

Quality of service management in telecommunication network using machine learning technique

Zhunossov Ayan¹, Baikenov Alimzhan¹, Manankova Olga², Zheltayev Timur¹, Ziyekenov Toktalyk¹

¹Department of Telecommunications and Space Engineering, Faculty of Telecommunications and Innovation Technologies, Almaty University of Power Engineering and Telecommunications, Almaty, Kazakhstan

²Department of Cybersecurity, Faculty of Computer Sciences and Cybersecurity, International University of Information Technology, Almaty, Kazakhstan

Article Info

Article history:

Received Oct 20, 2022

Revised Jun 18, 2023

Accepted Aug 2, 2023

Keywords:

Machine learning

Monitoring

PADT

PPPoE

Quality of service management

Virtual local area network

ABSTRACT

Designing and implementing a fail-safe, real-time telecommunications network is complex. In modern networks, traditional quality of service (QoS) methods for monitoring and analyzing data have some problems, such as accuracy and efficient processing of big data in real time. To solve this problem, should use an appropriate intelligent crash classification system to detect and diagnose runtime errors. The article proposes to use a comprehensive fault detection system that includes QoS and machine learning technologies using information about the state of a point-to-point protocol over ethernet (PPPoE) session on PPPoE active discovery termination (PADT) virtual local area network (VLAN) routes. This intelligent system is built using the machine learning method and is independent of the main real-time system. Demonstrated the operation of seven machine learning algorithms and presented the results of training and fault detection. Based on the received information about the state of the PPPoE session, the PADT type allows you to control the behavior of the real-time system.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Manankova Olga

Department of Cybersecurity, Faculty of Computer Sciences and Cybersecurity

International University of Information Technology

Almaty, Kazakhstan

Email: olga.manank@gmail.com

1. INTRODUCTION

In modern communication networks, effective troubleshooting plays a pivotal role in ensuring optimal service quality for end users. Internet service providers (ISPs) grapple with the challenge of delivering high-quality services to customers while upholding network stability and dependability. Detecting and rectifying network failures is a significant hurdle faced by ISPs [1]. Consequently, a robust fault detection system becomes indispensable to quality of service (QoS) and uphold customer satisfaction.

A notable issue in network communication pertains to the occurrence of session failures within point-to-point protocol over ethernet (PPPoE) sessions [2], [3]. Detecting failures in PPPoE sessions poses challenges since this operates at the data link layer, responsible for transmitting and receiving data packets between nodes. However, this layer lacks mechanisms to verify packet delivery to their intended destinations. In response, this article proposes an all-encompassing fault detection system that melds QoS and machine learning technologies. This system scrutinizes the status of PPPoE sessions across virtual local area network (VLAN) routes, utilizing real-time PPPoE active discovery termination (PADT) data from network devices. Its role is to detect, diagnose, and address network failures, thus providing actionable insights to network administrators.

Existing research on detecting network communication failures has chiefly centered on QoS mechanisms that prioritize traffic based on its importance and utilization [4], [5]. More recently, machine-

learning algorithms have emerged as tools for detecting and prognosticating network failures. These algorithms learn from historical traffic data, proficiently predicting future traffic patterns to allocate resources effectively and shape traffic. Another crucial facet of QoS management pertains to fault detection and diagnosis. Machine learning algorithms can be trained to discern and categorize failures according to network performance metrics and logs, facilitating proactive issue identification and resolution. Moreover, machine learning methods can be utilized to tackle network optimization hurdles, such as enhancing route efficiency, distributing workloads evenly, and strategizing network expansion. By dissecting network performance data, machine-learning models can spot optimization opportunities and propose network alterations.

Alqudah and Yaseen [6] conducted a comparison between the effectiveness of machine learning and statistical methods based on software fault prediction models. The findings of this empirical analysis indicated that machine learning outperformed classical statistical models in predicting whether a class/module is prone to errors or not. The use of machine learning techniques has proven to be more effective in identifying instances where software might encounter faults or errors in the future. This suggests that the advanced algorithms and patterns learned by machine learning models enable them to achieve a greater accuracy and reliability in foreseeing these issues, ultimately contributing to enhanced software quality and performance management.

In the domain of analyzing data network traffic, machine learning plays a significant role in achieving multiple objectives related to the assessment of network operational performance, management, and security. Machine learning techniques contribute to enhancing various aspects of network analysis [7]. By integrating internet protocol (IP) network methodologies with data mining techniques, it becomes possible to conduct a thorough and qualitative analysis of current IP networks. This synergy between disciplines facilitates the identification and revelation of regions within the network that experience congestion. Data mining, which involves extracting valuable insights from large datasets, when combined with established IP network methodologies, empowers organizations to gain a deeper understanding of their network's performance [8].

