Dual Adaptive Windows toward Concept-Drift in Online Network Intrusion Detection

Name: Xiaowei Hu
Address: Institute of Information Engineering, Chinese Academy of Sciences
Email: huxiaowei@iie.ac.cn

Outline

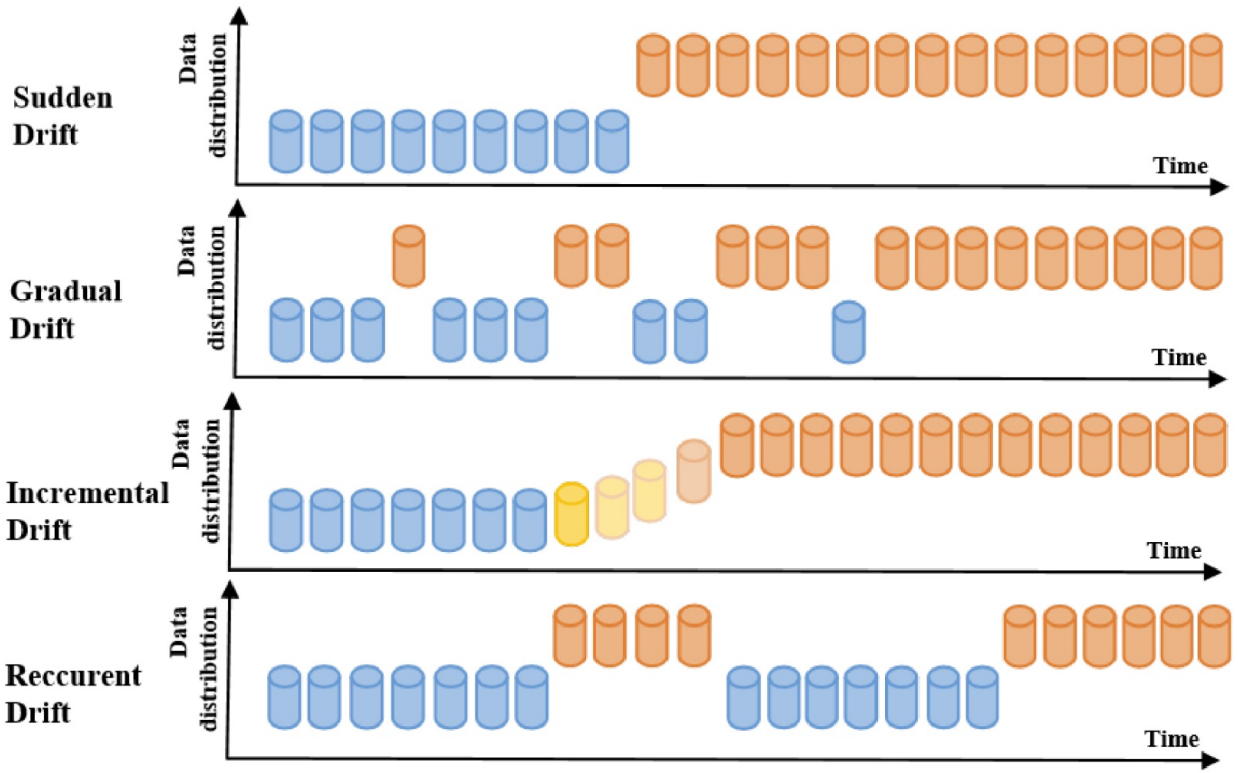
1. Background
2. Research Method
3. Experimental Evaluation
4. Conclusions

