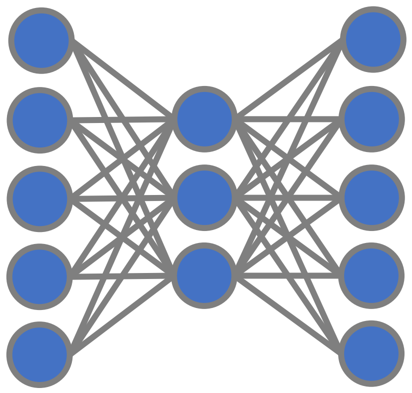
ENIDrift: A Fast and Adaptive Ensemble System for Network Intrusion Detection under Real-world Drift

Xian Wang
The Hong Kong University of Science and Technology
xwanggj@connect.ust.hk

Background: ML-based NIDS

Multiple machine learning (ML) and deep learning (DL) algorithms have been applied to network intrusion detection systems (NIDS).



The basic structure
of artificial neural network

Artificial Neural Network (ANN)
e.g., Kitsune [NDSS'18]

Principal Component Analysis (PCA)
e.g., Camacho et al. [TIFS'18]

Comparative Learning
e.g., CADE [USENIX Security'21]

Clustering Method
e.g., ACID [INFOCOM'21]

ML-based NIDSs have the following two main common advantages

- No need to prepare a signature database in advance
- Can detect unknown attacks

They can reach 95% (even 100%) accuracy, achieving high experimental performance.

Limitations 1: A narrow cover on real-world drift

There are three main sources of network behavior drift in real-world detection

- Concept drift [Ditzler et al., TKDE'13] of network distribution
- Imbalanced and changeable network packet ratio
- Well-crafted ML attack [Demontis et al., TDSC'19]

But none model has a wide cover on real-world drift.

Research Scope	ACDWM [30]	ACID [10]	CADE [57]	DeepAID [21]	Kitsune [35]	Whisper [18]
Concept Drift	●	○	●	○	○	○
Imbalanced Data	○	◐	○	○	◐	◐
Well-crafted ML attack	○	○	○	●	○	○

● considers the corresponding problem in its research scope. ◐ does not focus on the problem in its work but uses a framework that basically does not have the problem.

○ has the problem and does not solve it well in its scope.

Limitations 2: Low processing speed

Existing ML-based approaches pose high runtime overhead and have low processing speeds for incoming network packets.

300 packets / s

40 packets / s

7 packets / s

Current approach

Common small-scale network

Large-scale network

The latency of detection can leave time interval for the attack to cause severe damage to our system.

ENIDrift

ENIDrift is a fast and adaptive ensemble system for network intrusion detection under real-world drift. It focuses on dynamic and incremental network packet streams.

The design of ENIDrift has three components
1) iP2V, incremental feature extraction method based on Word2Vec
2) Sub-classifier generation module
3) ENIDrift update module

Advantage # 1 – having wide adaptability to real-world drift

Research Scope	ACDWM [30]	ACID [10]	CADE [57]	DeepAID [21]	Kitsune [35]	Whisper [18]	ENIDrift
Concept Drift	●	○	●	○	○	○	●
Imbalanced Data	○	◐	○	○	◐	◐	●
Well-crafted ML attack	○	○	○	●	○	○	●

● considers the corresponding problem in its research scope. ◐ does not focus on the problem in its work but uses a framework that basically does not have the problem.

○ has the problem and does not solve it well in its scope.

Table: Comparison of related approaches on their research scope

Advantage # 2 – improving network packet processing speed to the speed of common network environment

