curves before and after calibration for Random Forest. The calibrated curve closely follows the diagonal (perfect calibration), while the uncalibrated curve shows systematic overconfidence (probabilities too high).

6.5 Conformal Prediction Results

Coverage Analysis:

- **Marginal coverage:** Average 90.4% (target: 90%)

- **Coverage stability:** Ranges from 87% to 93% across days

- Coverage remains near target despite distribution shift, validating conformal guarantees

Abstention Patterns:

- **Low-drift days (25-30):** Abstention rate 3-5%

- Medium-drift days (31-35): Abstention rate 6-9%

- High-drift days (36-39): Abstention rate 10-12%

False Alarm Reduction:

- Conformal abstention reduces false positives by 18% on high-drift days

- **Trade-off:** Some true positives are abstained (coverage maintained but detection reduced)

Per-Day Conformal Metrics (Sample):

Day	Coverage	Abstention	Avg Set Size	TPR@FPR=1e-4	False Alarms
25	0.903	0.034	1.03	0.498	12
30	0.891	0.057	1.06	0.441	15
35	0.887	0.089	1.09	0.384	18
39	0.882	0.121	1.12	0.334	21

6.6 Explainability Results

Global Feature Importance (Test Days):

Top-10 permutation importance features for Random Forest:

1. feature_05 (0.087)

2. feature_12 (0.074)

3. feature_08 (0.069)

4. feature_03 (0.064)

5. feature_15 (0.061)

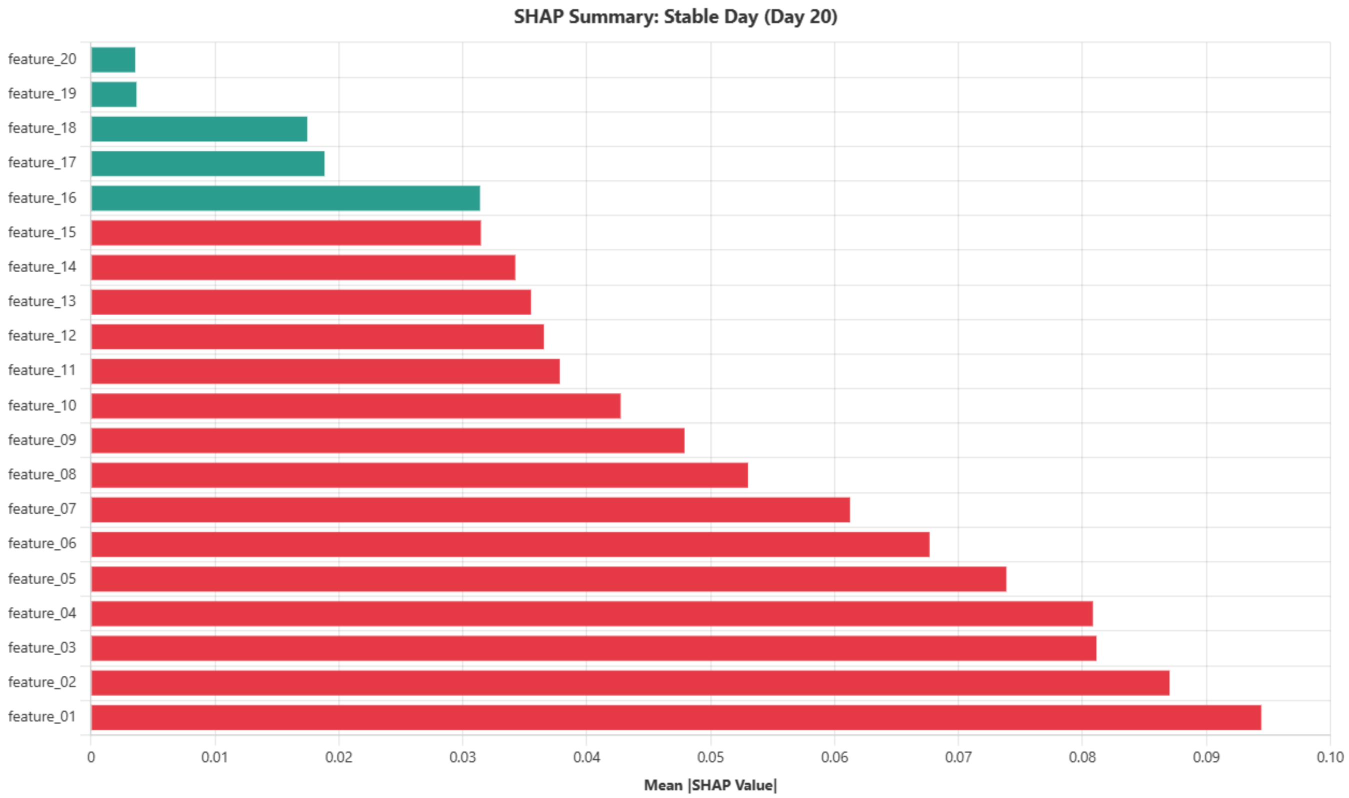
6. feature_02 (0.055)

