



Article

An Intelligent Content Prefix Classification Approach for Quality of Service Optimization in Information-Centric Networking

Cutifa Safitri ^{1,*}, Yoshihide Yamada ¹, Sabariah Baharun ¹, Shidrokh Goudarzi ², Quang Ngoc Nguyen ³ , Keping Yu ³ and Takuro Sato ³

¹ Department of Electronic System Engineering, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, 54100 Kuala Lumpur, Malaysia; yoshihide@utm.my (Y.Y.); sabariah@utm.my (S.B.)

² Department of Advanced Informatics School, Universiti Teknologi Malaysia, 54100 Kuala Lumpur, Malaysia; gshidrokh2@live.utm.my

³ Department of Communications and Computer Engineering, Faculty of Science and Engineering, Waseda University, Shinjuku-ku, Tokyo 169-0051, Japan; quang.nguyen@fuji.waseda.jp (Q.N.N.); keping.yu@aoni.waseda.jp (K.Y.); t-sato@waseda.jp (T.S.)

* Correspondence: scutifa2@live.utm.my

Received: 1 March 2018; Accepted: 2 April 2018; Published: 9 April 2018



Abstract: This research proposes an intelligent classification framework for quality of service (QoS) performance improvement in information-centric networking (ICN). The proposal works towards keyword classification techniques to obtain the most valuable information via suitable content prefixes in ICN. In this study, we have achieved the intelligent function using Artificial Intelligence (AI) implementation. Particularly, to find the most suitable and promising intelligent approach for maintaining QoS matrices, we have evaluated various AI algorithms, including evolutionary algorithms (EA), swarm intelligence (SI), and machine learning (ML) by using the cost function to assess their classification performances. With the goal of enabling a complete ICN prefix classification solution, we also propose a hybrid implementation to optimize classification performances by integration of relevant AI algorithms. This hybrid mechanism searches for a final minimum structure to prevent the local optima from happening. By simulation, the evaluation results show that the proposal outperforms EA and ML in terms of network resource utilization and response delay for QoS performance optimization.

Keywords: information-centric networking (ICN); Intelligent classifications; artificial intelligence (AI); quality of service (QoS)

1. Introduction

Information-centric networking (ICN) is a future internet architecture that implements a naming scheme for content forwarding instead of referring the content to its location as in traditional host-to-host IP-based networks [1]. ICN transforms content data into first-class entities and loosens the required binding between content requesters and content providers. This communication paradigm shift is desirable to meet the expected exponential growth in the number of content exchanges on the Internet. With the growth of smart devices and next-generation networks, information becomes more accessible on a global scale. Particularly, the projection of global mobile data traffic will increase sevenfold, reaching 49.0 exabytes per month by 2021 [2]. As the amount of information is increasing day by day, the need to efficiently find and retrieve the most relevant content is becoming more challenging [3].

The major difference between IP and ICN is that the latter applies content name prefix format as the identifier for the forwarding process. Name prefix is a hierarchical segment that is concatenated by “/” as of the well-known uniform resource identifier (URI) full name representation. The general format of the content prefix in ICN can be depicted as ‘/publisherID/routerID /content_name/content_type’. The content name, itself, is composed of keywords that can be registered and bonded as the name identifier of the content. Nevertheless, as of the current IP-based Internet, users still cannot express the intended content request with exact key terms. Thus, to obtain the desired data, users need to try several keywords using the search engine then choose the content that is most related to them [4]. As current network devices require fast and efficient services, it is important to dynamically learn from user’s inputs, like a string of keywords as content attributes then estimate and return the most important name prefix to ensure quality of service (QoS).

This motivates the need for an intelligent classification mechanism to obtain the most appropriate content prefix component for content discovery improvement then accomplish QoS optimization. We focus on a generic framework which performs content prefix classification by utilizing artificial intelligence (AI) approach to evaluate the relevance between the inputted keywords and the desired content. This is a potential approach given that up to now, AI approach in ICN is still at early stage, though prior studies in this area suggest opportunities for intelligent processing of content prefix in ICN.

Different from other notable related work in this field, our study focuses on finding the most suitable intelligent approach that supports efficient classification in the application and network levels. In this paper, we evaluate and observe the performance of various AI algorithms, then discuss their classification performances with our selected QoS matrices. Specifically, the selected algorithms are well-known AI algorithms including evolutionary algorithm (EA), swarm intelligence (SI), and machine learning (ML) methods. Their performances are then evaluated under four different criteria, which are number of function evaluation, cost function, standard deviation, and computation time. From the extensive investigation using MATLAB, we select the most suitable intelligent classification methods for the proposed framework. Then, we show how the framework can optimize ICN efficiency by solving the localization problem in relevant comparable related AI-based work for classification.

The contribution of the proposal is as follows: (1) we propose a new generic intelligent classification framework in ICN by aggregating content requests from a large number of users to optimize QoS and (2) the simulation results show that the proposal improves the network efficiency in terms of reductions in network resource utilization and response delay by handling aggregated content packets for content dissemination. This acts as a potential approach towards the realization of ICN for the future internet architecture.

2. Related Work

This section presents the overview of fundamental concepts and features of ICN, Artificial intelligence (AI), and its implementation in ICN.

2.1. Information-Centric Networking

ICN is a promising candidate for the future internet architecture (FIA). ICN forwarding engine comprises of three primary data structures: Content Store (CS), the cache space of content nodes; Pending Interest Table (PIT), which stores the pending incoming request and its ingress face; and Forwarding Interest Base (FIB), the table that stores the forwarding egress of the potential content provider.

Regarding ICN concept, Content-Centric Networking (CCN) is a well-known research project in ICN initiated since 2007 [1]. The main idea of CCN is that a content request broadcasted by a user (namely Interest packets) contains the desired content name. When an Interest packet arrives, the CCN forwarding engine checks the information base to find the requested content’s possible providers which are the nodes with original content or its valid replica. Then the Interest packet is forwarded

through any outgoing interface that moves it closer to the user location [4]. In this research, we utilize Named-Data Networking (NDN) as it is an enhanced version of the CCN architecture, which has started in 2010 with detailed protocol and forwarding algorithms to form the fully functional prototype in ICN.

