



HAL
open science

A Profit-aware Adaptive Approach for In-Network Traffic Classification

Muhammad Saqib, Halima Elbiaze, Roch Glitho

► **To cite this version:**

Muhammad Saqib, Halima Elbiaze, Roch Glitho. A Profit-aware Adaptive Approach for In-Network Traffic Classification. IEEE International Conference on Communications, May 2023, Rome, Italy. hal-04139681

HAL Id: hal-04139681

<https://hal.science/hal-04139681v1>

Submitted on 27 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Profit-aware Adaptive Approach for In-Network Traffic Classification

Muhammad Saqib*, Halima Elbiaze*, Roch Glitho†

* Université du Québec à Montréal, Montreal, Canada

† Concordia University, Montreal, Canada

saqib.muhammad@courrier.uqam.ca, elbiaze.halima@uqam.ca, glitho@ece.concordia.ca

Abstract—In-network traffic classification is a new paradigm in developing accurate and early-stage traffic classification solutions. However, despite having good accuracy, the one-fit machine learning model becomes outdated as the traffic pattern changes over time. This changing traffic pattern leads to misclassification, i.e., incorrect mapping of traffic flows to the Quality of Service (QoS) classes, resulting in a service quality violation and the imposition of a penalty. This paper proposes a profit-aware adaptive traffic classification approach in the data plane. We particularly design an economic model to measure the impact of per-class misclassification rate on the infrastructure provider's profit and use an adaptive method to handle misclassification directly inside a programmable data plane. The evaluation result shows that optimal path allocation for various traffic classes determines the targeted revenue, while improving classifier accuracy reduces penalty and maintains the maximum profit.

Index Terms—In-Network traffic classification, misclassification, QoS, programmable data plane.

I. INTRODUCTION

With a surge in latency-sensitive services and applications in the Internet of Things (IoT) domain, applying diversified Quality of Service (QoS) policies and efficiently utilizing network resources is crucial. Network Slicing (NS) emerged as a promising solution for resource orchestration to achieve QoS isolation by overlaying multiple virtual networks on top of a shared network domain [1]. It enables efficient usage and management of network resources and provides differentiated services at scale. Hence, a particular service can use a dedicated network slice to facilitate the QoS requirements [2]. Along with this, network traffic classification is also an essential task in computer network management. It enables the provision of application-specific QoS guarantees [3] by mapping the incoming traffic flows to the appropriate QoS slice and thereby simplifies the enforcement of Service Level Agreements (SLAs).

Recently, in-network traffic classification turned out to be the key enabler in accurate and early-stage traffic classification solutions [4]. However, in IoT domain, the traffic presently includes a variety of behaviours such as communication types, events, sources, patterns and volumes etc [5], [6]. These behaviours considerably impact traffic patterns, management, and control. As a result, based on the number and type of active devices in the network, the devices' behaviour might

generate a variety of characteristics, i.e., variability in data transmission period and payload size. Machine learning (ML) has rapidly become a viable option for identifying the source devices and application types from Internet traffic. Despite having a traffic classifier with good accuracy, a single-time trained ML-model becomes outdated as the traffic pattern changes over time [7]. This changing traffic pattern leads to *misclassification*, i.e., incorrect mapping of traffic flows to the QoS class.

The work in [8] approaches misclassification from a cost-aware model training perspective, having a minimal risk of inaccuracy. The authors argue that a solution for device identification (or classification) should priorly consider feature extraction's cost (computational and memory). They particularly define budget constraints for features and perform a combinatorial search over the solution space to obtain the optimal (i.e., cost-effective) features. Similarly, another work in [9] considers a cost-sensitive learning strategy to ensure the robustness of traffic classifiers against unbalanced datasets. They consider the misclassification cost during the training process and subsequently minimize the training model's cost. However, despite having a cost-effective model with good classification accuracy, the one-fit ML model loses relevance over time as the traffic pattern changes. This loss of accuracy leads to incorrect mapping of traffic flows to the QoS class, which further results in SLA violations and affects customer satisfaction in return. As a result, on the one hand, the service provider tries to increase revenue through priority-based traffic scheduling [10]. On the other hand, incorrect QoS class mapping by the classifier leads to SLA violations, resulting in penalties that negatively impact the provider's profit. A significant challenge for Infrastructure Providers (InPs) is ensuring multi-priority traffic demands while maintaining maximum profit.

