



# SVM-based Analysis for Predicting Success Rate of Interest Packets in Information Centric Networks

Nitul Dutta, Sudeep Tanwar, Shobhit K. Patel & Gheorghita Ghinea

To cite this article: Nitul Dutta, Sudeep Tanwar, Shobhit K. Patel & Gheorghita Ghinea (2022) SVM-based Analysis for Predicting Success Rate of Interest Packets in Information Centric Networks, Applied Artificial Intelligence, 36:1, 2020488, DOI: [10.1080/08839514.2021.2020488](https://doi.org/10.1080/08839514.2021.2020488)

To link to this article: <https://doi.org/10.1080/08839514.2021.2020488>



© 2022 The Author(s). Published with license by Taylor & Francis Group, LLC.



Published online: 31 Dec 2021.



[Submit your article to this journal](#)



Article views: 1136



[View related articles](#)



[View Crossmark data](#)



Citing articles: 1 [View citing articles](#)

# SVM-based Analysis for Predicting Success Rate of Interest Packets in Information Centric Networks

Nitul Dutta<sup>a</sup>, Sudeep Tanwar<sup>b</sup>, Shobhit K. Patel<sup>c</sup>, and Gheorghita Ghinea<sup>d</sup>

<sup>a</sup>Computer Science And Engineering Department, SRM University, Amaravati, India; <sup>b</sup>Computer Engineering Department, Nirma University, Ahmedabad, India; <sup>c</sup>Electronics and Communication Engineering Department, Marwadi University, Rajkot, India; <sup>d</sup>Computer Science Department, College of Engineering Design and Physical Sciences., Brunel University, London, UK

## ABSTRACT

A consumer in Information Centric Network (ICN) generates an Interest packet by specifying the name of the required content. As the network emphasizes on content retrieval without much bothering about who serves it (a cache location or actual producer), every Content Router (CR) either provides the requested content back to the requester (if exists in its cache) or forwards the Interest packet to the nearest CR. While forwarding an Interest packet, the ICN routing by default does not provide any mechanism to predict the probable location of the content searched. However, having a predictive model before forwarding may significantly improve content retrieval performance. In this paper, a machine learning (ML) algorithm, particularly a Support Vector Machine (SVM) is used to forecast the success of the Interest packet. A CR can then send an Interest packet in the outgoing interface which is forecasted successful. The objective is to maximize the success rate which in turn minimizes content search time and maximizes throughput. The dataset used in is generated from a simulation topology designed in *ndnSim* comprising 10 K data points having 10 features. The linear, RBF and the polynomial kernel (with degree 3) are used to analyze the dataset. The polynomial kernel shows the best behavior with 98% accuracy. A comparative retrieval time with and without ML demonstrates around 10% improvement with SVM enable forwarding compared to normal ICN forwarding.

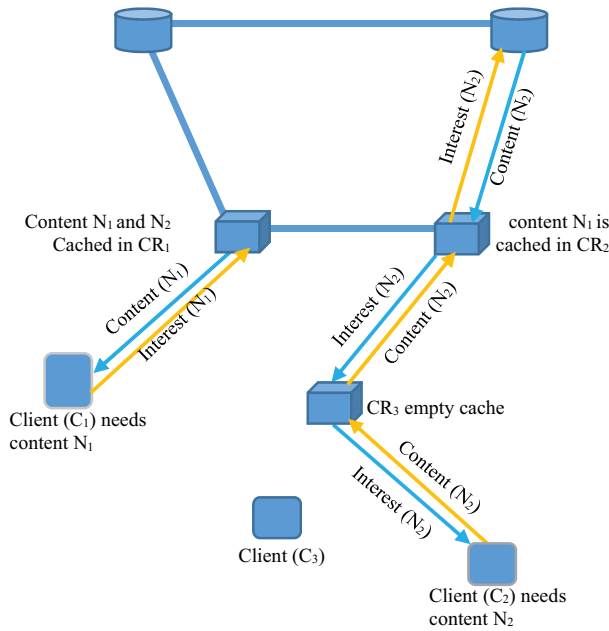
## ARTICLE HISTORY

Received 2 February 2021  
Revised 10 December 2021  
Accepted 15 December 2021

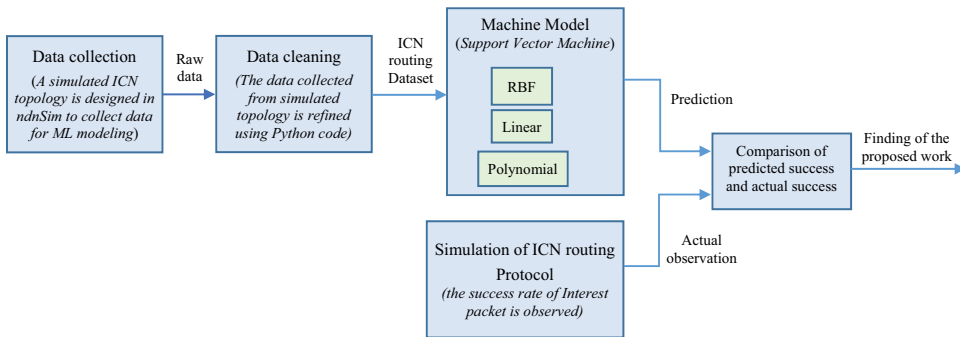
## Introduction

The internet applications of current time is extremely sensitive to access delay in one hand and data hungry on the other (Chegini and Mahanti 2019; Cisco Systems, 2019; Talluri et al. 2021). Users are more particular about retrieval speed of the desired content rather than access location. In such situations, the end users could be served better if the underlying network allows to obtain the requested content from any of the source without

**CONTACT** Nitul Dutta  [nituldutta@gmail.com](mailto:nitul Dutta@gmail.com)  Computer Science And Engineering Department, SRM University, Amaravati, Andhra Pradesh, India



**Figure 1.** The diagram demonstrates the flow of Interest packet to search content. There are two clients C<sub>1</sub> and C<sub>2</sub> requesting content N<sub>1</sub> and N<sub>2</sub> respectively. P<sub>1</sub> and P<sub>2</sub> are the generators of N<sub>1</sub> and N<sub>2</sub>. Moreover N<sub>1</sub> is cached in CR<sub>1</sub> and hence C<sub>1</sub> is may be served by CR<sub>1</sub> and C<sub>2</sub> is served by the actual producer of N<sub>2</sub> (as it is not cached in any CR). Our algorithm tries to predict the appropriate location of the cache store using ML models.