Threat Model

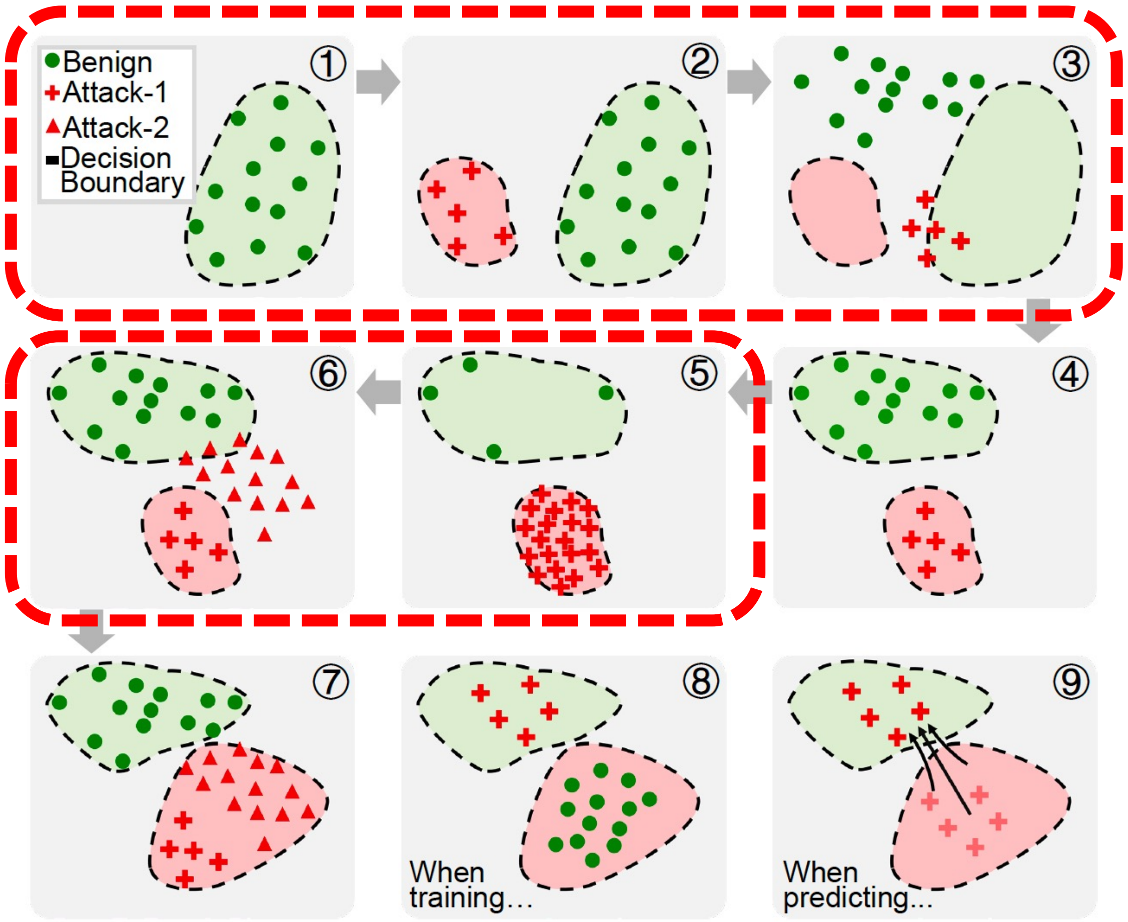


Figure: Examples of real world drift in network environment

We consider the drift of network environment in real world

 # 1 – Concept drift

Concept drift is the statistical properties of the target variable that change over time in unforeseen ways [Lu et al., TKDE'29]

For NIDS:

- the change of normal network behavior
- the change of anomalous network behavior

