

Machine Learning-based fixed access Network Interface Congestion Prediction

Anees Akhtar¹, Irfan Zafar², Muhammad Ahmed Zaki³, Usman Amjad^{4*} and Muhammad Khurram⁵

^{1,2} Department of Computer Science and Information Technology, University of Engineering and Technology, Peshawar, Pakistan

^{3,4} Department of Computer Science and Information Technology, NED University of Engineering and Technology, Karachi, Pakistan

⁵ Department of Engineering, University of Technology and Applied Sciences-Nizwa, Sultanate of Oman

*Corresponding author: usmanamjad@neduet.edu.pk

Abstract:

Network interface congestion refers to a situation in which network interfaces or communication channels within a network experience high levels of traffic, leading to performance degradation or potential network failures. In modern networking, this issue remains relevant and can manifest in various contexts due to a surge in data traffic, Internet of Things (IoT) devices can generate substantial data, increased use of VPNs for remote work and when multiple users or applications concurrently access and transfer data in and out of the cloud. Efforts to forecast and address network interface bottlenecks are crucial to sustaining the performance and consistency of modern networks. In the PTCL fixed access network infrastructure, network blocking is a serious problem that necessities to be addressed. The main objective of this study is to overcome network congestion problems that occur in PTCL's fixed access network through the use of machine learning techniques. Despite having real-time monitoring tools such as SolarWinds, there remains a dearth of software capable of predicting imminent bottlenecks in the network interface. The primary aim is to build an analytical model that can predict potential blockages within 30 days and facilitate proactive management strategies for better administration. The study classifies the most effective machine learning technique for this purpose and evaluates its performance against current methods. To achieve the goal, a diagnostic dataset was created, leveraging appropriate configurations, software tools, and hardware resources. Data was sourced from SolarWinds, consisting of information from over 23,000 interfaces across 11,000 access network nodes. This data was analyzed in PTCL's Data Warehouse, where a historical aggregated dataset was constructed. Exploratory data analysis was performed using Python libraries, including Matplotlib, NumPy, Pandas, and Seaborn. This investigation sets the foundation for proactive network management and enhanced user experiences within PTCL's fixed access network, addressing the critical essential for advanced predictive tools to combat congestion and enhance network competence.

Keywords: Machine Learning, Bandwidth Utilization, Fixed Access Network, Congestion Prediction

I. INTRODUCTION

The foundation of an increasingly dispersed digital world is network communications [1]. With a single tap on a screen, innumerable businesses and individuals can conduct daily operations and carry out tasks that have become routine thanks to technologies like peer-to-peer networks, blockchain, ad-hoc and mesh networks, cloud/edge/fog computing systems, peer-to-peer networks, 5G networks, space communications and networks, and software and infrastructure as a service [2]. Live streaming is becoming more and more important in the era of new media ecology. Studies focused on a variety of user experiences have developed as a result of the extensive use of live streaming and its enormous user base [3]. The need for high-speed internet services has propelled the telecommunications sector's

impressive global growth. Particularly in Pakistan, the telecom industry has experienced significant growth [4]. Pakistan Telecommunication Company Limited PTCL the country's largest Internet Service Provider (ISP), faces the challenge of meeting the growing demand for high-speed internet while minimizing network congestion [5], [6]. Despite real-time monitoring with SolarWinds, PTCL lacks a proactive solution for precise congestion forecasting. This shortfall hampers effective resource management and capacity planning, impacting customer satisfaction and network performance [7]. The design of PTCL's access network is precisely crafted to give trustworthy, high-speed, and cheap services to its customer base in Pakistan [8]. The entire PTCL network design is shown in Figure 1.