2.2. Artificial Intelligence

The most fundamental parts of ‘intelligence’ in AI engine are learning and adapting processes. By enabling the adaptive learning, the development and applications of Evolutionary Algorithm (EA) have made it become one of the fastest growing research fields in AI. EA includes genetic algorithm, biogeography-based optimization, and differential evolution. Recently, swarm intelligence (SI) algorithms, including ant colony optimization, artificial bee colony, and particle swarm optimization have also been proposed as optimization methods. Machine learning, which includes supervised, unsupervised and reinforcement learning, enables computers to modify and adapt their actions (such as making a prediction) so that the learning process can be more accurate.

For EA, genetic algorithm (GA) is based on genetic structure analogy and chromosomes’ behavior within a population. GA’s advantage is the flexibility in modeling both time-dependent and coupling constraints. However, since GA is a stochastic optimization algorithm, the optimality of its solution cannot be guaranteed. As a global optimization technique, GA can gain good initial convergence characteristics. However, it may slow down considerably once the region of optimal solutions has been identified [5].

The mathematical model bio-geography based optimization (BBO) describes how a species migrates, arises, and becomes extinct [6]. In BBO, the individual is termed as species and has the suitability index variables to evaluate its quality as a solution. As habitat suitability index improves, the species count increases, emigration increases, and immigration decreases. BBO has common characteristic features with GA. Operators in GA are crossover and mutation, whereas, in BBO, they are migration and mutation.

Differential evolution (DE) is a simple population-based search algorithm for global optimization with a minimum number of control parameters. DE is a powerful search engine in single objective optimization, but its usage in multi-objective optimization still raises some issues, because the use of differential evolution in such problems requires additional alternative encodings (e.g., combinatorial optimization problems) [7].

Regarding the swarm intelligent algorithms, ant-colony optimization (ACO) is based on the exploration principles of the ants’ foraging process from their nest to the food source by efficiently using their pheromones’ trail. Ants perform random walks for food, and when they reach the destination, the ants will return to their nest. While returning, a pheromone trail is produced leading back to the food source, and the following ants can follow that trail. The collaboration of suitable pheromones with stronger trail will further intensify and produce environmental changes towards the shortest path to food source [8].

The artificial bee colony (ABC) algorithm emulates the intelligent foraging behavior of the honeybee swarm [9]. A food source represents a possible solution to the problem optimization which corresponds to the quality of the solution. ABC algorithm has gains higher performance in both of the global and local searches for each of the iterations compared to other algorithms; hence, the probability of finding the optimal parameters is significantly increased.

Particle swarm optimization (PSO) was motivated by social behavior of birds (as particles) when attempting to get to an unknown destination [10]. The particles swarm through the search space and update their positions. Advantages of PSO include computational feasibility and effectiveness, smooth implementation, and consistency in performance. However, the PSO may lack the global search ability at the end of a run due to the utilization of a linearly decreasing inertia weight.

In supervised learning, the system is fed and learned from a provided set of examples with the correct responses. The supervised learning, such as multi-layer perceptron (MLP), aims to minimize the

error criterion, like the squares deviation error, based on the difference between targets and the outputs. Different from this, the unsupervised learning algorithm intends to discover the similarities between the inputs, in which inputs with common attributes are categorized into groups [11]. The reinforcement learning (RL) is placed between supervised and unsupervised learning. The RL algorithms are informed when the answer is wrong but do not get instructed on how to correct it [11].

2.3. Implementation of Artificial Intelligent in ICN

Shanbhag et al. used ACO as an optimization forwarding strategy in CCN for selective router service to promote load balancing in service-centric networking [12]. However, this approach did not consider CCN local traffic and redundant Interest packets from the ants in the network. Researchers in [13] extends ACO-multipath behaviors and addresses the probabilistic ant-routing mechanism to enable multipath transmissions for CCN nodes. Recent work in [14] shows optimization in ACO by using bidirectional ants to diffuse and exploit multiple content replicas. The study aims to obtain an optimal cache and efficient utilization of available cache resources within a specific area.

Study on PSO implementation in CCN applies PSO for the Forwarding Information Base (PSO-FIB) [15] to enhance the QoS of the forwarding experiences. PSO-FIB uses particles to maintain the forwarding probability of each entry in the FIB. Researchers in [10] proposed a hybrid scheme of PSO and K-means clustering algorithm over CCN to gain a fuzzy anomaly detection system for future kinds of security challenges in CCN.

The machine learning is applied in CCN in [16] to discover temporary copies of content items not addressed in routing tables, then forward requests to the best face by calculating Q values for exploration and exploitation in every hop. A study in [17] proposes Q-routing to address packet routing problem in dynamically changing networks. In particular, the authors propose RL based method to solve the problem of content placement and routing by employing the Q-routing with cost-to-go computation for the optimization of caching routing decisions.

The proposal in [18] evaluates the forwarding strategy in CCN based on Multi-Armed Bandits Strategy (MABS), a ϵ -greedy technique of RL. MABS probabilistically explores the network for each Interest packet request and exploits the acquired knowledge using the best-classified interface. Evaluation results show that MABS can reduce the number of hops to find content.

Work in [19] proposes a content discovery system, which is a content announcement based on deep exponential network and cache replacement algorithms. By applying a restricted Boltzmann machine, the proposed model shows improvements regarding reduced average latency, cache utilization, and network capacity.

3. Intelligent Content Prefix Classification Techniques

In ICN, the format of the content prefix is human readable and can be categorized into different groups according to pre-determined rules [19]. This research proposes a generic framework which enables the classification of user's input keywords to guarantee content discovery and retrieval. Our presented framework is initiated when users express their content requests with the assumption that each request comprises of a set of inputted keywords. These keywords are firstly fed into the pre-processing stage to remove unnecessary components, such as duplicate words, blank spaces, and other undesired characters. The filtered inputs are then transferred to the intelligent classification engine which processes and extracts the critical features to obtain the most valuable keywords. The output of the intelligent classification process is the known content prefix attributes that act as the unique content identity, which, in turn, provides the link to bind the prefix and the content that the user asks for. Finally, this output is handled in the post-processing stage before being dispatched to the network as the ICN content name prefix for the interest packet. The procedure overview is depicted in Figure 1.