Changing traffic patterns are the root cause of a trained ML model's accuracy degradation. Therefore, it is essential to investigate the impact of misclassified flows and handle the misclassification problem while streaming data is being collected. In this work, we investigate the impact from an economic perspective on the provider's profit and propose an adaptive traffic classification approach to handle misclassification inside the data plane. In particular, we first specify the service quality requirements boundaries for a few emerging applications and formulate the problem for multi-priority traffic scheduling

This work was fully supported by CHIST-ERA program under the "Smart Distribution of Computing in Dynamic Networks (SDCDN)" 2018 call.

TABLE I: Service quality requirements of a few emerging applications [11]

Use case / Applications	Traffic Class	Data rate(γ)	E2E Latency(φ)	Priority (α)
Personalized BAN	Non-real-time/Critical rate ($X_t R_a$)	1-100 Mbps	≤ 1000 ms	α_1
Internet of Everything	Non real-time/Non-critical rate ($X_t X_a$)	100-1000 Mbps	≤ 100 ms	α_2
Smart Grid 2.0	Real-time/Critical rate ($R_t R_a$)	100-2000 Mbps	≤ 10 ms	α_3
Live streaming	HD real-time/Highly critical rate ($HR_t HR_a$)	2000-4000Mbps	≤ 1 ms	α_5

TABLE II: Table of notations

Notation	Description
N	Set of nodes
L	Set of links
S	Set of network slices
K	Set of paths between source and destination pairs
C	Set of priority class (i.e., QoS group)
F	Set of traffic flows
f_i^c	Traffic flow i of class c
$\varphi_{p,i}^c$	Associated delay with packet p belongs to f_i^c
$\gamma_{l,i}^c$	Data rate of link l on flow i belongs to class c
b_l	Bandwidth of link l
x_k^s	Traffic volume of slice s on path k
α_i^c	Service priority of flow i belongs to class c
β	Monetary penalty unit
δ	The selling price for a unit of bandwidth
λ^c	Misclassification rate of a particular class c

to increase the provider's revenue. We then put forth an economic model to quantify traffic misclassification's impact on the InP's profit and use an adaptive traffic classification method to handle misclassification. Improving the classifier's accuracy ensures that multi-priority traffic demands are met while maximum profit is maintained. Our contributions, in particular, are as follows:

- We characterize the traffic demands from end users into various QoS classes for priority-based traffic scheduling;
- We design an economic model that uses penalty calculation and revenue generation functions to generate maximum revenue by guaranteeing multi-priority user demands. Also, we investigate the impact of per-class misclassification on the provider's profit;
- We propose a profit-aware adaptive traffic classification approach to handle misclassification directly inside a programmable data plane.

The remaining of this paper is organized as follows. Section II presents the problem definition and network model. Section III represents the proposed solution. The evaluation results are shown in Section IV. Finally, Section V represents the concluding remarks and future directions.

II. PROBLEM FORMULATION AND NETWORK MODEL

A. Toward traffic characterization

We begin by discussing the motivation for categorizing Internet traffic. Regarding service quality metrics such as latency φ and data rate γ , we translated the QoS requirements of a few emerging applications to various priority classes. TABLE I depicts the set of traffic priority classes, along with their appropriate service quality requirements. Each class has its own characteristics, such as high definition HD , real-time R_t or non-real-time traffic X_t - indicating the bandwidth and latency requirements; critical R_a or non-critical data rate X_a - expressing the level of delay tolerance [12].