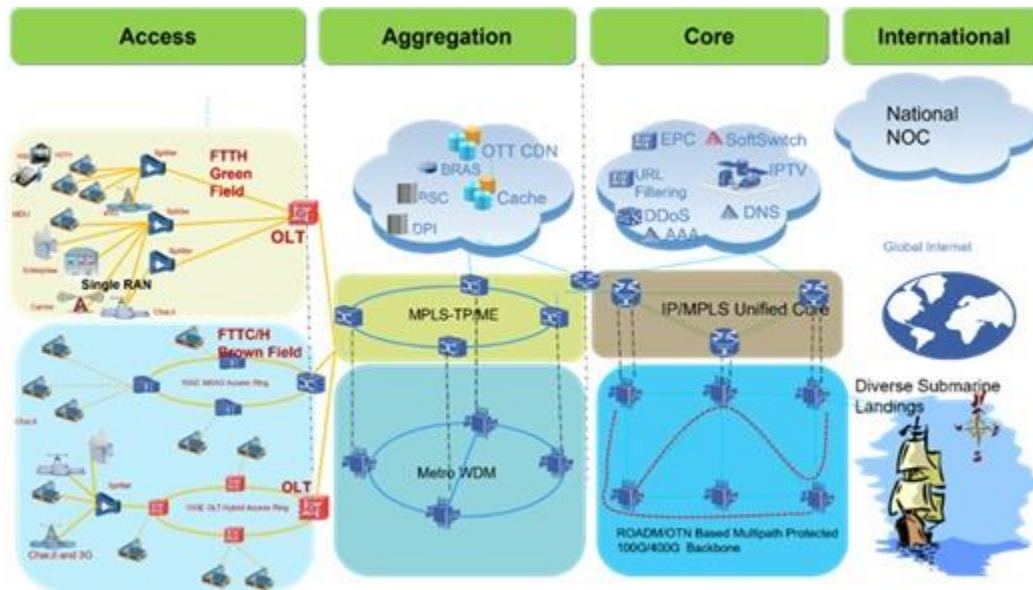


Figure 1: PTCL Network Architecture

When the network's capacity is exceeded, fixed access networks become congested, which lowers performance and creates operational inefficiencies. Overutilization, high traffic volumes, network bottlenecks, and technical issues are a few common causes of congestion in fixed-access networks. Packet loss, slow data rates, and interruptions are just a few of the effects of access network congestion that eventually cause customer resentment and harm the brand's reputation. Furthermore, it can degrade user experience quality, obstructing internet services and corporate activities. Blocking can also threaten network security and make it more problematic for internet service providers to efficiently participate in the market [9]. Many academic works have looked at how to use ML in network circumstances [10]. For example, Miller et al.'s analysis of deep neural networks (DNNs) for network defense against cyberattacks was thorough. Thanks to advancements in processing power, ML has gained a lot of respect from both academia and business for its ability to efficiently extract insights from large datasets [11]. It's known as a transformative technology that empowers systems to make predictions or find solutions on their own. In several fields including image recognition, speech recognition [12]–[15], autonomous systems [16], and communication networks [17], ML has made significant contributions. Its exceptional classification and improvisation skills are well-known; it can even outperform human performance rarely seen otherwise [18].

Supervised, unsupervised, and reinforcement learning (RL) algorithms play a notable role in the context of network frameworks. Due to its adverse effects on whole network performance, network bottleneck is a chief source of worry for network operators. Networks may suffer delays and unproductive resource use if a bottleneck is not meritoriously controlled and accomplished. To maintain network stability, equitable resource use, and a reasonable packet loss ratio, congestion control is essential [19]. Different network environments put their congestion control strategies into practice. Traditional routing systems can't draw lessons from the past when it comes to network oddities like congestion. The network is constantly under stress due to the rise in traffic, which creates problems with resource management

and allocation. As a result, network traffic quality of service (QoS) can suffer, especially as the majority of networks continue to use routing systems from decades ago. Historically, fixed networks used conventional routing techniques to determine the shortest path based on distance vectors or link prices. However, when faced with a high volume of traffic, these policies may compound the problem and worsen performance. Conventional routing techniques in such circumstances frequently replicate prior routing decisions, leading to unmanageable increases in delay and packet loss rates. Congestion-related problems can be lessened via predictive ML models.

ML models like SVM and Gaussian Process Regressions predict optical network OSNR. The most effective routes for incoming traffic are then chosen using these projections. information, QoT estimation might be problematic. Proietti et al. [24] addressed this with an alien wavelength performance monitoring method and ML. The method predicts QoT using an Artificial Neural Network (ANN), attaining a 95% accuracy rate and significantly boosting network performance. An in-band OSNR estimate method using ML algorithms like SVM and Gaussian Process Regressions was proposed by Locatelli et al. [25]. These models were trained using simulated spectral data, and in this case, Gaussian Process Regressions were performed more accurately. To optimize the path selection for incoming traffic, Choudhury et al. [26] developed a hybrid ML model to predict multi-vendor traffic flow and optical efficiency. To forecast path performance, several regression models were used, including Ridge, Gaussian Process, Gradient Boosted Trees, and Random Forest. Gaussian Process and Random Forest produced the lowest Mean Square Error (MSE) of 0.81. Compared to non-ML-based methods, this method improved efficiency and cut expenses by 9%.

This research aims to identify the most effective method for forecasting network interface congestion within PTCL's fixed access network nodes through a thorough examination including a wide range of machine learning techniques and algorithms. This study intends to significantly advance the field of network management by creating a reliable and accurate prediction model. It will also provide ISPs with priceless information and tools to improve their operational effectiveness and network performance. This research diligently prepares the way for a future characterized by proactive network management and improved customer experiences inside PTCL's fixed access network infrastructure by attempting to satisfy the essential requirement for an advanced predictive solution.

