Detection of Network Intrusions and Attacks

1st Yaroslav Hayduk
Faculty of Information and Communication Technology
Wrocław University of Science and Technology
Wrocław, Poland
293883@student.pwr.edu.pl

2nd María Capilla
Faculty of Information and Communication Technology
Wrocław University of Science and Technology
Wrocław, Poland
maria.capzap@gmail.com

3rd Lukas Marović
Faculty of Information and Communication Technology
Wrocław University of Science and Technology
Wrocław, Poland
293855@student.pwr.edu.pl

I. Introduction

Modern society heavily relies on the proper functionality of communication networks, which form the backbone of information systems and support critical infrastructure, industrial control, and everyday services [1]. Ensuring their security is therefore a key requirement for maintaining continuous and reliable operation. Network infrastructures are regularly targeted and exposed to malicious activities such as jamming, signal manipulation, denial-of-service, or unauthorized access. These intrusions can cause severe disruptions in data delivery and integrity across different layers of the network stack.

Traditional signature-based or rule-based intrusion detection systems (IDS) are limited in their ability to recognize previously unseen types of attacks [2], [3]. As network technologies and threat patterns continually evolve, new types of attacks frequently appear. Consequently, these static mechanisms often fail to provide early warnings or adapt to emerging behaviors. To address these challenges, machine learning (ML) techniques have become widely used for network security monitoring, as they allow models to automatically learn from telemetry data and detect subtle deviations from normal network behavior [2], [4].

Even though ML-based intrusion detection methods can provide high accuracy for recurring attacks, they are constrained by the dynamic nature of real networks. Attack patterns and background traffic continuously change, which may lead to model drift and loss of prediction quality over time [5]. Additionally, many ML models require large, labeled datasets and extensive retraining to maintain performance. This motivates the exploration of adaptive and explainable ML approaches that can continuously update themselves, generalize across different attack types, and remain reliable in real-time operation [5]–[7].

The goal of this project is to study and evaluate ML-based methods for the detection of network intrusions and attacks. Using network telemetry datasets that include multiple types of intrusion events, the task consists of training and analyzing binary classifiers that distinguish between attack and no-attack conditions across various scenarios. The obtained results will help assess how different learning strategies maintain performance when exposed to evolving or previously unseen attack types.

II. Literature Review

Machine learning (ML) has become a fundamental component of modern network intrusion detection systems (IDS). Due to the increasing scale and sophistication of cyberattacks, traditional rule-based security mechanisms often fail to recognize novel or evolving threats [2], [3]. Recent research by Knapińska and Furdek and their collaborators explores how ML can be applied to detect attacks across different layers of communication systems, with a particular focus on adaptability, explainability, and real-time performance [5]–[7].

A. Dynamic ML Model Updating Strategies for Detection of Evolving Network Attacks

In [5], Knapińska and Furdek present two dynamic machine learning updating strategies designed to enhance the adaptability of attack detection systems. The first strategy involves incremental updates to a single classifier, while the second utilizes an ensemble method based on streaming algorithms. These approaches are validated on multiple public datasets, demonstrating that dynamic model updating significantly improves sustained detection accuracy in the face of evolving attack scenarios compared to static models. The work highlights the necessity of flexible models that can generalize across attack types and retain robust performance over time.

B. Adaptive ML for Optical Network Attack Detection

In [6], the authors experimentally evaluate the performance of adaptive ML classifiers, including Multilayer Perceptron (MLP), Support Vector Machine (SVM), and eXtreme Gradient Boosting (XGB), to detect physical-layer attacks in optical networks. Telemetry data from a real-world testbed is used to train models in six representative attack categories. The results show that joint models trained on aggregated data can efficiently identify both known and new attacks, with MLP particularly excelling in rapid adaptation through partial fitting. Achieving up to 0.936 balanced accuracy, the research underscores the critical role of adaptive ML in maintaining network security against emerging threats.

C. Explainable and Resilient ML-Based Physical-Layer Attack Detectors

The work in [7] focuses on the explainability and resilience of ML-based attack detectors for network infrastructure. Using SHAP-based feature importance analysis, the authors enhance the interpretability of models trained on physical-layer intrusion data. They demonstrate that optimally selected feature subsets can drastically speed up detection time but warn that such streamlined models may be more susceptible to adversarial noise. The findings emphasize the importance of balancing computational efficiency and robustness against attacks, pointing to explainable AI as a crucial factor for trustworthy and effective security systems.