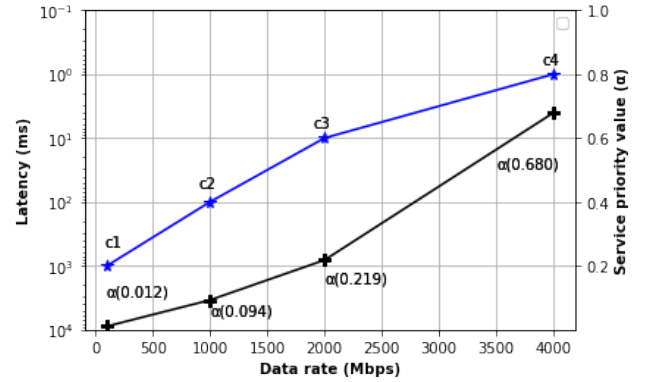


Fig. 1: Service priority factor calculation

B. Problem definition

Let $c \in C$ be the set of priority classes, each with its service quality parameters, namely the latency φ and data rate γ . Here, the latency φ is defined in terms of end-to-end packet delay on the chosen route, and the data rate γ is given in bits per second as specified by the user capacity requirement. As a result, each class c is bounded by threshold values of latency φ_{max}^c and data rate γ_{min}^c . All the defined variables are summarized in Table II.

Further, the priority of a class c to be privileged in data delivery is defined in terms of its service quality requirements. The service priority factor, α_i^c defines the initial priority value of the flow i belongs to class c for allocating resources. The α_i^c increases as the required data rate and latency requirements tighten. Therefore, we grouped the multi-priority traffic demands into different classes and assigned a priority value α to each class c (see TABLE I). We defined α as being based on the criticality level, i.e., the service quality requirements of the classes, and that it is proportional to the amount of required data rate and inverse of latency. The blue line in Fig. 2 highlights the criticality level of each class and shows how it expands with the increase in data rate and decrease in latency requirements. Since α reflects the relative importance of the classes, it is then used to define the penalty and revenue generation functions.

Substrate network and constraints: We represent the underlying physical infrastructure as a directed graph $G = (N, L)$ with a set of nodes $n \in N$ and links $l \in L$, each link with a bandwidth $b_l > 0$ (measured in bps). On top of the substrate network, we consider the co-existence of multiple network slices indexed by $s \in S = \{1, \dots, S\}$. For clarity and simplicity, we focus on a scenario where each slice s serves network traffic from a single traffic class c with a single source to a single destination pair [13] having different service parameters in terms of φ and γ . Hence, we represent a slice by a source-

destination (S/D) pair (u, v) where the following constraints must be respected for traffic flows between any pair:

$$\varphi_{p,i}^c \leq \varphi_{max}^c, \forall p \in f_i^c \in F^c \quad (1)$$

$$\gamma_{l,i}^c \geq \gamma_{min}^c, \forall l \in L, \forall f_i^c \in F^c \quad (2)$$

The above constraint (1) ensures an end-to-end delay threshold along the multi-hop route. The associated end-to-end delay φ with a packet p across the multi-hop route between source node u and destination node v , in particular, shall not exceed the maximum latency limit for a given class c , i.e., φ_{max}^c . Furthermore, constraint (2) assures that the data rate γ at any link $l \in L$ should be sufficient to meet the capacity demand of passing flow i belongs to class c .

In addition, the paths between (u, v) pairs over links L are indexed by $k \in K = \{1, \dots, K\}$. We further denote the traffic volume each slice s generates as x_k^s going through path k . Since multiple slices share the same physical network, the bandwidth consumption at each link may not exceed the available bandwidth. Thus, we also define the link's bandwidth constraint:

$$\sum_{s \in S} \sum_{k \in K} x_k^s \leq b_l, \forall l \in L \quad (3)$$

Provider's revenue: The service provider generates revenue by optimal resource allocation to heterogeneous traffic demands. The revenue mainly depends on the resources requested by the service. That is the product of the selling price of a bandwidth unit and a class's service priority factor. The pricing policy determines the charge per unit bandwidth for each substrate link $l \in L$. A differential pricing policy is considered based on the class's criticality level. Thus, the revenue gained by the provider at the time t by selling the bandwidth resource can be expressed as:

$$\sigma(t) = \sum_{c \in C} \sum_{f \in F^c} \sum_{s \in S} \sum_{k \in K} \delta * \alpha_f^c * x_k^s(t) \quad (4)$$

The selling price for a unit of bandwidth is denoted as δ . The priority of flow f of class c (i.e., service quality priority) is represented as α_f^c , and $x_k^s(t)$ stands for satisfying bandwidth over a path k between (u, v) pairs of slices $s \in S$.