Threat Model

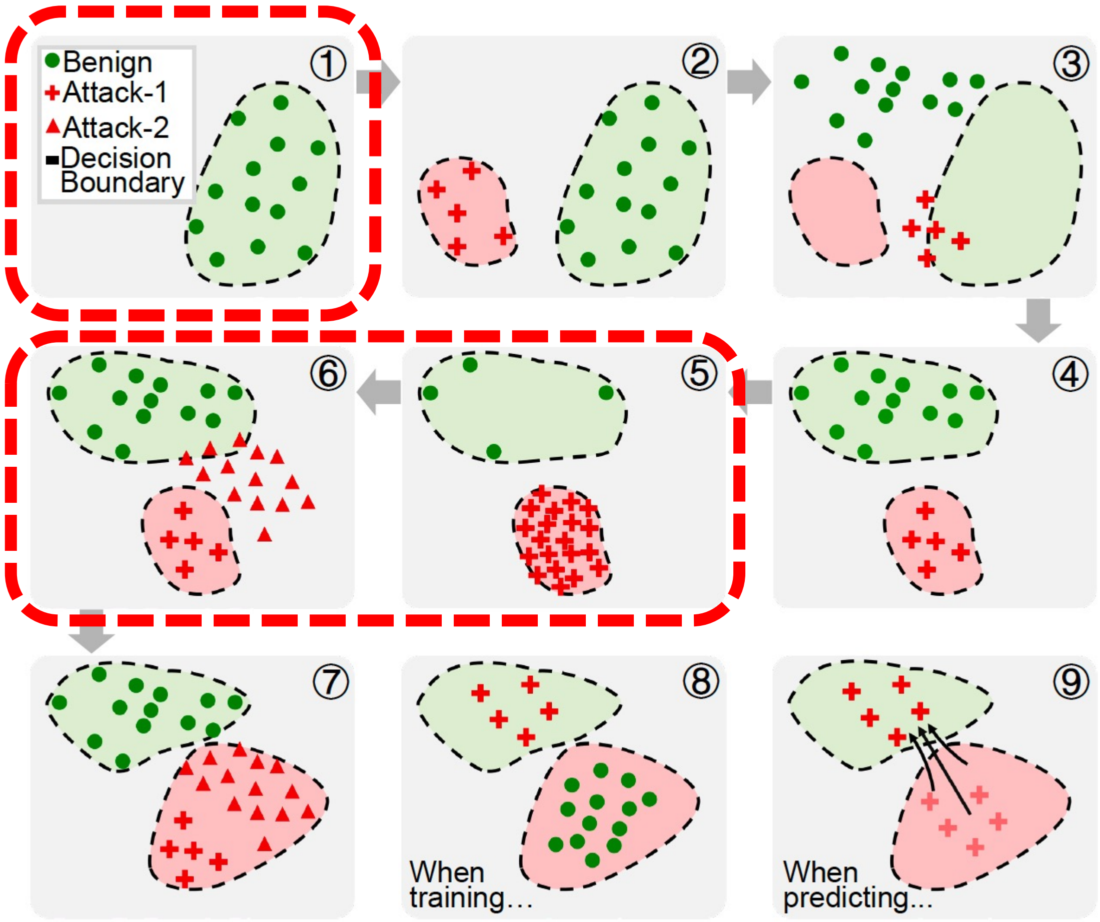


Figure: Examples of real world drift in network environment

We consider the drift of network distribution in real world

 # 2 – Imbalanced data

The ratio of normal and anomalous network packets is not 50:50 and always changing.

Specifically:

