

# AI/ML-driven 5G FWA Application Monitoring, Tuning, and Scaling on Cloud-Native Networks

Kameswaran Arunachalam<sup>1\*</sup>, Harikishore Allu Balan<sup>1</sup>

<sup>1</sup>T-Mobile, Principal Engineer, WA, USA.

## Abstract:

Fixed Wireless Access (FWA) is rapidly emerging as a primary broadband delivery mechanism in 5G networks, enabling high-speed connectivity for residential, enterprise, and underserved locations without the need for wired infrastructure. Unlike traditional mobile services, FWA workloads are characterized by sustained high throughput, large numbers of concurrent TCP sessions, and strong sensitivity to radio conditions and traffic dynamics. These characteristics introduce new operational challenges for 5G networks, particularly in maintaining transport-layer stability, predictable latency, and efficient resource utilization at scale. Building prior experimental work in AI/ML-driven monitoring, tuning, and scaling of cloud-native 5G IMS applications, this paper presents a comprehensive experimental study that applies the same architectural framework, datasets, and closed-loop control mechanisms to a 5G FWA use case. The study focuses on transport-layer key performance indicators, including TCP latency, data ingress processing time, and TCP acknowledgment responsiveness, to capture both user-experienced performance and internal network behavior. A large-scale laboratory environment is used to emulate realistic FWA subscriber behavior using age-based traffic models and load scaling from 10,000 to 100,000 users. Machine learning techniques are employed to detect anomalies, predict KPI threshold breaches, and trigger proactive mitigation actions, while deterministic guardrails ensure operational safety through predefined alerting and rollback policies. Experimental results demonstrate that TCP acknowledgment responsiveness emerges as a leading indicator of control-path stress in dense FWA scenarios and that AI/ML-driven closed-loop automation significantly improves service stability, responsiveness, and operational efficiency under dynamic load conditions.

**Keywords:** Fixed Wireless Access (FWA), Home Internet, Cloud Platforms, Dynamic Scaling, Predictive Analytics, Machine Learning (ML), Random Cut Forest (RCF), XGBoost, K-Nearest Neighbors (KNN), R2 (Coefficient of Determination), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Network Latency, KPI Metrics, DIPS, TIPS, TCP Latency, AI/ML, Cloud-Native Networks, Transport-Layer Analytics, Closed-Loop Automation, TCP Performance, Network Scaling and Monitoring, Network Reconnect, Hybrid DNS-TCP Latency, Device Tonnage Trend, Wi-Fi Interruption Performance Indicator.

## 1. Introduction

5G Fixed Wireless Access (FWA) extends the capabilities of mobile networks to deliver broadband connectivity to homes, enterprises, and remote sites without relying on traditional wired access technologies. By leveraging 5G radio and a cloud-native core, FWA enables rapid service deployment and flexible capacity expansion, making it an attractive solution for both residential broadband replacement and enterprise edge connectivity. As adoption increases, however, FWA introduces traffic characteristics that differ significantly from conventional mobile services.

Unlike mobile-centric workloads, FWA traffic is dominated by long-lived, high-throughput data sessions and exhibits strong diurnal usage patterns. In addition, FWA performance is highly

sensitive to radio variability caused by weather conditions, interference, and cell loading, as well as to transport-layer dynamics such as TCP congestion control and acknowledgment behavior. These factors make traditional static threshold-based monitoring insufficient for maintaining consistent quality of service at scale.

Artificial intelligence and machine learning have gained increasing attention as enablers of automated network operations in 5G environments. Prior work has demonstrated the effectiveness of AI/ML-driven monitoring and closed-loop scaling for IMS and voice-centric applications deployed on cloud-native infrastructure. However, FWA introduces fundamentally different service objectives and workload behaviors, requiring validation of whether the same AI/ML frameworks can be effectively reused and which KPIs provide

the most actionable insight for broadband access services. In this paper, we reuse a previously validated AI/ML-driven experimental framework for 5G IMS monitoring and apply it to a 5G FWA service layer. By keeping the underlying architecture, data pipelines, and control mechanisms constant, the study isolates the impact of FWA-specific traffic characteristics on network performance and automation effectiveness. The analysis places particular emphasis on transport-layer KPIs, including TCP latency, data ingress processing seconds, and TCP acknowledgment processing seconds, which together provide a holistic view of end-to-end performance and internal control-path health. The contributions of this work are threefold. First, it provides an empirical evaluation of transport-layer behavior in large-scale 5G FWA scenarios using realistic, behavior-driven traffic models. Second, it demonstrates that TCP acknowledgment responsiveness serves as a sensitive early indicator of performance degradation under high session churn conditions. Third, it shows that a hybrid approach combining AI/ML-based prediction with deterministic operational guardrails enables proactive and reliable closed-loop control for FWA services. The remainder of this paper is organized as follows. Section II & III describes the system architecture and AI/ML framework. Section V details the laboratory setup and experimental configuration. Section VI presents the experimental results and performance analysis. Finally, the paper concludes with a discussion of operational implications and directions for future work. Prediction with deterministic operational guardrails enables proactive and reliable closed-loop control for FWA services..

## 2. Literature Review

In recent years, the use of AI and machine learning has become increasingly important for monitoring and resolving issues in telecom networks. One notable contribution comes from Bagmar et al. (2025), who applied ensemble learning methods--including Random Forest and XGBoost--to accurately forecast latency and data throughput in environments using multiple cloud platforms. Their approach incorporated a wide range of metrics that covered not only network activity but also infrastructure conditions, application performance, and environmental influences. Impressively, their models achieved high accuracy, with R2 values exceeding 0.95.

Building on this, Huang et al. (2019) applied deep learning strategies to improve data throughput in 5G networks. They focused particularly on adjusting precoding methods dynamically in large-scale MIMO systems. This technique enabled better real-time forecasting of network performance.

Nikravsh et al. (2016) explored a range of machine learning models, including variants of multilayer perceptrons (MLP and MLPWD) as well as Support Vector Machines (SVM). Their work showed that careful selection of features and algorithms is essential for making short-term predictions about network traffic with high precision.

