



Cluster-Based Random Forest Regression for Internet of Things Bandwidth Prediction Using Leakage-Free Features

Naek Parulian Hutagalung^{1*}, Bambang Tutuko², Ahmad Zarkasi³

^{1,2,3}Department of Computer Science, Faculty of Computer Science,
Universitas Sriwijaya, Indonesia

E-Mail: ¹naekparulian12@gmail.com, ²bambang_tutuko@unsri.ac.id, ³ahmadzarkasi@unsri.ac.id

Received May 12th 2026; Revised Jun 29th 2026; Accepted Jul 10th 2026; Available Online Jul 27th 2026

Corresponding Author: Naek Parulian Hutagalung

Copyright © 2026 by Authors, Published by Institut Riset dan Publikasi Indonesia (IRPI)

Abstract

Heterogeneous Internet of Things (IoT) deployments generate highly variable network traffic, yet most existing bandwidth management approaches rely on static provisioning or device-agnostic models that fail to capture per-device traffic heterogeneity. This study proposes a hybrid machine learning framework combining K-Means clustering and Random Forest regression to enable cluster-aware, leakage-free bandwidth consumption prediction for IoT environments. Using 994,145 flow records from the UNSW HomeNet dataset, the Log1p transformation was applied to address heavy-tailed traffic distributions, and direct bandwidth-derived features were strictly excluded to prevent data leakage. Devices were segmented into three traffic profiles (Low, Medium, and High), validated by a Silhouette Score of 0.4153 and a Davies–Bouldin Index of 1.0357. Within the structural constraints of this single-dataset, controlled setting, Random Forest regression achieved R^2 values above 0.99 across all clusters, with stable cross-validated performance, substantially outperforming Linear Regression, which achieved R^2 values as low as 0.0374 in the Medium cluster. Feature importance analysis revealed that Inter-Arrival Time is the primary driver of bandwidth consumption, superseding raw payload volume, with direct implications for temporally-driven feature design. These results indicate that cluster-aware, leakage-free models offer a viable architecture for IoT bandwidth management, with cross-domain generalization as a primary direction for future work.

Keywords: Bandwidth Prediction, Feature Importance, IoT Traffic, K-Means Clustering, Random Forest

1. INTRODUCTION

The continuous expansion of Internet of Things (IoT) deployments has introduced a growing diversity of connected devices into residential, industrial, and urban network environments. By the end of 2024, the global number of connected IoT devices had reached 18.5 billion, and this figure is projected to surpass 21 billion by the close of 2025, with long-range forecasts placing the total above 39 billion by 2030 [1]. This expansion encompasses device categories whose communication requirements differ by several orders of magnitude, from battery-constrained temperature sensors that transmit periodic bursts of a few bytes, to high-definition surveillance cameras sustaining multi-megabit per second flows around the clock [2]. When devices of such contrasting profiles coexist on a shared network, a uniform bandwidth policy that ignores per-device behavioral differences will inevitably over-provision capacity for low-demand endpoints while denying adequate resources to high-demand ones, degrading the overall Quality of Service (QoS) delivered to all device classes [3].

Network Traffic Prediction (NTP) has been extensively studied as a means of enabling proactive, demand-aware resource management, and machine learning approaches have largely displaced classical statistical models in this domain owing to their capacity to capture non-linear temporal patterns in traffic data [4], [5]. Among these, ensemble methods such as Random Forest have demonstrated favorable trade-offs between predictive accuracy, robustness to outliers, and interpretability through native feature importance quantification, properties particularly valuable in heterogeneous IoT traffic contexts [6], [7]. While gradient boosting methods such as Extreme Gradient Boosting (XGBoost) offer competitive accuracy, they provide less transparent feature attribution at the cluster level [8]. Recurrent architectures such as Long Short Term Memory (LSTM) require temporally ordered sequential inputs, which per-flow CICFlowMeter records do not satisfy [4]. Furthermore, Linear Regression is retained as the primary baseline in this study to serve as a structural diagnostic probe, consistent with established practice in comparative ML studies for network traffic prediction

[5]. Its empirical inability to capture the non-linear dynamics of IoT traffic provides direct justification for deploying more computationally intensive ensemble architectures, as demonstrated in cluster-aware IoT resource allocation frameworks [7]. However, the majority of existing NTP studies operate at the aggregate network level, treating the total traffic volume as a single time series and optimizing predictions for that consolidated signal [5], [9]. This aggregate perspective conceals the structural heterogeneity among device categories, making it impractical to derive differentiated allocation policies for individual device segments [10]. In heterogeneous IoT environments in particular, the range of flow characteristics is so wide that no single regression surface can simultaneously provide accurate predictions for low-traffic sensors and high-traffic multimedia devices, and attempts to do so generally yield models biased toward the dominant traffic class [11].

Several research efforts have explored segmentation-based approaches to address the heterogeneity of Internet of Things (IoT) traffic. Clustering methods, in particular K-Means, have been applied to group IoT devices based on shared behavioral characteristics before applying differentiated management policies, with studies demonstrating that cluster-aware schemes achieve higher prediction accuracy than device-agnostic baselines [7], [12]. Combined frameworks that chain K-Means segmentation with per-cluster supervised regression have reported improvements in resource allocation precision compared to uniform provisioning strategies [7], [8]. Nonetheless, a review of this body of work reveals three limitations that recur across studies. First, cluster-level generalization is rarely verified through stratified cross-validation, leaving reported performance metrics without statistical evidence of stability. Second, feature importance analysis at the per-cluster level is rarely reported, leaving the behavioral drivers of bandwidth consumption within each segment unexplained. Third, and most critically, features that are direct arithmetic transformations of the bandwidth target variable are frequently retained in the predictor set, introducing data leakage that inflates evaluation metrics beyond what would be observed in operational deployment [13], [14], [15].

Data leakage, defined as the inadvertent incorporation of target-derived or temporally forbidden information into model training, is a well-documented threat to the validity of machine learning systems across multiple domains [16], [17]. In the context of network traffic datasets generated by tools such as CICFlowMeter, features such as flow byte rate and packet rate are computed directly from total transferred bytes and flow duration, which are also the quantities used to derive the bandwidth consumption target. A model trained with such features does not learn a generalizable relationship between observable traffic patterns and bandwidth demand. Instead, it learns an identity transformation that produces artificially high accuracy on historical data but provides no predictive utility in real-time deployment, where the target value is unknown at prediction time [6], [16]. Despite the documented impact of leakage on model validity [17], its systematic treatment in IoT bandwidth prediction research has received little attention, and no prior cluster-based framework has explicitly reported the exclusion of bandwidth-derived features as a methodological design criterion [15].

