HAL
open science

Q-ICAN: A Q-learning based Cache Pollution Attack Mitigation Approach for Named Data Networking

Abdelhak Hidouri, Haifa Touati, Mohamed Hadded, Nasreddine Hajlaoui, Paul Mühlethaler, Samia Bouzefrane

To cite this version:

Abdelhak Hidouri, Haifa Touati, Mohamed Hadded, Nasreddine Hajlaoui, Paul Mühlethaler, et al.. Q-ICAN: A Q-learning based Cache Pollution Attack Mitigation Approach for Named Data Networking. Computer Networks, 2023, 235, pp.109998. 10.1016/j.comnet.2023.109998 . hal-04425117

HAL Id: hal-04425117

https://hal.science/hal-04425117

Submitted on 29 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Q-ICAN: A Q-learning based Cache Pollution Attack Mitigation Approach for Named Data Networking

Abdelhak Hidouri[a,b], Haifa Touati[a,*], Mohamed Hadded[c,d], Nasreddine Hajlaoui[a,e], Paul Muhlethaler[f] and Samia Bouzefrane[g]

[a] Hatem Bettaher IResCoMath Research Lab, University of Gabes, Gabes, Tunisia
[b] National School of Computer Science (ENSI), University of Manouba, Manouba, Tunisia
[c] Institute of Research and Technology (IRT SystemX), Paris, France
[d] Abu Dhabi University, Abu Dhabi, United Arab Emirates
[e] Unit of Scientific Research, Applied College, Qassim University, Unayzah, Saudi Arabia
[f] National Institute for Research in Digital Science and Technology (INRIA), Paris, France
[g] CEDRIC Lab, Conservatoire national des arts et métiers (Cnam), Paris, France

ARTICLE INFO

Keywords:
Named Data Networking
Cache Pollution Attack
Q-Learning

ABSTRACT

The Cache Pollution Attack (CPA) is a recent threat that poses a significant risk to Named Data Networks (NDN). This attack can impact the caching process in various ways, such as causing increased cache misses for legitimate users, delays in data retrieval, and exhaustion of resources in NDN routers. Despite the numerous countermeasures suggested in the literature for CPA, many of them have detrimental effects on the NDN components. In this paper, we introduce Q-ICAN, a novel intelligent technique for detecting and mitigating cache pollution attacks in NDN. More specifically, Q-ICAN uses Q-Learning as an automated CPA prediction mechanism. Each NDN router integrates a reinforcement learning agent that utilizes impactful metrics such as the variation of the Cache Hit Ratio (CHR) and the interest inter-arrival time to learn how to differentiate between malicious and legitimate interests. We conducted several simulations using NDNSim to assess the effectiveness of our solution in terms of Cache Hit Ratio (CHR), Average Retrieval Delay (ARD) and multiple artificial intelligence evaluation metrics such as accuracy, precision, recall, etc. The obtained results confirm that Q-ICAN detects CPA attacks with a 95.09% accuracy rate, achieves a 94% CHR, and reduces ARD by 18%. Additionally, Q-ICAN adheres to the security policy of the NDN architecture and consumes fewer resources from NDN routers compared to existing state-of-the-art solutions.

1. Introduction and Motivation

The primary focus in network delivery has always been on content, rather than on the identification of endpoints and representation of hosts. Additionally, a recent white paper from Cisco (1) highlights that by 2024, multimedia content consumption, such as video streams, is projected to account for up to 96% of Internet data usage. This shift in Internet usage patterns is anticipated to have a significant impact on the existing Internet performance, rendering traditional host-based communication models inadequate to meet the demands of this extensive content distribution. Consequently, various strategies have emerged, with the Content Distribution Network (CDN) (2)(3)(4) being a prominent solution. CDN was specifically designed to distribute data across a network of servers. It ensures that content is stored in proximity to the requester by utilizing nearby servers that hold the most frequently requested data by neighboring users.

Another architecture known as Peer-to-Peer (P2P) was introduced to facilitate faster content delivery. The P2P architecture operates as a network overlay, enabling direct communication among users for sharing content without relying on centralized servers (5) (6) (7). While CDN and P2P systems improve upon the traditional TCP/IP architecture, they do possess certain vulnerabilities, particularly in terms of lacking robust security measures. Several notable attacks include Man-in-the-Middle (MiTM) attacks, replay attacks, spoofing attacks, sniffing attacks, and data hijacking (8) (9).

*Corresponding author
haifa.touati@cristal.rnu.tn, haifa.touati@univgb.tn (H. Touati)
ORCID(s):
1

On the other hand, achieving efficient P2P systems remains a significant challenge for researchers. Additionally, ensuring seamless interconnection among nodes in the CDN architecture poses difficulties for developers and researchers (10). In recent years, the Information-Centric Networking (ICN) paradigm has emerged as a result of extensive research aimed at developing a scalable and efficient approach to content delivery over the Internet. This architectural concept emphasizes a content-centric model, decoupling named content items from the specific hosts on which they are stored. Various ICN architectures have been proposed in the literature, with the Named Data Networking (NDN) project being widely acknowledged as a highly promising ICN architecture capable of addressing the limitations of the current Internet architecture.

NDN represents a significant paradigm shift towards a revolutionary Internet that prioritizes the content itself rather than relying solely on endpoint identification. The NDN architecture excels in identifying the source and destination of content without encountering any difficulties, thanks to its robust content naming scheme (11). In NDN, every piece of content is identified uniquely using a hierarchical naming system, establishing a robust foundation for security, integrity, and scalability.

Moreover, compared to the traditional TCP/IP architecture, NDN achieves lower content retrieval delay thanks to its caching mechanism. In NDN, each router has the capability to store a copy of the requested content. This feature allows for faster response times and helps reduce network congestion. This caching mechanism also improves content availability by allowing consumers to access content even when the content producer is offline or unavailable (12; 13). Another important feature of NDN is its built-in security mechanism. In NDN, content producers employ digital signatures to sign their content objects, ensuring that only legitimate requesters (Consumers) can access the specific data content (14). This security measure guarantees that content retrieval is only possible using shared public keys, while the private key remains with the content provider (15).

Since its proposal in 2010, the NDN architecture has been receiving considerable attention, and the research community is consistently exploring its potential in various network environments, including Wireless Sensor Networks (WSNs) (19), IoT (20; 21), VANET (22) and for several applications such as Web (24; 25), Big Data delivery (23) and edge cloud computing (26). Over the past five years, numerous trial deployments have been initiated to accelerate the adoption of NDN by the industry. Examples of these prototypes were showcased during the NDN Community Meeting in 2023 (27). The deployments mentioned are driven by the intrinsic functionalities of NDN, especially its caching and security capabilities, which underscore the significance of secure communications in NDN. These prototypes include initiatives such as Genomics datasets access based on NDN, as well as the N-DISE project for a petascale data distribution system in NDN catering to significant science programs.

Numerous efforts have been made to improve NDN's fundamental functionalities, such as naming, forwarding, and interest rate control (28; 29). However, it is worth noting that the security aspect of NDN is still in its early stages, as the architecture is designed with security in mind. While NDN has shown promise in enhancing content distribution, it has also given rise to several security concerns. In recent times, multiple attacks have specifically targeted this architecture, highlighting the need for robust security measures. Indeed, one of the significant attacks that has impacted NDN is the Cache Pollution Attack (CPA) (16) (17) (18), which can be easily developed and manipulated, and poses a serious threat to the main components of NDN, particularly the Content Store (CS) cache, which can result in an increased number of retransmissions, which ultimately leads to a reduction in network throughput.

To address the challenges of security in NDN, we propose a novel Q-Learning mitigation mechanism called Q-ICAN (Q-learning based Intrusion prevention system for CPA Attack in NDN). Q-ICAN represents a significant improvement over our previous solution, ICAN (30), which was based on statistical measures. While ICAN was effective in identifying some types of cache pollution attacks, it was not a real-time or dynamic solution. In contrast, Q-ICAN, being an RL-based approach, is highly adaptable and able to learn in real-time, making it a more dynamic and robust solution for detecting and preventing cache pollution attacks in NDN. The Q-Learning algorithm enables Q-ICAN to continuously update its comprehension of the network environment and adapt to changing conditions. This enables Q-ICAN to achieve optimal performance and security by making informed decisions based on the knowledge acquired through Q-Learning. The main novelties and contributions of our paper can be summarized as follows:

- As far as our knowledge extends, we are the first to design a scheme to detect and mitigate CPA attacks using the Q-Learning algorithm. Our online and light-weighted Q-Learning strategy allows us to collect the data in real-time and train the model without burdening the resources of the NDN routers.
- Our AI model introduces a unique approach in which the *Action* is based on handling interests rather than potentially malicious data packets. From the initialization phase, our approach involves discarding interests

that our trained model identifies as malicious, while legitimate data packets are stored based on a pre-designed caching strategy. This approach provides a higher level of security for NDN by preventing potential attacks at an early stage.

- To model the *State* of our Q-Learning agent, we have carefully selected four parameters that have not been used in previous literature. These parameters include the Cache Hit Ratio (CHR), Average Interest Inter-arrival Time (IAT), Hop Count Variation, and Prefix Variation. By incorporating these specific parameters, our model can more accurately identify potential cache pollution attacks and improve the overall security of NDN.
- We conducted extensive simulations on two of the largest and most realistic topologies, considering the number of nodes and communication between them. Additionally, our model distinguishes itself as one of the pioneering approaches to address both the Locality-Disruption Attack (LDA) and False Locality Attack (FLA) scenarios.
- Our model has demonstrated exceptional performance in detecting the presence of CPA attacks, achieving an impressive accuracy rate of 95.09%. Additionally, our model can be seen as a caching replacement strategy that enhances the effectiveness of the CS caching mechanism.

The rest of this paper is organized as follows. Section 2 presents an introduction to the basic NDN forwarding mechanism. Section 3 focuses on the essential security aspects of NDN. In Section 4, we provide a brief overview of the related work relevant to our paper. In Section 5, we provide a detailed explanation of our proposed CPA detection scheme. The performance evaluation of Q-ICAN and qualitative comparison with ICAN are presented in Section 6. Finally, we conclude by summarizing our findings and discussing future directions in Section 7.

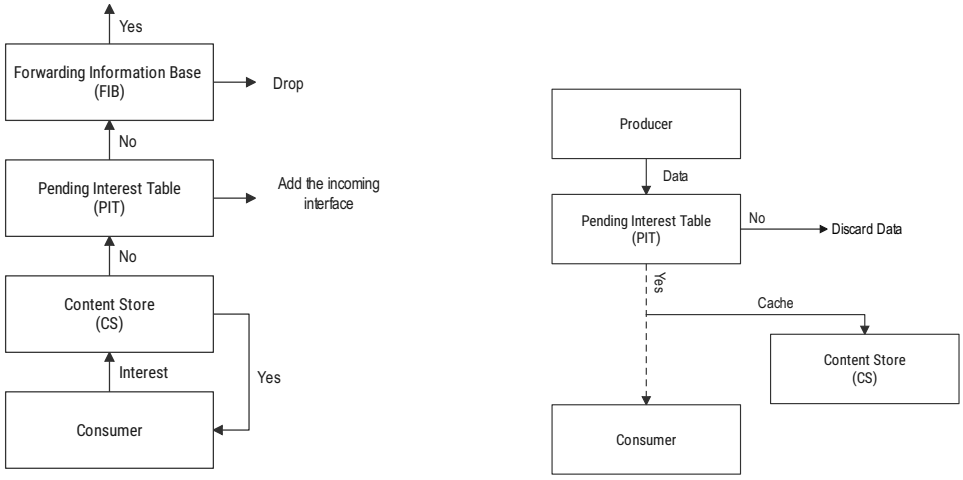
2. Basic NDN Forwarding Mechanism

In NDN, the delivery of content relies on two types of packets: (1) Interest Packets and (2) Data Packets. The Interest Packet, which is responsible for initiating a request for specific content, while the Data Packet carries the requested content in response to a received Interest. The Data packet is typically cryptographically signed, ensuring that only the consumer who initiated the request can verify and manipulate its contents (31). The forwarding process in NDN relies on three main essential components:

- **The Content Store (CS) :** It acts as a temporary cache, storing requested content for future use and enabling faster content delivery by serving it locally instead of retrieving it from the original provider.
- **The Pending Interest Table (PIT) :** It saves the information of each interface in relation to the corresponding interest received.
- **The Forwarding Information Base (FIB) :** It is responsible for storing the content names and their corresponding forwarding strategy and route.

In the NDN architecture, a node performs several checks and actions when it receives an interest packet, as illustrated in Figure 1. Firstly, the node checks the CS, if it finds a match, it returns the requested content to the incoming interface. If there is no match, it performs a PIT lookup. If the same content is already requested in the PIT, the interface from which the interest is received will be appended to the PIT entry. If there is no matching PIT entry, the node creates a new one and then looks up the FIB to find a suitable node that can forward the interest. If a suitable forwarder is found, the interest packet is transmitted via the optimal path according to the forwarding strategy. However, if there is no available forwarder node, the interest packet is discarded.

