Detection and Defense of Cache Pollution Based on Popularity Prediction in Named Data Networking

Lin Yao, Yujie Zeng, Xin Wang, Ailun Chen and Guowei Wu

Abstract—Named Data Networking (NDN) as one of the most promising information-centric networking architectures can improve the network performance by supporting the large scale content distribution. However, the use of in-network caching mechanism increases the opportunity of cache pollution attack, where the attackers intend to reduce the cache hit of legal users by releasing fake requests to fill the precious cache with non-popular contents. To prevent the degradation of network performance caused by such an attack, it is becoming particularly important to detect the attack and then throttle it. In this paper, we propose a detection and defense scheme with the help of grey forecast, which can effectively exploit the regularity of past Interests and popularity by comprehensively considering three major factors to predict the future popularity of each cached content. If the predicted popularity of any content differs too much from the actually calculated one in several consecutive slices, the pollution attack will be determined. Once the attack is detected, the defense will be taken by suppressing the popularity increase of the suspicious content to mitigate the damage of the pollution attack. We also consider a special case, where there exists a sudden burst of traffic from legal users that cannot be simply dropped. The simulations in ndnSim indicate that our proposed method is effective in detecting and defending the pollution attack with higher cache hit, higher detecting ratio, and lower hop count compared to other state-of-the-art schemes.

Index Terms—Cache Pollution Attack; Grey Prediction; Popularity Prediction; NDN

1 Introduction

Named Data Networking (NDN) is one of five projects funded by the U.S. National Science Foundation under its Future Internet Architecture Program [1], and is one of the most promising information-centric networking architectures. Different from the current host-centric network, a user in NDN does not care where the content comes from, but is only interested in what the content is. Instead of routing a packet based on its destination IP address, a router in NDN routes the incoming packet following its inside content name. In order to alleviate the traffic pressure on the network bandwidth and make full use of the network resource, a common approach of NDN is to provide the in-network caching to speed up content distribution [2]. To obtain a content, the pull-based mode is implemented in NDN. An Interest packet can be sent by a user to request contents. In Fig. 1, the Interest packet is forwarded until it arrives at a content provider $R1$ which sends Data packet towards the requester along the reverse path of the Interest. $R4$ and $R6$ decide whether to cache the Data packet based on their cache replacement policies.

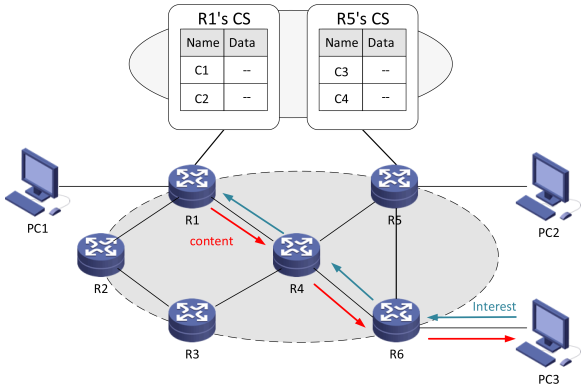


Fig. 1: Data path in NDN

Though the ubiquitous in-network caching can improve the network performance by reducing the delay of content retrieval, the pervasive caching is subject to cache pollution attack [3] [4]. Under the pollution attack, an adversary intentionally sends a large number of Interests for non-popular data, with an aim of compromising the caching balance and thereby tricking NDN routers into caching non-popular contents. As a result, the quality of service for legal users is degraded. There are two types of cache pollution attack [5]: Locality-Disruption Attack (LDA) and False-Locality Attack (FLA). A locality-disruption attacker continuously generates Interest packets for different non-popular contents, and fills the cache buffer with these unpopular contents to degrade the caching performance. A false-locality attacker repeatedly requests a specific set of non-popular contents to replace the popular contents. Both FLA and LDA aim to

- *Lin Yao, Yujie Zeng, Ailun Chen and Guowei Wu are with the Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province. Lin Yao are also with the DUT-RU International School of Information Science & Engineering, Dalian University of Technology, Dalian 116600, China. Ailun Chen, Yujie Zeng and Guowei Wu are also with the School of Software, Dalian University of Technology.(e-mail: yaolin@dlut.edu.cn).*
- *Xin Wang is with the Department of Electrical and Computer Engineering, Stony Brook University, New York State, U.S.A. (e-mail: x.wang@stonybrook.edu).*

degrade the hit ratio. In FLA, an attacker needs to infer the distribution of popular contents by sniffering the Interests of legitimate users. In LDA, an attack can be made successfully by requesting the contents with the uniform distribution without knowing the distribution of popular contents.

1.1 Motivation

It is critical to detect and throttle LDA and FLA. Some cache replacement algorithms such as LFU remove the least frequently requested content whenever the cache is overflow, which may help to alleviate the impact of LDA but are ineffective in preventing FLA [6]. Thus we mainly focus on dealing with FLA in this paper.

Some efforts have been made in recent years to address this issue. Generally, each NDN router stores statistics on incoming requests and sets some thresholds such as the number of repeated requests and the ratio of the number of repeated requests to the number of cache hits. If any given threshold is exceeded, FLA is considered to exist [7]. Some approaches detect FLA based on the cache replacement policy [3] [7], where the relationship between the inherent characteristics of the cached content and the content type (i.e., attack or non-attack) is considered. The path diversity of Interest packets is also exploited to detect the attack because the requests from attackers are unlikely to traverse as many different paths as the requests from legal users [8].

Despite the various efforts in detecting FLA, most of them do not consider the defense technique once FLA is detected and the subsequent requests from malicious users are simply discarded [5] [7] [9]. Furthermore, most of works have ignored a special case that a large number of users suddenly become interested in some non-popular data such as the old songs of a singer that becomes famous over night. In this situation, these requests should be considered as the legal packets rather than the malicious ones. For example, the Nobel prize on Literature was awarded to Bob Dylan in 2016, and he became the first musician awarded with the Novel prize. The news came as a surprise to people in the world. Then, some of his songs at TheTopTens such as "Blow in the Wind" and "Like a rolling stone" received a large number of hits suddenly. Without the capability of handling this kind of burst requests, the network performance can be significantly compromised. Therefore, appropriate defensive measures should not only reduce the extent of damage from FLA but also meet the requirements of legal users' requests.

1.2 Contributions and Organization

Considering the above challenges, we propose a detection and defense scheme of FLA based on Grey Prediction (FLAGP) for NDN. Different from the literature work on pollution detection which considers either only a specific factor or different factors independently, we design an algorithm that an NDN router can effectively predict the popularity of each cached content with grey theory by fully exploiting the past patterns of Interests and popularity of each cached content. FLA is then determined based on the difference between the predicted popularity and the computed value when the subsequent Interests are received. Given all the above considerations, this paper has the following contributions:

(1) To the best of our knowledge, we are the first to determine FLA according to the popularity difference between the predicted value and the actual one. To more accurately predict the popularity of a cached content, we design a function that can effectively integrate three types of statistics (request ratio, variance of repeated interests and standard deviation of request intervals) into one metric, with the three associated series predicted based on the grey prediction model $GM(1,n)$ [10].

(2) We are the first that controls the popularity increase to defend against FLA.

(3) We are the first to consider the special case that a large number of users suddenly become interested in some non-popular contents.

(4) FLAGP is compared with two other state-of-the-art pollution attack detection schemes CPMH [7] and LMDA [6], and the extensive simulations show that our scheme can achieve up to 50% higher detection ratio, 12.04% lower cache hit miss.

The rest of this paper is organized as follows. Section II reviews related work. In Section III, network model, problem statement, and grey prediction are introduced. The details of our proposed FLAGP scheme are presented in Section IV. We evaluate FLAGP and compare it with other schemes in Section V. Finally, we conclude our work in Section VI.

2 Related Work

Though the work on cache pollution attack has been proposed for Internet in [11] [12], only recently, researchers are drawn attention to cache pollution attack in the emerging NDN environment. There are two types of cache pollution attack: Locality-Disruption Attack (LDA) and False-Locality Attack (FLA). For more complete reviews, we discuss literature studies related to both, but we only consider FLA in this paper.

FLA-based Most approaches detected FLA based on the related statistics of the received Interests. In [7], Mauro et al. first proved that the cache pollution attack was a realistic threat in the large-scale NDN deployment and designed a detection scheme based on the change of probability in requesting a cached content. In [3], FLA was detected based on the goodness value of each content which was calculated from the cached content's longevity, access frequency and the corresponding hit ratio. If the goodness value was equal to a certain threshold value, FLA would be detected. Some monitoring nodes close to clients were selected to cooperatively record the network information such as the request rate and the hit ratio to detect the pollution attack [13]. In [14], the coefficient of variation was proposed to quantify the locality of each content. The lower the coefficient, the higher the locality. The content popularity and locality were combined to make caching decision so as to mitigate the effect of pollution attack. Though attackers can make some fake contents popular by frequently requesting them, they cannot change the locality of each router which can reflect the degree of equilibrium distribution of all Interests. Some approaches use the hierarchy of content name prefixes to determine FLA, because they assumed the unpopular contents requested by attackers had the same name prefixes. Flajolet-Martin sketch [5] was adopted to distinguish between the attack traffic and regular traffic by identifying prefixes of unpopular contents. Similar to [5], each router could determine FLA based on the prefixes requested by attackers in [6]. In [8], the authors exploited the path diversity of Interest packets to thwart FLA, with the assumption that requests from attackers were unlikely to traverse as many different paths as those from legal users.