**Figure 2.** System model for applying ML in ICN routing is presented in the diagram. Flow of operation starting from data collection till the findings of the proposed work as comparison of predicted and actual success rate of Interest packet is depicted.

necessarily relying on the actual producer of the data. Certainly, authenticity and security need to be preserved for such data retrievals. The ICN (Jacobson et al. 2012; Xylomenos et al. 2014) is envisioned to enable access of data from anywhere keeping in mind all the necessary security precautions. The ICN is focused on reducing content delivery latency and increasing throughput by enabling content (Data) serving to the client from any

data store in the network instead of relying only on data producers (or actual servers). In order to do so, it suggests several changes in the current Internet architecture like unique content naming, name based routing, self-secured data, etc. Consumers or clients in the network search a content (data) by specifying the content name instead of using host IP address of the content producer as in traditional TCP/IP network. Any node in the network having the requested content may serve the requester by sending the data to it. As a potential future architecture, the ICN opens up ample of opportunities to research as none of the protocols are matured enough. Out of many such areas of research, naming (Mangili, Martignon, and Capone 2015), caching (Abdullahi, Arif, and Hassan 2015; Mick, Tourani, and Misra 2016; Nguyen et al. 2019), routing (Banerjee, Kulkarni, and Seetharam 2018; Coulom 2007; Detti et al. 2018; Modesto and Boukerche 2018; Wang et al. 2012) and security (Goleman, Boyatzis, and Mckee 2019, 2019) are few of the topics that is drawing attention of many researchers. The work discussed in this paper is focused on ICN routing in general and forwarding Interest packets in particular. Following, a brief description of data retrieval in ICN in Named Data Networking (NDN) architecture (Jacobson et al. 2012).

The underlying architecture of this paper to realize the proposed approach for Interest forwarding follows the structure described in NDN architecture (Fayazbakhsh et al. 2013; Mahmudul Hoque et al. 2013; Modesto and Boukerche 2018; Wang et al. 2012). Following the said architecture, a client in the network generates an 'Interest' packet containing the name of the desired data (or content shown in Figure 1,2) (Coulom 2007). Every intermediate CR forwards the 'Interest' packet toward a location from where the requested content may be retrieved. Once the requested data is found, it is sent back to the client in the reverse path of the 'Interest' packet propagation. Hence, the routing process involves two activities: the forwarding of Interest packets to the content store and the sending the data back toward the requester. To forward Interest packets every CR maintains a Forward Information Base (FIB) table as per the specification of NDN architecture. Forwarding of Interest packets in right direction toward the content cache helps in retrieving data with shorter delay and maximized throughput. However, the Interest forwarding module in ICN by default does not provide any mechanism to know in advanced which path of propagation is better so that the content could be retrieved in less time. If the forwarding CR can figure out the best interface to forward the Interest packet to end up with successful retrieval of searched data, then data access process will be quite fast.

The research question addressed in this paper is "does machine learning models help in predicting the contents' cached locations in ICN?." Our objectives are to apply machine learning models on a routing and forwarding dataset of ICN network to forecast location information. And, to verify the models' prediction through a simulation testbed. In order to analyze the stated

question, we apply Support Vector Machine (SVM) on the ICN routing dataset. Three variations of SVM are applied on the available dataset and their prediction performance is evaluated as a first step of the research. Then in the second phase of the work, the best variation of SVM which shows maximum accuracy, is integrated in ICN Interest forwarding strategy. This integration helps in verifying the predictions made by the machine model. The observation in the simulated testbed depicts a significant improvement in the performance of content retrieval while SVM is integrated with ICN forwarding approach.

### **Motivations**

This work is motivated by several factors. First, ICN is drawing the attention of many researchers as a game changing architecture for the future Internet. However, many of the areas in this area are still not sufficiently explored. The Interest forwarding technique is one of such unexplored areas. Second, machine learning algorithms for modeling behavior of complex systems is another prominent area. Machine learning-based prediction models are proven methods of forecasting future trends. The ICN in general and the routing in particular is undoubtedly a complex system and seems to be well modeled with the help of machine learning algorithms. Moreover, there is a need for an efficient Interest forwarding mechanism in ICN for its wide acceptance and deployment. We believe, if the Interest packets are smartly forwarded, the routing in ICN will be improved. Motivated by the above mentioned facts, the presented paper models the Interest packet forwarding strategy in ICN network using ML algorithms. The CRs take help of the designed model to judge the accurate location of the searched content. The contribution of the work is majorly focused on modeling the success rate of Interest packets in finding the content using Support Vector Machine (SVM). The objective is to maximize the success rate which in turn minimizes content search time and maximizes throughput. Moreover, the comparison of routing with and without applying machine learning model is also presented for better interpretation of the idea presented here.

### **Research Contributions**

The contributions of the proposed work are as follows:

- Use of Support Vector Machine (SVM) to predict the probable cache location of searched data to make content retrieval faster.
- Dataset for applying the ML model is created from the simulation using *ndnSim* 2.0. The dataset comprises of 10 K data points having 10 features.

- Success prediction of Interest packets using linear, RBF, and polynomial kernel for comparing the prediction accuracy.
- Results of ML prediction is further validated through simulation.

Although there is a lot of research going on in ICN routing and forwarding, the integration of ML techniques into ICN Interest forwarding to predict contents' location is not widely explored. Few of the works such as Yao et al. (2019), Khelifi et al. (2019), Liu et al. (2017), are found in applying AI techniques in the ICN environment. However, their focus is completely different from ours. These works focus mainly on cache management or security provisioning. In the contrast, our proposal deals with predicting the probable locations of the requested content and guides the Interest packet to appropriate location. Such guidance improves the content retrieval time and hence increases efficiency of ICN. This work not only predicts the location information through ML techniques, it also verifies the predicted results by integrating SVM into Interest forwarding techniques.

The dataset used for modeling is prepared from a ns-3 based *ndnSim* simulation testbed designed to demonstrate ICN forwarding mechanism. The produced dataset comprises of 10 K data points with 10 features as listed in Table 2 described in Section III. Three variations of SVM are used in predicting the content locations from the dataset. The best version shows an accuracy of 98% and hence that said version is integrated with the forwarding strategy (details of the process is discussed in Section IV). The performer SVM variation is then integrated with Interest forwarding and predictions of machine learning models are verified. The Comparisons of SVM integrated forwarding mechanism is done with normal ICN forwarding techniques without integrating SVM. The comparison shows that the content retrieval time reduces around 10% in SVM enable forwarding compared to normal ICN forwarding.

## Organization

Paper is organized as follows. Work related to the use of machine learning in network in general and in ICN in particular is discussed in Section II. The problem description, dataset generation, and the fundamentals of SVM is found in Section III. Predicted results from SVM model and implementation is presented in Section IV. Validation of ML prediction is given in Section V. The paper is concluded with a discussion in Section VI.

## Related Work

Machine learning (ML) is one of the most promising artificial intelligence tools, conceived to support the prediction of behavior of a system. In recent years, ML is seen to be applied in every possible field. The networking system is one of such key infrastructures to use machine learning. ML is noticed to apply in various security analysis, as in IDS, traffic prediction for congestion control, and many more (Boutaba et al. 2018; Jain et al. 2020; Wang et al. 2018; Yap et al. 2020). Recently, few works in applying ML in vehicular networks and SDN-enabled networks for resource management is also noticed. There are few literatures reported on the application of ML in ICN as well (Khelifi et al. 2019; Liu et al. 2017; Xu et al. 2019; Yao et al. 2019). The focus of this paper is to apply ML techniques for ICN in determining the success rate of Interest packets. Few relevant research in the same line of directions is discussed in this section.