Background

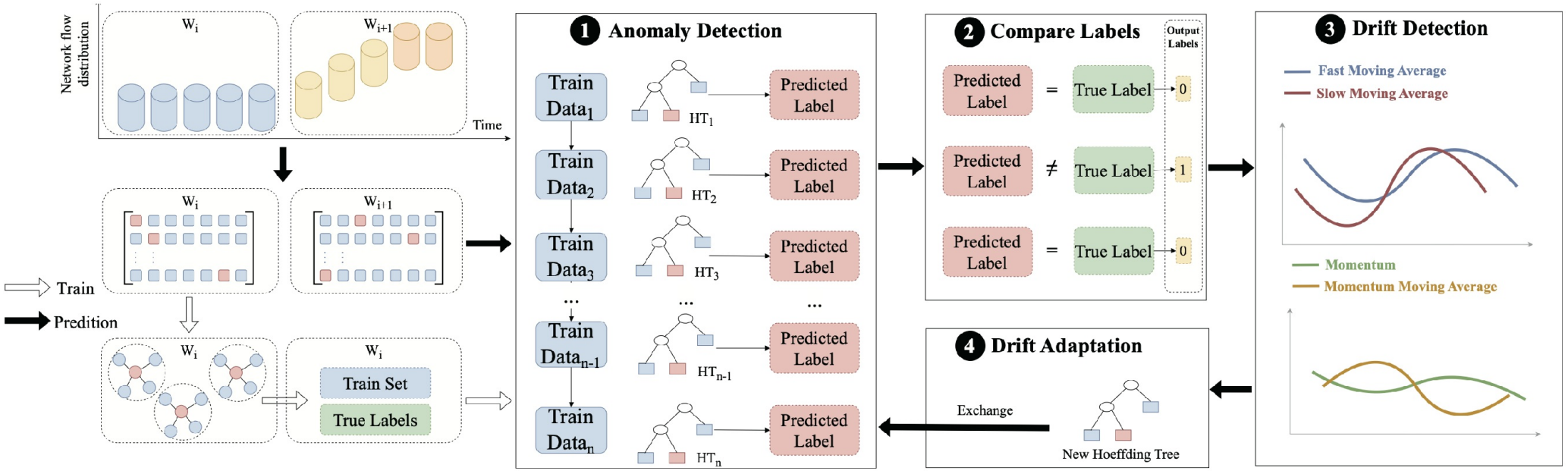
1. Concept drift refers to the non-stationarity of data stream distribution over time. In the context of network security, it manifests as unpredictable changes in the distribution of network data streams. The right figure shows four cases of concept drift

2. Conventional approaches typically assume that data streams within a network originate from a single source and exhibit a stable distribution.

3. Thus, when confronted with dynamic changes in data distribution, trained machine learning or deep learning intrusion detection models may fail to adapt to these shifts, consequently leading to a degradation in their predictive performance.



Research Method



Overall architecture

Research Method

0. Data preprocessing stage. To enhance the usability of traffic data and reduce noise interference, the framework adopts one-hot encoding, min-max normalization, and the BIRCH clustering algorithm for data preprocessing. One-hot encoding is used to process categorical features, making them compatible with the input format of machine learning models. Min-max normalization can normalize numerical features, avoiding the impact of feature scale differences on model training. Meanwhile, the BIRCH clustering algorithm is used to efficiently extract representative training data, improving the model's learning efficiency and generalization ability.

Research Method

1. Anomaly Detection Based on Hoeffding Tree. After data preprocessing, the framework adopts the Hoeffding decision tree algorithm to perform incremental learning on network traffic features within the current sliding window. This algorithm can continuously update the decision tree based on the order of data arrival without requiring a complete training set, adapting to the dynamic changes in the network environment. As a result, it can real-time predict detection labels based on the learned parameters of the decision tree.

Research Method

2. Comparison of Labels. By comparing the difference between the predicted labels and the true labels, the final result label is output. If the predicted label matches the true label, the result label is 0; otherwise, it is 1.

Research Method

3. Concept Drift Detection. Subsequently, the output labels are added to the fast window and the slow window, and the difference between the average values of these two windows, i.e., the momentum moving average difference of the dual windows, is calculated to measure changes in the data distribution. To further improve the accuracy of concept drift detection, this study introduces an estimator based on the difference between the momentum moving averages of the fast and slow windows. This estimator is capable of sensitively capturing the trend of changes in the data distribution. When the sign of the estimator at the current time step is opposite to that of the previous time step, it indicates a significant change in the data distribution, and the study triggers a concept drift signal.