II. METHODOLOGY

The study's primary objective was to apply machine learning algorithms practically to predict instances of excessive bandwidth utilization (congestion) using authentic fixed-access network traffic data. To achieve this, the study involved the creation of an Analytical Dataset (ADS) and the acquisition of relevant system configurations, software tools, and hardware resources, all executed with a meticulously planned approach. The collection of interface traffic statistics is made possible by the seamless integration of SolarWinds with all of the interfaces of PTCL access network nodes. Then, every day, this data is methodically transmitted to PTCL's Data Warehouse (DWH) as shown in Figure 2.



Figure 2: Data Sourcing for Predictive Model

The study used the SMOTE (Synthetic Minority Over-sampling Technique) approach to overcome the class imbalance problem as shown in Figure 3. The study then carried out a stratified random sampling from both groups to collect 3 samples each including 90,000 observations.

- A ratio of 70% for the minority class and 30% for the majority class will be maintained in this 1st sample.
- A ratio of 50% for the minority class and 50% for the majority class will be maintained in this 2nd sample.
- A ratio of 30% for the minority class and 70% for the majority class will be maintained in this 3rd sample.

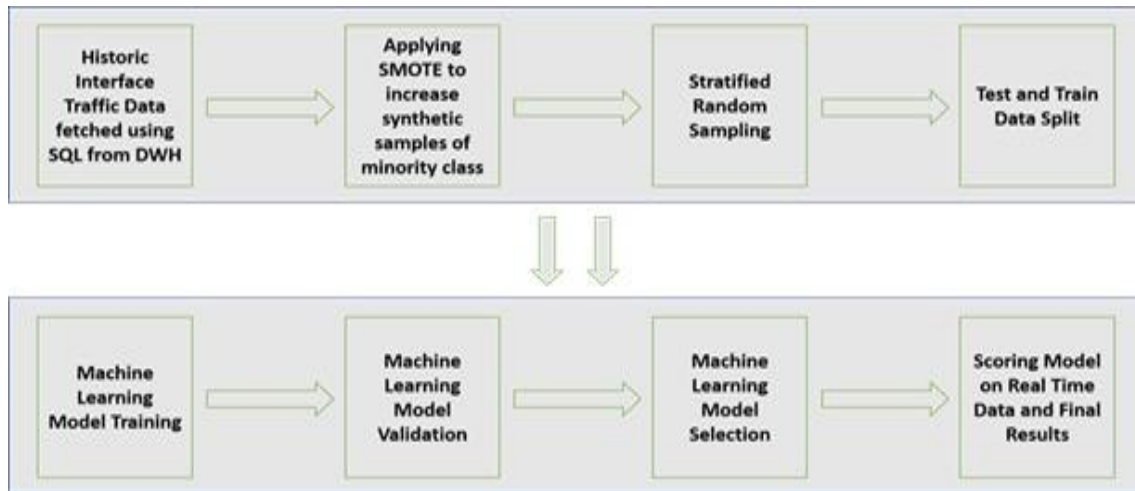


Figure 3: Approach and Methodology for Predictive Model

To divide the data into separate test and training/validation datasets, the study used a stratified test-train split process. To train the model multiple machine learning techniques were used such as: Regression, Random Forest, Support Vector Machine, and XGBoost. The model that performs the best will be picked as the final candidate after these models have been validated using the test dataset. Additionally, perform real-time scoring for validation to make sure that the model is practically applicable in real-world circumstances.

A. Data Sourcing

We collected data for this study from a sizable dataset made up of more than 23,000 interfaces connected to more than 11,000 access network nodes. The sophisticated network management system SolarWinds was used to obtain the data. We extracted the required data from the SolarWinds database and then moved it to our Data Warehouse for further examination.

B. Data Storage and Processing

We created a historical aggregated dataset within PTCL's Data Warehouse to streamline our analytical procedures. The Teradata platform, which is famous for its powerful capabilities in managing and processing large information effectively, is used to build this data warehouse.

C. Analytical Dataset (ADS) Preparation for Predictive Modeling

To get the Analytical Dataset (ADS) ready for predictive modeling several steps are followed:

ADS Features: Numerous characteristics have been integrated into the Analytical Dataset (ADS), each of which was painstakingly created to encompass key elements of network interfaces. These characteristics include aspects such as interface speed, in-bandwidth, out-bandwidth, traffic type (unicast/multicast), and the thorough utilization profile. Together, these characteristics act as priceless indicators, enhancing our capacity to accurately predict and anticipate interface congestion.

