# Random Probit Regressive Decision Forest Classification based IoT aware Content Caching with Healthcare Data

R. Sangeetha and T.N. Ravi

***Abstract*—Information-Centric Networking (ICN) is promising system architecture for distributing popular information across the network. Its important feature is the node cache. ICN with caching is a very promising featured network structural design. There are numerous cache nodes distributed in the ICN. Excessive consumption of cache information on a tremendous number of nodes dissipates a large storage space and causes data redundancy, which reduces the cache hit rate. Therefore, an efficient caching deployment approach is required to improve the cache hit rate. A novel RAndom probit regressive Bucklin DEcision Forest classifier (RADEF) technique is introduced in Information-Centric Networking (ICN) for minimal network latency and higher cache hit rate. In RADEF Technique, probit regression and random forest classification processes are carried out for ICN with healthcare patient data. First, the patient information is collected from different IoT devices and registered. Every request to the router node is analyzed in the content storage (CS) by using a random probit regressive Bucklin decision forest classifier. The Bucklin decision forest classifier is an ensemble technique that includes a set of weak learners as decision trees (i.e., probit regression tree) and IoHT data are selected randomly for each decision tree. The Probit regression tree is constructed to analyze the patient healthcare data request search in the content storage (CS) of the router node in the ICN network by using a simple matching coefficient. If the copy is present in the content storage, the particular router nodes are chosen and deliver the content. If the copy is absent, the patient information is stored in the cache. Then, the weak learner's results are combined to make a strong output. Then the votes are generated for each decision tree. The votes of all decision trees are combined to identify the majority votes of data for classification by minimizing the error using the Bucklin voting method. In this way, content catching is effectively performed in ICN with minimum latency and a higher cache hit ratio. Experimental evaluation is carried out on factors such as cache hit rate, network latency, average request length, average response time, server traffic ratio, and hop reduction ratio regarding respect to the number of patient healthcare data. The analyzed results demonstrate the superior performance of our proposed RADEF technique when compared with existing methods.***

***Index Terms*—Information-Centric Networking (ICN), IoHT, Content catching, Probit regression, Probit decision tree, Bucklin decision forest classifier, Bucklin voting method, Content storage, Cache hit ratio**

Manuscript received on 9 Oct 2023, revised on 4 Apr 2024

R.Sangeetha is a Research Scholar of PG & Research Department of Computer Science, Thanthai Periyar Government Arts and Science College, Bharathidasan University, Tiruchirappalli, India - 620 023 (e-mail: sangeethar.scholar@gmail.com)

T.N.Ravi is an Associate Professor of PG & Research Department of Computer Science, Thanthai Periyar Government Arts and Science College, Bharathidasan University, Tiruchirappalli, India - 620 023 (e.mail: proftnravi@gmail.com)

## I. Introduction

ICN (Information-Centric Networking) has gained popularity as a network that mainly focuses on transmitted and received data. The Internet of Things applies the characteristics of content caching which is a typical feature of ICN. Therefore, designing an efficient caching strategy for IC-IoT is able to improve the efficiency of smart IoT devices. A Pre-Caching Strategy based on the Relevance of smart device request Content, called PCSRC was developed in [1] to improve the cache hit ratio and minimize the traffic ratio. But the network latency was not reduced. In [2] a cooperative caching method was designed to permit distributed edge servers for cooperating with each other. However, the cache hit rate was not focused.

The context-based caching mechanism was developed in [3] to achieve the content availability and network efficiency. But the average request length was not considered. A Co-adjuvant Caching Joint Request Forwarding in Information-Centric Networks (CCJRF-ICN) was developed in [4] to reduce the content discovery delay, content server load, and network congestion. The average request length was not minimized by ICN with the communication model. An effective isotonic regressive adaptive boost classification-based information centric network (IRABC-ICN) model was developed in [5] with IoHT data.

A Content Placement based on Normalized Node Degree and Distance (CPNDD) was introduced in [6] for content caching. However, the dimensions and network characteristics were not analyzed to improve the caching decisions and certainly minimize the complexity. A cooperative caching method was introduced in [7] for information-centric IoT networks to optimize the cache hit rate and decrease the network delay. But the cooperative caching scheme was not supported the IoT with Named Data Networking (NDN). ICN-based caching strategies were developed in [8] to minimize the content retrieval latency and improve the cache hit ratio with a focus on IoT-based environments. However, the average request length was not reduced.

An efficient content store-based caching strategy was introduced in [9] to improve the cache hit ratio, and stretch ratio and minimize the content retrieval latency. But it failed to integrate caching, forwarding, and optimizations with IoT technology. A fog-assisted healthcare IoT system was introduced in [10] to attain rapid patient data retrieval with minimum latency. However, the analysis of the cache hit ratio was not performed.

An ICN-IoT communication model was developed in [11] for a high dynamic platform. However, an efficient mobility-aware forwarding method was not applied to update the forwarding status and information while maintaining distributed in-caching and processing features. An Edge Linked Caching (ELC) method was introduced in [12] for information-centric IoT to minimize the response latency. However, the performance of ELC failed to test the network with rapid mobility. A Central Control Caching (CCC) method was developed in [13] for energy-efficient IoT content caching. The method minimizes the response latency and increases the cache hit but it failed to extend the scheme with the mobility of the key feature of IoT devices.

### *A. Major Contributions*

In order to overcome the existing issues, a novel RADEF technique is introduced with the following novel contributions.

- To improve the caching performance in ICN-IOT, a novel RADEF is introduced. That is based on the random probit regressive Bucklin decision forest classifier.
- To increase cache hit rate, RADEF uses the random probit regressive Bucklin decision forest classifier that combines the results of the weak learner as a probit regression tree to improve content delivery by analyzing the incoming patient healthcare data request in the content storage of the router with the help of simple matching coefficient. If the content is present, the requested data are delivered with minimum response time resulting in reducing the network latency.
- Finally, comprehensive experiment evaluations are carried out to estimate the performance of the RADEF technique and other techniques along with the various metrics.

### *B. Outline of paper*

The rest of the article is organized into six dissimilar sections as follows. Section 2 reviews the related works. Section 3 provides a brief description of the proposed RADEF with a neat architectural diagram. Section 4 describes the experimental settings with the dataset description. In section 5, the performance results of the proposed RADEF with existing methods are discussed for different metrics. Finally, Section 6 concludes the paper.

## II. Related Works

A double-layer network structure cache robustness strategy (CRS) was introduced in [14] based on content popularity and node significance. A Packet Update Caching (PUC) method was introduced in [15] for Energy Efficient IoT-based Information-Centric Networking. But the performance of a higher cache hit rate was not archived. A content popularity ranking (CPR) method was developed in [16] for content caching and minimizing the content delivery time.

A new content caching method was developed in [17] based on the hop count and bandwidth parameters to decrease the content redundancy and caching operations. However, the performance of the proposed strategy failed to analyze mobility-based networks and recent network topologies. A Context-based Cache Admission Policy (ctxCAP) was introduced in [18] to deal with mobile environments for enhancing the performance of caches.

An enhanced ICN-IoT content caching method was introduced in [19] by enabling Artificial Intelligence (AI)-based collaborative filtering to support heterogeneous IoT architecture. However, content retrieval latency was not reduced. A new ICN-based testbed approach was designed in [20] for naming, routing, and caching that provide mobility support. An efficient caching technique named PoolCache was developed in [21] for the effective caching capacity of nodes. However, the latency was not minimized. The content availability at a network was derived in [22] for ICN network topology. But it failed to analyze the content availability in large-scale ICN networks to investigate the impact of the network size on the content availability. A novel method called "NB-Cache" was developed in [23] to address CS's performance to improve the throughput and minimize the latency. In [24], the combination of elliptic curve cryptography-based identity-based cryptosystems and edge nodes was implemented in order to securely manage the patients' health data. In [25], Improved Context-Aware Data Fusion and Enhanced Recursive Feature Elimination Model was implemented for IoT-based patient health data.

## III. Proposal Methodology

Information-Centric Networking (ICN) is an assured network architecture used for efficiently delivering content. The Information-Centric Internet-of-Things (IC-IoT) connect more devices to the Internet, which allows for several significant applications like digital health to become a reality. Excessive deployment of cache information on a large number of devices wastes more storage space and directs data redundancy, which decreases the cache hit rate. Caching is the major key solution to this problem. Therefore, a novel caching deployment approach is required for cache deployment strategy and cache replacement strategy to greatly improve the throughput and performance of ICN and minimize network latency. The efficiency of caching also directly determines the overall performance of ICN. Its focus is on the delivery of data objects to the end user rather than communication between the client and host. To address these aspects, in this work a novel method called, the RADEF technique for accurate information caching with a higher hit ratio is proposed. Figure 1 shows the block diagram of the proposed RADEF technique for accurate information catching in IoHT.

