IRSTI 49.33.29 DOI https://doi.org/10.26577/JMMCS1302202612

B. Kumalakov[1], **N. Kassymbek**[2], **S. Aibagarov**[2*], **A. Mukhanbet**[2], **Y. Nurakhov**[2], **T. Imankulov**[2]
[1] Astana IT University, Astana, Kazakhstan
[2] Farabi University, Almaty, Kazakhstan
*e-mail: awer1307dot@gmail.com

COMPARATIVE ANALYSIS OF CLASSIFIER FAMILIES FOR INTENT-BASED 5G NETWORK SLICING

This study investigates the classification of 5G network slice types using the public "Network Slicing in 5G"dataset. We address a critical scientific pitfall by identifying and mitigating evaluation leakage caused by near-deterministic rule-encoded binary indicators and heavy data duplication. By excluding these artificial context flags and utilizing only a minimal set of QoS-observable telemetry — specifically packet delay, packet loss rate, time, and equipment category — we establish a rigorous, leakage-aware evaluation protocol. Five representative classifier families were evaluated using a group-safe splitting strategy to ensure results reflect real-world operational conditions. Our experimental results demonstrate that tree-based ensembles significantly outperform linear models, with the strongest ensembles reaching about 95% accuracy (Histogram-based Gradient Boosting at 94.74% and Extra Trees marginally higher at 95.14%). This research underscores that while measurable network telemetry provides sufficient signal for high-accuracy slice recognition, non-linear models are necessary to navigate the complex, stochastic overlaps inherent in real wireless environments.
Keywords: Network slicing, Machine learning, Quality of service (QoS), Data leakage mitigation, 5G/6G Networks.

Б. Кумалаков[1], Н. Қасымбек[2], С. Айбагаров[2*], А. Мұханбет[2], Е. Нурахов[2], Т. Иманкулов[2]
[1] Astana IT University, Астана, Қазақстан
[2] Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан
*e-mail: awer1307dot@gmail.com

5G желісін ниетке негізделген слайсингтеуге арналған классификаторлар отбасын салыстырмалы талдау

Бұл зерттеуде «Network Slicing in 5G» жалпыға қолжетімді деректер жинағын пайдалана отырып, 5G желілік слайстар түрлерін жіктеу қарастырылады. Біз ережелермен кодталған, дерлік детерминирленген бинарлы индикаторлар мен деректердің қатты қайталануынан туындаған бағалаудағы ағып кетулерді (evaluation leakage) анықтау және азайту арқылы маңызды ғылыми мәселені шешеміз. Осы жасанды контексттік жалаушаларды алып тастау және тек QoS бойынша бақыланатын телеметрияның минималды жиынтығын атап айтқанда, пакеттердің кешігуі, пакеттердің жоғалу деңгейі, уақыт және жабдық санатын пайдалану арқылы біз ағып кетуді ескеретін қатаң бағалау хаттамасын орнатамыз. Бес репрезентативті

классификаторлар отбасы нәтижелердің нақты пайдалану жағдайларын көрсетуін қамтамасыз ету үшін қауіпсіз топтық бөлу (group-safe splitting) стратегиясын қолдана отырып бағаланды. Біздің тәжірибелік нәтижелеріміз ағашқа негізделген ансамбльдердің сызықтық модельдерден айтарлықтай асып түсетінін көрсетеді, ал ең күшті ансамбльдер шамамен 95% дәлдікке қол жеткізді (Histogram-based Gradient Boosting — 94,74%, Extra Trees сәл жоғары — 95,14%). Бұл зерттеу өлшенетін желілік телеметрия слайстарды жоғары дәлдікпен тану үшін жеткілікті сигнал бергенімен, шынайы сымсыз орталарға тән күрделі, стохастикалық қабаттасуларда навигация жасау үшін сызықты емес модельдер қажет екенін көрсетеді.
Түйін сөздер: желілік слайсинг, машиналық оқыту, QoS, деректердің ағып кетуін азайту, 5G/6G желілері.

Б. Кумалаков[1], Н. Касымбек[2], С. Айбагаров[2*], А. Муханбет[2], Е. Нурахов[2],
Т. Иманкулов[2]
[1] Astana IT University, Астана, Казахстан
[2] Казахский национальный университет имени аль-Фараби, Алматы, Казахстан
*e-mail: awer1307dot@gmail.com

Сравнительный анализ семейств классификаторов для слайсинга сетей 5G на основе намерений

В данном исследовании изучается классификация типов сетевых слайсов 5G с использованием общедоступного набора данных ”Network Slicing in 5G”. Мы решаем критическую научную проблему, выявляя и устраняя утечку данных при оценке (evaluation leakage), вызванную почти детерминированными бинарными индикаторами, закодированными в виде правил, и сильным дублированием данных. Исключив эти искусственные контекстные флаги и используя лишь минимальный набор телеметрии, наблюдаемой по QoS, — в частности, задержку пакетов, уровень потери пакетов, время и категорию оборудования, — мы устанавливаем строгий протокол оценки, учитывающий утечки данных. Пять репрезентативных семейств классификаторов были оценены с использованием стратегии безопасного группового разделения (group-safe splitting), чтобы гарантировать, что результаты отражают реальные условия эксплуатации. Наши экспериментальные результаты показывают, что ансамбли на основе деревьев значительно превосходят линейные модели, при этом наиболее сильные ансамбли достигают около 95% точности (Histogram-based Gradient Boosting — 94,74%, Extra Trees немного выше — 95,14%). Данное исследование подчеркивает, что, хотя измеримая сетевая телеметрия обеспечивает достаточный сигнал для высокоточного распознавания слайсов, нелинейные модели необходимы для навигации в сложных, стохастических перекрытиях, присущих реальным беспроводным средам.
Ключевые слова: слайсинг сети, машинное обучение, QoS, устранение утечки данных, 5G/6G сети.