D. Unsupervised Network Intrusion Detection

Casas, Mazel, and Owezarski [8] discuss the limitations of IDSs that were dominant in security devices at the time. They highlight how signature-based IDS systems, which rely on known attack signatures, cannot detect new types of network attacks. They then examine anomaly-based IDS systems, which, while capable of detecting new attacks, require time-consuming training to build profiles that minimize false positives. To address these issues, the authors propose UNIDS, an unsupervised Network Intrusion Detection System designed to detect unknown attacks without labeled data, signatures, or prior training. The method combines subspace clustering, density-based clustering, and evidence accumulation clustering to identify and rank outliers in network traffic flows. When tested on the KDD'99 dataset, UNIDS successfully detected over 90% of attacks, performing similarly to a signature-based NIDS on known attacks and significantly outperforming it on unknown attacks.

III. Problem Definition

The application of machine learning (ML) to network attack detection offers significant promise, yet still confronts several critical challenges that limit the effectiveness and practicality of existing solutions. Traditional approaches that depend on static ML models, each trained for a specific known attack, suffer from both insufficient flexibility and limited scalability: it is infeasible to maintain a separate model for every possible threat in a constantly evolving attack landscape [2]. As a result, these static systems lack comprehensive coverage and quickly experience performance degradation when new or modified attack types emerge, leading to a failure in generalization and necessitating costly manual retraining from scratch each time the threat landscape changes [5].

To address the inability of static models to generalize toward unseen attacks, the field has explored unsupervised and anomaly-based learning techniques, which have demonstrated improved detection of novel threats [3], [4], [8]. Additionally, recent research has focused on incorporating data from lower layers of the network, such as the physical layer, to further enhance detection capabilities [6], [9]. However, both of these approaches are often hampered by the absence of efficient dynamic update mechanisms and either lose accuracy over time or require frequent, resource-intensive retraining, rendering them suboptimal for real-world deployment.

More recent studies introduce dynamically updating models that learn incrementally from new data, showing promising improvements in long-term performance [5]. Still, these dynamic approaches typically incur considerable computational cost because all incoming traffic is processed and used for updates. To address this open challenge, the present paper proposes a hybrid solution that combines the high efficiency and low overhead of static models for initial traffic screening with the adaptability of dynamically updating models, triggered only upon detection of suspicious behavior. This design aims to maintain effective detection of previously unseen attacks while substantially reducing the computational resources traditionally required for continuous dynamic updates.

IV. Proposed Methods

A. Overview

To address the trade-off between adaptability and computational cost in ML-based network attack detection, we propose a hybrid two-stage classification framework that combines a lightweight static anomaly detector with a dynamically updated supervised classifier. The intuition is to use an inexpensive model to continuously monitor all traffic and flag only suspicious flows, while reserving the more computationally intensive dynamic model for a smaller subset of potentially anomalous samples. In real-world scenarios most traffic is benign, so such filtering can substantially reduce the load on the adaptive component. In this way, the system aims to preserve the ability to detect evolving or previously unseen attacks, as in dynamic ML schemes [5], while reducing the overhead associated with frequent updates on the entire traffic stream.

Our approach is conceptually related to hybrid IDS architectures that chain anomaly and misuse detection components [8], [10], but it differs in several aspects. First, the first-stage detector operates directly on network telemetry features (KDD'99 and NetFlow-based datasets), focusing only on distinguishing normal from suspicious traffic rather than specific attack types. Second, the second-stage detector is not a static misuse classifier but a dynamically updated ML model, following the updating strategies proposed in [5]. This combination is intended to be both adaptable to new threats and more efficient in resource usage than purely dynamic systems.

B. Stage I: Static Anomaly Screening

The first stage of the proposed framework is a static anomaly detection model trained to characterize normal network behavior. Only benign samples are used during training, leveraging the “normal” class in KDD’99 and benign flows in the NetFlow dataset. This model is trained once offline and is not updated during operation.

In the implementation phase, we plan to experiment with classical anomaly detection techniques and density-based approaches inspired by UNIDS [8]. Features will be preprocessed using standard scaling and normalization. The decision threshold will be tuned to keep the false positive rate low, as excessive false alarms would negate its purpose as an efficient filter.