This study addresses the identified gaps by proposing a cluster-aware, leakage-free bandwidth prediction framework for heterogeneous IoT networks, applied to 994,145 flow records from the UNSW HomeNet dataset [2], which contains 105 heterogeneous IoT and non-IoT devices. The framework combines Log1p-assisted K-Means clustering, strict exclusion of bandwidth-derived features [16], [17], and per-cluster Random Forest regression evaluated through stratified held-out testing and 5-Fold Cross-Validation [18]. The main contributions are as follows: (1). Proposing a cluster-aware and leakage-free bandwidth prediction framework that integrates K-Means clustering with cluster-specific Random Forest regression, while employing Log1p transformation and excluding bandwidth-derived features to ensure reliable model generalization. (2). Demonstrating that behavioral segmentation of heterogeneous IoT traffic into Low, Medium, and High bandwidth profiles reduces prediction bias and enables highly accurate bandwidth estimation, achieving R^2 values above 0.99 across all clusters. (3). Identifying Inter-Arrival Time as the most influential predictor of bandwidth consumption through cluster-level feature importance analysis, highlighting the critical role of temporal communication patterns in IoT traffic behavior. (4). Validating the robustness and reliability of the proposed framework using stratified 5-fold cross-validation across clusters with unequal traffic distributions.

The remainder of this paper is organized as follows. Section II presents the theoretical background, dataset, preprocessing procedures, and experimental methodology. Section III reports the clustering and prediction results, including feature importance analysis and cross-validation performance. Section IV discusses the findings and their implications for IoT bandwidth prediction. Finally, Section V concludes the paper and outlines future research directions.

2. MATERIALS AND METHOD

This study proposes a four-phase cluster-aware machine learning framework for predicting bandwidth consumption in heterogeneous Internet of Things (IOT) environments. The experimental pipeline encompasses dataset acquisition and preprocessing, leakage-free feature selection, behavioral traffic segmentation via K-Means clustering, per-cluster predictive model training and hyperparameter optimization, and rigorous performance evaluation through held-out testing and stratified cross-validation. The complete research

workflow is illustrated in Figure 1. All experiments were conducted on Google Colaboratory (free tier) with 12 GB RAM using Python 3.12, scikit-learn 1.6.1, and a fixed random seed of 42 throughout to ensure full reproducibility.

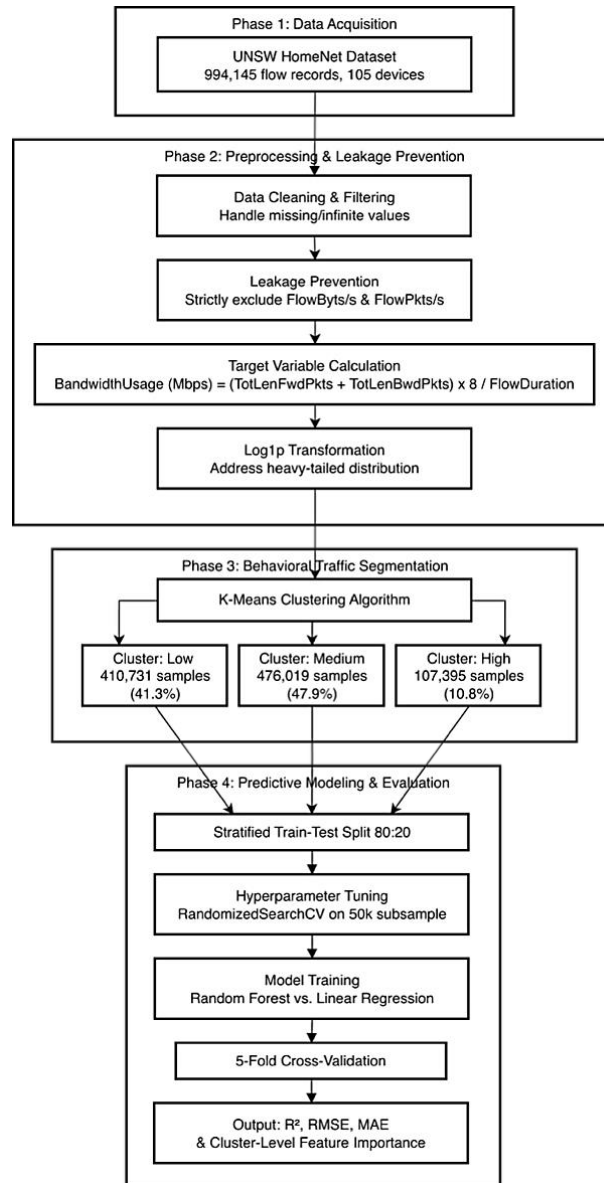


Figure 1. Research Methodology Flowchart of the Proposed Framework

2.1. Theoretical Background

K-Means Clustering: K-Means is an unsupervised partitioning algorithm that divides a dataset of N observations into k clusters by iteratively minimizing the within-cluster sum of squared Euclidean distances. The algorithm alternates between two steps: assigning each observation to its nearest centroid, and recomputing each centroid as the mean of all assigned observations, until convergence is reached [7], [12]. Its computational efficiency on large-scale datasets makes it well-suited for behavioral segmentation of IoT network flows prior to per-cluster predictive modeling.

Random Forest Regression: Random Forest is an ensemble learning method that constructs multiple decision trees on bootstrapped subsets of the training data and averages their predictions to reduce variance. Each tree is trained on a random subset of features at each split node, introducing decorrelation among trees and improving generalization [7]. For regression tasks, the ensemble output is the mean prediction of all constituent trees. Random Forest natively supports per-feature contribution analysis through impurity-based importance measures, making it particularly valuable for cluster-level interpretability in heterogeneous IoT traffic contexts [7].

Data Leakage: Data leakage occurs when information that would not be available at prediction time is incorporated into model training, producing inflated evaluation metrics that do not reflect operational

performance [16], [17]. In CICFlowMeter-derived datasets, features such as flow byte rate and packet rate are direct arithmetic functions of the same quantities used to derive the bandwidth consumption target, and their inclusion causes the model to learn a trivial identity mapping rather than a generalizable traffic-to-bandwidth relationship.