The work of Yao et al. (2019) presents a brief discussion on some related technologies of AI in the ICN including, routing, naming, security, and content distribution. An analysis on future directions in this area is also highlighted. Ample of ML based research in ICN cache enhancement is noticed. A study on the applicability of Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Reinforcement Learning (RL) with ICN is carried out in Khelifi et al. (2019). A Deep-Learning-based Content Popularity Prediction (DLCPP) is proposed in Liu et al. (2017) to predict content popularity. It creates a distributed and reconfigurable DL network of SDN switches for prediction. Deep Q-Networks (DQN) which is based on Q-learning is adopted to provide caching solution in Xu et al. (2019). Authors have considered an SDN IoT application for the deployment the protocol. The effectiveness of the proposed solution is verified through simulation. The Deep Cache framework for content caching in ICN is described in Narayanan et al. (2018) through RNN model to boost cache performance. A Long Short-Term Memory (LSTM) encoding is the heart of this approach. These researches do not explore routing or forwarding. Mostly, the above mentioned researches are trying to improve caching process through the use of AI-based techniques. Undoubtedly, caching is one of the basic concerns to make serving possible from anywhere in the network. But, unless the request for content is forwarded efficiently, the main goal of faster access of content cannot be achieved. Hence, there is a necessity to explore the content request forwarding process in ICN routing.

Although an considerably important concern, work on efficient forwarding of Interest packet is comparatively less investigated (Banerjee et al. 2017; Lehman et al. 2016; Zhu and Afanasyev 2013). Certainly, an efficient selection of path to the content significantly improves content



**Table 1.** A comparison of similar work with proposed forwarding strategy.

Reference	Objective	Contribution of the work	Comparison with our work	Use of ML model
Banerjee et al. (2017)	A routing strategy based on characteristic time of content in a cache to improve content search latency.	The proposed routing is dependent on the underlying caching strategy. Provides a probabilistic approach for content retrieval.	This work does not use any heuristic knowledge to predict content location. <i>In our work</i> we use a priori knowledge recorded in dataset.	No machine learning approach is used in this work like in our approach.
Zhu and Afanasyev (2013)	Exploits the features of the NDN naming rules from condensed cryptographic dataset which securely route data.	This work deals with security provisioning in forwarding and routing content.	Although they use heuristic dataset they apply the priori knowledge to secure data routing. <i>Like ours</i> , they do not deal with the access latency of contents.	No ML prediction models are used in the work.
Lehman et al. (2016)	An Adaptive Smoothed RTT-based Forwarding (ASF) mechanism to mitigate hierarchical routing path selection.	An experimental investigation is demonstrated for packet delivery delay and overhead under and compared with NLSR.	No special care is taken while forwarding Interest packet to search content. <i>In our approach</i> , ML model helps in predicting the forwarding path.	ML models are not included in their discussions.
Proposed Work	A machine learning based forwarding strategy with prediction of data availability in cache locations.	A Interest forwarding approach guided by ML model to increase success rate of content retrieval and minimizes search latency.	--	No ML prediction models are used in this work (Zhu and Afanasyev 2013).

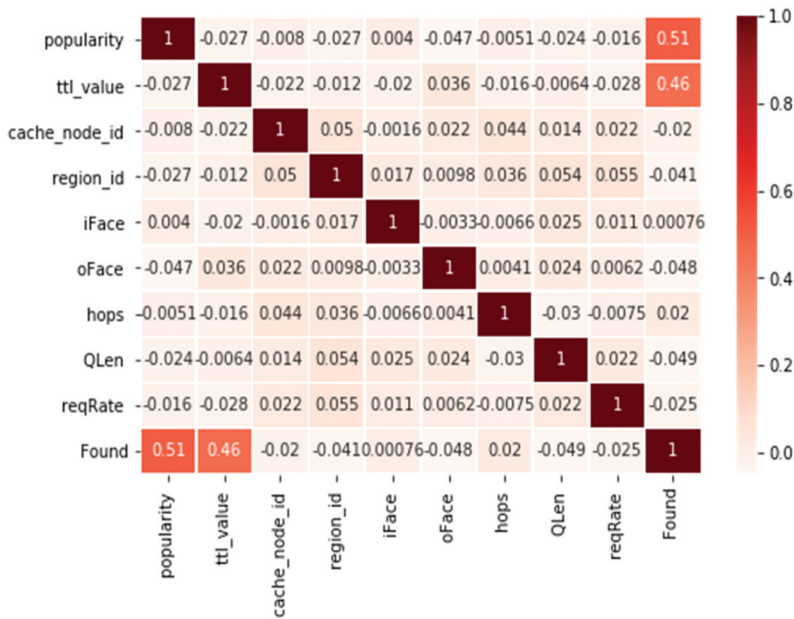


access time and network overhead. The work of Banerjee et al. (2017) and Zhu and Afanasyev (2013) emphasizes on this idea by forwarding Interest packets to the suitable nearby content cache. Further, if machine learning models are used in content searching, the efficiency is supposed to be remarkably high. However, to the best of our knowledge, there is no research found in the domain of ICN to predict the location of a searched content using ML techniques. This fact motivates us to carry out this research. SVM technique is used to predict the cache where the searched content is stored. Based on the suggestion by the machine model, the Interest packet is forwarded in the direction of the content. For this purpose, first the dataset is created from an simulation environment (Section III, (b)). The SVM model with different kernels (linear, RBF and polynomial) are applied on the dataset to predict contents' cached locations. Based on the results of the machine based analysis, a model is designed and integrated with the interest forwarding algorithm for ICN. The designed model is implemented and results obtained from the implementation is used to verify the predictions of the SVM-based machine learning model.

The Table 1 shows a list of similar work with their primary objectives, contribution, and difference with the proposed work. Facts are compared with our proposed scheme for better understanding of the benefits.

**Table 2.** List of features in the dataset with their description. These are the features that captured at the time of collecting data from the simulation environment designed for ICN network using *ndnSim*. Later these features are analyzed to find the most important features to classify the dataset.

Feature name	Description
<i>nodeID</i>	Unique identifier of the CR node that generates the Interest packet.
<i>contentName</i>	Unique name given to a content. Searching is done based on this name.
<i>popularity</i>	Every content has a popularity value computed based on its' access frequency. The interest packet carries the last known popularity value for the requested content.
<i>contentSize</i>	The size in byte of the requested content and included in the Interest packet. The parameter 'contentSize' propagates through Content Advertisement (CA) message disseminated periodically by actual content producers. The detailed description of the CA process is not explained in this paper as this variable has no direct impact on the performance of the work presented here. Interested readers may refer to (Marandi et al. 2017; Quan et al. 2014).
<i>ttlValue</i>	The last known time of the requested content from the time of its caching in the network. This value is computed by the underlying caching mechanism.
<i>inIF</i>	ID of the interface through which the Interest packet is received.
<i>outIF</i>	ID of the interface through which the Interest packet is forwarded.
<i>timeofReturn</i>	Time period in which the response to the Interest is received by the CR.
<i>queLength</i>	Queue length of the outgoing interface through which the Interest packet is forwarded.
<i>hopTraversed</i>	Count of total number of hops traversed by the Interest packet so far including the current CR.
<i>reqArrivalRate</i>	The frequency of Interest packet arrival at the time of forwarding the packet.
<i>dstCR</i>	The ID of the CR from where the data is received.
<i>Success</i>	Target variable. Set to 0 initially to indicate not successful. Once the CR receives the response it sets to 1.