1 Introduction

5G networks are engineered to support highly heterogeneous services ranging from enhanced mobile broadband (eMBB) to ultra-reliable low-latency communications (URLLC) and massive machine-type communications (mMTC) over a common physical infrastructure. A key architectural mechanism to operationalize this heterogeneity is network slicing, which enables multiple logically isolated networks (slices) to coexist, each configured with its own performance targets and policy constraints. In the 5G system architecture, slicing is formalized via network slice instances and slice subnet instances spanning the RAN and core network domains, providing the basis for differentiated behavior and end-to-end service realization [1].

Beyond architecture, slicing must be managed as a lifecycle-driven operational capability. 3GPP management specifications define concepts, use cases, and requirements for slice management, including provisioning, modification, monitoring, and assurance emphasizing the need for KPI-driven automation and cross-domain coordination [2]. Industry frameworks also influenced early deployments by structuring slicing into clear layers (service, slice, and resource layers) and by clarifying how service requirements map into slice instantiation and isolation objectives [3]. In practice, slicing is tightly coupled with network softwarization: ETSI explicitly discusses how NFV/MANO constructs and network services relate to slice realization, clarifying touchpoints between slicing concepts and deployable virtualized network functions and services [4].

Despite maturity in standardization, operational slicing remains challenging. Slices must meet diverse SLA targets under fluctuating radio conditions, varying traffic mixes, and multi-tenant constraints. Survey work highlights unresolved issues such as isolation-efficiency trade-offs, end-to-end orchestration complexity, and the practical difficulty of closing the loop between monitoring and corrective actions at runtime [5]. Complementary architectural studies in the SDN/NFV context demonstrate how slicing can be implemented using programmable networking and virtualization, while also identifying gaps in end-to-end realization, monitoring consistency, and scalable control [6]. Broader surveys on slicing and softwarization further reinforce that, although enabling technologies are well established, reproducible validation is often limited by a lack of consistent datasets and evaluation methodologies that reflect real operational constraints [7].

From a performance perspective, slicing is not merely an administrative partitioning mechanism; it is deeply connected to how wireless resources are allocated and how reliability/latency constraints are enforced. Communication-theoretic analyses show that supporting eMBB/URLLC/mMTC concurrently requires careful resource management and highlights the non-trivial interactions between traffic demands, reliability, and delay in the RAN [8]. Consequently, data-driven and learning-based approaches have become increasingly prominent for slice operations. For example, deep reinforcement learning is widely explored to dynamically allocate resources among slices under time-varying demand, aiming to satisfy heterogeneous constraints while improving overall utilization [9].

A foundational step in many slicing pipelines is slice type recognition/selection from observable measurements. Given telemetry such as packet delay and loss (and potentially device-capability indicators or coarse context signals), an operator may need to infer which slice type is being served or should be selected for an incoming demand. This task aligns with slice management studies and standardization work that emphasize intent-to-policy mapping and assurance based on observed KPIs [10]. Within the logical 5G architecture discussion around 3GPP slicing, slice selection and composition can be viewed as part of forming an end-to-end logical network tailored to service requirements, motivating practical classification-style formulations when direct policy labels are not explicitly available at runtime [11]. At the same time, surveys on machine learning for slicing resource management outline both the promise of ML for automation and the risks in experimental practice, including data bias and non-stationarity, which can lead to overly optimistic conclusions if evaluation protocols are weak [12].

A particularly important pitfall for slice type classification on public datasets is evaluation leakage. Many datasets contain heavy duplication and near-deterministic mappings between "service indicator" features (e.g., application-domain flags) and the slice label. If identical samples appear in both training and test sets, models can achieve deceptively perfect results without learning generalizable relationships. Leakage is a well-documented issue in applied ML and data mining, and must be explicitly detected and avoided to ensure scientific validity [13]. Therefore, leakage-aware splitting (e.g., group-based splitting that blocks identical feature rows from crossing train/test boundaries) is essential for credible slice type classification results.

In this paper, we study slice type classification using the public Kaggle dataset "Network Slicing in 5G" [14] to address a key scientific question: under leakage-aware evaluation, how accurately can slice type be inferred from a minimal set of measurable features, and which model families remain robust when the dataset contains heavy duplication and potentially rule-encoded indicators? To answer this, we evaluate five representative classifier families (linear, margin-based, and tree-ensemble models) under a group-safe split that prevents duplicate leakage. The primary novelty and contribution of this work is the introduction of a leakage-aware evaluation protocol and baseline suite for slice type classification on a widely used public dataset, paired with feature-set comparisons that distinguish QoS-observable signals from rule-encoded context indicators. Performance is reported using standard multi-class metrics: accuracy, macro-F1, and balanced accuracy. All experiments are implemented with widely adopted open-source tooling to support reproducibility [15,16].