Mobile technologies are increasingly using machine learning methods to optimize configurations and improve various aspects of mobile communication, computing and resource allocation [9], [10] and solving network design problems [11]. In software-defined networking (SDN) software-defined networks, machine learning can predict and deliver customized QoS based on customer requirements. This approach reduces the load on network equipment by minimizing signal traffic flow, identifying and categorizing conflicting data flows [12]–[15]. A separate study [16] presents several machine learning algorithms such as decision trees (DT), support vector machines (SVM), extremely fast decision trees (EFDT), and hybrid decision trees (DT-SVM), demonstrating their use to solve the above problems in SDN networks [17], [18]. These algorithms make it possible to quickly detect and classify conflicts that arise in SDN networks, and also help prevent distributed denial of service (DDoS) attacks [19]–[21]. As SDN becomes a foundational element of 5G mobile communications systems, the focus is shifting towards reducing costs and improving the quality of service. Through the use of machine learning mechanisms in OpenFlow, SDN provides efficient resource allocation in operator networks, taking into account mobility and reducing over-provisioning of resources [22].

The fundamental essence of machine learning in QoS management lies in its capacity to autonomously and adaptively optimize service delivery based on data-driven insights. Through the utilization of machine learning algorithms, systems can learn from historical data and real-time observations to dynamically adjust resource allocation, prioritize traffic, and predict potential service disruptions. This empowers networks to enhance QoS by responding to changing conditions in a proactive and efficient manner, ultimately leading to improved user experiences and network performance [23].

In light of this analysis, machine-learning methodologies assume increasing relevance as network monitoring grows intricate with rising traffic demands. By deploying these methods across diverse scientific and technological domains, swifter and more effective solutions emerge. This article presents a machine learning methodology for identifying bottlenecks in telecommunications networks while monitoring service quality. This innovative approach employs router data on PADT values, indirectly indicating bottleneck presence [24]–[27]. This machine learning-based method scrutinizes PPPoE session status across VLAN routes, gathering router data at specified intervals for analysis on a server. Resultant diagrams provide guidance for necessary service diagnostics. The novelty of this approach lies in leveraging PADT information as a criterion for evaluating service quality.

Researchers have also proposed various methods for detecting PPPoE session failures, including the use of Ping messages, simple network management protocol (SNMP), and syslog messages. However, these methods have limitations, such as the inability to detect failures caused by a sudden increase in network traffic [28]–[30]. Major contributors in the field have explored various error detection techniques, including anomaly detection [31]–[33], data mining and machine learning. These studies highlighted the importance of real-time monitoring, accurate data collection, and efficient analysis methods for fault detection. Despite the progress made in fault detection systems, there are still areas for improvement. One of the outstanding problems is the detection of failures in PPPoE networks, which are widely used by ISPs. PPPoE networks involve the encapsulation of PPP frames in Ethernet frames and rely on the exchange of control packets to establish and

maintain sessions between client and server. Existing failure detection systems do not consider PPPoE session state and PADT packet types in VLAN routes. Therefore, there is a need for an efficient fault detection system that considers these factors to improve quality.

Compared with our previous works [34], [35], the main contribution of this article is that a system was created to solve the problems of improving the quality of services by training a neural network in order to automatically find and classify the reasons for the unsatisfactory quality of services provided. The system uses advanced data analysis techniques to identify patterns and anomalies in network traffic and provides network administrators with actionable insights. Thus, unlike [34], [35] we expand the original troubleshooting system. Another difference from [34], [35], is the results of neural network training on previously collected data.

The rest of the manuscript is organized as follows. Section 2 presents the methodology for the proposed fault detection system, including architecture and components. Section 3 describes the results of the experiments. Section 4 concludes the article and discusses future research directions in this area.

2. METHOD

PPPoE sessions can fail due to various factors, including network congestion, hardware failures, software bugs, misconfiguration, and security attacks. When a PPPoE session failure occurs, the PPPoE client sends a PADT packet to the PPPoE server, indicating that the session should be terminated. This PADT packet contains information about the reason for the session termination, which can include authentication failures, idle timeouts, or network errors. By analyzing the PADT packets on VLAN routes, our proposed fault detection system can identify and diagnose the causes of PPPoE session failures, enabling network administrators to take appropriate action to address the issues promptly [36].

To grasp the evolving trend of packet drops in the given direction, it's essential to compute the packet drop ratio employing the (1):

$$K = (PADT2 - PADT1) / S, \quad (1)$$

where: K is the rate of dropped packets,

PADT2 - value of sent PADT packets during polling *t2*,

PADT1 - value of sent PADT packets during polling *t1*,

S is the arithmetic mean of the total number of sessions in the time interval (*t2-t1*).

The drop rate of packets exposes the proportion of unsuccessful authorization attempts [37], [38]. To enhance the acquisition of PADT packet statistics and pinpoint root causes, the utilization of machine learning methods is suggested. To achieve this, a Python script was crafted, facilitating data collection from routers at designated time intervals and transmission to a server. At the server, the amassed data from tables is translated into graphical representations.