Network traffic classifier: The InP increases revenue by allocating resources to various classes of traffic. While a traffic classifier at the network's edge assigns the incoming traffic flows to the correct traffic class, i.e., QoS slice. In the case of incorrect traffic class mapping, the InP would be unable to implement the appropriate QoS policies, which would impact the CSL. In order to maintain good classification accuracy, it is crucial to regularly check the classifier's prediction and penalize the classifier for inaccurate mappings.

Let \mathcal{C} be the classifier that classifies the incoming traffic flows f_i to the corresponding class. Any misclassification,

i.e., incorrect QoS slice mapping, might result in an SLA violation, adding a penalty that can be calculated as the per-class criticality level.

Misclassification rate: We use *fl-score* as a performance metric to measure the per-class misclassification rate. *fl-score* assesses the classification model's performance starting from the confusion matrix; it aggregates *Precision* and *Recall* measures under the concept of harmonic mean [14]. The formula of *fl-score* can be interpreted as a weighted average between *Precision* and *Recall*:

$$f1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (5)$$

where

$$Precision = \frac{TP}{(TP + FP)} \quad (6)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (7)$$

TP means the observation is positive, and the sample is predicted to be positive. *FN* is that the observation is positive, but the sample is predicted to be negative. *TN* describes that observation is negative, and the sample is predicted to be negative. And *FP* represents that observation is negative, but the sample is predicted to be positive.

The *fl-score* reaches its best value at 1 and the worst score at 0. Hence, the misclassification rate of a particular class c can be defined as:

$$\lambda^c = 1 - f1^c, \forall c \in C \quad (8)$$

Penalty: The provider is required to return the penalty incurred due to the misclassification. The penalty ρ for a particular class c over time t can be calculated as a product of the monetary penalty unit, the class's priority, and the misclassification rate of that class. The total penalty ρ associated with each class c at the time t can be computed as follows:

$$\rho(t) = \sum_{t \in T} \sum_{c \in C} \beta * \alpha^c * \lambda^c \quad (9)$$

where β is the monetary penalty unit, α^c is the class priority and λ^c is the misclassification rate of class c .

Objective function: The objectives are to maximize the provider's profit and minimize SLA violations. Maximizing the provider's profit can be achieved by maximizing σ and minimizing ρ . The objective function \mathcal{P} can therefore be written as follows:

$$\max_{\mathcal{P}} \sum_{t \in T} (\sigma(t) - \rho(t)) \quad (10)$$

Subject to constraint (1), (2) and (3).

III. PROPOSED SOLUTION

This section discusses the system design and outlines the steps needed to increase revenue while maintaining maximum profit. Fig. 2 explains the high-level system design, mainly consisting of two phases: (i) offline training of the ML model and determining the optimal routing path at the control plane; (ii) traffic class identification and network slice allocation in the data plane.

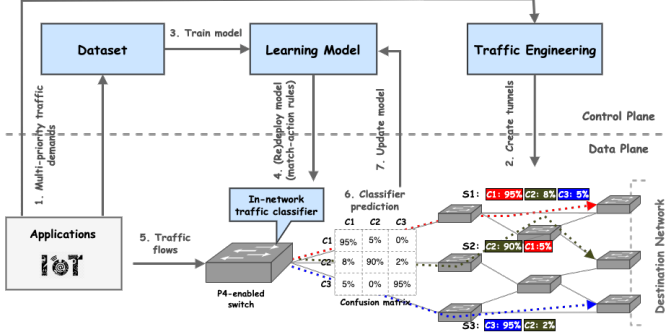


Fig. 2: System design

In the first phase, the control counterpart determines the optimal routing paths for multi-priority traffic demands to meet the traffic demands and maximize the provider revenue. The revenue function takes traffic demands and available routes with bandwidth capacity as inputs, resulting in maximum revenue as an output. Thus, the optimal allocation of traffic demands to routing paths between source and destination networks is determined by formulating the optimization problem as an Integer Linear Program (ILP). In addition, the control plane trains the ML model on a given dataset. A decision tree classifier is used to train the model based on the sequential packet size information. The resulting output (i.e., *if-else* conditions) is translated into a P4-enabled switch [15] to identify the flow class at runtime in the data plane and apply the corresponding traffic forwarding rules.