Next, when a data packet arrives at the NDN node, it first searches the PIT to check if any interest requests were made for the received data. If there is a matching PIT entry, the Data packet is sent back through the recorded reverse path to the interface that made the initial interest request. According to the caching policy, the data packet could be stored in the content store (CS) for quick retrieval in future interest requests. If there is no matching PIT entry, the incoming Data packet is discarded (32).



(a) Interest packet.

(b) Data packet.

Figure 1: Interest and Data packets forwarding in NDN.

3. Security in NDN

In this section, we will provide an overview of the fundamental security mechanisms employed in NDN. We will also identify several threats to NDN network, with a particular focus on the emerging threat of Cache Pollution Attacks (CPA). We will delve into the principle behind CPA and explore the different types of CPA that can occur. By analyzing these threats in detail, we can gain a better understanding of the challenges facing NDN security and design effective countermeasures to mitigate these risks.

3.1. Basic NDN Security Mechanims

NDN incorporates several security measures, including cryptographic verification, access control, and trust management, to guarantee the confidentiality, integrity, and authenticity of data. Data packets are signed by the producer using their private key, and the resulting signature is included in the packet's `Signature` field, which comprises two sub-fields: `SignatureInfo` and `SignatureValue`. `SignatureInfo` identifies the publisher's public key, and `SignatureValue` is decrypted using this key. Upon receiving a Data packet, the consumer verifies its authenticity and integrity by checking the `Signature` field and using the identified public key to authenticate the signature. While the use of signature fields in NDN helps prevent data tampering and ensures trust in received data, intermediate routers may not necessarily need to verify the content due to the substantial overhead it can impose, as verifying the signature requires access to numerous public key certificates to trust the validating public key (33) (34).

3.2. Cache Pollution Attacks in NDN

The NDN architecture is gaining attention as a potential replacement for the traditional Internet architecture that relies on TCP/IP. NDN is designed to focus on content rather than carriers, and researchers have been investigating various aspects of this new architecture, particularly forwarding, caching, and recently security. While NDN is not vulnerable to traditional attacks that target TCP/IP, such as DoS, DDoS, relay, MiTM, or flooding attacks (35), it is still susceptible to new types of attacks, including Cache Pollution Attack (CPA), Cache Poisoning Attack, Interest Flooding Attack (IFA), and Cache Privacy Attack (36). A wider survey of potential attacks in Named Data Networks can be found in (16) (17). The proposed mechanism in the current study specifically targets the Cache Pollution Attack, so in this subsection, we closely investigate the principles and sub-types of CPA.

CPA is one of the most impactful and easy-to-manipulate attacks in NDN. The attacker's goal is to store unpopular items in the CS, thus preventing regular users from accessing the cache. This attack mainly targets cache hits on NDN routers. To manipulate the content's priority in the CS of nearby routers, the attacker node sends a large number of

intensive interest packets. This causes a significant amount of malicious packets to be cached in the router's CS, making it more difficult for legitimate users to access the content they need.

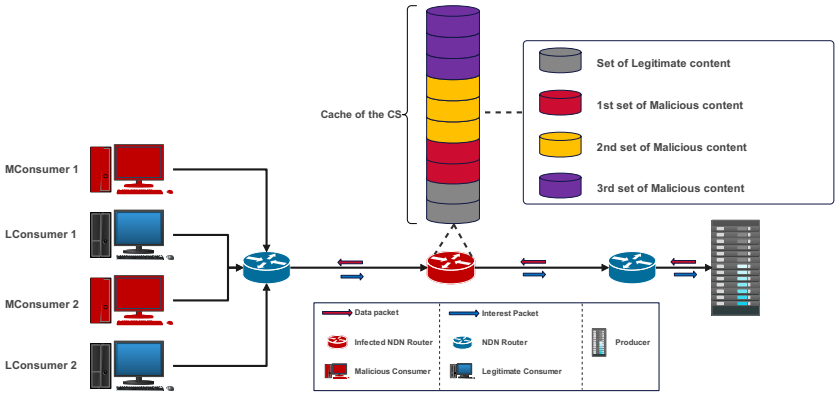


Figure 2: LDA attacks in NDN.

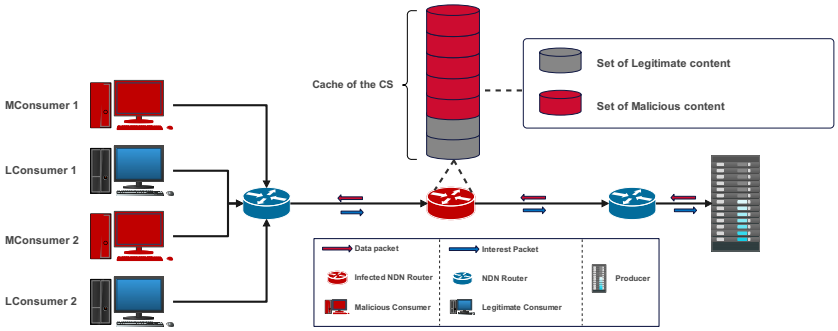


Figure 3: FLA attacks in NDN.

In CPA, there are two main types of attacks: Locality Disruption Attack (LDA) and False Locality Attack (FLA).

In an LDA attack, the attacker generates a continuous stream of requests for new and unpopular contents, intentionally disrupting the correlation structure of the original request stream. These less popular contents are then cached in the NDN gateway and nearby routers. However, since these cached contents are not frequently requested by legitimate consumers, the overall hit ratio experienced by regular consumers decreases. On the other hand, in FLA, the attacker manipulates the popularity of already cached contents by repeatedly requesting the same set of unpopular contents. This strategy artificially inflates the overall hit ratio of the cache, as the attacker keeps requesting the same contents. However, the hit ratio for legitimate consumers is negatively affected since a portion of the cache becomes occupied by the repeatedly requested.

In our previous work (18), we conducted simulations to assess the impact of CPA attacks on the consumer side. We observed that these attacks led to increased wait times for consumers trying to obtain the required content, reduced bandwidth for legitimate consumers, and even Time-Outs. The aforementioned negative effects collectively impact user experience. It is crucial for researchers in the network security community to acknowledge and address the impact of CPA attacks.

4. Related Works

Recently, the literature has seen a significant amount of work proposed to tackle the issue of Cache Pollution Attacks in NDN. These solutions can generally be categorized into statistical and Machine Learning (ML) based approaches.

Our approach falls under the ML-based category. Therefore, in this section, we present a concise review of recent ML-based solutions for mitigating CPA. Further details and a comprehensive review of both categories of CPA mitigation techniques can be found in the referenced work by (16). In Table 1, we provide an overview of the ML-based CPA mitigation solutions discussed in this section.

Nasserala et al. proposed the Cache nFace mechanism in (39). This mechanism employs the CS cache clustering technique to detect the presence of CPA attacks. The mechanism divides the cache of the CS into sub-caches based on the frequency of receiving different contents and assigns them labels corresponding to the associated interface numbers. As a result, if a CPA attack occurs, it affects only one sub-cache, while the other sub-caches continue serving requested contents to legitimate consumers. However, this mechanism has a potential drawback: if the most requested content resides in the infected sub-cache, legitimate consumers may encounter difficulties in accessing it. Additionally, the authors did not address the scenario involving low-popularity contents.

Yang et al. (40) recently proposed a novel CPA mitigation mechanism based on Support Vector Machine (SVM). This approach utilizes various parameters to detect the occurrence of CPA attacks, including the number of interests received per second, the number of content requesters, the distribution of requesters, the core network traffic, and the Cache Hit Ratio of legitimate users. The proposed pre-trained SVM model leverages the aforementioned parameters to classify all received contents and determine their legitimacy or maliciousness. However, this mechanism is subject to a few limitations. Firstly, if the pattern of malicious content differs from what the model was trained on, the attack may go undetected. Secondly, both malicious and legitimate interests are processed, without differentiation, which can lead to unnecessary processing of malicious interests. Lastly, this mechanism may have higher storage requirements for NDN routers due to the inclusion of the SVM model.

Buvanesvari R. et al. (41) proposed a detection mechanism based on a combination of multi-classifier and meta-heuristic approaches. This mechanism comprises two main phases. In the first phase, features such as the number of nodes, frequency of sending interests, pattern of interests, and execution time of the attack are extracted. These features are then used as input to a Deep Convolutional Neural Network (CNN). However, instead of using the traditional CNN functionality, the authors optimize it by incorporating a Fuzzy Decision Tree (FDT). This mechanism exhibits a certain level of greediness in terms of NDN resource consumption, including storage space and CPU usage.

Gini impurity (GI) is a mechanism proposed by Vishwa et al. (42). This mechanism utilizes the decision tree technique and incorporates two main metrics: interest probability and time interval of receiving the interest.

The mitigation technique employed in this mechanism involves adding the suspected malicious interface, from which the malicious interest originates, to a blacklist. However, there are certain drawbacks associated with this approach. One drawback is the heavy resource consumption it imposes on the NDN router, including high CPU usage and potential space storage exhaustion. Additionally, this mechanism may encounter limitations when dealing with malicious content that is influenced by low-popularity content, as it may fall outside the scope of detection.

Lin Yao et al. (43) proposed a mechanism for popularity prediction based on three main metrics: the frequency of interest demand, the standard deviation of request intervals, and the request ratio. These metrics are used to calculate the popularity of content. The calculated values are then incorporated into the Grey Model prediction model (GMP). However, since the mitigation occurs in the later stages of the attack, there is a possibility of the CS cache being polluted even before determining the authenticity of the content. Therefore, this mechanism could be further improved if the decision-making process takes place at an earlier stage.

Gradient Boosting Decision Tree (GBDT) is a mechanism proposed by Dapeng Man et al. in (44). This mechanism involves a learning process that utilizes specific features, including the node status and the information along the path. However, this mechanism has the potential to deplete the limited resources of NDN routers. The authors in (45) proposed a mechanism based on the Radial Basis Function Neural Network (RBFNN). This mechanism utilizes a dataset as input, which includes various features such as the durability of content in the cache over a time interval, the content density, the standard deviation of content, the least recapture of processing content, the hit ratio, and the interface from which the content originates. The output of the model is classified as either 0 (indicating a FLA type of attack), 0.5 (indicating an LDA type of attack), or 1 (indicating a healthy content). However, it is worth noting that this mechanism also faces the challenge of resource greediness within the limited NDN environment.

Hybrid Heterogeneous Multi-classifier Ensemble learning, proposed by Lin Yao et al. (46), is a novel mechanism that utilizes two key parameters: the request frequency for a specific content from the Content Store (CS) cache and the hit ratio of a content in the CS cache. However, it is important to note that this mechanism has a limitation in that it can only effectively identify highly popular content while struggling to detect low-popularity malicious content.

Vimala Rani et al. (47) proposed a defense mechanism designed to counter CPA attacks. The mechanism utilizes Fuzzy C-Means Clustering (FuCL) for detecting these attacks and employs Active Queue Management (AQM) to manage content delivery time. The mechanism makes a decision to either revoke or permit suspected content from the cache. Vimala Rani et al. (48) further investigated the application of the Fuzzy Restricted Boltzmann Machine (FuRBM) learning framework for the detection of CPA attacks. Their proposed mechanism comprises three key steps: the construction of a consumer reward table, the provision of rewards to consumers, and the mitigation of attacks. The system assigns ranks to each content, with low-priority content being revoked from the reward table as a countermeasure. Both the mechanisms presented in (47) and (48) can potentially impose a significant negative impact on NDN resource consumption, resulting in increased bandwidth usage, storage space requirements, CPU overload, and potentially higher false positive rates.

Lin Yao et al. (49) recently introduced a new method to mitigate NDN networks against CPA attack based on Non-Cooperative Game theory (NCG). This technique explores scenarios where the benefits of each agent depend not only on their own actions but also on the actions of other agents. The mechanism involves collecting suspicious requests and storing them in a dedicated suspicious list, a process carried out by all nodes in the network. Each node then determines its utility and caching strategy, utilizing the Non-Cooperative Game technique to achieve balance and equitable utility distribution among the nodes while eliminating suspected malicious nodes. This solution tends to be resource-intensive but can effectively reduce the delivery time caused by constant verifications.

Recently, Liang Liu et al. (50) proposed a novel detection mechanism for identifying CPA attacks based on Bayesian optimization and CatBoost. This mechanism utilizes features extracted by Bayesian optimization, including the cache miss rate during the attack phase and the number of cache misses within the time interval of the CPA attack. BO-CatBoost is employed to predict the network state. However, this mechanism faces certain challenges. One of them is the lack of persistence due to the limited amount of network behavior data available for accurate attack detection. Additionally, this mechanism focuses solely on the detection phase and does not introduce any mitigation strategies. Babu et al. proposed in (51) a signature-based solution to mitigate abnormal behaviors in the NDN network. This mechanism utilizes supervised artificial intelligence techniques, specifically the Dynamic Forest of Random Subsets (DRF). However, this approach has a few performance limitations. Firstly, it exhibits greediness in terms of space storage requirements. Additionally, the mechanism can increase the retrieval delay for content in the network.