Feature Engineering and Behavioral Modeling: We calculated statistical metrics such as minimum, maximum, average, and summation values concerning relevant features, as judged appropriate. With the use of this method, we were able to aggregate different aspects of interface consumption throughout a range of time frames, including daily, weekly, and monthly aggregations.

Data Labeling: The research created three unique samples with varying proportions of congested and non-congested interfaces to balance this imbalance. This strategy was utilized to make sure that the information provided for training and evaluation appropriately reflected the distribution at its core.

D. Machine Learning Model

To create effective machine learning models following steps are involved:

Data Pre-processing: The researcher used a variety of Python tools, including NumPy, Pandas, Matplotlib, and Seaborn, to undertake an exploratory data analysis. To increase the effectiveness of our data analysis, we started by importing the necessary libraries and customizing Pandas' display settings. To make subsequent analysis easier, the three different samples (sample_10_30_70_TRAIN.csv, sample_10_50_50_TRAIN.csv, and sample_10_70_30_TRAIN.csv) were imported into independent Pandas Data Frames. The initial sampling is shown in Table 1.

Table 1: Sampling (Target Class, Numeric Value, No of Observations, %)

Dataset	Target Class	Numeric Value	Number of Observations	Percentage
All	Non-Congested Interfaces	0	3431327	98.20%
	Congested Interfaces	1	62903	1.80%
Sample 01	Non-Congested Interfaces	0	63112	70%
	Congested Interfaces	1	27048	30%
Sample 02	Non-Congested Interfaces	0	45919	50%
	Congested Interfaces	1	45919	50%
Sample 03	Non-Congested Interfaces	0	26958	30%
	Congested Interfaces	1	62903	70%

Feature Selection: To find and keep the most important traits, the researcher used variance thresholding with a threshold of 0. With the use of this method, low-variance elements that might only slightly affect the predicting job can be eliminated as shown in Table 2.

Table 2: Sampling (No of Features, Features Dropped & Remaining)

Dataset	Number of Features	Features Dropped after Correlation Analysis	Remaining Features
Sample 01	327	165	162
Sample 02	327	168	159
Sample 03	327	171	156

Train-Test Split: To assess the effectiveness of the models, each sample dataset was divided into training and testing sets. The models were trained on the training set, and their prediction performance was assessed on the testing set using a separate dataset as shown in Table 3.

Table 3: Sampling (x_train, x_test, y_train, y_test)

Dataset	X_train	X_test	y_train	y_test
Sample 01	67620	22540	67620	22540
Sample 02	68878	22960	68878	22960

Sample 03	67395	22466	67395	22466
------------------	-------	-------	-------	-------

Classification: To categorize the ADS samples, various classification algorithms such as XGB Classifier, Random Forest Classifier, Extra Trees Classifier, and Bagging Classifier were trained by the study. Other classifiers like LGBM Classifier and KNeighbors classifier also contributed but only one model known as the XGB classifier was able to outperform for ADS Sample 2 in terms of performance after evaluation. Additionally, the Decision Tree Classification Algorithm along with the SVC AdaBoost classifier Linear Discriminant Analysis, Linear SVM, Ridge Classifier CV, Ridge Classifier, and SGD Classifiers were among others that underwent training during this research project.

Model Deployment: The chosen XGBClassifier model was serialized and stored using the pickle library, ensuring it was prepared for deployment in a production environment. This comprehensive methodology was followed by the study to construct a framework for data collection, pre-processing, feature engineering, model training, and system implementation. In the area of network interface congestion analysis, this meticulous procedure improves the research's validity, reproducibility, and trustworthiness, allowing for perceptive observations and reliable projections. Doing high-quality research always requires having access to a dataset that has adequate data for training and validation. We considered this when creating our ADS construction schedule.

III. MODEL EVALUATION

Achieving the best outcomes in machine learning and artificial intelligence applications requires careful model selection. The study can increase the predictability and accuracy of their findings by selecting the best model, which will provide more insightful information and solutions for a range of real-world issues. Table 4 displays the selection of the Model. Ensuring the model or algorithm of choice is appropriate for the given job, dataset, and goals is an essential stage in the machine learning pipeline that will eventually improve the machine learning application's effectiveness. Figure 5 displays the metrics for the model evaluation.