2 Materials and Methods

This study utilizes the public Kaggle dataset "Network Slicing in 5G". It contains records of network parameter measurements and includes both continuous and binary categorical features reflecting service configurations and their Quality of Service (QoS) requirements. The target variable is *Slice Type*, representing the network slice label. The Table 1 below provides a description of all columns in the original dataset.

Table 1. Description of columns

Column Name	Description
LTE/5g Category	User Equipment (UE) category/class that describes its capabilities and performance specifications.
Time	Timestamp or time index of the observation/measurement.
Packet Loss Rate	The fraction of lost packets: the number of undelivered packets divided by the total number of packets sent.
Packet delay	Time taken for a packet to be delivered (packet-level latency).
Slice Type	Network slicing configuration label (the target class).
GBR / Non-GBR	Indicators of Guaranteed Bit Rate (1/0) and Non-Guaranteed Bit Rate (1/0).
LTE/5G	Binary indicator of the LTE or 5G connectivity being used (1/0).
IoT / IoT Devices	Binary indicators of an Internet of Things service context and the presence of IoT devices (1/0).
AR/VR/Gaming	Indicator of AR/VR or gaming usage (1/0), typically sensitive to latency.
Healthcare	Usage in healthcare scenarios (1/0).
Industry 4.0	Usage in digital enterprise / Industry 4.0 scenarios (1/0).
Public Safety	Usage for public welfare and safety purposes (1/0).
Smart City & Home	Usage in smart city automation or household tasks (1/0).
Smart Transportation	Usage in connected/public transportation scenarios (1/0).
Smartphone	Indicator of traffic corresponding to smartphone usage (1/0).

The original dataset has a dimensionality of 31,583 rows by 17 columns. To initially understand the internal relationships within the data and identify potential multicollinearity, a pairwise correlation matrix was constructed for all features except the target variable. As shown in Figure 1, many binary context indicators exhibit specific co-occurrence patterns, indicating their artificial nature stemming from the simulation or dataset generation process.

2.1 Description of the Evaluated Models

To solve the network slice type classification task, five machine learning models representing different algorithm families were selected.

Logistic Regression. A linear classifier that models the probability of an object belonging to a class using a logistic function. Mathematically, the probability for the binary case (which generalizes to multiclass) is expressed as:

$$P(y = 1 \mid X) = \frac{1}{1 + e^{-(w^T X + b)}} \tag{1}$$

where w represents the feature weights, X is the feature matrix, and b is the bias.

Support Vector Machine (LinearSVC). LinearSVC constructs a hyperplane that best separates the classes by maximizing the margin. The optimization problem consists of minimizing:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \tag{2}$$

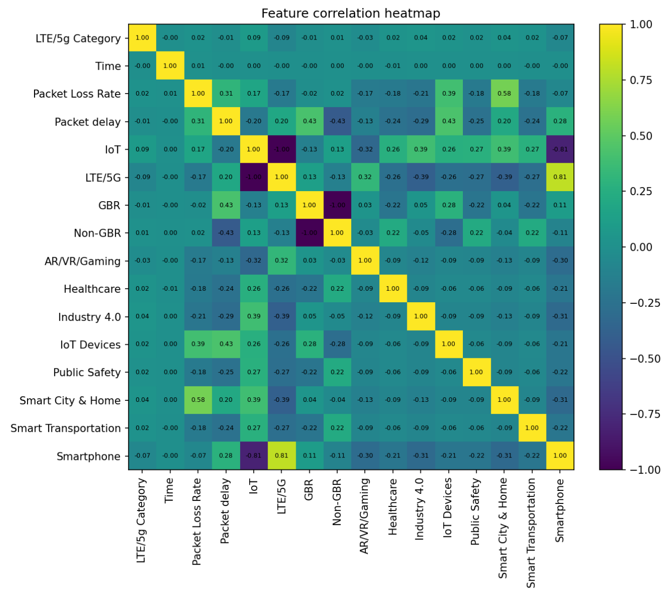

Figure 1: Correlation matrix.

where ξ_i allows for a certain proportion of errors (soft margin), and C regulates the regularization strength.

Random Forest (RandomForestClassifier). An ensemble method utilizing bootstrap aggregating (bagging) over decision trees. The final prediction is obtained by averaging or voting across multiple independent trees, each trained on a random subsample of the data and a random subset of features, which significantly reduces the model's variance.

Extra Trees (ExtraTreesClassifier). A modification of the random forest where the thresholds for splitting nodes in the trees are chosen randomly rather than by searching for the optimal information gain. This adds additional stochasticity to the algorithm and accelerates the training process.

Histogram-based Gradient Boosting (HistGradientBoostingClassifier). An efficient implementation of gradient boosting that pre-groups continuous features into discrete bins (histograms). The ensemble builds trees sequentially: each new tree is trained on the negative gradient of the loss function L of the previous composition:

$$F_m(x) = F_{m-1}(x) + \nu \, h_m(x), \tag{3}$$

where ν is the learning rate.

To ensure full reproducibility, all models were trained with fixed random seeds (`random_state` = 42) and the exact hyperparameter configurations summarized in Table 2. Continuous features were median-imputed, and for the distance/margin-sensitive linear models a min-max scaler to the range $[0, 1]$ was additionally applied within the pipeline; the tree-based ensembles were trained on the imputed features without scaling, as they are invariant to monotonic feature transformations. All experiments were implemented with scikit-learn.