Ref.	Year	Machine Learning technique	Attack Category
Yang Cao et al. (40)	2023	SVM	FLA
Babu et al. (51)	2023	DRF	FLA
Liu et al. (50)	2022	BO-CatBoost	FLA
Lin Yao et al. (49)	2022	NCG	LDA and FLA
Buvanesvari R. et al. (41)	2022	CNN and FDT	LDA and FLA
Lin Yao et al. (46)	2022	HHMCEL	FLA
Dapeng Man et al. (44)	2021	GBDT	LDA and FLA
Vimala Rani et al. (47)	2021	FuCL	LDA and FLA
Vishwa et al. (42)	2020	GI	LDA and FLA
Lin Yao et al. (43)	2020	GMP	LDA and FLA
Buvanesvari et al. (45)	2020	RBFNN	LDA and FLA
Vimala Rani et al. (48)	2020	FuRL	LDA and FLA
Andre Nasserala et al. (39)	2018	Clustering	FLA

Table 1: Machine learning based CPA detection and mitigation mechanisms.

5. The Proposed CPA Detection Scheme

In this section, we present our proposed solution for mitigating CPA in NDNs. Our solution is based on the utilization of Machine Learning (ML) techniques, which harness the capabilities of ML algorithms to understand the traffic patterns within the network. By analyzing these patterns, our solution can detect and flag any abnormal behavior that may indicate the occurrence of a CPA. More specifically, our proposed solution utilizes a Q-Learning-based approach to make informed decisions regarding packet filtering in order to prevent the propagation of CPA. We opted for Q-Learning due to its ability to learn the optimal policy for decision-making based on the observed

system state. In the case of CPA detection, this means that Q-Learning can learn the best strategy for identifying malicious interests based on observed network traffic patterns (52). This ability to learn from experience and adjust the policy accordingly makes Q-Learning an attractive choice for CPA detection. Moreover, Q-Learning is a model-free approach, meaning that it does not require prior knowledge of the underlying system dynamics. This property makes it well-suited for non-stationary environments. This is particularly important in the context of CPA detection since the behavior of the attacker and the patterns of malicious traffic are constantly evolving and changing. Q-Learning can adapt to these changes in real-time without requiring updates to the model or knowledge of the precise attack patterns. Overall, Q-ICAN's Q-Learning-based detection strategy is a powerful tool for detecting cache pollution attacks, and its ability to adapt to changing circumstances makes it a valuable addition to any NDN security system.

The main idea of our proposed solution involves integrating a Q-Learning agent into each NDN router. These agents learn the traffic patterns of interest packets by monitoring key metrics such as the average Cache Hit Ratio (AVG-CHR), average Inter-Arrival Time (AVG-IAT), and Hop Count of received Interests. When a notable deviation is detected in these metrics, it indicates a potential cache pollution attack. Consequently, the corresponding interest packet is identified as malicious and subsequently discarded (refer to Figure 4).

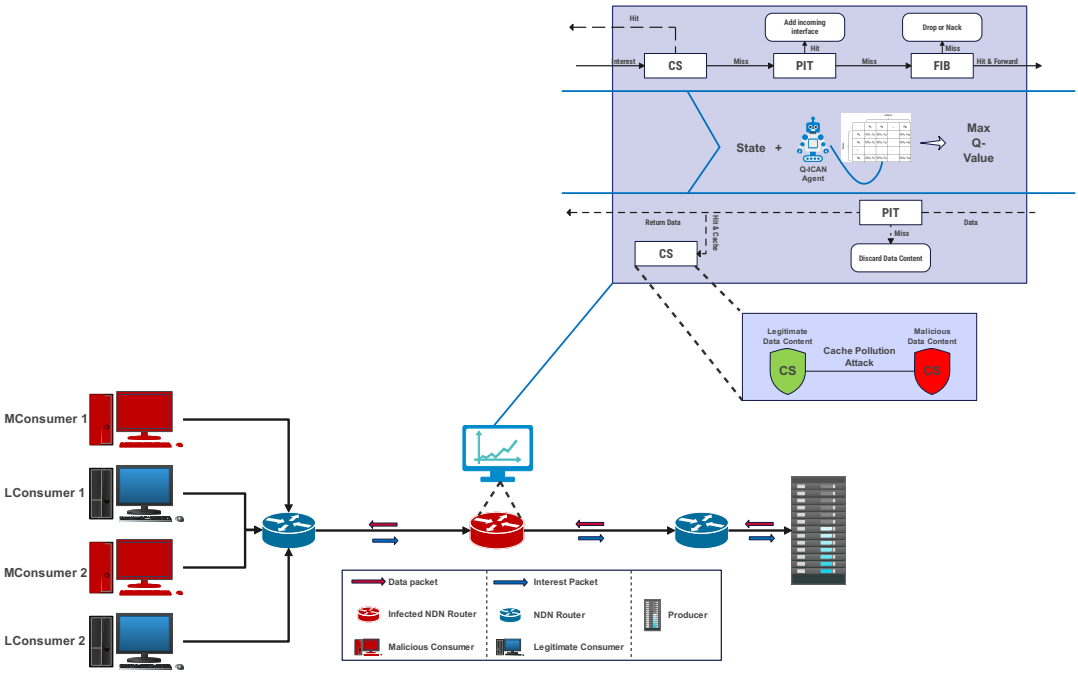


Figure 4: Q-ICAN global system model.

As shown in Figure 5, four key components need to be carefully selected for the implementation of the Q-Learning agent: the state, the action, the reward function, and the transition between the exploration and exploitation phases. The next subsections delve into greater detail regarding the selection process for these components.

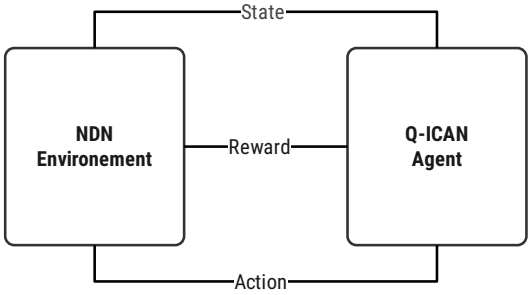


Figure 5: Interaction process between Q-ICAN and the NDN environment.

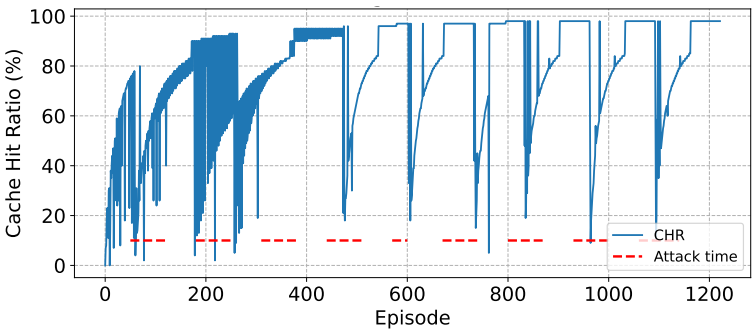


Figure 6: Cache Hit Ratio variation under CPA.

5.1. The state representation

The state in Q-Learning specifies the data that the agent needs to make decisions and update its Q-values. The selection of an appropriate state representation is crucial as it directly impacts the agent's performance in the environment. Mathematically, the state in Q-Learning is represented by equation (1), where $V^{\pi}(s)$ denotes the state value. It represents the estimated total reward starting from state s and following the policy π. When an agent chooses actions based on a particular policy π, the corresponding value function can be expressed as follows:

$$V^{\pi}(s) = \mathbb{E}\,[\sum_{i=1}^{T} \gamma^{i-1}\, r_i]\,; \forall s \in \mathbb{S} \tag{1}$$

The highest possible value function for all states compared to other value functions is referred to as the optimal state value function, represented by the function (2) :

$$V^{*}(s) = \max_{\pi} V^{\pi}(s); \forall s \in \mathbb{S} \tag{2}$$

If we are aware of the optimal value function, then the policy that it relates to is the best policy, denoted as π^{*} as represented by the function (3) :

$$\pi^{*} = arg \max_{a} V^{\pi}(s)\,; \forall s \in \mathbb{S} \tag{3}$$

In order to use Q-Learning effectively for CPA detection, the state parameters should capture the relevant features of the network traffic that enable the NDN router to detect and respond to intrusions effectively. In our previous work (30), we conducted a study to analyze the impact of CPA on NDN routers' performance and studied the evolution of different parameters under the attack. Our findings revealed that the variation of the Cache Hit Ratio (CHR), interest inter-arrival time, and interest hop count could serve as good indicators of the presence of cache pollution attacks. For example, Figure 6 illustrates the variation of CHR under CPA, with the periods of attack highlighted by red dotted lines. It is evident from the figure that the CHR experiences a significant decline and reaches below 5% during the attack periods. Based on the results of this study (30), we choose to represent the state in Q-ICAN by the tuple Average Cache Hit Ratio, Average Inter-Arrival Time, and the Hop Count:

$$S(AVG_CHR, AVG_IAT, HC).$$

Where:

- AVG_CHR represents the average, on each time period, of the Cache Hit Ratio (CHR). The value of CHR is computed as follows:

$$CHR = \left(\frac{\sum CacheHits}{\sum CacheHits + \sum CacheMisses}\right) \times 100$$

  Where `CacheHits` refers to the number of interests that are successfully served from the cache server (CS), and `CacheMisses` refers to the number of interests that are not served from the cache and must be sent to the content producer for retrieval.

- AVG_IAT refers to the average, on each time period, of the IAT. As shown in Figure 7 the IAT represents the inter-arrival time, at the NDN router, of two consecutive interest packets requesting the same prefix.

- In our model, the variation of the Hop Count (HC) refers to the number of nodes, typically NDN routers, that a piece of data traverses. Specifically, in our model, the HC is measured at the targeted NDN router. This metric is evaluated for each prefix and denoted as $HC_{prefix(i)}$.

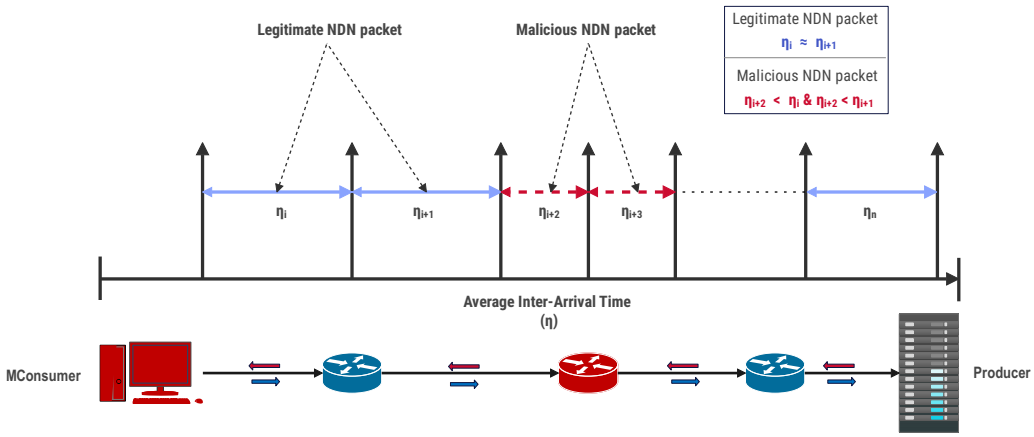


Figure 7: Average Inter-Arrival Time model under CPA attack.

In addition to selecting the best parameters to represent the state, reducing the number of states in Q-Learning is also crucial in order to reduce the computational complexity of the algorithm and improve its ability to generalize to new situations. When there are too many states, the Q-learning algorithm may converge slowly or fail to converge altogether. One effective approach to reducing the number of states in Q-Learning is through a technique called *state aggregation*. By grouping similar states together, the algorithm can learn more efficiently and generalize better to novel scenarios. In the context of Q-ICAN, we employ the following steps to reduce the number of states:

As illustrated in Figure 8, we have implemented a technique to reduce the variation of the Average CHR by grouping its values into ranges of 10% each. For instance, if the CHR value falls between 0% and 10%, we assign the value of 0.1 to the corresponding metric in the current state.

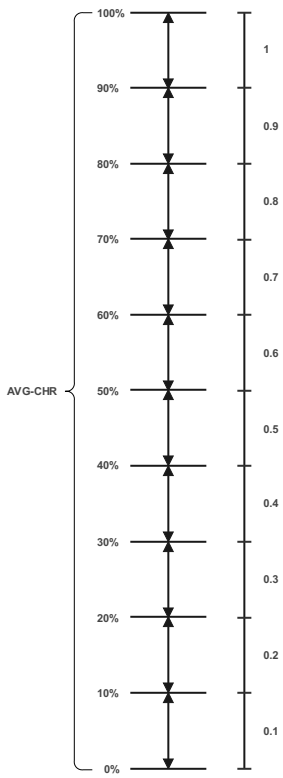


Figure 8: Reducing the variation of the AVG-CHR paramter.