LDA-based In 2012, Park et al. first proposed a detection approach against the pollution attack for content centric networking by using random checks [9]. In this scheme, the incoming packets are aggregated into a matrix. When the entropy of the matrix falls below a pre-defined threshold, the attack will be detected. CacheShield [15] was designed to detect LDA by exploiting the distinction of characteristics between normal requests and malicious requests. The schemes proposed in [3] [5] [7] can detect LDA too.

Summary- Compared with the detection and defense schemes in the literature, to more accurately detect FLA, we fully exploit the past patterns of Interests and popularity of each cached content to predict the future content popularity with grey theory. If the predicted popularity of any content and the actually calculated one differ too much in several consecutive slices, we can determine FLA. In order to suppress the excessive growth of popularity of those unpopular contents requested by attackers, our defense algorithm can dynamically adjust the content popularity by reducing the popularity increase constant and thereby avoiding the popular contents being removed from the buffer. To better serve the legitimate requests, our defense algorithm considers a specific but often observed case where a large number of users suddenly become interested in some non-popular data. However, existing schemes on FLA or LDA detection simply drop the suspicious packets or do not cache the suspicious contents.

3 Network model and Grey Prediction

In this section, we first introduce the network model and then briefly introduce the grey prediction method.

3.1 Network Model

NDN is a significant common approach taken by several future Internet research activities [16]. Its goal is to achieve efficient and reliable content distribution, with the focus on contents rather than the data carriers as done in conventional communications. In NDN, all content objects are identified by their names regardless of their locations and sources. It has two major types of packets, Interest and Data, which carry a name that identifies the content object requested or transmitted. Usually, a name in NDN assumes a hierarchically structure [1]. For example, a video produced by DLUT may have the name /dlut/videos/dance.mpg, where ”/” delineates the name components in text representations, similar to URLs.

Each NDN node maintains three data structures, Forwarding Information Base (FIB), Pending Interest Table (PIT), and Content Store (CS), as shown in Fig. 2. CS keeps a record of each cached content. PIT is a table that records all the Interests that a router has forwarded but are not satisfied yet. An Interest item in PIT contains the Interest name, incoming interface(s), and outgoing interface(s). FIB is a table for routing incoming Interest packets based on their name prefixes. Once an Interest arrives at a router with the requested content cached in its CS, it will reply with the Data packet sent towards the interface receiving the Interest. If the router does not have the requested content, it will aggregate the incoming interface of the Interest into PIT if the same Interest has been forwarded; Otherwise, the Interest will be forwarded. When a Data packets is received, if there is a matching entry in the FIB, the NDN router decides whether to cache it based on its cache replacement policy .

This information retrieving procedure makes NDN attractive to use in-network caching to speed up content distribution and improve network resource utilization. However, when the adversary can know some non-popular objects in advance by monitoring the request packets, the in-network caching mechanism is vulnerable to the pollution attack. By injecting a large number of requests for a small set of non-popular objects into the NDN network, the adversary attempts to waste precious cache resources of routers to cache non-popular contents. This will reduce the cache hit ratio for popular contents and increase the time delay for legitimate users to get the requested content. To quantize the attackers' strategy, we adopt two parameters to launch FLA in this paper, attacking power ratio φ and range ratio ξ [8]. The power ratio φ is a comparison between the number of malicious Interest packets from attackers and the number of normal Interest packets from legal uses. The range ratio is denoted as $\xi = \frac{|\mathcal{A}|}{Q}$, where $\mathcal{A}$ is the number of content objects attacked, and Q is the total number of content objects cached in each router. Based on these two parameters, we can easily construct a false-locality attack and evaluate the effectiveness of FLAGP in defending against FLA.

Problem Statement (Detecting and Defending FLA effectively in NDN):

The goal of our paper is to detect the attack and then throttle it. There are a few questions we would like to answer. First, how to implement such an attack in NDN using limited resources and apply it to a more practical and complex network topology? How to distinguish between the malicious and legal requests, and then to detect high and low rate attacks with a high accuracy? Furthermore, how to make proactive countermeasures effectively once FLA is detected?

3.2 Grey Prediction

Grey system theory was first proposed in 1982 [17], where Grey means that the quality is poor, incomplete, uncertain, etc. The primary characteristic of a grey system is the incompleteness of information. Grey system theory can be applied to prediction, decision-making, etc. In grey theory, each stochastic variable is considered as a grey number that varies within a certain range and time, and the corresponding stochastic process is regarded as a grey process. With Grey prediction, the future states of the system are

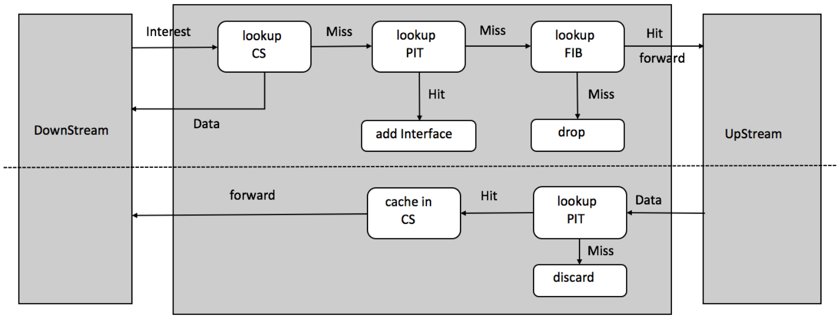


Fig. 2: Data process in NDN

forecasted by excavating the hidden laws of the system from unascertained characteristics of the original data sequences. Compared with methods such as those based on the conventional linear statistical models and artificial neural networks, the adoption of grey prediction theory can alleviate the difficulty of modeling the nonlinear, stochastic and highly non-stationary time series without requiring a large number of training data and relatively long training period [18]. $GM(1,1)$ and $GM(1,n)$ are two mainly existing grey prediction models.

GM (1, 1) $GM(1,1)$ stands for a first order grey model with one variable. Considering the original series $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), ..., x^{(0)}(m))$ as a nonnegative series and $X^{(1)} = (x^{(1)}(1), x^{(1)}(2), ..., x^{(1)}(m))$ as 1-Accumulated Generating Operation (1-AGO) series of $X^{(0)}$, where $x^{(1)}(k) = \sum_{j=1}^{k} x^{(0)}(j)$, $k = 1, 2, ..., m$ holds and the 1-AGO data of the original series $X^{(0)}$ are used as the intermediate information by the grey prediction model [10]. Let $Z^{(0)} = (z^{(0)}(2), z^{(0)}(3), ..., z^{(0)}(m))$ be the sequence generated from 1-AGO through the calculation of the mean of adjacent neighbors. The basic form of $GM(1,1)$ is defined as:

$$x^{(0)}(k) + az^{(1)}(k) = b, \tag{1}$$

where a is the development coefficient, b is the grey action quantity [19] and $z^{(1)}(k) = \frac{1}{2}(x^{(1)}(k) + x^{(1)}(k-1))$, $k = 2, 3, ..., m$. a and b satisfy $[a, b]^T = (B^T B)^{-1} B^T Y$, where $Y^T = [x^{(0)}(2), x^{(0)}(3), ..., x^{(0)}(m)]$ and B is a matrix shown as follow:

$$B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots & \vdots \\ -z^{(1)}(m) & 1 \end{bmatrix}.$$

Based on Eq.(1), $x^{(0)}(k)$ can be predicted

$$x^{(0)}(k) = \beta - \alpha x^{(1)}(k-1), \tag{2}$$

where $\alpha = \frac{a}{1+0.5a}$ and $\beta = \frac{b}{1+0.5a}$.

$GM(1,1)$ is a model constructed on a single sequence. It uses only the behavioral sequence of the system without considering any externally acting sequences.

