Machine Learning Based 5G Network Slicing Management and Classification

Zong-Xun Wu
Department of Computer Science and Information Engineering
National Central University
Taoyuan, 320317 Taiwan
Mrplastic@networklab.csie.ncu.edu.tw

Yun-Zhe You
Department of Computer Science and Information Engineering
National Central University
Taoyuan, 320317 Taiwan
victor@networklab.csie.ncu.edu.tw

Chien-Chang Liu
Department of Computer Science and Information Engineering
National Central University
Taoyuan, 320317 Taiwan
c34lcd@yahoo.com.tw

Li-Der Chou
Department of Computer Science and Information Engineering
National Central University
Taoyuan, 320317 Taiwan
cld@csie.ncu.edu.tw

***Abstract*—Due to the rapid development of the Internet, network bandwidth and stability are becoming more and more important. With the increase in the number of users, how to make each user have a high Quality of Service (QoS) is an urgent problem to be solved. 5G slicing allows flexible management of each user's network usage, which in turn optimizes the overall network usage and reduces the consumption of network resources. The 5G slicing can flexibly manage each user's network usage to optimize overall network usage and reduce network resource consumption. In this paper, use machine learning to analyze the network traffic, and analyze a total of 141 different applications on the network, and conduct experiments on different machine learning models. Based on the above experimental results, propose an algorithm for 5G slice management. Based on the above traffic analysis results, we will dynamically configure and optimize the resources of each slice according to the current network traffic of each user.**

Keywords—5G, network slicing, machine learning, network throughput, network function virtualization.

I. INTRODUCTION

In recent years, the growth of IoT devices has also led to machine-to-machine (MTM) communication application services. In order to enable these emerging application services to be more flexibly deployed and programmatically controlled in the 5G network environment to meet the quality of service requirements of these emerging application services, such as high bandwidth and low latency, the key technology is Software Defined Network (SDN) and Network Functions Virtualization (NFV) [1], which can be centrally managed and programmatically controlled by SDN to enable real-time deployment of these services. With the centralized management and programmable features of SDN [2], dynamic changes in the network environment can be monitored in real time, while NFV can virtualize network functions to achieve the efficiency of flexible configuration and deployment of different types of services as well as to reduce deployment costs. Therefore, how to utilize microservices in the 5G network environment to respond to the large number of service requests from various devices and deploy them efficiently, while ensuring the independence of each network service to avoid information security issues, will be one of the difficult problems faced by 5G equipment manufacturers and network service providers.

Network slicing is a key technology in 5G networks that aims to provide user-specific network services, such as cloud services [3]. Therefore, using machine learning technology to allocate and optimize resources in network slicing management can improve the throughput of network traffic [4]. This study aims AI technology to learn different service types and classify traffic, analyze historical network traffic to predict possible future network behaviors, formulate slice allocation strategies based on the classification and prediction results, and automatically expand slices and manage the usage status of each slice by judging the current network conditions through dynamic scaling technology, while the selection of slices relies on the large amount of accurate and real-time information collected from the network. In the paper an AI-based traffic classifier and slicing resource allocation mechanism, along with slicing allocation strategies, will be proposed and developed. We adopt machine learning to predict the dynamic changes in network traffic, so as to make optimal decisions for allocating network slices to various services. In additional, based on resource availability and workload, a dynamic scaling algorithm for the dynamic resource allocation of slices is proposed, ensuring adaptability to the high dynamics and scalability of slices. Experimental results show that the AI traffic classifier, trained using Random Forest [5] and Gradient Boosting Decision Tree (GBDT) [6] models, achieved the best performance with an accuracy rate of up to 95.73%.

The remaining sections of this paper are organized as follows: Section 2 discusses background knowledge and related research; Section 3 introduces the system architecture and algorithms; Section 4 describes the experiments and discussions; and Section 5 concludes the paper.

II. BACKGROUND AND RELATED WORKS

This section introduces the research background and basic knowledge of this paper. Section A discusses 5G microservices and related research; Section B introduces network slicing; and Section C discusses auto-scaling.

A. 5G Microservices

Microservices is a software architecture where traditional applications often integrate a large number of services into a single monolithic deployment, but this deployment approach has its limitations. As the services provided by an application grow, the size of the program significantly increases, leading to deployment and usage overload issues. However, 5G microservices leverage the advantages of microservices to enhance the overall scalability of the network, providing greater flexibility in resource allocation.

