FRANCE 2030 PROGRAMME DE RECHERCHE CYBERSÉCURITÉ

Evaluating and Enhancing Test Set Quality for ML- and DL-based IDS Testing

Omar Anser*, Jérôme François*†, Isabelle Chrisment*

IETF 121 NMRG

* Inria, Université de Lorraine, CNRS, LORIA, Nancy, France
† SnT - University of Luxembourg, Luxembourg

Emails: firstname.lastname@inria.fr, firstname.lastname@uni.lu

Introduction to ML- and DL-Based IDS Evaluation

ML & DL-Based IDS

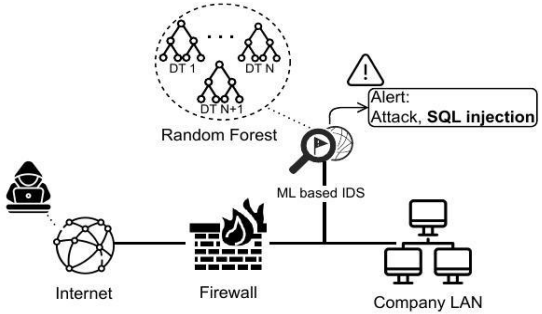
- IDS using **Machine Learning (ML)** and **Deep Learning (DL)** for **traffic classification**
- Classification models trained to distinguish between normal and malicious network traffic

Classification Algorithms

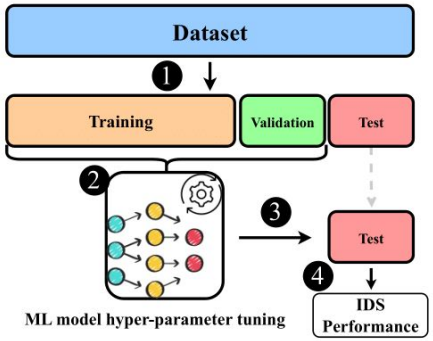
- **ML Algorithms**: Random Forest (RF), Support Vector Machine (SVM), Decision Trees (DT)
- **DL Models**: Deep Neural Networks (DNN), Convolutional Neural Networks (CNN) for complex feature extraction

Model Training Workflow

- **Dataset Splits**: Train / Validation / Test sets
- **Hyperparameter Tuning**: Optimizing model parameters for best performance
- **Issues in Training**
  - **Overfitting**: Model too specific to training data, poor generalization



ML-based IDS



Model Training Workflow

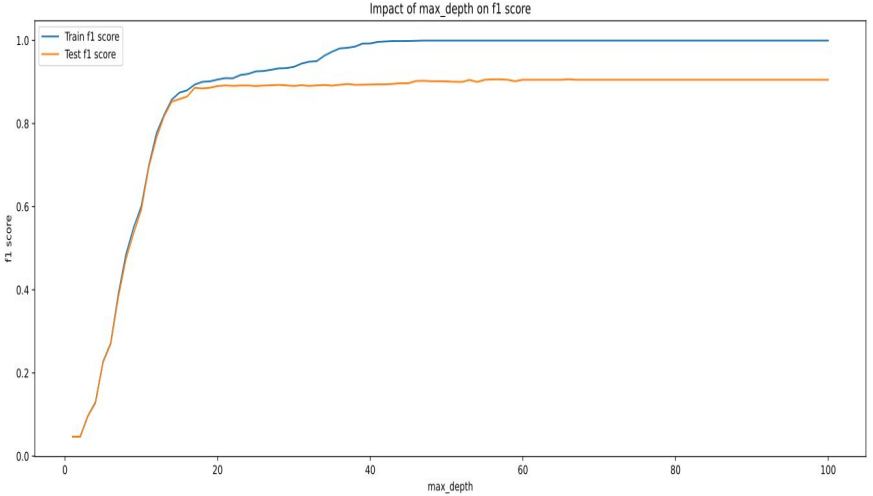
Problem Statement

Datasets Used in Prior Work

- Experiments conducted on **IDS2017** and **IDS2018** datasets for Intrusion Detection Systems