Feature Importance: Impurity-based feature importance in Random Forest measures the average reduction in node impurity weighted by the number of samples reaching each node, accumulated across all trees in the ensemble [7]. Higher importance scores indicate features that contribute more substantially to reducing prediction error. Conducting this analysis independently per cluster, rather than on the aggregate dataset, reveals which traffic characteristics are behaviorally dominant within each device class.

Bandwidth Prediction: Bandwidth consumption, expressed in Mbps, quantifies the volume of data transferred per unit time for a given network flow. Accurate per-cluster bandwidth prediction enables proactive resource allocation in Internet of Things (IoT) gateway systems, allowing differentiated traffic management policies to be applied across device segments with heterogeneous demand profiles [3], [11].

2.2. Dataset

The UNSW HomeNet dataset [2] was selected as the experimental benchmark for this study. The dataset comprises network traffic flow records captured from a residential smart home environment containing 105 heterogeneous Internet of Things (IoT) and non-IoT devices, including smart speakers, IP cameras, thermostats, and general-purpose computing devices. Raw flow statistics were extracted using CICFlowMeter, yielding 85 per-flow features that characterize each bidirectional network flow. Following data cleaning and filtering procedures described in Section 2.3, a total of 994,145 valid flow records were retained for analysis.

2.3. Preprocessing and Leakage-Free Feature Selection

The preprocessing pipeline was designed with explicit attention to preventing data leakage before any feature engineering or target derivation. The first step involved converting all candidate features to numeric values, removing rows with missing or non-finite values, and filtering out flow records with a FlowDuration of zero to prevent division-by-zero errors in subsequent target-variable computation.

Leakage prevention was enforced as a mandatory design constraint before the target variable was computed. Specifically, the features FlowByts/s and FlowPkts/s were strictly excluded from the predictor set at this stage. These features are direct arithmetic transformations of TotLenFwdPkts, TotLenBwdPkts, and FlowDuration, which are precisely the quantities used to derive the bandwidth consumption target. Retaining such features would cause the model to learn an identity transformation rather than a generalizable relationship between observable traffic patterns and bandwidth demand, leading to inflated evaluation metrics with no predictive utility in real deployment conditions [16], [17].

After leakage prevention, the target variable BandwidthUsage was computed from the remaining raw flow statistics using Equation (1).

$$\text{BandwidthUsage (Mbps)} = \frac{(\text{TotLenFwdPkts} + \text{TotLenBwdPkts}) \times 8}{\text{FlowDuration}} \quad (1)$$

where TotLenFwdPkts and TotLenBwdPkts denote the total byte lengths of forward and backward packets, respectively, the factor of 8 converts bytes to bits, and FlowDuration is expressed in microseconds. All records with a computed BandwidthUsage below zero were subsequently discarded.

Eight leakage-free input features were retained for both clustering and predictive modeling. These features capture temporal and volumetric flow characteristics that are observable in real time without knowledge of the target value, as summarized in Table 1.

Table 1. Leakage-Free Input Features Used for Clustering and Prediction

Feature	Category	Description
FlowIATMean	Temporal / Timing	Mean inter-arrival time between consecutive packets
FlowIATMax	Temporal / Timing	Maximum inter-arrival time (idle period detection)
FlowIATMin	Temporal / Timing	Minimum inter-arrival time (burst detection)
FlowIATStd	Temporal / Timing	Standard deviation of inter-arrival time
FwdPktLenMean	Volumetric / Size	Mean forward packet payload length
BwdPktLenMean	Volumetric / Size	Mean backward packet payload length
FwdPktLenStd	Volumetric / Size	Standard deviation of forward packet payload length
BwdPktLenStd	Volumetric / Size	Standard deviation of backward packet payload length

2.4. Log1p Transformation for K-Means Clustering

Internet of Things (IoT) network traffic follows a heavy-tailed distribution, wherein low-demand sensor flows vastly outnumber the high-throughput flows of data-intensive devices [13]. In the present dataset, BandwidthUsage has a mean of 0.083 Mbps yet reaches 34.6 Mbps at maximum, confirming extreme skewness. Under such conditions, the Euclidean distance metric used by K-Means assigns disproportionately large weights to high-magnitude outlier flows, causing the algorithm to collapse the vast majority of devices into a single cluster rather than producing meaningful behavioral segments [5].

To address this limitation, a Log1p transformation was applied to all eight input features prior to K-Means clustering, defined formally in Equation (2).

$$x' = \log_e(1 + x) \quad (2)$$

The transformation was preceded by a lower-bound clip operation, setting negative values to zero, as certain CICFlowMeter-derived IAT features may occasionally produce negative values due to timestamp resolution anomalies; $\log(1/p)$ of x is undefined for $x < -1$. The Log1p function was selected over alternative transformations because it handles zero-valued inputs gracefully ($\log1p(0) = 0$), preserves the monotonic ordering of the original data, and compresses extreme ranges proportionally without discarding any observations [13]. It is important to note that the Log1p transformation was applied solely for K-Means clustering. All subsequent model training, hyperparameter tuning, and performance evaluation were conducted on the original untransformed feature values to preserve interpretability and real-world predictive validity.

2.5. Behavioral Traffic Segmentation via K-Means Clustering

As described in Section 2.1, K-Means minimizes within-cluster inertia through iterative centroid assignment and recomputation, and was applied here to the Log1p-transformed feature space. To prevent scale dominance among attributes with varying magnitudes, the data were subsequently standardized using Z-score normalization (StandardScaler) as defined in Equation (3).

$$z = \frac{x - \mu}{\sigma} \quad (3)$$

where x represents the original feature value, μ is the mean, and σ is the standard deviation of the feature. Following this standardization, the algorithm partitioned the Internet of Things (IoT) devices into distinct behavioral traffic profiles. The number of clusters was set to $k = 3$, corresponding to Low, Medium, and High bandwidth utilization segments. This value was determined empirically using the Elbow Method, which plots the within-cluster sum of squares (inertia) against increasing values of k ; the inflection point at $k = 3$ indicated that additional clusters would yield diminishing reductions in inertia [7], [12]. The K-Means algorithm was run $n_init = 10$ times with distinct random centroids, and the configuration with the lowest inertia was retained.