Table 2. Hyperparameter configuration of the evaluated models

Model	Key hyperparameters
LogReg (multinomial)	`solver=lbfgs`, `max_iter=4000`, multinomial loss, default L_2 regularization; pipeline: median imputation + min-max scaling.
LinearSVC	squared-hinge loss, default $C = 1.0$, `random_state=42`; pipeline: median imputation + min-max scaling.
RandomForest	`n_estimators=500`, `random_state=42`, default Gini criterion and unrestricted depth.
ExtraTrees	`n_estimators=800`, `min_samples_leaf=2`, `random_state=42`, randomized split thresholds.
HistGradientBoosting	`max_depth=6`, `learning_rate=0.05`, `max_iter=600`, `random_state=42`.

2.2 Evaluation Metrics

To comprehensively evaluate the models' performance, the following metrics were applied:

Accuracy: overall percentage of correctly classified instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4}$$

Macro-F1 (Macro-averaged F-score): The harmonic mean of Precision and Recall. In macro-averaging, the metric is calculated independently for each class, and then their arithmetic mean is taken. This allows for an equal evaluation of model performance across all classes.

$$F1_{macro} = \frac{1}{N}\sum_{i=1}^{N} 2 \cdot \frac{Precision_i \cdot Recall_i}{Precision_i + Recall_i} \tag{5}$$

Balanced Accuracy: Calculated as the average Recall for each class. This metric is particularly useful for objectively evaluating imbalanced datasets.

$$BalancedAccuracy = \frac{1}{N}\sum_{i=1}^{N} \frac{TP_i}{TP_i + FN_i} \tag{6}$$

2.3 Data Analysis and Feature Selection

During the exploratory data analysis (EDA), it was determined that most of the features (12 out of 16) are binary flags denoting high-level service context (e.g., Healthcare, Smart Transportation, GBR).

To evaluate the statistical relationship between each feature and the target variable Slice Type, the Mutual Information method was applied. Unlike linear correlation, mutual information can capture any dependencies, including non-linear ones, by measuring the reduction in uncertainty of one feature given the known value of another. The results of this analysis are presented in Figure 2.

As can be seen in Figure 2, the binary context flags exhibit an anomalously high predictive power. In fact, these features act as rule-encoded indicators that deterministically define the slice

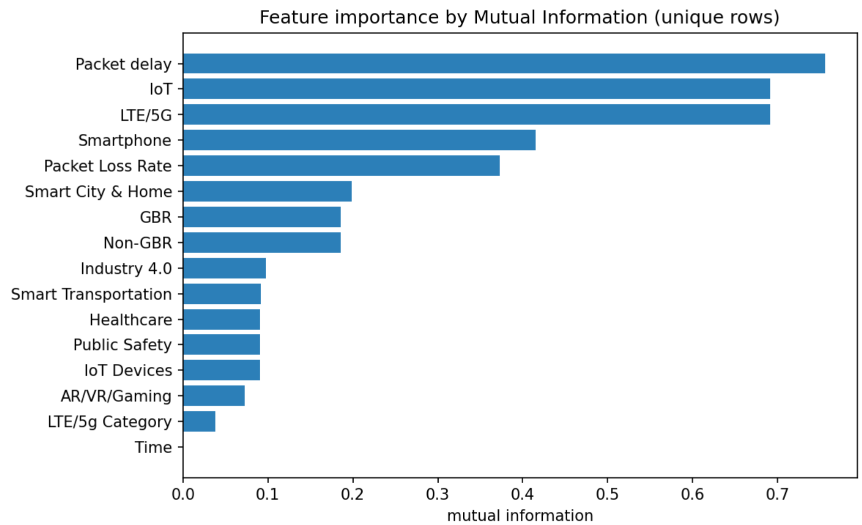

Figure 2: Feature importance based on mutual information.

type. Utilizing such features during modeling leads to severe evaluation leakage: the model does not discover hidden network patterns but simply memorizes an artificially created look-up table.

Furthermore, in real-world 5G network operations, operators typically do not have access to explicit high-level application labels during the traffic routing and resource allocation phase. Such information is extremely difficult to extract from the data stream in real time. At the same time, basic network telemetry (QoS-observable signals), such as Packet delay and Packet Loss Rate, is an easily measurable and accessible metric.

Based on these facts, a critical decision was made: to completely exclude all binary context indicators from the dataset. Only a minimal set of objectively measurable parameters was retained for further modeling: Packet delay, Packet Loss Rate, Time, and LTE/5g Category. This allowed us to formulate a fair protocol (leakage-aware evaluation) and verify how accurately the models can recognize the slice type relying exclusively on realistic telemetry.

It is important to clarify the distinction between the two LTE/5G-related fields, since they are conceptually different despite the similar naming. The field *LTE/5G* is a one-hot binary context flag (1/0) that encodes which radio-access technology the scenario was generated under; like the other rule-encoded indicators, it carries no continuously measurable physical magnitude and was therefore discarded together with the rest of the binary context group. In contrast, *LTE/5g Category* is an ordinal device-capability class (UE category) describing the throughput and feature tier of the user equipment. This quantity is observable from the radio control plane during normal operation, takes a graded range of values rather than a binary scenario label, and does not deterministically encode the slice; for these reasons it is retained as a legitimate telemetry-adjacent feature.