Figure 1 depicts the architecture diagram of the proposed RADEF technique to provide accurate information catching with lesser latency and higher hit ratio by implementing the healthcare scenario. The healthcare architecture comprises two entities such as patients $P_1, P_2, P_3, \ldots P_n$ and who send the patient healthcare data request $DR_1, DR_2, DR_3 \ldots DR_n$ collected by the IoT device.

The ICN includes a router node 'RN' and cache called Content Storage ('CS'). ICN routers are not only used for data forwarding but also cache the data during transit, thereby improving hop count and minimize the delay. A cache is a high-speed content storage layer which stores a subset of data request. The efficiency of caching effectively improves the overall performance of ICN.

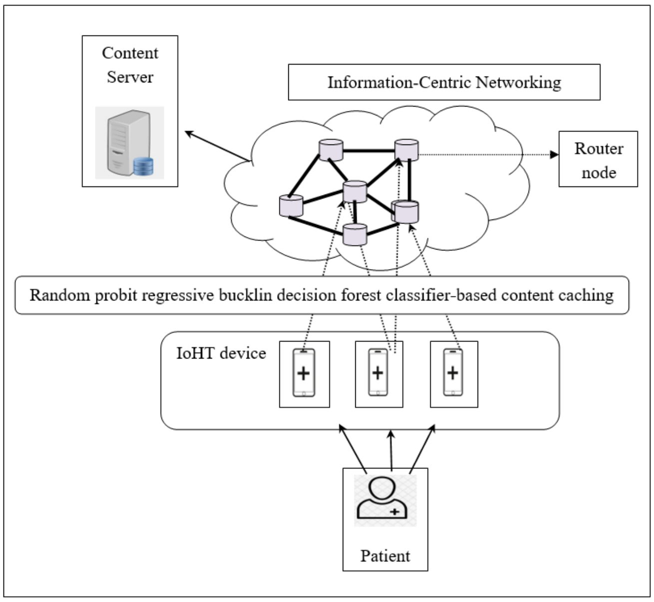


Fig 1. Architecture diagram of proposed RADEF technique

The patient healthcare data request $DR_1$, $DR_2$, $DR_3$ . . . $DR_n$ are obtained from maternal health care dataset through the IoHT device for predicting health risks of pregnant patients. Then the Probit Regressive Analysis considered as weak classifier analyze patient healthcare data request in the Content Storage ('CS'). During the regression analysis, it checks whether the data request with a similar name in CS. If the content is present, the particular router node is selected to place that content. If the copy is absent, the patient information is stored in the cache for further processing. In this regression analysis, a router's cache is frequently hit by a patient data request and responds immediately with minimum latency and improves the throughput. The detailed process of the proposed RADEF technique

*A. Random probit regressive bucklin decision forest classifier-based content caching in ICN*

By applying the proposed RADEF technique, first, the patient healthcare data requests are collected from the maternal healthcare dataset with the help of an IoT device in order to predict the health risks of the patient during their pregnancy. The data obtained are formulated in matrix format as given below.

$$X = \begin{bmatrix} DR_1F_1 & DR_1F_2 & \dots & DR_1F_n \\ DR_2F_1 & DR_2F_2 & \dots & DR_2F_n \\ \dots & \dots & \dots & \dots \\ DR_mF_1 & DR_mF_2 & \dots & DR_mF_n \end{bmatrix} \quad (1)$$

Where, X denotes an input matrix which contains the patient healthcare data request 'DR' obtained from dataset for each patient '$P_i$' with the features '$F_j$'. Then the request is forwarded to the routers for verifying whether the data request with similar name in CS.

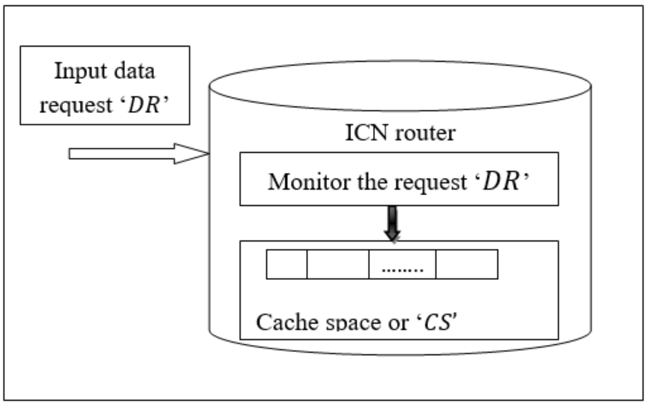


Fig 2. Process of data request monitoring in router

Figure 2 illustrates the process of the patient healthcare data request monitoring in router. By applying random probit regressive analyzed decision forest classification, it checks whether the patient healthcare data request with the same name is present in 'CS'. When the copy is present, the particular router node is selected to place the content. If copy is not present, the patient information is stored in cache for further processing.

The Bucklin decision forest classifier is a machine learning ensemble technique that converts the weak classification results into strong ones. The weak classifier is a base classification algorithm that difficult to provide accurate classification results. But the ensemble classifier provides accurate results by combining the results of the weak classifier with minimum error.

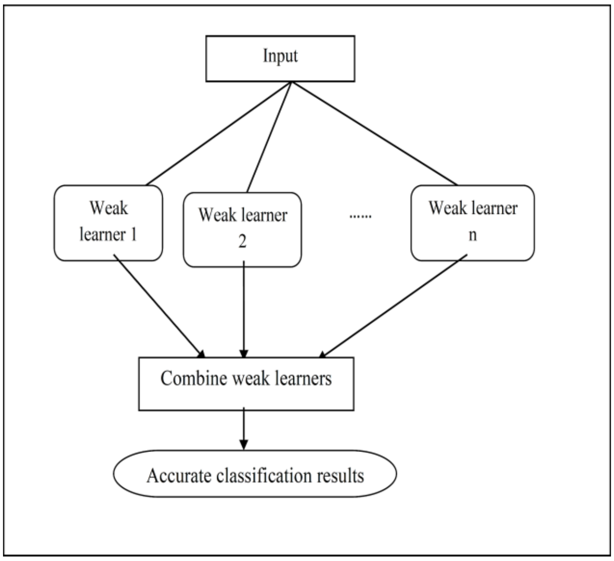


Fig 3. Structural design of random probit regressive bucklin decision forest classification

Figure 3 given above depicts the process of the random probit regressive bucklin decision forest algorithm to obtain strong output results. Let us consider the input of the decision forest algorithm is a number of patient healthcare data requests $DR_1$, $DR_2$, $DR_3$ . . . $DR_n$. The bucklin random decision forest algorithm constructs a set of weak learners as $w_i \in w_1, w_2, w_3, \dots, w_k$ from a probit regression tree. A probit regression tree is the machine learning technique that is used to analyzes patient healthcare data request search.

A probit regression tree is a decision tree that includes a root node linked to the terminal nodes (leave node). A probit model is a type of regression where the dependent variable (i.e. outcome) takes only dichotomous or binary outcome variables either ‘0’ or ‘1’. The regression tree provides a classification result based on Simple Matching Coefficient (SMC).

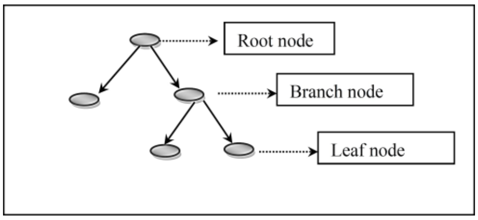


Fig 4. Probit regression tree

Figure 4 illustrates the structure of the probit regression tree which includes root node, branch node and leaf node. The regression tree is a flowchart-like structure in which each root node represents a ‘test’ on an input (e.g.) whether a patient healthcare data request with the same name is present in ‘CS’, each branch represents the outcome of the test, and each leaf node represents a class label (decision taken 0 or 1). The root node uses the simple matching coefficient for analyzing the patient healthcare data request.