Cluster labels were assigned based on the mean BandwidthUsage of each resulting cluster computed on the original, untransformed target values, ensuring that the ordering Low, Medium, and High reflects actual bandwidth consumption magnitudes rather than the compressed log-space distances used during clustering. The quality of the resulting partition was validated using two complementary metrics: the Silhouette Score, which measures intra-cluster cohesion relative to inter-cluster separation on a scale of negative one to positive one, where higher values indicate better-defined clusters; and the Davies-Bouldin Index, which quantifies the ratio of within-cluster scatter to between-cluster separation, where lower values indicate more compact and well-separated clusters [7]. The distribution of the resulting behavioral profiles and their validation scores are presented in Section 3.

2.6. Predictive Model Training and Hyperparameter Optimization

The full labeled dataset was partitioned into training and test sets using a stratified 80:20 split with a fixed random seed, ensuring that the proportional representation of each cluster was preserved across both partitions. The training set comprised 795,316 records, and the test set comprised 198,829 records. Per-cluster predictive models were trained independently for each of the three traffic segments, using the original untransformed feature values normalized by a per-cluster *StandardScaler* fitted exclusively on the corresponding training partition to prevent information leakage from the test set.

Two regression algorithms were trained and compared for each cluster: Linear Regression, serving as an interpretable baseline, and Random Forest Regression, which was expected to outperform the linear model given the non-linear dependencies and high variance of IoT traffic data [19], [6]. The theoretical foundations of Random Forest regression and feature importance quantification are described in Section 2.1. Random Forest hyperparameters were optimized using *RandomizedSearchCV* with $n_iter = 10$, with candidate configurations evaluated via 3-Fold cross-validation on a stratified subsample of 50,000 records per cluster. This subsampling strategy was adopted to ensure computational feasibility within the constraints of the

experimental environment while maintaining statistical representativeness, following established practice for large-scale machine learning experiments [18]. The hyperparameter search space is defined in Table 2.

Table 2. Random Forest Hyperparameter Search Space

Hyperparameter	Search Range
n_estimators	randint(100, 300)
max_depth	{10, 20, 30}
min_samples_split	randint(2, 10)
min_samples_leaf	randint(1, 5)
max_features	{sqrt, log2}

Following hyperparameter optimization on the subsample, the best-performing configuration for each cluster was used to retrain a final Random Forest model on the complete training partition for that cluster, ensuring that the predictive models benefit from the full available training data.

2.7. Evaluation Metrics

Model performance was assessed using three complementary regression metrics evaluated on the held-out test set for each cluster. The coefficient of determination (R^2) quantifies the proportion of variance in BandwidthUsage explained by the model, with a value of 1.0 indicating perfect prediction and values approaching 0 indicating that the model performs no better than a constant baseline. Root Mean Squared Error (RMSE) measures the average magnitude of prediction errors in the original unit of the target variable (Mbps), penalizing large errors more heavily than small ones. Mean Absolute Error (MAE) provides a linear measure of average prediction error that is more robust to outliers than RMSE. These metrics were computed per cluster to enable fine-grained comparison between Random Forest and Linear Regression across traffic segments of different distributional characteristics [9], [3].

To assess the generalization stability of the trained Random Forest models beyond a single train-test split, 5-Fold Cross-Validation was performed on the same stratified 50,000-record subsample used for hyperparameter tuning. Using an identical subsample for both tuning and cross-validation ensures distributional consistency between the two evaluation procedures, following the methodological guidance of Bischl et al. [18]. Cross-validated R^2 and RMSE are reported as mean plus or minus one standard deviation across the five folds, providing a measure of model stability under distributional variation within each cluster.

3. RESULTS AND DISCUSSION

This section presents the empirical findings of the proposed framework, structured sequentially across traffic segmentation validation, predictive model comparison, cross-validation stability analysis, cluster-level feature importance, and a synthesizing discussion contextualizing the results against prior work.

3.1. Traffic Behavioral Segmentation Results

The application of K-Means clustering ($k = 3$) on the Log1p-transformed and StandardScaler-normalized feature space successfully partitioned the 994,145 valid flow records into three behaviorally distinct profiles. The optimal value of k was empirically confirmed by the Elbow Method (Figure 2), which exhibited a pronounced inflection point at $k = 3$, indicating that additional clusters would yield only marginal reductions in within-cluster inertia.

As detailed in Table 3, the resulting cluster distribution reflects the inherently skewed nature of Internet of Things (IOT) network traffic. The Low and Medium profiles constitute the vast majority of the dataset, representing devices with intermittent to moderate transmission patterns, such as environmental sensors. Conversely, the high-bandwidth profile accounts for a distinct minority (10.8%), successfully capturing data-intensive endpoints like IP cameras whose flows constitute the heavy-tailed extreme of the traffic distribution.

Table 3. K-Means Cluster Distribution and Validation Metrics

Cluster Profile	Samples	Proportion (%)	Silhouette Score	Davies-Bouldin Index
Low	410,731	41.3	0.4153 (approx., n=20,000)	1.0357 (full dataset)
Medium	476,019	47.9		
High	107,395	10.8		
Total	994,145	100.0		

These metrics represent the overall dataset segmentation quality

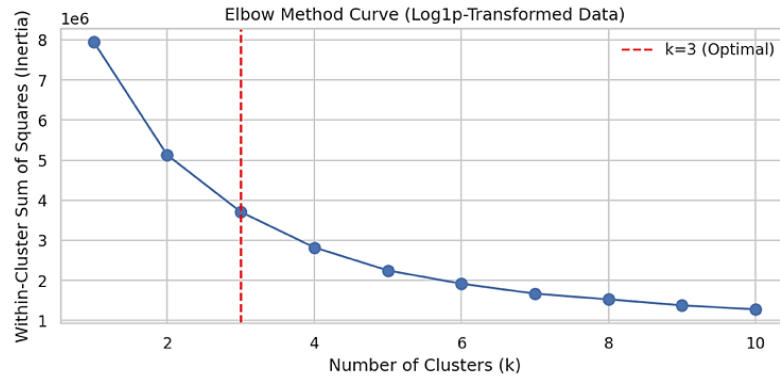


Figure 2. Elbow Method Evaluation for Determining the Optimal Number of Clusters

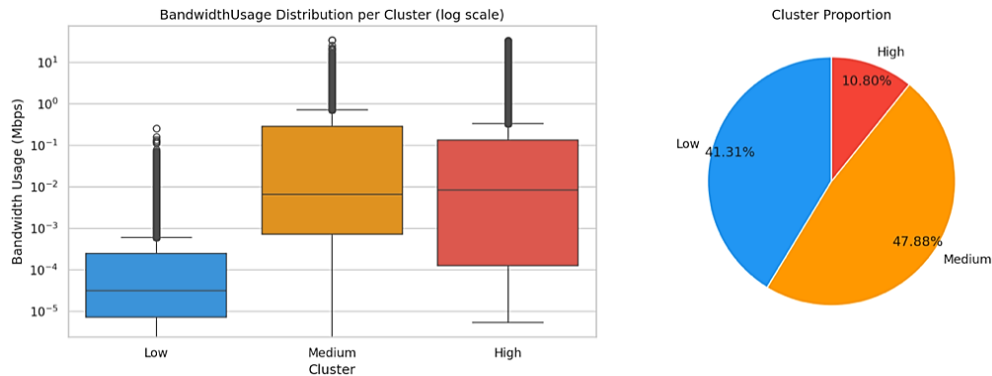


Figure 3. Visualization of the K-Means Behavioral Traffic Segmentation and Cluster Proportions