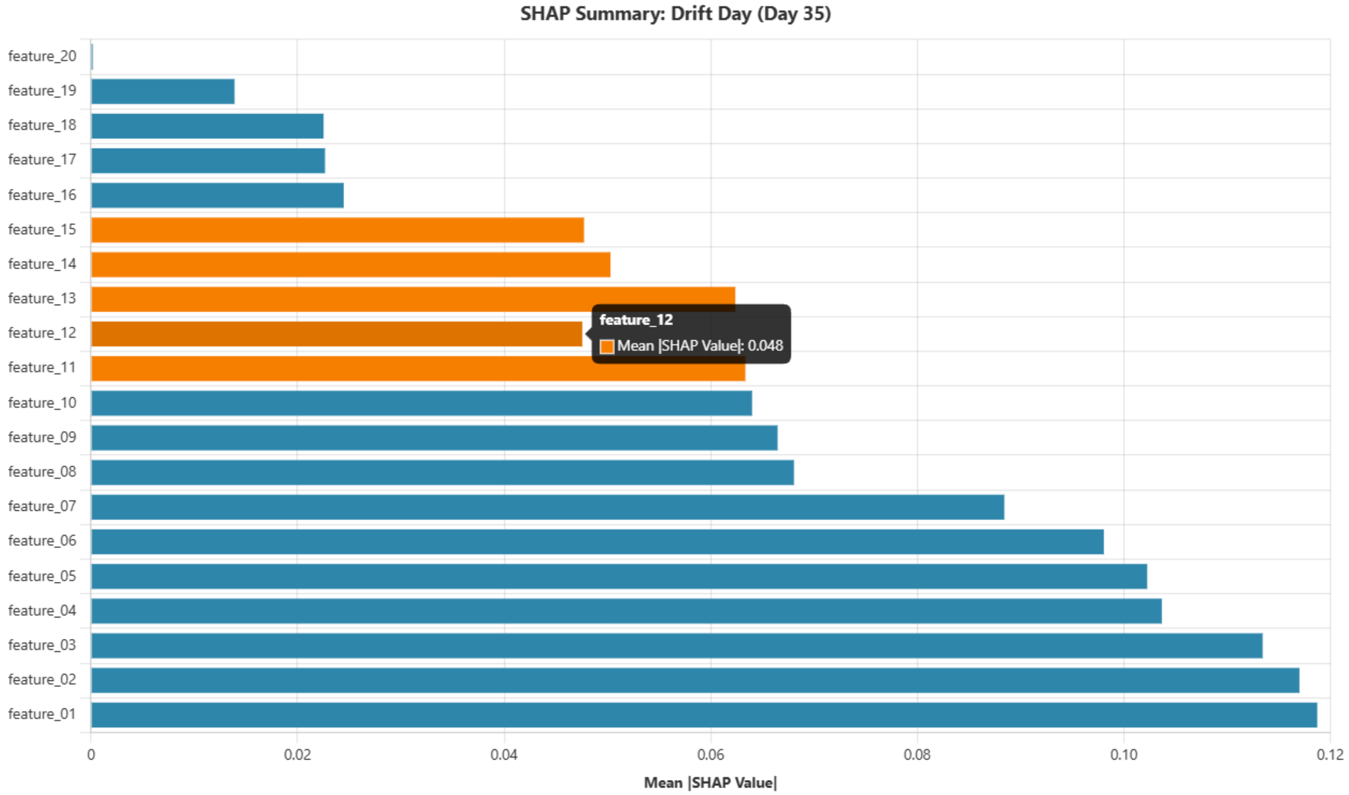
7. feature_11 (0.052)

