Triangle Area Based Multivariate Correlation Analysis for Detecting and Mitigating Cache Pollution Attacks in Named Data Networking

Muhammad Sohail
Department of Automation
University of Science and Technology of China
Hefei, China
sohailkhan@mail.ustc.edu.cn

Quan Zheng*
Department of Automation
Institute of Advanced Technology, University of Science and Technology of China
Hefei, China
qzheng@ustc.edu.cn

Zeinab Rezaiefar
School of Computing
Ulster University, United Kingdom
z.rezaiefar@ulster.ac.uk

Muhammad Alamgeer Khan
Department of Automation
University of Science and Technology of China
Hefei, China
alamgeer@mail.ustc.edu.cn

Rizwan Ullah
Department of EEIS
University of Science and Technology of China
Hefei, China
rizwanul@mail.ustc.edu.cn

Xiaobin Tan
Department of Automation
University of Science and Technology of China
Hefei, China
xbtan@ustc.edu.cn

Jian Yang
Department of Automation
University of Science and Technology of China
Hefei, China
jianyang@ustc.edu.cn

Liu Yuan
NEL-PSRPC
China Academy of Electronic and Information Technology of CETC, Bejing, China.
yuanliu@cetc.com.cn

***Abstract*— The key feature of NDN is in-network caching that every router has its cache to store data for future use, thus improve the usage of the network bandwidth and reduce the network latency. However, in-network caching increases the security risks - cache pollution attacks (CPA), which includes locality disruption (ruining the cache locality by sending random requests for unpopular contents to make them popular) and False Locality (introducing unpopular contents in the router's cache by sending requests for a set of unpopular contents). In this paper, we propose a machine learning method, named Triangle Area Based Multivariate Correlation Analysis (TAB-MCA) t at detects the cache pollution attacks in NDN. This detection system has two parts, the triangle-area-based MCA technique, and the threshold-based anomaly detection technique. The TAB-MCA technique is used to extract hidden geometrical correlations between two distinct features for all possible permutations and the threshold-based anomaly detection technique. This technique helps our model to be able to distinguish attacks from legitimate traffic records without requiring prior knowledge. Our technique detects locality disruption, false locality, and combination of the two with high accuracy. Implementation of XC-topology, the proposed method shows high efficiency in mitigating these attacks. In comparison to other ML-methods, our proposed method has a low overhead cost in mitigating CPA as it doesn't require attackers' prior knowledge. Additionally, our method can also detect non-uniform attack distributions.**

Keywords*— *Named Data Networking, False Locality, Locality Disruptions, Multivariate Correlation Analysis, In-network Caching.

I. INTRODUCTION

Recently, several approaches have been adopted to design an operable replacement for current IP-based Internet to serve today's needs in a better and efficient way [1-3]. Scalability, mobility and most importantly strong security are some of the major design features in the latest developments.

Named Data Networking (NDN) [4] is an eminent example of Content-Centric Networking (CCN) also known as Information Centric Networking (ICN). In NDN, the name identifier is used to access a data content instead of using content location. Users convey their requests by disseminating interest messages which contain names instead of IP addresses. The network can satisfy these interest messages from any nearest intervening nodes having the requested contents. Content message follow the reverse path followed by interest message. Any intervening node can save this content message for subsequent request in future efficiently. The pervasive in-network caching [5] is the crucial feature to reduce latency and usage of bandwidth. However, employing extensive caching invigorates security concerns of cache pollution attacks such as locality disruption and false locality. In locality disruption, attackers consistently send request messages for new unpopular contents to force routers (victims) to cache those requested unpopular contents, thus deteriorating the efficiency of cache by demolishing the cache file locality while in False Locality, attackers make requests for a limited set of unpopular contents to make them popular, resultantly the hit ratio is reduced.

Contribution: This paper addresses cache pollution attacks in NDN using a machine learning method. There are three main factors in this method including popularity of a content messages, time distribution of request messages and interface distribution of interest messages for a content. The employed method addresses the two major concerns including the locality-disruption and false-locality attacks. Here, some of the

smart behaviors of attackers are also taken into account which previous methods were failed to do. On the basis of these three factors, the router will determine to cache or release the content messages for thwarting cache pollution attacks.