GM(1, n) To obtain more accurate prediction, the prediction model $GM(1,n)$ was proposed in 1988 with $n-1$ relative factors being acted as the series $X_2^{(0)}, ..., X_n^{(0)}$ associated with the predicted series $X_1^{(0)}$, where $X_i^{(0)} = (x_i^{(0)}(1), x_i^{(0)}(2), ..., x_i^{(0)}(m))$, $i = 2, 3, \cdots, n$. $X_1^{(0)}$ represents the original series, $X_i^{(0)}$ is the data sequence of relevant factors, $X_i^{(1)}$ is the 1-AGO of $X_i^{(0)}$, $i = 2, 3, \cdots, n$, and $Z_1^{(1)}$ is formed with the mean of the adjacent neighbors of the sequence $X_1^{(1)}$ [10]. $GM(1,n)$ is defined as

$$x_1^{(0)}(k) + az_1^{(1)}(k) = \sum_{i=2}^{n} b_i x_i^{(1)}(k), \tag{3}$$

where a and b_i satisfy $[a, b_2, b_3, ..., b_n]^T = (B^T B)^{-1} B^T Y$. $Y^T = [x_1^{(0)}(2), x_1^{(0)}(3), ..., x_1^{(0)}(k)]$ and B is defined as follows:

$$B = \begin{bmatrix} -z_1^{(1)}(2) & x_2^{(1)}(2) & ... & x_n^{(1)}(2) \\ -z_1^{(1)}(3) & x_2^{(1)}(3) & ... & x_n^{(1)}(3) \\ \vdots & \vdots & \vdots & \vdots \\ -z_1^{(1)}(k) & x_2^{(1)}(k) & ... & x_n^{(1)}(k) \end{bmatrix}$$

To predict $X_1^{(0)}$, we adopt the following steps:

Step (1): From Eq. (3), we can get

$$x_1^{(0)}(k) = \sum_{i=2}^{n} \beta_i x_i^{(1)}(k) - \alpha x_1^{(1)}(k-1).$$

Step (2): Considering the correlation between the related sequences, we use the other $n-2$ associated sequences to generate $GM(1, n-1)$ model and get $X_2^{(0)}$.

$$x_2^{(0)}(k) = \sum_{i=3}^{n} \beta_i x_i^{(1)}(k) - \alpha x_2^{(1)}(k-1),$$

Similarly, we can get $X_3^{(0)}$, $X_4^{(0)}$,..., $X_n^{(0)}$.

Step (3): At last, we can get n models listed in Eq.(4), where a_j and $b_{j,i}$ in $GM(1, n-j+1)$ model satisfy $[a_j, b_{j,j+1}, b_{j,j+2}, ..., b_{j,n}]^T = (B_j^T B_j)^{-1} B_j^T Y_j$ with $j = 2, 3, \cdots, n-1$.

$$\begin{cases} GM(1,n): & x_1^{(0)}(k) + a_1 z_1^{(1)}(k) = \sum_{i=2}^{n} b_{1i} x_i^{(1)}(k) \\ GM(1,q): & x_j^{(0)}(k) + a_j z_j^{(1)}(k) = \sum_{i=j+1}^{n} b_{j,i} x_i^{(1)}(k) \\ (q = n-j+1) & \\ GM(1,1): & x_n^{(0)}(k) + a_n z_n^{(1)}(k) = b_{n0} \end{cases} \tag{4}$$

Then, $X_1^{(0)}$ can be deduced from the following equations,

$$\begin{cases} x_1^{(0)}(k) = \sum_{i=2}^{n} \beta_{1i} x_i^{(1)}(k) - \alpha_1 x_1^{(1)}(k-1) \\ x_j^{(0)}(k) = (\sum_{i=j+1}^{n} \beta_{j,i} x_i^{(1)}(k)) - \alpha_j x_j^{(1)}(k-1) \\ x_n^{(0)}(k) = \beta_{n0} - \alpha_n x_n^{(1)}(k-1), \end{cases} \qquad (5)$$

where $\alpha_j = \frac{a_j}{1+0.5a_j}$ and $\beta_{j,i} = \frac{b_{j,i}}{1+0.5a_j}$.

4 Detection and Defense of Cache Pollution Based on Grey Prediction

Our main research goal is to protect the NDN routers from FLA. In this section, we first introduce our basic framework and then elaborate the details of our FLAGP scheme.

4.1 Overview

As shown in Fig. 3, our FLAGP scheme includes two major modules, used for detecting FLA and defending FLA respectively. Under FLA, the adversaries select a subset of non-popular contents and issue Interests for this subset. The increase in the percentage of non-popular requests changes the distribution of Interests compared with that under the normal condition. As a result, the caches of NDN routers are polluted by replacing popular contents with non-popular ones. Detection aims to identify the Interests that are most likely generated by attackers. Defense aims to reduce the risk of FLA by controlling the fast popularity increment of non-popular contents. FLAGP works with the following basic steps:

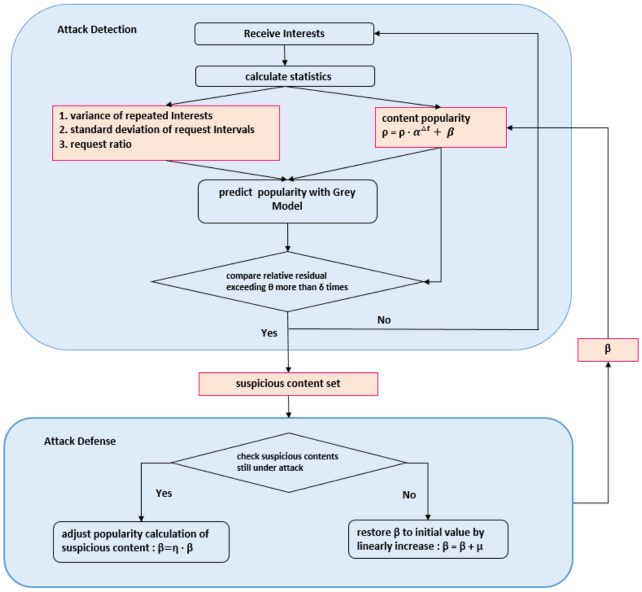

Attack Detection
Receive Interests
calculate statistics
1. variance of repeated Interests
2. standard deviation of request Intervals
3. request ratio
content popularity
ρ = ρ · α^Δt + β
predict popularity with Grey Model
compare relative residual exceeding θ more than δ times
No
Yes
β
suspicious content set
Attack Defense
check suspicious contents still under attack
Yes
No
adjust popularity calculation of suspicious content : β=η · β
restore β to initial value by linearly increase : β = β + μ

Fig. 3: Architecture of our FLAGP

1) An NDN router periodically makes statistics on the request ratio for the same content c_i, the variance of repeated Interests and standard deviation of request intervals between two adjacent Interests for c_i.
2) Periodically, each router computes the popularity increment of c_i based on the above statistics. If the calculated value deviates significantly from the one predicted by $GM(1,n)$ model, we add c_i into a suspicious list.
3) If c_i is judged as a suspicious content in the subsequent δ time slices, where a slice is a time period, we can determine FLA is launched on c_i.
4) Once FLA is detected, the defensive measure will be taken by controlling the popularity increment.

In our scheme, we adopt $GM(1,n)$ model to achieve the popularity prediction. As discussed before, grey theory mainly focuses on model uncertainty and information insufficiency when analyzing and understanding systems via research on prediction, decision making, etc. [18]. As a comparison, the statistics models and neural network-based approaches are too complex to be used in the time series prediction that demands a great deal of training data. Our detection model FLAGP can be considered as a grey system. $GM(1,n)$ uses accumulated generation operations to build Eq.(4) and Eq.(5). One of its significant characteristic is to consider the correlation between the input sequences. Besides, $GM(1,n)$ model can get a better prediction result even with less data, so it can overcome the data sparsity. The following three sequences in the preprocessing stage, the request ratio for c_i, the variance of repeated Interests and standard deviation of request intervals between two adjacent Interests for c_i, are the input required for model construction and subsequent forecasting.

4.2 Detection of FLA

Generally, the traffic under FLA must be different from the normal traffic. We exploit the traffic difference to detect FLA. Before presenting the detection module, we list frequently used notations in Table 1 and define some terminologies.

TABLE 1: Frequent notations

Notation	Description
$\mathcal{C}$	The set of content
c_i	The *i-th* content in $\mathcal{C}$
φ	The attacking power ratio
ξ	The range ratio
$\rho(c_i)$	The popularity of c_i
$\nu_k(c_i)$	Variance of repeated Interests for c_i in the *k-th* slice
$\sigma_k(c_i)$	Standard deviation of request intervals for c_i in the *k-th* slice
$\gamma_k(c_i)$	The request ratio of c_i in the *k-th* slice
$t(c_i)$	The time of receiving c_i
$E(X)$	The average of all elements in the set X
θ	The threshold of difference between the real and predicted popularity
δ	The threshold of suspicious Interest number in a time slice
$GM(1,n)$	Grey Prediction Model

Definition 1 (Request Ratio). $\gamma(c_i)$ is defined as the ratio between the number of Interests for a content c_i and that of all the requests received by the router during a slice,

$$\gamma(c_i) = \frac{n(c_i)}{\sum_{c_k \in \mathcal{C}} n(c_k)}, \qquad (6)$$

where $n(c_i)$ is the number of requests for c_i.