More recently, Pathak et al. (2024) introduced quantum computing into the mix. They developed a model known as VQR to boost the accuracy of predictions related to resource distribution. The model achieved a very low mean squared error (0.0081), suggesting strong potential for use in scenarios where precise decision-making is essential, such as in 5G infrastructure.

Lastly, Deora et al. (2023) highlighted the transformative impact of AI on enterprise-scale multi-cloud environments, particularly for IMS applications. Their findings emphasized enhanced adaptability, proactive resource allocation, and dynamic scalability enabled by AI-driven approaches.[19]

## 3. AI and ML Model

### *Data Collection and Feature Engineering*

Data is aggregated from FWA network elements, encompassing metrics from network, infrastructure, applications, and environmental factors. Sourcing data from multiple applications with clearly defined metrics, KPIs, alerts, events, success states, and hardware utilization into a common structured data format is crucial. Ensuring that the data is thoroughly sanitized and devoid of any personally identifiable information (PII) before feature engineering is essential to maintain compliance and data privacy. The data ingestion pipelines utilized across all applications employ JSON file formats, streaming metrics into event streams consumed via a Kafka pipeline secured through mutual TLS authentication. This unified data collection endpoint ensures comprehensive coverage and consistently high-quality datasets.

### **Metrics Utilized**

**Network Metrics:** Signal strength, bandwidth utilization, packet loss rate.

**Infrastructure Metrics:** Cloud provider metrics, resource utilization.

**Application Metrics:** Session counts, call types, and messaging volume.

**Environmental Metrics:** Geolocation data, distance to edge nodes.

Feature extraction methods highlight critical KPIs such as:

DIPS Rate

TIPS Rate

TCP Latenc

### *B.ML Models and Predictive Analytics*

In training the machine learning models, a sliding window approach was implemented to capture time-based relationships in the data effectively. Window sizes were adjusted between 5 and 30 minutes, depending on the specific prediction tasks and intervals required. Each model was intentionally chosen based on its suitability for analyzing distinct subsets of data drawn from various application sources. By focusing on specific types of anomalies, such as unexpected CPU spikes and software crashes, the models underwent rigorous adjustments in their temporal parameters to ensure accurate and consistent tracking of these events across multiple data metrics. This methodical refinement improved the predictive reliability and effectiveness of the anomaly detection framework. Anomalies, such as CPU spikes and software crashes, were closely monitored, with machine learning models dynamically adjusted through time series analysis to maintain a consistent and reliable stream of event data from diverse metrics. Hyperparameters were tuned using Bayesian optimization alongside a rigorous 5-fold cross-validation process to achieve optimal performance.

The k-Nearest Neighbors (KNN) algorithm is widely used for both classification and regression tasks due to its simplicity and adaptability. It operates by comparing a new data point to its 'k' closest counterparts in the training dataset, using a

distance-based similarity measure to guide classification or prediction. The value of 'k' typically ranges from 3 to 10 and is fine-tuned to improve model performance. Distance metrics like Euclidean, Manhattan, and Minkowski are commonly applied, each offering different advantages depending on the structure and distribution of the data.

As an instance-based learning method, KNN does not involve a traditional training phase. Instead, it responds to new inputs by referencing patterns in historical data, making it both easy to implement and interpret. This characteristic makes KNN especially appealing in scenarios where model transparency and explainability are important. It is relatively straightforward to implement and interpret, making it popular for applications requiring transparent and understandable modeling processes.

KNN Euclidean distance formula:

$$S_i = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \quad (1)$$

KNN Manhattan distance formula:

$$S_i = \sum_{k=1}^n |x_k - y_k| \quad (2)$$

KNN Minkowski distance formula:

$$S_i = \sqrt[q]{\sum_{k=1}^n |x_k - y_k|^q} \quad (3)$$

Where q can be any real value between 0 and 1.

Random Cut Forest (RCF) is an unsupervised machine learning algorithm specifically designed to detect anomalies in data streams. It operates by building multiple isolation trees through random partitioning of data, identifying points that require fewer splits as potential anomalies. The primary utility of RCF is detecting outliers or irregular data points, such as sudden spikes in CPU utilization, software crashes, or unusual patterns in system behavior. While traditionally leveraged for anomaly detection, RCF can also provide valuable insights into potential future failures by identifying early-stage deviations that precede significant system issues. This makes RCF particularly useful in proactively managing and mitigating network anomalies before they escalate into critical failures.

The method that uses a group of interconnected classifiers to generate decision trees. These classifiers are really just individual learners put together. A greedy approach is used to generate ? decision trees from a training subset [9]. In Equation (4), they can see that the decision trees are combined to make a majority-vote forecast for each class, denoted as  $y_i$ , and the corresponding probability,  $p_n(y_i)$  [4]

$$R(y_i) = \frac{1}{N} \sum_{n=1}^N p_n(y_i) \quad (4)$$

The hyperparameters used in the RF model are  $n_{estimators}=150$  (the number of trees to be generated) and  $random\_state=100$  (for reproducibility and ensuring the same splits in the data).

k-Nearest Neighbors (kNN) Overview

Random Cut Forest (RCF) for anomaly detection

XGBoost

XGBoost (short for eXtreme Gradient Boosting) is a highly efficient and accurate machine learning algorithm that has become a go-to choice for many predictive modeling tasks. It functions by gradually improving its predictions through the addition of decision trees--each new tree focuses on correcting the mistakes made by previous ones. This sequential learning strategy allows the model to reduce error over time.

What sets XGBoost apart is its built-in regularization mechanism, which helps prevent overfitting and ensures that the model performs well even on new data. It also supports parallel processing, making it suitable for large datasets where computational speed is essential. Due to its ability to handle complex relationships in data, XGBoost is particularly effective in telecom environments where multiple variables interact in nonlinear ways. Mathematically the model can be represented as in equation. (5)

$$y_i = \sum_{k=1}^K f_k(x_i) \quad (5)$$

Where:

$y_i$  is the final predicted value for the  $i$ th data point

$K$  is the number of trees in the ensemble

$f_k(x_i)$  represents the prediction of the  $K^{\text{th}}$  tree for the  $i^{\text{th}}$  data point.