Table 4: Model Selection

Dataset	Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken (seconds)
Sample 01	XGBClassifier	0.976	0.972	0.972	0.976	27.172
Sample 01	ExtraTreesClassifier	0.974	0.970	0.970	0.974	14.374
Sample 01	BaggingClassifier	0.974	0.968	0.968	0.974	49.199
Sample 01	RandomForestClassifier	0.972	0.965	0.965	0.972	45.060
Sample 01	LGBMClassifier	0.966	0.956	0.956	0.966	3.159
Sample 01	DecisionTreeClassifier	0.959	0.953	0.953	0.959	7.736
Sample 01	KNeighborsClassifier	0.933	0.925	0.925	0.933	24.831
Sample 01	ExtraTreeClassifier	0.932	0.921	0.921	0.932	0.541
Sample 01	SVC	0.932	0.912	0.912	0.932	286.944
Sample 01	AdaBoostClassifier	0.934	0.903	0.903	0.932	26.837
Sample 01	SGDClassifier	0.898	0.880	0.880	0.898	2.423
Sample 01	LogisticRegression	0.894	0.876	0.876	0.894	1.709
Sample 01	RidgeClassifierCV	0.887	0.875	0.875	0.888	1.452
Sample 01	RidgeClassifier	0.887	0.875	0.875	0.888	0.690
Sample 01	LinearSVC	0.890	0.875	0.875	0.890	44.904

Sample 01	LinearDiscriminantAnalysis	0.885	0.874	0.874	0.886	2.703
Sample 01	CalibratedClassifierCV	0.905	0.874	0.874	0.903	121.228
Sample 01	Perceptron	0.833	0.833	0.833	0.837	1.009
Sample 01	NuSVC	0.853	0.813	0.813	0.851	757.229
Sample 01	NearestCentroid	0.789	0.791	0.791	0.795	0.606
Sample 01	BernoulliNB	0.730	0.744	0.744	0.740	0.545
Sample 01	PassiveAggressiveClassifier	0.781	0.708	0.708	0.772	0.692
Sample 01	GaussianNB	0.386	0.550	0.550	0.313	0.558
Sample 01	QuadraticDiscriminantAnalysis	0.338	0.523	0.523	0.220	1.191
Sample 01	DummyClassifier	0.700	0.500	0.500	0.577	0.388
Sample 02	XGBClassifier	0.982	0.982	0.982	0.982	34.465
Sample 02	RandomForestClassifier	0.981	0.981	0.981	0.981	35.189
Sample 02	ExtraTreesClassifier	0.980	0.980	0.980	0.980	15.578
Sample 02	BaggingClassifier	0.979	0.979	0.979	0.979	56.886
Sample 02	LGBMClassifier	0.968	0.968	0.968	0.968	3.181
Sample 02	DecisionTreeClassifier	0.961	0.961	0.961	0.961	9.970
Sample 02	ExtraTreeClassifier	0.942	0.942	0.942	0.942	0.648
Sample 02	KNeighborsClassifier	0.932	0.932	0.932	0.932	29.741
Sample 02	SVC	0.914	0.914	0.914	0.914	315.917
Sample 02	AdaBoostClassifier	0.909	0.909	0.909	0.909	29.333
Sample 02	LogisticRegression	0.874	0.874	0.874	0.874	1.662
Sample 02	CalibratedClassifierCV	0.874	0.874	0.874	0.874	156.688
Sample 02	SGDClassifier	0.873	0.873	0.873	0.873	2.213
Sample 02	RidgeClassifier	0.870	0.870	0.870	0.870	0.618
Sample 02	LinearDiscriminantAnalysis	0.870	0.870	0.870	0.870	2.545
Sample 02	RidgeClassifierCV	0.870	0.870	0.870	0.870	1.223
Sample 02	LinearSVC	0.870	0.869	0.869	0.870	44.772
Sample 02	NuSVC	0.847	0.847	0.847	0.847	862.616
Sample 02	GaussianNB	0.809	0.809	0.809	0.808	0.593
Sample 02	Perceptron	0.806	0.806	0.806	0.806	0.649
Sample 02	NearestCentroid	0.804	0.804	0.804	0.804	0.589
Sample 02	BernoulliNB	0.796	0.796	0.796	0.796	0.658
Sample 02	PassiveAggressiveClassifier	0.628	0.629	0.629	0.612	0.835
Sample 02	QuadraticDiscriminantAnalysis	0.565	0.563	0.563	0.481	0.917
Sample 02	DummyClassifier	0.498	0.500	0.500	0.331	0.425
Sample 03	XGBClassifier	0.986	0.980	0.980	0.986	25.339