Definition 2 (Variance of Repeated Interests). $\nu(c_i)$ is defined as the variance of repeated Interests from all the router interfaces.

$$\nu(c_i) = \sum\nolimits_{j=1}^{n}(m_j - E(m))^2,$$

where n is the number of router interfaces, m_j is the number of requests for c_i received from the *j-th* interface, and $E(m) = \frac{1}{n}\sum_{j=1}^{n} m_j$ represents the expectation of all requests received by the router.

Definition 3 (Standard Deviation of Request Intervals). σ is defined as the standard deviation of request intervals for the same content in a slice,

$$\sigma(c_i) = \sqrt{\frac{1}{m}\sum\nolimits_{k=1}^{m}(\Delta t_k - E(\Delta t))^2}, \tag{7}$$

where Δt_k is the time interval between the *k-th* and *(k-1)-th* request for c_i in a slice, m is the total number of intervals for c_i in that slice and $E(\Delta t) = \frac{1}{m}\sum_{j=1}^{m}\Delta t$ is the interval expectation for c_i.

Definition 4 (Content Popularity). When an NDN router receives a new Interest for c_i, the popularity of c_i is computed as

$$\rho(c_i) = \rho(c_i) \cdot \alpha^{\Delta t} + \beta, \tag{8}$$

where α is a decay constant, Δt is the time interval between two consecutive Interests for c_i and β is the popularity increase constant. When a node receives a request for c_i, it will compute the time interval Δt between the current time and the last time of receiving the same request for c_i. Popularity should decrease exponentially with the time interval Δt. This equation considers both the frequency and freshness of the request.

Then the following steps show that how to detect FLA in Algorithm 1, which is divided into the following steps:

a. When an Interest for c_i is received, its popularity is computed with Eq. (8) (Line 6).

b. The popularity of c_i is predicted by $GM(1,4)$ model, where there are one original main series on the popularity and three associated series on the request ratio, variance of repeated requests and standard deviation of time intervals respectively. In FLAGP, we make statistics of the above three factors in the past 10 slices to generate the associate series. These associated series can reflect the receiving status of Interests for a given content, which can indirectly reflect the popularity change and help more accurately predict the popularity of the next time slice. Three steps in $GM(1,4)$ model are implemented to predict the future popularity if an Interests for c_i is received again.

c. If the popularity computed in **Step** (a) is θ times bigger than the predicted value in **Step** (b), the file name of c_i is added into a suspicious list and its corresponding counter is incremented by one. Otherwise, the popularity of c_i will be updated as usual (Lines 8 - 12).

d. If the number of times for c_i to be detected suspicious is more than δ in the sequence of slices followed, we can determine that FLA is launched. The defense will be taken in Algorithm 2.

Algorithm 1 FLA_Detection

```
Input:
  $\mathcal{P}$            ▷ set of popularity for all the contents
  $\mathcal{V}$     ▷ set of $\nu$ for all cotent during the past $n-1$ slices
  $\mathcal{S}$     ▷ set of $\sigma$ for all cotent during the past $n-1$ slices
  $\mathcal{R}$     ▷ set of $\gamma$ for all cotent during the past $n-1$ slices
  $\mathcal{S}\_req$           ▷ set of request in the $n$-th slice
Output:
  true $or$ false           ▷ true means FLA may happen
 1: $det\_res \leftarrow false$                  ▷ detection result
 2: $N \leftarrow 0$                 ▷ counter of suspicious Interests
 3: for each  $c_i \in \mathcal{S}\_req$ do
 4:     $\Delta t \leftarrow t(c_i) - t_{pre}(c_i)$
 5:                 ▷ $t_{pre}(c_i)$: time of receiving the last $c_i$
 6:     $\rho(c_i) \leftarrow \rho(c_i) \cdot \alpha^{\Delta t} + \beta$
 7:     $\rho'(c_i) \leftarrow GM(1,4)$
 8:     if $|\frac{\rho(c_i)-\rho'(c_i)}{\rho(c_i)}| \geqslant \theta$ then
 9:         $N \leftarrow N + 1$
10:     else
11:         $\mathcal{P}(c_i) \leftarrow \mathcal{P}(c_i) \cup \rho(c_i)$
12:     end if
13: end for
14: if $N \geqslant \delta$ then
15:     $det\_res \leftarrow$ true
16: end if
17: $return\ det\_res$
```

4.3 Defense of Cache Pollution

As soon as FLA is detected, the defense function will be taken against such pollution attack. There are two methods to resist the attack, avoiding storing unpopular content and discarding malicious Interests [6]. Our FLAGP adopts the first method by avoiding storing non-popular contents to protect the cache. However, in a special case, many people are suddenly interested in some non-popular contents such as the past news of one Nobel Prize winner. Considering it, we execute the cache replacement by dynamically evaluating the content popularity. The defense function (Algorithm 2) works as follows:

a. Once the defense procedure is initiated, FLAGP will dynamically adjust the popularity of the suspicious content c_i by reducing β in Eq. (8) as $\beta \leftarrow \eta \cdot \beta$, where η decreases with the increase of the number of malicious requests j for c_i, $\eta \leftarrow \eta \cdot \epsilon^j$ and $\epsilon < 1$ (Lines $8 - 13$).

b. If FLA can still be detected in the subsequent slices, β continues to decay as $\beta \leftarrow \eta \cdot \beta$, until the requests for c_i become normal or β approaches 0 so that the popularity tends to be constant.

c. Else, if the attack is stopped in the next slice, β will stop to decrease. If there is no suspicious Interest for c_i in the next three slices, β will increase linearly until restoring to the initial value (Lines $14 - 23$).

d. The probability is computed repeatedly as described in (b) and (c).

Algorithm 2 FLA_Defense

```
$\beta' \leftarrow \beta$                                          ▷ initial $\beta$ value
$flag\_stop \leftarrow 1$                            ▷ flag of stopping simulation
$det\_res \leftarrow false$                  ▷ detection result of current slice
$j$        ▷ number of Interest packets for $c_i$ of current slice
$y$                ▷ number of continuous slices not under FLA
while $flag\_stop$ do
    $det\_res \leftarrow FLA\_Detection(\mathcal{P}, \mathcal{V}, \mathcal{S}, \mathcal{R}, \mathcal{S}\_req)$
    if $det\_res == true$ then
        $y \leftarrow 0$
        if $\beta > 0$ then
            $\eta \leftarrow \eta \cdot \epsilon^j$                                      ▷ $\epsilon < 1$
            $\beta \leftarrow \eta \cdot \beta$
        end if
    else
        $y \leftarrow y + 1$
        if $y \geq 3$ then
            if $\beta + \mu < \beta'$ then
                $\beta \leftarrow \beta + \mu$
            else
                $\beta \leftarrow \beta'$
            end if
        end if
    end if
    if $sim\ is\ done$ then
        $flag\_stop \leftarrow 0$                          ▷ simulation done
    end if
end while
```

5 Performance Evaluation

In order to evaluate the performance of our scheme, we conduct our simulations using ndnSIM [20], a popular open-source NDN simulator based on NS-3. The primary parameters used in ndnSIM are provided in Table 2. The total number of content chunks used in the simulation is 10,000 and each chunk has the same size, 1MB. For each router, the CS size is set to 1% of the chunk number, and the PIT size is set to 12,000 entries. The cache is replaced based on the content popularity, with each router caching contents of higher popularity. In our simulations, we consider two topologies, DFN topology [6] [7] in Fig. 4 and AS-3967 network topology [13] in Fig. 5. DFN is a realistic topology including all of the necessary components of a useful network to understand the work principles of different schemes. AS-3967 network topology is a snapshot of a realistic ISP network. In DFN, there are 16 legitimate consumers and 4 attackers. In AS-3967, there are 80 nodes. Among them, 92 are legitimate consumers and 6 are attackers.

All our simulations span over 24 hours, and follow a similar pattern. We divide our simulations into two phases. During the first 12 hours, all interests are issued by legitimate users at a speed of 3,000 interests per second following the Zipf-like distribution [21] [22]. Then, adversaries launch attacks during the second phase, following the pattern introduced in Section III with the power ratio φ and range ratio ξ. φ is a ratio between the rate of malicious Interests from attackers and the rate of normal Interests from legal users. ξ is a ratio between the number of content objects attacked and the total number of content objects cached in each router.