The next section will evaluate the performances of the selected AI algorithms to identify an intelligent classification method. For this, the relevant Evolutionary algorithms, swarm intelligence,

and machine learning algorithms will be selected and analyzed to realize the suitable method with high performance in the context of the ICN content prefix.

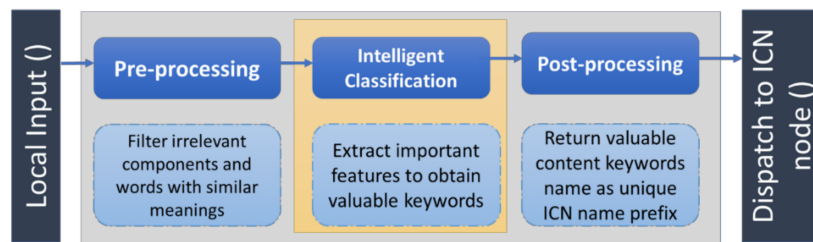


Figure 1. Proposed intelligent classification framework.

3.1. Performance Evaluation of Various AI Algorithms

In this section, we select and compare the performance of relevant AI algorithms in ICN. These algorithms are evaluated using four criteria, which are: the number of function evaluations, the average of the cost function obtained in each trial, the standard deviation of the function values, and the computation time.

The input for this performance evaluation is obtained from a series of keywords that present as independent user input with a data size of 498×8 characters. The input data set firstly enters the pre-processing stage. After, these inputs are fed into the selected algorithm for their performance evaluation. We then elaborate each algorithm’s performance by collecting results from thirty different runs. The computation of the performance analysis is conducted using MATLAB R2016b software (MathWorks Inc., Natick, MA, USA). For this, we designed a migration and mutation strategy, as well as the crossover operator of selected algorithms in the same way to reduce the impact from different operators. Hence, the selected well-known algorithms can be compared in similar conditions. Table 1 shows the key parameter used for the performance evaluations of different algorithms.

Table 1. Parameter settings.

| | Parameters |
|--------------------------|------------|
| Max iteration | 200 |
| NPop | 100 |
| Alpha | 0.99 |
| Initial Temp. | 10 |
| Crossover Percentage | 1 |
| Mutation Percentage | 1 |
| Crossover Inflation Rate | 0.2 |
| Mutation Rate | 0.1 |

3.1.1. Number of Function Evaluations (Nfe)

The Nfe is used to measure the algorithms’ performances and define the optimal model. Nfe represents the number of trials required for the objective function to reach its optimum global value. The efficiency is determined by collecting the Nfe for each selected algorithm in which a lower value of Nfe means higher efficiency. Hence, the most efficient algorithm is the one that consumes the fewest Nfe to solve the problem. For all the selected algorithms, the numbers of function evaluations were calculated by applying the Teaching-learning-based optimization (TLBO), which is a meta-heuristic optimization algorithm based on the natural phenomenon of teaching and learning [20]. TLBO optimization algorithm requires only common controlling parameters like population size and a number of generations for its operation. This means TLBO does not require the determination of any algorithm-specific controlling parameters, such as the mutation ratio and crossover ratio, as in GA. For the TLBO algorithm, the number of function evaluation (Nfe) is calculated by Equation (1):

$$Nfe = 2 \times G_n \times P_n + P_n \tag{1}$$

where G_n is the number of generations in which the best solution was obtained, and P_n is the number of populations [5]. Figure 2 presents the average Nfe required to reach optimum global value for each algorithm. The results taken from 30 different trials (each with 200 iterations) suggest that GA has the lowest required Nfe among evolutionary algorithms, and PSO has the lowest Nfe among the swarm intelligence group. Overall, reinforcement learning shows the lowest Nfe, i.e., it owns the highest efficiency over the other selected algorithms.

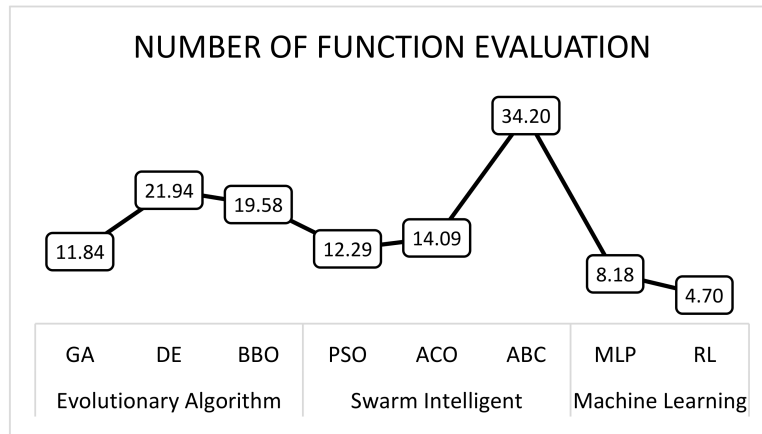


Figure 2. Average number of function evaluation.

3.1.2. Cost Function (CF)

CF is a measurement of the cost utilization which manages the resource needed to satisfy the objective function. CF values of the selected algorithms are then examined to observe each algorithm’s cost efficiency. CF can be calculated using the analytic hierarchy process (AHP), as identified by previous work [5], based on the ability to vary the weighting factors and the optimized cost is selected among different parameter preferences. AHP can also adopt different units of various parameters for QoS into normalized cost value from the cost function.