When an application exhibits high coupling [7], maintenance and refactoring become challenging and time-consuming. In the event of a failure in a single module or service, the entire system becomes unavailable, impacting the

availability of other services. Utilizing the architecture of 5G microservices offers the following benefits: team members can independently work on individual services, accelerating the development process; adding new functionalities to microservices is relatively straightforward; due to the independence of microservices, the interruption of one microservice does not affect another when multiple microservices are in operation.

However, accommodating diverse service requirements poses new challenges for efficient 5G resource utilization. To address this, reference [8] introduce a novel Stochastic Optimization framework for Green Multimedia Services (SOGMS). SOGMS aims to maximize system throughput while minimizing energy consumption in data delivery. Leveraging Lyapunov optimization, it decomposed the optimization into three manageable subproblems: quality-of-experience-based admission control, cooperative resource allocation, and multimedia service scheduling. Extensive simulations compare SOGMS with state-of-the-art solutions in dense 5G networks, demonstrating its effectiveness.

B. *Network Slicing*

The primary function of network slicing [9] is to partition the physical network of a network operator into different virtual networks, each catering to specific service requirements such as latency, bandwidth, security, reliability, and more. This is achieved by interconnecting these virtual networks, allowing for the fulfillment of 5G environment demands.

Reference [10] proposed a SDN/FNV framework named "STREK" offers adaptable Quality-of-Experience (QoE) [11], security, and authentication functions across multi-domain cloud to edge networks. Its components include a holistic SDNFV data plane, NFV service-chaining, network slicing, and TREK, a lightweight hybrid cipher scheme. An open RESTful API lets applications deploy custom policies. In multi-domain/small-cell deployments, STREK uses dynamic flow/session-level key generation and efficient handover authentication, meeting 5G's low-latency requirements.

The allocation of 5G network slices is one of the hot topics in recent years, and it is very important to allocate suitable users to the corresponding slices. In order to solve this problem, an enhanced learning-based slice allocation technique is proposed in [12]which can dynamically allocate users to suitable slices according to the environment changes.

C. *Auto-Scaling*

The technology of dynamic scaling [13] is particularly crucial in 5G networks. Dynamic scaling allows for the automatic allocation of new resources such as CPU, memory, and storage resources during peak usage to maintain user Quality of Service (QoS). During off-peak periods, it automatically releases excess resources, reducing server costs. This enables 5G network slicing to achieve dynamic balance and automated resource allocation, enhancing the flexibility of the 5G network architecture.

III. System Architecture

This section introduces the overall system architecture and operation flow in detail. Section A introduces the system architecture; Section B introduces AI-based traffic classifier; Section C introduces slice resource configuration mechanism and slice allocation strategy.

A. *System Architecture*

The system architecture of this paper, as depicted in Figure 1, is divided into three main components: the AI traffic classifier [14], resource management, and slicing allocation strategy [15]. The AI traffic classifier is responsible for monitoring application traffic and utilizing 5G slicing to ensure a satisfactory user experience within the constraints of limited resources. When users connect to the base station, the system identifies and categorizes their application traffic based on packet information, and then allocates the traffic to one of three types of slices: lightweight, hybrid, or heavyweight. In cases where slicing network resources become insufficient, the system employs slicing management algorithms to reorganize resources and migrate users to slices with ample resources.

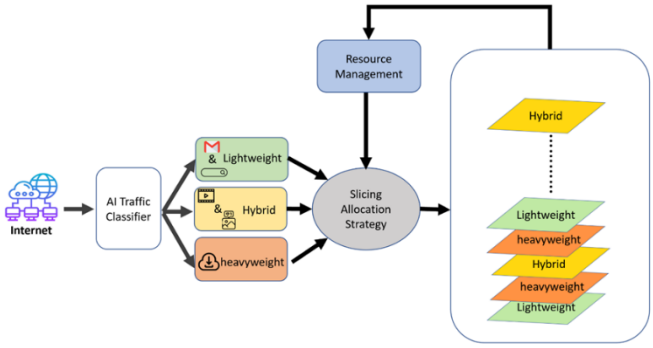

Fig. 1. System Architecture.