Sample 03	RandomForestClassifier	0.986	0.980	0.980	0.986	32.131
Sample 03	ExtraTreesClassifier	0.986	0.979	0.979	0.986	13.242
Sample 03	BaggingClassifier	0.985	0.979	0.979	0.985	46.108
Sample 03	LGBMClassifier	0.978	0.970	0.970	0.978	2.469
Sample 03	DecisionTreeClassifier	0.973	0.966	0.966	0.973	7.690
Sample 03	ExtraTreeClassifier	0.948	0.935	0.935	0.948	0.543
Sample 03	KNeighborsClassifier	0.942	0.916	0.916	0.941	25.187
Sample 03	SVC	0.918	0.909	0.909	0.918	283.989
Sample 03	AdaBoostClassifier	0.903	0.901	0.901	0.905	30.609
Sample 03	LinearDiscriminantAnalysis	0.876	0.857	0.857	0.877	2.445
Sample 03	LinearSVC	0.878	0.856	0.856	0.878	42.464
Sample 03	RidgeClassifierCV	0.875	0.855	0.855	0.876	1.114
Sample 03	RidgeClassifier	0.875	0.855	0.855	0.876	0.579
Sample 03	SGDClassifier	0.877	0.853	0.853	0.877	2.110
Sample 03	LogisticRegression	0.877	0.852	0.852	0.877	1.637
Sample 03	BernoulliNB	0.833	0.829	0.829	0.837	0.547
Sample 03	CalibratedClassifierCV	0.860	0.814	0.814	0.857	148.288
Sample 03	NearestCentroid	0.817	0.811	0.811	0.821	0.545
Sample 03	PassiveAggressiveClassifier	0.836	0.810	0.810	0.837	0.834
Sample 03	Perceptron	0.836	0.803	0.803	0.836	0.827
Sample 03	NuSVC	0.845	0.793	0.793	0.841	758.633
Sample 03	GaussianNB	0.402	0.571	0.571	0.332	0.590
Sample 03	QuadraticDiscriminantAnalysis	0.329	0.519	0.519	0.201	0.896
Sample 03	DummyClassifier	0.700	0.500	0.500	0.577	0.367

The Receiver Operating Characteristic (ROC) curve, is a graph that is recurrently used in binary classification tasks to assess how well a machine learning model is performing, particularly when the class distribution is disturbed. At various threshold values for the model, it plots the true positive rate (TPR) versus the false positive rate (FPR). Depending on how well the model performs, a typical ROC curve is a plot of TPR (on the y-axis) vs FPR (on the x-axis). About different threshold selections, the optimal ROC curve fits into the upper-left corner of the plot, indicating low FPR and high TPR. As seen in Figure 4, the ROC curve and AUC are useful in real-world applications for comparing several models and selecting the model that works best for a specific classification assignment. They offer insightful information about a model's capacity to distinguish between positive and negative cases at various decision thresholds. Figure 6 shows the top 20 features of our XGB Classifier model.