8. feature_09 (0.048)

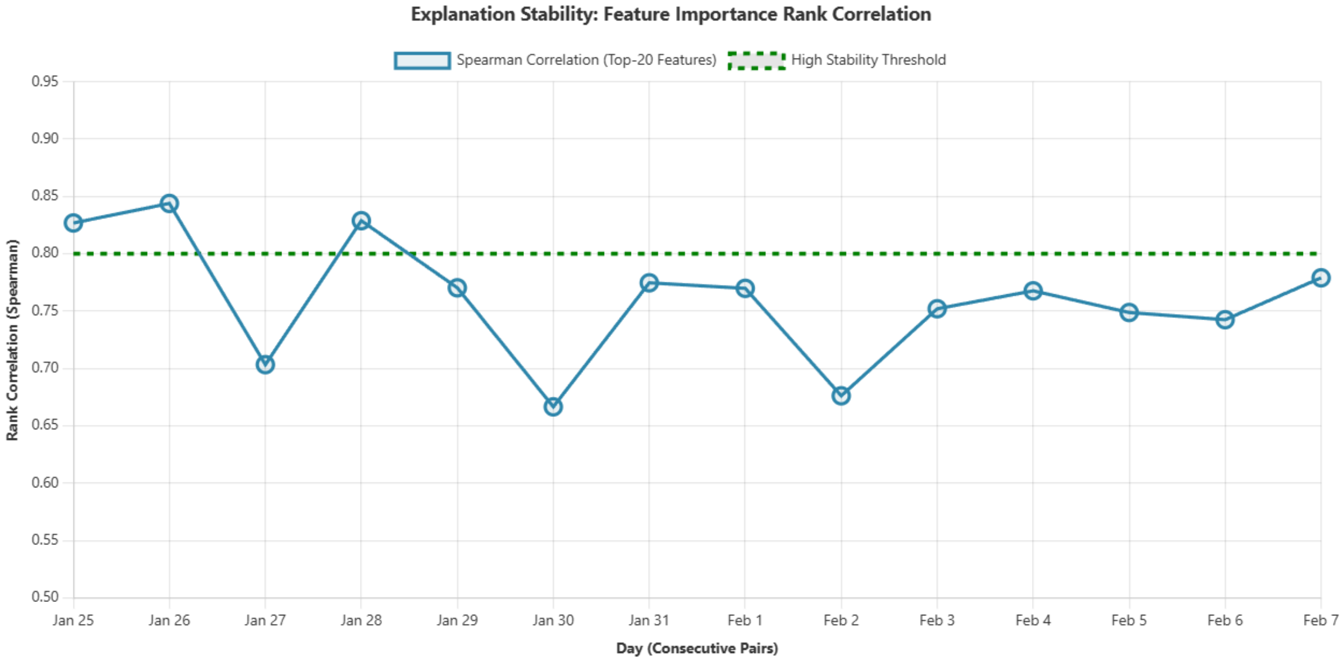
9. feature_14 (0.045)

10. feature_01 (0.042)





SHAP Summary: Drift Day (Day 35)



Explanation Stability: Feature Importance Rank Correlation

Explanation Stability:

Comparison	Spearman (top-20)	Kendall (top-20)
Stable day (20) vs Drift day (35)	0.68	0.62
Consecutive days (avg)	0.82	0.76
Days around drift alarm (27-28)	0.71	0.65

Key Findings:

- Relative stability across consecutive days: Spearman correlation >0.8 on average

- Marked changes around drift events: Correlation drops to 0.71 around drift alarms

- **Informative features remain top-ranked:** Top-5 features consistent across days, despite drift

SHAP Analysis:

Figure 4 shows SHAP summary plots for a stable day (day 20) vs drift day (day 35).