The inclusion of *Time* warrants particular scrutiny, since a raw time index can act as a proxy for the data-generation order and thereby reintroduce a subtle form of leakage. We therefore examined its role explicitly. First, as Figure 2 shows, *Time* has very low *individual* mutual information with the target, so it does not single-handedly determine the slice label and is not a dominant rule-encoded indicator. Second, the grouping and deduplication procedure described in Section 2.4 operates on the full retained feature tuple, so identical observations are never split across the train/test boundary regardless of their time stamp. Third, although its univariate signal is weak, retaining *Time* increases the number of *distinct* feature combinations in the data (the count of unique feature groups rises from a few hundred to several thousand once *Time* is included), which both strengthens the group-safe split and lets the non-linear ensembles exploit weak interaction effects that are invisible to a univariate measure. The principal discriminative signal nonetheless originates from the physical QoS measurements (delay, loss, device category), with *Time* acting only as a secondary, weakly informative covariate rather than a label-revealing ordering artifact. We report results with *Time* included for transparency, and we flag the removal of any residual ordering dependence as a direction for future dataset curation.

2.4 Leakage-Aware Group-Safe Splitting

The central methodological safeguard of this study is the group-safe train/test split, which we now define formally. Let each observation be represented by its retained feature vector $x_i = (\text{Packet delay}, \text{Packet Loss Rate}, \text{Time}, \text{LTE/5g Category})$. We define a deterministic grouping function $g(x_i)$ that assigns a group identifier to every sample via a hash of its complete feature tuple, so that two samples receive the same identifier if and only if they are identical across all retained features:

$$g(x_i) = \text{hash}\big(x_i\big), \qquad g(x_i) = g(x_j) \iff x_i = x_j. \tag{7}$$

The grouping variable is therefore the unique feature-row identity, not the time index or any external key. All samples sharing a group identifier are constrained to fall entirely within either the training or the test partition; no group may straddle the boundary. This directly blocks the duplicate-leakage mechanism, because the heavy row duplication present in the dataset (the raw data contains 31,583 rows but only 5,449 unique feature groups) can no longer place identical rows on both sides of the split.

A small number of groups (474) contained inconsistent labels across their identical feature rows; for these we assigned the majority (modal) label within the group to obtain a well-defined target before splitting. Partitioning was performed with `StratifiedGroupKFold` ($k = 5$, `shuffle=True`, `random_state=42`), taking the first fold as the held-out test set. This simultaneously preserves the class proportions and enforces group integrity. The resulting partition contains 25,266 training samples and 6,317 test samples with zero group overlap, and the class distribution is essentially identical in both partitions (Slice 1 ≈53.4%, Slice 2 ≈23.2%, Slice 3 ≈23.4%).

To quantify the severity of the leakage that this protocol removes, we performed an ablation comparing the group-safe split against a naive stratified random split (which allows duplicated rows to leak across the boundary). The results are reported in Table 3. The gap is modest but systematic for the ensemble models: a naive split mildly inflates apparent performance for some configurations while the group-safe split yields the more conservative and operationally honest estimate. The relatively small magnitude of the gap (well under two percentage points) indicates that, once the binary rule-encoded indicators are removed, the residual leakage attributable to row duplication on the continuous telemetry is limited rather than catastrophic — a finding that itself supports the validity of the retained feature set.

Table 3. Ablation — naive random split vs. group-safe split (Accuracy)

Model	Naive split	Group-safe split	Δ (naive − safe)
LogReg (multinomial)	0.8292	0.8259	+0.0033
LinearSVC	0.8368	0.8344	+0.0024
RandomForest	0.9381	0.9462	−0.0081
ExtraTrees	0.9387	0.9514	−0.0127
HistGradientBoosting	0.9424	0.9474	−0.0051

3 Results

This section presents the results of computational experiments for classifying network slice types (Slice Type) based on a filtered set of measurable network parameters (QoS-observable signals), such as packet delay, packet loss rate, time, and equipment category. The evaluation was conducted using a leakage-aware strategy, which ensures the objectivity and realism of the obtained metrics.

To compare the effectiveness of the five selected machine learning algorithms, we used three key metrics: overall accuracy (Accuracy), macro-averaged F-measure (Macro-F1), and balanced accuracy (Balanced Accuracy). The evaluation results of the models on the test set are summarized in Table 4.

Table 4. Comparative model performance table

Model	Accuracy	Macro-F1	Balanced Accuracy
ExtraTrees	0.9514	0.9503	0.9523
HistGradientBoosting	**0.9474**	**0.9455**	**0.9436**
RandomForest	0.9462	0.9448	0.9462
LinearSVC	0.8344	0.8382	0.8491
LogReg (multinomial)	0.8259	0.8268	0.8334

As can be seen from Table 4, tree-based ensemble methods demonstrated significant superiority over linear models. To visually demonstrate this gap, we constructed a comparative visualization of the metrics, which illustrates the overall performance trend.