As shown in Figure 9, we have implemented a technique to reduce the number of values for each Inter-Arrival Time. This reduction is achieved by defining four main values, denoted as $V1$, $V2$, $V3$, and $V4$, based on the values of (IAT_{min}, IAT_{moy}, IAT_{max}). For example, if an IAT value falls within the range of $0s$ to IAT_{min}, it will be set to $V1$, and so on.

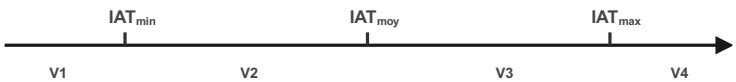


Figure 9: Reducing the variation of the the AVG-IAT parameter.

5.2. The Action Modelling

In Q-Learning, the action value function, denoted as $Q(s, a)$, represents an estimate of the expected reward an agent will receive by taking action a in state s and following the optimal policy thereafter. The optimal policy is the one that maximizes the expected return over time. In a given state, there may be multiple available actions, each leading to a

different estimate of the Q-value. The agent typically selects the action with the highest Q-value as the optimal action to take in the current state. Mathematically, the optimal value function $V^*(s)$, as defined in equation (1), is defined as the maximum value of $Q^*(s, a)$ for all actions a in state s. In other words, $V^*(s)$ represents the highest predicted total reward an agent can achieve when starting from state s and following the optimal policy. The relationship between $Q^*(s, a)$ and $V^*(s)$ can be expressed straightforwardly as follows:

$$V^*(s) = \max_a Q^*(s, a) \; ; \forall s \in \mathbb{S} \tag{1}$$

Thus, by selecting the action a that produces the highest $Q^*(s, a)$ for a given state s, we can determine the optimal policy. This can be expressed as shown in equation (2), assuming that we have knowledge of the maximum action-value function $Q^*(s, a)$.

$$\pi^*(s) = arg \max_a Q^*(s, a) \; ; \forall s \in \mathbb{S} \tag{2}$$

In practice, it is essential to carefully design the action values to ensure that they accurately represent the reward structure of the environment and the agent's goals. In the context of CPA in NDN, we have to choose between two possible directions: (1) Actions applied to data packets, i.e. whether to cache or not the Data packet (2) Actions applied to interest packets, i.e. whether to discard or to process the interest packet.

In Q-ICAN, we chose the action that is most appropriate in terms of conserving the NDN router's CPU usage and storage space. Using the first type of action can overload intermediate routers and waste network bandwidth, compared to the second action, which involves discarding malicious interests at the beginning without forcing the NDN routers to unnecessarily forward and satisfy attackers' interests. Hence, in our solution, we set the action as follows:

$$A = \begin{cases} \rightarrow Process\ Interest \\ \\ \rightarrow Discard\ Interest \end{cases}$$

5.3. The Reward Modelling

The reward specification is crucial in reinforcement learning, since it serves as a signal for the direction of training. The reward signal informs the agent how well it is performing during a specific state and action. Consequently, A well-designed reward function can speed up the agent's convergence to an optimal policy.

To model the reward in Q-ICAN, we follow the following logic. When an action A_i has been taken in step i, the reward needs to be calculated in the step $i + 1$, in this period of time as shown in Figure 10, two situations only can lead to an increase of the AVG_CHR such as :

- If the attack appears and the decision is correct
- If there is no attack and the decision is correct

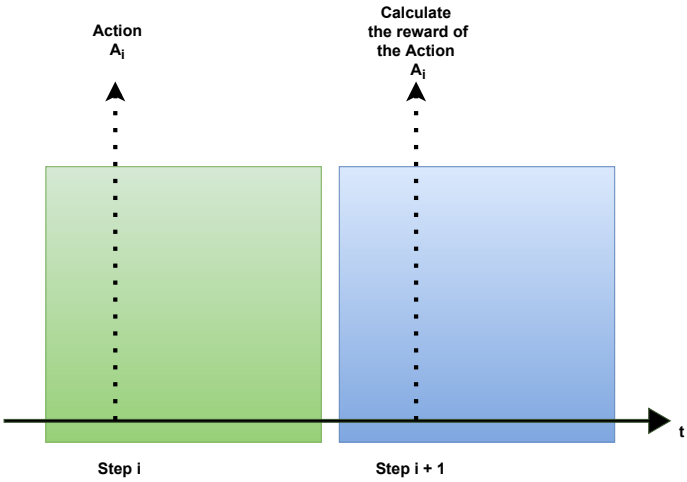


Figure 10: Reward modelling.

This variation of the CHR (AVG_CHR) leads us to model the Reward as follows :

$$Reward = \left\{AVG_CHR_{t+1} - AVG_CHR_t\right\}$$

where;

$$Step(t) = RTT_{moy}$$

5.4. The Q-Function Modelling

In Q-learning, equation (1) is used to calculate the Q-value for an action taken from state s. This value is determined by summing the immediate rewards obtained using a behaviour policy, such as an ϵ-greedy policy. At state s_{t+1}, the policy starts acting greedily. The value iteration in Q-learning, as shown in equation(2), involves updating the $Old\ Q\ value$ with the addition of the $TD\ error$ term to create a $new\ Q\ value$. This new value reflects the difference between the predicted Q-value and the actual Q-value obtained from taking action a at state s.

$$Q(s,a) = r(s,a) + \gamma \max_a Q(s',a) \tag{1}$$

$$Q(s,a) \leftarrow old\ Q\ value + TD\ error \tag{2}$$

The $new\ Q\ value$ is subtracted from the previous Q value to calculate the $TD\ error$ as demonstrated in (3) and (4).

$$Q(s_t,a_t) \leftarrow old\ Q\ value + (new\ Q\ value - old\ Q\ value) \tag{3}$$

$$new\ Q\ value = [r_t + \gamma\ max\ Q(s_{t+1},a_t)] \tag{4}$$

The Q-Function that insures the Q-Value updates is represented as follows:

$$q^{new}(s,a) = (1-\alpha) \times \underbrace{q(s,a)}_{\text{Old Value}} + \alpha \times \overbrace{(R_{t+1} + \gamma \times \max_{a'} q(s',a'))}^{\text{Learned Value}} \tag{5}$$

Where, γ represents the discount factor and α the learning rate.

In Q-ICAN, we have introduced a new structure called the Q-table, which is used to store the Q-values associated with each state-action pair. This Q-table is an essential component of the Q-ICAN algorithm. To illustrate this concept, we have provided an example of the contents of the Q-table in Figure 11. This table shows the Q-values for different state-action pairs and is used to determine the best action to take from each state.

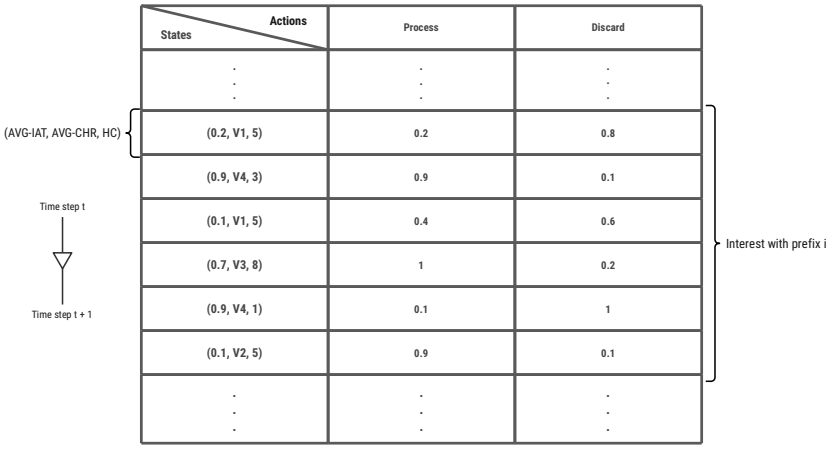
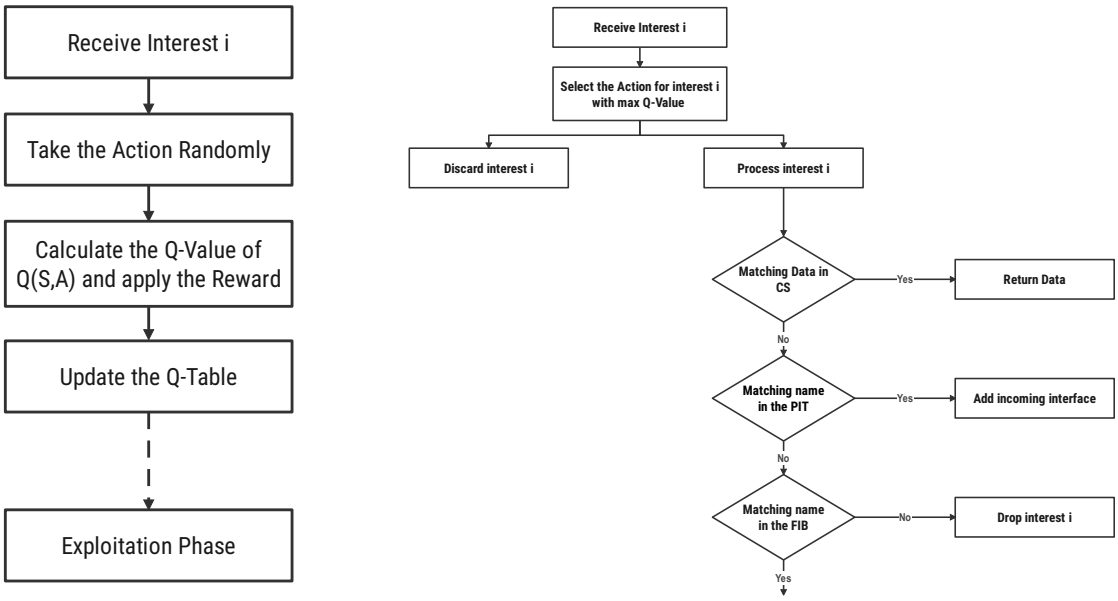


Figure 11: Q-ICAN associated Q-Table.

5.5. The Exploration and Exploitation Phases

In Q-Learning, the exploitation phase refers to the process of selecting the action with the highest Q-value based on the Q-Table. On the other hand, the exploration phase involves randomly selecting actions to explore new possibilities. By striking a balance between exploitation and exploration, the Q-Learning algorithm can effectively learn and improve its decision-making capabilities.

In Q-ICAN, during the exploration phase, the agent attempts to discover new features of the NDN network environment by selecting sub-optimal action (see Figure (a) 12). The Q-ICAN agent starts by receiving an interest i,and takes an action randomly. For each action, the Q-Value of $Q(s,\ a)$ is calculated, representing the expected future rewards associated with taking the action a in state s. Once the Q-Value is obtained, a reward is assigned based on the observed outcome of the action. This reward is used to update the Q-Table. The agent iteratively updates the Q-Table until it reaches the optimal Q-Value for each state-action couple (s, a).



(a) Exploration phase.

(b) Exploitation phase.

Figure 12: Q-ICAN exploration and exploitation phases.

During the exploitation phase, the Q-ICAN agent uses the knowledge that has been learned about the NDN network environment. As shown in Figure (b) 12, upon receiving an interest i, Q-ICAN selects the Action associated with the maximum Q-Value. If the interest is malicious, the associated action basically is to discard the interest. If the interest is legitimate, the associated action is to process the interest and the normal NDN forwarding mechanism is applied.

Achieving optimal performance in Q-Learning depends on finding the right balance between exploration and exploitation phases. In the literature (54), several strategies have been suggested to achieve this trade-off, including:

- Epsilon-greedy strategy.
- Decaying Epsilon-greedy.
- Optimistic initialization
- Upper Confidence Bound
- Thompson Sampling
- etc.

Our choice has been made based on the usability of the Epsilon-greedy strategy in the state of the art. This method introduces a hyperparameter "epsilon" that determines the probability of selecting a random action during the exploration phase. The value of epsilon is usually gradually reduced over time to prioritize exploitation as the Q-Table becomes more accurate. Recent works have shown that Epsilon-greedy strategy is suitable for named data networks (55)(56)(57), as it yields better performance in terms of time consumption and the switching between the exploration and the exploitation phases. The algorithm of the Epsilon-greedy strategy is given in Algorithm 1.

6. Performance Evaluation

In this section, we will first discuss the simulation settings that were used in our performance evaluation. Next, we present and analyse the experimental results we obtained. Finally, we conduct a comparative analysis of Q-ICAN with state-of-the-art solutions including our previous ICAN solution.