TABLE 2: Simulation parameters

Parameter description	Value
LinkDelay (Router to Router)	5ms
Number of Content	10,000
CS Size	100
PIT Size	12,000
Content Size	1MB
Legitimate Request Speed	3000 Interests/s
Popularity Weight (α)	0.5
Popularity Increment (β)	0.5
Detection Threshold (δ)	5
β Increment (μ)	0.1
β Decay (ϵ)	0.9

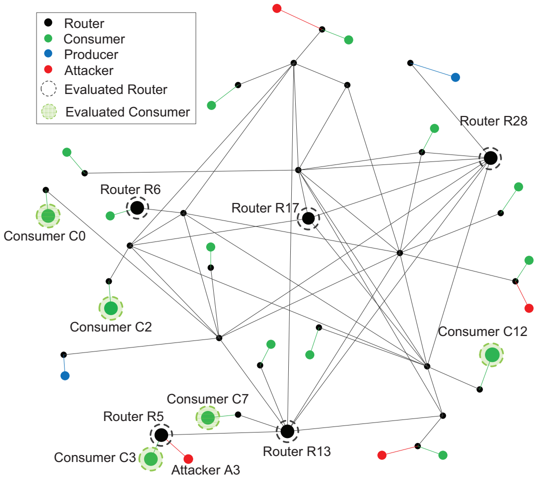


Fig. 4: DFN topology

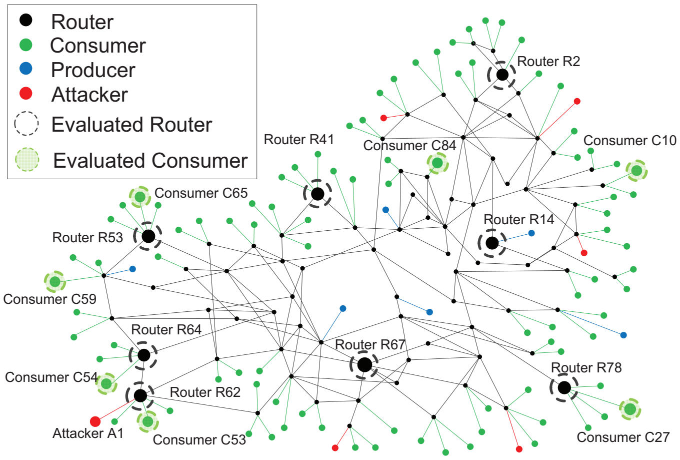
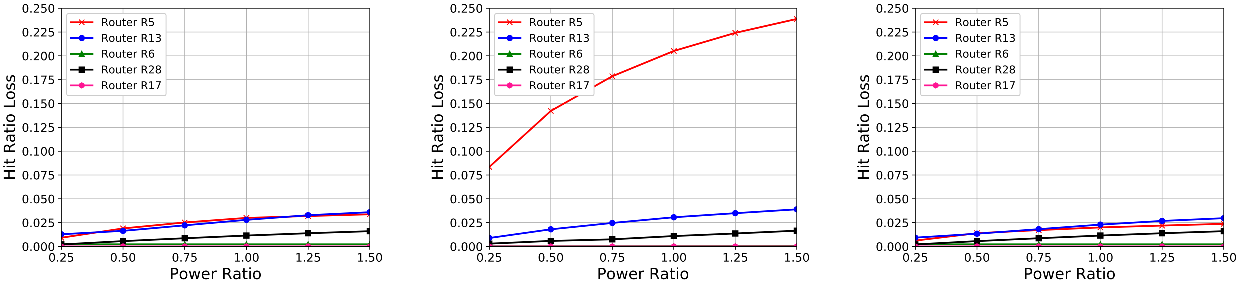


Fig. 5: AS-3967 topology

5.1 Performance Metrics

We compare our FLAGP scheme with LMDA [7] and CPMH [6]. FLAGP detects FLA according to the popularity increment between the actual popularity and the value predicted based on the past pattern of Interests and content popularity. In LMDA [7], FLA is determined by evaluating



(a) Hit ratio loss with LFU

(b) Hit ratio loss with LRU

(c) Hit ratio loss with Popularity

Fig. 6: Hit ratio loss with cache replacement policies under LDA in DFN

the probability variation computed based on the ratio of Interests for the same content. In CPMH [6], each router identifies the file prefixes of requested contents to detect FLA. Under FLA, the file prefixes of original non-popular contents become frequent in the requests. To compare them, we mainly use the following performance metrics:

1) **Hit Ratio Loss**, the comparison between the hit ratios with and without FLA.
2) **Average Hop Count**, the average number of hops through which a legitimate user receives the corresponding Data.
3) **Detection Ratio**, the ratio between the number of correct detections(including safe events and attack events) and the total number of detections.
4) **False Positive Ratio**, the ratio of the detections that falsely detects the safe event as an attack one.
5) **False Negative Ratio**, the ratio of the detections that falsely detects the attack event as a safe one.

5.2 Impacts of cache replacement strategies

In this set of simulations, we evaluate the damage effect from LDA and FLA by adopting different cache replacement strategies without any defense, LFU (Least Frequently Used) [23], LRU (Least Recently Used) [23] and the popularity-based cache policy in Eq. (8). LRU applies new data to replace data in storage locations that have not been accessed for the longest period, while LFU replaces cached data that are used the least often. The popularity-based cache policy replaces the data that have lower content popularity. Due to the page limit, we finish our simulations in the complex DFN and AS-3967 topologies and choose those representative consumers and routers to illustrate the performance. In Fig. 4, we choose consumers $C0$, $C2$, $C3$, $C7$ and $C12$ to evaluate the average hop count and routers $R5$, $R6$, $R13$, $R17$ and $R28$ to evaluate the hit ratio loss. In Fig. 5, we choose consumers $C10$, $C27$, $C53$, $C54$, $C59$, $C65$ and $C84$ to evaluate average hop count and $R2$, $R14$, $R41$, $R53$, $R62$, $R64$, $R67$ and $R78$ to evaluate the hit ratio loss. As the LinkDelay (Router to Router) in Table 2 is set to a fixed value 5ms, we use the average hop count to evaluate the delay from sending an Interest to receiving the corresponding Data. The power ratio φ varies from 0.25 to 1.5. The range ratio ξ is set to be 1.

5.2.1 *Impact under LDA*

Fig. 6 shows the impact of LDA on the caching performance at different routers under different cache replacement strategies in the DFN topology. In Fig. 6(b), only router $R5$ experiences the increase of hit ratio loss with φ under LRU, because $R5$ is connected with an attacker $A3$. Although the hit ratio losses in Fig. 6(a) and Fig. 6(c) only increase slightly, the loss in Fig. 6(b) increases by 186.7%. Fig. 7 shows the impact of LDA on different consumers in DFN topology. Only the average hop count of consumer $C3$ which is connected with the same router as the attacker $A3$ increases with φ in Fig. 7(b), while the average hop count of all consumers in Fig. 7(a) and Fig. 7(c) remains stable. Fig. 6 and Fig. 7 show that LFU and cache policy based on content popularity can alleviate the effect of LDA.

In Fig. 8 and Fig. 9, AS-3967 network is seen to have the similar results on the hit ratio loss and the average hop count, and we omit the analyses for space saving.

5.2.2 *Impact under FLA*

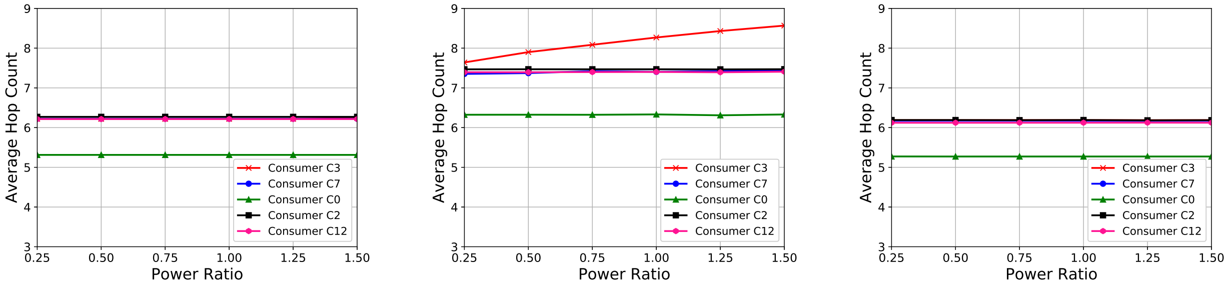
Fig. 10 shows the impact of FLA on the caching performance under different cache replacement strategies in the DFN topology. As $R5$ is connected with an attacker $A3$, compared with other routers, its hit ratio loss is the highest and increases with φ by 528.1%(LFU), 160.4%(LRU) and 325%(Popularity) respectively. The hit ratio loss of routers $R13$ increases slightly with φ because it is one hop away from the attacked router $R5$. The hit ratio loss of other routers remains low, because they are a few hops away from the attacker. In Fig. 11, only the average hop count of $C3$ increases with φ, because its router connects with an attacker $A3$ in Fig. 4. Fig. 10 and Fig. 11 show that all the cache replacement strategies cannot mitigate FLA.

Again, S-3967 network has the similar results on the hit ratio loss (Fig. 12) and the average hop count (Fig. 13), so we omit its similar analysis.

Since LFU and the popularity-based method are effective in mitigating LDA, we focus on FLA in the following experiments.