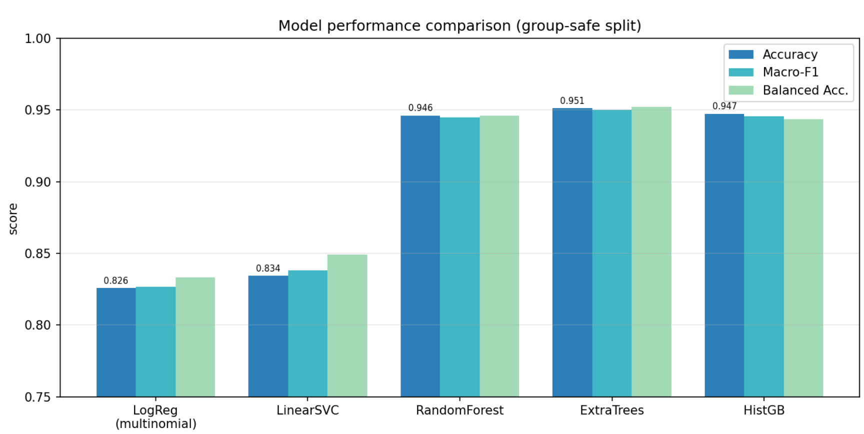

Figure 3: Comparative chart of Accuracy, Macro-F1, and Balanced Accuracy metrics for various models.

Analyzing Figure 3 and the table data, it can be concluded that linear classifiers Logistic Regression (LogReg) and Support Vector Machine (LinearSVC) show a baseline level of quality with an Accuracy of around 0.83 and a Balanced Accuracy of 0.83–0.85. This indicates that the measurable network parameters possess a certain degree of linear separability; however, the boundaries between the classes are too complex for simple hyperplanes. It is interesting to note that for linear models, the Balanced Accuracy slightly exceeds the overall Accuracy. This suggests their relative robustness to minority classes when class weights are properly adjusted, but their overall predictive capacity remains limited.

Decision tree-based ensembles handled the task substantially better, forming a tightly clustered top tier. RandomForest achieved an Accuracy of 0.9462, HistGradientBoosting reached 0.9474, and ExtraTrees attained the highest Accuracy of 0.9514. The three ensembles are statistically close (within roughly half a percentage point of one another), indicating that once a model can express non-linear decision boundaries, the measurable telemetry is sufficient to recover the slice type with high fidelity. In the remainder of this section we analyze HistGradientBoosting in detail as a representative high-performing ensemble; its behavior is characteristic of the tree-based family as a whole.

To better understand the physical nature of class separation and the reasons for the strong advantage of non-linear tree-based algorithms over linear ones, a scatter plot was constructed.

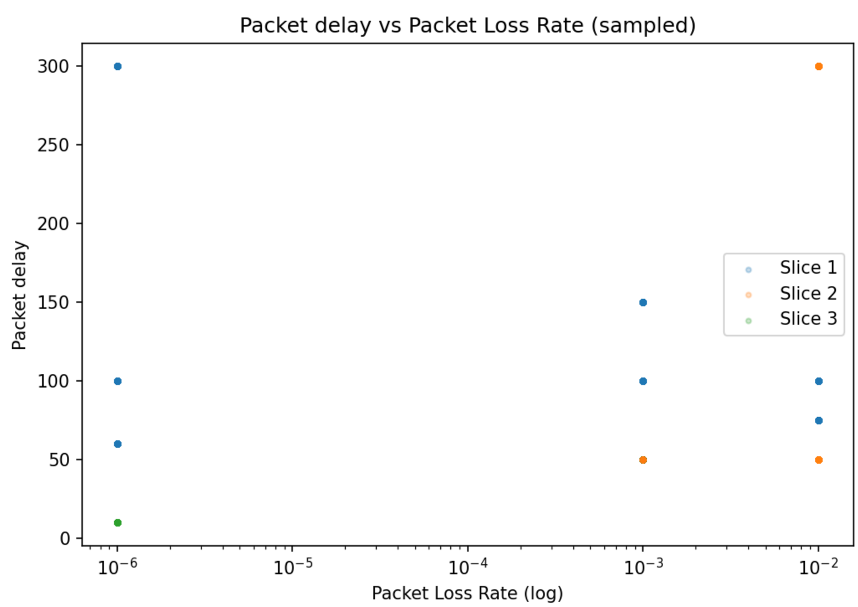

Figure 4: Scatter plot illustrating the distribution of various network slice types in the continuous feature space.

As shown in Figure 4, the distribution of samples in the space of key telemetric features (such as Packet delay and Packet Loss Rate) has a highly non-trivial structure. The classes form dense clusters that heavily overlap in certain areas. Linear algorithms draw straight decision boundaries through this “cloud” of points, which leads to underfitting in class mixing zones. In contrast, HistGradientBoosting builds complex, multidimensional, step-like decision boundaries and successfully segments this non-linear space.

Despite the high effectiveness of histogram gradient boosting, the algorithm still makes errors in about 5.3% of cases. To study the nature of these inaccuracies in detail, a confusion matrix was constructed on the test set.