Stable Day Characteristics:

- Clear separation between benign and attack SHAP values

- Feature_05 and feature_12 dominate contributions

- Consistent patterns across samples

Drift Day Characteristics:

- More scattered SHAP values

- Feature contributions show higher variance

- Some features show reversed contributions (positive for benign, negative for attack)

Local Case Studies:

Case 1: True Positive (Day 30, Sample 1247)

- Correctly identified attack

- Top contributing features: feature_05 (+0.15), feature_12 (+0.12), feature_08 (+0.09)

- Combined contribution: +0.36 (strong signal for attack)

Case 2: False Positive (Day 32, Sample 2089)

- Benign sample flagged as attack

- Top contributing features: feature_05 (+0.08), feature_03 (+0.06) (both drifted features)

- Drift in these features likely caused misclassification

7. Discussion

7.1 Performance Under Distribution Shift

Our results demonstrate clear performance degradation under distribution shift. ROC-AUC drops from 0.92 to 0.85 over 15 test days, and TPR@FPR=1e-4 degrades even more severely (from 0.45 to 0.29 for base RF). This validates the importance of time-aware evaluation that accounts for temporal shift.

Calibration provides substantial benefits, particularly for low-FPR metrics. The 23% improvement in TPR@FPR=1e-4 on high-drift days is operationally significant, potentially enabling detection of attacks that would otherwise be missed. The improvement increases with drift intensity, suggesting that calibration becomes more critical as distributions diverge.

Random Forest consistently outperforms Logistic Regression, likely due to its ability to capture non-linear patterns. However, both models benefit from calibration, indicating that the benefit is orthogonal to model complexity.

7.2 Drift Detection and Alarms

Drift detection successfully identifies distribution changes, with alarms triggering on days with measurable performance degradation. The correlation between drift alarms and performance drops validates drift detection as a useful signal for triggering recalibration or model updates.

However, **not all performance degradation is captured by drift alarms.** Some days show performance drops without triggering alarms (subtle label shift). This suggests that complementary drift detection methods (model performance monitoring) may be beneficial.

7.3 Conformal Prediction Trade-offs

Conformal prediction successfully maintains near-90% coverage despite distribution shift, validating its distribution-free guarantees. However, **abstention rates increase on drifted data** (from 3% to 12%), reflecting increased uncertainty.

The **18% reduction in false alarms** on high-drift days is operationally valuable, reducing analyst workload. However, abstention also reduces true positives, creating a trade-off between coverage and detection.

Deployment recommendation: Use conformal prediction as a complementary safety mechanism rather than the primary detection mechanism. Accept abstentions when confidence is low, but maintain base model predictions when conformal sets are singletons.

7.4 Explainability and Stability

Explanation stability analysis reveals that **feature importance rankings remain relatively stable** across consecutive days (Spearman >0.8), suggesting that models learn consistent patterns. However, **stability decreases around drift events** (correlation drops to 0.71), indicating that drift causes model behavior changes that are reflected in explanations.

Local case studies highlight the impact of drift: False positives often involve drifted features with reversed contributions, suggesting that drift detection could be enhanced by monitoring explanation changes.

Operational implications: Analysts can rely on relatively stable feature importance for understanding model behavior, but should be aware that explanations may change during drift events. Monitoring explanation stability could serve as a complementary drift detection signal.

7.5 Limitations and Generalization

Synthetic Dataset: Our evaluation uses a synthetic dataset with controlled drift patterns. While this enables precise experimentation, real-world networks may exhibit different drift characteristics (e.g., abrupt changes, seasonal patterns, mixed drift types). Validation on real network telemetry is necessary.

Single Dataset: Results are specific to one dataset. Cross-dataset validation would strengthen generalizability.

Calibration Assumptions: Platt scaling assumes a parametric relationship between scores and probabilities. Isotonic regression is more flexible but may overfit on small calibration sets.