The following step is correctly mapping incoming traffic flows to the appropriate QoS class. Once the switch receives the first few packets for an individual flow, it extracts relevant features such as five tuples (IP address, ports, and protocol) and sequential packet size information and determines the corresponding class for that flow. The switch then directs the flow's subsequent packets to the determined slice for that specific class. We refer to [4] for more details about traffic flow classification inside a programmable data plane.

The next step is calculating the per-class misclassification rate from the classifier's prediction and periodically updating the ML model to maintain classification accuracy. We periodically monitor the classifier's performance by calculating the *f1-score* from the data plane's prediction. Because of changes in traffic patterns, the deployed model loses accuracy over time, resulting in flows being mapped to the incorrect QoS class. The traffic flows from a set of classes C are to be mapped to a set of slices S ; therefore, the traffic engineering

module is to create a separate tunnel for each slice. As shown in Fig. 2, for example, the arrows coloured as *red*, *green*, and *blue* represent three different slices with distinct bandwidth capacities and propagation delays. The first arrow shows that misclassification causes 8% flows of c_2 and 5% flows of c_3 to be routed through slice s_1 . Likewise, 5% flows of c_1 and 2% flows of c_2 are routed via s_2 and s_3 , respectively. The desired QoS policies are not applied to that fraction of flows, resulting in a service quality violation and penalty accumulation. Therefore, we regularly fit newly received data into the existing model to maintain an updated model inside the data plane, ensuring that flows are predicted correctly, and misclassification rates are kept to a minimum.

Algorithm 1: Maximizing InP's profit

Input: C, K, F
 // C : A list of classes with varying traffic demands
 // K : A list of capacity-varying routes connecting source and destination networks
 // F : Network traffic flows from different class of traffic (C)
Output: \mathcal{P}
 // \mathcal{P} : Profit
 $\mathcal{P} = 0$;
for $t \in T$ **do**
 $\sigma(t) = \text{Scheduling}(C, K)$; // Algorithm 2
 $\rho(t) = \text{Misclassification}(F)$; // Algorithm 3
 $\mathcal{P}(t) = \sigma(t) - \rho(t)$; // Profit over period ' t '

Algorithm 2: Scheduling

Input: C, K
Output: max_revenue
 $\text{max_revenue} = 0$;
for $c \in C$ **do**
 for $k \in K$ **do**
 for each feasible solution do
 $\sigma = \sum_{c \in C} \sum_{k \in K} \delta * \alpha^c * x_k^c$;
 if $\sigma > \text{max_revenue}$ **then**
 $\text{max_revenue} = \sigma$;
return max_revenue

Algorithm 3: Misclassification

Input: F
Output: ρ
 $\text{penalty} = 0$;
for $c \in C$ **do**
 $\lambda^c = 1 - f_{1F}^c$;
 $\rho += \beta * \alpha^c * \lambda^c$;
return ρ

Algorithm 1-3 outlines the entire InP profit maximization process. First, the scheduling function takes traffic demands and network resources, such as route bandwidth capacity, as input and returns the maximum revenue resulting from optimal path allocation. The misclassification function is then periodically called to compute the per-class misclassification rate and the associated penalty. The revenue and penalty are essentially calculated by multiplying the class priority factor to service quality requirements and the misclassification rate of that specific class, respectively. In other words, the revenue and penalty will increase in proportion to the criticality level

of the class. Finally, the penalty is periodically subtracted from the targeted revenue and the maximum profit is obtained over time.

IV. EVALUATION RESULTS

The evaluation section consists of two parts: meeting multi-priority traffic demands for maximizing revenue and measuring the misclassification impact on the InP's profit.