Prior Work

- Focused on **hyperparameter tuning** and **automation** for IDS
- Implemented **RF** classifier using **scikit-learn**
- Initially used **default hyperparameters**:
  - `max_depth=None`: unrestricted growth
  - `min_samples_split=2`: complex trees from splitting nodes with minimal samples
  - `min_samples_leaf=1`: high variance with leaves holding single samples
  - **Expected outcome:** Overfitting, as model captures noise over patterns



Impact of Tree Depth on IDS Performance During Training and Testing

Unexpected Result

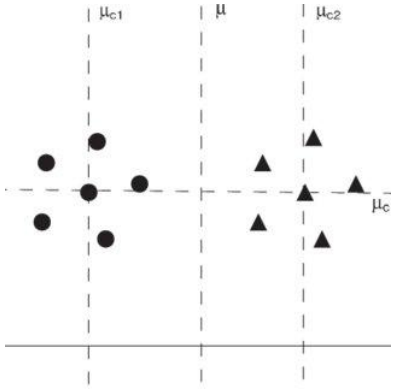
- Despite overfitting, the model achieved **near 100% detection** on test data
- Raises a question: **Is the dataset potentially the problem?**

Evaluating Dataset Complexity

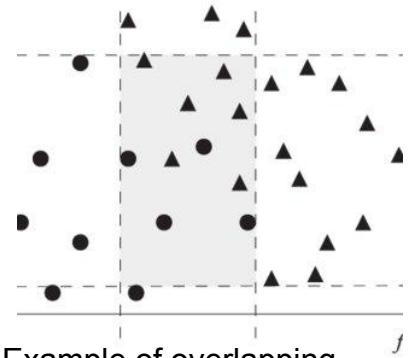
- Is the dataset too simple with a lack of variability?
  - Ana C. Lorena et al. 2019. How Complex Is Your Classification Problem? A Survey on Measuring Classification Complexity
  - Common Measures: Feature-based and Linearity-based,
    ranging from 0 (low complexity) to 1 (high complexity)

- Feature-Based Measures:
  - Quantify data complexity based on feature interactions
    and discriminative power between classes (e.g., how easily features separate normal vs. malicious traffic)

- Results
  - IDS scores near 0; CIFAR-10 scores higher close to 1
  - Indicates IDS data may lack variability or complexity



Example of F1 computation for a two-class dataset.



Example of overlapping region

Measures	IDS	CIFAR-10
F1	0.0177	0.8029
F2	0.0000	0.8524
F3	0.0000	0.9998
F4	0.0078	0.2660

Features-based Measures IDS vs

Research Questions

- Evaluating IDS Performance
  - **Are we effectively assessing model performance using the IDS dataset?**

- Quantifying Test Set Quality
  - **How can we measure the quality of a test set for robust model evaluation?**

- Enhancing the Test Set
  - **What strategies can help improve the test set to better capture model performance?**

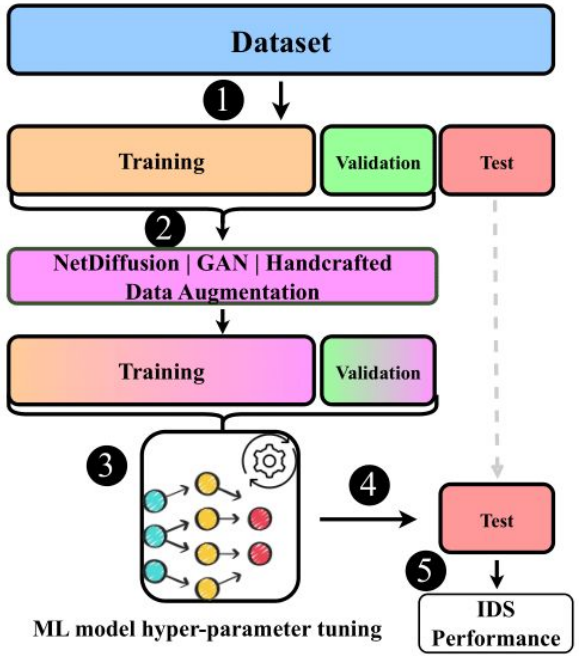
Data Augmentation: State of the Art (SoA)