To tackle this challenge using machine learning, neural networks were employed as the algorithm of choice. Neural networks enable the comprehension of intricate patterns within data and the classification of PADT packets into either standard or anomalous categories, based on the derived characteristics. The features by which PADT packets are classified by fault class include:

- The timestamp of the PADT packet is used to determine patterns over time.
- The PADT packet size is used to detect abnormal packet size.
- The source and destination IP addresses are used to identify the source and destination of PADT packet.
- The protocol type is used to distinguish between different types of packets.
- The PADT packet payload data reveals patterns that indicate a QoS issue.

For ease of visualization, the ratio is quantified as a percentage. After collecting and processing the information is placed in tables for further work with machine learning scripts. It is proposed to study the neural network training algorithms stochastic gradient descent (SGD), root mean square propagation (RMSprop), adaptive moment estimation (Adam), Adadelta, adaptive gradient algorithm (Adagrad), Adamax, nesterov-accelerated adaptive moment estimation (Nadam) for the possibility of fault detection based on PADT packet data collected over 2 weeks for seven specialized services, including failures according to the architecture shown in Figure 1. Crucial pieces of code using various optimizers (neural network training algorithms) are shown in Figure 2.

For example, the time stamp of a PADT (PPP active terminate) packet on the network can help identify network latency-related network congestion problems, such as delays caused by long routes or bandwidth limitations. If the PADT packet sizes are inconsistent or exceed the maximum size allowed by the PPP protocol, this is a hardware or software failure in the network equipment. If the timestamps of the PADT packets indicate unexpected behavior, such as sudden bursts of traffic, this indicates a security risk or an attack on the network.