Algorithm 1 Epsilon-greedy strategy algorithm

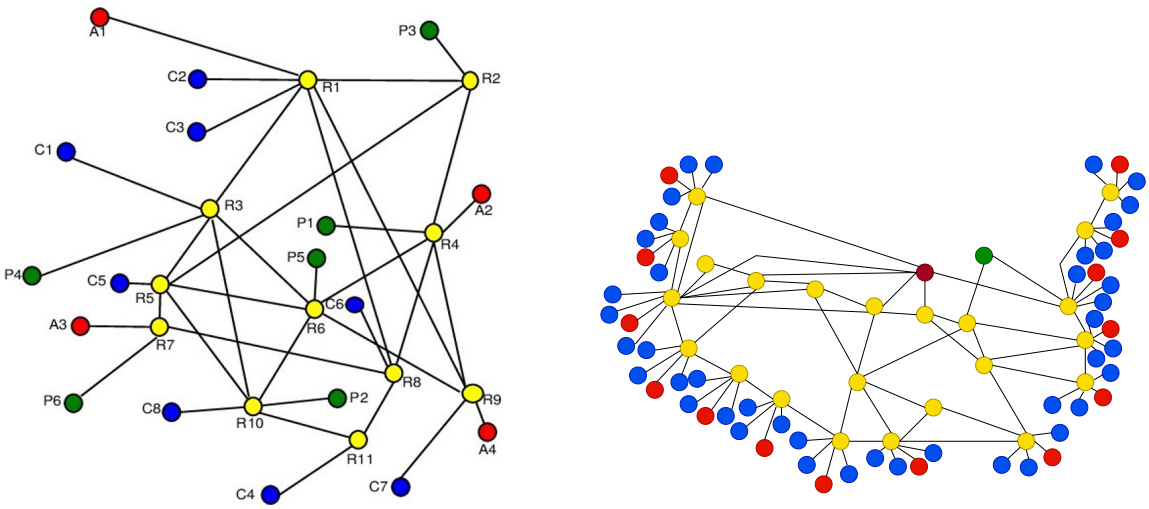
```
Epsilon ← 1
Rand ← [0..1]
if Rand ≤ Epsilon then
    Exploration()
else
    Exploitation()
end if

Update(Epsilon)
```

6.1. Simulation Settings

In this subsection, we provide detailed information about the simulation settings used in our investigation. We conducted our simulations using the official ndnSIM module of Network Simulator 3 (NS-3). We used two real-world topologies: the DFN topology (see Figure 13.a) and the larger AT&T topology, with up to 80 nodes (see Figure 13.b). We explored various settings within each topology to capture a wide range of scenarios. During the total simulation time, we alternated between the attack period and the non-attack period (normal state). The attack period lasted for 7 seconds, while the normal state lasted for 6 seconds. To investigate the effectiveness of Q-ICAN, we varied several parameters, including the number of the attackers, the frequency of malicious interests, the cache size. Additionally, we tested different caching policies, such as LRU and LFU, which are commonly used in NDN. A summary of the simulation parameters is presented in Table 2.

Simulation time	106s
Number of Nodes	DFN:29 / AT&T:80
Number of Legitimate Consumers	DFN:8 / AT&T : 42
Number of Attackers	DFN: 1, 2, 3, 4 /AT&T: 14
Consumer type	ConsumerZipfMandelbrot
Attackers Interest rate	120, 160, 200, 220, 260
Legitimate Consumers Interest rate	120
Time of Launching the Attack	5-12, 18-25, 31-38, 44-51, 57-60, 67-74, 80-87, 93-100s
Router CS size	DFN : 50, 100, 150 / AT&T : 100
Cache policy	DFN : LRU and LFU / AT&T : LRU

Table 2: Simulation parameters.



(a) DFN topology.

(b) AT&T topology.

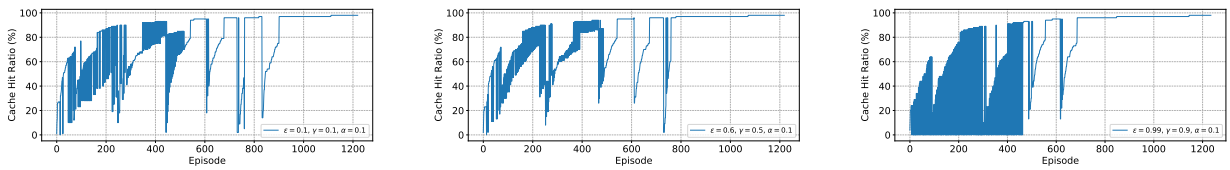
Figure 13: Simulation topologies.

6.1.1. Setting the Hyperparameters for Q-Learning

Setting the hyperparameters in Q-Learning plays a crucial role in determining the performance of the algorithm, and it is particularly important for achieving optimal results with the Q-ICAN agent. There are three key parameters that need to be appropriately configured to achieve the optimal performance of the Q-ICAN agent.

1. Epsilon (ϵ): It is used to balance exploration and exploitation in the Q-ICAN agent. As explained in section 5.5, we consider using a decaying schedule for epsilon. By gradually reducing epsilon over time, the agent starts with a higher exploration rate in the initial stages of learning, allowing it to explore various actions and learn about the environment. As the learning progresses, the agent shifts its focus toward exploitation.
2. The learning rate (α): It determines the speed at which the Q-ICAN agent learns from new experiences. Selecting an appropriate learning rate is crucial for achieving optimal learning and convergence in the Q-Learning process.
3. The Discount factor (γ): It determines the trade-off between immediate and future rewards. A value of 0 indicates that the agent is fully short-sighted and only considers immediate rewards, while a value of 1 indicates that the agent considers all potential future rewards equally.

In order to determine the most suitable hyperparameter settings for our specific problem, we conducted experiments involving various combinations of ϵ, α, and γ values. The performance of the Q-ICAN agent was evaluated using these different hyperparameter settings, and the results are depicted in Figure 14. Figure 14.(a) demonstrates that when the hyperparameters are set to (0.1, 0.1, 0.1) respectively, the achieved Cache Hit Ratio (CHR) stabilizes at episode 910. However, by increasing the values to (0.6, 0.1, 0.5), as shown in Figure 14.(b), the CHR stabilizes faster, reaching stability at episode 780. Furthermore, by further increasing the hyperparameters to (0.99, 0.1, 0.9) as depicted in Figure 14.(c), the stabilization time is reduced to 605 episodes, with a 1% improvement in the CHR compared to Figure 14.(b). Based on the analysis of these results, we have selected the hyperparameters $\epsilon = 0.99$, $\alpha = 0.1$, and $\gamma = 0.9$, as summarized in Table 3.



(a) $\varepsilon = 0.1$, $\alpha = 0.1$, $\gamma = 0.1$ (b) $\varepsilon = 0.6$, $\alpha = 0.1$, $\gamma = 0.5$ (c) $\varepsilon = 0.99$, $\alpha = 0.1$, $\gamma = 0.9$

Figure 14: Performance of CHR under the variation of the Q-learning settings.

Epsilon (ϵ)	Learning Rate (α)	Discount Factor (γ)
0.99	0.1	0.9

Table 3: Q-ICAN agent settings.

6.2. Simulations Results

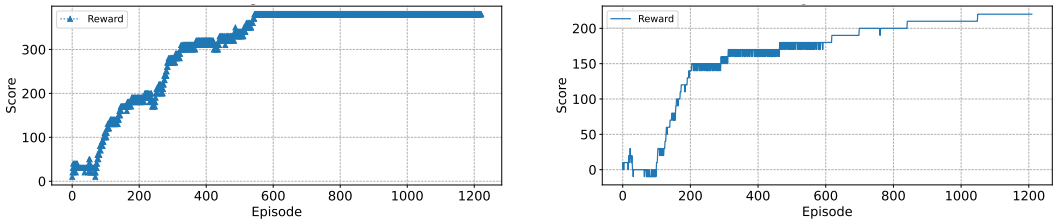
To assess the effectiveness of our proposed method, we studied various performance metrics, namely (1) the score values, (2) the Cache Hit Ratio, (3) the Average Retrieval Delay, and (4) the efficiency in terms of several AI-related performance parameters (i.e. Accuracy, Precision, Recall, Specificity, and F1 Score). In the following subsections, we present and analyse the obtained results.

6.2.1. Evaluation of the Score values

To evaluate the performance of our Q-ICAN agent in the presence of CPA attacks, we have, first, measured the Score values to track when the rewarding values converge and stabilize. We model the Score as the sum of the reward or penalty value R on each episode e :

$$Score = \sum R_e$$

Our evaluation of the Score metric reveals that the score stabilizes after 580 episodes for the DFN topology and after 590 episodes for the AT&T topology (see Figure 15.) These results demonstrate that the Q-ICAN agent performs well in both topologies, as evidenced by the stable convergence of the Score. The agent effectively learns to navigate the environment without making any critical mistakes, highlighting its ability to exploit the environment while avoiding erroneous judgments.



(a) DFN topology. (b) AT&T topology.

Figure 15: Q-ICAN score under DFN and AT&T topology.

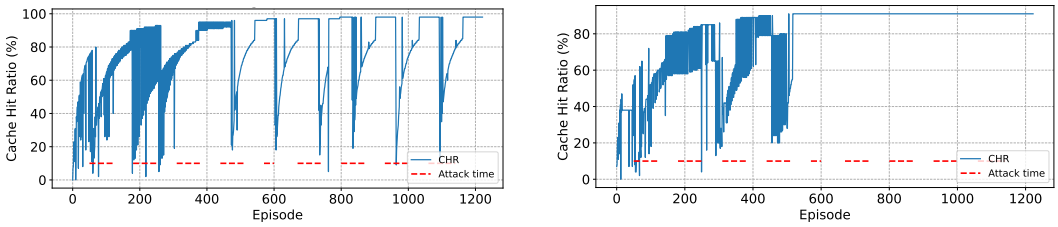
6.2.2. Evaluation of the Cache Hit Ratio (CHR)

In addition to the Score metric, we have evaluated, in a second step, the Cache Hit Ratio (CHR) metric, where the Cache Hit Ratio (CHR) is the percentage of data requests that were successfully supplied by the cache. This indicates that a consumer received the content requested from the cache without having to request it from the producer. Moreover, the decease of the value of the CHR refers to a suspected attack state.

- **The CHR while varying the network size** we first examined the impact of network size on CHR using two real-world topologies: the DFN topology, which consists of 29 nodes, and the AtT topology, which has 80 nodes.

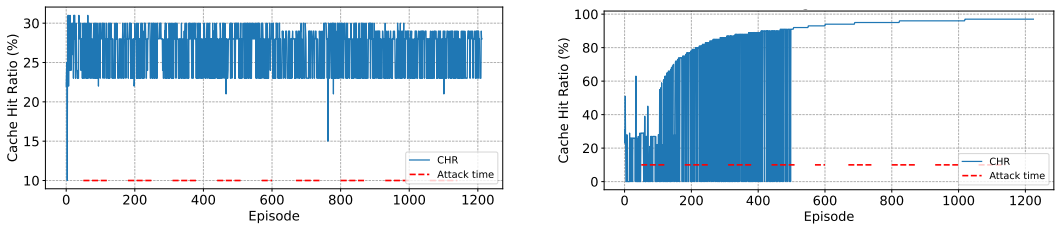
  As shown in Figure 16.a, in the DFN topology, an attack can cause significant damage to the CHR if Q-ICAN is not implemented. In most cases during the attack, the CHR drops to under 20%, as outlined in Table 2, and in some instances, it reaches 0%. However, with the implementation of Q-ICAN, as depicted in Figure 16.b, the CHR stabilizes at 94% during the exploitation phase (Episode = 520).

  Similarly, in the AT&T topology, the CHR decreases to under 23% in most cases during the attack state, as shown in Figure 17.a. However, with the presence of our agent, as depicted in Figure 17.b, the CHR stabilizes at almost 92% by the end of the 500th time episode.



(a) Without Q-ICAN. (b) With Q-ICAN.

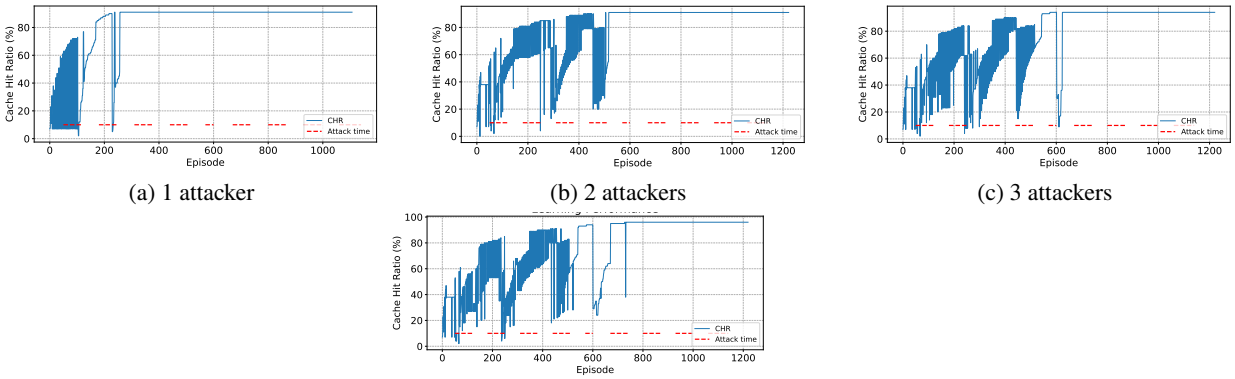
Figure 16: Evaluation of the CHR with and without Q-ICAN in the DFN topology.