- **Recent Methods in Traffic Classification**
  - **Chao Wang et al., 2024**: Data Augmentation for Traffic Classification
  - **Xi Jiang et al., 2024**: NetDiffusion – Protocol-Constrained Traffic Generation
  - **Yucheng Yin et al., 2022**: GAN-based Synthetic IP Header Trace Generation

- **These data augmentation methods solve a different problem and do not address our specific issue:**

→ Improving test accuracy with training set data augmentation, without prioritizing consistency

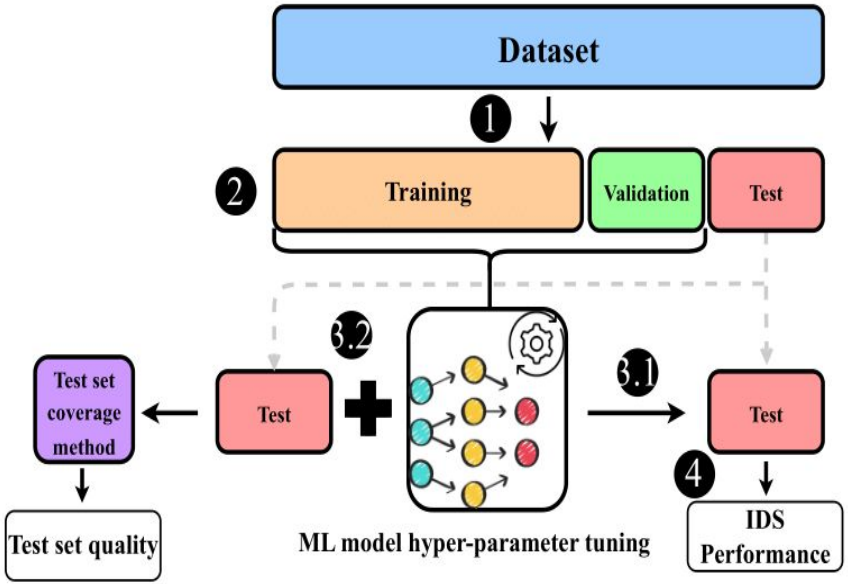
→ Our issue is not accuracy, but ensuring the relevance of model testing



Data Augmentation Application

Test set coverage SoA: Measuring the diversity

- Neuron coverage: Focuses on the internal structure, ensuring a broad range of neurons are activated
- Surprise adequacy: Focuses on the novelty of the inputs compared to the training data
    - Jinhan Kim et al. 2019. Guiding deep learning system testing using surprise adequacy. In Proceedings of the 41st International Conference on Software Engineering (ICSE '19).
- Mutation testing: Evaluates test set quality by introducing small changes to the model and checking if the tests can detect them.
    - Nargiz Humbatova et al. 2021. DeepCrime: Mutation Testing of Deep Learning Systems Based on Real Faults.



Test set coverage application

Test set augmentation: DeepMetis

- DeepMetis uses evolutionary algorithms to generate new test inputs that aim to detect faults introduced by mutations.
- DeepMetis uses DeepCrime to introduce mutations into the model and identifies inputs that fail to detect these changes, enhancing the test set's capability.
- The tool measures how many mutants (faults) are detected by the test set, aiming to increase the mutation score—an indicator of the test set's effectiveness.
- Based on multi-objective optimization (NSGA-II), DeepMetis iteratively improves test inputs by evolving a population of candidate inputs over multiple generations to better detect faults.