During operation, every incoming flow is first processed by the static anomaly detector. If it is classified as normal with high confidence, the flow is immediately labeled as benign and not forwarded to the second stage. Only flows flagged as suspicious are passed to the dynamic model for further analysis. This stage therefore serves as a cheap, always-on screening layer that reduces the volume of traffic that the more expensive dynamic model must handle.

C. Stage II: Dynamically Updated Binary Classifier

The second stage of the framework is a supervised binary classifier that distinguishes between attack and no-attack conditions on the subset of flows flagged as suspicious by the first stage. This model follows the dynamic updating strategies studied by Knapińska and Furdek [5]:

- **Dynamic single model:** a single classifier (e.g., MLP or SGD) updated incrementally using mini-batches of newly observed data;
- **Dynamic ensemble:** an ensemble of base learners created on consecutive batches of data, with older models gradually replaced or down-weighted.

In our framework, the key difference is that the dynamic model is trained and updated only on flows that have been pre-filtered as anomalous by Stage I. During an initial warm-up phase, the dynamic model is trained on labeled attack and benign samples from the offline training datasets, but only using instances that would be flagged as suspicious by the static detector. This simulates the real operational pipeline where the second stage never sees clearly benign traffic.

During live operation, the system continuously processes incoming flows in batches. For each batch:

1) Stage I labels each flow as normal or anomalous.
2) All anomalous flows are passed to Stage II, which outputs a binary decision (attack / no attack).
3) Once the ground-truth labels for these flows are available (e.g., in an emulated environment), Stage II is updated incrementally using only this subset, according to the chosen dynamic strategy.

The hypothesis is that by focusing the dynamic updates only on suspicious traffic, a smaller but more informative subset of data will (i) reduce training and inference time per unit of traffic, and (ii) maintain or even improve long-term detection performance compared to dynamic models trained on all flows, especially when attacks are relatively rare.

D. Baseline Methods and Evaluation Plan

To assess the effectiveness and benefits of the proposed hybrid framework, we will compare it against the following baselines:

- **Static baseline:** a conventional static supervised classifier trained once on labeled data and then applied without further updates, as commonly done in ML-based IDS studies [2], [4].
- **Dynamic-only baseline:** the dynamic single and dynamic ensemble models from [5] operating directly on the full traffic stream, without the static pre-filter.

Performance will be evaluated on standard network intrusion detection datasets (KDD’99 and NetFlow-based datasets) using metrics such as accuracy, precision, recall, F1-score, and computational cost (inference time and update time). Particular attention will be paid to the behavior of each method under evolving attack scenarios and to its ability to detect previously unseen attack types.

By quantifying both detection performance and resource consumption, we aim to determine whether the proposed hybrid approach provides a practically meaningful improvement over existing purely static or purely dynamic ML-based intrusion detection systems.

V. Solution Implementation

A. Stage I: The Filter

For the initial development of our solution, we focused on the first stage: training an unsupervised Isolation Forest model to detect even the slightest anomalies and forward them to the dynamic model. We selected Isolation Forest because this algorithm excels at outlier detection, and trained it using 80% of the RoEduNet-SIMARGL2021 dataset [11], after removing all anomalous traffic (to avoid biasing the model toward known anomalies) and irrelevant features that introduced noise during training. The model was configured with 100 estimators and a contamination parameter tuned up to 0.5 to favour high recall on anomalous traffic. We then tested it on a dedicated test dataset with the anomaly distribution shown in Figure 1.

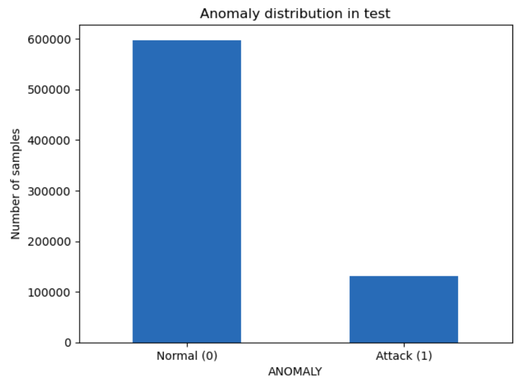

Fig. 1. Anomaly distribution in test dataset.