(a) Without Q-ICAN. (b) With Q-ICAN.

Figure 17: Evaluation of the CHR with and without Q-ICAN in the AT&T topology.

- **The CHR while varying the number of attackers :** The performance of Q-ICAN is not significantly affected by the number of attackers, as confirmed by the obtained results. In the case of one or two attackers, Q-ICAN can easily detect and mitigate them, as shown in Figure 18.a and Figure 18.b at episodes 263 and 520, respectively. With three or four attackers, as depicted in Figure 18.c and Figure 18.d, the detection episode typically falls between 615 and 790, consecutively. Despite the high stress that Q-ICAN experiences when the number of attackers varies, the CHR remains stable and typically ranges between 91% and 94%.

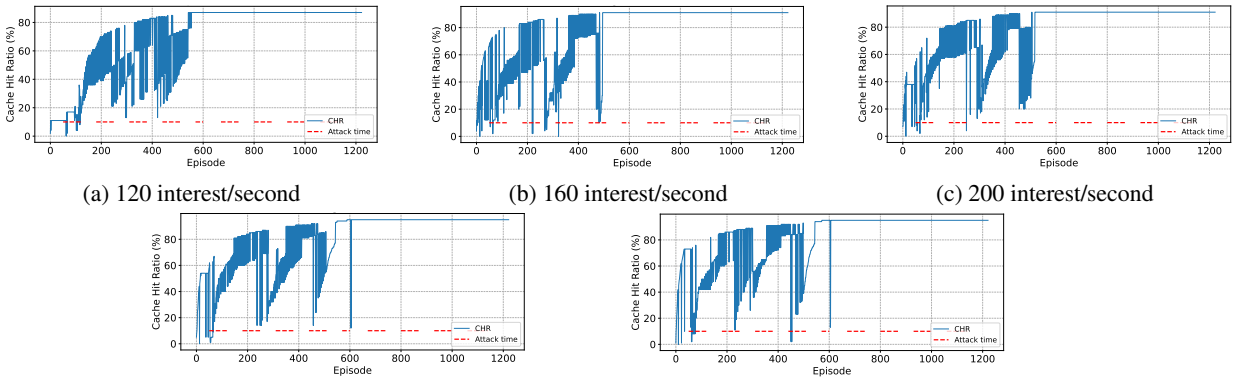


(a) 1 attacker (b) 2 attackers (c) 3 attackers

(d) 4 attackers

Figure 18: Performance of CHR under the variation of the number of the attackers.

- **The CHR while varying the rate of malicious interests :** As the rate of attacker interests increases, distinguishing between legitimate and malicious traffic becomes increasingly challenging for any mitigation solution. Hence, it is crucial for CPA mitigation solutions to be capable of handling high rates of attacker interests while maintaining performance and preventing legitimate traffic from being blocked. This study examines the impact of varying the sending rate of malicious interests on the CHR using Q-ICAN. The study employs five main frequencies, as shown in Figure 19. The results confirm that Q-ICAN's performance remains unaffected by an increase in the malicious interest rate, and it efficiently detects and blocks malicious interests even under stressful scenarios of up to 260 interests per second. This resistance to a malicious traffic increase ensures that the Q-ICAN solution can continue to function optimally even under high-stress scenarios. In all cases, our agent successfully detects this impactful attack, and the CHR exceeds the threshold of 90%.



(a) 120 interest/second (b) 160 interest/second (c) 200 interest/second

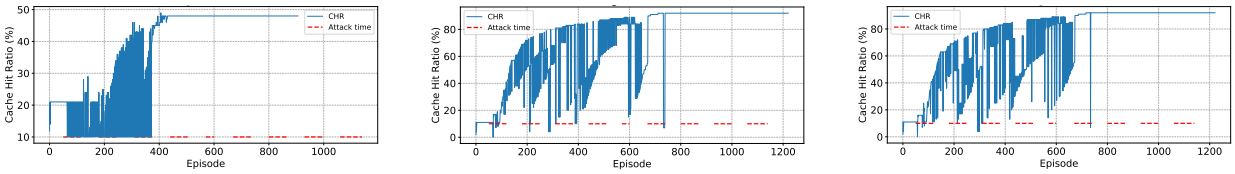
(d) 220 interest/second (e) 260 interest/second

Figure 19: Performance of CHR with a frequency variation of requesting malicious content.

- **The CHR while varying the cache size :** The variation of such parameter is important, where it shows that the attack still damaging the main cache resources and reduces the recover of such Data content. In our simulation,

3 cases are under our study such as the cache size of 50,100 and 150. Despite the high impact of CPA in the cache of the CS, Q-ICAN is able to mitigate and improve the CS cache from CHR = 0% to CHR = 94% as demonstrated in Figure 20.a, 20.b, 20.c consecutively.

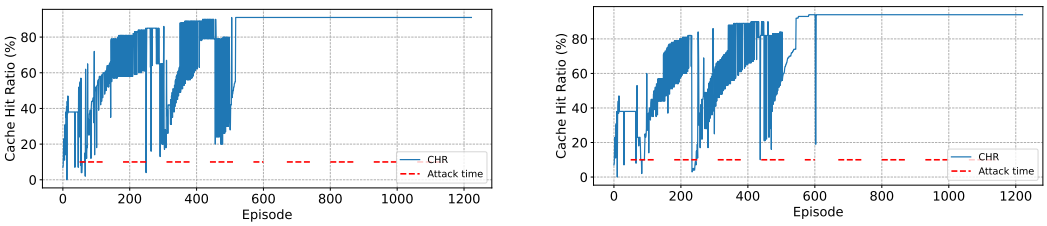
Varying the cache size is a critical parameter as it demonstrates the extent to which an attack can damage the main cache resources and reduce data content recovery. In our simulation, we examine three cases with cache sizes of 50, 100, and 150. Despite the significant impact of CPA on the cache of the CS, Q-ICAN can improve the CHR from 0% to 94%, as illustrated in Figure 20.a, 20.b, and 20.c. This improvement in CHR shows that Q-ICAN can effectively identify and block malicious interests, preventing them from damaging the cache resources of the CS.



(a) CS = 50. (b) CS = 100. (c) CS = 150.

Figure 20: Performance of CHR under the variation of the CS cache size.

- **The CHR using different caching policies :** The caching policy has a huge impact on the caching mechanism, where variating this metric can set Q-ICAN into a challenging position of detecting the attack upon the variation of such important metric that has a direct relation with the CHR. Q-ICAN is able to detect the CPA attack in both LRU and LFU caching strategies. The CHR increased from 0% to above 90% as shown in Figure 21.a and 21.b. The choice of caching policy has a significant impact on the caching mechanism, and varying this setting can pose a challenge for Q-ICAN to detect the attack since it has a direct impact on the CHR. Nevertheless, our study confirms that Q-ICAN can accurately detect the CPA attack in both LRU and LFU caching strategies. As shown in Figure 21.a and 21.b, the CHR increases from 0% to above 90%. This increase in CHR demonstrates Q-ICAN's ability to detect and block malicious interests effectively, even when the caching policy varies.



(a) LRU policy. (b) LFU policy.

Figure 21: Performance of CHR using different caching policies.

To sum up, our study has demonstrated that Q-ICAN is highly effective in mitigating CPA attacks and optimizing the CHR to exceed a threshold of 90% in all cases, even under challenging scenarios and stressful conditions.

6.2.3. Evaluation of the Average Retrieval Delay (ARD)

The Average Retrieval Delay (ARD) is a useful evaluation metric that measures the time it takes for a consumer to retrieve the desired content. Our study, as shown in Figure 22, highlights that a legitimate consumer's retrieval time in the presence of a CPA attack is 18.3 ms. However, the introduction of Q-ICAN agent results in a remarkable reduction of retrieval time to 15.5 ms, demonstrating the Q-ICAN model's efficiency in conserving and optimizing the ARD under critical attacks. With an average time difference of 2.8 ms, Q-ICAN effectively detects and mitigates potential threats in real-time, maintaining a fast and efficient content retrieval operation. These results indicate the significance of CPA detection solutions such as Q-ICAN in improving user experience and increasing network availability.

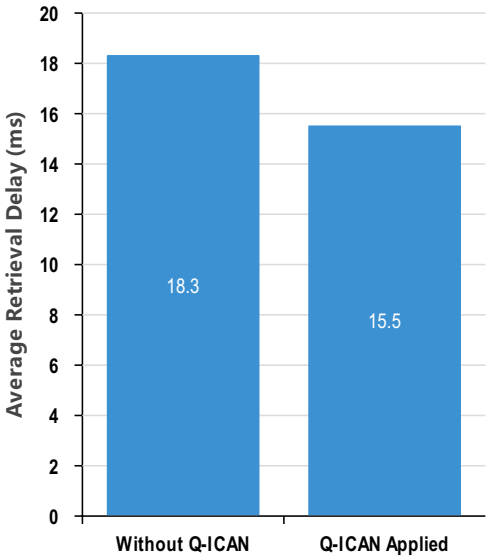


Figure 22: Average Retrieval Delay under CPA.

6.2.4. Q-ICAN efficiency

To assess the effectiveness of our mitigation mechanism Q-ICAN, we measured its efficiency using several AI performance parameters, namely (1) Accuracy, (2) Precision, (3) Recall, (4) Specificity, and (5) F1 Score. These parameters are defined mathematically as follows:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{2}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{3}$$

$$Specificity = \frac{TN}{(FP + TN)} \tag{4}$$

$$F1\ Score = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \tag{5}$$

Where:

- *False Positive (FP) :* refers to the number of legitimate packets that are incorrectly classified as malicious.
- *False Negative (FN) :* refers to the number of malicious packets that are incorrectly classified as legitimate.
- *True Positive (TP) :* refers to the number of legitimate packets that are correctly classified as legitimate.

- *True Negative (TN) :* refers to the number of malicious packets that are correctly classified as malicious.

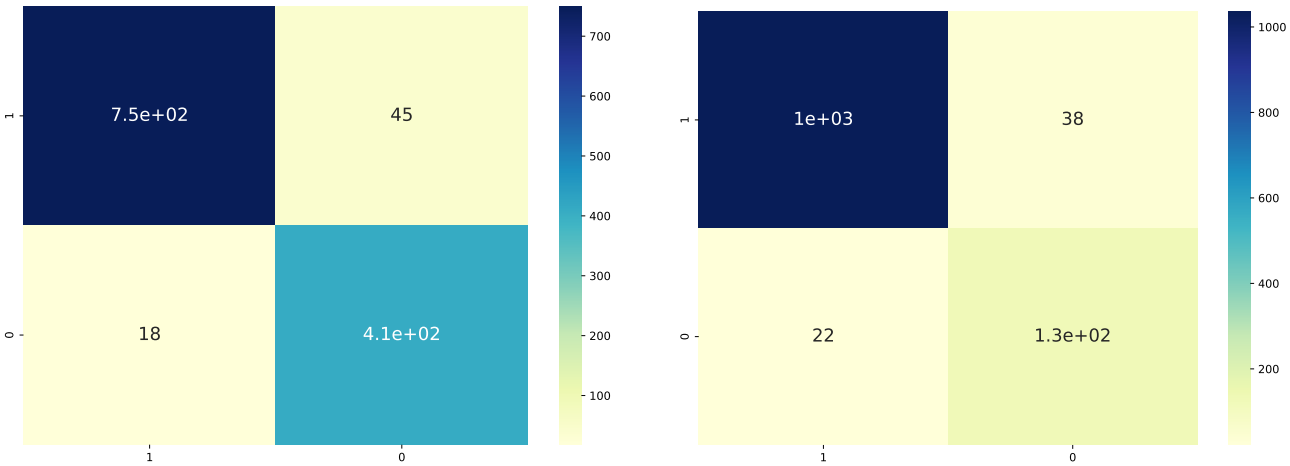
For a comprehensive evaluation of the effectiveness of the proposed intelligent mechanism, we present in Table 4 and Figure 23 the performance parameters: Accuracy, Precision, Recall, Specificity, and F1 Score, along with the corresponding confusion matrix for both the DFN and AT&T topologies. These metrics provide valuable insights into the performance and efficiency of the mechanism in different network scenarios.