The coefficient is a statistical method used for comparing the similarity and diversity of sample sets. Here, every incoming patient healthcare data request to the router node is analyzed in the content storage. The matching coefficient is formulated as given below,

$$M_{\text{coef}} = \left[ \frac{DR_i \cap \text{Content}_{\text{CS}}}{\text{Number of contents in CS}} \right] \quad (2)$$

Where $M_{coef}$ indicates a simple matching coefficient used to analyze the incoming patient healthcare data requests ‘$DR_i$’. $Content_{CS}$ denotes a router node that seeks the content in the content storage ‘CS’. The $M_{coef}$ returns the output value from 0 to ‘1’.

$$w_i \rightarrow M_{\text{coef}} = \begin{cases} 1, & DR_i \text{ present in CS} \\ 0, & DR_i \text{ absent in CS} \end{cases} \quad (3)$$

Where $M_{coef}$ returns ‘1’ indicate that the patient healthcare data request is present in ‘CS’. When the copy is present, the particular router node is selected to place the content. The $M_{coef}$ returns ‘0’ indicates that the patient information is not present in ‘CS’ and it is stored in cache for further processing. When the copy is present, the specified router node is chosen for delivering the content. The stored information in cache is forwarded to the next hop router node for patient healthcare data analysis.

The base classification technique is not accurate. In order to obtain strong results, the entire weak learner’s results are combined. The strong classification results are achieved by summing all the weak learner results as given below,

$$Y = \sum_{i=1}^{k} w_i \quad (4)$$

Where, Y denotes an outcome of the ensemble technique, $w_i$ denotes a predicted result of base classifier (i.e. probit regression tree). For each weak learner, the error is computed based on the difference between the actual and predicted outcomes. Therefore, the error is mathematically formulated as given below,

$$Er = \frac{1}{n} |Y_a - Y|^2 \quad (5)$$

From (5), $E_r$ indicates an error, $Y_a$ stands for the actual outcomes, Y indicates observed results. After that, the Bucklin Voting is applied to rank the results based on the error rate.

The weak learner results having the minimum errors are ranked first than the other results. After the ranking process, the result with a higher error is removed. This process helps to improve the accurate results and minimizes the complexity of achieving the ensemble results. This voting scheme also prevents an undesirable outcome. Finally, the obtained results of weak learners are counted, and identifies the majority votes to be elected.

$$Y = \arg\max_k q(w_i) \quad (6)$$

From (6), Y signifies the strong ensemble learning outcomes, arg max represents an argument of the maximum function to find the majority votes (q) of the output ‘$W_i$’ whose conclusion is identified to the $k^{th}$ weak learner results. Finally, the ensemble technique results provide the majority of the output as final results. In this way, the incoming patient healthcare requests are processed and to provide the faster data response speed. This helps to minimize the traffic in ICN. As a result, the proposed technique accurately responses the number of requests in specific time interval resulting it improves the cache hit ratio. The algorithm of **ra**ndom probit regressive bucklin **de**cision **f**orest classifier technique is given below.

**Algorithm 1: Ra**ndom probit regressive bucklin **De**cision Forest classifier-based Information-Centric Networking Caching

**Input:** Dataset ‘DS’, features ‘$F = F_1, F_2, \ldots, F_n$’, Router node ‘$RN = RN_1, RN_2, \ldots, RN_n$’, patient healthcare data requests $DR_1, DR_2, DR_3 \ldots, DR_n$

**Output:** Improve the data transmission

**Begin**

**Step 1: for** each Dataset ‘DS’ with Features ‘F’

**Step 2:** Formulate input with data requests $DR_1, DR_2, DR_3 \ldots, DR_n$ and features stored in Content Storage ‘CS’

**Step 3:** Construct ‘K’ set of weak learners ‘$W_i$’

**Step 4**: **for each** data requests $DR_i$

**Step 5:** Analyze patient healthcare data request search in content storage

**Step 6**: Apply simple matching coefficient ‘$M_{coef}$’

**Step 7: if ($M_{coef}$ = 1) then**

**Step 8:** Patient healthcare data request is present in ‘CS’

**Step 9**: Particular router node is selected to place the content

**Step 10**: **else**

**Step 11:** Patient healthcare data request is absent in ‘CS’

**Step 12:** Patient information is stored in cache

**Step 13**: **end if**

**Step 14:** Combine weak learner results

**Step 15:** **for each** weak learner results

**Step 16:** **Calculate** error ‘$E_r$’

**Step 17:** **Rank the** weak learners

**Step 18:** Find weak learners with minimum error

**Step 19:** **Apply** the Bucklin voting

**Step 20:** **Find** arg $\max_k q(w_i)$

**Step 21: Return strong classified results**
**Step 22: End for**
**Step 23: End for**
**Step 24: End for**
End

Algorithm 1 describes the step-by-step process of random probit regressive Bucklin decision forest classifier-based Information-Centric Networking Caching. The number of patient healthcare data requests is taken as input to the random probit regressive Bucklin decision forest classifier. First, the set of weak learners is constructed. For each patient healthcare data request, the simple matching coefficient is applied for verifying whether the patient healthcare data request is present or absent in content storage. When the data is present in content storage, the particular router node is selected to transfer the content. Otherwise, the content is stored in the cache for further processing. The output of weak learners is combined to construct strong classification results. As a result, higher efficiency of content distribution is performed in ICN and minimal latency.

## IV. Experimental Setup

In this section, experimental assessment of proposed RADEF and existing PCSRC [1] cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5] are implemented in Python language using Maternal Health Risk dataset taken from https://www.kaggle.com/datasets/csafrit2/maternal-health-risk-data. The main aim of the dataset is to predict risk intensity level during pregnancy. The Data has been gathered from different hospitals, community clinics, and maternal healthcare's with the help of IoT based risk monitoring system. The dataset includes seven features and the patient healthcare data or instances are 1014. The seven features are discussed as given below,

1. Age: Age in years when a woman is pregnant.
2. SystolicBP: Upper value of Blood Pressure in mmHg
3. DiastolicBP: Lower value of Blood Pressure in mmHg
4. BS: Blood glucose levels is in terms of a molar concentration, mmol/L.
5. BodyTemp: body temperature in Fahrenheit (F)
6. HeartRate: A normal resting heart rate in beats per minute.
7. Risk Level: Predicted Risk Intensity Level during pregnancy

Based on the above patient healthcare data, the content matching is performed in ICN.

## V. Performance Results And Analysis

In this section, performance analysis of the proposed RADEF and existing methods namely PCSRC [1] cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5] are discussed with different metrics such as cache hit rate, network latency, average request length, average response time, server stretch ratio, and hop reduction ratio with respect to the number of patient healthcare data. The parameters are described as given below,

**Cache hit rate:** It is measured ratio of response number of requests and total number of requests in specific time interval. The formula for calculating the cache hit rate is expressed as follows,

$$\mathrm{CHR} = \left[\frac{\mathrm{Resp}}{\mathrm{Count}}\right] \times 100 \tag{7}$$

Where 'CHR' denotes a cache hit rate, 'Resp' denotes a response for the number response requests and the number of requests obtained by nodes 'Count'. It is measured in terms of percentage (%).

**Network latency:** It is an assessment of delay in communication over an ICN. Latency is measured as the amount of time consumed for a packet of healthcare data to be acquired, transmitted, and processed via multiple routers. This is mathematically formulated as given below.

$$\mathrm{NL} = \sum_{i=1}^{n} \mathrm{Req}_i \times \mathrm{Time}\ [\mathrm{DP_{acq}} + \mathrm{DP_{trans}} + \mathrm{DP_{recv}}] \tag{8}$$

Where, 'NL' denotes a network latency, basis of the data packets to be obtained '$DP_{acq}$', number of data to be transmitted '$DP_{trans}$' and the number of data received at the destination end '$DP_{recv}$' respectively with respect to the requests '$Req_i$' involved in the simulation. Latency is measured in milliseconds (ms).

**Average Request Length**: It is evaluated that refers to the average number of router nodes transmitted when an interest patient data hits the request content. Therefore, the ARL is mathematically computed as given below,

$$\mathrm{ARL} = \frac{\mathrm{HDP_{resp}}}{\mathrm{RDP_{resp}}} \tag{9}$$

Where, 'ARL' average request length, '$HDP_{resp}$' denotes a number of hops that a data is responded, '$RDP_{resp}$' indicates a number of responses that a data packet is responded. It is measured in percentage (%).