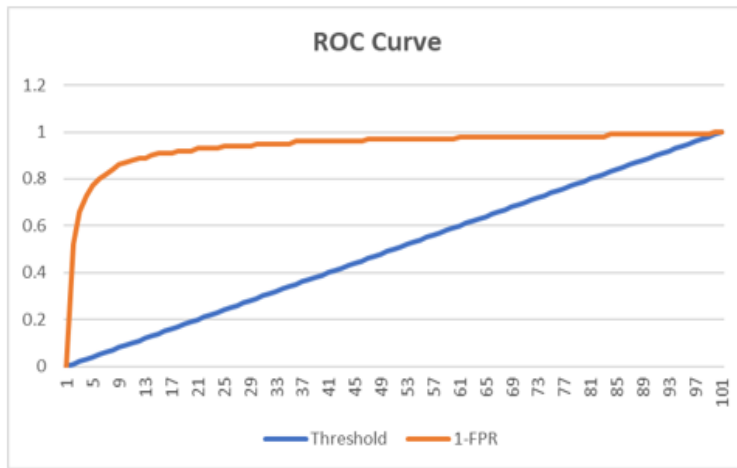


Figure 4: AUC/ROC Curve of XGB Classifier

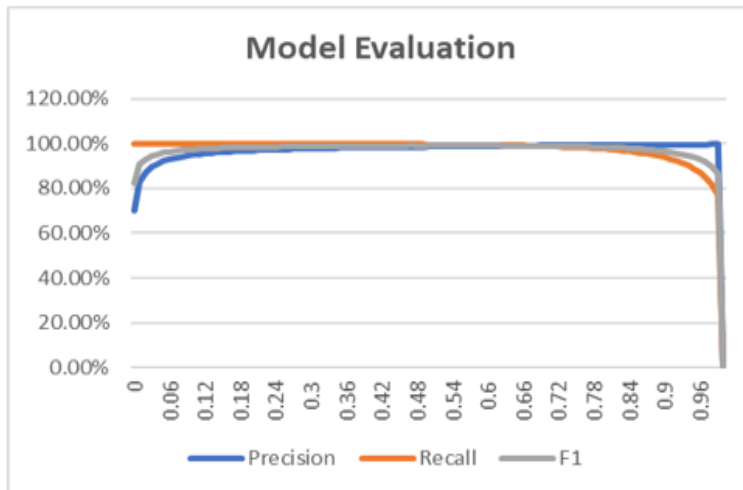


Figure 5: Evaluation Metrics of XGB Classifier

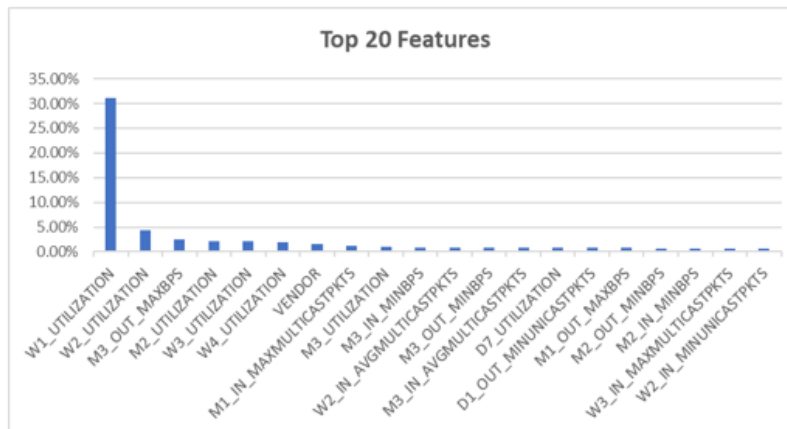


Figure 6: Top 20 Features of XGB Classifier

IV. CONCLUSION AND FUTURE WORK

The primary goal of this study was to identify the most effective machine learning method for forecasting network congestion in PTCL's fixed access network nodes. The objective was to enhance network performance and reliability by providing valuable insights and practical solutions. Network congestion in PTCL's fixed access network infrastructure posed a significant challenge, with potential impacts on customer satisfaction, overall network performance, service quality, latency, and connectivity. To achieve this goal, the study conducted a comprehensive assessment and comparison of various machine-learning approaches. These techniques included many different models and algorithms, each with unique benefits and advantages. Through a thorough evaluation procedure, the research aimed to identify the machine learning approach that best exhibited the highest prediction accuracy and appropriateness to predict network congestion.

Research is changing network performance and our experience of it thanks to work on predicting congestion in fixed access networks with machine learning. Future research should look at three key areas. First, improving ways to spot anomalies with methods like self-organizing maps. Second, adjusting congestion thresholds in line with network changes through dynamic thresholding. Third, making real-time predictions to manage data streaming rapidly. Also, merging different data sources boosts the accuracy of congestion prediction. This includes weather data and network performance indicators. By using designs such as convolution neural networks (CNNs) and recurrent neural networks (RNNs), we can grasp complex spatial and temporal network patterns. Critically, ensemble methods, online learning for flexibility, and explainability research can be harnessed to create human-sounding models. The use of cross-domain knowledge transfer expands the horizons of congestion prediction. Resource allocation optimization can help to distribute bandwidth. Using domain knowledge infusing models can be facilitated. The security of models can also be made a prime objective. Scalability measures are recommended for networks of large size and networks of networks. Network data used in congestion prediction can be protected from privacy loss using privacy-preserving algorithms. The use of standards and best practices in the industry can improve interoperability and trust.