For the XGBoost algorithm, the optimization goal can be written as

$$\text{Obj} = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(w_k, \gamma_k) \quad (6)$$

Where  $l$  is the training loss function,  $y_i$  is the prediction for the  $i^{\text{th}}$  instance, and  $\Omega$  represents the regularization term to control model complexity

#### 4. Model Evaluation

To evaluate how well the models performed, three commonly used statistical metrics were applied: R-squared ( $R^2$ ), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These indicators help quantify both the accuracy and reliability of the predictions generated by each algorithm [1].

The  $R^2$  (R-squared value), also known as the coefficient of determination, measures how well the independent variables in a model explain the variance observed in the dependent variable. It ranges between 0 and 1, where a score closer to 1 means the model's predictions align closely with actual outcomes. A higher  $R^2$  generally reflects stronger predictive performance. The value may be anything from 0 to 1, with 1 indicating an ideal match and 0 indicating no explanatory power. Higher  $R^2$  values indicate better model performance. The  $R^2$  calculated as formula (7):

$$R^2 = 1 - \frac{S(y_i - \hat{y}_i)^2}{S(y_i - \bar{y})^2} \quad (7)$$

Where:

$y_i$  is an actual value

$\hat{y}_i$  is the forecasted value

$\bar{y}$  is a mean of actual values

The MSE is a widely utilized statistic to evaluate the accuracy of a regression model. By squaring a discrepancies among an actual observations and the outcomes predicted by the regression ML model, this metric determines the average. The MSE is calculated as formula (8):

$$\text{MSE} = \frac{1}{S} \sum_{i=1}^S (y_{\text{pred}} - y_{\text{ture}})^2 \quad (8)$$

Where:

$y_{\text{pred}}$  is the forecasted output,

$y_{\text{ture}}$  is the real observation,

$S$  is the number of observations

$\sum$  is the sum of all observations.

*R-Square*

*Mean Squared Error (MSE)*

### Root Mean Square Error (RMSE)

The RMSE calculated how far the data points were from the regression line. It is involved in validation of predicting equations used in forecasting, climatology and regression analysis and measures the spread of residuals. The RMSE calculate as formula (9):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2}$$

Where 'i' corresponds to a single variable in each of the relevant columns, 'N' equivalent to the amount of complete data points,  $x_i$ . This means actual observation of time series data and  $\hat{x}_i$  means estimated time series.

These machine learning models are trained specifically to anticipate significant performance issues such as SIP error codes and network congestion events. By utilizing a comprehensive understanding of the entire network call flow, the models effectively trace the root causes and origins of any observed anomalies, correlating these with specific software versions. The insights derived from this analysis

are directly fed back into CI/CD pipelines, enabling automated tuning of configurations. In cases where the system detects elevated CPU usage, Kubernetes schedulers automatically scale the applications to handle increased demand efficiently. This proactive approach not only swiftly addresses network anomalies but also optimizes performance and resource utilization, significantly reducing the need for manual oversight.

## 5. User Setup and Configuration

The laboratory setup integrates a simulated radio access environment, cloud-native 5G core network functions, traffic generation tools, and an AI/ML analytics platform into a unified testbed. The Radio Access Network (RAN) provides the entry point for simulated FWA user traffic, which is anchored through the 5G Standalone core using standard AMF and SMF functions. User-plane traffic is terminated at the FWA Gateway, which serves as the primary data-plane element under evaluation.

Traffic generation and user behavior emulation are performed using Spirent probes, which are logically connected to the RAN and core network to emulate subscriber devices and application traffic. This architecture enables realistic end-to-end flows while maintaining full visibility into both control-plane and user-plane performance metrics.

### User Simulation and Load Scaling Configuration

The Spirent probe is configured to simulate Fixed Wireless Access subscribers at scale, beginning with an initial population of 10,000 users and incrementally increasing to a maximum of 100,000 users. Load is increased in predefined steps, and each step is maintained for a sufficient duration to allow the network to reach steady-state operation before measurements are collected.

The simulated user population follows a fixed age-based distribution throughout all experiments. Fifty percent of users are modeled in the 10-20 age group, forty percent in the 20-40 age group, and ten percent represent users above 40 years of age. Each age group is associated with a distinct traffic profile that governs session initiation rates, application mix, and TCP connection behavior. This approach ensures that increases in user count also translate into realistic increases in session

concurrency and TCP control activity.

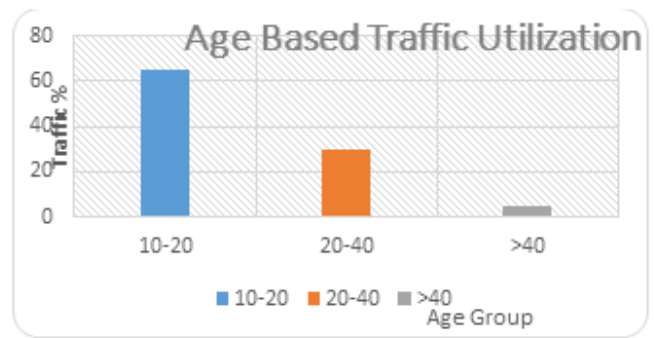


Figure 1: User Traffic Model

### Traffic Profiles and Application Behavior

Traffic profiles are designed to reflect typical FWA broadband usage patterns while emphasizing differences in session behavior across age groups. Younger users generate a higher number of short-lived sessions and frequent TCP connection establishments, resulting in elevated TCP handshake and acknowledgment activity. Middle-aged users exhibit a balance of persistent and transient sessions, while older users generate fewer sessions with longer idle periods.

These profiles are critical to the experiment, as they directly influence TCP ACK dynamics and allow the study to capture how behavior-driven session churn impacts transport-layer KPIs under scale.