Consider a set of candidate Algorithm $A_N = \{A_1, A_2, \dots, A_n\}$ and a set of quality of service factors $q_m = \{q_1, \dots, q_m\}$, where n is the number of candidate algorithms and m is the total number of QoS factors. Supposed that each QoS factor q_j has weight value W_j , and this weight shows the effect of the factor on the CF algorithms as following:

$$A_N = \sum_{j=1}^M q_j \times W_j \tag{2}$$

The relative scores among the QoS score set can be calculated using Equation (3), where $R_{q_i q_j}$ is the relative score between parameters q_i and q_j , and S_{q_i} and S_{q_j} are their respective scores.

$$\begin{cases} R_{q_i q_j} = \left(1 - \frac{S_{q_i}}{S_{q_j}}\right) \times 10 & ; j > i \\ R_{q_j q_i} = \frac{1}{R_{q_i q_j}} & ; j < i \\ R_{q_i q_j} = 1 & ; i = j \end{cases} \tag{3}$$

$X = \{X_{ij}\}$ is a $M \times M$ matrix which X_{ij} represents the priority scores of each factor, is initialized as follows:

$$X = \begin{bmatrix} 1 & R_{q1q2} & R_{q1q3} & R_{q1q4} & R_{q1q5} \\ \frac{1}{R_{q1q2}} & 1 & R_{q2q3} & R_{q2q4} & R_{q2q5} \\ \frac{1}{R_{q1q3}} & \frac{1}{R_{q2q3}} & 1 & R_{q3q4} & R_{q3q5} \\ \frac{1}{R_{q1q4}} & \frac{1}{R_{q2q4}} & \frac{1}{R_{q3q4}} & 1 & R_{q4q5} \\ \frac{1}{R_{q1q5}} & \frac{1}{R_{q2q5}} & \frac{1}{R_{q3q5}} & \frac{1}{R_{q4q5}} & 1 \end{bmatrix} \tag{4}$$

Then the normalized relative weight of X_{ij} in Equation (5) is obtained when each element of the matrix X is divided by the summations of its column in Equation (4):

$$X_{ij} = \frac{X_{ij}}{\sum_{i=1}^M X_{ij}} \tag{5}$$

The normalized matrix X , w_{norm} , is shown in Equation (6):

$$w_{norm} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \\ w_{31} & w_{32} & w_{33} & w_{34} & w_{35} \\ w_{41} & w_{42} & w_{43} & w_{44} & w_{45} \\ w_{51} & w_{52} & w_{53} & w_{54} & w_{55} \end{bmatrix} \tag{6}$$

Next, the average values of each row are calculated to give the priorities for each factor as shown in Equation (7):

$$\bar{w}_i = \frac{w_{i1} + w_{i2} + w_{i3} + w_{i4} + w_{i5}}{5} \tag{7}$$

From this, we build the normalized vector W_j which is also the priority vector as it shows the relative weights among its elements. Note that the sum of all the elements in priority vector is 1:

$$W_j = \begin{bmatrix} \bar{w}_1 \\ \bar{w}_2 \\ \bar{w}_3 \\ \bar{w}_4 \\ \bar{w}_5 \end{bmatrix} \tag{8}$$

Since W_j it is normalized from this, the set of QoS parameters where the sum of all the elements in priority vector is 1, denoted by the following vector:

$$q_j = [S \quad D \quad E \quad R \quad V] \tag{9}$$

where five parameters including SINR (S), delay (D), energy (E), RSSI (R), and velocity (V) are used as the QoS cost function parameter.

The cost function is a measurement of the cost utilization to manage the resource needed to allocate for serving the user request with specified QoS requirements. In the context of future wireless networks, [21] predicted that the mobile users will mainly generate content traffic, hence, we consider QoS mobility for user mobility as a QoS factor. The SINR factor is selected to adapt the QoS to the network quality condition and to ensure that the PoA gets good signal quality. Additionally, the delay and energy factor is to model the tradeoff between average delay cost and the average power (energy) cost. Thus, the key for optimal scheduling is on the network balancing between delay and power

cost. Next, the RSSI factor can rank the available wireless network based on the network priority assignment from the list of all available network within user’s coverage at a particular period of time, whereas velocity shows the speed and direction of the user.

For CF evaluation, we use MATLAB, which provides sets of functions for measuring the absolute cost to calculate overall cost function obtained in each trial by mapping each QoS parameter to the corresponding element of vector W_n in sequential order. Table 2 shows the best, worst, and average cost of the selected algorithms obtained after 30 iterations. The results indicate the overall cost of the evaluated algorithms, with the lowest cost, indicates the most efficient QoS allocation based on network condition, achieved by the genetic algorithm (GA). The evaluations for machine learning are not conducted in this trial because the characteristic differences in the cost function and computation make it unsuitable for comparison with other algorithms.

Table 2. Value of best, worst, and average cost function over 30 runs.

| | Evolutionary Algorithm (EA) | | | Swarm Intelligent (SI) | | |
|---------|-----------------------------|--------|--------|------------------------|--------|--------|
| | GA | DE | BBO | PSO | ACO | ABC |
| Best | 117.65 | 117.26 | 117.78 | 118.37 | 362.26 | 117.76 |
| Worst | 158.40 | 182.32 | 273.50 | 180.47 | 550.19 | 167.87 |
| Average | 118.81 | 121.38 | 121.09 | 120.41 | 370.67 | 121.41 |

3.1.3. Standard Deviation (SD)

The SD performance suggests the algorithms’ components stability. The result in Figure 3 shows that BBO has the lowest average SD among evolutionary algorithms, and PSO has the lowest average SD among the swarm intelligence group. Overall, reinforcement learning gives smaller SD values than others, indicating a more stable solution quality as a more massive SD quality shows a less stable solution quality.