5.3 Impact of range ratio under FLA

Fig. 10, Fig. 11, Fig. 12 and Fig. 13 have proven that both the hit ratio loss and average hop count increase with the power ratio φ under FLA. In this section, we evaluate the reverse effect from FLA as the range ratio ξ varies. To

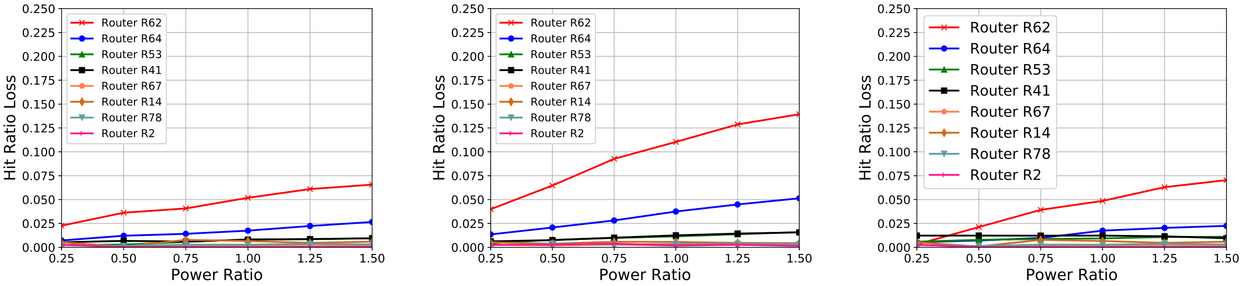


(a) Average hop count with LFU

(b) Average hop count with LRU

(c) Average hop count with Popularity

Fig. 7: Average hop count with different cache replacement policies under LDA in DFN

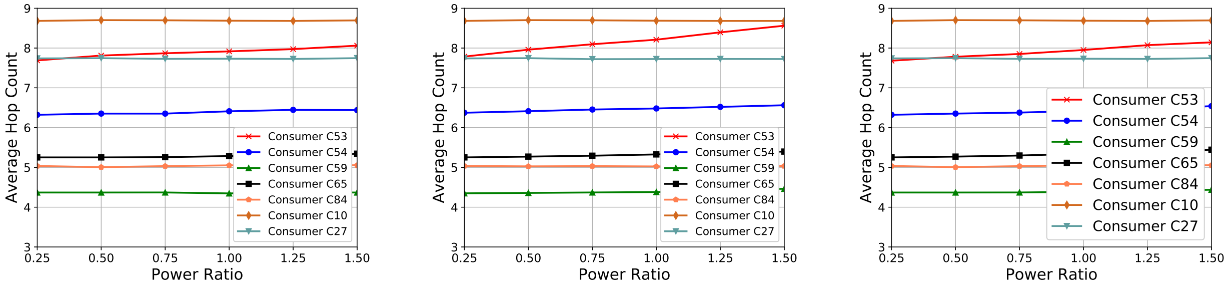


(a) Hit ratio loss with LFU

(b) Hit ratio loss with LRU

(c) Hit ratio loss with Popularity

Fig. 8: Hit ratio loss with different cache replacement policies under LDA in AS-3967

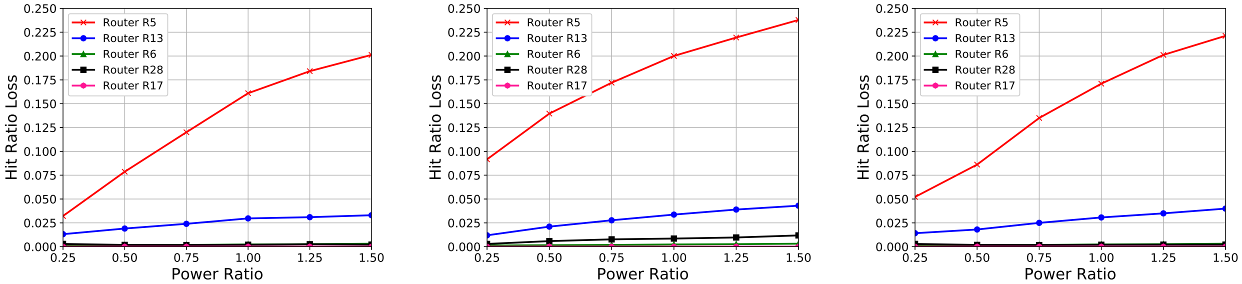


(a) Average hop count with LFU

(b) Average hop count with LRU

(c) Average hop count with Popularity

Fig. 9: Average hop count with different cache replacement policies under LDA in AS-3967



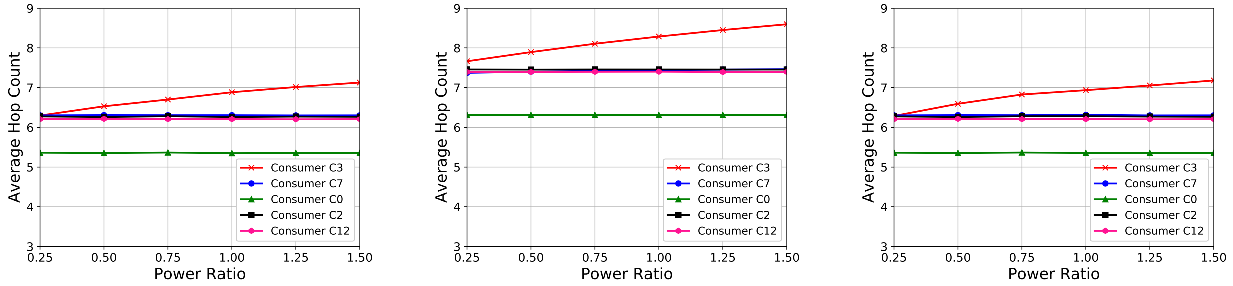
(a) Hit ratio loss with LFU

(b) Hit ratio loss with LRU

(c) Hit ratio loss with Popularity

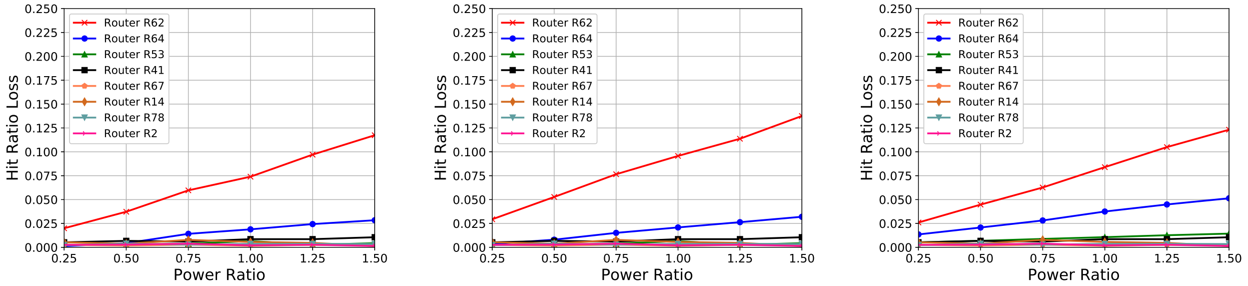
Fig. 10: Hit ratio loss with different cache replacement policies under FLA in DFN

simplify the figures, we adopt the same router notations as those in CPMH [6], upstream routers, downstream routers next to legitimate consumers, and downstream routers next to attackers. In Fig. 4, we choose $C0$, $C2$, $C3$ and $C7$ as consumers and $R6$, $R13$ and $R5$ to evaluate the hit ratio. In Fig. 5, we choose four consumers $C53$, $C54$, $C59$ and $C65$ and $R62$, $R64$ and $R53$ to evaluate the hit ratio. Two attackers, $A3$ in Fig. 4, and $A1$ in Fig. 5 are connected to the



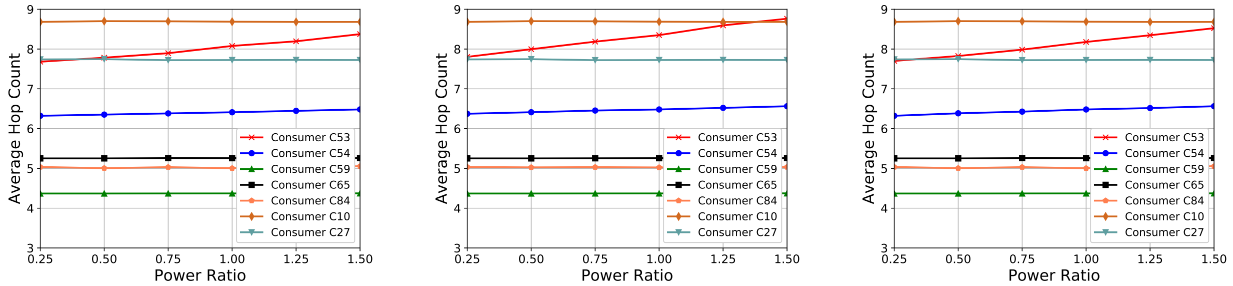
(a) Average hop count with LFU (b) Average hop count with LRU (c) Average hop count with Popularity

Fig. 11: Average hop count with different cache replacement policies under FLA in DFN