B. *AI-based Traffic Classifier*

With the emergence of new network services, computational resources exhibit high dynamics. In such cases, traditional traffic allocation mechanisms cannot effectively distribute resources, potentially leading to issues such as slice traffic overload or underutilization. AI-based traffic classifiers not only enable highly automated traffic classification and identification but also significantly reduce the workload of network administrators [16].

To address the issue of uneven resource allocation in 5G slicing, this paper employs an AI traffic classifier for traffic steering. Traffic is allocated to the appropriate slice based on the current service, as illustrated in Figure 2. Before model training, relevant network traffic features are extracted, including source IP (Internet Protocol Address), packet size, network protocol, and various other characteristics for discerning network service content. Subsequently, we categorize the data into three types of slices: lightweight, hybrid, and heavyweight.

1) Lightweight slices primarily serve network services with small traffic loads, such as HTTP, Wikipedia, Messenger, and other services characterized by low traffic volume and high interaction frequency.

2) Hybrid slices are intended for hybrid network applications. These applications may experience variations in network traffic due to different user service usage. To avoid frequent slice switching that can degrade service quality, such applications are placed within this slice. Examples include Facebook, Instagram, Google, and other hybrid network applications.

3) Heavyweight slices cater to applications requiring high bandwidth and stability, such as Cloudflare, Google Drive, and services demanding substantial network resources over extended periods.

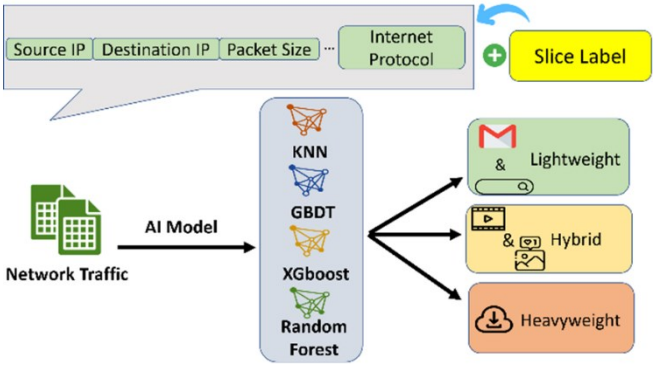

Fig. 2. AI-based traffic classifier.

C. *Slice Resource Configuration Mechanism and Slice Allocation Strategy*

In the face of diverse service demands, the current 5G computational resource allocation mechanism is static, and the available resource pool is limited. To address the issue of resource scarcity leading to network service disruptions, Auto-Scaling technology is crucial.

Common auto-scaling strategies for 5G network slicing are based on metrics such as slice load, traffic usage, and service requirements. Two common auto-scaling strategies are as follows:

1) Dynamic resource auto-scaling: This strategy adjusts resources based on the current slice's demand and resource status. When resources are insufficient, additional resources are requested from the higher-level resource manager, and when resources are excessive, resources are released.

2) Priority-based auto-scaling: Users and network service providers can ensure different levels of QoS based on contractual agreements. When users have higher priority, they receive higher QoS.

In this paper, the dynamic resource auto-scaling strategy is selected. Virtual resources for each network slice are categorized into three classes based on network traffic. These virtual resources are organized in a scalable manner. The paper will develop a dynamic slice resource allocation method that can dynamically allocate resources based on their resource quantity and workload.

The process of dynamic slice resource allocation is as follows: When the User Plane Function (UPF) [17] processes traffic flows exceeding its capacity, it sends a scaling request to the Session Management Function (SMF), requesting horizontal resource scaling. The SMF, through the slice classifier, selects a new network slice for the user. If all network slices are currently overloaded, the UPF notifies the Global Session Manager (GSM) to perform resource vertical scaling [18], such as increasing CPU cores, memory, and bandwidth. For detailed processes, refer to Figure 3 and Algorithm 1.