The segmentation quality was quantitatively validated through two complementary indices. The Silhouette Score of 0.4153 indicates a moderate but meaningful separation between the three traffic profiles on a scale of -1 to 1. The Davies-Bouldin Index of 1.0357 confirms that inter-cluster distances are sufficiently larger than intra-cluster scatter to produce coherent behavioral groupings. It is important to note that perfect cluster separation (Silhouette Score approaching 1.0) is not expected for continuous IoT traffic data, where no sharp behavioral boundaries exist between device categories. The obtained values are consistent with clustering outcomes reported in comparable IoT segmentation studies [7], [12] and are sufficient to support the per-cluster regression task that follows.

3.2. Predictive Model Performance on the Held-Out Test Set

Prior to evaluating predictive accuracy, the hyperparameter optimization phase yielded an architecturally notable result: the Random Forest algorithm produced identical optimal configurations across all three clusters, uniformly selecting a tree depth of 10 and 158 estimators with log2 feature sampling, as detailed in Table 2. The architectural significance of this convergence is addressed in Section 3.5.

The predictive performance of both models, evaluated on the 20% held-out stratified test set, is presented in Table 4. The Random Forest regression substantially outperformed the Linear Regression baseline across all clusters and metrics. The performance gap was most pronounced in the Medium cluster, where Random Forest attained an R^2 of 0.9938 against the baseline's 0.0374. The complete failure of Linear Regression to model the target variable in the Medium and High segments constitutes empirical evidence that the relationship between fundamental flow-level communication behaviors and bandwidth consumption is highly non-linear [6], [19].

Table 4. Predictive Model Performance on the Held-Out Test Set ($n = 198,829$)

Cluster	Model	R^2	RMSE (Mbps)	MAE (Mbps)	N Test
Low	Linear Regression	0.4242	0.002746	0.001393	82,146
Low	Random Forest	0.9993	0.000099	0.000011	82,146
Medium	Linear Regression	0.0374	0.377084	0.140700	95,204
Medium	Random Forest	0.9938	0.030270	0.004277	95,204
High	Linear Regression	0.2333	1.006854	0.322128	21,479
High	Random Forest	0.9912	0.107803	0.016855	21,479

Bold values indicate the best-performing model per cluster and metric.

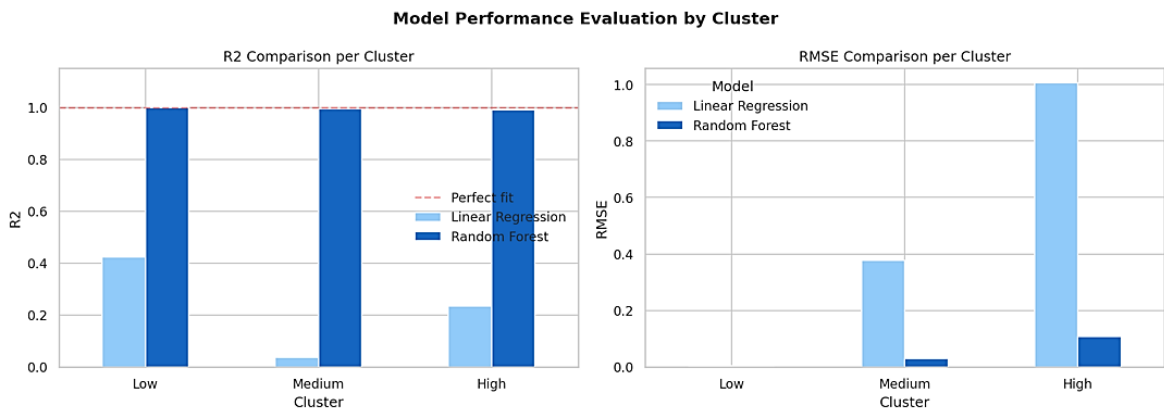


Figure 4. Comparative R^2 and RMSE Performance of Random Forest and Linear Regression per Cluster

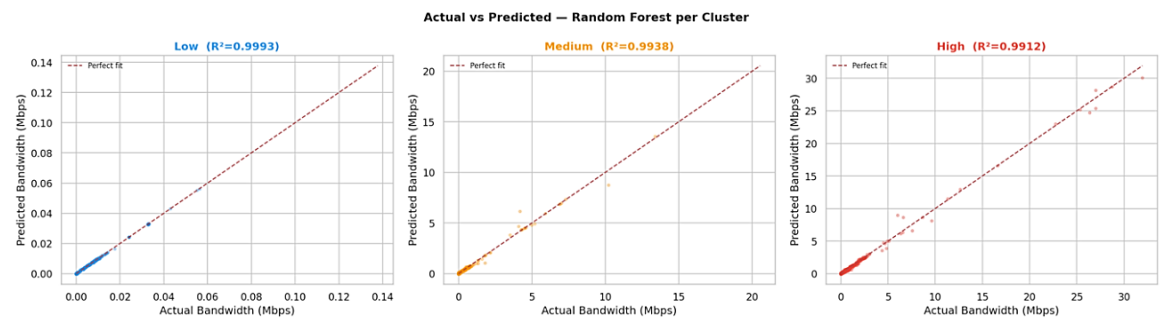


Figure 5. Actual Versus Predicted Bandwidth Consumption Across the Three Cluster Segments (Random Forest)

To formally validate the statistical significance of this performance gap, a Wilcoxon signed-rank test was applied to the absolute prediction errors of both models across all test records within each cluster, with effect sizes computed using the rank-biserial correlation coefficient to account for potential inflation of statistical significance under large sample sizes. Additionally, 95% bootstrap confidence intervals for R^2 were computed on subsamples of 10,000 records per cluster using 1,000 resamples. The complete statistical validation results are presented in Table 5.