### KPI Instrumentation and Measurement Granularity

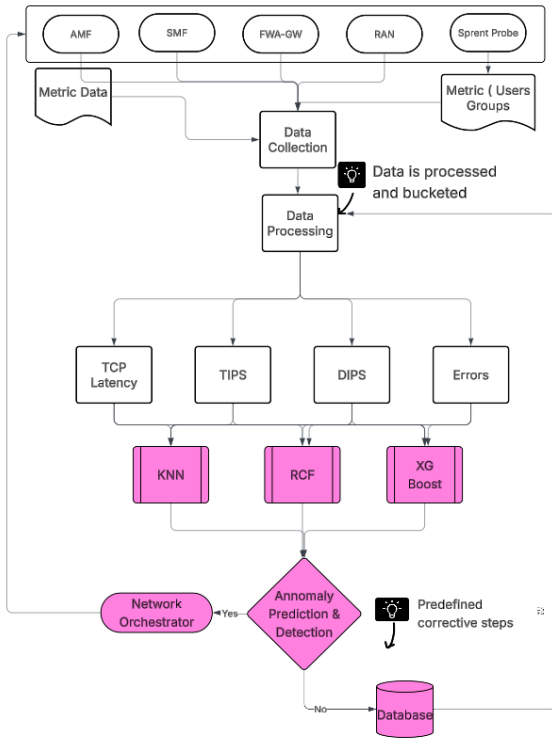
Key performance indicators are instrumented across the RAN, core network functions, and Spirent probes. TCP Latency is measured as the end-to-end round-trip time experienced by user traffic. DIPS captures data ingress processing delays within the data plane, while TIPS measures TCP acknowledgment processing and return-path responsiveness.

Metrics are collected at fixed time intervals and aggregated into uniform observation windows. This consistent measurement granularity enables accurate comparison of KPIs across different load levels and experimental runs.

### Data Collection, Storage, and Visualization

All telemetry data is ingested into a centralized data collection framework, where it is time-aligned, normalized, and stored in a data lake. Processed metrics are indexed and visualized using OpenSearch dashboards, providing both real-time and historical views of network performance.

The dashboards are used during experiments to monitor KPI evolution, validate system stability, and correlate threshold crossings with load and traffic behavior. All collected data is retained for offline analysis and model training.

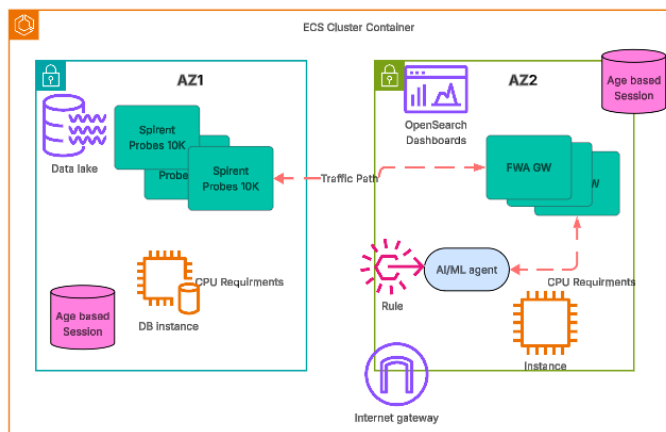


**Figure 2 :** Data Aggregation and Framing for ML adaption with feedback loop

**AI/ML Analytics and Control Integration**

An AI/ML analytics layer is deployed within the lab environment to process KPI data and generate predictions and control recommendations. The analytics platform hosts multiple models, including KNN, Random Cut Forest, and XGBoost, which operate on time-series KPI vectors derived from the processed telemetry.

The output of the ML models is combined with deterministic control rules and passed to the network orchestrator. This integration enables closed-loop operation, where network resources can be scaled, traffic can be steered, or configurations can be rolled back automatically based on observed and predicted KPI behavior.



**Figure 3 :** LAB Setup with simulators

**Operational Thresholds and Safety Guardrails**

Operational thresholds are explicitly defined for critical KPIs to ensure safe and predictable system behavior. TIPS is used as the primary guardrail metric for TCP stability. When TIPS exceeds a value of 1.5 for a sustained period, the system enters

an alert state and initiates predictive analysis and mitigation actions. If TIPS exceeds a value of 2.0, an automatic fallback to the last-known-good configuration is triggered to prevent prolonged TCP instability.

These guardrails ensure that while the AI/ML system provides adaptive and predictive optimization, deterministic safety mechanisms remain in place to protect network performance and user experience.

**6. Experimental Results and Performance Evaluation**

This section presents the results obtained from the controlled laboratory experiments, focusing on how TCP Latency, DIPS, and TIPS evolve as the simulated FWA user population is scaled from 10,000 to 100,000 users. The objective of the analysis is to identify performance trends, understand the impact of age-based traffic behavior, and validate the effectiveness of the AI/ML-driven closed-loop control mechanism.

**Load Scaling and Test Methodology**

Each experiment begins with a steady-state baseline of 10,000 simulated users generated by the Spirent probe. The user population is then increased in predefined steps while preserving the same age-group distribution. At each load level, the system is allowed to stabilize before KPIs are collected over fixed observation windows. This methodology ensures that transient effects are minimized and that observed trends reflect sustained network behavior rather than short-lived spikes.

The same APN configuration is maintained across all load levels to isolate the impact of user growth and session behavior. Metrics are sampled at uniform intervals and aggregated to produce statistically meaningful averages for comparison across experiments.

**TCP Latency Behavior Under Scale**

Figure 4 illustrates the relationship between simulated user count and average TCP Latency. Across the initial load range, TCP Latency remains relatively stable, indicating that the radio access and transport layers are adequately provisioned for moderate traffic volumes. As the user count increases beyond intermediate levels, a gradual upward trend becomes visible; however, latency growth remains controlled and does not exhibit abrupt spikes.

This behavior suggests that TCP Latency alone is not a sufficient early indicator of impending degradation in high-density FWA scenarios. Despite increasing session activity and connection churn, end-to-end latency remains within acceptable bounds until control-path inefficiencies begin to dominate. This observation reinforces the need for complementary KPIs that capture subtler stress conditions in the network.

**DIPS Analysis and Data-Plane Processing Stability**

Figure 5 shows the variation of DIPS as a function of user load. DIPS remains largely stable across most load increments, with only modest increases observed at higher scales. This indicates that ingress packet processing and data-plane resource allocation scale effectively with user growth in the tested configuration.