A. Satisfying multi-priority traffic demands

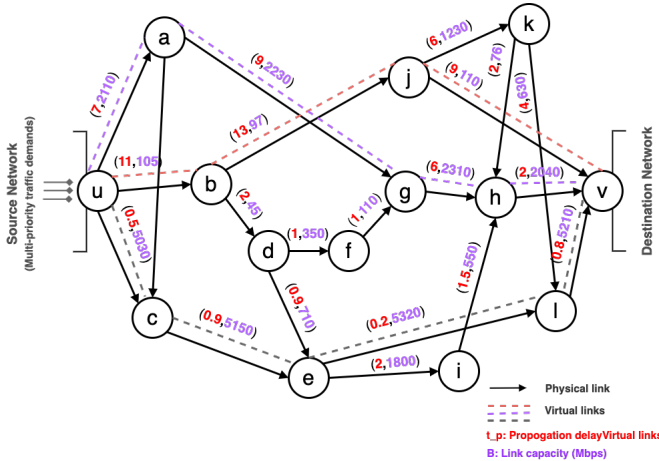


Fig. 3: Network topology

We consider the performance requirements for various classes of traffic defined in TABLE I. The InP makes revenue by meeting the traffic demands with distinct service quality parameters. The selling price for a single unit of bandwidth is supposed to be \$5. As the optimal allocation of traffic demands to routing paths between source and destination networks is formulated as an optimization problem, we let the solver find the optimal paths over a network topology (See Fig. 3). The provider then allocates the routing paths to meet the traffic demands. Based on the satisfiable traffic demands, the provider calculates the targeted revenue using the revenue generation function defined in Section II. The generated revenue for the specified classes in Table I is, for example, shown in Fig. 4. The bar graph demonstrates that the correlation between bandwidth demand and revenue is more robust when a class has a high criticality level, such as *class-4*. *Class-2* exhibits the opposite behaviour due to its lower criticality level. So, the InP finds the optimal route for each traffic class to meet the demands with maximum revenue.

B. Impact of misclassification

The next step is measuring the misclassification's impact on the provider's profit. Since the network's traffic pattern evolves rather than staying constant, this phenomenon is referred to as *concept drift* [6]. As a result, a model's accuracy deteriorates over time as the traffic pattern changes. Therefore, a trained model could be updated continuously or infrequently while streaming data is being collected [16].

In our evaluation, we use packet capture (PCAP) traces of IoT devices released by [5] as our dataset. From the available dataset instances, we selected the PCAP files for seven days

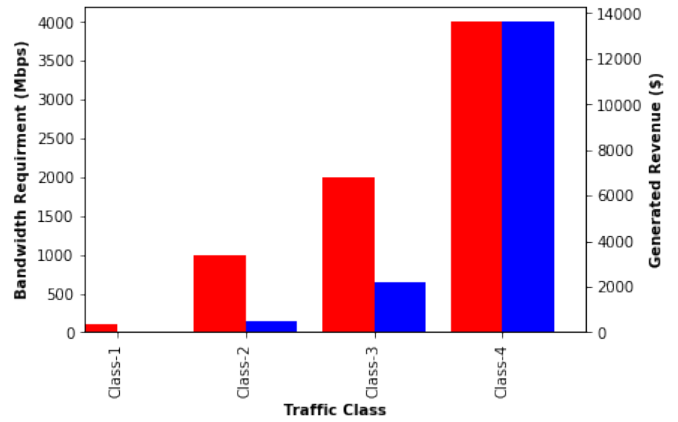


Fig. 4: Per-class revenue determined by optimal path allocation

(from 23 to 29 Sep 2016), containing flows related to four applications comprising different IoT devices. The devices have been divided into several classes with varying degrees of priority. We selected devices that can be assigned to various QoS groups: from high bandwidth and low latency to best effort. The devices belonging to the same class share the same traffic characteristics. Therefore, we selected a single device from each class for evaluation purposes. TABLE III shows a summary of the dataset for these chosen devices.