Table 5. Statistical Significance and Bootstrap R^2 Confidence Intervals (Random Forest vs Linear Regression)

Cluster	Wilcoxon p-value	Effect Size r	Interpretation	95% CI R^2
Low	< 0.001	0.8656	Large	[0.9997, 0.9998]
Medium	< 0.001	0.8649	Large	[0.9967, 0.9989]
High	< 0.001	0.8559	Large	[0.9782, 0.9959]

Effect size interpreted per Cohen [20]. Bootstrap CI computed using 1,000 resamples, $n = 10,000$ per cluster.

The uniformly large effect sizes ($r > 0.85$) across all three clusters indicate that the performance superiority of Random Forest over Linear Regression is not only statistically significant but also practically meaningful, as the magnitude of error reduction is substantial enough to be consequential in real-world bandwidth allocation decisions. The narrow bootstrap confidence intervals further confirm that these conclusions are stable and not sensitive to the specific data partition used for evaluation.

3.3. Cross-Validation and Model Generalization Stability

To verify that the high R^2 scores observed on the held-out test set were not the product of a favorable random data partition, the trained Random Forest models were subjected to 5-Fold Cross-Validation on a stratified subsample of 50,000 records per cluster, identical to the subsample used for hyperparameter tuning. The cross-validation results are presented in Table 6.

As shown in Table 6, all three clusters maintained cross-validated R^2 values exceeding 0.93, confirming that the leakage-free predictor set provides robust forecasting capabilities independent of specific data partitions. While the Low and High clusters demonstrated tight R^2 stability, the Medium cluster exhibited a

relatively larger RMSE standard deviation. This fluctuation is not a model instability artifact but rather a reflection of the inherently diverse, transitional nature of the devices contained within this segment [5], [13].

Table 6. 5-Fold Cross-Validation Results of Random Forest per Cluster ($n = 50,000$)

Cluster	Mean R ² (plus or minus Std)	Mean RMSE (plus or minus Std) (Mbps)	Mean MAE (plus or minus Std) (Mbps)	N Samples
Low	0.9879 (0.0197)	0.000311 (0.000288)	0.000016 (0.000003)	50,000
Medium	0.9324 (0.0729)	0.096976 (0.065357)	0.005859 (0.000746)	50,000
High	0.9712 (0.0300)	0.176392 (0.104089)	0.018588 (0.003228)	50,000

Values in parentheses represent the standard deviation across the five folds.

3.4. Cluster-Level Feature Importance Analysis

A principal advantage of the Random Forest algorithm is its native capacity to quantify per-feature contributions through mean decrease in impurity (Gini importance). The cluster-level feature importance scores are summarized in Table 7, ordered by importance variance across clusters to highlight the most behaviorally discriminative features.

Table 7. Random Forest Feature Importance Scores per Cluster (Ordered by Variance)

Feature	Low	Medium	High	Variance
FlowIATMean	0.1100	0.4520	0.4486	0.0386
FlowIATMax	0.3217	0.2924	0.0251	0.0267
FwdPktLenMean	0.2776	0.0191	0.0653	0.0190
FlowIATStd	0.0248	0.0202	0.2435	0.0163
BwdPktLenMean	0.2204	0.0835	0.1779	0.0049
FlowIATMin	0.0179	0.0910	0.0002	0.0023
BwdPktLenStd	0.0095	0.0361	0.0076	0.0003
FwdPktLenStd	0.0180	0.0058	0.0318	0.0002

Bold values indicate the top-ranked feature per cluster. Variance measures importance dispersion across the three clusters.

The analysis reveals two fundamentally different behavioral regimes (Table 7 and Figure 6). In the Medium and High clusters, Inter-Arrival Time metrics (specifically FlowIATMean and FlowIATStd) completely dominate the predictive decision boundaries. This indicates that data-intensive devices are characterized primarily by dense, consistent packet transmission frequencies rather than individual payload sizes. Conversely, the low-bandwidth cluster is heavily influenced by volumetric features (FwdPktLenMean and BwdPktLenMean) alongside FlowIATMax. This finding directly corresponds to the physical nature of low-power duty-cycled sensors, where the bandwidth footprint is determined by sparse payloads separated by dominant idle periods.

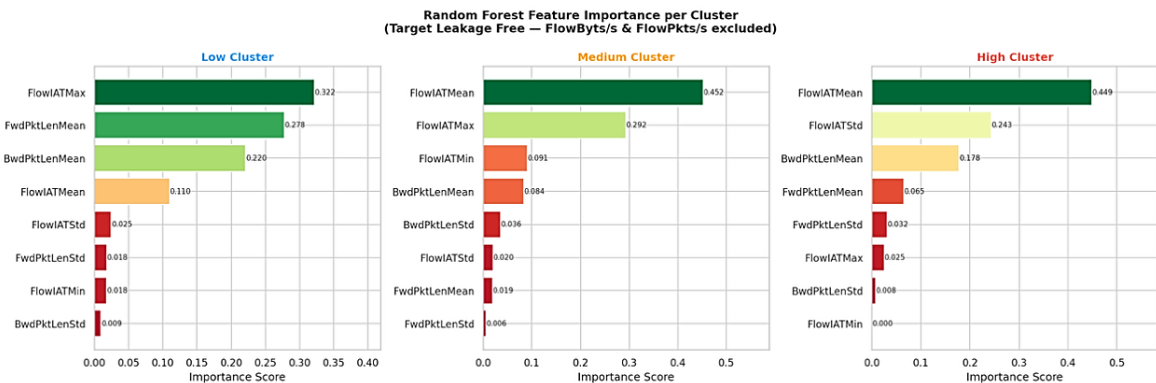


Figure 6. Random Forest Feature Importance Profiles Across the Three Bandwidth Clusters

3.5. Discussion

The empirical results collectively address all four research contributions established in the introduction of this study and warrant discussion on three interconnected dimensions: the validity of the near-perfect predictive accuracy, the structural significance of hyperparameter convergence, and the implications of the feature importance findings for Internet of Things (IoT) network engineering. Validity of Near-Perfect Accuracy ($R^2 = 0.9993$ and $R^2 > 0.99$). The R^2 values exceeding 0.99 across all test clusters, particularly the

R^2 of 0.9993 in the Low cluster, are values that legitimately invite scrutiny in any machine learning evaluation. In predictive network modeling, such performance levels frequently indicate residual data leakage or overfitting rather than genuine generalization. This study directly addresses both concerns through multiple independent lines of evidence.