For testing our model, we tried multiple values of contamination, and we obtained the results displayed in Figure 2.

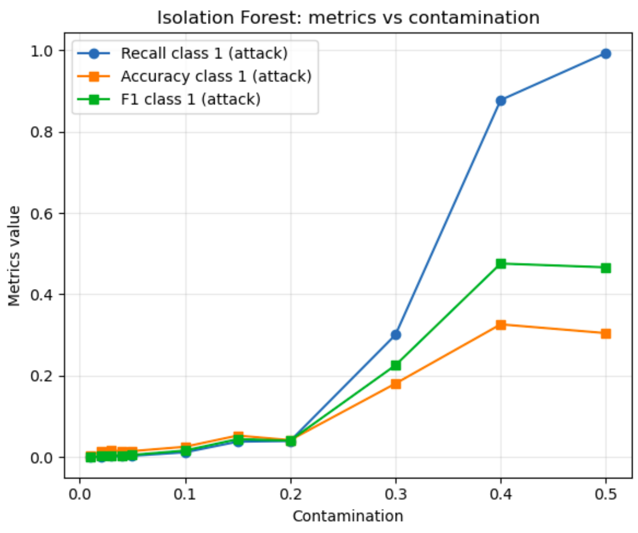

Fig. 2. Isolation Forest: metrics vs contamination (class 1).

In this context, contamination = 0.5 is the most appropriate value because it maximizes the model's ability to detect attacks, which is the primary goal in an intrusion detection system. At 0.5, the recall for the attack class (class 1) reaches its highest value, meaning the model misses far fewer attacks than with lower contamination values, and the F1-score for attacks is also maximized, indicating the best overall balance between precision and recall for that class. Although precision for attacks is slightly lower than at 0.4, the significant gain in recall and F1-score is more important, since in this domain failing to detect attacks is more critical than generating some additional false positives.

Although the chosen configuration generates a relatively high number of false positives, this behaviour is acceptable – and even desirable – in our architecture. The anomalies flagged by the Isolation Forest in Stage 1 (including false positives) will be used as additional labelled data for the Stage-2 model, which is designed to learn dynamically and refine the decision boundaries between normal and attack traffic. Therefore, Stage 1 acts as a high-recall, coarse filter that feeds a more powerful supervised classifier, rather than as a final detector on its own.

Also, this model still filters out around 50% of benign traffic, which significantly reduces the computational cost of the second-stage model, since that stage only needs to process the remaining, more suspicious flows shown in Figure 3.

B. Stage II: The Expert

The second stage of the solution, which should discern false positives from real attacks, supports multiple strategies, allowing comparison between different levels of complexity and adaptability. The strategies that we have chosen for experimentation are two dynamic single models and a dynamic ensemble model:

1) SGD (Stohastic Gradient Descent) – a linear dynamic single classifier for fast, linear updates.

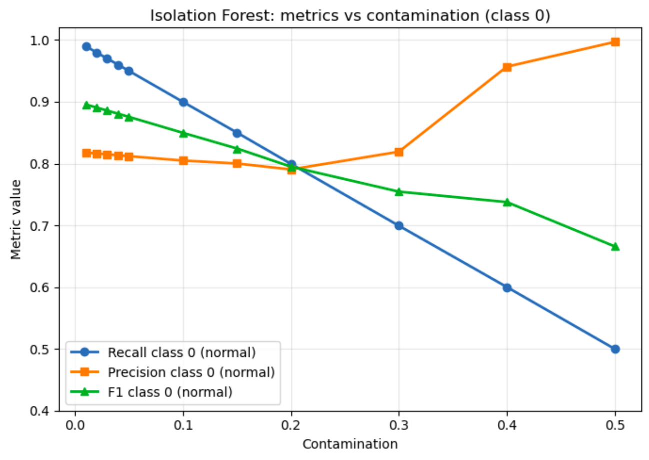

Fig. 3. Isolation Forest: metrics vs contamination (class 0).

2) MLP (Multi-layer Perceptron) – a neural network-based dynamic single classifier for capturing non-linear patterns.
3) Dynamic ensemble – a strategy using an SEA-inspired pool of classifiers that trains a new model for each batch and prunes the worst performer, offering high stability. Prediction is done using a majority vote.