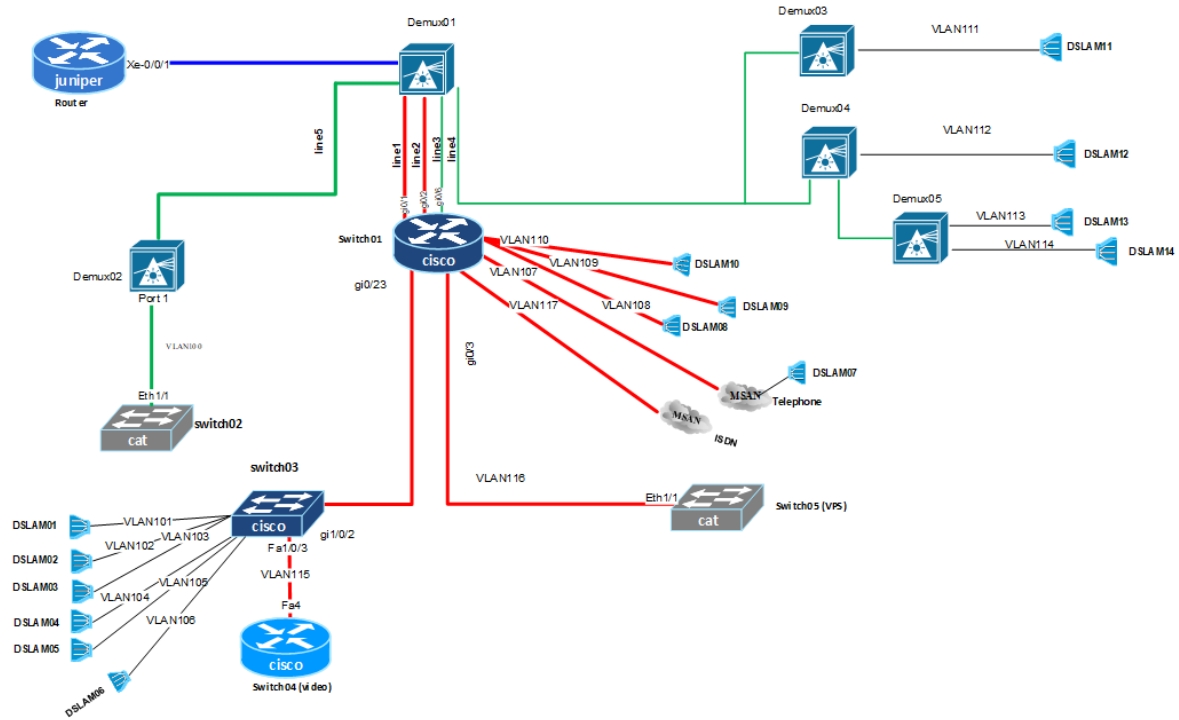


Figure 1. PPPoE network architecture

```
import numpy as np
import matplotlib.pyplot as plt
# Make numpy values easier to read.
np.set_printoptions(precision=3, suppress=True)

import tensorflow as tf
from tensorflow.keras import layers
data = pd.read_csv('data2.csv')
test = pd.read_csv('test.csv')
print(data.shape)
data_feat = data.copy()
data_labels = data_feat.pop('PADI')
test_labels = test.pop('PADI')
data_feat = np.array(data_feat)
test_feat = np.array(test)
normalize = layers.Normalization()
normalize.adapt(data_feat)
model = tf.keras.Sequential([
    normalize,
    layers.Dense(64),
    layers.Dense(1)
])
f = open('PADI_2_K.txt', 'w')
optimizers = ('sgd', 'RMSprop', 'Adam', 'Adadelta', 'Adagrad', 'Adamax', 'Nadam')
losses = (tf.losses.MeanAbsoluteError(), tf.losses.MeanSquaredError())
losses_name = ('MeanAbsoluteError', 'MeanSquaredError')
fig = plt.figure(figsize=(16, 10), dpi=60, layout='constrained')
ax = fig.add_subplot(111)
for i in range(len(losses)):
    for opt in optimizers:
        print(losses_name[i], opt)
        model.compile(loss = losses[i],
                      optimizer = opt,
                      metrics = ['accuracy'])
        history = model.fit(data_feat, data_labels, epochs = 16)
        plt.plot(history.epoch, history.history["loss"], label=f'{losses_name[i]}_{opt}')
        plt.title('Graph Train')
        plt.xlabel('Epochs')
        plt.ylabel('Loss')
        plt.legend(bbox_to_anchor=( 1.05 , 1 ))
        eva = model.evaluate(test, test_labels, verbose=2)
        print(losses_name[i], opt, file = f)
        print(eva, file = f)
        model.save(f'model2_{losses_name[i]}_{opt}')
plt.savefig('graph')
f.close()
```

Figure 2. Pieces of code using TensorFlow optimizers

3. RESULTS AND DISCUSSION

Using the freely available resource TensorFlow with an open software library for machine learning developed by Google. A script was written to solve the above problems and train a neural network in order to automatically find and classify the reasons for the unsatisfactory quality of services provided. The system uses several different modules (optimizer), such as SGD, RMSprop, Adam, Adadelta, Adagrad, Adamax, Nadam for a script written in the python programming language.

SGD is a simple and popular optimization algorithm for training neural networks. It updates the network's weights based on the gradients of a randomly selected subset of the training data. SGD can be computationally efficient but may require tuning of the learning rate and batch size.

RMSprop is an adaptive learning rate optimization algorithm. It uses a moving average of the squared gradients to normalize the learning rate. RMSprop can be effective in avoiding the problem of the learning rate being too large or too small.

Adam is an adaptive learning rate optimization algorithm that combines the advantages of RMSprop and momentum optimization. It adapts the learning rate based on the first and second moments of the gradients. Adam can be computationally efficient and effective in handling sparse gradients.

Adadelta is an adaptive learning rate optimization algorithm that uses a combination of running average gradients and running average updates. It adapts the learning rate based on the ratio of these two running averages. Adadelta can be effective in handling noisy gradients and avoiding the need for manual tuning of the learning rate.

Adagrad is an adaptive learning rate optimization algorithm that adapts the learning rate based on the history of the gradients. It uses a diagonal matrix of the squared gradients to normalize the learning rate. Adagrad can be effective in handling sparse gradients but may become too aggressive in reducing the learning rate.

Adamax is a variant of the Adam algorithm that uses the infinity norm to normalize the gradient updates. It can be more robust to outliers than the standard Adam algorithm. Adamax can be effective in handling large-scale and high-dimensional problems.

Nadam is a variant of the Adam algorithm that includes Nesterov momentum optimization. It uses the gradient computed with the updated weights to update the weights instead of the current gradient. Nadam can be effective in reducing oscillations and avoiding local minima.

During the initial iterations, the error of different modules varied from 28% to 71% as shown in Figure 3. When predicting events, the modules looked at previous results and gave their predictions for the possible cause of the current problem. Further, the results are entered into the database and the network is trained again. In Table 1 is showed and in Figure 4, the graph shows the predictions of different modules for 7 directions.

Layer (type)	Output Shape	Param #
normalization (Normalization)	(None, 40)	81
dense (Dense)	(None, 64)	2624
dense_1 (Dense)	(None, 1)	65

Total params: 2,770		
Trainable params: 2,689		
Non-trainable params: 81		

None		
Epoch 1/16		
1/1 [-----]	- 1s 595ms/step	- loss: 0.7085 - accuracy: 0.4000
Epoch 2/16		
1/1 [-----]	- 0s 3ms/step	- loss: 0.6544 - accuracy: 0.4000
Epoch 3/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.6233 - accuracy: 0.4000
Epoch 4/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.5932 - accuracy: 0.4000
Epoch 5/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.5636 - accuracy: 0.4000
Epoch 6/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.5346 - accuracy: 0.4000
Epoch 7/16		
1/1 [-----]	- 0s 3ms/step	- loss: 0.5062 - accuracy: 0.4000
Epoch 8/16		
1/1 [-----]	- 0s 3ms/step	- loss: 0.4785 - accuracy: 0.4000
Epoch 9/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.4515 - accuracy: 0.4000
Epoch 10/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.4252 - accuracy: 0.4000
Epoch 11/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.3996 - accuracy: 0.4000
Epoch 12/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.3746 - accuracy: 0.5000
Epoch 13/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.3501 - accuracy: 0.5000
Epoch 14/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.3262 - accuracy: 0.5000
Epoch 15/16		
1/1 [-----]	- 0s 2ms/step	- loss: 0.3028 - accuracy: 0.5000
Epoch 16/16		
1/1 [-----]	- 0s 3ms/step	- loss: 0.2799 - accuracy: 0.5000