**Figure 3.** Correlation of features for determining the classification of data points. The two features namely 'popularity' and 'ttl\_value' shows highly related correlation with values of 0.51 and 0.46, respectively. These two features are used for SVM classification in the later part of this work.

## Problem Formulation and System Model

This section discusses the problem formulation and system model. It is divided into four subsections. First, we discuss the problem specification followed by detailed dataset description. In subsection three, the mathematical interpretation of SVM in success prediction is presented. The fourth subsection describes the system model with a block diagram.

(a) *Problem specification:* The Interest packet forwarding process is modeled as a binary classification problem. One class defines the successful (Class 1) retrieval of content and other is unsuccessful (Class 0). The objective is to predict accurately the successful retrieval of content from the training and testing datasets. The SVM is used to model the problem as it is one of the efficient ML algorithms.

b) *Dataset description:* The dataset used in this research is generated from a simulation topology designed in *ndnSim2.0*, a *ns-3* based simulator. A topology identical to US26 network is designed with 26 CR nodes and 10 clients. These clients generate content requests for 1000 unique named contents. Data for analysis is gathered in 10 different CRs in the topology with Centrality betweenness (Delvadia, Dutta, and Ghinea 2019; Lal and Kumar

2018) from 3 to 5 to capture desired data for modeling (CRs marked with dotted circle in Figure 3). All the data are later combined to a single dataset and feed into the SVM model. The final dataset comprises of unique records identified by the content name. Please note that the data points in the dataset includes information only when the content is retrieved from any intermediate cache. If the data is obtained from the producer then it is not included in the dataset. Because the intention of the designed model is to find the probability of accessing data from a cache rather than the actual producer. The dataset contains 10 K records and 10 features. From the dataset, 70% data is used for training and 30% is used for testing the models. The description of features is given in Table 2.

*SVM in success prediction:*

Let us consider a dataset  $D$  as

$$D = \{x_i, y_i\}_{i=1}^n$$

$$x_i \in \mathbb{R}^d, y_i \in \{+1, -1\}$$

Where,  $n$  is the data points,  $x_i$  is the  $i^{\text{th}}$  data point,  $d$  is the dimension and  $y_i$  is the binary target variable that takes positive class (+1) or negative class (-1). The goal of the SVM is to find the optimum hyperplane [28] that can best separate the data points into given two classes. The hyper plane is represented as

$$y_i = W^T X + B \quad (1)$$

Where,  $X$  is the matrix representation of the dataset  $D$ ,  $W$  is the weight vector, and  $B$  is the bias vector. To obtain the optimum hyper plane both  $W$  and  $B$  should be selected optimally. Based on the equation of the hyper plane, the following hypothesis is defined:

$$h(x_i) = \begin{cases} +1 & \text{if } wx + b \leq 0 \\ -1 & \text{if } wx + b < 0 \end{cases} \quad (2)$$

Following the hyperplane, a point either on the hyperplane or above it, is classified as a positive class. Whereas, point below the hyperplane is classified as negative class. There may be multiple hyperplanes exist to separate the two classes, but we are interested in finding the optimal plane. Hence, there is a need of SVM optimization technique. Let us define

$$f = y(wx + b) \quad (3)$$

The sign of  $f$  (or polarity) is positive if the point is correctly classified and negative if incorrectly classified. The geometric margin  $\gamma$  is computed by dividing  $f$  with the norm of the weight vector  $w$ , such as

$$\gamma = y \left( \frac{wx + b}{\|w\|} \right)$$

With regards to the given dataset  $D$ , considering  $m$  training data points in the training set, the  $\gamma$  is

$$\gamma = \min_{i=1,\dots,m} \left\{ y_i \left( \frac{w}{\|w\|} x_i + \frac{b}{\|w\|} \right) \right\}$$

The above equation selects the largest margin for minimum values of  $w$  and  $b$  and considered as optimal hyperplane. The optimization problem may be rewritten as

$$\max_{w,b} \gamma = \max_{w,b} \left\{ y_i \left( \frac{wx_i + b}{\|w\|} \right) \right\}$$

or

$$\max_{w,b} \gamma = \max_{w,b} \left\{ \left( \frac{1}{\|w\|} \right) \right\}$$

Subject to,  $y_i(wx_i + b) - 1 \geq 0$  for  $i = 1 \dots m$ . Alternatively, it may also be represented as

$$\max_{w,b} \gamma = \min_{w,b} \|w\|$$

Which is equivalent to,

$$\min_{w,b} \frac{\|w\|^2}{2} \quad (4)$$

Subject to,  $y_i(wx_i + b) - 1 \geq 0$  for  $i = 1 \dots m$ . The above optimization function with given inequality constraint may solved using LaGrange formulation. The Lagrangian function is represented as

$$L(w, b, \alpha) = \frac{\|w\|^2}{2} - \sum_{i=1}^m \alpha_i y_i \{wx + b - 1\} \quad (5)$$

Where,  $\alpha_i$  is the Lagrangian multipliers and  $\alpha_i \geq 0$ . The optimization problem now becomes,

$$\min_{w,b} \max L(w, b, \alpha)$$

Subject to  $\alpha_i \geq 0$  for  $i = 1 \dots m$ . The LaGrange multipliers satisfies the following conditions:

- i.  $\sum_{i=1}^m (\alpha_i - \alpha_i^*) = 0$
- ii.  $\bar{\alpha}_i^1, \alpha_i^* \in [0, C[$

iii.  $\alpha_i, \alpha_i^* \neq C$

For condition (iii),  $|f(x_i) - y_i| < \epsilon$ , and  $\alpha_i, \alpha_i^*$  must be zero. It is stated that the model can tolerate lower than  $\epsilon$ . So, all  $x_i$ 's need not be included in the training data set rather, the  $S_v < S$  training points are sufficient to train the model. These set of points  $x_i$  in  $S_v$  who meets the condition  $\alpha_i, \alpha_i^* = 0$  forms the *support vectors* for the model. Our objective is to find the values of  $w$  and  $b$  for which we have maximum value of the LaGrange function  $L$ . For that, the function is differentiated first, it is differentiated with respect to  $w$  and then with respect to  $b$ . So, it derives the following two equations:

$$\frac{d}{dw}L(w, b, \alpha) = w - \sum^{\alpha_i} y_i x_i = 0$$

$$\frac{d}{db}L(w, b, \alpha) = - \sum^{\alpha_i} y_i = 0$$

Or

$$w = \sum^{\alpha_i} y_i x_i$$

$$\sum^{\alpha_i} y_i = 0$$

Substituting  $w$ , and  $b$  in  $L$ ,

$$w(\alpha, b) = \max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j .x_j$$