Then the proposed method is analyzed for XC topology. Finally, proposed method in NDN environment is validate with extensive simulation. The evaluation results demonstrate its efficacy in detection, mitigation and overhead over preexisting approaches.

The remaining sections of this paper are arranged as in the following. Section 2 explains related work. TAB-MCA system architecture is given in section 3. In section 4, proposed method is introduced and discussed. The section 5 brings the evaluation results to manifest the effectuality of the proposed strategy, and finally, the conclusion is given in section 6.

II. RELATED WORK

Since NDN is employed as a new architecture to replace current IP-based internet architecture, few researchers have focused to address several kinds of attacks. A. Afanasyev et al. [8], A. Compagno et al. [9] and K. Wang et al. [10] have addressed Denial of Service attacks. A. Afanasyev et al. [11], B. Alzahrani et al. [12] and B. Alzahrani et al. [13] have addressed naming, routing and forwarding attacks respectively. Z. Rezaeifar et al. [14], D. Kim et al. [15] and C. Ghali et al. [16] have addressed cache poisoning attacks. However, this approach addresses cache pollution attacks in NDN. In cache poisoning, attackers try to cache malicious or fake contents in router's content store by sending request for malicious contents, while in cache pollution; attackers try to disrupt cache locality by sending request for unpopular contents and force routers (victims) to cache these unpopular contents in its content store. Successful attacks can reduce link utilization and cache hit ratio from normal users.

Park et al. [17] introduced an approach based on randomness checks of a matrix for the detection of locality disruption attacks in CCN. This methodology is effective when applied in simple scenario and this cannot be applicable to large CCN topologies. Moreover, the computational cost is increased when tested on multiple caching nodes.

Xie et al. [18] determined a method with an objective of enhancing cache robustness to tackle locality disruption attacks using Cache-Shield. The Cache-Shield defines a shielding function which is based on logistic function. This function decides whether the content should be kept or not, if the answer is yes CS stores this content and if answer is no CS stores content's name instead of storing the whole content. Cache-Shield should imperatively operate even in the absence of attack and accumulate more data on each router which can reduce data storage for popular contents.

Conti et al. [19] proposed a technique to detect cache pollution attacks in the NDN called as lightweight mechanism. They mentioned that Cache shielding is ineffective against some attacks, even in some situations the performance of this approach is worse than without them. In this technique, they use sampling interest distribution to detect cache pollution attacks. The method only shows how to detect cache pollution attacks but doesn't show any suppressing method to reduce the influence of attacks.

Karami et al. [20] developed a cache replacement policy called ANFIS- based cache replacement. ANFIS (adaptive neuro-fuzzy inference system) is an integration of neural network with fuzzy inference system. They use inherent properties of cached content as ANFIS inputs and output of the ANFIS is categorized into three types (i.e. false locality, Locality disruption and Healthy). ANFIS is a machine learning method having high computational overhead. They introduce a cache replacement method, which works only in the condition when CS is full and having no place for new incoming popular content.

Guo et al. [21] exploiting the diversity of the interests traversing paths to detect False Locality attacks in NDN. This method is based on the idea that interests traversing paths for malicious content objects are not as many as for legitimate contents. A request is considered as malicious and expelled if such a path diversity is not observed. In their work, Probabilistic Counting with Stochastic Averaging (PCSA) and Bloom Filter are used for path tracking.

Zhang et al. [22] introduce a non-collaborative technique for cache decision making and thereby mitigate cache pollution attacks in CCN. In their method, popularity is integrated with locality to construct decision-making matrix. The basic idea this method based on is, attackers cannot send interest messages from as many interfaces for unpopular content objects as popular content objects receive from.

Yao et al. [23] propose clustering, an unsupervised machine learning method for detection and mitigation of cache pollution attacks. Clusters are made on the basis of interest probability and average time interval between two consecutive interests for the same content object using Euclidean Distance. Based on these clusters results, attack type (i.e. FLA or LDA) is determined.