Figure 3. The error of different modules

Table 1. Result of modules

Loops	sgd	RMSprop	Adam	Adadelta	Adagrad	Adamax	Nadam
Loop 1	1.4521005	1.4121289	1.6095896	2.0218556	1.9068718	1.6437837	1.6557081
Loop 2	1.0359179	1.9709009	2.0210886	2.0213861	1.878932	1.9029763	1.9416497
Loop 3	1.2487546	1.6180602	1.9484773	1.8841248	1.8703012	1.9039562	1.8330187
Loop 4	1.4535244	0.8115978	0.51684785	1.9890528	1.4873731	0.51669	0.6559415
Loop 5	0.1594775	0.4738749	0.58863926	1.8992579	1.5753918	0.64257264	0.7714031
Loop 6	2.106946	2.1208556	2.3559039	1.9296553	2.1502793	2.372746	2.288053
Loop 7	1.9144098	0.2673669	0.24379101	0.0901962	0.0580795	0.0909132	0.1162455

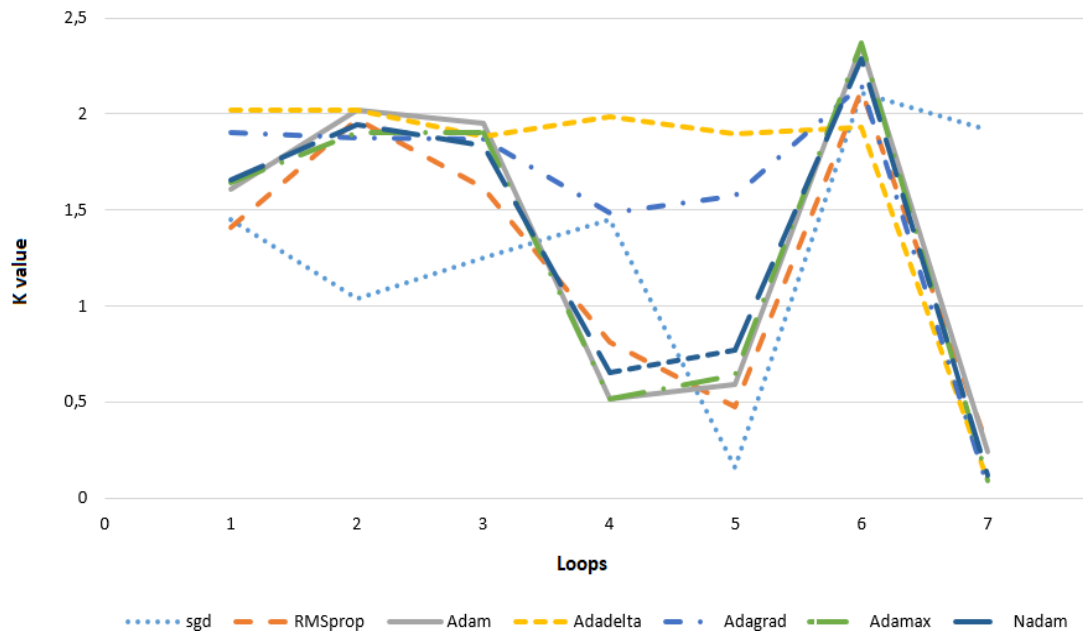


Figure 4. Schedule for identifying faults according to the algorithms used

A value close to 0.5 suggests a potential issue with wireless communication. This could indicate that the device is struggling to maintain a reliable connection to the network due to various factors. Signal interference from nearby electronic devices or physical obstacles could be hindering the device's ability to establish a strong and consistent wireless link.

A value of 1 indicates a possible problem with the physical interface of the device. This numeric reading might be indicative of hardware-related issues that affect the device's ability to interact seamlessly with other devices or networks. Damaged connectors, ports, or cables could be causing connectivity problems, making it difficult for the device to establish proper communication channels. Such issues might result in intermittent connections; data transfer errors, or even complete device unresponsiveness, highlighting the importance of inspecting and addressing potential physical interface concerns.

A value of 1.5 implies a potential problem related to memory or processor load on the device. This intermediate reading signifies that the device's performance might be compromised due to resource limitations. Inadequate memory availability could lead to slower response times and application crashes, as the device struggles to manage its running processes efficiently. Similarly, an excessive processor load could result in laggy performance, delayed user interactions, and an overall diminished user experience.