- the ratio is imbalanced
- sometimes very extreme
- always changing

Threat Model

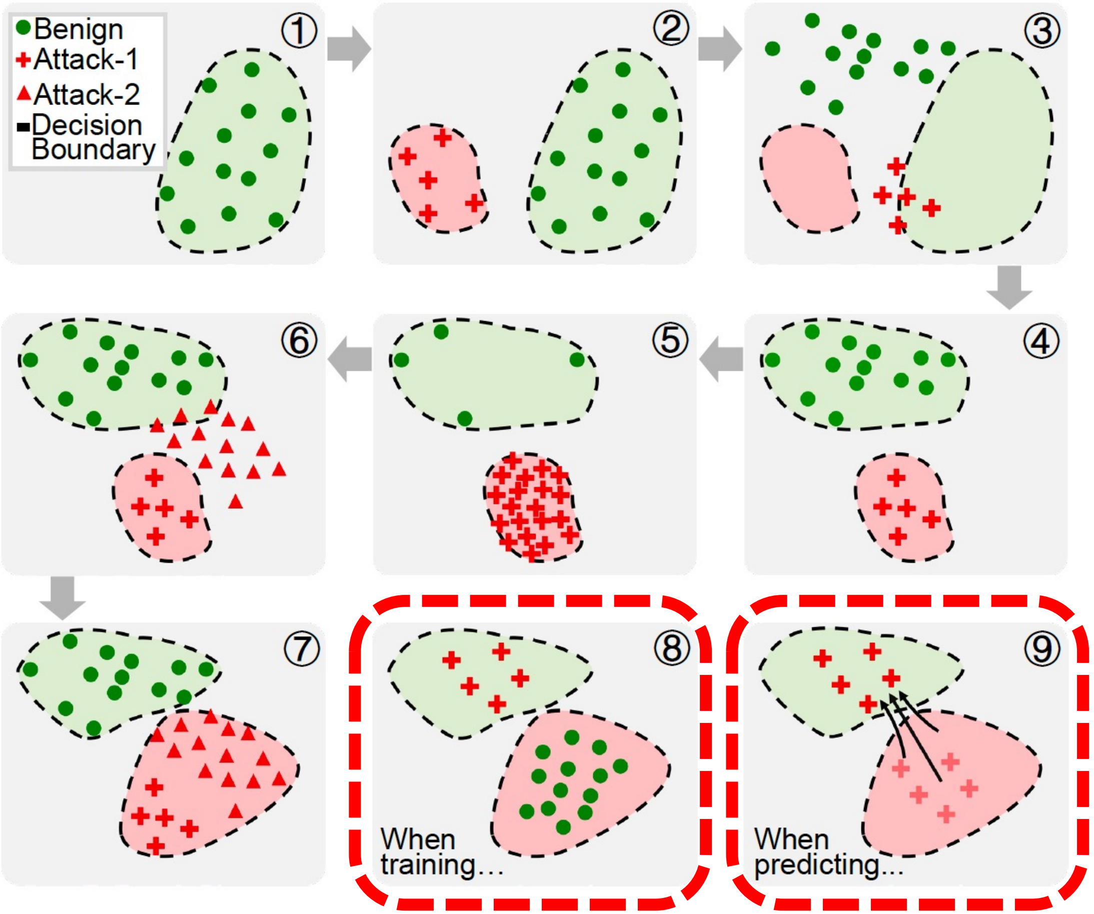


Figure: Examples of real world drift in network environment

We consider the drift of network distribution in real world

 # 3 – Well-crafted ML attack

There are well-crafted attacks for ML-based NIDS

We only consider two specific attacks:
- data contamination for training data
- adversarial attack for NIDS [MACGAN, WASA'20]

The ENIDrift NIDS - Overview

ENIDrift is a fast and adaptive ensemble system for network intrusion detection under real-world drift. It focuses on dynamic and incremental network packet streams.

The design of ENIDrift has three components

1) iP2V, incremental feature extraction method based on Word2Vec
2) Sub-classifier generation module
3) ENIDrift update module