Once we get  $w$ , to compute the value of  $b$ , we got:

$$y_i (wx^* + b) - 1 = 0$$

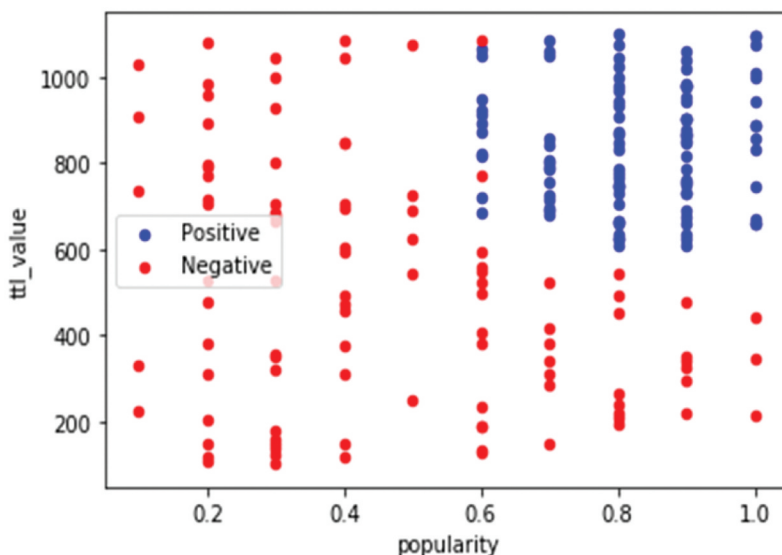
Multiplying both sides by  $y_i$  and putting  $y_i^2 = 1$ , generates

$$b = y_i - wx^*$$

Hence,

$$b = \frac{1}{S} \sum_{i=1}^m y_i + wx \tag{6}$$

$S$  is the number of support vectors selected to use to draw the hyperplane. The SVM does not provide any restriction on the type and number of features to be selected for predicting the results. Taking the above mentioned mathematical basis the SVM is used in the dataset to derive a prediction model for the given dataset.



**Figure 4.** Scatter plot of the ICN dataset to observe the separability for classification. correlation of features for determining the classification of data points. The data could be linearly separated with little misclassification. So, SVM is used for modeling.

### **System Model**

The dataset collection phase is as described in subsection b above. This phase produces the raw data. To make raw data suitable for feed into the ML model it is refined by removing redundant data points and irrelevant columns. The cleaning phase generates the dataset that is suitable for applying to the model. This dataset is used as an input to the SVM with RBF, Linear, and Polynomial kernel. Simultaneously, a simulation topology is designed and success rate is observed. The observed result is compared with ML predicted result to draw a conclusion of the proposed work. In next section results of ML model as well as validation of the same is presented.

## **Results of Machine Learning Model**

### **Correlation of Features**

The dataset comprises of 10 features. But, not all of them contributes to the classification. To figure out the most relevant features for classification, a heatmap of the dataset is plotted. Figure for correlation-map (Figure 3) shows the observation. It depicts that the *popularity* and *ttl\_value* has highest correlation to the classification. Hence, these two features can well predict the behavior of the model without considering other features.

**Table 3.** Comparison of prediction and accuracy of the content retrieval for different features. This tables shows the best results with two features, popularity and ttl\_value. (Parameters Gamma = 0.7, C = 100; all values are in %-age).

Feature count	Selected features	Failure prediction	Success prediction	Accuracy
	RBF kernel			
1	ttl_value	84	48	78
1	popularity	84	46	78
2	<b>popularity, ttl_value</b>	<b>86</b>	<b>52</b>	<b>80</b>
3	popularity, ttl_value, QLen	82	69	81
12	All features	79	0	79
	Linear kernel			
1	ttl_value	93	78	90
1	popularity	93	78	90
2	<b>popularity, ttl_value</b>	<b>93</b>	<b>88</b>	<b>92</b>
3	popularity, ttl_value, QLen	92	86	91
12	All features	93	78	90
	Polynomial kernel (with degree 3)			
1	ttl_value	79	0	79
1	popularity	84	46	78
2	<b>popularity, ttl_value</b>	<b>79</b>	<b>100</b>	<b>90</b>
3	popularity, ttl_value, QLen	79	0	51
12	All features	79	0	79

### Separability Check

After finding the two most important features for classification, an analysis of the dataset with respect to the selected features is carried out to see whether the dataset is linearly separable or not. Because, to select the type of SVM kernel it is necessary to know the degree of linearity of the dataset. For the purpose a scatter plot of the dataset with popularity and ttl\_value is observed. The result is shown in Figure 4. The plot reveals that the dataset is separable with little miss classification. Hence, a SVM may suitably be used for classification. But to find the best kernel for proper classification, different SVM kernels are tried and results are presented in Table 3.

Step	Algo ml_model_for_prediction()
1	<b>Input:</b> Content_name, popularity, ttl_value <b>Output:</b> olFace
2	Start
3	olFace = null
4	If (popularity $\geq$ 0.6 and ttl_value $\geq$ 400) then
5	olFace = lookup(popTab, popularity)
6	Else
7	olFace = null
8	Endif
9	If (olFace $\neq$ null) then
10	Forward(Interest(Content_name, olFace))
11	Else
12	Forward(Interest(Content_name))
13	Endif
13	Stop

### Results of SVM Classifier

The Table 3 shows a comparison of negative and positive class prediction along with the accuracy of prediction for different feature counts. It also shows the results of three different kernels used in SVM namely, the RBF, the Linear, and Polynomial (with degree 3). The table demonstrates the best performance with two features *popularity* and *ttl\_value*. This result verifies the results of heatmap presented in Figure 3.

In Table 3, the 'Failure Prediction' implies that response to the 'Interest' is failed and value of 'Found' is 0 in the dataset. On the other hand, 'Success Prediction' implies the response to the 'Interest' packet is successful and value of 'Found' is 1. While implementing the model in the routing module, the importance of 'Success prediction' is more. This analysis is primarily focused on the prediction of successful dissemination of Interest. The results of SVM classifier are depicted in Table 3, reveals several facts regarding the behavior of the ICN routing process. First, for all the three kernels tested, better results are found with two features, the 'popularity' and 'ttl\_value.' As far as 'Success Prediction' is concerned, the polynomial kernel shows best result followed by 'Linear' kernel. However, the accuracy of prediction in 'Polynomial' kernel is much better compared to 'Linear' kernel. Hence, it may be concluded that the both the kernels are potential candidate for modeling the problem under consideration. The parameter 'gamma' is taken as 0.7 and 'C' is 100. These two parameters are selected after examining results for multiple parameters and values that gives the best values for all kernels are finally selected.