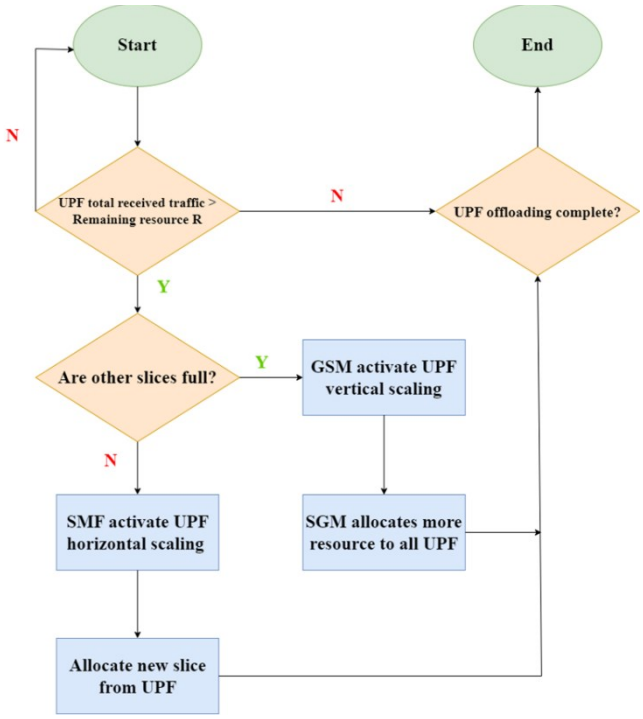

Fig. 3. Allocation strategy flow chart.

Algorithm 1: Slice Resource Allocation Strategy

```
def auto_scaling_pstrategy(P: Packet, S: SliceReousrce):
    M = model
    pkgType = PkgClassify(P, M)
    for s in S:
        if slice == pkgType:
            if enough_resource(P, s):
                allocate_resource(P, s)
        else if not allocate_from_other_slice(P, s, S):
            request_GSM(P)
```

IV. Performance Evaluation and Slicing Management Algorithm

This section provides a detailed overview of the AI classifier experiment and the auto-scaling configuration strategy. Section A discusses data preprocessing and the handling of dataset imbalances. In Section B, we compare the training speed and resource utilization between CPU and GPU. Section C delves into the evaluation of the AI traffic classifier's performance.

A. *Data Preprocessing and Addressing Data Set Imbalance*

The network traffic data for this experiment is obtained from the network traffic dataset collected by Universidad Del Cauca Popayán Colombia. The dataset comprises 141 network applications and includes 50 network traffic features, totaling approximately two million seven hundred thousand records of different traffic, as shown in Figure 4.

Using raw, unprocessed data directly for model training would result in difficulties in achieving convergence in the overall model learning process. To effectively enhance learning outcomes, data preprocessing is necessary before model training. To ensure the model's ability to accurately

classify under different circumstances, eight features that possibly reflect traffic source and time labels, namely "Flow key", "Src IP numeric", "Src IP", "Src Port", "Dst IP", "Dst Port" and "Proto" were removed. Data imbalance can also bias the model's training performance, causing it to lean excessively toward classes with a larger quantity. This, in turn, leads to inaccurate data predictions. To address the issue of dataset imbalance, this project removed data samples with too few instances and employed the Synthetic Minority Oversampling Technique (SMOTE) [19] oversampling technique to address the problems of data scarcity and uneven data distribution.

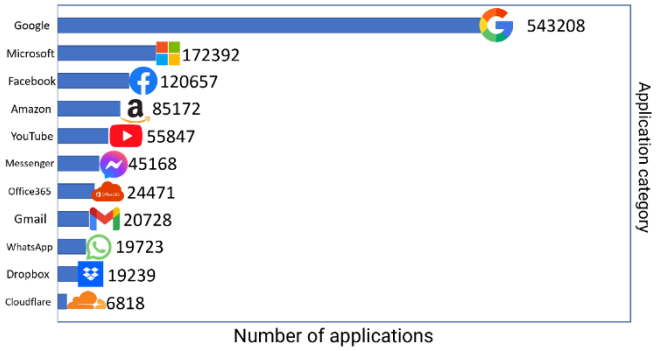

Fig. 4. Data class distribution.

B. Comparing Training Speed Between CPU and GPU

In recent years, the rise of neural network technology has been greatly facilitated by the improved computational capabilities of GPUs. Therefore, this experiment first focuses on testing Random Forest, GBDT, XGBoost [20], and K-Nearest Neighbor (KNN) [21] models. Using 170,000 packet training data as input, the models are employed to classify packets into three categories. To confirm that GPUs do indeed enhance computational efficiency compared to CPUs, this experiment compares the speed differences between the two using the Scikit-learn machine learning framework [22] as the computational foundation.