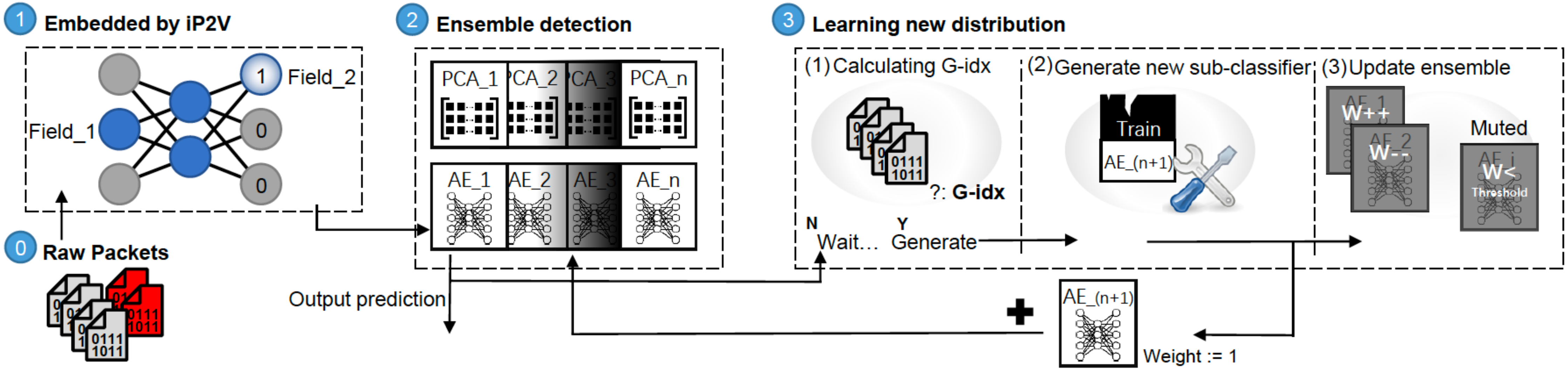


Figure: An overview of ENIDrift

The ENIDrift NIDS - iP2V

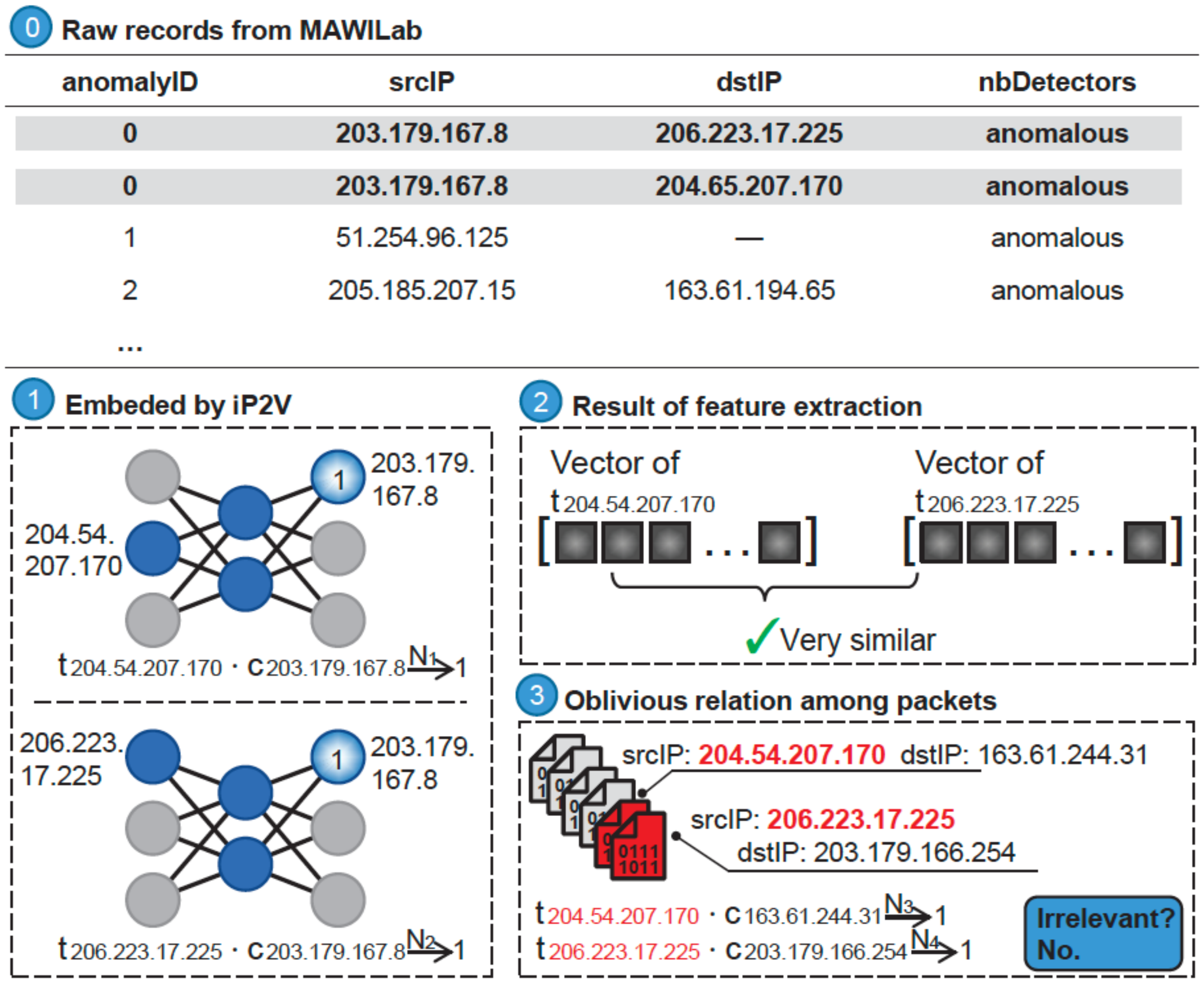


Figure: An overview of iP2V

We have an incremental Packet to Vector tool, iP2V.