Research Method

4. Concept Drift Adaptation. Once concept drift is detected, the framework automatically replaces the current Hoeffding tree classifier with an optimized classifier retrained under the new data distribution. Through this dynamic adaptation mechanism, the method effectively reduces the performance degradation caused by concept drift in traditional approaches, while avoiding the computational overhead of frequent global model retraining. This enhances the detection efficiency and adaptability of the intrusion detection system in dynamic and adversarial network environments.

Experimental Evaluation

- Dataset

This study utilized the CIC-IDS-2017 and NSL-KDD datasets for evaluation, with CIC-IDS-2017's statistical analysis illustrated by a table.

Attack Types	Count	Proportion
Infiltration	38	0.0088%
Brute Force	151	0.0348%
SQL Injection	12	0.0028%
XSS	27	0.0062%
DoS GoldenEye	7,567	1.7456%
DoS Hulk	158,469	36.5561%
DoS Slowhttptest	1,742	0.4019%
DoS Slowlori	4,001	0.9230%
Heartbleed	11	0.0025%
FTP Patator	3,973	0.9165%
SSH Patator	2,980	0.6874%
Botnet	738	0.1702%
DDoS	94,763	21.8602%
Portscan	159,023	36.6839%
SUM	433,495	100.0000%

Experimental Evaluation

- Experiment Setup

Classification Experiment

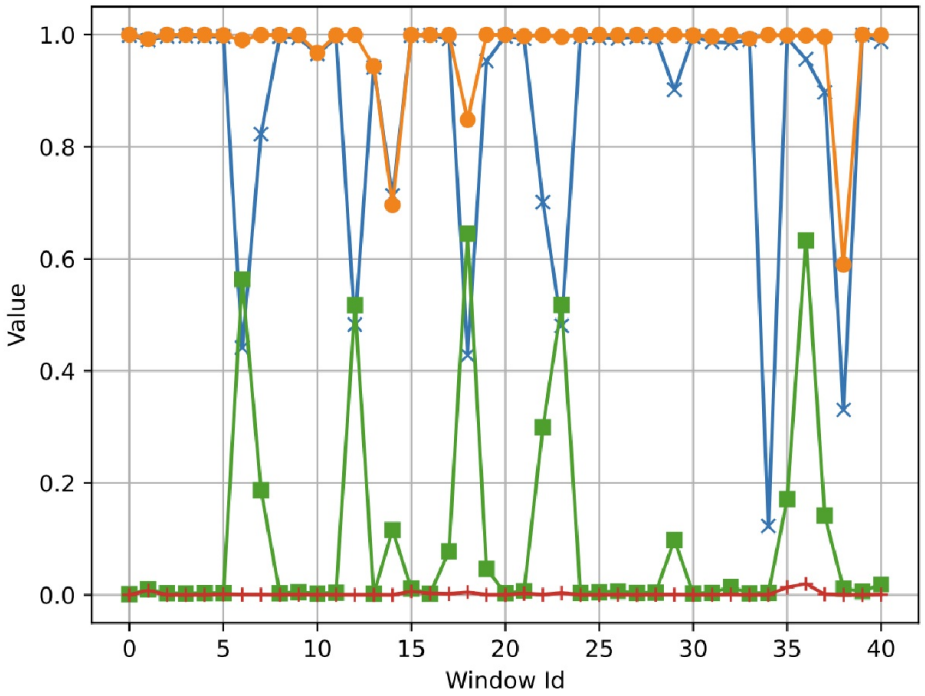
This study employed the same window segmentation method as prior works: 2,090,564 CIC-IDS-2017 traffic samples were partitioned into 41 non-overlapping sliding windows, each with a capacity of 50,000 entries, covering a continuous 5-day network activity cycle. Meanwhile, 125,973 NSL-KDD training samples were divided into 17 windows, each containing 7,000 samples. This approach was used to compare the impact of introducing DAWMA on the predictive performance of the baseline model.