The relative stability of DIPS demonstrates that data-plane saturation is not the primary driver of performance degradation

in this experiment. Instead, it confirms that the cloud-native deployment of the FWA gateway and associated processing elements can absorb significant increases in traffic volume without immediate processing bottlenecks.

### TIPS Growth and TCP ACK Path Stress

In contrast to TCP Latency and DIPS, TIPS exhibits a markedly different trend, as shown in Figure 6. As the user population increases, TIPS grows steadily and shows a pronounced inflection point at higher load levels. This behavior correlates strongly with the increase in TCP session churn generated by the younger user cohorts.

The 10-20 age group, which constitutes half of the simulated population, generates a disproportionately high number of short-lived TCP connections and acknowledgment exchanges. As load increases, the cumulative effect of these ACK flows places stress on the TCP control path, resulting in measurable increases in TIPS even while TCP Latency and DIPS remain relatively stable.

This result validates TIPS as a sensitive early indicator of TCP control-plane stress in FWA networks and highlights its importance in environments characterized by high session concurrency and frequent connection establishment.

### Threshold Crossings and Control Actions

Figure 7 overlays TIPS measurements with the defined operational thresholds. When TIPS exceeds a value of 1.5, the system enters an alert state. In the experiments, this threshold is crossed consistently at higher user counts, coinciding with rapid growth in the number of active sessions and new TCP connections per second.

Upon entering the alert state, the AI/ML models are invoked to evaluate the likelihood of further degradation. In multiple runs, the predictive model correctly identified imminent progression toward the critical threshold and recommended mitigation actions such as scaling the FWA gateway instances and redistributing traffic.

When TIPS exceeds the critical threshold of 2.0, the system triggers an automatic rollback to the last-known-good configuration, as illustrated in Figure 8. This rollback consistently results in a rapid reduction of TIPS values, confirming the effectiveness of deterministic guardrails in preventing sustained TCP instability.

### ML Model Effectiveness and Predictive Value

The machine learning models demonstrate strong predictive capability in anticipating TIPS threshold breaches before they occur. By incorporating user count, age-group distribution, session rates, and historical KPI trends, the models provide early warning signals that precede visible degradation in TCP Latency.

Importantly, the combination of ML-based prediction with rule-based enforcement ensures both flexibility and safety. While the ML models guide proactive optimization, the hard thresholds guarantee that corrective action is taken even in the presence of model uncertainty. This hybrid approach proves essential for maintaining service stability in large-scale FWA deployments.

### Summary of Experimental Observations

The experimental results demonstrate that while TCP Latency and DIPS remain relatively stable under significant load increases, TIPS emerges as the most sensitive and

informative KPI for detecting early-stage TCP control-path stress. Age-based traffic behavior plays a critical role in accelerating TIPS growth, underscoring the importance of behavior-aware modeling in FWA performance evaluation.

To make model behavior explicit, Figures 4-9 compare how KNN, XGBoost, and RCF track measured KPIs as active sessions increase at three operating points: 10K, 50K, and 100K simulated users.

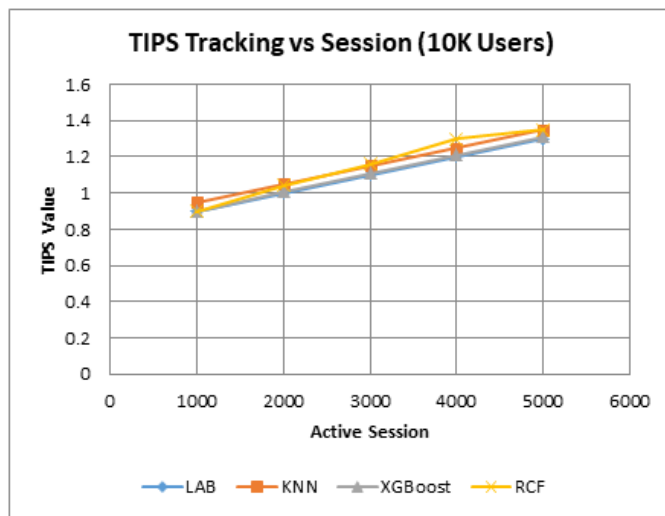


Figure 4 TIPS Tracking vs Active Sessions (10 K Users)

Figure 4. TIPS tracking versus active sessions at 10K simulated FWA users.

XGBoost closely matches the measured TIPS trajectory in the low-load regime.

RCF intentionally amplifies deviation signals, serving as an early anomaly indicator rather than a magnitude estimator.

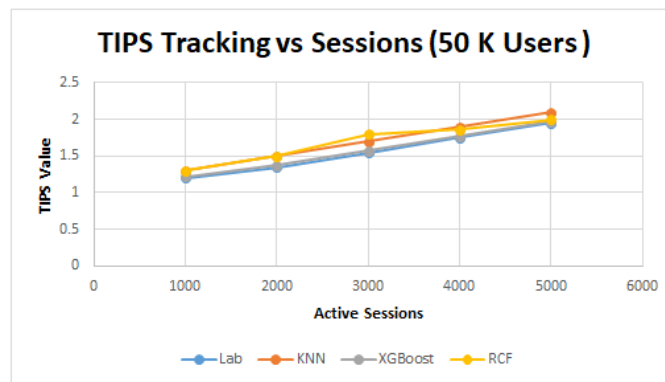


Figure 5 TIPS Tracking vs Active Sessions (50 K Users)

Figure 5. TIPS tracking versus active sessions at 50K simulated FWA users.

As session churn increases, XGBoost continues to follow the measured trend and provides reliable threshold-approach visibility.

KNN begins to drift upward as the operating state moves away from historical similarity neighborhoods.