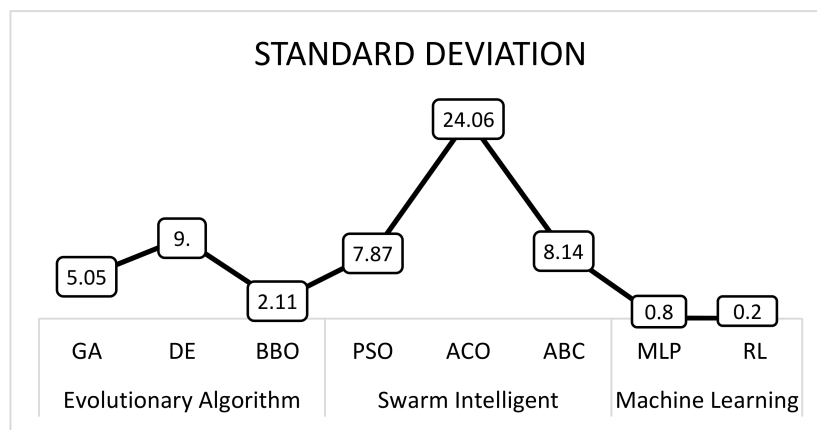


Figure 3. Average standard deviation of different evaluated algorithms.

3.1.4. Computation Time (CT)

The computation time is the total time needed to complete a trial of the algorithm. CT then reflects the CPU computation time, and the smallest number indicates the least amount of time required to finish one operating cycle for each trial. This value is particularly relevant in processing a real-time content request as the response time should be quick to minimize latency.

Figure 4 presents the CT results obtained from each algorithm. The results show that the BBO and ABC algorithms perform better than the rest of Evolutionary algorithm and swarm intelligent groups, respectively. However, as a whole, RL gives smallest CT values compared to other techniques. Overall,

the performances of the selected AI algorithms are summarized in Table 3. The results suggest the high efficiency of reinforcement learning performance over other algorithms in terms of Nfe, SD, and CT. Additionally, results from Table 1 show that GA achieves the lowest cost value.

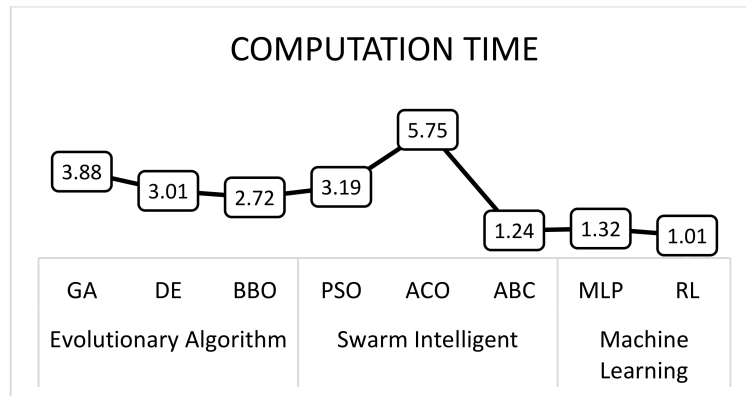


Figure 4. Average standard deviation of different evaluated algorithms.

Table 3. Performance results of the selected AI algorithms.

| | EA | | | SI | | | ML | |
|-----|-------|-------|-------|-------|-------|-------|------|------|
| | GA | DE | BBO | PSO | ACO | ABC | MLP | RL |
| Nfe | 11.84 | 21.94 | 19.58 | 12.29 | 14.09 | 34.20 | 8.18 | 4.70 |
| SD | 5.05 | 9.70 | 2.11 | 7.87 | 24.06 | 8.14 | 0.08 | 0.02 |
| CT | 3.88 | 3.01 | 2.72 | 3.19 | 5.75 | 1.24 | 1.32 | 1.01 |

3.2. Proposed Intelligent Hybrid Technique

The previous section has evaluated the performance of the selected AI algorithms and strongly suggest the superiority of reinforcement learning (RL) under machine learning (ML), and the genetic algorithm (GA) under the evolutionary algorithm (EA) group, for solving classification problem, especially in the case of ICN content prefix.

The application of EA with ML in AI has been one of the growing research fields with rapid development. Studies in this field which attempt to apply the integration of EA and ML techniques have been proven to be beneficial in both convergence speed and solution quality. One of the well-known adaptations of this approach is the learning classifier system (LCS) [22] which has become a powerful tool in a wide range of applications.

Simulated annealing (SA), originating from the process of cooling metal, includes searching for a final minimum energy structure. After going through several stages, the final structure is achieved where the structure gets the minimum value. Different from SA, the GA, as discussed in the previous Section 2.2, has several concerns related to the premature convergence in its optimization, due to a high reliance on the crossover operation. This may diminish the overall performance by producing a more homogeneous population and searching for the best solution in the mutation stage. Another concern of GA relates to the way to reach the optimal solution after finding a near-optimal solution. To attain the global optimum and resolve the occurrence of local optima, we propose a hybrid GA with the SA technique to realize an intelligent content prefix classification in ICN. We show the pseudo-code in Table 4.

The reasons behind the choice of the GA and SA hybrid are because they are proved to be efficient and robust in search processes, making them suitable for solving large combinatorial optimization problems. Unlike the other algorithms, GA has a strong global search ability, while SA has strong local search ability and no premature problems. Therefore, the hybridization of GA and SA can overcome

the limits of each of the two methods, bringing into play their respective advantages, and improve the solving efficiency. The combination of SA rules serves as a validation algorithm for the outcome of the GA and to detect unacceptable calculation results.

Figure 5 summarizes our proposal of intelligent adaptive classification technique which integrates previously-evaluated AI algorithms and the LCS adaptation. There are two main components of the classification techniques which are learning component and discovery component. The learning component features RL, observes the environment, then selects and performs actions [23]. If the action is favorable, it obtains rewards in return. Otherwise, it receives penalties in the form of negative rewards. The discovery component features genetic algorithm–simulated annealing (GA-SA), processes the population evolution by introducing a fitness function, which is proportional to the precise prediction of the reward.

We then assess the performance of the discovery component using GA and compare with our hybrid GA-SA approach. The performance result of the proposed hybrid technique is depicted in Figure 6. Based on this result, we conclude that SA rules shape the integration into the new population phase of GA to increase discovery component effectiveness by merging populations. Hence, we choose to implement GA-SA hybridization as the discovery component in the proposed intelligent classification technique for keyword classification.