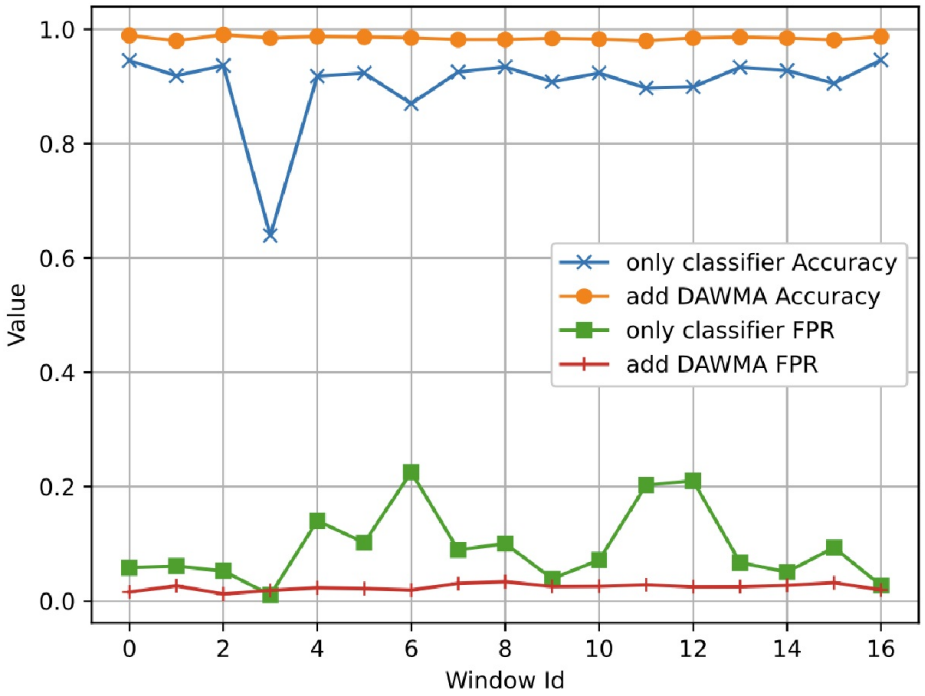
Intrusion Detection with Concept Drift Experiment

For the CIC-IDS2017 dataset, this study selected 1,396,914 traffic samples from the last three days for validation, with a window size of 50,000 to simulate real-world network data stream variations. For the NSL-KDD dataset, 22,544 test samples were used in the experiments with a window size of 2,000. This comparative framework evaluates the predictive performance against established baselines.

Experimental Evaluation

- Classification Experiment Result





Experimental Evaluation

- Intrusion Detection with Concept Drift Experiment Result

Method	Dataset	Accuracy	Precision	Recall	F1 Score
Naive Bayes*	NSL-KDD	0.8563	0.8199	0.8730	0.8345
	CIC-IDS2017	0.8787	0.9212	0.9067	0.8758
LightGBM*	NSL-KDD	0.9418	**0.9829**	0.8802	0.9284
	CIC-IDS2017	0.9401	0.8657	0.8020	0.7982
ARF+ADWIN	NSL-KDD	0.9340	0.9426	0.9028	0.9210
	CIC-IDS2017	0.9383	0.9362	**0.9972**	0.9532
INSOMNIA	NSL-KDD	0.5455	0.4666	0.8178	0.5940
	CIC-IDS2017	0.8656	0.3941	0.4857	0.3758
We Proposed DWOIDS	NSL-KDD	**0.9463**	0.9524	**0.9214**	**0.9364**
	CIC-IDS2017	**0.9608**	**0.9480**	0.9969	**0.9653**

* indicates that this method does not address concept drift.

Conclusions

First, concept drift is empirically confirmed to exist in network data streams, posing significant challenges to traditional classifiers and frequently causing performance degradation.

Second, the DAWMA strategy proposed in this study enables classifiers to achieve rapid recovery and stable operation, thus highlighting the critical importance of addressing concept drift in online intrusion detection for network flows.

Third, when concept drift occurs in the current window, the proposed framework promptly replaces outdated classifiers with newly trained ones and subsequently maintains effective monitoring of subsequent windows.