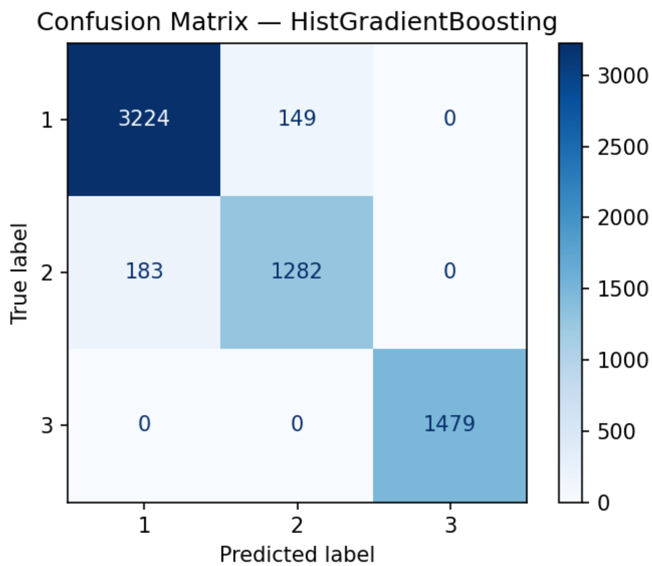

Figure 5: Confusion Matrix for the HistGradientBoosting model.

The analysis of Figure 5 confirms that errors are concentrated between conceptually similar network slices. From the perspective of 5G network architecture, these misclassifications have a physical justification: during short-term congestion or anomalies, telemetry profiles can become temporarily distorted, making classes statistically closer.

To make the balanced-accuracy figure interpretable and to expose how performance is distributed across the imbalanced classes, Table 5 reports per-class precision, recall, and F1 for the HistGradientBoosting model on the group-safe test set. Slice 3 is perfectly separable in the continuous telemetry space (precision = recall = 1.000), which is consistent with its isolated cluster in Figure 4. The residual error is confined almost entirely to the Slice 1/Slice 2 boundary: 149 Slice 1 samples are predicted as Slice 2 and 183 Slice 2 samples as Slice 1, yielding a slightly lower recall for the minority Slice 2 (0.8751). This asymmetry explains why the balanced accuracy (0.9436), which weights each class equally, sits marginally below the overall accuracy (0.9474): the dominant, well-separated classes pull the raw accuracy up, while the harder minority boundary is what the balanced metric surfaces.

Table 5. Per-class performance of HistGradientBoosting (group-safe test set)

Slice Type	**Precision**	**Recall**	**F1-score**	**Support**
1	0.9463	0.9558	0.9510	3373
2	0.8959	0.8751	0.8854	1465
3	1.0000	1.0000	1.0000	1479
Macro avg	0.9474	0.9436	0.9455	6317

4 Conclusions

This study demonstrates that 5G network slice types can be accurately inferred using a minimal set of objectively measurable QoS parameters. While linear classifiers provide a baseline level of utility, they are insufficient for capturing the non-linear, overlapping clusters inherent in network telemetry. Tree-ensemble methods proved decisively stronger than linear models by effectively segmenting complex decision spaces; the three ensembles performed comparably, with HistGradientBoosting reaching 94.74% accuracy and ExtraTrees marginally higher at 95.14%.

A primary contribution of this work is the implementation of a leakage-aware evaluation protocol, supported by a formal definition of the grouping variable, an explicit naive-versus-group-safe ablation, per-class performance reporting, and a full hyperparameter specification (Tables 2–5) to ensure reproducibility. By identifying that binary context flags in public datasets often act as "cheating" indicators, we proved that removing these labels is essential for developing models that generalize to real-world operations where such high-level information is unavailable. The remaining misclassifications in our best-performing model are attributed to the natural stochastic overlap of signals during network congestion rather than algorithmic failure. Ultimately, this research provides a robust framework for automated, intent-based slice management, emphasizing the need for rigorous data validation in applied machine learning for telecommunications.

5 Funding

This research was funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. BR24993211).

Author Contributions

Conceptualization, Y.N. and T.I.; methodology, Y.N. and S.A.; software, N.K. and A.M.; validation, S.A. and A.M.; formal analysis, Y.N. and N.K.; investigation, Y.N., N.K. and A.M.; resources, S.A. and B.K.; data curation, N.K. and A.M.; writing—original draft preparation, Y.N. and S.A.; writing—review and editing, N.K. and A.M.; visualization, N.K. and A.M.; supervision, B.K. and T.I.; project administration, T.I.; funding acquisition, T.I. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] 3GPP, "System architecture for the 5G System (5GS)," 3GPP TS 23.501 / ETSI TS 123 501, Release 18, v18.8.0.

[2] 3GPP, "Management and orchestration; Concepts, use cases and requirements for network slicing management," 3GPP TS 28.530 / ETSI TS 128 530, Release 18, v18.0.0.

[3] NGMN Alliance, “Description of Network Slicing Concept (Network Slicing Framework),” 2016.

[4] ETSI, “Network Slicing and NFV—Relationship and Touchpoints,” ETSI GR NFV-REL 010, v3.1.1, 2019.

[5] 5G Americas, “Network Slicing for 5G and Beyond,” white paper.

[6] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, “Network slicing in 5G: Survey and challenges,” *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017, doi: 10.1109/MCOM.2017.1600951.

[7] B. Han and H. D. Schotten, “Machine learning for network slicing resource management: A comprehensive survey,” arXiv, 2020, doi: 10.48550/ARXIV.2001.07974.

[8] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, “5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view,” *IEEE Access*, vol. 6, pp. 55765–55779, 2018, doi: 10.1109/ACCESS.2018.2872781.

[9] S. Malta, P. Pinto, and M. Fernández-Veiga, “Optimizing 5G network slicing with DRL: Balancing eMBB, URLLC, and mMTC with OMA, NOMA, and RSMA,” *J. Netw. Comput. Appl.*, vol. 234, p. 104068, Feb. 2025, doi: 10.1016/j.jnca.2024.104068.