The Shortcomings of Existing Solutions

- Limitations of Neuron Coverage Solutions
  - Model-Dependent: They are closely tied to specific models
  - Interpretability Issues: Very difficult to interpret the results
- Limitations of Mutation Testing
  - Operators for creating mutants are model-dependent
  - Changing the model requires changing the mutation operators
- Limitations of Surprise Adequacy
  - Focuses only on the proximity of data points to the training set
  - The final trained model must be provided to calculate the metrics
- Gaps in DeepMetis
  - Utilizes mutation testing and inherits the same limitations
  - Employs a data generator that may produce unrealistic data (impossible to verify in our domain)
  - Solution is not reusable across different datasets

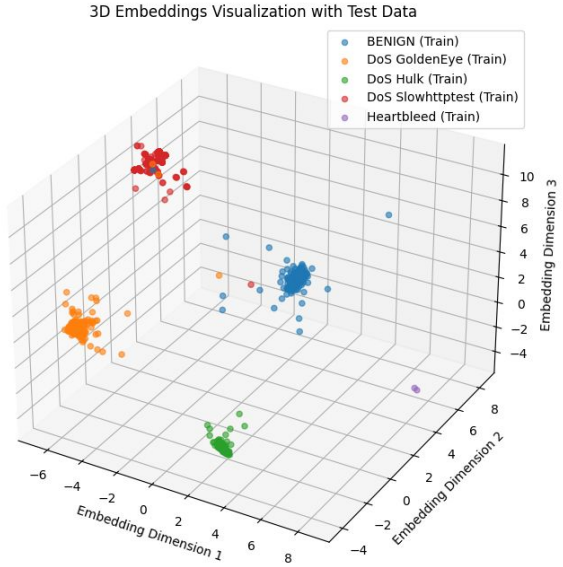
Addressing the Gaps in Existing Solutions

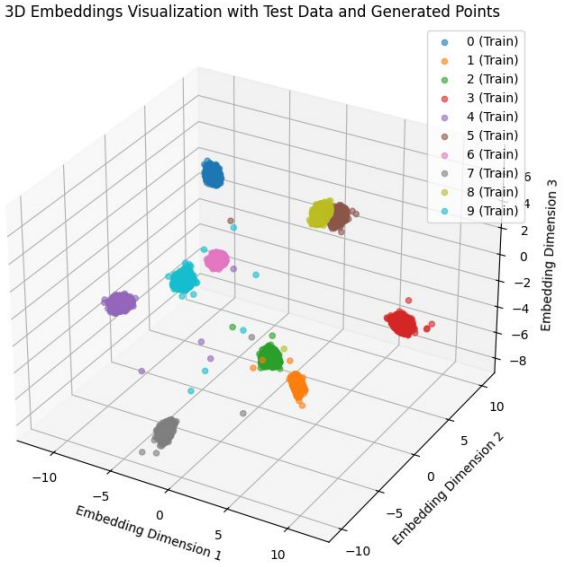
- Decouple from the model by focusing solely on data: apply contrastive learning to form clusters by training an Autoencoder (AE) on training data and then passing test data through it

- Propose a New Metric for Test Data Quality by Integrating Proximity, Scarcity, and Diversity to Provide a More Comprehensive Assessment

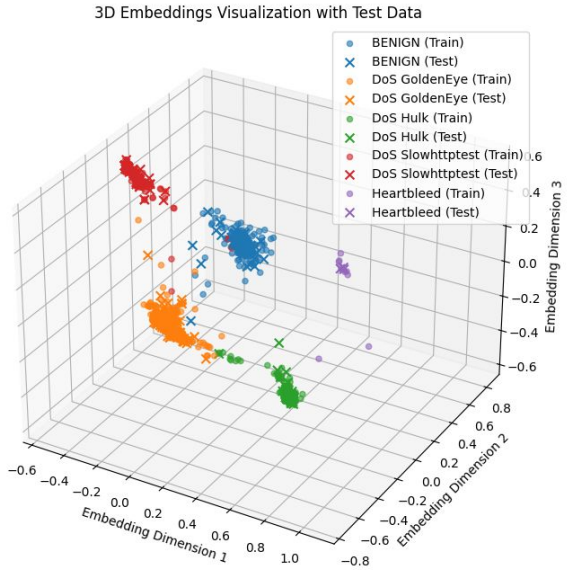
- Guide the Generation of Test Data Points Using This Metric
  - Utilize Reinforcement Learning (RL) to find tc configurations that increase the metric's value
  - The trained agent can be applied to different datasets