REFERENCES

- [1] M. A. Ridwan, N. A. M. Radzi, F. Abdullah, and Y. E. Jalil, "Applications of machine learning in networking: a survey of current issues and future challenges," *IEEE Access*, vol. 9, pp. 52523–52556, 2021.
- [2] R. Galliera, A. Morelli, R. Fronteddu, and N. Suri, "MARLIN: Soft Actor-Critic based Reinforcement Learning for Congestion Control in Real Networks," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, 2023, pp. 1–10.
- [3] X. Ji, B. Han, C. Xu, C. Song, and J. Su, "Adaptive QoS-aware multipath congestion control for live streaming," *Comput. Networks*, vol. 220, p. 109470, 2023.
- [4] C. Daraojimba, A. D. Bakare, J. O. Olurin, K. M. Abioye, M. I. Obinyeluaku, and D. O. Daraojimba, "A REVIEW OF POST-COVID TELECOMMUNICATION INVESTMENT TRENDS: IMPACTS ON INFRASTRUCTURE DEVELOPMENT," *Comput. Sci. IT Res. J.*, vol. 4, no. 1, pp. 1–19, 2023.
- [5] M. Abubakr, U. Amjad, A. Raza, M. Khurram, H. Farman, and A. Ali, "COVID-19 diagnosis using Transfer Learning on Xray Images," *Pakistan Journal of Engineering, Technology & Science*, vol. 11, no. 1, pp. 15–28, 2023.
- [6] I. W. Katono, "Entrepreneurship in Education in a Developing Country Context: Constraints Facing Private Schools in Uganda," in *Restructuring Leadership for School Improvement and Reform*, IGI Global, 2023, pp. 359–377.
- [7] T. Cerquitelli, M. Meo, M. Curado, L. Skorin-Kapov, and E. E. Tsiropoulou, "Machine Learning Empowered Computer Networks," *Computer Networks*. Elsevier, p. 109807, 2023.
- [8] M. Soomro, U. Memon, M. Ali, and N. A. Qureshi, "Learning and development at PTCL and advent of COVID-19," *Emerald Emerg. Mark. Case Stud.*, vol. 13, no. 2, pp. 1–29, 2023.
- [9] Z. Li et al., "Deterministic Full-Fiber Service Network for the Fifth-Generation Fixed Network (F5G)," *IEEE Netw.*, 2023.
- [10] D. J. Miller, Z. Xiang, and G. Kesidis, "Adversarial learning targeting deep neural network classification: A comprehensive review of defenses against attacks," *Proc. IEEE*, vol. 108, no. 3, pp. 402–433, 2020.
- [11] L. Liu, B. Yin, S. Zhang, X. Cao, and Y. Cheng, "Deep learning meets wireless network optimization: Identify critical links," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 167–180, 2018.
- [12] K. Nugroho, E. Noersasongko, and H. A. Santoso, "Javanese gender speech recognition using deep learning and singular value decomposition," in *2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2019, pp. 251–254.

- [13] H. A. N. Zhiyan and W. Jian, "Speech emotion recognition based on deep learning and kernel nonlinear PSVM," in 2019 Chinese Control and Decision Conference (CCDC), 2019, pp. 1426–1430.
- [14] A. Sarkar, S. Dasgupta, S. K. Naskar, and S. Bandyopadhyay, "Says who? deep learning models for joint speech recognition, segmentation and diarization," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5229–5233.
- [15] M. Satsangi, M. Yadav, and P. S. Sudhish, "License plate recognition: A comparative study on thresholding, OCR and machine learning approaches," in 2018 International Conference on Bioinformatics and Systems Biology (BSB), 2018, pp. 1–6.
- [16] V. D’Orazio, J. Honaker, R. Prasady, and M. Shoemate, "Modeling and forecasting armed conflict: AutoML with human-guided machine learning," in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 4714–4723.
- [17] X. Zhou, M. Sun, G. Y. Li, and B.-H. F. Juang, "Intelligent wireless communications enabled by cognitive radio and machine learning," *China Commun.*, vol. 15, no. 12, pp. 16–48, 2018.
- [18] Z. Mammeri, "Reinforcement learning based routing in networks: Review and classification of approaches," *Ieee Access*, vol. 7, pp. 55916–55950, 2019.
- [19] R. Boutaba et al., "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *J. Internet Serv. Appl.*, vol. 9, no. 1, pp. 1–99, 2018.
- [20] F. Tang et al., "On removing routing protocol from future wireless networks: A real-time deep learning approach for intelligent traffic control," *IEEE Wirel. Commun.*, vol. 25, no. 1, pp. 154–160, 2017.
- [21] P. Geurts, I. El Khayat, and G. Leduc, "A machine learning approach to improve congestion control over wireless computer networks," in Fourth IEEE International Conference on Data Mining (ICDM’04), 2004, pp. 383–386.
- [22] S. Li, H. C. Leung, E. W. M. Wong, and C. S. Leung, "Enhancement of extreme learning machine for estimating blocking probability of OCS networks with fixed-alternate routing," *IEEE Access*, vol. 7, pp. 52319–52330, 2019.
- [23] N. Yuvaraj and P. Thangaraj, "Machine learning based adaptive congestion window adjustment for congestion aware routing in cross-layer approach handling of wireless mesh network," *Cluster Comput.*, vol. 22, pp. 9929–9939, 2019.
- [24] R. Proietti et al., "Experimental demonstration of machine-learning-aided QoT estimation in multi-domain elastic optical networks with alien wavelengths," *J. Opt. Commun. Netw.*, vol. 11, no. 1, pp. A1–A10, 2019.
- [25] F. Locatelli, K. Christodouloupoulos, M. S. Moreolo, J. M. Fabrega, and S. Spadaro, "Machine learning-based in-band OSNR estimation from optical spectra," *IEEE Photonics Technol. Lett.*, vol. 31, no. 24, pp. 1929–1932, 2019.
- [26] G. Choudhury, D. Lynch, G. Thakur, and S. Tse, "Two use cases of machine learning for SDN-enabled IP/optical networks: Traffic matrix prediction and optical path performance prediction," *J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D52–D62, 2018.