TABLE III: Dataset summary

Class	Device type	# of flows	# of packets
C1($X R_a$)	Sensor	6411	233329
C2($X X$)	Appliance	5439	23561
C3($R_t R_a$)	Controller	16788	270840
C4($H R_t H R_a$)	Camera	1601	144187

TABLE IV: Per-class $f1$ -score

Class	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Class-1	1.00	0.95	0.98	0.98	0.97	0.99	1.00
Class-2	0.99	0.50	0.91	0.97	0.99	1.00	0.95
Class-3	0.98	0.93	0.93	0.95	0.94	0.98	1.00
Class-4	0.93	0.84	0.82	0.98	0.94	0.98	1.00

The dataset is divided into multiple chunks to evaluate model drift identification and adaptation (i.e., to learn the newly received data samples), each one representing the daily streaming data. The first data chunk has been used for training, while the subsequent fragments have been used for testing purposes. To identify the model drift, we inspected new data patterns on the *2nd day* and sporadically monitored how well the model performed on contemporary data patterns. As shown in TABLE IV, the $f1$ -score decreased to 50% on the *2nd day* due to input changes in the source data.

The similarities between the input data, which resulted in the incorrect mapping of the incoming traffic to the QoS class, caused accuracy degradation, resulting in QoS degradation. Based on per-class's misclassification rate and priority factor, we calculated the penalty, which is supposed to be \$50 for each % of $f1$ -score loss. Due to the distinct priority factor of various traffic classes, the inaccuracy of each class contributed to the penalty at a different rate. For instance, it is clear from TABLE IV that *class-2's* $f1$ -score of the *2nd day* is significantly lower than *class-4*. However, the penalty added by *class-4* is considerably higher. (see Fig. 5). Therefore,

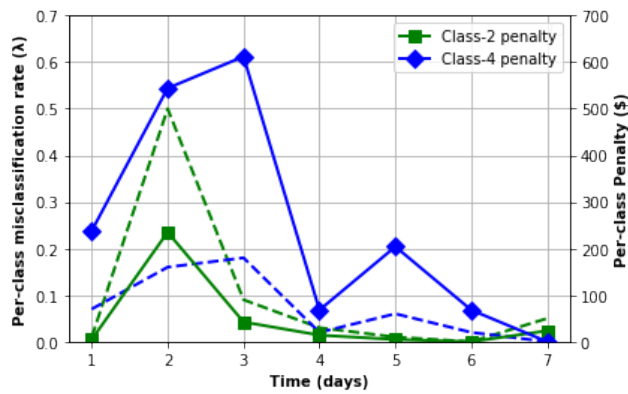


Fig. 5: Per-class misclassification rate vs. penalty variation

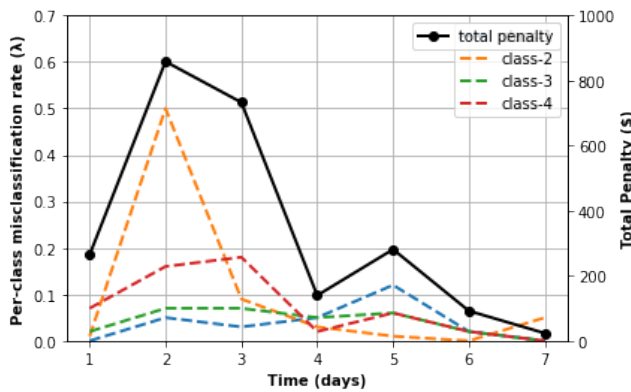


Fig. 6: Per-class misclassification rate vs. total penalty

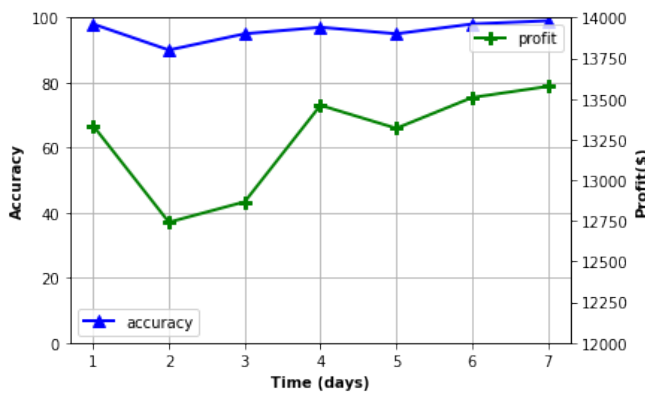


Fig. 7: Impact of improving accuracy on the InP's profit

the impact of the misclassification rate on the total penalty depends on the significance of traffic class, as shown in Fig 6.