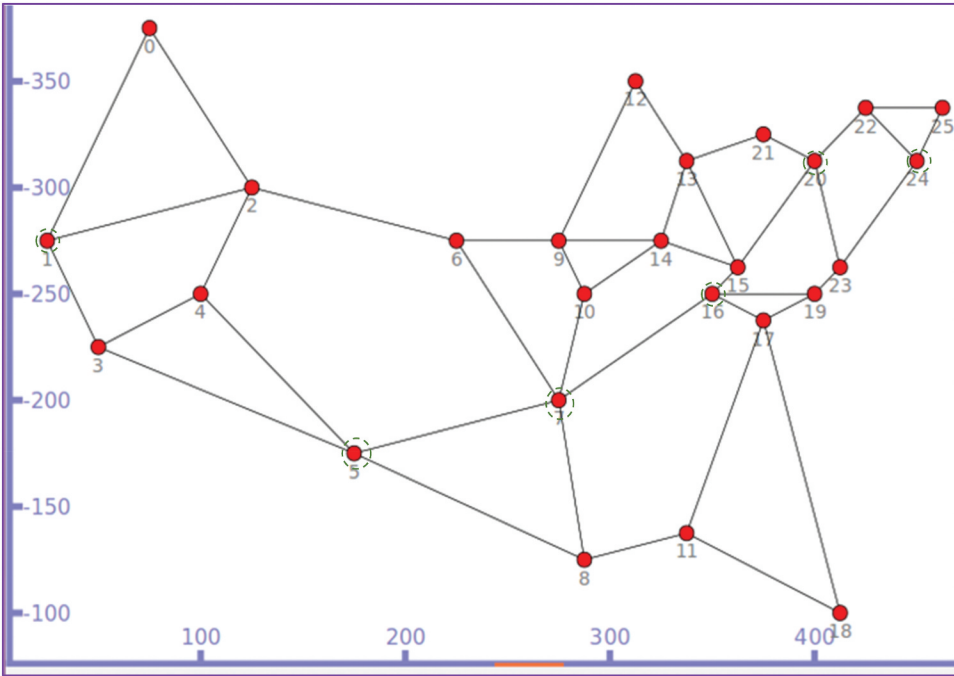
In this paper, the prediction of the machine model is verified with implementation in the simulation environment. So, a model is designed based on the prediction suggested by the SVM algorithm and is integrated in the *ns-3* based simulation. The objective is to check the accuracy of prediction of the machine model in semi real time environment. For the said purpose, a model is designed so that it can be implemented in *ns-3* code. The model description is given next.

### The Model

Based on the findings of the analysis through machine learning algorithm (SVM in particular), a machine model is designed to integrate with our *ns-3*-based simulator. As suggested by the SVM the '*popularity*' of content and caching period '*ttl\_value*' carried by the Interest packet is used in the machine model. The description of the model follows.

The CR maintains a table (implemented using hash table) to keep the output interface for popular data. This Table is formed by collecting data from the previous Interest packet forwarding and last known success record. In the algorithm (Step 1) based on the prediction, the model suggests the



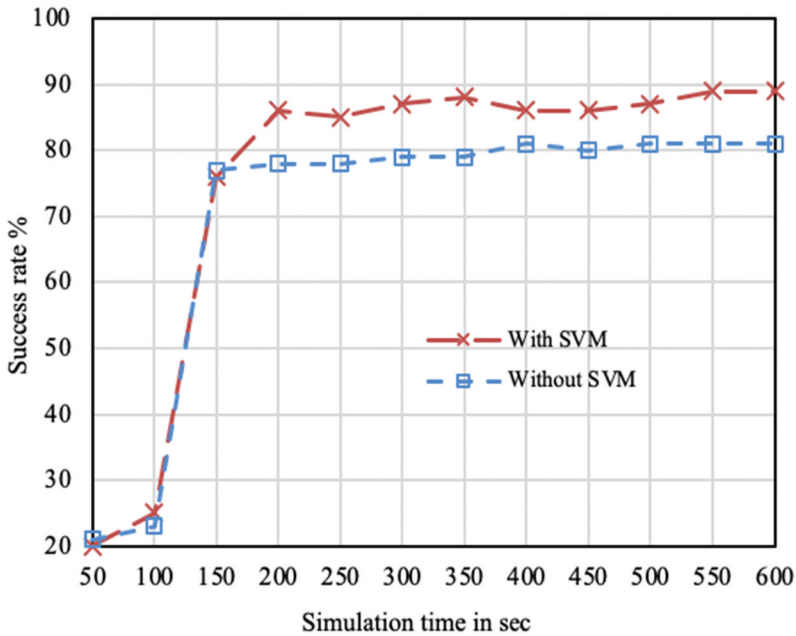


**Figure 5.** A sample simulation topology based on US26 network. Approximately 1000 unique contents with a size of 5000B is taken for simulation. The cache store size is of 50–150 contents. Nodes with dotted circles have betweenness centrality (BC) value in the range 3–5 and results are collected in these CR nodes.

**Table 4.** Simulation parameter.

Parameter	Typical values
Network size	US26 network with 26 nodes
Number of contents	1000 contents
Content size	5000B
Cache size	50–150 contents
Betweenness centrality	1–5
Simulation time	100–650 Sec

possibility of finding the content in a cached location (recall that the dataset includes only those entries where data is found in the cache. Entries for data retrieval from the producer is excluded from the dataset). Then it searches its database for the out interface taking the popularity as the key value. There may be multiple entries for out interface (max 5 is assumed) and it picks the first interface to send the ‘Interest’ packet. However, if that out interface has a queue length larger than a prespecified threshold (assumed 30 in the simulation) then it selects the next out interface listed in the table. The default ICN checks the FIB table to see if the ‘Interest’ packet is already forwarded for the newly requested content. If yes then it does not forward the ‘Interest,’ rather it waits for the reply. But in this proposed scheme the algorithm checks the time



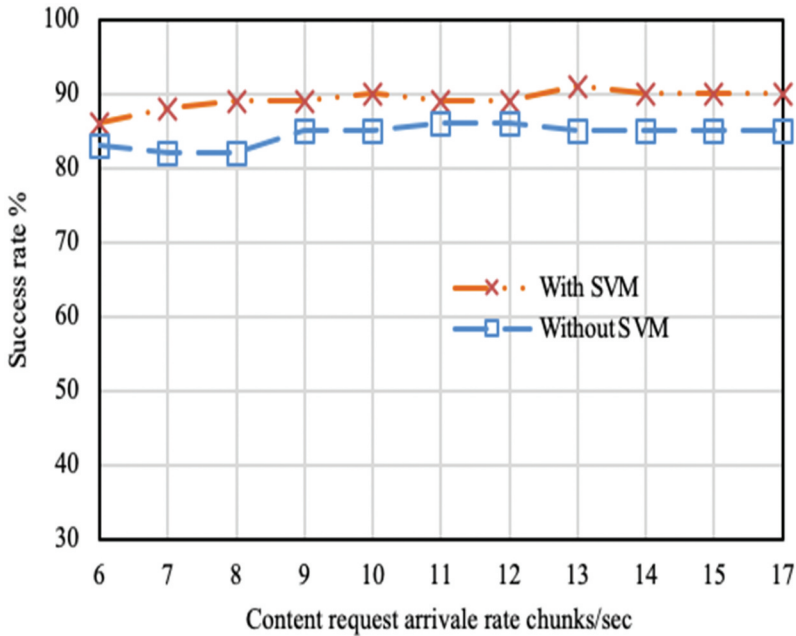
**Figure 6.** Interest packet success rate over simulation time. The plot considers those successful processing whose content is found in intermediate cache. Serving from the actual producer is not counted in the above results.

of sending the interest packet. If it is long time back as specified by a threshold time then it opts for resending. But it uses a different interface then it sent earlier. Of course, the out interface is selected from the list in the table.