A value of 2 suggests a potential bandwidth problem. This reading serves as an indicator that the device's data transfer capabilities are under strain, likely due to the demands placed on its network connection. High data traffic or large file transfers could be saturating the available bandwidth, leading to slower data transfer rates and increased latency.

The lower the K value, the better the algorithm performs in terms of minimizing the absolute differences between predicted and actual values. From the provided K values, the algorithm with the lowest K for the given dataset appears to be Adagrad, as it has the lowest K value among the listed algorithms for most of the data points in Figure 4. These diagnostic values provide valuable insights into different aspects of a device's functionality and performance. By leveraging these indicators, users and technicians can work together to troubleshoot and optimize the device's performance, ensuring a more seamless and reliable experience.

4. CONCLUSION

In this work, a machine learning technique was used to optimize the tracking of the quality of services provided. The algorithm is based on the method of detecting failures on the operator's route using information from PPPoE sessions in a specific VLAN. Seven algorithms used to predict events in a neural network based on machine learning were investigated. Signs by which a malfunction is detected are determined. The weight coefficients of faults are found, by which the nature of the failure is determined. Based on the test results of the proposed machine learning methods, it is worth highlighting RMSprop, Adam, Adadelta and Nadam. They can be efficient optimization algorithms for training neural networks to classify PADT bursts as normal or abnormal.

In the future, there are ambitious plans to expand the range of fault tracking parameters, a strategic move aimed at enhancing the precision and scope of diagnostic processes. This evolution is poised to revolutionize the field of fault detection and troubleshooting by delving deeper into the intricate web of device behaviors and potential issues. By incorporating an array of novel parameters, the diagnostic framework will gain the ability to detect and isolate problems that were previously elusive, thereby bolstering the overall efficacy of the fault tracking system.




REFERENCES

- [1] E. C. Molero, S. Vissicchio, and L. Vanbever, "FAST in-network GraY failure detection for ISPs," in *SIGCOMM 2022 - Proceedings of the ACM SIGCOMM 2022 Conference*, Aug. 2022, pp. 677–692, doi: 10.1145/3544216.3544242.
- [2] T. Bhumrawi, C. Netramai, K. Kaemarungsi, and K. Limtanyakul, "Impact of multi-services over service provider's local network measured by passive and active measurements techniques," in *Advances in Intelligent Systems and Computing*, vol. 209, 2013, pp. 41–50, doi: 10.1007/978-3-642-37371-8_8.
- [3] F. Liu, T. Xie, Y. Feng, and D. Feng, "On the security of PPPoE network," *Security and Communication Networks*, vol. 5, no. 10, pp. 1159–1168, Feb. 2012, doi: 10.1002/sec.512.
- [4] T. A. Assegie and H. D. Bizuneh, "Improving network performance with an integrated priority queue and weighted fair queue scheduling," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 19, no. 1, pp. 241–247, Jul. 2020, doi: 10.11591/ijeecs.v19.i1.pp241-247.
- [5] B. Ramasamy and G. F. Sudha, "Instantaneous channel characteristics and progression factor based collaborative routing," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 28, no. 2, pp. 918–925, Nov. 2022, doi: 10.11591/ijeecs.v28.i2.pp918-925.
- [6] N. Alqudah and Q. Yaseen, "Machine learning for traffic analysis: A review," *Procedia Computer Science*, vol. 170, pp. 911–916, 2020, doi: 10.1016/j.procs.2020.03.111.
- [7] A. S. Jaradat, M. M. Barhoush, and R. B. Easa, "Network intrusion detection system: Machine learning approach," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 25, no. 2, pp. 1151–1158, Feb. 2022, doi: 10.11591/ijeecs.v25.i2.pp1151-1158.
- [8] A. Nacef, A. Kaci, Y. Aklouf, and D. L. C. Dutra, "Machine learning based fast self optimized and life cycle management network," *Computer Networks*, vol. 209, p. 108895, May 2022, doi: 10.1016/j.comnet.2022.108895.
- [9] S. K. Pandey, R. B. Mishra, and A. K. Tripathi, "Machine learning based methods for software fault prediction: a survey," *Expert Systems with Applications*, vol. 172, p. 114595, Jun. 2021, doi: 10.1016/j.eswa.2021.114595.
- [10] S. Suthaharan, "Big data classification: problems and challenges in network intrusion prediction with machine learning," *Performance Evaluation Review*, vol. 41, no. 4, pp. 70–73, Apr. 2014, doi: 10.1145/2627534.2627557.
- [11] Z. A. Khan and A. Samad, "A study of machine learning in wireless sensor network," *International Journal of Computer Networks And Applications*, vol. 4, no. 4, Aug. 2017, doi: 10.22247/ijcna/2017/49122.
- [12] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network traffic classification techniques and comparative analysis using machine learning algorithms," in *2016 2nd IEEE International Conference on Computer and Communications, ICCCC 2016 - Proceedings*, Oct. 2017, pp. 2451–2455, doi: 10.1109/CompComm.2016.7925139.
- [13] M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine learning at the network edge: a survey," *ACM Computing Surveys*, vol. 54, no. 8, pp. 1–37, Oct. 2022, doi: 10.1145/3469029.
- [14] V. Labayen, E. Magaña, D. Morató, and M. Izal, "Online classification of user activities using machine learning on network traffic," *Computer Networks*, vol. 181, p. 107557, Nov. 2020, doi: 10.1016/j.comnet.2020.107557.
- [15] O. Nassef, W. Sun, H. Purmehdi, M. Tatipamula, and T. Mahmoodi, "A survey: distributed machine learning for 5G and beyond," *Computer Networks*, vol. 207, p. 108820, Apr. 2022, doi: 10.1016/j.comnet.2022.108820.
- [16] R. A. Rahman, S. Masrom, N. H. A. Samad, R. M. Daud, and E. Mutia, "Machine learning prediction of video-based learning with technology acceptance model," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 29, no. 3, pp. 1560–1566, Mar. 2023, doi: 10.11591/ijeecs.v29.i3.pp1560-1566.
- [17] Y. Zhao, B. Yan, D. Liu, Y. He, D. Wang, and J. Zhang, "SOON: self-optimizing optical networks with machine learning," *Optics Express*, vol. 26, no. 22, p. 28713, Oct. 2018, doi: 10.1364/oe.26.028713.
- [18] W. S. Saif, M. A. Esmail, A. M. Ragheb, T. A. Alshawi, and S. A. Alshebeili, "Machine learning techniques for optical performance monitoring and modulation format identification: A survey," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 4, pp. 2839–2882, 2020, doi: 10.1109/COMST.2020.3018494.
- [19] Z. Xiong and N. Zilberman, "Do switches dream of machine learning?: Toward in-network classification," in *HotNets 2019 - Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, Nov. 2019, pp. 25–33, doi: 10.1145/3365609.3365864.
- [20] M. H. H. Khairi et al., "Detection and classification of conflict flows in SDN using machine learning algorithms," *IEEE Access*, vol. 9, pp. 76024–76037, 2021, doi: 10.1109/ACCESS.2021.3081629.
- [21] M. Reza, M. Javad, S. Raouf, and R. Javidan, "Network traffic classification using machine learning techniques over software defined networks," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 7, 2017, doi: 10.14569/ijacsa.2017.080729.