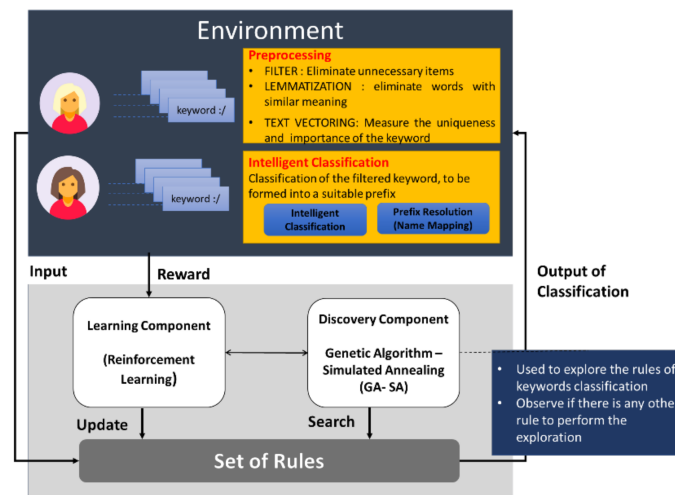


Figure 5. Proposal for intelligent classification using RL with a hybrid GA-SA.

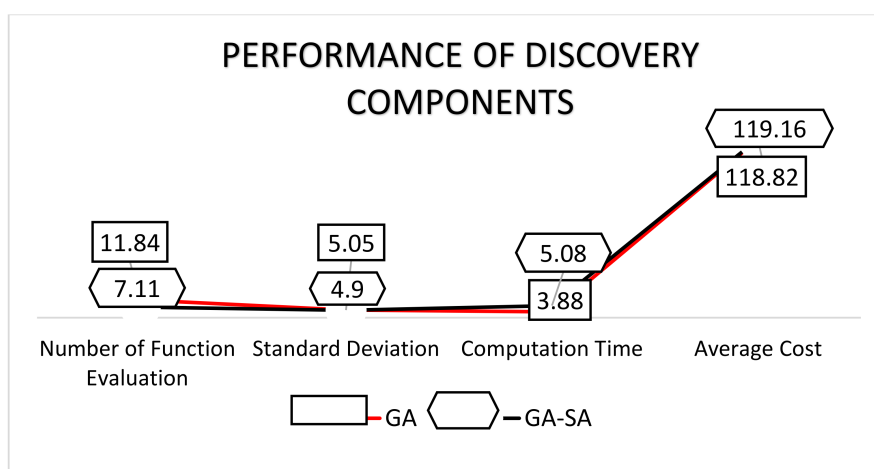


Figure 6. Comparison between GA and hybrid GA-SA for discovery components.

Table 4. GA-SA Discovery Components Pseudo-code.

```

1: Function GA-SA Discovery Component ()
2:   Initialize GA Population of Chromosome
3:   Initialize SA Temperature
4:   For each iteration
5:     Set mutation and crossover
6:     Evaluate the cost function
7:     Calculate priority vector by AHP
8:   End For
9:   If termination criteria are not achieved, then
10:    Select a pair of Chromosome for Mating
11:    For it = 1: MaxIteration
12:      // Perform Crossover
13:      For SubIteration = 1: MaxSubIteration
14:        Perform Crossover
15:        Evaluate Offspring
16:      End For
17:      // Perform Mutation
18:      For SubIteration = 1: MaxSubIteration
19:        Perform Mutation
20:        Evaluate Mutants
21:      End For
22:    End For
23:  End If
24:  Merge Offspring's in the Population
25:  Sort New Population
26:  Compare New Population using SA Rule
27:  Update New Population
28:  Temperature Reduction
29:  Update Best
30:  Solution Found Store Best Cost

```

3.3. Performance Evaluation and Discussion

After the content prefix classification has successfully performed and shaped into valuable content attributes as content name, the name prefix is ready to be dispatched into ICN. Notably, upon receiving a request packet, ICN intermediate router first checks its content store and then pending interest table, as stated in typical ICN forwarding [1].

To evaluate the benefits of the classification methods, we simulate our proposal using ndnSIM [24], a widely-used emulator of name data networking for ICN platform under the ns-3 framework. The network topology used in this simulation is five-layer tree topology as depicted in Figure 7. We assume that the root node at first layer acts as the data center located and it is connected to three core content router nodes in the second layer. At the third layer, these core nodes connect to five edge nodes, which further connect to five aggregated nodes in the fourth layer. Users in the same area connect to their respective aggregated node and send requests (interest packets) for interested content in ICN, given that all router nodes are implemented with the full function as an ICN node (NDN protocol).

Using this topology, we simulate two scenarios of interest-data communication. For the first simulation, we implement content request using an unclassified prefix, whereas, for the second simulation, we implement the already classified prefix using our intelligent framework. In both scenarios, each node has three NDN fundamental data structures (CS, FIB, and PIT) with the same size of CS for cache storage. For simplicity, we also assume that all content objects have the same size. For each scenario, the simulations were executed using two different interest arrival rates. In the first part, user nodes generate a stable, uniform distribution rate of 10 Interest packets per second, whereas, in the

second simulation, we use Zipf-Mandelbrot for modeling content popularity distribution with 10 interest packets per second as the request frequency. The simulation time used for both cases is 100 s.

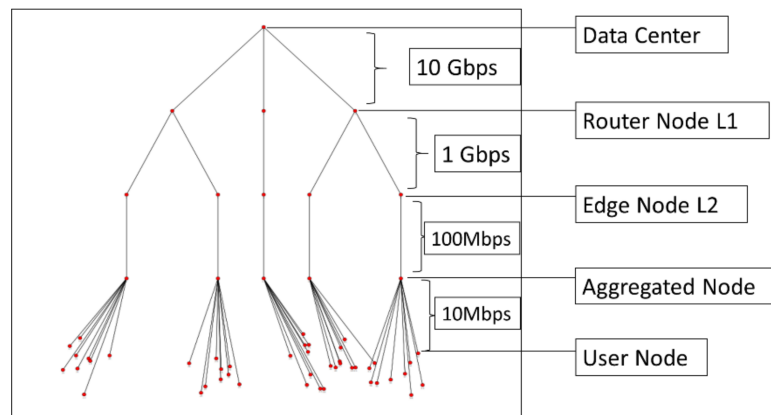


Figure 7. Network topology.