As it is seen earlier in the [Table 3](#) that the prediction is 100% for finding the data cache with an accuracy of 90% (considering the best case with polynomial kernel). Hence forwarding the Interest packet to oFace as discussed in the algorithm seems to be worthy. Moreover, the model also suggest the negative response prediction of 79% with 90% accuracy. It implies if the model suggest that the requested data is not cached so far then the probability of the data is not in the cache is 0.96. Hence, the Interest packet should be forwarded following the underlying forwarding strategy. However, as a proof of concept (PoC), we have implemented the algorithm *ml\_model\_for\_prediction* in the simulation environment and results are compared to validate the machine model.

### Simulation Results for Validating the Machine Model

- (a) *Simulation setup*: For verification of predicted results the same topology which is used for dataset preparation described in [Section III](#).(b) is used. The setup is designed in the *ns-3* based



**Figure 7.** Interest packet success rate over request arrival rate. The plot considers only successful Interest requests served from the cache. Results are recorded after 100sec of simulation progress. By this time elapse most of the caches are filled with sufficient amount of data. Hence, there is a high possibility of getting data in cache.

NDN simulator, ndnSIM 2.0 (Afanasyev, Moiseenko, and Zhang 2012). The topology is similar to US26 network with 26 CRs as depicted in Figure 5. A set of 1000 unique contents with content size of 5000 Bytes is modeled. The content requests follow Zipf distribution where the skewness parameter ranges from 0.1 to 1.2. The cache is considered empty initially and the capacity is ranging from 50 to 150 with an average of 70 contents. All simulations are executed for a duration of 600 seconds and content requests are started after 10 seconds of the simulation. Through this simulation it is aimed to validate the results of the machine model. Hence, all the parameters are kept same as that used in data collecting process (stated in Table 4).

- (b) *Success rate (with and without ML)*: It is the measure of what percent of the ‘Interest’ packet forwarding made by a CR is successful. Observation is done in 6 CRs (CR1(3), CR5 (4), 7(5), 16(4), 20(4), and 24(4). The values in the bracket indicates the BC value). The results shown are the average of success in all these routers. Please note that, we are counting only those cases where

data is obtained from the intermediate CRs. If content is found in the producer, then it is not counted as success. The results are shown against varying simulation time in [Figure 6](#).

Till 100 seconds of simulation success ratio does not match with the machine prediction. The reason is till date time the cache is not filled and majority of the content is retrieved from the producer. Over time, the caches are filled with popular contents and success ratio increases. In a condition where caches are sufficiently filled by contents the success ratio is between 78 and 90 approx. The presented result depicts that the SVM integrated version of the model works with nearly 10% increased success ratio compared to the model without SVM integration. It implies that the SVM model can predict the success reasonably.

In [Figure 7](#), the success ratio is shown with respect to request arrival rate. The path followed by the machine enabled methods shows better performance over other. In this case also the improvement is around 8–10% for SVM enabled version of Interest forwarding compared to non-SVM version. In conclusion, our machine model can well predict the success ratio of any Interest packet forwarding mechanism.

## Discussion and Conclusion

In this paper we tried to address the question “does machine learning models help in predicting the contents’ cached locations in ICN?” As the ‘Interest’ packets are used to search content in ICN, the work in this paper tries to forecast the success of the ‘Interest’ packet in finding contents through applying SVM model. We have found that yes, it is possible to enhance the performance of the content search process by integrating Interest forwarding with mL techniques. Our claims are supported by applying SVM into a dataset collected from ns-3-based ICN testbed (Section). A dataset with 10 K records and 10 features are used with training-testing ratio of 70–30. A dataset with 10 K records and 10 features are used with training-testing ratio of 70–30. Three variations of SVM model are applied on the available dataset and their prediction performance is evaluated during the research. Based on the outcome of the SVM, a model is designed and integrated in the ICN forwarding module. The findings of machine models are verified by integrating the best variation of SVM with forwarding strategy and testing again in simulated environment. This integration helps in verifying prediction made by the machine model.

The observation in the simulated testbed depicts a significant improvement in the performance of content retrieval while SVM is integrated with ICN forwarding approach. Based on the prediction made from apriori datasets, a CR in the network makes use of this prediction to forward future ‘Interest’

packets. A comparative success ratio with and without applying the machine based prediction is presented. The results shows that the prediction sufficiently fits the simulated results and significant improves in the performance of content retrieval is noticed. The results demonstrates that the use of SVM is advantageous compared to normal ICN based 'Interest' packet forwarding technique.

In this paper we have used the NDN architecture where the contents searching is done by generating Interest packets. Other approaches, those who use Name Resolution Server (NRS) for content searching have comparatively less problems with finding the nearest cached copy of a content. The reason of selecting the NDN for this paper is the clean slate implementation of ICN approach without relying on TCP/IP protocol stack. In the future, we are planned to investigate the impact of machine models on such protocols that relies on NRS for content searching.

## Disclosure Statement

No potential conflict of interest was reported by the author(s).

## ORCID

Nitul Dutta  <http://orcid.org/0000-0002-3399-0042>

## References

- Abdullahi, I., S. Arif, and S. Hassan. 2015. Survey on caching approaches in information centric networking. *Journal of Network and Computer Applications* 56:48–59. doi:10.1016/j.jnca.2015.06.011.
- Afanasyev, A., I. Moiseenko, and L. Zhang. 2012. ndnSIM: NDN simulator for NS-3. *NDN, Technical Report NDN-0005*.
- Banerjee, B., A. Kulkarni, and A. Seetharam. 2018. Greedy caching: An optimized content placement strategy for information-centric networks. *Computer Networks* 140:78–91. doi:10.1016/j.comnet.2018.05.001.
- Banerjee, B., A. Seetharam, A. Mukherjee, and M. Kanti Naskar. 2017. Characteristic time routing in information centric networks. *Computer Networks* 113:148–58. doi:10.1016/j.comnet.2016.12.009.
- Boutaba, R., M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo. 2018. A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities. *Journal of Internet Services and Applications*. doi:10.1186/s13174-018-0087-2.
- Chegini, H., and A. Mahanti. 2019). A framework of automation on context-aware internet of things (IoT) systems. *UCC 2019 Companion - Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, Auckland New Zealand. doi:10.1145/3368235.3368848.
- Cisco Systems, Inc. 2019. Cisco visual networking index: forecast and trends, 2017–2022 white paper. *Cisco Forecast and Methodology*, 2019,1-38.

