# On Performance Analysis of Quantized Unsupervised Learning for In-Kernel Flow-Based Intrusion Detection Systems*

**Hotaka TAGUCHI**[†a)], ***Student Member***, **Takanori HARA**[†b)], ***Member***, ***and*** **Shoji KASAHARA**[†c)], ***Fellow***

## 1. Introduction

*Quantization* is a technique that converts a floating-point model to a lower-bitwidth integer model, reducing the model size and inference speed at the expense of inference accuracy [1]. To maintain inference accuracy, the quantization parameters should be adjusted appropriately. In this paper, we examine how quantization affects inference accuracy and speed of an unsupervised learning model, specifically *autoencoders* (AEs), by comparing two representative quantization algorithms (i.e., *post-training quantization* (PTQ) and *quantization-aware training* (QAT)). In [2], the authors proposed a PTQ-based supervised learning model for an in-kernel (eBPF-assisted) flow-based intrusion detection system (IDS) as a use case. We further explore the potential of the two quantization algorithms, including the appropriate algorithm selection and the applicability to the in-kernel IDS, in terms of accuracy, speed, and model size.

## 2. Quantized Autoencoder Model for IDS

An AE model consists of an encoder and a decoder. The AE-based IDS is trained with the 5-tuple flow-related features [2] only consisting of normal packets such that the gap between the $D$-dimensional input vector $\boldsymbol{x} = (x_1, \ldots, x_D)$ and output one $\boldsymbol{y} = (y_1, \ldots, y_D)$ (i.e., reconstruction error $e = 1/D \cdot \sum_{i=1}^{D}(x_i - y_i)^2$) is minimized. In the inference phase, the AE-based IDS estimates a flow as abnormal if its reconstruction error $e$ is larger than a predefined threshold $\theta$. The quantization process is performed by $x_q = \mathrm{round}(s/x) + z$, where $x$ and $x_q$ mean a floating value and a quantized one and $s$ and $z$ stand for a scale factor and a zero-point value. The PTQ algorithm adjusts $s$ and $z$ by feeding calibration data to the trained model, while the QAT algorithm fine-tunes them during the training process.

## 3. Evaluation Results

For the evaluation, a server with Apple M1 Ultra and 128 GB memory is used. For comparison purposes, we prepare three models: Floating model, PTQ-based model, and QAT-based model. We first train the three models using the CICIDS-2017 dataset [3] and then evaluate the inference accuracy and speed of the three models over flow duration $t$. We confirm from Fig. 1 that all models improve the inference accuracy at $t \geq 20$ and then saturate at $t \geq 100$ because the flow-related information is updated over time $t$. On the other hand, the QAT-based model shows almost the same accuracy as the floating model, thanks to the fine-tuned quantization parameters during training. The training time for the floating model is 30.7 s, while those of the PTQ-based and QAT-based models are 31.55 s and 79.36 s, respectively. However, the inference time (resp. the model size) of floating model is 37.6 $\mu$s (resp. 4.46 KB), while those of the PTQ-based and QAT-based models are 20.5 $\mu$s and 19.6 $\mu$s (resp. 1.99 KB and 1.99 KB). These results indicate that the QAT-based model is suitable for the in-kernel IDS.

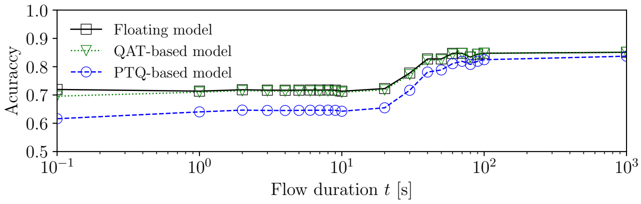


**Fig. 1** Inference accuracy over flow duration $t$.

## 4. Conclusion

In this paper, we examined the impact of quantization on the inference accuracy and speed of an unsupervised learning model for in-kernel IDS, comparing two quantization algorithms. The results show that the QAT-based model achieves similar inference accuracy to the floating model while reducing both model size and inference speed.

[†]The authors are with the Division of Information Science, Nara Institute of Science and Technology, Nara, 630-0192 Japan.

*This work was supported in part by the JSPS Grant-in-Aid for Young Scientists under Grant 23K16869 and the JSPS KAKENHI (B) under Grant 24K02931.

a) E-mail: taguchi.hotaka.tc3@is.naist.jp
b) E-mail: hara@ieee.org
c) E-mail: kasahara@ieee.org

### References

[1] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," pp.2704–2713, 2018.

[2] T. Hara and M. Sasabe, "Practicality of In-Kernel/User-Space Packet Processing Empowered by Lightweight Neural Network and Decision Tree," Computer Networks, vol.240, p.110188, Feb. 2024.

[3] I. Sharafaldin, A. Habibi Lashkari, and A.A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," Proc. of ICISSP, pp.108–116, 2018.