- Its extraction is based on relationships among network packets and has good performance.
- It only has simple operations and is very fast.
- We decouple the computation of its ANN and make it incremental.

The ENIDrift NIDS - G-idx based Sub-classifier Generation Module

Previous approaches:

- Using <u>variance</u> to control the sub-classification generation
- Using <u>accuracy</u> to control the sub-classification generation.

We consider both variance and accuracy in our generation index (G-idx) so that the generation can be more stable and defend the well-crafted ML attack.

$$r = \lambda Var(\boldsymbol{o}) + (1 - \lambda) Err(\boldsymbol{o}),$$

We also re-construct the generation workflow so that the time complexity of the generation is reduced from O(n) to O(1).

The ENIDrift NIDS - Ensemble Update Module

The update module is made to maintain and strengthen the adaptability of ENIDrift.

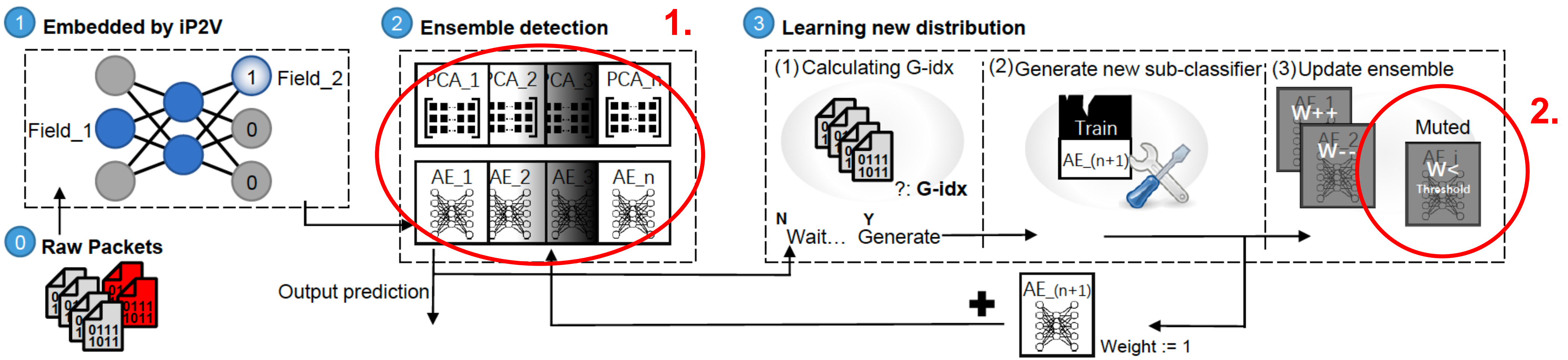


Figure: An overview of ENIDrift

We make three main adjustments based on its original framework:

1. Because there are extremely imbalanced network packets, the adaptive ensemble framework is changed from a supervised to an unsupervised model;
2. There are upper and lower bounds for the sub-classifier weights. Sub-classifiers with weights lower than a pre-defined threshold will be muted for some time for ENIDrift ensemble classification;
3. There is also an upper bound for the dataset size of the sub-classifier generation. It can exclude bad datasets, e.g., data contamination.

Implementation and Evaluation

Test models:

- Original Kitsune
- Retainable Kitsune
- ENIDrift with PCA
- ENIDrift with AutoEncoder

Datasets:

- CICIDS2017
- MAWILab
- RWDIDS

The experiment has four levels.

- ➢ Level 1: test by network packets collected over a long period of a day, where real world drift is light.
- ➢ Level 2: test by network packets from different days and attacks, where real world drift is heavy.
- ➢ Level 3: test by real world drift from the three specific real world drifts

  (using our own real world drift dataset for network intrusion detection)

Evaluation Result

- ENIDrift significantly outperforms the state-of-the-art solutions by up to 69.78% of F1 and
- ENIDrift achieves a 100% F1 against our adversarial attack and is adaptive to various real-world drifts.