In the experimental environment, both Scikit-learn-CPU, which runs on CPUs, and Scikit-learn-GPU, which runs on GPUs, were installed. They were each used to train the aforementioned datasets and models. The average training time per epoch (in seconds) is shown in Figure 5. For instance, in the case of Random Forest, the CPU takes 287 seconds per epoch, while the GPU only takes 28 seconds, which is nearly a tenfold difference.

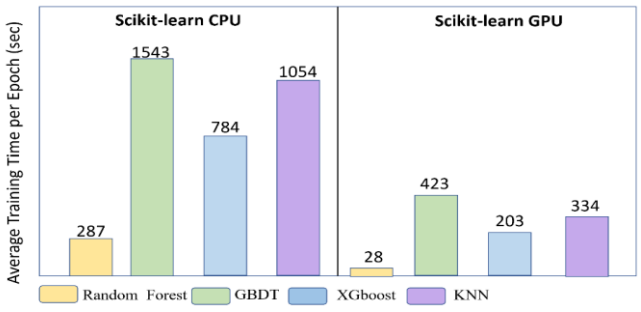

Fig. 5. Comparison of CPU and GPU computing time.

In addition to training speed, GPU-based computations can significantly reduce the burden on the CPU during training. Figure 6 illustrates the CPU and GPU utilization when using Scikit-learn-CPU and Scikit-learn-GPU for computations. When using Scikit-learn-CPU, the CPU utilization reaches as high as 94%, which affects the basic operation of the host machine. However, when using Scikit-learn-GPU, the CPU utilization is only 20%, while the GPU utilization is at 65%. Therefore, the GPU can help alleviate the computational load on the CPU, allowing the host machine to maintain normal operation.

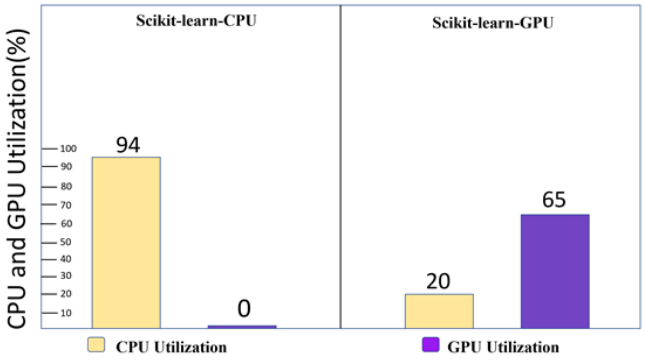

Fig. 6. Comparison of CPU and GPU utilization.

C. Evaluation of AI Traffic Classifier's Performance

The performance of models using Random Forest and GBDT is better than that of XGBoost and KNN, and these algorithms are effective in classifying slices with an impressive accuracy of up to 95.73%, as shown in Table I.

TABLE I
DIFFERENT SPLIT-RATIO ACCURACY COMPARISON

Split-Ratio	KNN	Random-Forest	XGboost	GBDT
80:20	93.99%	95.73%	91.42%	95.76%
70:30	93.77%	95.66%	91.31%	95.14%
60:40	93.01%	95.45%	91.33%	91.49%

V. CONCLUSIONS

This paper introduces supervised learning to learn initial parameters, during the experimental process, it was found that the AI traffic classifier performed best under the training of Random Forest and GBDT models, achieving an accuracy rate of up to 95.73%. In the experimental environment, GPU demonstrates a significant tenfold difference in training speed compared to CPU. Simultaneously, when utilizing Scikit-learn-GPU, it effectively alleviates the computational burden on the host machine. This advantage holds crucial reference value for large-scale data processing and complex model training.

ACKNOWLEDGMENT

This research was supported in part by the National Science and Technology Council, Taiwan, under Grant no. 111-2221-E-008-063-MY3, 111-2221-E-008-062-MY2 and 111-2218-E-415-001-MBK.

REFERENCES

[1] A. A. Barakabitze, A. Ahmad, R. Mijumbi and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy architectures and future challenges", *Comput. Netw.*, vol. 167, Feb. 2020.

[2] D. C. Li, P. H. Chen and L. D. Chou, "GAP4NSH: A Genetic Service Function Chaining with Network Service Header for P4-based Software-Defined Networks", *Journal of Supercomputing*, Mar. 2023.