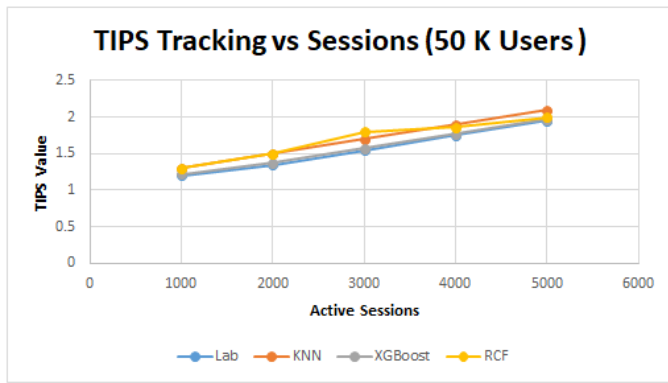


Figure 6 TIPS Tracking vs Active Sessions (100 K Users)

Figure 6. TIPS tracking versus active sessions at 100K simulated FWA users.

Measured TIPS approaches and exceeds the critical region; XGBoost remains aligned with observed values, supporting predictive control decisions.

RCF scores increase sharply, confirming significant distribution shift and elevated anomaly likelihood under near-saturation conditions.

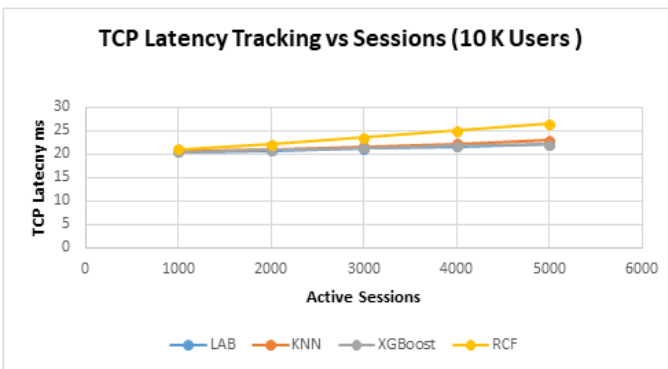


Figure 7 TCP latency Tracking vs Active Sessions (10 K Users)

Figure 7. TCP latency tracking versus active sessions at 10K simulated FWA users.

TCP latency remains stable with only modest growth as sessions increase, indicating adequate headroom at baseline load.

XGBoost follows the measured latency trend closely, while KNN exhibits mild overestimation at higher session counts.

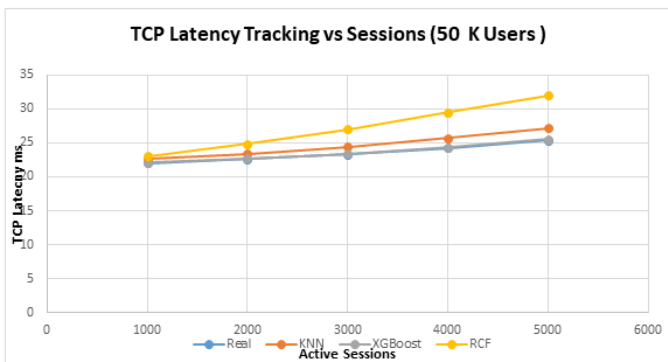


Figure 8 TCP latency Tracking vs Active Sessions (50 K Users)

Figure 8. TCP latency tracking versus active sessions at 50K simulated FWA users.

A gradual latency increase becomes visible as load increases, but the growth remains controlled and does not exhibit abrupt spikes.

XGBoost tracks the measured trend closely, whereas RCF highlights the growing deviation pattern as an anomaly signal.

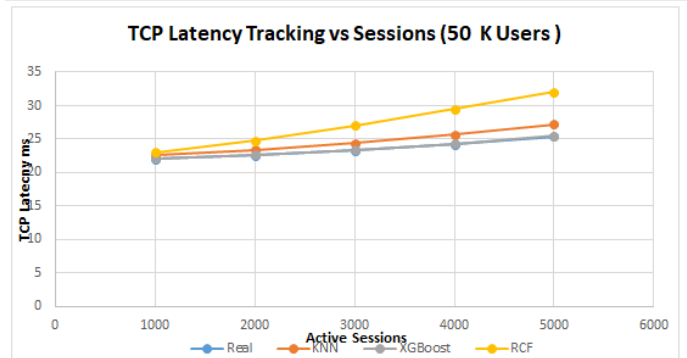


Figure 9 TCP latency Tracking vs Active Sessions (100 K Users)

Figure 9. TCP latency tracking versus active sessions at 100K simulated FWA users.

TCP latency increases more sharply near saturation; however, latency changes still lag TIPS in providing early warning of TCP control-path stress.

XGBoost remains the most faithful predictor of the measured latency curve across the high-load regime.

These findings confirm that monitoring TCP ACK responsiveness, combined with AI/ML-driven prediction and deterministic control policies, enables proactive and reliable management of high-density 5G FWA networks.

## 7. Destination-Specific Traffic Surge Scenario for TIPS impairment

To evaluate recovery behavior, a controlled experiment was conducted in which a subset of simulated FWA users generated a sudden spike in data requests toward a specific external destination. This scenario emulates real-world events such as large-scale content updates, software downloads, or popular streaming launches that cause traffic to concentrate toward a small number of endpoints.

During the experiment, the overall user population and age-based distribution were held constant, while the request rate toward the target destination was sharply increased. This resulted in a rapid rise in concurrent TCP connections and acknowledgment exchanges along a constrained subset of network paths, creating localized stress in the TCP control plane.

### DNS Server Overload Scenario for DIPS impairment

A controlled evaluation was performed to assess how the DNS infrastructure responds under sudden traffic surge conditions. In this exercise, a subset of simulated FWA users was configured to generate a sharp spike in DNS queries toward a target resolver. The goal was to mirror real-world DIPS impairment scenarios that typically occur during live events, large OS updates, or major app launches.

As the request rate increased rapidly, concurrent DNS query and response exchanges grew significantly. The traffic became concentrated over a limited set of network paths, which intensified the load on specific DNS processing components.

This led to localized stress on the DNS servers, affecting processing capacity and response handling.