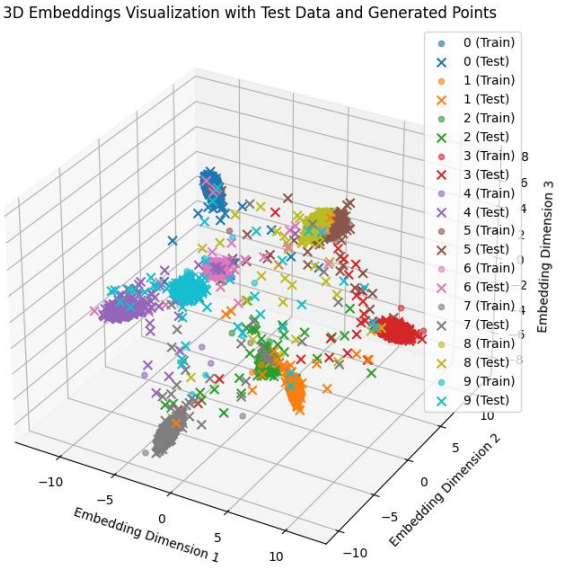
Decoupling from the Model: Focusing Solely on Data with Contrastive Learning





Decoupling from the Model: Focusing Solely on Data with Contrastive Learning





Introducing a Comprehensive Metric: Proximity, Scarcity, and Diversity in Test Data Quality

Proximity Score

- Measures how close test data points are to the target clusters.

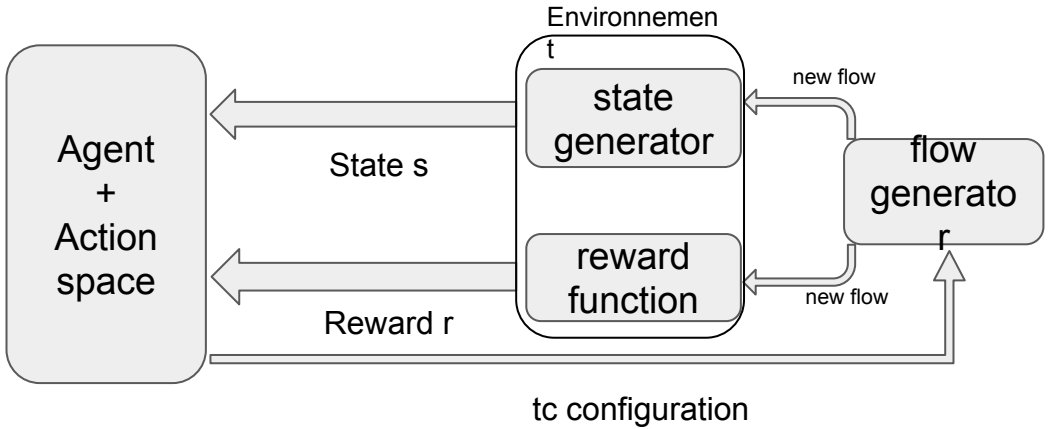
Scarcity Score

- Assesses the distribution balance of test data across clusters using the Gini Coefficient.

Diversity Score

- Evaluates the variety within each cluster of test data using the Vendi Score.