Simulation under a uniform distribution of the Interest packet arrival rate shows that the classified content prefix achieves lower packet drop as well as lower network resource utilization, as shown in the Figures 8 and 9, respectively. The similar tendency is also observed in Figures 10 and 11 when Zipf-Mandelbrot modeled the interest frequency distribution rate. The results show that the proposal achieves lower packet drop rate as well as lower network resource utilization for interest packets by using the classified content prefix format, especially when the number of user increase. This tendency suggests that the proposal is highly scalable and fit into the goal of future interest design as the proposed intelligent prefix classification system can achieve higher benefit for network performance when the network gets bigger with lots of users' content.

Overall, the evaluation results show that the classification method can improve QoS performance in terms of reducing network load and packet drop efficiently. This is because the classified keywords take part in assisting the discovery of content with relatively low overhead for handling the content name prefix. This improvement suggests that the proposed intelligent framework can identify and filter the essential input bits among a large number of irrelevant input keywords from users for specific content.

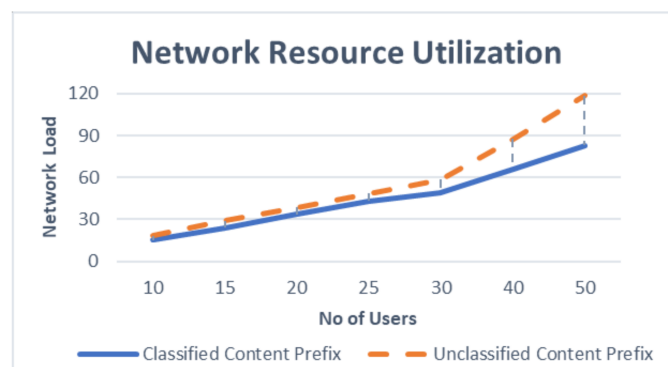


Figure 8. Performance comparison of network resource utilization between classified and unclassified content prefixes under uniform distributions.

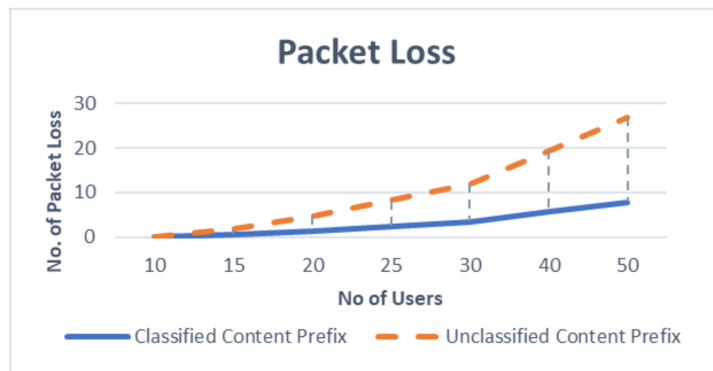


Figure 9. Performance comparison of packet loss between classified and unclassified content prefixes under uniform distributions.

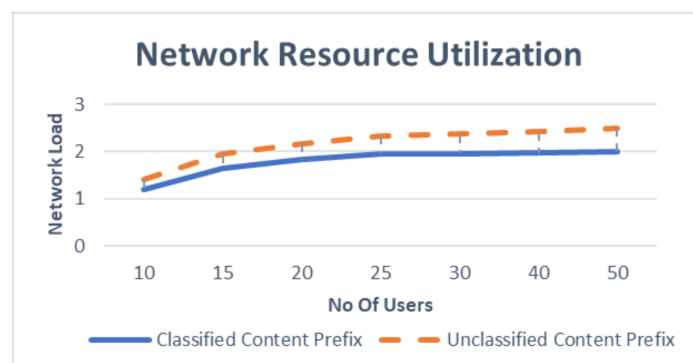


Figure 10. Performance comparison of network resource utilization between classified and unclassified content prefixes.

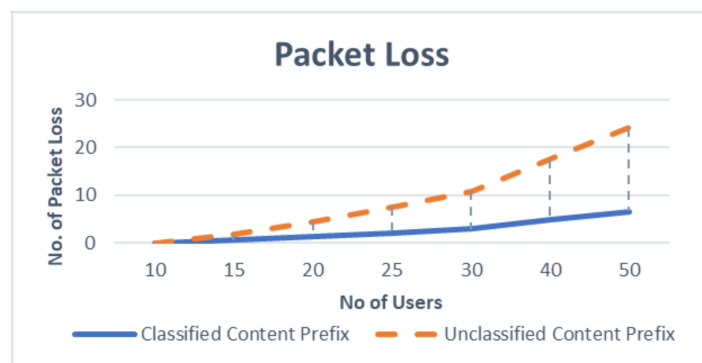


Figure 11. Performance comparison of packet loss utilization between classified and unclassified content prefixes.

4. Conclusions

ICN undoubtedly will play a vital role in communication paradigm shift in the near future, where the number of content items is expected to grow exponentially. In this study, a novel AI-based hybrid classification model is proposed to realize an intelligent classification technique. For this purpose, we integrate the simulated annealing (SA) to enhance the genetic algorithm (GA) as a GA-SA. The GA-SA acts as the hybrid discovery model component of ICN content prefix classification technique to reduce possible occurrences of local optima and premature convergence. Additionally, we implement the GA-SA with the RL-based scheme for learning component to improve the classification performance.