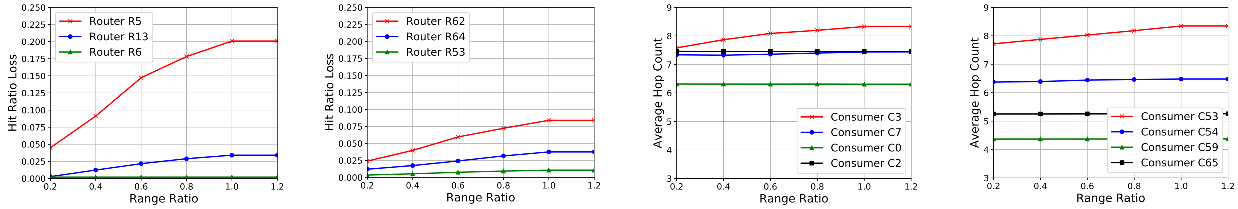
(a) Hit ratio loss with LFU (b) Hit ratio loss with LRU (c) Hit ratio loss with Popularity

Fig. 12: Hit ratio loss with different cache replacement policies under FLA in AS-3967



(a) Average hop count with LFU (b) Average hop count with LRU (c) Average hop count with Popularity

Fig. 13: Average hop count with different cache replacement policies under FLA in AS-3967

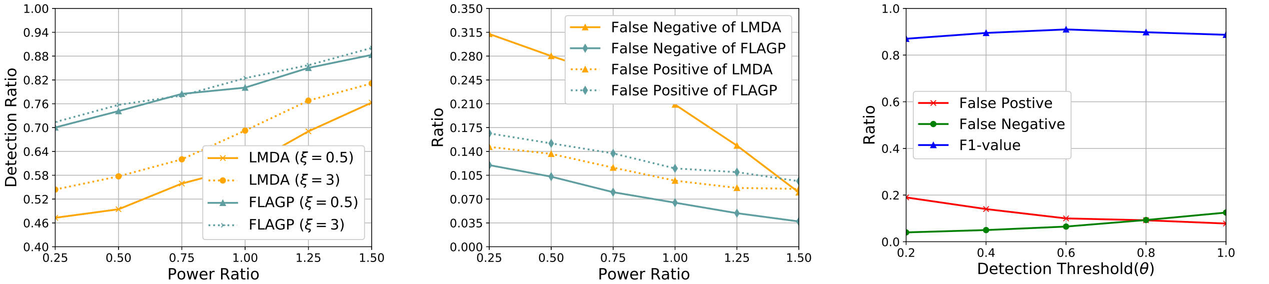


(a) Hit ratio loss in DFN (b) Hit ratio loss in AS-3967 (c) Average hop count in DFN (d) Average hop count in AS-3967

Fig. 14: Impact of Range Ratio

routers $R5$ and $R62$ respectively. The range ratio ξ varies from 0.2 to 1.2 and φ is set to be 1.

Fig. 14 shows the impact of different φ on the routers and consumers. Fig. 14(a) shows the hit ratio loss under different ξ in DFN topology. The loss of router $R6$ keeps stable, because it has a relatively large number of hops from the attacker $A3$. The loss of router $R5$ is higher than that of $R13$ because it is connected to $A3$ directly. In Fig. 14(c), only the hop count of consumer $C3$ increases with ξ because it is connected to the same router as $A3$. Similar to the power ratio φ, the range ratio only has some effect on the router connected with the attacker and the consumer connected



(a) Detection Ratio

(b) False Positive/Negative Ratio of FLAGP and LMDA with φ

(c) False Positive/Negative Ratio of FLAGP with θ

Fig. 15: Performance Comparison

with the same router as the attacker.

AS-3967 network has the similar results on the hit ratio loss in Fig. 14(b) and the average hop count in Fig. 14(d).

5.4 Performance Comparison

5.4.1 *Comparison on Detection*

Because CPMH adopts the detection of LMDA, we only compare our scheme with LMDA in terms of detection ratio, false positive ratio and false negative ratio, where we set θ to 0.6. We evaluate the false positive ratio and false negative ratio by tuning the parameter θ to change the number of suspicious requests.

Fig. 15(a) shows that the ratio of both schemes increases with the power ratio φ. Obviously, our FLAGP scheme maintains a much higher detection ratio than LMDA. The ratio of FLAGP keeps relatively stable as ξ varies, which shows FLAGP has a better and stable performance on the detection ratio in different cases. Though both schemes detect the attack events based on the statistics of Interests, our FLAGP makes use of three types of statistics (request ratio, variance of repeated interests and standard deviation of request intervals) before adopting the grey prediction. However, only the first type is used in LMDA. Our strict detection scheme brings both a higher ratio in Fig. 15(a) and a lower false negative ratio than LMDA in Fig. 15(b) as φ increases. In Fig. 15(b), it is obvious that FLAGP has a much better performance on the false negative ratio to more effectively detect FLA at the cost of slightly higher false positive ratio.

As discussed before, if the popularity computed for a received Interest is θ times of the predicted value, the file name in the Interest will be added into a suspicious list and its corresponding counter is incremented by one. As θ increases, more attacks are detected as safe events and fewer safe events are detected as attack events, causing false negative ratio to increase and false positive to decrease in Fig. 15(c). F1 value is the highest when θ is equal to 0.6. Hence θ is set to 0.6 in our later simulations.

5.4.2 *Comparison on Defense*

To evaluate the effect of defense, we compare FLAGP, LMDA and CPMH on the hit ratio loss and average hop count. As the earlier simulations have shown that FLA can have more damaging effect on the router connected with the attacker, we only make some simulations on router $R5$ in DFN topology. Similarly, we only make some simulations on router $R62$ in AS-3967 topology. Fig. 16 shows LMDA possesses the largest hit ratio loss because it does not support any defense against FLA. On $R5$ in DFN topology, CPMH maintains a hit ratio loss of 0.174 and FLAGP has a loss of 0.150 under FLA (13.7% better). On $R62$ in AS-3967 topology, CPMH maintains a hit ratio loss of 0.083 and FLAGP has a loss of 0.073 under FLA (12.04% better). Though both CPMH and FLAGP can defend against FLA, CPMH has a lower detection ratio in Fig. 15(a) by adopting the same detection method as LMDA, leading to a larger hit ratio loss in Fig. 16.

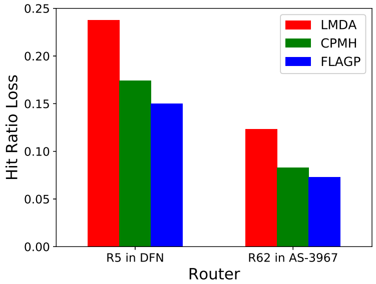


Fig. 16: Hit Ratio Loss Under Defense

TABLE 3: Average Hop Count of $C3$ in DFN and $C53$ in AS-3967 under Attack

Algorithm	Before Attack (DFN)	After Attack (DFN)	Before Attack (AS-3967)	After Attack (AS-3967)
LDMA	7.332	8.595	7.675	8.761
CPMH	7.332	7.904	7.675	8.220
FLAGP	7.332	7.762	7.675	8.117

Fig. 14(c) and Fig. 14(d) have proven that only the hop count of consumers which are connected with the same routers as attackers increase with ξ. Therefore, we only select $C3$ in DFN topology and $C53$ in AS-3967 topology to evaluate the average hop count. In Table 3, these three schemes have the same hop count under FLA if no defense is executed. When the defense is applied, LMDA without adopting any defense keeps the constant count. Our FLAGP has a lower count than CPMH because we can reduce the chance of caching non-popular data by controlling the growth of their content popularity, while CPMH identifies prefixes which are requested by attackers and puts suspicious file prefixes in a blacklist without storing the corresponding Data packets.

5.4.3 Comparison on Algorithmic Complexity

We compare the algorithmic complexity of our FLAGP and the other two state-of-the-art schemes and summarize the results in Table 4.

TABLE 4: Complexity of Algorithms

Algorithm	Detection Complexity	Defense Complexity
LDMA	$O(n)$	×
CPMH	$O(n)$	$O(n)$
FLAGP	$O(n)$	$O(n)$

LMDA detects FLA by calculating the probability variation based on the ratio of Interests for the same content object. Thus the complexity of LMDA is $O(n)$.

CPMH includes a detection phase which is the same as LMDA and also defense phase. CPMH identifies the attackers' prefixes by calculating the weighted request rate variation and store them in blacklist. The detection complexity is $O(n)$. By checking whether the content prefix in each Interest is contained in the blacklist, the router decides whether or not to cache the content. This defense process incurs a complexity of $O(n)$.

Our FLAGP detects CPA by exploiting the popularity change with grey prediction. During the detection phase, FLAGP detects Interests received and constructs a grey model based on the popularity and three traffic parameters. The complexity is $O(n)$. During the defense phase, we control the content popularity of every suspicious Interest with the complexity $O(n)$.

From the above discussion, we can obtain Table 4. LMDA has the least complexity because it only detects FLA. CPMH and FLAGP have the same complexity of detection and defense.