The pipeline for a chosen strategy consists of two phases: a warm-up phase and an online testing phase. The warm-up phase consists of initializing our dynamic models using training data that the Isolation Forest from the first stage flagged as anomalous. The models need to be trained with some initial data, or else they will not be able to comprehend the inputs during the second phase. For the dynamic ensemble strategy this means creating an initial pool of classifiers by splitting the training data up into chunks for each new classifier to be trained on.

In the online testing (simulation) phase, the test data is split into batches which are processed one by one. The batch data is first processed by the first stage and is then sent to the dynamic model for prediction. The dynamic model is then updated using the test data, and both the predicted labels and the true labels are stored for further evaluation. The dynamic single models are updated using partial fitting, while the dynamic ensemble is updated by replacing the worst performing model in the pool with a new one trained on the batch data.

VI. Results

Metric	SGD	MLP	Dynamic Ensemble
Balanced Accuracy	85.71%	90.04%	95.94%
Recall (Attacks)	76.84%	85.41%	97.57%
F1-Score (Attacks)	0.7631	0.8150	0.8738
Processing Time	3.66s	6.29s	38.58s

TABLE I
Comparison of model performance metrics.

The most important metric that should be observed is the recall of attacks. Because an ideal IDS should be able to deal with all threats, we want to ensure that our model detects an attack when it happens, even if a certain degree of false alarms is present. The recall of attacks gives us a score of how many attacks the model correctly detected, which is exactly what we need. As we can see in Table 1, the ensemble has by far the highest score, having caught 97.6% of attacks, with the MLP only catching 85.41% of them, and the SGD even less with a 76.84% detection rate. The ensemble performs the best even across other metrics such as balanced accuracy, showing that it can even correctly detect whether a piece of traffic is benign or malignant most of the time.

The metric where the ensemble predictably lags behind the other models is processing time. The ensemble is computationally heavy and at 38.58 seconds of processing time, it is about six times slower than the MLP and ten times slower than the SGD. The MLP therefore provides a decent balance of speed and accuracy, if the system does not require a high degree of security. However, our stage 1 filter successfully marked 45.25% of traffic as benign with high confidence. This means the expensive ensemble only had to process 54.75% of the data, which drastically decreased the workload, making it a feasible option for real-time detection.

References

[1] C. Alcaraz and S. Zeadally, "Critical infrastructure protection: Requirements and challenges for the 21st century," *International Journal of Critical Infrastructure Protection*, vol. 8, no. 4, pp. 53–66, 2015.

[2] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 20, pp. 1–22, 2019.

[3] R. Samrin and D. Vasumathi, "Review on anomaly based network intrusion detection system," in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2017, pp. 141–147.

[4] T. Zoppi, G. Fenza, D. Furno, V. Loia, F. Orciuoli, and S. Senatore, "Which algorithm can detect unknown attacks? comparison of supervised, unsupervised and meta-learning algorithms for intrusion detection," *Computers & Security*, vol. 127, p. 103107, 2023.

[5] A. Knapińska and M. Furdek, "Dynamic ml model updating strategies for detection of evolving network attacks," in *2025 15th International Workshop on Resilient Networks Design and Modeling (RNDM)*, 2025, pp. 1–7.

[6] ——, "Experimental analysis of adaptive ml classifiers for dynamic detection of emerging physical-layer attacks," *Conference Proceedings or Journal Name*, pp. 1–4, 2025.

[7] ——, "Explainable and resilient ml-based physical-layer attack detectors," 2025. [Online]. Available: https://arxiv.org/abs/2509.26530

[8] P. Casas, J. Mazel, and P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," *Computer Communications*, vol. 35, no. 7, pp. 772–783, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366412000266

[9] C. Natalino *et al.*, "Microservice-based unsupervised anomaly detection loop for optical networks," in *OFC Conference*, 2022.

[10] S. T. Powers and J. He, "A hybrid artificial immune system and self organising map for network intrusion detection," *Information Sciences*, vol. 178, no. 15, pp. 3024–3042, 2008.

[11] M.-E. Mihailescu, D. Mihai, M. Carabas, M. Komisarek, M. Pawlicki, W. Holubowicz, and R. Kozik, "The proposition and evaluation of the roedunet-simargl2021 network intrusion detection dataset," *Sensors*, vol. 21, no. 13, p. 4319, 2021.