- Coulom, R. 2007. Efficient selectivity and backup operators in Monte-Carlo tree search. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Turin, Italy. doi:10.1007/978-3-540-75538-8\_7.
- Delvadia, K., N. Dutta, and G. Ghinea. 2019. An efficient routing strategy for information centric networks. *International Symposium on Advanced Networks and Telecommunication Systems, ANTS, 2019-December*, Goa, India. doi:10.1109/ANTS47819.2019.9118123.
- Detti, A., L. Bracciale, P. Loreti, G. Rossi, and N. Blefari Melazzi. 2018. A cluster-based scalable router for information centric networks. *Computer Networks* 142:24–32. doi:10.1016/j.comnet.2018.06.003.
- Fayazbakhsh, S. K., Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. C. Ng, V. Sekar, and S. Shenker. 2013. Less pain, most of the gain: Incrementally deployable ICN. *ACM SIGCOMM Computer Communication Review* 43 (4):147–58. doi:10.1145/2534169.2486023.
- Goleman, D., R. Boyatzis, and A. Mckee. 2019. Information-centric networking: Evaluation and security considerations - RFC. *Journal of Chemical Information and Modeling*, 7945,1-40.
- Jacobson, V., D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard. 2012. Networking named content. *Communications of the ACM* 55 (1):117–24. doi:10.1145/2063176.2063204.
- Jain, D. K., A. Mahanti, P. Shamsolmoali, and R. Manikandan. 2020. Deep neural learning techniques with long short-term memory for gesture recognition. *Neural Computing and Applications* 32 (20):20. doi:10.1007/s00521-020-04742-9.
- Khelifi, H., S. Luo, B. Nour, A. Sellami, H. Moun gla, S. H. Ahmed, and M. Guizani. 2019. Bringing deep learning at the edge of information-centric internet of things. *IEEE Communications Letters* 23 (1):52–55. doi:10.1109/LCOMM.2018.2875978.
- Lal, K. N., and A. Kumar. 2018. A centrality-measures based caching scheme for content-centric networking (CCN). *Multimedia Tools and Applications* 77 (14):17625–42. doi:10.1007/s11042-017-5183-y.
- Lehman, V., A. Gawande, B. Zhang, L. Zhang, R. Aldecoa, D. Krioukov, and L. Wang. 2016. An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN. *2016 IEEE/ACM 24th International Symposium on Quality of Service, IWQoS 2016*, Tokyo, Japan. doi:10.1109/IWQoS.2016.7590394.
- Liu, W. X., J. Zhang, Z. W. Liang, L. X. Peng, and J. Cai. 2017. Content popularity prediction and caching for ICN: A deep learning approach with SDN. *IEEE Access* 6:5075–89. doi:10.1109/ACCESS.2017.2781716.
- Mahmudul Hoque, A. K. M., S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang. 2013. NLSR: Named-data link state routing protocol. *ICN 2013 - Proceedings of the 3rd, 2013 ACM SIGCOMM Workshop on Information-Centric Networking*, New York, US. doi:10.1145/2491224.2491231.
- Mangili, M., F. Martignon, and A. Capone. 2015. Optimal design of information centric networks. *Computer Networks* 91:638–53. doi:10.1016/j.comnet.2015.09.003.
- Marandi, A., T. Braun, K. Salamatian, and N. Thomos. 2017. BFR: A bloom filter-based routing approach for information-centric networks. *2017 IFIP Networking Conference, IFIP Networking 2017 and Workshops*, Stockholm, Sweden. doi:10.23919/IFIPNetworking.2017.8264842.
- Mick, T., R. Tourani, and S. Misra. 2016. MuNCC: Multi-hop neighborhood collaborative caching in information centric networks. *ACM-ICN 2016 - Proceedings of the 2016 3rd ACM Conference on Information-Centric Networking*, Kyoto, Japan. doi:10.1145/2984356.2984375.

- Modesto, F. M., and A. Boukerche. 2018. SEVeN: A novel service-based architecture for information-centric vehicular network. *Computer Communications* 117:133–46. doi:10.1016/j.comcom.2017.07.013.
- Narayanan, A., S. Verma, E. Ramadan, P. Babaie, and Z. L. Zhang. 2018. DEEPCACHE: A deep learning based framework for content caching. *NetAI 2018 - Proceedings of the 2018 Workshop on Network Meets AI and ML, Part of SIGCOMM 2018*, Budapest, Hungary. doi:10.1145/3229543.3229555.
- Nguyen, Q. N., J. Liu, Z. Pan, I. Benkacem, T. Tsuda, T. Taleb, S. Shimamoto, and T. Sato. 2019. PPCS: A progressive popularity-aware caching scheme for edge-based cache redundancy avoidance in information-centric networks. *Sensors (Switzerland)*. doi:10.3390/s19030694.
- Quan, W., C. Xu, J. Guan, H. Zhang, and L. A. Grieco. 2014. Scalable name lookup with adaptive prefix bloom filter for named data networking. *IEEE Communications Letters* 18 (1):1. doi:10.1109/LCOMM.2013.112413.132231.
- Talluri, L. S. R. K., R. Thirumalaisamy, R. Kota, R. P. R. Sadi, U. Kc, R. K. Naha, and A. Mahanti. 2021. Providing consistent state to distributed storage system. *Computers* 10:2. doi:10.3390/computers10020023.
- Wang, L., A. K. M. M. Hoque, C. Yi, A. Alyyan, and B. Zhang. (2012). OSPFN: An OSPF based routing protocol for named data networking. *NDN Technical Report NDN-0003*, July 2012.
- Wang, M., Y. Cui, X. Wang, S. Xiao, and J. Jiang. 2018. Machine learning for networking: Workflow, advances and opportunities. *IEEE Network* 32 (2):92–99. doi:10.1109/MNET.2017.1700200.
- Xu, F., F. Yang, S. Bao, and C. Zhao. 2019. DQN inspired joint computing and caching resource allocation approach for software defined information-centric internet of things network. *IEEE Access*. doi:10.1109/ACCESS.2019.2916178.
- Xylomenos, G., C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. C. Polyzos. 2014. A survey of information-centric networking research. *IEEE Communications Surveys and Tutorials* 16 (2):1024–49. doi:10.1109/SURV.2013.070813.00063.
- Yao, H., M. Li, J. Du, P. Zhang, C. Jiang, and Z. Han. 2019. Artificial intelligence for information-centric networks. *IEEE Communications Magazine*. doi:10.1109/MCOM.2019.1800734.
- Yap, N., M. Gong, R. K. Naha, and A. Mahanti. 2020. Machine learning-based modelling for museum visitations prediction. *2020 International Symposium on Networks, Computers and Communications, ISNCC 2020*, Montreal QC, Canada. doi:10.1109/ISNCC49221.2020.9297182.
- Zhu, Z., and A. Afanasyev. 2013. Let's ChronoSync: Decentralized dataset state synchronization in named data networking. *Proceedings - International Conference on Network Protocols, ICNP*, Texas, US. doi:10.1109/ICNP.2013.6733578.