The critical point of this study is motivated by studies on the examination of performance evaluations between the relevant, AI-based algorithms in the context of ICN. This ensures the feasibility of the proposed hybrid technique in ICN content prefix classification. The evaluation results show that the proposed method using hybrid GA-SA achieves a lower score of the number of function evaluations and demonstrates higher performance than GA alone, i.e., reaching the optimization state faster with higher efficiency. This shows that our proposed hybrid classification model realizes an intelligent solution so that it can shape the user's inputted keywords as a content prefix and improves the overall system performance, especially for boosting the QoS performance. Additionally, the network load and packet drop matrices indicate that the proposed classified prefix achieves lower values compared to that of the raw/default content prefix as in conventional ICN.

Acknowledgments: The authors are grateful to the Japan-Asean Integration Fund (JAIF) Scholarship for their financial support and to the Communication Systems and Networks (CSN) Research Laboratory of Malaysia-Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia (UTM). We also extend our sincere appreciation to the Sato Laboratory of Waseda University for or their joint-supervision program.

Author Contributions: All authors conceived, formulate and designed the experiments; C.S. performed the experiments; C.S., S.G., and Q.N.N. analyzed the data; Y.Y., S.B., K.Y., T.S. supervised, advice and guide the overall project; C.S., S.G., Q.N.N. and K.Y. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plass, M.F.; Briggs, N.H.; Braynard, R.L. Networking named content. In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, Rome, Italy, 1–4 December 2009; pp. 1–12.
2. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021; White Paper; Cisco Systems, Inc.: San Jose, CA, USA, 2017.
3. Seo, Y.W.; Zhang, B.T. A reinforcement learning agent for personalized information filtering. In Proceedings of the 5th International Conference on Intelligent User Interfaces, New Orleans, LA, USA, 9–12 January 2000; pp. 248–251.
4. Edens, G.; Scott, G. A Better Way to Organize the Internet: Content-Centric Networking. *IEEE Spectrum Blogs*, 23 March 2017.
5. Goudarzi, S.; Hassan, W.H.; Anisi, M.H.; Soleymani, S.A. Comparison between hybridized algorithm of GA-SA and ABC, GA, DE and PSO for vertical-handover in heterogeneous wireless networks. *Sādhanā* **2016**, *41*, 727–753.
6. Hordri, N.F.; Yuhaniz, S.S.; Nasien, D. A Comparison Study of Biogeography based Optimization for Optimization Problems. *Int. J. Adv. Soft Comput. Appl.* **2013**, *5*, 1–16.
7. Karaboğa, D.; Ökdem, S. A simple and global optimization algorithm for engineering problems: Differential evolution algorithm. *Turk. J. Electr. Eng. Comput. Sci.* **2004**, *12*, 53–60.
8. Dressler, F.; Akan, O.B. A survey on bio-inspired networking. *Comput. Netw.* **2010**, *54*, 881–900. [[CrossRef](#)]
9. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [[CrossRef](#)]
10. Karami, A.; Guerrero-Zapata, M. A fuzzy anomaly detection system based on hybrid pso-kmeans algorithm in content-centric networks. *Neurocomputing* **2015**, *149*, 1253–1269. [[CrossRef](#)]
11. Marsland, S. *Machine Learning: An Algorithmic Perspective*; CRC Press: Boca Raton, FL, USA, 2015.
12. Shanbhag, S.; Schwan, N.; Rimac, I.; Varvello, M. SoCCeR: Services over content-centric routing. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, Toronto, ON, Canada, 19 August 2011; pp. 62–67.
13. Eymann, J.; Timm-Giel, A. Multipath transmission in content centric networking using a probabilistic ant-routing mechanism. In Proceedings of the International Conference on Mobile Networks and Management, Cork, Ireland, 23–25 September 2013; Springer: Cham, Switzerland, 2013; pp. 45–56.

14. Wang, N.; Wu, M.; Peng, L.; Liu, H. ACO-Based Cache Locating Strategy for Content-Centric Networking. In Proceedings of the IEEE 82nd Vehicular Technology Conference (VTC Fall), Boston, MA, USA, 6–9 September 2015; pp. 1–5.
15. Hou, R.; Chang, Y.; Yang, L. Multi-constrained QoS routing based on PSO for named data networking. *IET Commun.* **2017**, *11*, 1251–1255. [[CrossRef](#)]
16. Chiochetti, R.; Perino, D.; Carofiglio, G.; Rossi, D.; Rossini, G. INFORM: A dynamic interest forwarding mechanism for information centric networking. In Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking, Hong Kong, China, 12 August 2013; pp. 9–14.
17. Caarls, W.; Hargreaves, E.; Menasché, D.S. Q-caching: An integrated reinforcement-learning approach for caching and routing in information-centric networks. *arXiv*, 2015.
18. Bastos, I.V.; Moraes, I.M. A forwarding strategy based on reinforcement learning for Content-Centric Networking. In Proceedings of the 7th International Conference on the Network of the Future (NOF), Buzios, Brazil, 16–18 November 2016; pp. 1–5.
19. Zhang, H.; Xie, R.; Zhu, S.; Huang, T.; Liu, Y. DENA: An intelligent content discovery system used in named data networking. *IEEE Access* **2016**, *4*, 9093–9107. [[CrossRef](#)]
20. Dede, T.; Ayvaz, Y. Combined size and shape optimization of structures with a new meta-heuristic algorithm. *Appl. Soft Comput.* **2015**, *28*, 250–258. [[CrossRef](#)]
21. Shang, W.; Bannis, A.; Liang, T.; Wang, Z.; Yu, Y.; Afanasyev, A.; Thompson, J.; Burke, J.; Zhang, B. Named data networking of things. In Proceedings of the 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI), Berlin, Germany, 4–8 April 2016; pp. 117–128.
22. Urbanowicz, R.J.; Moore, J.H. Learning classifier systems: A complete introduction, review, and roadmap. *J. Artif. Evolut. Appl.* **2009**, *1*. [[CrossRef](#)]
23. Géron, A. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media, Inc.: Newton, MA, USA, 2017.
24. Afanasyev, A.; Moiseenko, I.; Zhang, L. *ndnSIM: NDN Simulator for NS-3*; Technical Reports; University of California: Los Angeles, CA, USA, 2012; Volume 4.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).