Complexity Evaluation: To verify the complexity, we evaluate using the experiments on processing time, CPU usage rate and memory usage. We only compare FLAGP with CPMH, because LMDA cannot provide a defense strategy and CPMH uses the same idea of LMDA in the detection phase.

The processing time is measured by the time consumption of handling 10,000 packets. As shown in Table. 5, our FLAGP scheme needs 0.1511s, CPMH needs 0.1099s. We also measure CPU usage rate and memory usage in Table. 5. We set the requesting rate to 3,000 packets per second. CPMH uses 29.7$\%$ of CPU, our FLAGP scheme uses 29.2$\%$ of CPU. As running time is longer, our FLAGP' s memory occupancy is up to more than 90.1MB, while CPMH occupies 73.7MB memory.

TABLE 5: Complexity Evaluation

Algorithm	Processing Time	CPU Usage	Memory Usage
$FLAGP$	0.1511s	29.2%	90.1MB
$CPMH$	0.1099s	29.7%	73.7MB

5.5 Comparison of Different Prediction Techniques

As discussed in **Section 4**, we adopt $GM(1,4)$ model to achieve the popularity prediction. Grey theory mainly focuses on the model uncertainty and information insufficiency. Compared with statistics model and neural network-based approach, it requires less data to run. In this section, we compare the grey model with Logistic Regression (LR) model [24] and Convolutional Neural Network (CNN) [25]. The LR model is a statistical model that is usually applied to a binary dependent variable. LR is also used in categorical prediction of a dependent variable based on its association with one or more independent (continuous or discrete) predictor variables [24]. CNN is a class of artificial neural network that uses convolutional layers to filter inputs for useful information. The convolution operation involves combining input data (feature map) with a convolution kernel (filter) to form a transformed feature map. CNN can reduce the complexity of feedback neural network effectively and is widely used in the field of pattern classification [25].

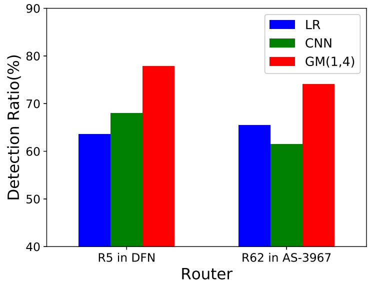


Fig. 17: Detection ratio of LR, CNN and GM(1,4)

Fig. 17 shows the detection ratio under FLA with LR, CNN and $GM(1,4)$ being used to predict the content popularity respectively, where both φ and ξ are set to 1 and Interests in the past ten slices are used. Obviously, our popularity predictive based on $GM(1,4)$ maintains the highest detection ratio whether in DFN or AS-3967.

6 Conclusion

In this paper, to alleviate the damage caused by the pollution attack in NDN network, we propose an efficient detection and defense scheme by exploiting the regularity of past Interests and popularity of each cached content. We adopt $GM(1,n)$ model to predict the future popularity of each content based on three associated series, request ratio, variance of repeated interests and standard deviation of request intervals. We detect the attack based on the popularity increment. To satisfy the quality requirement of normal uses, we further design a defense scheme by dynamically adjusting the popularity of the suspicious contents so as not to fill the cache buffer with those unpopular contents. We have performed extensive simulations to compare our scheme with several other state-of-the-art schemes, and the results demonstrate that our scheme achieves a much better performance on the hit ratio loss, FLA detection ratio, and hop count. In our future work, we plan to have a real-world implementation to further verify the performance.

Acknowledgments

This research is sponsored in part by the National Natural Science Foundation of China (contract/grant number: No.61872053) and US NSF CNS 1526843.

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *Acm Sigcomm Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[2] G. Zhang, Y. Li, and T. Lin, "Caching in information centric networking: A survey," *Computer Networks*, vol. 57, no. 16, pp. 3128–3141, 2013.

[3] A. Karami and M. Guerrero-Zapata, "An anfis-based cache replacement method for mitigating cache pollution attacks in named data networking," *Computer Networks*, vol. 80, pp. 51–65, 2015.

[4] T. Chatterjee, S. Ruj, and S. D. Bit, "Security issues in named data networks," *Computer*, vol. 51, no. 1, pp. 66–75, 2018.

[5] Z. Xu, B. Chen, N. Wang, Y. Zhang, and Z. Li, "Elda: Towards efficient and lightweight detection of cache pollution attacks in ndn," in *Local Computer Networks*, 2016, pp. 82–90.

[6] T. Kamimoto, K. Mori, S. Umeda, and Y. Ohata, "Cache protection method based on prefix hierarchy for content-oriented network," in *IEEE Consumer Communications & NETWORKING Conference*, 2016, pp. 417–422.

[7] M. Conti, P. Gasti, and M. Teoli, "A lightweight mechanism for detection of cache pollution attacks in named data networking," *Computer Networks*, vol. 57, no. 16, pp. 3178–3191, 2013.

[8] H. Guo, X. Wang, K. Chang, and Y. Tian, "Exploiting path diversity for thwarting pollution attacks in named data networking," *IEEE Transactions on Information Forensics & Security*, vol. 11, no. 9, pp. 2077–2090, 2017.

[9] H. Park, I. Widjaja, and H. Lee, "Detection of cache pollution attacks using randomness checks," in *IEEE International Conference on Communications*, 2012, pp. 1096–1100.

[10] T. L. Tien, "A research on the grey prediction model gm(1,n)," *Applied Mathematics & Computation*, vol. 218, no. 9, pp. 4903–4916, 2012.

[11] L. Deng, Y. Gao, Y. Chen, and A. Kuzmanovic, "Pollution attacks and defenses for internet caching systems," *Computer Networks the International Journal of Computer & Telecommunications Networking*, vol. 52, no. 5, pp. 935–956, 2008.

[12] Y. Gao, L. Deng, A. Kuzmanovic, and Y. Chen, "Internet cache pollution attacks and countermeasures," in *IEEE International Conference on Network Protocols, ICNP 2006, November 12-15, 2006, Santa Barbara, California, Usa*, 2006, pp. 54–64.

[13] H. Salah, M. Alfatafta, S. SayedAhmed, and T. Strufe, "Comon++: Preventing cache pollution in ndn efficiently and effectively," in *Local Computer Networks (LCN), 2017 IEEE 42nd Conference on*. IEEE (42nd IEEE Conference on Local Computer Networks, LCN), 2017, pp. 43–51.

[14] G. Zhang, J. Liu, X. Chang, and Z. Chen, "Combining popularity and locality to enhance in-network caching performance and mitigate pollution attacks in content- centric networking," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2017.

[15] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking," *Proceedings - IEEE INFOCOM*, vol. 131, no. 5, pp. 2426–2434, 2012.

[16] B. Ahlgren, C. Dannewitz, C. Imbrenda, and D. Kutscher, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.

[17] D. Ju-Long, "Control problems of grey systems," *Systems & Control Letters*, vol. 1, no. 5, pp. 288–294, 1982.

[18] E. Kayacan, B. Ulutas, and O. Kaynak, "Grey system theory-based models in time series prediction," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1784–1789, 2010.

[19] W. Zhou and J. M. He, "Generalized gm (1, 1) model and its application in forecasting of fuel production," *Applied Mathematical Modelling*, vol. 37, no. 9, pp. 6234–6243, 2013.

[20] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, Oct. 2012. [Online]. Available: http://named-data.net/techreports.html

[21] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: evidence and implications," in *INFOCOM '99. Eighteenth Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1999, pp. 126–134 vol.1.

[22] ——, "Web caching and zipf-like distributions: evidence and implications," in *INFOCOM '99. Eighteenth Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002, pp. 126–134 vol.1.

[23] S.Min, D. Lee, C. Kim, J. Choi, J. Kim, Y. Cho, and S. Noh, "Lrfu: a spectrum of policies that subsumes the least recently used and least frequently used policies," *IEEE Transactions on Computers*, vol. 50, no. 12, pp. 1352–1361, 2001.

[24] S. Menard, "Applied logistic regression analysis," *Technometrics*, vol. 38, no. 2, pp. 192–192, 2002.

[25] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

Lin Yao is a professor in DUT-RU International School of Information Science & Engineering, Dalian University of Technology (DUT), China. Her research interests include privacy of big data and security of Vanet, CCN, and Social network.

Yujie Zeng is an M.E. candidate in School of Software, Dalian University of Technology (DUT), China. His research interests include security and privacy in social network.

Xin Wang is currently an associate professor of the department of Electrical and Computer Engineering, Stony Brook University, New York State, U. S. A. Her research interests include mobile and ubiquitous computing, wireless communications and network systems, networked sensing and fusion, detection and estimation.

Ailun Chen is an M.E. candidate in School of Software, Dalian University of Technology (DUT), China. Her research interests include security and privacy in social network.

Guowei Wu is a professor in the School of Software, Dalian University of Technology (DUT), China. His research interests include smart edge computing, software defined networks, and NFV.