In the DFN topology, Q-ICAN achieved an accuracy of 95% with an FP value of 38 and an FN value of 22. The precision value reached 96.46%, indicating the percentage of correctly identified positive detections. The recall value was 97.92%, indicating the percentage of truly positive values detected in all CPA states. Specificity refers to the ability to correctly identify positive results, and in this case, the probability of correctly detecting a positive CPA attack was measured at 76.82%. Finally, the F1 score, calculated as the harmonic mean of precision and recall, yielded a value of 293.76% using Q-ICAN. Similarly, in the AT&T topology, Q-ICAN achieves an accuracy of 95% with FP and FN values of 45 and 18, respectively. The precision reaches 96.33%, confirming the high level of accuracy in identifying positive detections even in large real-world topologies. The recall value of 97.92% indicates a high percentage of correctly identified true positive values. However, the specificity and F1 score produce values of 90.13% and 292.96%, respectively, suggesting the possibility of some false positives in the results.

These results indicate the high efficiency of Q-ICAN's CPA detection mechanism. Additionally, the lower FN value and higher FP value in both topologies demonstrate Q-ICAN's advantage in prioritizing legitimate consumers' content retrieval while avoiding excessive blocking that could adversely impact the NDN network's performance, as shown in the confusion matrix in Figure 23.a and 23.b.

Topology	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1 score (%)
DFN	95,09	96,46	97,92	76,82	97,19
AT & T	95,01	94,33	97,65	90,13	95,97

Table 4: Q-ICAN Efficiency.



(a) DFN topology.

(b) AT&T topology.

Figure 23: Confusion Matrix of Q-ICAN in DFN and AT&T topologies.

6.3. Comparative Analysis of Q-ICAN and state-of-the-art solutions

In this section, we provide a comparative analysis of Q-ICAN and state-of-the-art solutions. First, we compare Q-ICAN to the previous solution ICAN, using various metrics, as summarized in Table 5. Firstly, we compare the two solutions in terms of CHR. Figure 24 illustrates the CHR values under CPA in the DFN Topology without a CPA mitigation mechanism (24.a), with ICAN (24.b) and with Q-ICAN (24.c). The obtained results reveal that in the absence of a CPA mitigation mechanism (Figure 24.a), the attack causes a sharp decline in the CHR, reaching 0%

in multiple time steps. By using ICAN (Figure 24.d), the CHR increases and stabilizes around 56%, while Q-ICAN (Figure 24.b) significantly improves the CHR, stabilizing at 94%. Throughout the simulation time, Q-ICAN achieves an average CHR of 72.55%, whereas ICAN is limited to 50.53%.

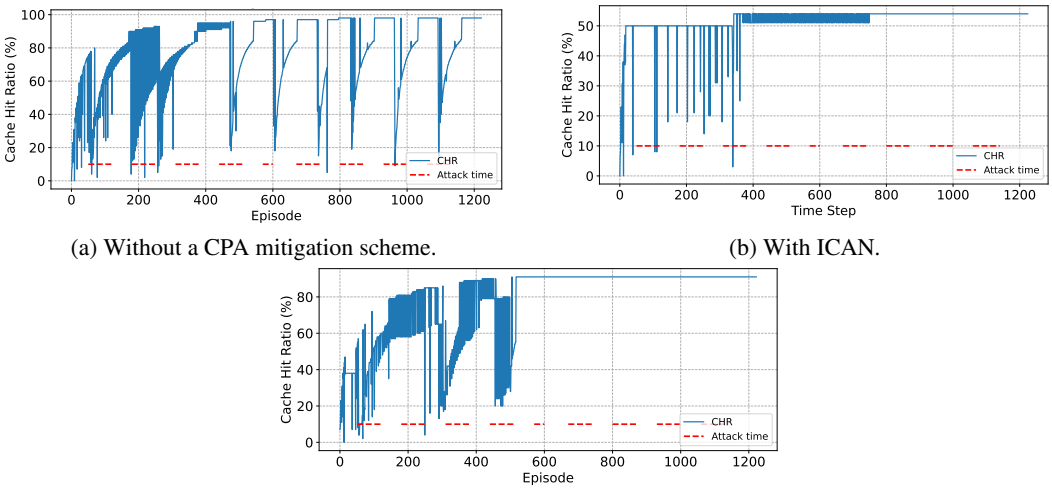
Secondly, we compare the *Accuracy* of Q-ICAN and ICAN. The results obtained confirm that Q-ICAN demonstrates superior accuracy compared to ICAN, even in larger and more complex topologies, with an accuracy of 95.09%, whereas ICAN averages an accuracy of 92.3%.

	AVG-CHR	**Accuracy**	**Identify leakage**	**Resource usage**		
				CPU	**Memory**	**Space storage**
ICAN (30)	50.53%	92.3%	No	low	low	low
Q-ICAN	72.55%	95.09%	No	low	low	low

Table 5: Comparative analysis of ICAN and Q-ICAN.

Finally, we conduct a qualitative comparison of the two solutions in terms of *Identity Leakage* and *Resource Usage*. The revelation of node identities in the NDN network can potentially compromise the philosophy of the architecture. Both ICAN and Q-ICAN successfully mitigate CPA without disclosing the identities of consumers, routers, or producers, affirming that the two solutions are conservative and respect the security semantics of the NDN architecture. Furthermore, in terms of *Resource Usage*, both solutions exhibit low CPU, memory, and space usage.

To summarize, Q-ICAN has demonstrated its efficiency in various performance aspects when compared to the solutions mentioned in the related work section (see Table 6). It effectively discovers and mitigates attacks without disclosing the identity of any nodes, compromising CPU usage efficiency, or consuming significant space on NDN routers. Q-ICAN is a lightweight solution that provides real-time protection to the content store of each NDN router. It acts as a reliable caching mechanism that selectively allows only legitimate content transmission. In comparison to state-of-the-art solutions mentioned earlier in Section 4, our solution consistently improves the cache hit ratio (CHR) to 72.55% (refer to Figure 25). Therefore, Q-ICAN outperforms other solutions in terms of CHR enhancement, making it a superior choice in the field.



(a) Without a CPA mitigation scheme.

(b) With ICAN.

(c) With Q-ICAN.

Figure 24: Comparison of ICAN and Q-ICAN performance in terms of CHR.

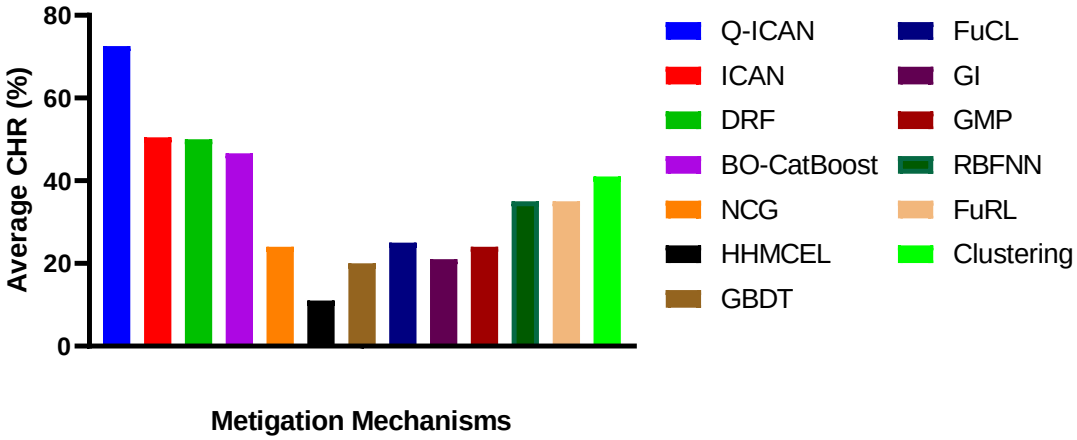


Figure 25: Comparative Analysis of Average CHR: Q-ICAN vs. State-of-the-Art Detection and Mitigation Mechanisms.

Mechanism	Memory Overhead	Computational Time	Privacy Conservation	Detection Pre/Post Attack	Accuracy	Topology	Space Storage
Q-ICAN	Low	Low	No leakage	Pre	High	DFN/AT&T	Low
ICAN	Low	Low	No leakage	Pre	High	DFN	Low
SVM (40)	Low	Low	No leakage	Post	Medium	XC/DFN	High
DRF (51)	High	High	No leakage	Post	Medium	Self-made	High
BO-CatBoost (50)	High	High	No leakage	Post	Medium	Tree/DFN	High
NCG (49)	High	High	No leakage	Post	Medium	Self Made	High
CNN and FDT (41)	High	High	No leakage	Post	Medium	Self Made	High
HHMCEL (46)	High	Low	No leakage	Post	High	Manhattan topology	High
GBDT (44)	Low	High	Leakage	Post	High	Self Made	Low
FuCL (47)	High	Medium	No Leakage	Post	Medium	Self Made	Low
GI (42)	High	High	No leakage	Post	Medium	Self Made	High
GMP (43)	High	High	Leakage	Post	High	AS-3967/DFN	High
RBFNN (45)	Low	High	No leakage	Post	Medium	XC/DFN	High
FuRL (48)	High	High	No leakage	Post	Medium	AS-3967/DFN	High
Clustering (39)	Low	Low	No Leakage	Post	High	Tree/Mesh	Medium

Table 6: Comparative Analysis: Q-ICAN and AI-based Solutions in the State-of-the-Art.

7. Conclusion and Future Work

NDN is susceptible to several vulnerabilities that can severely disrupt the functionality of its essential components. CPA is one of the most harmful attacks, with the highest impact on the availability of contents in the cache of the CS. Multiple mitigation mechanisms have been suggested by the researcher for such attack, but the question always

remains: whether the attack is still effective under all these mitigation mechanisms and whether these mechanisms negatively affect the performance of the essential components of NDN.

In this paper, we propose Q-ICAN, a lightweight detection and mitigation mechanism that relies on the Q-Learning algorithm to prevent CPA attacks. Through several simulations, Q-ICAN has demonstrated its beneficial effect on mitigating the cache of the CS, using a variety of metrics that directly relate to the CS and consumers within the same security path. Furthermore, our real-time detection and mitigation mechanism exhibits high accuracy in detecting the presence of the attack and defending the core of the CS. Moreover, it improves the cache process, leading to better performance even in the normal state where no attack is present.

Although Q-ICAN has proven to be an effective countermeasure implemented in each router of the NDN network to detect and mitigate CPA attacks. our future work entails designing a collaborative communication system among Q-ICAN agents to provide even greater protection. Specifically, we aim to create a global model of Q-ICAN that can communicate with the sub-model in each NDN router, facilitating the sharing of information about suspected attacks and enabling early-stage mitigation. Furthermore, we plan to investigate the use of Deep Reinforcement Learning (DRL) to optimize the performance of our agents. Our objective is to enhance the detection rate of malicious attacks through the application of advanced deep learning techniques.

Abbreviations

The following abbreviations are used in this manuscript:

PIT	Pending Interest Table
CS	Content Store
FIB	Forwarding Information Base
CPA	Cache Pollution Attack
CDN	Content Delivery Network
P2P	Peer to Peer
DDB	Distributed Database
LRU	Least Recently Used
LFU	Least Frequently Used
FLA	False Locality Attack
LDA	Locality disruption attack
CHR	Cache Hit Ratio
ARD	Average Retrieval Delay
MiTM	Man-in-the-Middle
ICN	Information-Centric Networking
RL	Reinforcement Learning
CHR	Cache Hit Ratio
LDA	Locality-Disruption Attack
SVM	Support Vector Machine
CNN	Convolutional Neural Network
FDT	fuzzy decision tree
GBDT	Gradient Boost Decision Tree
RBFNN	Radial Basis Function Neural Network
HHMCEL	Hybrid Heterogeneous Multi-classifier Ensemble learning
FuCL	Fuzzy C-Means Clustering
AQM	Active Queue Management
FuRL	Fuzzy RBM Learning framework
IAT	Inter-Arrival Time
AVG-IAT	Average Inter-Arrival Time
HC	Hop Count
AVG-CHR	Average Cache Hit Ratio
ICAN	Intrusion detection system for CPA attack in NDN
DRF	Dynamic Random Forest
NCG	Non-Cooperative Game
GMP	Grey Model Prediction
GI	Gini Impurity

Acknowledgement

This research work has been carried out jointly at IRT SystemX, IReSCoMath Research Lab and INRIA of Paris in the scope of the project EXPLO, which has received funding from IRT SystemX.

References

[1] Cisco Annual Internet Report (2018–2023) White Paper. (n.d.). Cisco. Retrieved February 27, 2023, from https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] Passarella, A. (2012). A survey on content-centric technologies for the current Internet: CDN and P2P solutions. Comput. Commun., 35, 1-32.

[3] Ji-rong, Z. (2011). A survey of streaming media technology based on P2P and CDN. Journal of Xi'an University of Posts and Telecommunications.