**Average Response time**: It is defined as the sum of time between request being sent to router and data being received by users. Therefore, the ART is mathematically computed as given below,

$$\mathrm{ART} = \sum_{i=1}^{n} \mathrm{Req_i} * \mathrm{Time}\ [\mathrm{DR_{sent}} + \mathrm{D_{recv}}] \tag{10}$$

Where, 'ART' average response time, '$Req_i$' denotes a number of requests. '$DR_{sent}$' denotes a request being sent to router, '$D_{recv}$' data received. Response time is measured in milliseconds (ms)

**Server Traffic Ratio:** STR is defined as the traffic ratio of the requests responded and the total requests. It is measured in percentage (%). The mathematical expression for determining the server traffic ratio is obtained as given below.

$$SRT = \frac{\text{Traffic created by request responded}}{\text{Total traffic in a network}} \times 100 \tag{11}$$

**Hop Reduction Ratio:** HRR is defined as the ratio between the number of hops of cache hits and the total number of hops of all accesses.

It is a metric for determining the efficiency of content caching decisions and indicates that the cache hit happens close to the requesters. The following is the mathematical formula for estimating the hop reduction ratio.

$$HRR = \frac{\text{Number of hopes of cache hits}}{\text{Total number of hopes of all accesses}} \times 100 \quad (12)$$

HRR refers to the hop count traversed by the patient data to reach the content provider (i.e., if a cache hit happens) to the hop count between the content requester and the server i.e., if network routers do not support caching. It is estimated in terms of percentage (%).

Table 1 and figure 5 illustrate a performance comparison of the cache hit rate versus cache size ranging from 80 to 800. As shown in the graph, the performance of the cache hit rate using RADEF is considerably improved than the other existing methods namely, PCSRC [1] cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5]. With the horizontal axis denoting the cache size for simulation, the vertical axis represents the cache hit rate. However, simulations conducted with a size of the cache is 80 resulted in a cache hit rate of 78.75% using PCSRC [1], 80% using Cooperative caching [2], 83.75% using Context-based caching mechanism [3], 82.25% using CCJRF-ICN [3], 85% using existing IRABC-ICN [4] and 88.75% using RADEF respectively.

For each method, different performance results were observed. Finally, the obtained results of the proposed RADEF are compared to the existing methods. After getting ten results, the average is taken to show the improvement of the proposed RADEF. The overall performance results indicate that the RADEF increases the cache hit rate by 12%, 9%, 6%, 8% and 4% when compared to PCSRC [1], Cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN **[5]** respectively.

The reason behind the improvement was due to the application of the **Ra**ndom probit regressive Bucklin **de**cision **f**orest classifier algorithm. By applying this algorithm, patient healthcare data requests are taken as input.

Then the simple matching coefficient is applied to verify whether the patient healthcare data request is present or absent in content storage. The error rate of the weak classifier is minimized and to make a better output by applying a Bucklin voting scheme. This helps to improve the cache hit rate.

TABLE 1
CACHE HIT RATE

| Cache Size (KB) | Cache Hit Rate (%) | | | | | |
|---|---|---|---|---|---|---|
| | PCSRC | Cooperative caching | Context-based caching mechanism | CCJRF-ICN | IRABC-ICN | RADEF |
| 80 | 78.75 | 80 | 81.25 | 83.75 | 85 | 88.75 |
| 160 | 79.32 | 81.36 | 82.15 | 83.98 | 86.12 | 89 |
| 240 | 80.02 | 82 | 82.96 | 84 | 86.63 | 89.74 |
| 320 | 80.76 | 82.85 | 83.65 | 84.85 | 87 | 90.05 |
| 400 | 81.2 | 83.1 | 84.33 | 85.12 | 87.45 | 90.74 |
| 480 | 82.05 | 83.89 | 84.96 | 85.96 | 88.1 | 91.1 |
| 560 | 82.86 | 84.06 | 85 | 86 | 88.25 | 92 |
| 640 | 83.44 | 84.65 | 85.74 | 87 | 89.1 | 92.74 |
| 720 | 83.98 | 85 | 86 | 87.65 | 90 | 93 |
| 800 | 84.08 | 85.65 | 86.78 | 88.2 | 90.8 | 94.3 |

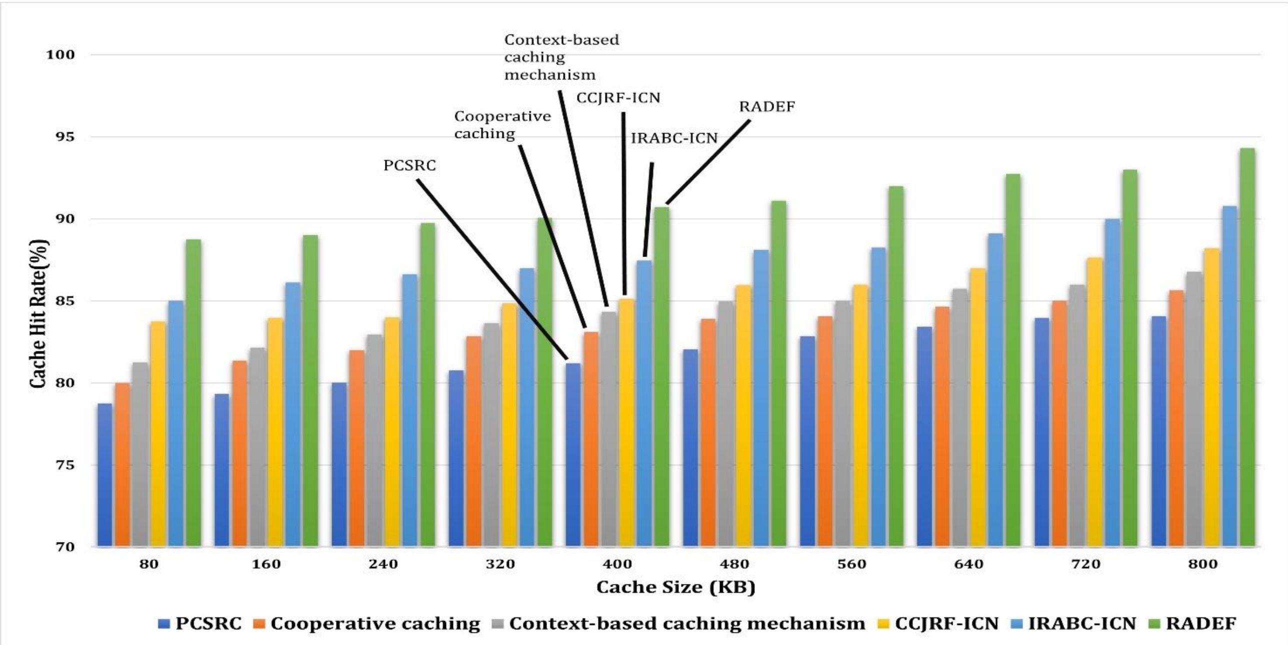


Fig 5. Performance comparison of Cache hit rate

TABLE 2
NETWORK LATENCY

| Cache Size (KB) | Network Latency (ms) | | | | | |
|---|---|---|---|---|---|---|
| | PCSRC | Cooperative caching | Context-based caching mechanism | CCJRF-ICN | IRABC-ICN | Proposed RADEF |
| 80 | 5.8 | 6.3 | 4.9 | 5.2 | 4.2 | 3.8 |
| 160 | 8.2 | 9 | 6.2 | 7.8 | 5.35 | 4.84 |
| 240 | 10.55 | 12.62 | 8.65 | 9.7 | 7.16 | 6.6 |
| 320 | 14.65 | 16.02 | 10.3 | 12.5 | 8.25 | 7.2 |
| 400 | 15.5 | 17.8 | 11.32 | 13.66 | 10.35 | 8.5 |
| 480 | 18 | 20.20 | 13.25 | 15.02 | 11.45 | 10.8 |
| 560 | 20.3 | 22.33 | 15.10 | 17.05 | 13.15 | 11.2 |
| 640 | 22.08 | 24.78 | 17 | 20.32 | 15 | 12.48 |
| 720 | 26.1 | 28.33 | 19.2 | 22.52 | 16.25 | 14.76 |
| 800 | 30 | 32.10 | 22.5 | 24.10 | 18 | 16 |