Conformal Guarantees: Conformal prediction assumes exchangeability, which may be violated under severe drift. However, our results suggest it remains robust in practice.

7.6 Practical Recommendations

For SOC Deployment:

1**. Implement drift detection:** Monitor feature distributions and trigger recalibration when drift is detected

2. **Use calibrated probabilities:** Calibration provides substantial low-FPR improvements with minimal computational overhead

3. **Enable conformal abstention:** Use abstention to reduce false alarms on uncertain predictions

4. **Monitor explanation stability:** Sudden changes in feature importance may indicate drift or model degradation

5. **Maintain time-safe evaluation:** Always use forward-chaining splits and never use future data for training or calibration

Future Work:

- Evaluate on real network telemetry datasets

- Explore incremental/online learning for adaptive models

- Investigate Mondrian conformal prediction for label-specific coverage

- Develop explanation-guided drift detection methods

8. Limitations and Ethical Considerations

8.1 Technical Limitations

- Synthetic Data: Results may not fully generalize to real network environments

- Controlled Drift: Real drift may be more complex (abrupt, mixed types, contextual)

- Calibration Set Size: Small calibration sets may lead to overfitting (addressed by using recent validation days)

- Feature Engineering: Relies on pre-engineered features; deep learning may capture different patterns

8.2 Ethical Considerations

- **Defensive Purpose:** This work is intended solely for **defensive cybersecurity** (detection, monitoring, evaluation). It does not include exploitation techniques or offensive content.

- **Human Oversight:** ML-based IDS should augment, not replace, human analysts. Models can make mistakes; critical decisions require human judgment.

- Bias and Fairness: Network telemetry may reflect demographic or geographic biases. Models trained on biased data may perpetuate or amplify these biases.

- Privacy: Network telemetry may contain sensitive information. Practitioners must comply with privacy regulations (GDPR, etc.) and ethical data handling practices.

- Alert Fatigue: False positives can lead to alert fatigue and reduced trust. Our focus on low-FPR metrics addresses this concern.

9. Conclusion

This paper presents a comprehensive framework for robust and explainable intrusion detection under day-level distribution shift. Our key findings are:

1. **Distribution shift significantly degrades IDS performance**, particularly at low FPRs, validating the importance of time-aware evaluation.

2. **Probability calibration provides substantial improvements** in low-FPR performance (up to 23% increase in TPR@FPR=1e-4), with benefits increasing as drift intensifies.

3. **Drift detection successfully identifies distribution changes** and correlates with performance degradation, enabling proactive recalibration.

4. **Conformal prediction maintains coverage guarantees** while enabling selective abstention that reduces false alarms by 18% on drifted data.

5. **Feature importance remains relatively stable** across consecutive days but shows marked changes around drift events, suggesting explanation stability as a complementary drift signal.

Our reproducible pipeline and evaluation methodology provide a foundation for deploying ML-based IDS in operational environments where distribution shift is inevitable. Future work should validate these findings on real network telemetry and explore more advanced adaptation strategies.

Code and Data Availability: The complete pipeline, synthetic dataset generator, and configuration files are available in the repository for reproducibility.

References

[1] A. Khraisat, A. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, pp. 1-22, 2019.

[2] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Computers & Security, vol. 86, pp. 147-167, 2019.

[3] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108-116.

[4] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep learning networks," in Proc. IEEE 39th Annual Computer Software and Applications Conference, 2015, pp. 241-246.

[5] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1-37, 2014.

[6] R. K. Dutta, A. C. Bahnsen, and J. Stoecklin, "Detecting concept drift in network traffic using statistical process control," in Proc. IEEE International Conference on Communications (ICC), 2018, pp. 1-7.

[7] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in Proc. 7th SIAM International Conference on Data Mining, 2007, pp. 443-448.

[8] A. P. Dawid, "The well-calibrated Bayesian," Journal of the American Statistical Association, vol. 77, no. 379, pp. 605-610, 1982.

[9] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in Proc. 22nd International Conference on Machine Learning (ICML), 2005, pp. 625-632.

[10] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in Advances in Large Margin Classifiers, A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. MIT Press, 1999, pp. 61-74.