Regarding data leakage: the two features most directly linked to bandwidth consumption in CICFlowMeter datasets, FlowByts/s and FlowPkts/s, were explicitly excluded from the feature set at the preprocessing stage, prior to target variable computation. The rigorous application of this leakage-free constraint is the central methodological contribution of this study and has been validated against the leakage taxonomy described in [16] and [17]. Regarding overfitting: the cross-validation results provide definitive evidence against memorization. Furthermore, the exceptionally high accuracy in the Low cluster ($R^2 = 0.9993$) is attributable to the homogeneous, structurally constrained nature of duty-cycled sensors. The model accurately approximates a relationship that is genuinely deterministic within this narrow distributional range, rather than memorizing noise [7].

Structural Significance of Hyperparameter Convergence. The convergence of identical optimal hyperparameters across all three clusters (max_depth=10, n_estimators=158) is architecturally meaningful rather than coincidental. A tree of depth 10 can partition the feature space into up to 1,024 distinct decision regions, which is empirically sufficient to approximate this bounded function regardless of the traffic magnitude at which it is evaluated. The convergence therefore reflects a fundamental invariance in the structural complexity required to model bandwidth from leakage-free features and provides evidence that the proposed framework does not require cluster-specific architectural redesign to achieve high-fidelity predictions across heterogeneous traffic profiles [19].

Feature Engineering Implications and Comparison to Prior Work. The identification of Inter-Arrival Time as the primary driver of bandwidth consumption in high-throughput clusters extends and refines the findings of prior cluster-based IoT resource allocation studies. Derakhshanfard et al. [7] and Koesnadi and Chung [12] demonstrated that behavioral clustering improves allocation precision but did not perform cluster-level feature importance analysis to identify which traffic characteristics drive the prediction within each segment. Pundir et al. [19] employed linear models for bandwidth prediction in IoT networks, which the present study demonstrates are inadequate for the non-linear traffic structures observed in the Medium and High clusters. To the best of our knowledge, the present work is among the first to provide cluster-stratified feature importance analysis on a large-scale heterogeneous IoT dataset with explicit leakage prevention, revealing that temporal communication frequency and payload volume serve as complementary rather than interchangeable predictors depending on the traffic intensity regime of the target device class.

Limitations and Future Directions. While the results demonstrate strong predictive performance across all traffic segments, several limitations warrant acknowledgment. The cluster imbalance, with the High bandwidth profile accounting for only 10.8% of the dataset, may limit the model's sensitivity to rare but high-impact traffic surges from devices poorly represented in the training data. Additionally, the current framework assumes static cluster membership; in practice, IoT devices may transition between behavioral profiles over time as their operational state changes, requiring periodic re-clustering or an online learning component. Future work should evaluate the framework under real-time deployment conditions, incorporate temporal drift detection for cluster reassignment, and extend the leakage-free feature analysis to additional IoT benchmark datasets to assess cross-dataset generalizability [3], [11].

4. CONCLUSION

This study successfully formulated and evaluated a cluster-aware, leakage-free machine learning framework for predicting bandwidth consumption in heterogeneous Internet of Things (IoT) environments. By applying Log1p-assisted K-Means clustering, the proposed architecture effectively isolated the heavy-tailed, high-throughput traffic profile (10.8% of the dataset) from the majority of low-demand sensors. The quality of this behavioral segmentation was quantitatively validated, achieving a Silhouette Score of 0.4153 and a Davies-Bouldin Index of 1.0357, which confirms coherent device groupings suitable for subsequent regression tasks.

The empirical results demonstrate that bandwidth consumption can be predicted with high accuracy, with the per-cluster Random Forest models achieving R^2 scores exceeding 0.99 across the held-out test segments. Crucially, this performance was attained without leakage-inducing features such as byte or packet rates. To explicitly validate this performance and ensure the models' statistical reliability across cluster segments of unequal size, 5-Fold Cross-Validation was applied on stratified subsamples. The models maintained high generalization stability across all clusters, as detailed in Table 6. Furthermore, the cluster-level feature importance analysis revealed a critical behavioral dichotomy: temporal communication patterns, specifically Inter-Arrival Time (IAT), are the dominant drivers of bandwidth consumption for data-intensive devices, whereas raw payload volume primarily dictates the network footprint of low-power sensors.

The empirical outcomes of this study provide a methodological foundation for cluster-aware bandwidth management in heterogeneous IoT environments. The demonstrated utility of temporally-driven, leakage-free

features establishes an objective basis for lightweight resource allocation design. However, it is important to acknowledge that this study was conducted under offline batch-inference conditions, and the computational overhead of deploying the trained Random Forest models under real-time streaming constraints has not been measured. Preliminary inference benchmarking conducted on the experimental environment (Google Colaboratory CPU) yielded per-sample prediction latencies of 10.20 μ s, 18.88 μ s, and 18.97 μ s for the Low, Medium, and High clusters respectively, corresponding to throughputs of approximately 98,065, 52,979, and 52,714 samples per second. These estimates suggest compatibility with gateway-level batch inference under moderate concurrency. However, a formal evaluation of end-to-end latency and memory footprint under live concurrent IoT flow conditions remains a necessary prerequisite before operational readiness in resource-constrained gateway systems can be claimed [3], [11]. Building upon these methodological foundations, future research should prioritize three directions. First, deploying this cluster-aware prediction framework within a Software-Defined Networking (SDN) controller to evaluate end-to-end inference latency under live traffic conditions [21]. Second, evaluating the leakage-free framework on cross-domain IoT datasets to assess generalizability across different network topologies and device ecosystems. Third, incorporating online re-clustering mechanisms to address concept drift in dynamically evolving IoT environments.