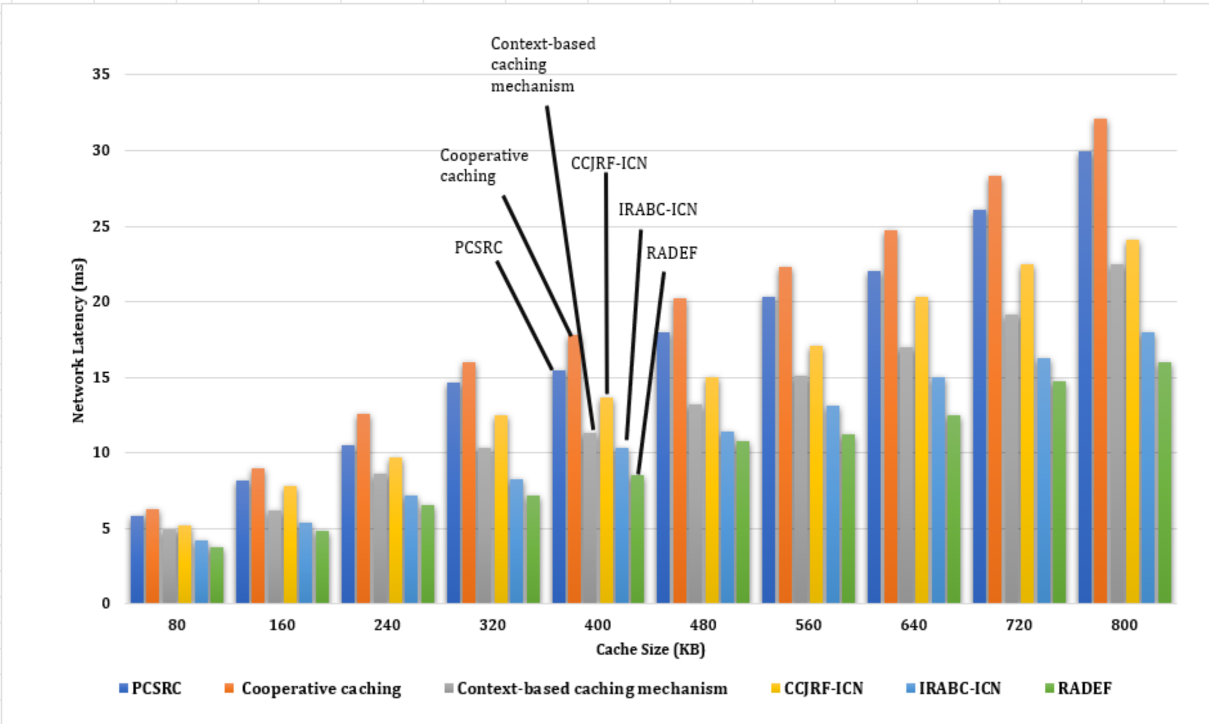


Fig 6. Performance comparison of network latency

Table 2 and figure 6 illustrate the network latency results using the six methods, PCSRC [1] cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5] and RADEF. From the above figure, with the network latency analysis provided on the y-axis and cache size given on the x-axis, both the performance factors are identified to be proportionate with each other. Here, the cache sizes are taken in the ranges from 80 to 800 and the number of requests ranges from 20 to 200 for calculating the network latency. As shown in figure 6, increasing the number of cache sizes results in an increase in the performance of network latency also. But, with simulations performed using 80 cache size and 20 requests, time consumed in content caching using the RADEF method was found to be 3.8ms, 5.8ms using PCSRC [1], 6.3ms using cooperative caching [2], 4.9ms using Context-based caching mechanism [3], 5.2ms using CCJRF-ICN [4], and 4.2ms using IRABC-ICN [5]. The averages of ten performance results are compared to existing methods. The overall performance results indicate that the performance of network latency was found to be comparatively minimized by 43%, 48%, 25%, 35% and 12% using RADEF when compared to existing methods PCSRC [1] cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5]. The reason for the improvement was owing to the incorporation of a probit regression tree for validating the incoming data request in the cache using a simple matching coefficient. Here, the successful validation with respect to the patient healthcare data request, a particular router node is chosen to deliver the content with minimum latency.

TABLE 3
AVERAGE REQUEST LENGTH

| Cache Size (KB) | Average Request Length (%) | | | | | |
|---|---|---|---|---|---|---|
| | PCSRC | Cooperative caching | Context-based caching mechanism | CCJRF-ICN | IRABC-ICN | RADEF |
| 80 | 80 | 75 | 65 | 70 | 60 | 55 |
| 160 | 75 | 72 | 62 | 68 | 56 | 52 |
| 240 | 73 | 70 | 60 | 66 | 55 | 51 |
| 320 | 69 | 66 | 58 | 64 | 53 | 50 |
| 400 | 65 | 62 | 56 | 60 | 52 | 49 |
| 480 | 62 | 58 | 54 | 56 | 51 | 48 |
| 560 | 61 | 56 | 50 | 54 | 48 | 45 |
| 640 | 60 | 54 | 49 | 52 | 46 | 43 |
| 720 | 59 | 52 | 47 | 50 | 44 | 41 |
| 800 | 58 | 50 | 46 | 48 | 43 | 39 |

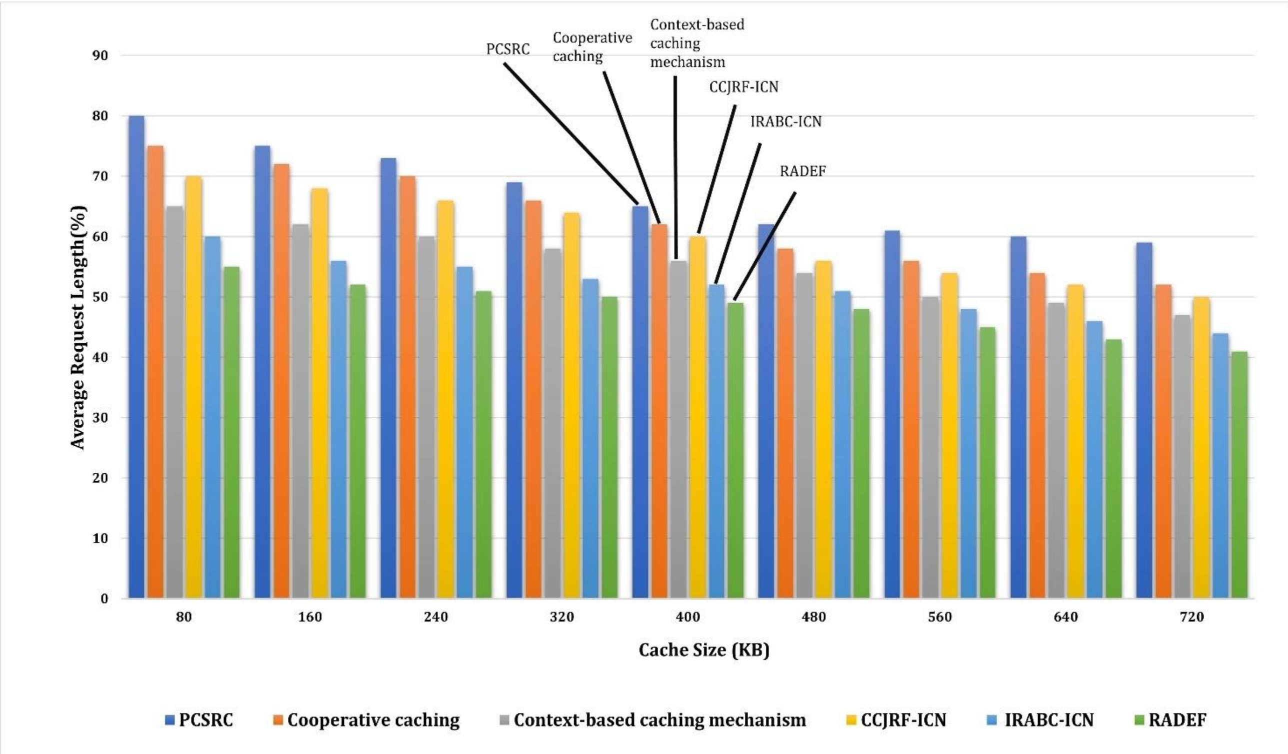


Fig 7. Performance comparison of average request length

Table 3 and figure 7 given above illustrate the graphical representations of average request length using six methods, RADEF, PCSRC [1] cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5]. With the horizontal axis denoting the cache size for simulation, the vertical axis represents the average request length. The number of the data request is taken from 20 to 200. As illustrated in the above figure, increasing the number of data requests and the average request length is found to be minimized. Since the requested content is progressively cached in routers through the simple matching coefficient. The average request length of RADEF is about 55% which is the best result in the other schemes due to its strategy of caching everywhere. The higher the hit ratio, the method reduces the average request length. The average of ten results indicates that the request length using RADEF was found to be minimized by 28% compared to [1] and 23% compared to [2], 13% compared to [3], 19% compared to [4] and 7% compared to [5] respectively.