- [22] S. V. Mahadevkar *et al.*, "A review on machine learning styles in computer vision - techniques and future directions," *IEEE Access*, vol. 10, pp. 107293–107329, 2022, doi: 10.1109/ACCESS.2022.3209825.
- [23] A. B. Dehkordi, M. R. Soltanaghaei, and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," *Journal of Supercomputing*, vol. 77, no. 3, pp. 2383–2415, Jun. 2021, doi: 10.1007/s11227-020-03323-w.
- [24] K. M. Sudar, M. Beulah, P. Deepalakshmi, P. Nagaraj, and P. Chinnasamy, "Detection of distributed denial of service attacks in SDN using machine learning techniques," Jan. 2021, doi: 10.1109/ICCCI50826.2021.9402517.
- [25] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: datasets and comparative study," *Computer Networks*, vol. 188, p. 107840, Apr. 2021, doi: 10.1016/j.comnet.2021.107840.
- [26] M. Bagaa, D. L. C. Dutra, T. Taleb, and K. Samdanis, "On SDN-driven network optimization and QoS aware routing using multiple paths," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4700–4714, Jul. 2020, doi: 10.1109/TWC.2020.2986408.
- [27] A. I. Khan and S. Al-Habsi, "Machine learning in computer vision," *Procedia Computer Science*, vol. 167, pp. 1444–1451, 2020, doi: 10.1016/j.procs.2020.03.355.
- [28] T. Z. Teshabaev, M. Z. Yakubova, and O. A. Manankova, "Analysis, research and simulation of a multiservice network based on the packet tracer software package to determine the value of delays to increasing value size of ICMP packet," *2020 International Conference on Information Science and Communications Technologies (ICISCT)*, Nov. 2020, doi: 10.1109/ICISCT50599.2020.9351479.
- [29] M. Z. Yakubova, O. A. Manankova, K. A. Tashev, and G. S. Sadikova, "Methodology of the determining for pearson's criterion based on researching the value of delays in the transmitting of information over a multiservice network," *2020 International Conference on Information Science and Communications Technologies (ICISCT)*, Nov. 2020, doi: 10.1109/ICISCT50599.2020.9351419.
- [30] O. A. Manankova, M. Z. Yakubova, and A. S. Baikenov, "Cryptanalysis the SHA-256 hash function using rainbow tables," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 930–944, Dec. 2022, doi: 10.52549/ijeei.v10i4.4247.
- [31] K. Fotiadou, T. H. Velivassaki, A. Voulkidis, D. Skias, S. Tsekeridou, and T. Zahariadis, "Network traffic anomaly detection via deep learning," *Information (Switzerland)*, vol. 12, no. 5, p. 215, May 2021, doi: 10.3390/info12050215.
- [32] L. K. Lok, V. A. Hameed, and M. E. Rana, "Hybrid machine learning approach for anomaly detection," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 27, no. 2, pp. 1016–1024, Aug. 2022, doi: 10.11591/ijeecs.v27.i2.pp1016-1024.
- [33] P. P. Ioulianiou and V. G. Vassilakis, "Denial-of-service attacks and countermeasures in the RPL-based internet of things," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11980 LNCS, Springer International Publishing, 2020, pp. 374–390, doi: 10.1007/978-3-030-42048-2_24.
- [34] A. Zhunussov, A. S. Baikenov, and D. Ilieva, "Monitoring the quality of services provided in a telecommunication network by analyzing the statistics of PPPoE packets," *2020 7th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)*, Nov. 2020, doi: 10.1109/EEAE49144.2020.9279089.
- [35] A. Zhunussov, A. Baikenov, T. Zheltayev, T. Serikov, and T. Ziyekenov, "Machine learning technique in QoS management network," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 2, pp. 904–911, 2023.
- [36] L. Mamakos, D. Simone, R. Wheeler, D. Carrel, J. Evarts, and K. Lidl, "A method for transmitting PPP over Ethernet (PPPoE), RFC 2516 (Informational)," *RFC Editor*, Feb. 1999, doi: 10.17487/rfc2516.
- [37] S. Bradner, "Benchmarking terminology for network interconnection devices," *RFC Editor*, Jul. 1991. [Online]. Available: <https://www.ietf.org/rfc/rfc1242.txt>.
- [38] A. Morton, "Round-trip packet loss metrics," *RFC Editor*, Aug. 2012, doi: 10.17487/rfc6673.