REFERENCES

- [1] S. H. Alsamhi et al., "Machine Learning for Smart Environments in B5G Networks: Connectivity and QoS," *Comput. Intell. Neurosci.*, vol. 2021, no. 1, p. 6805151, Jan. 2021, doi: <https://doi.org/10.1155/2021/6805151>.
- [2] M. M. Rahman, F. Bouhafs, S. A. Hoseini, and F. den Hartog, "UNSW HomeNet: A network traffic flow dataset for AI-based smart home device classification," *Comput. Ind. Eng.*, vol. 204, no. C, Art. no. 111041, Jun. 2025, doi: [10.1016/j.cie.2025.111041](https://doi.org/10.1016/j.cie.2025.111041).
- [3] Z. Ali et al., "Edge Based Priority-Aware Dynamic Resource Allocation for Internet of Things Networks," *Entropy*, vol. 24, no. 11, Art. no. 1607, 2022, doi: [10.3390/e24111607](https://doi.org/10.3390/e24111607).
- [4] O. Aouedi, V. A. Le, K. Piamrat, and Y. Ji, "Deep Learning on Network Traffic Prediction: Recent Advances, Analysis, and Future Directions," *ACM Comput. Surv.*, vol. 57, no. 6, Art. no. 151, pp. 1–37, Feb. 2025, doi: [10.1145/3703447](https://doi.org/10.1145/3703447).
- [5] I. Lohrasbinasab, A. Shahraki, A. Taherkordi, and A. Delia Jurcut, "From statistical- to machine learning-based network traffic prediction," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, p. e4394, Apr. 2022, doi: <https://doi.org/10.1002/ett.4394>.
- [6] M. T. Abdelaziz et al., "Enhancing Network Threat Detection with Random Forest-Based NIDS and Permutation Feature Importance," *Journal of Network and Systems Management*, vol. 33, no. 1, p. 2, 2024, doi: [10.1007/s10922-024-09874-0](https://doi.org/10.1007/s10922-024-09874-0).
- [7] N. Derakhshanfard, L. Hosseinzadeh, F. R. Jafari, and A. Ghaffari, "Intelligent resource allocation in internet of things using random forest and clustering techniques," *Sci. Rep.*, vol. 15, no. 1, p. 30522, 2025, doi: [10.1038/s41598-025-15931-8](https://doi.org/10.1038/s41598-025-15931-8).
- [8] K. Saurabh, S. Mahapatra, and M. M. Tripathi, "An adaptive bandwidth scheduling algorithm for IoT devices using machine learning techniques," *Franklin Open*, vol. 14, Art. no. 100492, 2026, doi: <https://doi.org/10.1016/j.fraope.2026.100492>.
- [9] C. Pan, Y. Wang, H. Shi, J. Shi, and R. Cai, "Network Traffic Prediction Incorporating Prior Knowledge for an Intelligent Network," *Sensors*, vol. 22, no. 7, Art. no. 2674, 2022, doi: [10.3390/s22072674](https://doi.org/10.3390/s22072674).
- [10] K. Murphy, A. Lavignotte, and C. Lepers, "Fault Prediction for Heterogeneous Telecommunication Networks Using Machine Learning: A Survey," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 2515–2538, 2024, doi: [10.1109/TNSM.2023.3340351](https://doi.org/10.1109/TNSM.2023.3340351).
- [11] W. Symor and Ł. Falas, "Ensuring Reliable Network Communication and Data Processing in Internet of Things Systems with Prediction-Based Resource Allocation," *Sensors*, vol. 25, no. 1, Art. no. 247, 2025, doi: [10.3390/s25010247](https://doi.org/10.3390/s25010247).
- [12] F. X. K. Koesnadi and S.-H. Chung, "Improved Cell Allocation Strategies Using K-Means Clustering in Congested 6TiSCH Environments," *Sensors*, vol. 24, no. 17, Art. no. 5608, 2024, doi: [10.3390/s24175608](https://doi.org/10.3390/s24175608).
- [13] Y. Xiong, S. Dong, R. Liu, F. Shi, and X. Jing, "IoT network traffic classification: a deep learning method with Fourier transform-assisted hyperparameter optimization," *Front. Phys.*, vol. 11, Art. no. 1273862, Oct. 2023, doi: [10.3389/fphy.2023.1273862](https://doi.org/10.3389/fphy.2023.1273862).
- [14] F. Afifi, F. Zaki, H. Hanif, N. Aqil, and N. B. Anuar, "Transformer-based tokenization for IoT traffic classification across diverse network environments," *PeerJ Comput. Sci.*, vol. 11, p. e3126, Aug. 2025, doi: [10.7717/peerj-cs.3126](https://doi.org/10.7717/peerj-cs.3126).
- [15] V. A. Phan, J. Jerabek, and L. Malina, "Comparison of Multiple Feature Selection Techniques for Machine Learning-Based Detection of IoT Attacks," in *Proc. 19th Int. Conf. Availability, Reliability and Security (ARES '24)*, Vienna, Austria, Jul. 2024, Art. no. 157, pp. 1–10, doi: [10.1145/3664476.3670440](https://doi.org/10.1145/3664476.3670440).

- [16] M. A. Bouke and A. Abdullah, "An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability," *Expert Syst. Appl.*, vol. 230, Art. no. 120715, Nov. 2023, doi: <https://doi.org/10.1016/j.eswa.2023.120715>.
- [17] A. Apicella, F. Isgrò, and R. Prevete, "Don't push the button! Exploring data leakage risks in machine learning and transfer learning," *Artif. Intell. Rev.*, vol. 58, no. 11, p. 339, 2025, doi: [10.1007/s10462-025-11326-3](https://doi.org/10.1007/s10462-025-11326-3).
- [18] B. Bischl et al., "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining and Knowledge Discovery*, vol. 13, no. 2, p. e1484, Mar. 2023, doi: <https://doi.org/10.1002/widm.1484>.
- [19] S. Pundir, G. Bhandari, and S. Tyagi, "Dynamic Bandwidth Allocation Using Linear Regression Model in IOT with Machine Learning Techniques," in *Micro-Electronics and Telecommunication Engineering*, D. K. Sharma, S. L. Peng, R. Sharma, and G. Jeon, Eds., Lecture Notes in Networks and Systems, vol. 617. Singapore: Springer, 2023, pp. 597–609. doi: [10.1007/978-981-19-9512-5_54](https://doi.org/10.1007/978-981-19-9512-5_54).
- [20] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.
- [21] R. Kulandaivel, R. MANIKANDAN, S. Veerappampalayam Easwaramoorthy, and J. Cho, "Gauss Markov and Flow Balanced Vector Radial Learning network traffic classification on IoT with SDN," *PLoS One*, vol. 19, no. 10, Art. no. e0308052, Oct. 2024, doi: [10.1371/journal.pone.0308052](https://doi.org/10.1371/journal.pone.0308052).