Figure 8 given below illustrates the graphical illustration of the average response time with 800 sizes of caches as input. From the figure, it is inferred that the average response time is increasing with the increasing cache sizes and the user requests provided as input. Among the four methods, the performance of average response time is found to be minimized using RADEF when compared to existing PCSRC [1] cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5]. Let us consider the cache size is 80 and the number of data requests is 20 in the first iteration for calculating the average response time. By applying RADEF, the average response time was found to be 14ms, 22%using PCSRC [1], 24%

using cooperative caching [2], 18% using Context-based caching mechanism [3], 20% using CCJRF-ICN [4] and 16.6% using IRABC-ICN [5] respectively. Likewise, different performance results are observed for each method. The average of ten results indicates that the overall performance of average response time is considerably minimized by 25%, 30%, 13%, 19% and 7% using RADEF when compared to PCSRC [1] cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5] respectively. This is because the RADEF effectively finds the user request content in the cache and delivers the content with minimum time.

TABLE 4
AVERAGE RESPONSE TIME

| Cache Size (KB) | Average Response Time (ms) | | | | | |
|---|---|---|---|---|---|---|
| | PCSRC | Cooperative caching | Context-based caching mechanism | CCJRF-ICN | IRABC-ICN | RADEF |
| 80 | 22 | 24 | 18 | 20 | 16.6 | 14 |
| 160 | 26 | 28 | 22 | 25 | 20 | 18 |
| 240 | 32 | 34 | 28 | 30 | 26 | 24 |
| 320 | 38 | 42.13 | 32 | 34 | 30 | 28 |
| 400 | 43 | 47 | 38 | 40 | 35 | 33 |
| 480 | 48 | 53 | 42 | 45 | 40 | 37.2 |
| 560 | 54.5 | 58.03 | 46.2 | 48.5 | 44.6 | 42 |
| 640 | 56.8 | 60 | 48.2 | 52.3 | 46 | 44.8 |
| 720 | 60 | 63 | 52.7 | 55.7 | 50.2 | 47.7 |
| 800 | 64 | 66 | 55.9 | 60.5 | 53.6 | 51 |

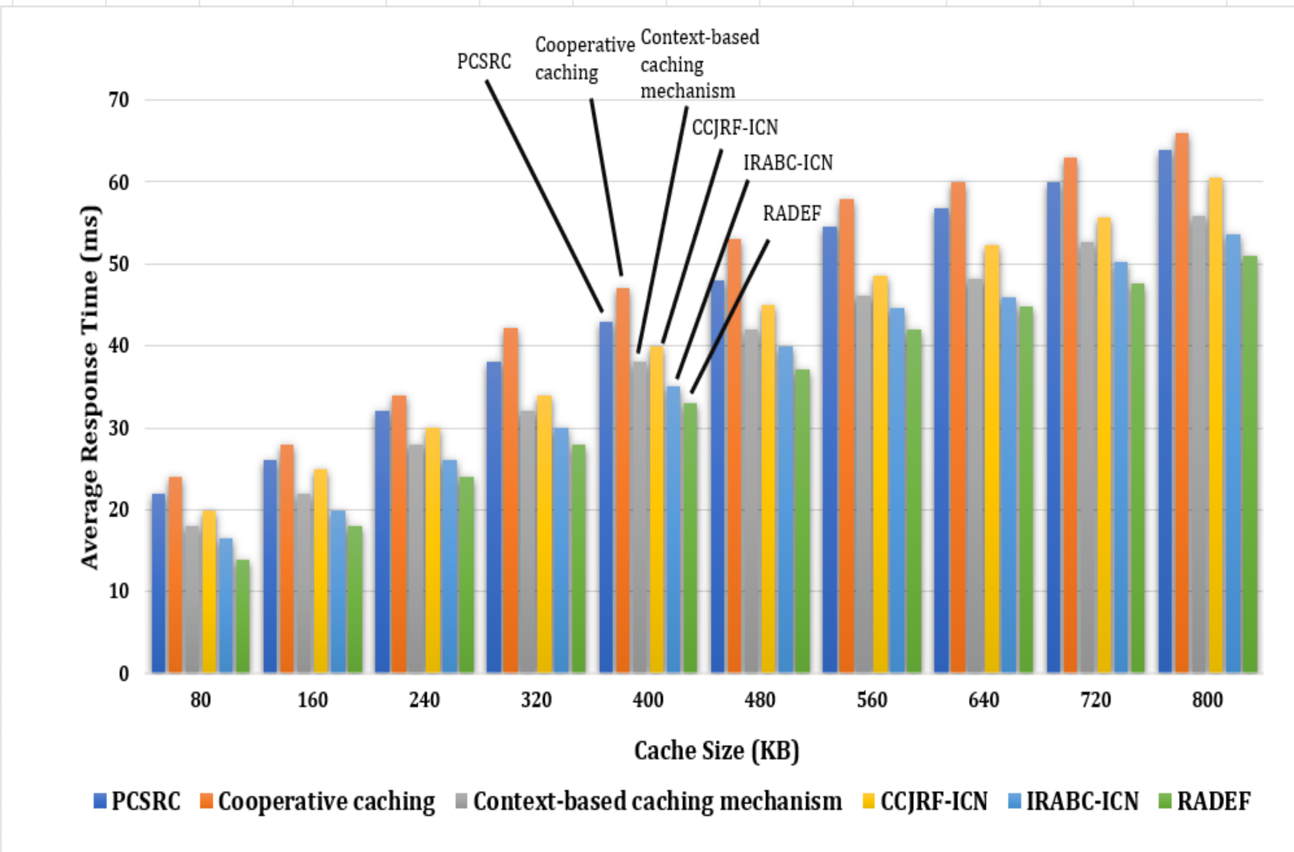


Fig 8. Performance comparison of average response time

TABLE 5
SERVER TRAFFIC RATIO

| Cache Size (KB) | Server Traffic Ratio (%) | | | | | |
|---|---|---|---|---|---|---|
| | PCSRC | Cooperative caching | Context-based caching mechanism | CCJRF-ICN | IRABC-ICN | RADEF |
| 80 | 6.4 | 6.8 | 5.4 | 5.8 | 5.2 | 4.8 |
| 160 | 6.8 | 7.2 | 6 | 6.4 | 5.7 | 5.3 |
| 240 | 9 | 10.3 | 7.9 | 8.2 | 7.2 | 6.8 |
| 320 | 12.3 | 13.2 | 10.5 | 11.6 | 9.8 | 8.9 |
| 400 | 13.5 | 15.5 | 11.8 | 12.3 | 10.2 | 9.2 |
| 480 | 14.6 | 17.6 | 12.4 | 13.8 | 11.3 | 10.2 |
| 560 | 18.2 | 20.2 | 15.6 | 16.8 | 13.4 | 12.3 |
| 640 | 20.6 | 22.3 | 16.2 | 17.2 | 14.2 | 13.5 |
| 720 | 21.3 | 23.4 | 17.7 | 19 | 15.6 | 14.5 |
| 800 | 22.5 | 24.2 | 18.5 | 20.3 | 17.3 | 15.8 |