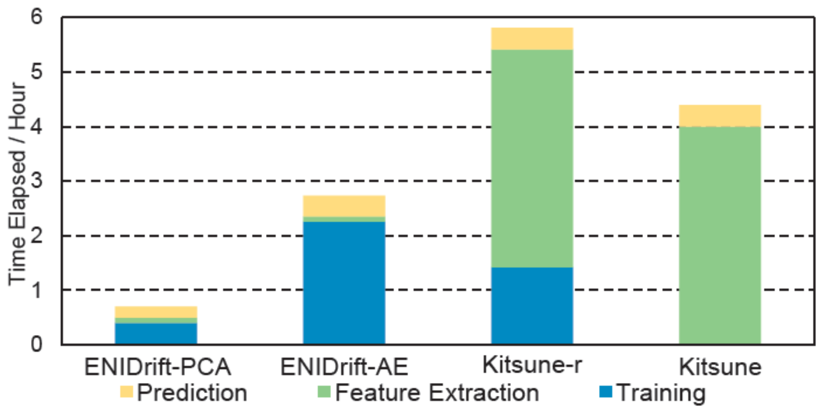


Figure: Breakdown of execution time in level-1

- reduces running time by 87.6%.

We also show the performance of the three components in detail.

Evaluation Result

We also evaluate the dependency on the releasing speed of datasets in level 4.

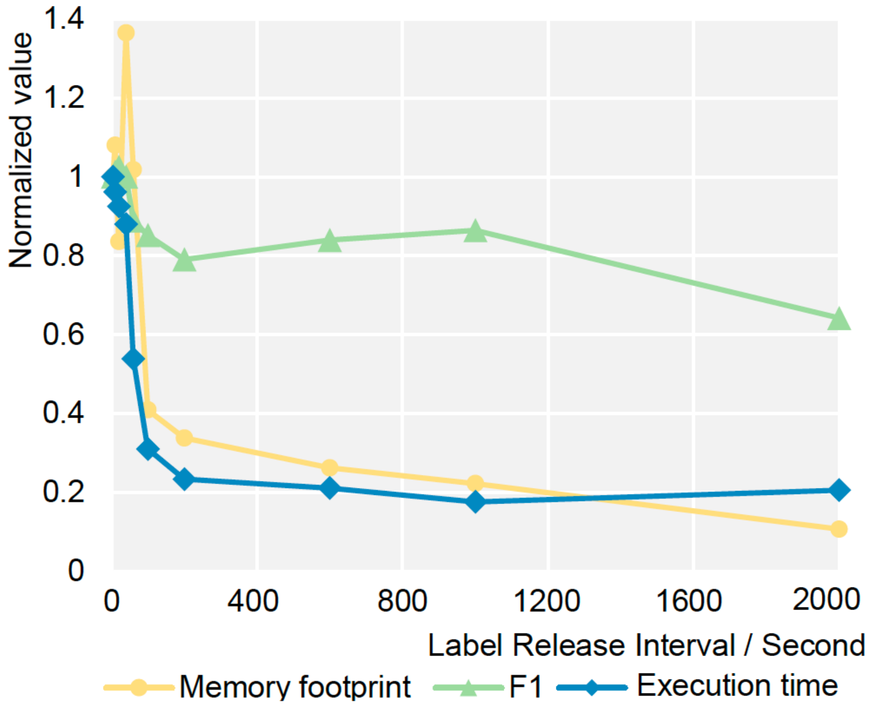


Figure: The memory footprint, F1-score and execution time (after normalization) of ENIDrift-PCA with respect to different LRI in the field test

ENIDrift can maintain 80% performance with a latency smaller than 1200s.

Our model ENIDrift does not require a real-time training network packet release and has a tolerance for inadequate annotated training data.

Conclusion

- We develop a new NIDS, ENIDrift with several new techniques:

  1) iP2V

  2) G-idx based sub-classifier generation modules

  3) ENIDrift update module

- New dataset with real-world drift

  We spent considerable effort collecting and constructing the first dataset **considering real-world settings and fierce drift** caused by concept drift, imbalanced data and well-crafted ML attack.

- Readily deployable performance

  Our evaluation demonstrates that ENIDrift has good performance on both accuracy and processing speed, and is sufficient for real-world deployment even under inadequate and delayed training data.

Thank you!