[10] R. Dangi and P. Lalwani, “Feature selection based machine learning models for 5G network slicing approximation,” *Comput. Netw.*, vol. 237, p. 110093, Dec. 2023, doi: 10.1016/j.comnet.2023.110093.

[11] X. de Foy and A. Rahman, “Network slicing in logical 5G architecture in 3GPP,” IETF Internet-Draft, 2017.

[12] 3GPP, “5G network slice management,” overview page and related TS/TR pointers, 2023.

[13] A. K. Alnaim and K. M. Albarrak, “Latency-aware NFV slicing orchestration for time-sensitive 6G applications,” *Systems*, vol. 13, no. 11, p. 957, Oct. 2025, doi: 10.3390/systems13110957.

[14] A. Mohan Kumar, “Network slicing in 5G,” dataset, Kaggle.

[15] A. Donatti *et al.*, “Survey on machine learning-enabled network slicing: Covering the entire life cycle,” *IEEE Trans. Netw. Serv. Manage.*, vol. 21, no. 1, pp. 994–1011, Feb. 2024, doi: 10.1109/TNSM.2023.3287651.

[16] “Detection of operating system vulnerabilities and network traffic analysis methods,” *J. Math. Mech. Comput. Sci.*, vol. 121, no. 1, pp. 99–109, 2024, doi: 10.26577/JMMCS2024121110.

Information about authors:

Bolatzhan Kumalakov – CEO at the Astana IT University (Astana, Kazakhstan, e-mail: bolatzhan.kumalakov@astanait.edu.kz).
Nurislam Kassymbek – Scientific researcher at Computer Science laboratory at Al-Farabi Kazakh National University (Almaty, Kazakhstan, e-mail: nuryslam.qassymbek@gmail.com).
Serik Aibagarov – Scientific researcher at Computer Science laboratory at Al-Farabi Kazakh National University (Almaty, Kazakhstan, e-mail: awer1307dot@gmail.com).
Aksultan Mukhanbet – Scientific researcher at Computer Science laboratory at Al-Farabi Kazakh National University (Almaty, Kazakhstan, e-mail: mukhanbetaksultan0414@gmail.com).
Yedil Nurakhov – Scientific researcher at Computer Science laboratory at Al-Farabi Kazakh National University (Almaty, Kazakhstan, e-mail: y.nurakhov@gmail.com).
Timur Imankulov – PhD, associate professor, Al-Farabi Kazakh National University (Almaty, Kazakhstan, e-mail: imankulov.timur@gmail.com).

Авторлар туралы мәлімет

Болатжан Кумалаков – Astana IT University ректоры (Астана, Қазақстан, e-mail: bolatzhan.kumalakov@astanait.edu.kz).
Нурислам Касымбек – Әл-Фараби атындағы Қазақ ұлттық университетінің Компьютерлік ғылымдар зертханасының ғылыми қызметкері (Алматы, Қазақстан, электрондық пошта: nuryslam.qassymbek@gmail.com).
Серик Айбагаров – Әл-Фараби атындағы Қазақ ұлттық университетінің Компьютерлік ғылымдар зертханасының ғылыми қызметкері (Алматы, Қазақстан, электрондық пошта: awer1307dot@gmail.com).
Аксултан Муханбет – Әл-Фараби атындағы Қазақ ұлттық университетінің Компьютерлік ғылымдар зертханасының ғылыми қызметкері (Алматы, Қазақстан, электрондық пошта: mukhanbetaksultan0414@gmail.com).
Едиль Нурахов – Әл-Фараби атындағы Қазақ ұлттық университетінің Компьютерлік ғылымдар зертханасының ғылыми қызметкері (Алматы, Қазақстан, электрондық пошта: y.nurakhov@gmail.com).
Тимур Иманкулов – PhD, доцент, әл-Фараби атындағы Қазақ ұлттық университеті (Алматы, Қазақстан, e-mail: imankulov.timur@gmail.com).

Сведения об авторах

Болатжан Кумалаков – Ректор Astana IT University (Астана, Казахстан, e-mail: bolatzhan.kumalakov@astanait.edu.kz).
Нурислам Касымбек – Научный сотрудник лаборатории компьютерных наук Казахского национального университета имени аль-Фараби (Алматы, Казахстан, электронная почта: nuryslam.qassymbek@gmail.com).
Серик Айбагаров – Научный сотрудник лаборатории компьютерных наук Казахского национального университета имени аль-Фараби (Алматы, Казахстан, электронная почта: awer1307dot@gmail.com).
Аксултан Муханбет – Научный сотрудник лаборатории компьютерных наук Казахского

национального университета имени аль-Фараби (Алматы, Казахстан, электронная почта: mukhanbetaksultan0414@gmail.com).
Едиль Нурахов – Научный сотрудник лаборатории компьютерных наук Казахского национального университета имени аль-Фараби (Алматы, Казахстан, электронная почта: y.nurakhov@gmail.com).
Тимур Иманкулов – PhD, доцент, Казахский национальный университет имени аль-Фараби (Алматы, Казахстан, e-mail: imankulov.timur@gmail.com).

Received: February 27, 2026
Accepted: June 12, 2026