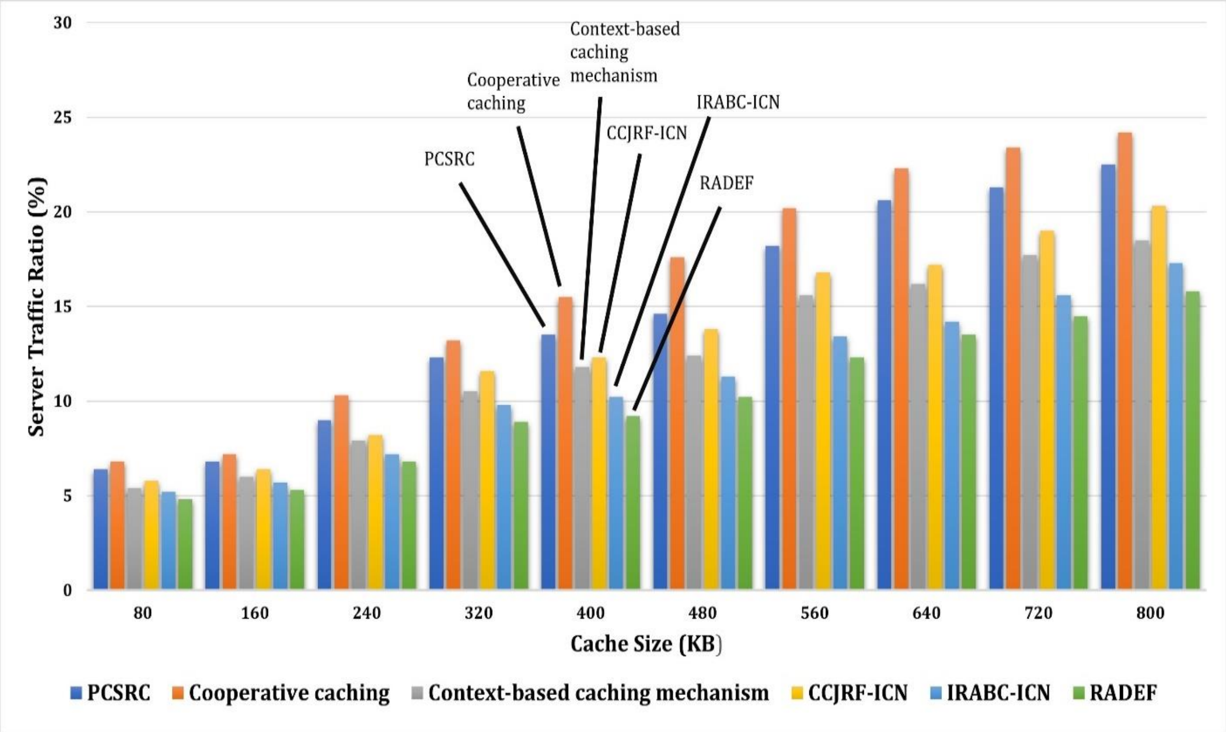


Fig 9. Performance comparison of server traffic ratio

The comparative examination of the server traffic ratio using four methods RADEF, PCSRC [1] cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5] is shown in Table 5 and Figure 9 depending on the catch size.In figure 9, the catch size is depicted along the horizontal axis, while the server traffic ratio is shown along the vertical axis.If the server traffic ratio is high, the likelihood of requests being answered in the cache path is likewise lowered. According to the aforementioned data, the proposed RADEF has less server traffic ratio than the other existing methods. The effective reduction of server traffic ratio is achieved in the proposed RADEF by the implementation of **Ra**ndom probit regressive **b**ucklin **de**cision **f**orest classifier. The patient healthcare requests are processed quickly due to the strong ensemble learning technique. That helps in reducing the traffic in a significant manner. Therefore, the average of 10 findings shows that employing RADEF reduces the overall performance of server traffic ratio by 29%, 36%, 16%, 22% and 8% when compared to PCSRC [1] cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5] respectively.

TABLE 6
HOP REDUCTION RATIO

| Cache Size (KB) | Hop Reduction Ratio (%) | | | | | |
|---|---|---|---|---|---|---|
| | PCSRC | Cooperative caching | Context-based caching mechanism | CCJRF-ICN | IRABC-ICN | RADEF |
| 80 | 0.5 | 0.54 | 0.6 | 0.58 | 0.66 | 0.7 |
| 160 | 0.61 | 0.6 | 0.7 | 0.7 | 0.8 | 0.9 |
| 240 | 0.7 | 0.7 | 0.92 | 0.85 | 1.1 | 1.4 |
| 320 | 1.5 | 1.6 | 2 | 1.8 | 2.4 | 2.6 |
| 400 | 2 | 2.4 | 2.8 | 2.6 | 3 | 3.5 |
| 480 | 2.8 | 3 | 3.8 | 3.6 | 4.2 | 4.7 |
| 560 | 4.2 | 4.5 | 6 | 5.7 | 6.3 | 7.5 |
| 640 | 5.3 | 5.8 | 7.2 | 6.5 | 7.8 | 8.2 |
| 720 | 5.8 | 6.4 | 8.1 | 7.6 | 8.5 | 9.2 |
| 800 | 6.1 | 6.8 | 8.6 | 8.2 | 9.1 | 9.8 |

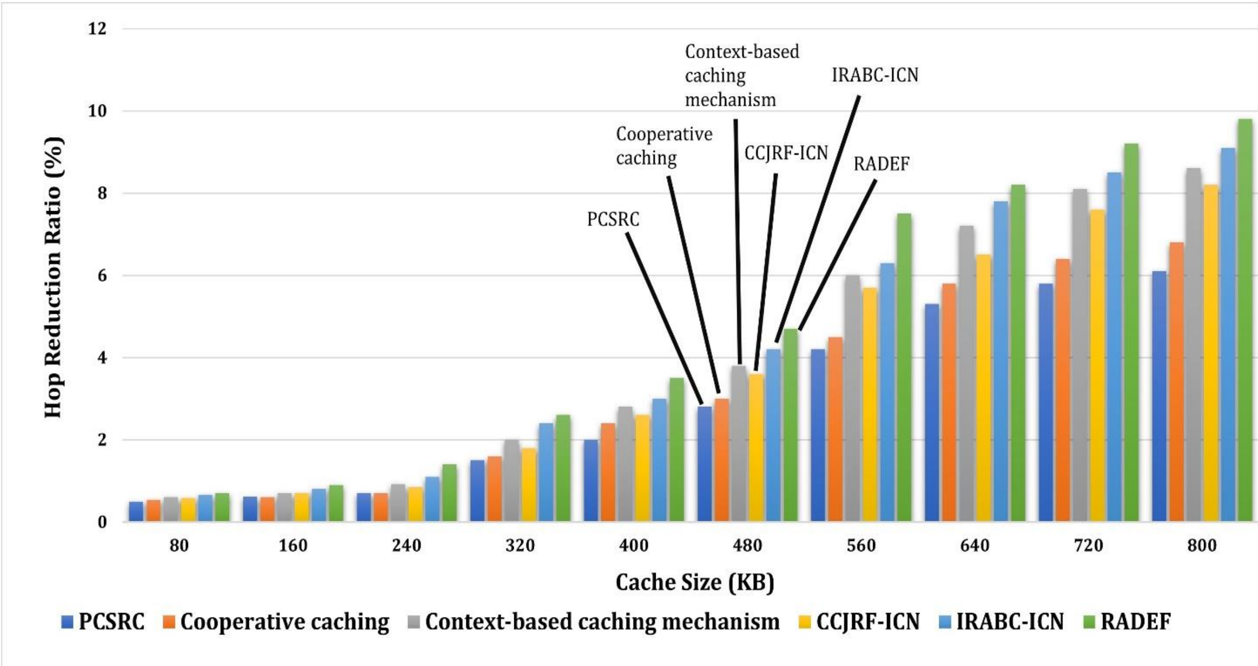


Fig 10. Performance comparison of hop reduction ratio

In accordance with the catch size, Table 6 and Figure 10 shows the hop reduction ratio using four distinct approaches RADEF, PCSRC [1], cooperative caching [2], Context-based caching mechanism [3], CCJRF-ICN [4], IRABC-ICN [5]. The higher hop reduction ratio suggests that data is transmitted more efficiently. The graphical representation shows that the proposed RADEF effectively improves the hop reduction ratio than the other three existing methods. Due to the application of **Ra**ndom probit regressive Bucklin **de**cision **f**orest classifier, the hop reduction ratio employing RADEF was shown to be improved by 66%, 52%, 23% 32% and 12% when compared to [1] [2], [3] [4],[5] correspondingly.

## VI. CONCLUSION

The policy of in-network caching holds crucial importance in facilitating quick and efficient communications within IoT networking technologies in ICN. Content caching methods significantly impact its efficiency. This paper introduces ran efficient caching method, RADEF, designed to enhance IoT-based healthcare performance by reducing response time and content retrieval latency. In RADEF, the random probit regressive Bucklin decision forest classifier performs content caching operations and delivers the interesting data from the router with minimum time resulting in it reducing the latency. By using healthcare risk prediction dataset, the comparative analysis is conducted between the RADEF technique and existing methods in terms of six different metrics. The observed result facilitates that the proposed RADEF is more efficient for IoT aware content caching with healthcare data with maximum cache hit ratio, hop reduction ratio, and minimum network latency, average request length, average response time, and server traffic ratio than the existing methods.

