



CHALMERS
UNIVERSITY OF TECHNOLOGY

Enhancing Operational Security of Human-to-Machine Applications through Concept Drift Detection

Downloaded from: <https://research.chalmers.se>, 2026-04-14 13:17 UTC

Citation for the original published paper (version of record):

Yu, X., Natalino Da Silva, C., Monti, P. et al (2025). Enhancing Operational Security of Human-to-Machine Applications through Concept Drift Detection. Conference on Optical Fiber Communication, Technical Digest Series. <http://dx.doi.org/10.1364/OFC.2025.W3J.5>

N.B. When citing this work, cite the original published paper.

be actioned (ML model retraining in the presence of *H2M application drift* or packet filtering in the presence of *malicious drift*). A complementary traffic classifier is also deployed in the ONU to filter out malicious packets when *malicious drift* is detected by the drift detector to optimize bandwidth utilization and improve operational security, thereby guaranteeing the overall performance of legitimate H2M applications.

2. Proposed Concept Drift Detection Framework

The inset of Fig. 1 illustrates our proposed framework. First, the ONU sends a group of packets stored in the buffer, which contains the packet arrival time (T_{arr}) and the payload information received from the end-user, in a round-robin manner to the CO. The *traffic predictor* in the CO, implemented using the long-short term memory (LSTM) and pre-trained using legitimate H2M traffic, is employed to predict the average packet inter-arrival time, termed \bar{T}'_{arr} , for the next polling cycle based on the received incoming packets. The *drift detector* alerts the concept drift by monitoring the prediction degradation of the *traffic predictor*, referred to as the percentage error rate between \bar{T}'_{arr} and the real average packet inter-arrival time (\bar{T}_{arr}). The *drift detector* dynamically adjusts window sizes based on the rate of change between two sub-windows without requiring predefined window sizes. A drift is detected when the difference between the means of two sub-windows ($\theta_{ADWIN} = |\hat{\mu}_{hist} - \hat{\mu}_{new}|$) exceeds a threshold within a predefined confidence value δ . Then, the detected drift would trigger the *traffic classifier* integrated into the *drift detector* to identify whether the concept drift is due to *H2M application drift* or *malicious drift*. Importantly, we use the XGBoost model as the *traffic classifier* due to its efficient inference time $\sim 100 \mu s$ and accuracy $\sim 99.995 \%$ as compared to other non-linear machine learning models, like support vector machine (SVM) and random forest.

Upon detecting any *H2M application drift*, the *drift adaptor* retrains the *traffic predictor* using \bar{T}_{arr} stream from newly incoming packets, adapting to the *H2M application drift*. Conversely, when *malicious drift* is detected, the *traffic classifier* in the ONU is activated to examine every incoming packet based on the payload and classify whether they are legitimate H2M packets or otherwise. Subsequently, legitimate H2M packets are queued for transmission and malicious packets are dropped to mitigate their impacts, thus reducing the uplink latency and enhancing overall operational security of H2M applications.

3. Performance Evaluation and Discussion

Simulations of a 10 km 10-GPON comprising 16 ONUs are conducted to test the effectiveness of the proposed framework compared to a baseline scenario, which is without its implementation. The *traffic classifier* is trained using the control signal of H2M applications, which contains a total of 63 features collected from the VR-based H2M experimental platform [1]. Two distinct H2M applications (H2M 1 and H2M 2) are used as legitimate traffic, with normalized traffic loads of 0.4 and 0.8, respectively. H2M 1 at normalized loads of 0.2 (low) and 0.8 (high) are used as the malicious control signal traffic. All H2M application traffic distributions are modelled as the Generalised Pareto distribution [1]. Three performance metrics, calculated as the average across all 16 ONUs, are considered to evaluate the framework performance: network bandwidth utilization, i.e., the percentage of the utilized bandwidth in Gbps over the total available bandwidth (10Gbps), uplink latency, and operational security enhancement, defined as the percentage of received legitimate H2M packets in the CO.

Fig. 2(a) illustrates the network bandwidth utilization under different traffic scenarios. In the baseline scenario, the injection of malicious traffic at the ONU results in increased bandwidth utilization, reaching up to 100 % when the malicious traffic load is high. Consequently, the excessive utilization of bandwidth by the malicious traffic results in less bandwidth available for transmitting legitimate H2M packets. Conversely, our proposed framework can detect and identify malicious traffic at 0.2 and 0.8 load levels, preventing extra bandwidth usage by malicious traffic and preserving it for legitimate H2M applications, with detection times of 32.09 ms and 13.78 ms, respectively. From Fig. 2(b), when the malicious traffic load is 0.2, the baseline uplink latency increases from