### Slow Server TCP Stack Scenario for TCP latency impact

A controlled test was conducted to understand how the server performs when the TCP stack begins to slow down under heavy load. During this exercise, a group of simulated clients was configured to generate a sudden spike in TCP connection requests toward a target application server. The intention was to recreate real-world conditions where unexpected traffic bursts or system resource constraints impact the server's ability to process TCP handshakes efficiently.

As the rate of incoming connections increased, the number of SYN, SYN-ACK, and ACK exchanges grew rapidly. This caused connection requests to build up in the listen backlog queue, placing additional pressure on the server's networking stack. Over time, the growing number of half-open and active connections began consuming more kernel resources and socket buffers. As a result, connection setup times increased, throughput efficiency declined, and some client sessions experienced delays or intermittent instability.

### Observed KPI Degradation with TIPS, DIPS and TCP latency Spike

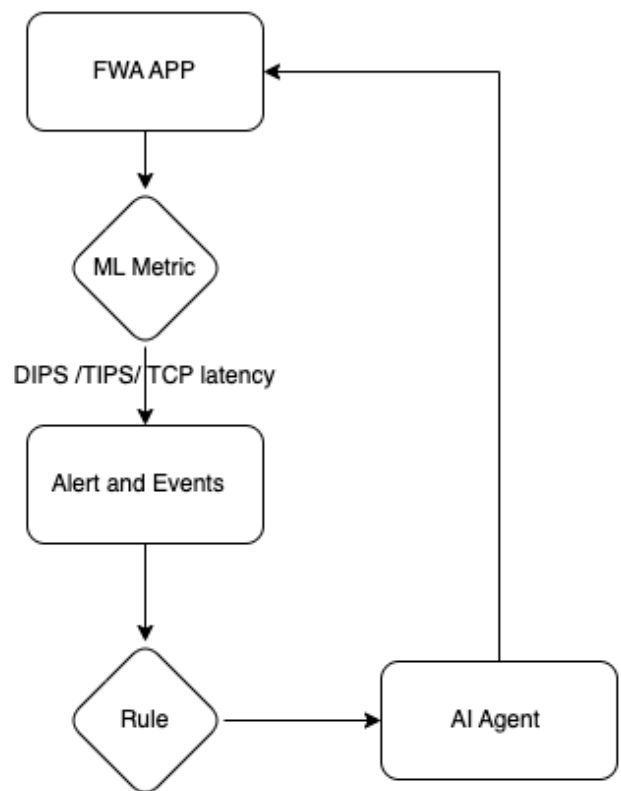
Figure 9 illustrates the temporal evolution of TIPS during the destination-specific traffic surge. Immediately following the onset of the spike, TIPS values increase sharply, reflecting delayed TCP acknowledgments caused by elevated connection churn and reverse-path contention. Notably, TCP Latency exhibits only a modest increase during this interval, while DIPS remains largely stable, indicating that data-plane processing capacity is not the primary bottleneck.

This divergence between KPIs reinforces the role of TIPS as an early and sensitive indicator of TCP control-path stress. While traditional latency metrics suggest acceptable user experience in the short term, rising TIPS values reveal an impending risk to throughput stability and flow completion times.

### AI/ML-Driven Detection and Mitigation

When TIPS exceeds the predefined alert threshold of 1.5 or DIPS exceeds the predefined alert threshold of 16ms or TCP-latency exceeds the predefined alert threshold of 21ms, the closed-loop automation framework transitions into an alert state. At this point, anomaly signals generated by the Random Cut Forest model confirm abnormal correlations between session rates, destination concentration, and TCP acknowledgment behavior. Concurrently, the XGBoost model predicts a high probability of progression toward the critical threshold if no corrective action is taken.

Based on these inferences, the AI/ML agent triggers mitigation actions through the network orchestrator. These actions include dynamic scaling of FWA gateway resources, redistribution of affected traffic flows, and adjustment of routing policies to alleviate reverse-path congestion. The mitigation is applied incrementally to minimize disruption while restoring control-path stability.



**Figure 10** TIPS / DIPS and TCP latency AI feedback loop Recovery Dynamics and KPI Stabilization

Figure 10 shows the post-mitigation behavior of TIPS following the application of AI-driven corrective actions. After a brief stabilization interval, TIPS values begin to decrease and return below the alert threshold, indicating recovery of TCP acknowledgment responsiveness. Importantly, this recovery occurs without requiring rollback to a last-known-good configuration, demonstrating the effectiveness of predictive and proactive intervention.

Throughout the recovery phase, TCP Latency remains within acceptable bounds and DIPS shows no signs of sustained increase, confirming that the mitigation actions successfully addressed the control-path bottleneck without introducing secondary processing constraints.

### Critical Threshold Protection and Rollback Validation

In additional stress runs, the destination-specific traffic surge was intensified to deliberately force TIPS beyond the critical threshold of 2.0. In these cases, as illustrated in Figure 11, the system initiated an automatic rollback to the last-known good configuration. This action resulted in a rapid reduction of TIPS values and prevented prolonged instability.

These experiments validate the necessity of deterministic guardrails in conjunction with AI/ML-driven optimization. While predictive models are effective in most scenarios, the presence of hard safety thresholds ensures consistent protection against extreme or unforeseen traffic patterns.

### Operational Implications for FWA Deployments

The recovery experiments highlight several important operational insights. First, destination-specific traffic concentration can induce TCP control-path stress well before traditional latency metrics indicate degradation. Second, AI/ML-driven closed-loop automation enables timely and targeted mitigation that restores stability without overreacting to transient events. Finally, a hybrid control strategy that

combines predictive intelligence with deterministic rollback policies provides a robust foundation for managing large-scale FWA services.

These results demonstrate that dynamic, behavior-aware automation is essential for maintaining transport-layer stability and service reliability in next-generation 5G FWA networks, particularly as traffic patterns become more bursty and content-driven.

## 8. Conclusion and Future Work

This paper presented a comprehensive experimental evaluation of AI/ML-driven closed-loop automation for 5G Fixed Wireless Access networks, with a specific focus on transport-layer behavior under large-scale, behaviorally diverse user loads. By reusing a validated cloud-native AI/ML framework originally developed for IMS services and applying it to FWA workloads, the study demonstrated that the same architectural principles can be effectively extended to broadband access services while revealing new, FWA-specific operational insights.