## REFERENCES


[1] Moli Zhang, Bowei Hao, Ruilin Wang and Ye Wang, “A Pre-Caching Strategy Based on the Content Relevance of Smart Device’s Request in Information-Centric IoT”, IEEE Access, Volume 8, April 2020, Pages 75761 – 75771. **DOI:** 10.1109/ACCESS.2020.2988926

[2] Yuming Zhang, Bohao Feng, Wei Quan, Aleteng Tian, Keshav Sood and Youfang Lin and Hongke Zhang, “Cooperative Edge Caching: A Multi-Agent Deep Learning Based Approach”, IEEE Access, Volume 8, July 2020, Pages 133212 – 133224. **DOI:** 10.1109/ACCESS.2020.3010329

[3] Luís Leira, Miguel Luís, Susana Sargento “Context-based caching in mobile information-centric networks”, Computer Communications, Elsevier, Volume 193, 2022, Pages 214-223, https://doi.org/10.1016/j.comcom.2022.07.017

[4] Krishna Delvadia, Nitul Dutta, Rajendrasinh Jadeja, “CCJRF-ICN: A Novel Mechanism for Coadjuvant Caching Joint Request Forwarding in Information Centric Networks”, Published in: IEEE Access, Volume 9, 2021, Pages 84134 – 84155. **DOI:** 10.1109/ACCESS.2021.3087558

[5] R. Sangeetha and T. N. Ravi, “IoT Content Cache Efficacy in Health Care Data Diagnostics Using Isotonic Regressive Adaptive Boost Classification”, International Journal of Intelligent Engineering and Systems, Volume 16, Issue3, 2023, Pages 593-603. DOI: 10.22266/ijies2023.0630.47

[6] Sumit Kumar & Rajeev Tiwari, “An efficient content placement scheme based on normalized node degree in content centric networking”, Cluster Computing, Springer, Volume 24, 2021, Pages 1277–1291. https://doi.org/10.1007/s10586-020-03185-0

[7] Satish Anamalamudi, Mohammed Saeed Alkatheiri; Eesa Al Solami; Abdur Rashid Sangi,“Cooperative Caching Scheme for Machine-to-Machine Information-Centric IoT Networks”,IEEE Canadian Journal of Electrical and Computer Engineering, Volume 44, Issue 2, 2021, Pages 228 – 237. **DOI:** 10.1109/ICJECE.2020.3046844

[8] Muhammad Ali Naeem, Rehmat Ullah, Yahui Meng, Rashid Ali and Bilal Ahmed Lodhi, “Caching Content on the Network Layer: A Performance Analysis of Caching Schemes in ICN-based Internet of Things”, IEEE Internet of Things Journal, Volume 9, Issue 9, 2022, Pages 6477 – 6495. **DOI:** 10.1109/JIOT.2021.3110977

[9] Ngoc-Thanh Dinh and Younghan Kim, “An Efficient Distributed Content Store-Based Caching Policy for Information-Centric Networking”, Sensors, Volume 22, Issue 4, 2022, Pages 1-12. https://doi.org/10.3390/s22041577

[10] Xiaonan Wang and Yanli Li, “Fog-Assisted Content-Centric Healthcare IoT”, IEEE Internet of Things Magazine, Volume 3, Issue 3, September 2020, Pages 90 – 93.**DOI:** 10.1109/IOTM.0001.2000005

[11] Boubakr Nour, Hatem Ibn-Khedher, Hassine Moungla, Hossam Afifi, Fan Li, Kashif Sharif, Hakima Khelifi and Mohsen Guizani, “Internet of Things Mobility over Information-Centric/Named-Data Networking”, IEEE Internet Computing, Volume 24, Issue 1, January-February 2020, Pages 14 – 24. **DOI:** 10.1109/MIC.2019.2963187

[12] Hamid Asmat, Ikram Ud Din, Fasee Ullah, Muhammad Talha, Murad Khan and Mohsen Guizani, “ELC: Edge Linked Caching for content updating in information-centric Internet of Things”, Computer Communications, Elsevier, Volume 156, 2020, Pages 172-184.https://doi.org/10.1016/j.comcom.2020.03.049

[13] Hamid Asmat, Fasee Ullah, Mahdi Zareei, Atif Khan and Ehab Mahmoud Mohamed, “Energy-Efficient Centrally Controlled Caching Contents for Information-Centric Internet of Things”, IEEE Access, Volume 8, July 2020, Pages 126358 – 126369. **DOI:** 10.1109/ACCESS.2020.3008193

[14] Gang Li and Zhijun Wu, “Information-Centric Networking Cache Robustness Strategy for System Wide Information Management”, IEEE Access, Volume 8, Pages 82432 – 82441. **DOI:** 10.1109/ACCESS.2020.2991762

[15] Ikram Ud Din, Suhaidi Hassan, Ahmad Almogren, Farrukh Ayub and Mohsen Guizani “PUC: Packet Update Caching for energy efficient IoT-based Information-Centric Networking”, Future Generation Computer Systems, Elsevier, Volume 111, October 2020, Pages 634-643.https://doi.org/10.1016/j.future.2019.11.022

[16] Nazib Abdun Nasir and Seong-Ho Jeong, “Content Management Based on Content Popularity Ranking in Information-Centric Networks”, Applied Science, Volume 11, Issue 13, 2021, Pages 1-21. https://doi.org/10.3390/app11136088

[17] Sumit Kumar, Rajeev Tiwari and Wei-Chiang Hong, “QoS Improvement Using In-Network Caching Based on Clustering and Popularity Heuristics in CCN”, Sensors, Volume 21, Issue 21, 2021, Pages 1-22. **DOI**: 10.3390/s21217204

[18] Luís Leira, Miguel Luís, Susana Sargento, “Context-based caching in mobile information-centric networks”, Computer Communications, Elsevier, Volume 193, 2022, Pages 214-223. https://doi.org/10.1016/j.comcom.2022.07.017

[19] Divya Gupta, Shalli Rani, Syed Hassan Ahmed, Sahil Verma, Muhammad Fazal Ijaz, and Jana Shafi, “Edge Caching Based on Collaborative Filtering for Heterogeneous ICN-IoT Applications”, Sensors, Volume 21, Issue 16, 2021, Pages 1-17. https://doi.org/10.3390/s21165491

[20] Nazib Abdun Nasir and Seong-Ho Jeong, “Fast Content Delivery Using a Testbed-Based Information-Centric Network”, IEEE Access, Volume 9, Pages 101600 – 101613. 2021. **DOI:** 10.1109/ACCESS.2021.3096042

[21] Bashaer Alahmri, Saad Al-Ahmadi, Abdelfettah Belghith, “Efficient Pooling and Collaborative Cache Management for NDN/IoT Networks”, IEEE Access, Volume 9, Pages 43228 – 43240. **DOI:** 10.1109/ACCESS.2021.3066133

[22] Ryo Nakamura, Noriaki Kamiyama, “Content Availability at Network Failure in Information-Centric Networking”, IEEE Transactions on Network and Service Management, Volume 18, Issue 3, 2021, Pages 3889 – 3899. **DOI:** 10.1109/TNSM.2021.3073233

[23] Tian Pan, Xingchen Lin, Enge Song, Cheng Xu, Jiao Zhang, Hao Li, Jianhui Lv, Tao Huang, Bin Liu, and Beichuan Zhan, “NB-Cache: Non-Blocking In-Network Caching for High-Performance Content Routers”, IEEE/ACM Transactions on Networking, Volume 29, Issue 5, 2021, Pages 1976 – 1989. **DOI:** 10.1109/TNET.2021.3083599

[24] Muhammad Izhar, Syed Asad Ali Naqvi, Adeel Ahmed, Saima Abdullah, Nazik Alturki, and Leila Jamel, “Enhancing Healthcare Efficacy Through IoT-Edge Fusion: A Novel Approach for Smart Health Monitoring and Diagnosis”, IEEE Access, Volume 11, 2023, Pages 136456-136467. **DOI**: 10.1109/ACCESS.2023.3337092.

[25] S. S. Saranya, and N. Sabiyath Fatima, “IoT-Based Patient Health Data Using Improved Context-Aware Data Fusion and Enhanced Recursive Feature Elimination Model”, Volume 10, 2022, Pages 128318-128335. **DOI:** 10.1109/ACCESS.2022.3226583.