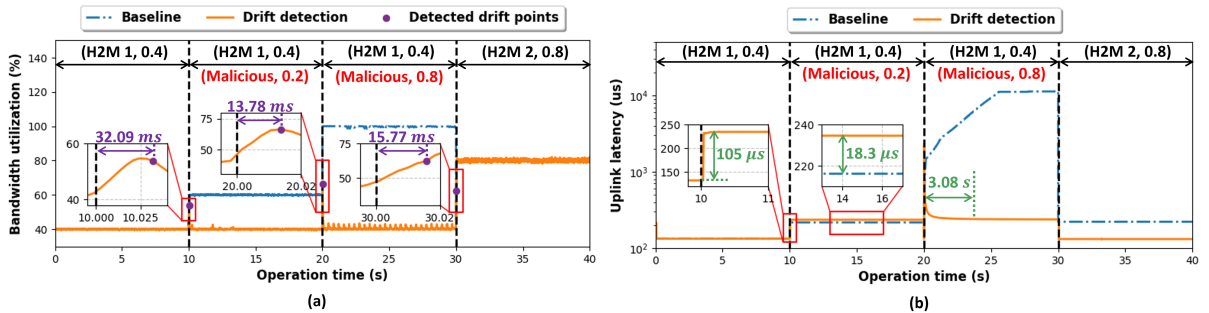


Fig. 2. Timing diagram of: (a) Bandwidth utilization; and (b) Uplink latency

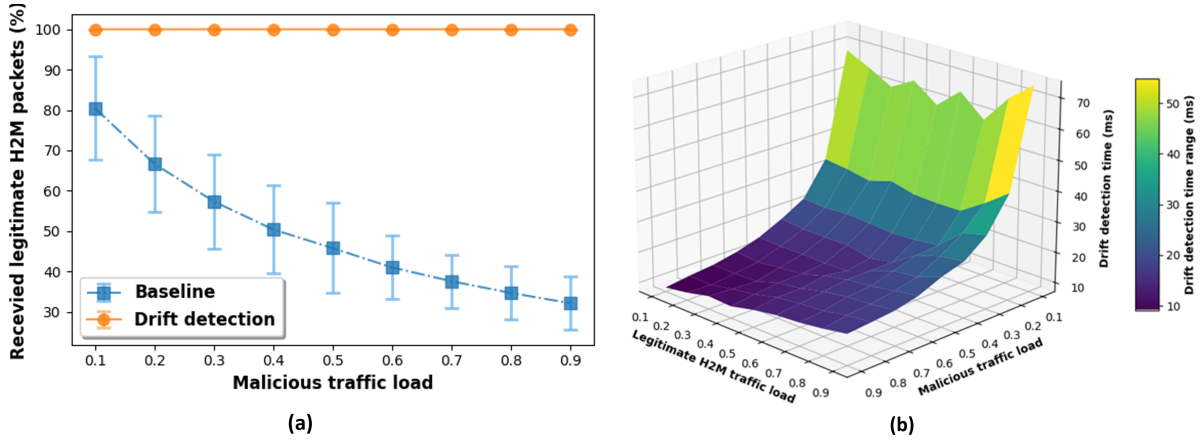


Fig. 3. (a) Operational security, (b) Drift detection time

132.22 μs to 218.92 μs . However, the framework causes the uplink latency to rise to 237.22 μs , primarily due to the additional time required for packet inspection and inference, i.e. the traffic classification inference time (100 μs) and the packet processing time through the entire stack of layers within the ONU (0.112 μs [5]). This indicates a slight impact of the *traffic classifier* on latency when handling lower-intensity malicious traffic. When the malicious traffic load is 0.8, the uplink latency of the baseline scenario reaches up to 11.17 ms , far exceeding the maximum allowable latency for H2M applications. In contrast, with the framework in place, after a recovery period of approximately 3.08 s , the uplink latency stabilises back to 237.22 μs , yielding 98% latency reduction compared to the baseline. Latency results from the last 10 seconds in Fig. 2(b) demonstrate the effectiveness of the framework in handling the concept drift between two legitimate H2M applications, with a drift detection time of 15.77 ms .

To quantify the enhancement of the operational security of H2M applications, Fig. 3(a) presents the percentage of received legitimate H2M packets over total packets received at the CO, as a function of the malicious traffic load. Specifically, a lower percentage reflects less secure operations, while a higher percentage—achieved by filtering out malicious packets—indicates enhanced operational security. In the baseline case, the percentage of received legitimate H2M packets drops to 66.68% and 34.72% at 0.2 and 0.8 malicious traffic loads, respectively. This highlights the excessive bandwidth utilization caused by the malicious traffic as shown in Fig. 2(a), leading to fewer legitimated H2M packets being transmitted and degrading the operational security of H2M applications. However, thanks to the *traffic classifier* at the ONU, most of malicious packets are filtered out. Despite the slight increase in overall uplink latency due to packet inference, the percentage of received legitimate H2M application packets improves significantly to near 100% for every malicious traffic load. This additional latency is a trade-off for the substantial gain in maintaining the operational security of H2M applications. Fig. 3(b) illustrates the average drift detection time incurred by the *drift detector*, as a function of the legitimate H2M traffic and the malicious traffic loads. Results indicate a negative correlation between these parameters, particularly when the legitimate H2M traffic load is constant, and can be attributed to a longer packet inter-arrival time caused by the low malicious traffic load.

4. Conclusion

This paper proposed a framework to detect and mitigate malicious traffic in H2M applications by leveraging concept drift detection methods. Simulation results validated that: (a) bandwidth utilization is optimized and up to 98 % latency reduction is achieved by using concept drift detection methods and a traffic classifier at the CO and at the ONU; and (b) a significant portion of malicious packets is filtered out to guarantee the operational security of H2M applications.

References

- [1] S. Mondal, *et al.*, "Remote human-to-machine distance emulation through AI-enhanced servers for tactile internet applications," in *Proc. of OFC*, 2020.
- [2] M. Freiburger-Verizon, *et al.*, "Low latency networks: future service level use cases and requirements," in *Proc. of OFC*, 2018.
- [3] E. Wong, *et al.*, "Machine learning enhanced next-generation optical access networks—challenges and emerging solutions," *JOCN*, vol. 15, no. 2, pp. A49–A62, 2023.
- [4] A. Azab, *et al.*, "Network traffic classification: Techniques, datasets, and challenges," *DCN*, vol. 10, no. 3, pp. 676–692, 2024.
- [5] G. Pongracz, *et al.*, "Removing roadblocks from SDN: OpenFlow software switch performance on Intel DPDK," in *Proc. of EWSDN*, pp. 62–67, 2013.