[3] F.-H. Tseng, Y.-M. Jheng, L.-D. Chou, H.-C. Chao and V. C. Leung, "Link-aware virtual machine placement for cloud services based on service-oriented architecture", *IEEE Trans. Cloud Comput.*, 2017.

[4] D. C. Li, M. R. Maulana and L.-D. Chou, "NNSplit-SOREN: Supporting the Model Implementation of Large Neural Networks in a Programmable Data Plane", *Computer Networks*, vol. 222, Feb. 2023.

[5] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning", *Stata J.*, vol. 20, no. 1, pp. 3-29, Mar. 2020.

[6] Z. Wen, Q. Li, B. He, and B. Cui, "Challenges and opportunities of building fast gbdt systems.," in *IJCAI*, 2021, pp. 4661-4668.

[7] E. Fregnan, T. Baum, F. Palomba and A. Bacchelli, "A survey on software coupling relations and tools", *Information and Software Technology*, vol. 107, pp. 159-178, 2019.

[8] T. Cao et al., "Stochastic optimization for green multimedia services in dense 5G networks", *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 3, pp. 1-23, Sep. 2019.

[9] S. Zhang, "An overview of network slicing for 5G", *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 111-117, Jun. 2019.

[10] P. Krishnan, K. Jain, P. G. Jose, K. Achuthan and R. Buyya, "SDN enabled QoE and security framework for multimedia applications in 5G networks", *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 17, no. 2, pp. 39:1-39:29, 2021.

[11] S. Saibharath, S. Mishra, and C. Hota. "Joint QoS and energy-efficient resource allocation and scheduling in 5G Network Slicing." *Computer Communications* 202 (2023): 110-123.

[12] C. C. Liu, and L. D. Chou. "5G/B5G Network Slice Management via Staged Reinforcement Learning." *IEEE Access*, vol. 11, pp. 72272 - 72280, 2023.

[13] A. Kwan, J. Wong, H.-A. Jacobsen and V. Muthusamy, "Hyscale: Hybrid and network scaling of dockerized microservices in cloud data centres", *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, pp. 80-90, 2019.

[14] S. K. Singh, M. M. Salim, J. Cha, Y. Pan and J. H. Park, "Machine learning-based network sub-slicing framework in a sustainable 5G environment", *Sustainability*, vol. 12, no. 15, pp. 6250, 2020.

[15] D. Wu, Z. Zhang, S. Wu, J. Yang and R. Wang, "Biologically inspired resource allocation for network slices in 5G-enabled Internet of Things", *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9266-9279, Dec. 2019.

[16] K. C. Chiu, C. C. Liu and L. D. Chou, "CAPC: Packet-based network service classifier with convolutional autoencoder", *IEEE Access*, vol. 8, pp. 218081-218094, 2020.

[17] I. Leyva-Pupo and C. Cervello-Pastor, "Efficient solutions to the placement and chaining problem of user plane functions in 5g networks", *Journal of Network and Computer Applications*, 2022.

[18] R. J. Patz, and Y. Lihua, "Methods and models for vertical scaling." *Linking and aligning scores and scales* (2007): 253-272.

[19] A. Fernandez, S. Garcia, F. Herrera and N. V. Chawla, "SMOTE for learning from imbalanced data: Progress and challenges marking the 15-year anniversary", *J. Artif. Intell. Res.*, vol. 61, pp. 863-905, Apr. 2018.

[20] H. Jiang, Z. He, G. Ye, and H. Zhang, "Network intrusion detection based on PSO-XGBoost model," *IEEE Access*, vol. 8, pp. 58392–58401, 2020.

[21] I. Triguero, D. García-Gil, J. Maillo, J. Luengo, S. García, and F. I. Triguero, D. García-Gil, J. Maillo, J. Luengo, S. García and F. Herrera, " Transforming big data into smart data: An insight on the use of the k -nearest neighbors algorithm to obtain quality data ", *Wiley Interdiscip. Rev. Data Mining Knowl. Discovery*, vol. 9, no. 2, 2019.

[22] F. Pedregosa et al., "Scikit-learn: Machine learning in python", *J. Mach. Learn. Res.*, vol. 12, pp. 2825-30, 2011.