We periodically fitted the newly received patterns to the existing model to reduce the impact of misclassification. It is evident from Fig. 7 that accuracy is gradually improved over time, directly affecting the overall profit. In other words, increasing classifier accuracy resulted in accurate QoS class mapping of incoming traffic, lowering penalties while maintaining a higher profit.

V. CONCLUSION & FUTURE DIRECTION

In this paper, we propose an approach to investigate the impact of misclassification from an economic perspective and

use an adaptive classification method to handle the misclassification problem. We evaluated the proposed method by characterizing multi-priority traffic to various QoS classes, optimally allocating routing paths to traffic classes, and mapping the incoming flows to the appropriate QoS class. The evaluation results show that the misclassification results in adding a penalty, where the penalty rate varies as per-class criticality level. The adaptive behaviour of the classifier improved the prediction's accuracy, resulting in accurate QoS class mapping of the incoming traffic flows, lowering penalties while maintaining a higher profit. This work can be extended to measure the impact of misclassified flows on the utilization of network resources and quantify the effect on QoS and Quality of Experience (QoE).

REFERENCES

- [1] Y. Wu, H.-N. Dai, H. Wang, Z. Xiong, and S. Guo, "A survey of intelligent network slicing management for industrial iot: integrated approaches for smart transportation, smart energy, and smart factory," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 1175–1211, 2022.
- [2] J. J. A. Esteves, A. Boubendir, F. Guillemin, and P. Sens, "Optimized network slicing proof-of-concept with interactive gaming use case," in *2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, 2020, pp. 150–152.
- [3] H. Yao, P. Gao, J. Wang, P. Zhang, C. Jiang, and Z. Han, "Capsule network assisted iot traffic classification mechanism for smart cities," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7515–7525, 2019.
- [4] M. Saqib, A. H. Zakaria, H. Elbiaze, and R. Glioth, "An accurate & efficient approach for traffic classification inside programmable data plane," in *IEEE Global Communications Conference*, 2022.
- [5] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2018.
- [6] I. Žliobaitė, M. Pechenizkiy, and J. Gama, "An overview of concept drift applications," *Big data analysis: new algorithms for a new society*, pp. 91–114, 2016.
- [7] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2018.
- [8] B. Chakraborty, D. M. Divakaran, I. Nevat, G. W. Peters, and M. Gurusamy, "Cost-aware feature selection for iot device classification," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 052–11 064, 2021.
- [9] A. Telikani, A. H. Gandomi, K.-K. R. Choo, and J. Shen, "A cost-sensitive deep learning-based approach for network traffic classification," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 661–670, 2021.
- [10] Y. Xu and D. Xu, "Maximizing profit of network inp by cross-priority traffic engineering," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.
- [11] C. De Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6g frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 836–886, 2021.
- [12] B. Afzal, S. A. Alvi, G. A. Shah, and W. Mahmood, "Energy efficient context aware traffic scheduling for iot applications," *Ad Hoc Networks*, vol. 62, pp. 101–115, 2017.
- [13] M. Leconte, G. S. Paschos, P. Mertikopoulos, and U. C. Kozat, "A resource allocation framework for network slicing," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2177–2185.
- [14] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," *arXiv preprint arXiv:2008.05756*, 2020.
- [15] "bmv2," <https://github.com/p4lang/behavioral-model/>, online; accessed 12 November 2022.
- [16] A. Shahraki, M. Abbasi, A. Taherkordi, and A. D. Jurcut, "A comparative study on online machine learning techniques for network traffic streams analysis," *Computer Networks*, vol. 207, p. 108836, 2022.