Guiding Test Data Generation: Reinforcement Learning for Enhanced Test Sets



Guiding Test Data Generation: Reinforcement Learning for Enhanced Test Sets

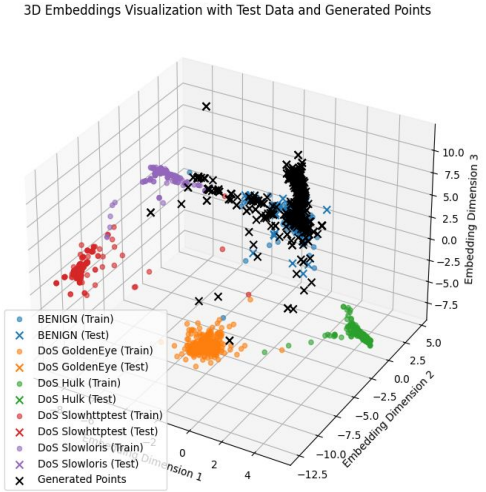
Action Space: Configurations of the **tc** parameters.

Reward Function: Improvement in the coverage score (ΔCoverage Score).

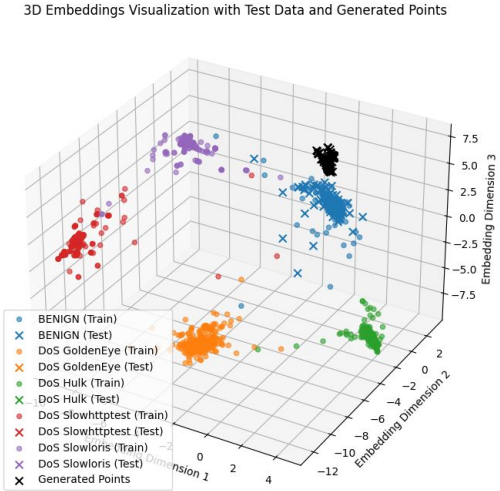
Workflow:

- **Initialize** an empty set of generated flows at the beginning of each episode.
- **At each step of the episode**:
  - **Generate a flow** using the current action.
  - **Add** the generated flow to the set of flows.
  - **Calculate** the marginal improvement in the coverage score and use it as the immediate reward.

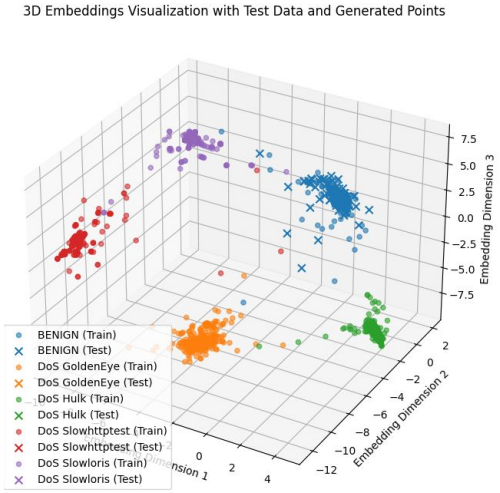
Preliminary Results



With tc



Without tc



Test original

Conclusion

- **Lack of Diversity:** IDS datasets lack the robustness and diversity seen in fields like computer vision.
- **Need for Complexity:** More complex, realistic network traffic datasets are needed to reflect real-world variability.
- **Dataset Quality:** High-quality datasets are crucial for developing IDS models that generalize effectively.
- **Enhanced Benchmarking:** Generate and share advanced datasets to raise IDS benchmarks to the standards of other fields.
- **Long-Term Vision:** Aim for IDS research driven by challenging, comprehensive datasets, enabling resilient and accurate models.

- **Acknowledgment:** This work has been partially supported by the French National Research Agency under the France 2030 label (Superviz ANR-22-PECY-0008). The views reflected herein do not necessarily reflect the opinion of the French government.

Questions?

Conclusion

- **Test Set Quality Quantification**
  - Our method enables **quantitative assessment of test set quality**

- **Improved Test Set Generation**
  - Provides a framework for **enhancing test sets** to better challenge IDS models

- **Assessing Model Robustness**
  - Ensures IDS models are **genuinely robust**

- **Acknowledgment:** This work has been partially supported by the French National Research Agency under the France 2030 label (Superviz ANR-22-PECY-0008). The views reflected herein do not necessarily reflect the opinion of the French government.