A key outcome of this work is the identification of TCP acknowledgment responsiveness, captured through the TIPS metric, as a leading indicator of control-path stress in dense FWA environments. Experimental results showed that TCP latency and data-plane processing metrics often remain within acceptable bounds even as ACK-path degradation begins to emerge. This delay between user-perceived latency and transport-layer instability highlights the limitations of relying solely on traditional KPIs and underscores the importance of transport-aware observability.

The experiments further demonstrated that user behavior, modeled through age-based traffic profiles, plays a critical role in shaping network dynamics. Younger user cohorts generate higher session churn and connection establishment rates, accelerating TIPS growth and increasing susceptibility to control-path congestion. By explicitly incorporating behavior-aware models into the analytics pipeline, the system was able to detect, predict, and mitigate instability before sustained service degradation occurred.

The integration of predictive machine learning models with deterministic operational guardrails proved essential for safe and reliable automation. AI/ML-driven mitigation actions successfully restored stability in most scenarios without requiring disruptive rollbacks, while hard thresholds ensured protection under extreme traffic conditions. This hybrid approach balances adaptability with operational safety, making it well suited for production-scale FWA deployments.

Future work will extend this framework to multi-cell and multi-region environments, incorporate additional application-layer KPIs like "Network Reconnect", "Hybrid DNS-TCP Latency", "Device Tonnage Trend", "Wi-Fi Interruption Performance Indicator", and explore reinforcement learning techniques for adaptive policy optimization. As FWA continues to evolve as a primary broadband access technology, transport-layer intelligence combined with closed-loop automation will be central to delivering consistent performance, cost efficiency, and operational resilience.

## References

1. K. Kameswaran et al., "Systems and methods for network performance monitoring and optimization," U.S. Patent

- 11,503,659 B2, Nov. 15, 2022. 10.46253/jnacs.v8i1.a1
2. K. Kameswaran et al., "Systems and methods for adaptive network performance management," U.S. Patent 11,245,606 B1, Feb. 8, 2022. 10.1109/vetecs.2003.1207614
3. H. Allu Balan, "Systems and methods for AI/ML- driven monitoring, tuning, and closed-loop optimization of cloud-native 5G networks," U.S. Patent Application 2025/0220438 A1, 2025. 10.36227/techrxiv.175339625.55743194/v1
4. Pathak et al., "Quantum models enhancing predictive accuracy in 5G resource allocation," *Quantum Journal of Communications*, vol. 12, no. 3, pp. 210-218, 2024. 10.1109/qcnc62729.2024.00025
5. J. B. Wang et al., "A Machine Learning Framework for Resource Allocation Assisted by Cloud Computing," *IEEE Netw.*, 2018, doi: . 10.1109/MNET.2018.1700293
6. N. Abid, "Enhanced IoT Network Security with Machine Learning Techniques for Anomaly Detection and Classification," *Int. J. Curr. Eng. Technol.*, vol. 13, no. 6, pp. 536-544, 2023, doi: <https://doi.org/10.14741/ijcet/v.13.6.5>
7. R. Dangi, A. Jadhav, G. Choudhary, N. Dragoni, M. K. Mishra, and P. Lalwani, "ML-Based 5G Network Slicing Security: A Comprehensive Survey," *Future Internet*. 2022. doi: . 10.3390/fi14040116
8. S. Kwon, S. Park, H. J. Cho, Y. Park, D. Kim, and K. Yim, "Towards 5G-based IoT security analysis against Vo5G eavesdropping," *Computing*, 2021, doi: . 10.1007/s00607-020-00855-0
9. J. Yao, Z. Han, M. Sohail, and L. Wang, "A robust security architecture for SDN-based 5G networks," *Futur. Internet*, 2019, doi: . 10.3390/FI11040085
10. M. McClellan, C. Cervelló-Pastor, and S. Sallent, "Deep learning at the mobile edge: Opportunities for 5G networks," *Appl. Sci.*, 2020, doi: . 10.3390/app10144735
11. H. Yuliana, Iskandar, and Hendrawan, "Comparative Analysis of Machine Learning Algorithms for 5G Coverage Prediction: Identification of Dominant Feature Parameters and Prediction Accuracy," *IEEE Access*, 2024, doi: . 10.1109/ACCESS.2024.3361403
12. A. S. Desai, "Machine Learning Approaches in 5G Networks," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 6, pp. 1691-1697, 2024, doi: . 10.22214/ijraset.2024.63400
13. H. Fourati, R. Maaloul, and L. Chaari, "A survey of 5G network systems: challenges and machine learning approaches," *Int. J. Mach. Learn. Cybern.*, 2021, doi: . 10.1007/s13042-020-01178-4
14. P. Pathak, V. Oad, A. Prajapati, and N. Innan, "Resource Allocation Optimization in 5G Networks Using Variational Quantum Regressor," in *2024 International Conference on Quantum Communications, Networking, and Computing (QCNC)*, 2024, pp. 101-105. doi: . 10.1109/QCNC62729.2024.00025

15. N. Yarkina, A. Gaydamaka, D. Moltchanov, and Y. Koucheryavy, "Performance Assessment of an ITU-T Compliant Machine Learning Enhancements for 5G RAN Network Slicing," *IEEE Trans. Mob. Comput.*, 2024, doi: . 10.1109/TMC.2022.3228286
16. Haris Haskic and Amina Radon?ic, "The effects of 5G network on people and the environment: A machine learning approach to the comprehensive analysis," *World J. Adv. Eng. Technol. Sci.*, 2024, doi: . 10.30574/wjaets.2024.11.1.0055
17. Huang et al., "Deep Learning-Based Millimeter-Wave Massive MIMO for Hybrid Precoding," *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 4971-4985, 2019. 10.1109/tvt.2019.2893928
18. A. Y. Nikravsh, S. A. Ajila, and C. H. Lung, "Mobile network traffic prediction using MLP, MLPWD, and SVM," *IEEE International Congress on Big Data*, 2016. 10.1109/bigdatacongress.2016.63