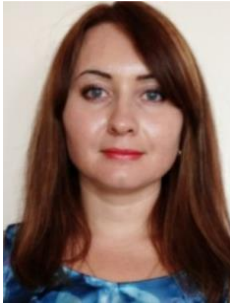
BIOGRAPHIES OF AUTHORS






Zhunussov Ayan    received the academic degree of Master of technical science in specialty radio engineering, electronics and telecommunications from Almaty University of Power Engineering and Telecommunications (AUPET), Almaty, in 2013. He is studying for a Ph.D. in Institute of Telecommunications and Space Engineering from Almaty University of Power Engineering and Telecommunications (AUPET), Almaty. He can be contacted at email: jarmale@mail.ru.






Baikenov Alimjan    is a Candidate of Engineering Sciences, Professor at the Department of Telecommunications and Innovation Technologies. He had overall experience of 41 years. He is familiar with software's like MATLAB, LABVIEW, Multisim, Packet Tracer, EVE NG, Wireshark. 2019 received the Title "Kurmetti baylanysshy", awarded in 2018 Medal "Bilim beru salasyn uzdigi". Now he is actively working with doctoral students in the field of multi-channel communication, Tele traffic theory, system modeling, wireless networks and systems (5G), fiber optic systems, internet of things (IoT), artificial intelligence (AI), information security in 5G and IoT. Expert IAAR NU "Independent Agency for Accreditation and Rating". He can be contacted at email: a.baikenov@aes.kz.






Manankova Olga    is a Ph.D. student at the Almaty University of Power Engineering and Telecommunications after Gumarbek Daukeyev (AUPET). After receiving a master's degree in 2010, she began working at AUPET as a teacher. During this time, under her supervision, more than 25 bachelors graduated, students became winners of the Republican competitions of research and development in the field of IT. Currently works at AUPET as a senior lecturer and is engaged in research in the field of radio engineering, electronics and telecommunications in accordance with the topic of the Ph.D. thesis "Research and creation of information security transmitted over an open communication channel using PBX Asterisk". She can be contacted at email: olga.manank@gmail.com.



Zheltayev Timur    received the academic degree of Master of technical science in speciality Radio engineering, electronics and telecommunications from Almaty University of Power Engineering and Telecommunications (AUPET), Almaty, in 2013. He is studying for a Ph.D. in Institute of Telecommunications and Space Engineering from Almaty University of Power Engineering and Telecommunications (AUPET), Almaty. Currently he is the lead data network engineer in JSC Kazakh telecom, the largest telecommunications company in Kazakhstan. His research interests include machine learning, telecommunications network and monitoring systems in networks. He can be contacted at email: zheltayev@gmail.com.



Ziyekenov Toktalyk    received Bachelor of Information technology from The Central Asian Innovation University, Almaty, in 2009. He received the Master degree of Economical science from Kazakh Economic University named after Ziyekenov Toktalyk, in 2011, Almaty. Currently he is the Company founder and Director of filter factory for mechanical water treatment. His research interests include management, telecommunications network and engineering. He can be contacted at email: toktalyk@mail.ru.