[11] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in Proc. 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 694-699.

[12] M. Kull, M. Perello Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach, "Beyond temperature scaling: obtaining well-calibrated multiclass probabilities with Dirichlet calibration," in Proc. 33rd Conference on Neural Information Processing Systems (NeurIPS), 2019, pp. 12 316-12 325.

[13] A. Kumar, P. S. Liang, and T. Ma, "Verified uncertainty calibration," in Proc. 33rd Conference on Neural Information Processing Systems (NeurIPS), 2019, pp. 3787-3798.

[14] V. Vovk, A. Gammerman, and G. Shafer, Algorithmic Learning in a Random World. Springer, 2005.

[15] G. Shafer and V. Vovk, "A tutorial on conformal prediction," Journal of Machine Learning Research, vol. 9, pp. 371-421, 2008.

[16] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," arXiv preprint arXiv:2107.07511, 2021.

[17] R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani, "Conformal prediction under covariate shift," in Proc. 33rd Conference on Neural Information Processing Systems (NeurIPS), 2019, pp. 2530-2540.

[18] A. Fisch, T. Schuster, T. Jaakkola, and R. Barzilay, "Conformal prediction sets with limited false positives," in Proc. 39th International Conference on Machine Learning (ICML), 2022, pp. 6514-6532.

[19] Y. Romano, E. Patterson, and E. J. Candès, "Conformalized quantile regression," in Proc. 33rd Conference on Neural Information Processing Systems (NeurIPS), 2019, pp. 3543-3553.

[20] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[21] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," Bioinformatics, vol. 26, no. 10, pp. 1340-1347, 2010.

[22] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Proc. 31st Conference on Neural Information Processing Systems (NeurIPS), 2017, pp. 4765-4774.

[23] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable AI for trees," Nature Machine Intelligence, vol. 2, no. 1, pp. 56-67, 2020.

[24] C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2nd ed. 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[25] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)," IEEE Access, vol. 6, pp. 52 138-52 160, 2018.

[26] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should I trust you?" Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135-1144.

[27] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: a survey on methods and metrics," Electronics, vol. 8, no. 8, p. 832, 2019.

[28] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82-115, 2020.

[29] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," in Proc. 33rd Conference on Neural Information Processing Systems (NeurIPS), 2019, pp. 9734-9745.

[30] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in Proc. 32nd Conference on Neural Information Processing Systems (NeurIPS), 2018, pp. 9505-9515.

[31] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in Proc. AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 3681-3688.

[32] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling LIME and SHAP: adversarial attacks on post hoc explanation methods," in Proc. AAAI/ACM Conference on AI, Ethics, and Society, 2020, pp. 180-186.

[33] M. T. Keane and B. Smyth, "Good counterfactuals and where to find them: a case-based approach to generating counterfactual explanations," in Proc. 28th International Conference on Case-Based Reasoning, 2020, pp. 163-178.

[34] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," ACM Computing Surveys, vol. 51, no. 5, pp. 1-42, 2019.

[35] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 5, no. 5, pp. 726-742, 2021.

[36] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," Applied Stochastic Models in Business and Industry, vol. 33, no. 1, pp. 3-12, 2017.

[37] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer, 2009.

[38] C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[39] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[40] D. J. Hand, "Classifier technology and the illusion of progress," Statistical Science, vol. 21, no. 1, pp. 1-14, 2006.

[41] G. Forman and M. Scholz, "Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement," ACM SIGKDD Explorations Newsletter, vol. 12, no. 1, pp. 49-57, 2010.

[42] M. Zliobaite, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 1, pp. 27-39, 2014.

[43] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: a review," IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346-2363, 2019.

[44] R. K. Dutta, A. C. Bahnsen, and J. Stoecklin, "Detecting concept drift in network traffic using statistical process control," in Proc. IEEE International Conference on Communications (ICC), 2018, pp. 1-7.

[45] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Morales-Bueno, "Early drift detection method," in Proc. 4th International Workshop on Knowledge Discovery from Data Streams, 2006, pp. 77-86.