[4] Ghaznavi, M., Jalalpour, E., Salahuddin, M.A., Boutaba, R., Migault, D., Preda, S. (2021). Content Delivery Network Security: A Survey. IEEE Communications Surveys Tutorials, 23, 2166-2190.

[5] Abudaqa, A.A., Mahmoud, A.S., Abu-Amara, M.H., Sheltami, T.R. (2020). Survey of Network Coding Based P2P File Sharing in Large Scale Networks. Applied Sciences.

[6] Niranchana (2019). Survey on Mobile Network Operators in P2P Applications.

[7] Bhatia, M., Rai, M.K. (2017). Identifying P2P traffic: A survey. Peer-to-Peer Networking and Applications, 10, 1182-1203.
[8] Nobakht, M., Mahmoudi, H., Rahmanzadeh, O. (2022). A Distributed Security Approach against ARP Cache Poisoning Attack. Proceedings of the 1st Workshop on Cybersecurity and Social Sciences.
[9] Özalp, A. N., Albayrak, Z., Çakmak, M., Özdoğan, E. (2022). Layer-based examination of cyber-attacks in IoT. 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE. doi: 10.1109/HORA55278.2022.9800047
[10] Hasimoto-Beltrán, R., López-Fuentes, F.D., Vera-Lopez, M. (2018). Hierarchical P2P architecture for efficient content distribution. Peer-to-Peer Networking and Applications, 12, 724-739.
[11] Shah, M.S., Leau, Y., Yan, Z., Anbar, M. (2022). Hierarchical Naming Scheme in Named Data Networking for Internet of Things: A Review and Future Security Challenges. IEEE Access, 10, 19958-19970.
[12] Quevedo, J., Corujo, D.N. (2022). Selective Content Retrieval in Information-Centric Networking. Sensors (Basel, Switzerland), 22.
[13] C. N., P., Vimala, H. S., J., S. (2023). A systematic survey on content caching in ICN and ICN-IoT: Challenges, approaches and strategies. Comput. Networks, 233, 109896. doi: 10.1016/j.comnet.2023.109896
[14] Rosli, A., Hassan, S., Omar, M. H. (2023). Data authentication mechanism using blockchain's proof-of-trust mechanism in named data networking. AIP Conf. Proc., 2608(1). doi: 10.1063/5.0128154
[15] Marques, D., Senna, C.R., Luís, M. (2022). Forwarding in Energy-Constrained Wireless Information Centric Networks. Sensors (Basel, Switzerland), 22.
[16] Hidouri, A., Hajlaoui, N., Touati, H., Hadded, M., Muhlethaler, P. (2022). A Survey on Security Attacks and Intrusion Detection Mechanisms in Named Data Networking. Comput., 11, 186.
[17] Hidouri, A., Hadded, M., Touati, H., Hajlaoui, N., Muhlethaler, P. (2022). Attacks, Detection Mechanisms and Their Limits in Named Data Networking (NDN). Communication Systems and Applications.
[18] Hidouri, A., Hadded, M., Hajlaoui, N., Touati, H., Mühlethaler, P. (2021). Cache Pollution Attacks in the NDN Architecture: Impact and Analysis. 2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 1-6.
[19] Aboud, A., Touati, H.: Geographic Interest Forwarding in NDN-Based Wireless Sensor Networks. In: 2016 IEEE/ACS 13th International Conference on Computer Systems and Applications (AICCSA). pp. 1-8(2016). https://doi.org/10.1109/AICCSA.2016.7945683
[20] Aboud, A., Touati, H., Hnich, B.: Efficient forwarding strategy in a NDN-based internet of things. Cluster Comput. 22(3): 805-818(2019). https://doi.org/10.1007/s10586-018-2859-7
[21] Touati, H, Aboud, A, Hnich, B. Named Data Networking-based communication model for Internet of Things using energy aware forwarding strategy and smart sleep mode. Concurrency Computat Pract Exper. 2022; 34(3):e6584. https://doi.org/10.1002/cpe.6584
[22] Aboud, A, Touati, H, Hnich, B. Hybrid 802.11p-cellular architecture for NDN-based VANET. Int J Commun Syst. 2023; 36(3):e5393. doi:10.1002/dac.5393
[23] Mejri, S., Touati, H., Kamoun F.: Are NDN Congestion Control Solutions Compatible with Big Data Traffic? In: 2018 International Conference on High Performance Computing & Simulation, (HPCS), pp. 978-984(2018). https://doi.org/10.1109/HPCS.2018.00154
[24] Nan, G., Qiao, X., Tu Y., Tan W., Guo L., Chen J.: Design and Implementation: the Native Web Browser and Server for Content-Centric Networking. Computer Communication Review 45(5): 609-610(2015). https://doi.org/10.1145/2829988.2790024
[25] Qiaoa, X., Rena, P., Chen, J., Tan, W., Blake, M. B., Xu, W.: Session persistence for dynamic web applications in Named Data Networking. Journal of Network and Computer Applications, 125:pp.220-235(2019). https://doi.org/10.1016/j.jnca.2018.10.015
[26] Ullah, R., Rehman, M.A.U., Kim, B.S.: Design and Implementation of an Open Source Framework and Prototype For Named Data Networking-Based Edge Cloud Computing System. IEEE Access, 7:57741-57759(2019). https://doi.org/10.1109/ACCESS.2019.2914067
[27] NDN Community Meeting 2023[Online].Available at: https://www.nist.gov/news-events/events/2023/03/ndncomm-2023 Accessed 10 Mar 2023.
[28] Mejri, S., Touati, H., Kamoun F.: Hop-by-hop interest rate notification and adjustment in named data networks. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6(2018).
[29] Touati, H., Mejri, S., Malouch, N. et al. Fair hop-by-hop interest rate control to mitigate congestion in named data networks. Cluster Comput 24, 2213–2230 (2021). https://doi.org/10.1007/s10586-021-03258-8
[30] Hidouri, A., Touati, H., Hadded, M., Hajlaoui, N., Muhlethaler, P. (2022). A Detection Mechanism for Cache Pollution Attack in Named Data Network Architecture. International Conference on Advanced Information Networking and Applications.
[31] Azamuddin, W. M. H., Aman, A. H. M., Sallehuddin, H., Abualsaud, K., 38; Mansor, N. (2023). The emerging of named data networking: Architecture, application, and technology. IEEE Access, 11, 23620–23633. https://doi.org/10.1109/access.2023.3243006
[32] Kutscher, D., Eum, S., Pentikousis, K., Psaras, I., Corujo, D., Saucez, D., Schmidt, T., 38; Waehlisch, M. (n.d.). RFC 7927: Information-Centric networking (ICN) research challenges. https://www.rfc-editor.org/rfc/rfc7927.html
[33] Muscariello, L., Papalini, M., Roques, O., Sardara, M., 38; Van, A. T. (2023). Securing scalable real-time multiparty communications with hybrid information-centric networking. ACM Transactions on Internet Technology. https://doi.org/10.1145/3593585
[34] Tschudin, C. (n.d.). File-Like ICN collections (FLIC). IETF Datatracker. https://datatracker.ietf.org/doc/draft-irtf-icnrg-flic/04/
[35] Kumar, N., Singh, A.K., Aleem, A. et al. Security Attacks in Named Data Networking: A Review and Research Directions. J. Comput. Sci. Technol.34, 1319–1350 (2019). https://doi.org/10.1007/s11390-019-1978-9
[36] M. S. M. Shah, Y. -B. Leau, M. Anbar and A. A. Bin-Salem, "Security and Integrity Attacks in Named Data Networking: A Survey," in IEEE Access, vol. 11, pp. 7984-8004, 2023, doi: 10.1109/ACCESS.2023.3238732.
[37] Zhou, J., Luo, J., Deng, L., 38; Wang, J. (2020, August 9). Cache pollution prevention mechanism based on cache partition in V-NDN. 2020 IEEE/CIC International Conference on Communications in China (ICCC). http://dx.doi.org/10.1109/iccc49849.2020.9238838
[38] H. Salah, M. Alfatafta, S. SayedAhmed and T. Strufe, "CoMon++: Preventing Cache Pollution in NDN Efficiently and Effectively," 2017 IEEE 42nd Conference on Local Computer Networks (LCN), Singapore, 2017, pp. 43-51, doi: 10.1109/LCN.2017.35.
[39] Nasserala, A., Bastos, I.V., Moraes, I.M. (2018). Cache nFace: a simple countermeasure for the producer-consumer collusion attack in Named Data Networking. Annals of Telecommunications, 74, 125-137.

[40] Cao, Y., Wu, D., Hu, M., Chen, S. (2023). Detection and Defense Schemes for Cache Pollution Attack in Content-Centric Network. In: Quan, W. (eds) Emerging Networking Architecture and Technologies. ICENAT 2022. Communications in Computer and Information Science, vol 1696. Springer, Singapore. https://doi.org/10.1007/978-981-19-9697-9_49

[41] R., B., Joseph K, S. (2022). Multi-classifier and meta-heuristic based cache pollution attacks and interest flooding attacks detection and mitigation model for named data networking. Journal of Experimental amp; Theoretical Artificial Intelligence, 1–26. https://doi.org/10.1080/0952813x.2022.2115141

[42] Singh, V. P., Ujjwal, R. L. (2020). Gini impurity based NDN cache pollution attack defence mechanism. Journal of Information and Optimization Sciences, 41(6), 1353–1363. https://doi.org/10.1080/02522667.2020.1809092

[43] Yao, L., Zeng, Y., Wang, X., Chen, A., Wu, G. (2020). Detection and defense of cache pollution based on popularity prediction in named data networking. IEEE Transactions on Dependable and Secure Computing, 1–1. https://doi.org/10.1109/tdsc.2020.2967724

[44] Man, D., Mu, Y., Guo, J., Yang, W., Lv, J., Wang, W. (2021). Cache pollution detection method based on GBDT in information-centric network. Security and Communication Networks, 2021, 1–10. https://doi.org/10.1155/2021/6658066

[45] Buvanesvari, R. M., Suresh Joseph, K. (2020). RBFNN: A radial basis function neural network model for detecting and mitigating the cache pollution attacks in named data networking. IET Networks, 9(5), 255–261. https://doi.org/10.1049/iet-net.2019.0156

[46] Yao, L., Zheng, Z., Wang, X., Zeng, Y., Wu, G. (2022). Detection of cache pollution attack based on ensemble learning in icn-based VANET. IEEE Transactions on Dependable and Secure Computing, 1–12. https://doi.org/10.1109/tdsc.2022.3196109

[47] Rani, V., Joshua T, A., Narasimma Mallikaarjunan, K., Harinarayan R, R. A., Dharani, J., Shalinie S, M. (2021, July 6). Exploiting queue-driven cache replacement technique for thwarting pollution attack in ICN. 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). http://dx.doi.org/10.1109/icccnt51525.2021.9579599

[48] Rani, P. V., Shalinie, S. M. (2020). FuRL: Fuzzy RBM learning framework to detect and mitigate network anomalies in Information Centric Network. Sādhanā, 45(1). https://doi.org/10.1007/s12046-020-01331-3

[49] Yao, L., Chen, Z., Dai, H., Wu, G. (2021). Exploiting Non-Cooperative Game Against Cache Pollution Attack in Vehicular Content Centric Network. IEEE Trans. Dependable Secure Comput., 19(6), 3873–3886. doi: 10.1109/TDSC.2021.3109046

[50] Liu, L., Peng, S. (2022). Detection of A Novel Dual Attack in Named Data Networking. 2022 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE. doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom57177.2022.00008

[51] Babu, V. J., Jose, M. V. (2023). Dynamic forest of random subsets-based one-time signature-based capability enhancing security architecture for named data networking. Int. j. inf. tecnol., 15(2), 773–788. doi: 10.1007/s41870-021-00786-9

[52] Alabadi, M., Albayrak, Z. (2020). Q-Learning for Securing Cyber-Physical Systems : A survey. 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE. doi: 10.1109/HORA49412.2020.9152841

[53] Kumar, N., Srivast, S. (2021). Fast detection and traceback-based mitigation of interest flooding attack. Research Square Platform LLC. http://dx.doi.org/10.21203/rs.3.rs-437520/v1

[54] Watkins, Christopher and Peter Dayan. "Q-learning." Machine Learning 8 (1992): 279-292.

[55] Ryu, S., Joe, I., Kim, W. (2021). Intelligent Forwarding Strategy for Congestion Control Using Q-Learning and LSTM in Named Data Networking. Mob. Inf. Syst., 2021, 5595260:1-5595260:10.

[56] Hnaien, H., Touati, H. (2020). Q-Learning Based Forwarding Strategy in Named Data Networks. Computational Science and Its Applications – ICCSA 2020, 12249, 434 - 444.

[57] Lan, Dehao et al. "A Deep Reinforcement Learning Based Congestion Control Mechanism for NDN." ICC 2019 - 2019 IEEE International Conference on Communications (ICC) (2019): 1-7.