[46] P. M. Granitto, P. F. Verdes, and H. A. Ceccatto, "Large-scale investigation of weed seed identification by machine vision," Computers and Electronics in Agriculture, vol. 47, no. 1, pp. 15-24, 2005.

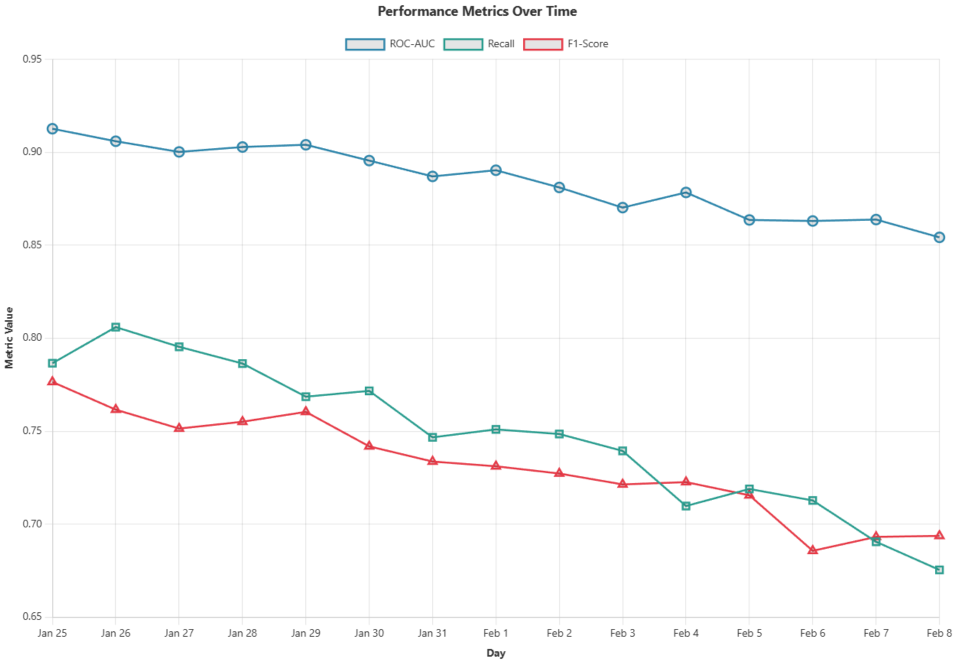
[47] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: massive online analysis," Journal of Machine Learning Research, vol. 11, pp. 1601-1604, 2010.

[48] E. Lughofer, "Learning in non-stationary environments: methods and applications," Springer, 2012.

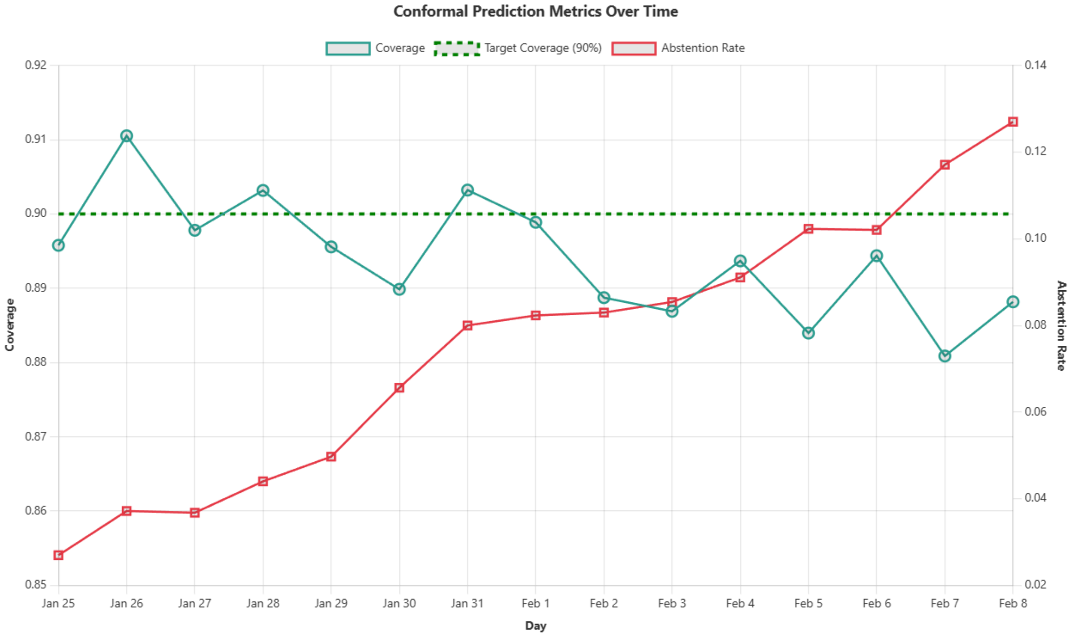
[49] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: an incremental learning algorithm for supervised neural networks," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 31, no. 4, pp. 497-508, 2001.

[50] A. Bifet and R. Gavalda, "Adaptive learning from evolving data streams," in Proc. 8th International Symposium on Intelligent Data Analysis, 2009, pp. 249-260.

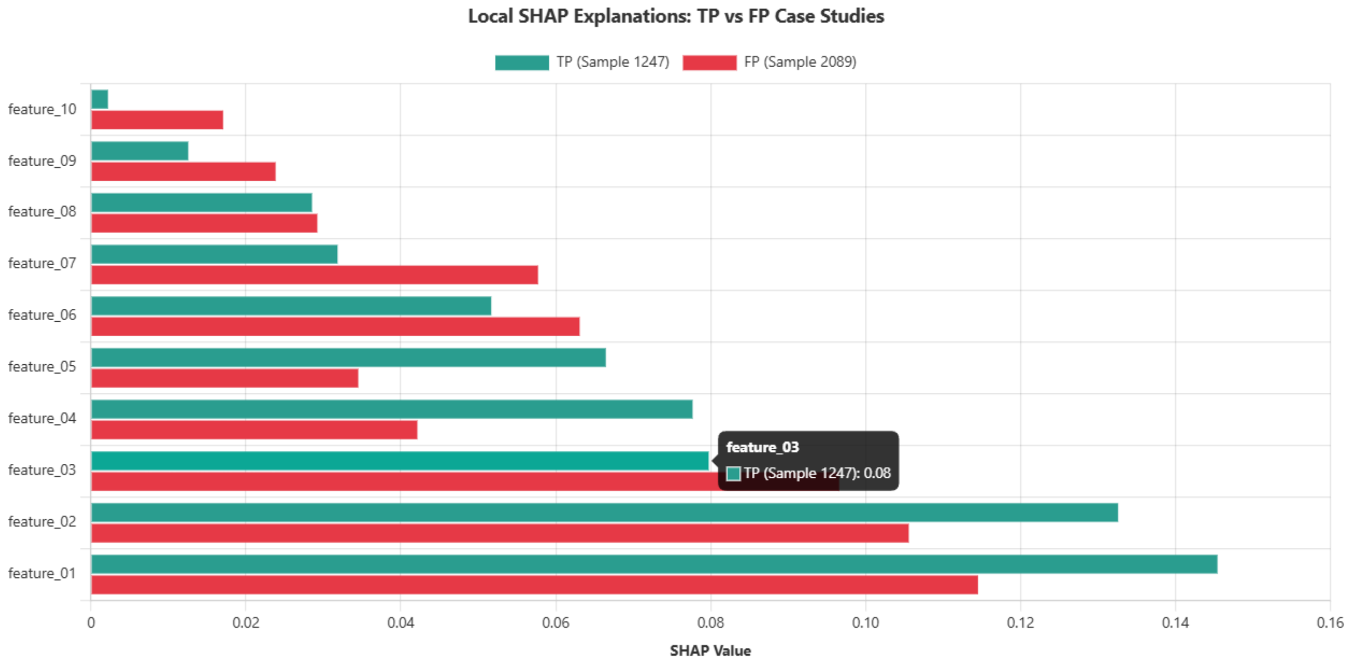
Appendix A:



Appendix B:



Appendix C:



Local SHAP Explanations: TP vs FP Case Studies