The work that is most similar to ours is [21], as we both address both types of cache pollution attacks (i.e. FLA or LDA using ML-Methods. However, there are two major differences: (1) In [21], inputs are used individually, while in our method we use correlation between every two. (2) In [21], training data for normal users as well as for attackers is also needed that enables it to detect attacks pre-defined in training data, while in our method we just need to provide information about normal users, and deviation from normal user's profile is considered as attack, that's why our method can detect any type of new attack.

After summarizing all the above discussion, the conclusion is that all the above mentioned methods are targeting cache pollution attacks in NDN. These methods have some drawbacks is given as: (1) Some of the pre-existing methods are only applicable to simple scenario while applying to complex scenario their effect is very less or zero effect. (2) Most of the pre-existing methods target either type of cache pollution attacks (LDA or FLA). (3) Almost all the pre-existing methods detect attacks based on content request distribution, which is not applicable to all types of malicious users. Our proposed method is applicable to all types of scenarios, targeting both types of CPA and detect all types of attacks.

III. TAB-MCA SYSTEM ARCHITECTURE

In this section, our proposed model for cache pollution detection and mitigation in NDN is overviewed, where framework of the proposed model is discussed.

A. Framework

There are three main steps in our proposed approach for mitigation of cache pollution attacks in NDN as shown in Fig. 1.

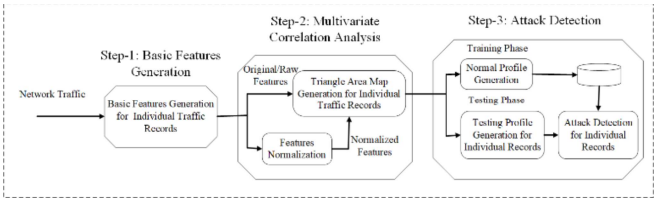

Fig. 1. TAB-MCA block diagram.

In step 1, fundamental features for individual records are extracted from the traffic entering to the internal network system where secured nodes are residing in and are used to collect network traffic records for a pre-determined time interval.

In step 2, the triangle area map (TAM) matrix is generated using network traffic records that originated from the first step. These records are used in original/raw form or after normalization by utilizing the "feature normalization" module. Each TAM contains different triangles areas, where each triangle area represents extracted correlations between two well defined features. Each traffic record has its own TAM so the total number of the TAMs is equal to the total number of the traffic records coming from step 1. The total number of triangle areas in each TAM is equal to all possible permutations of the extracted correlation between two distinct features. After obtaining TAM, we do not need actual fundamental features (non-normalized) or normalized features as they are replaced by triangle areas to represent traffic records. When network intrusions occur, it causes variations to these correlations and those variations can be exploited to determine and identify the malignant activities. This provides more accurate information that help to distinguish normal traffic records from attacks. Proposed methodology is explained in coming sections in detail.

Step 3 is the decision-making stage, which can detect any type of attack without requiring any prior knowledge about it. Two phases incorporated in decision-making are training and testing phase. The training phase constitutes the "Normal Profile (NP) generation" module to build normal profile for different legitimate traffic records, and the obtained profile is then stored in a database. In the testing phase, the "testing profile generation" module is used to generate profile for individual detected traffic records. Then, "attack detection" module is applied to compare the generated tested profile for an observing content with the already saved normal profile. A threshold value is used in the "attack detection" module to classify traffic behavior and distinguish cache pollution attacks from normal traffic. The methodology is developed in the Section 4.

IV. PROPOSED METHOD

In this portion, we explain the architecture of the proposed model (TAB-MCA) for detection and suppression of cache pollution attacks in NDN. Then, simulations are carried out to analyze the efficacy of the proposed methodology using considered most well used XC topology. The nomenclature of the proposed method is depicted in Fig. 1 followed by the detailed explanation of the proposed method. The considered network topology (XC) is shown in Fig. 2.

A. Basic Feature Generation

To detect the cache pollution attacks in NDN, the traditional method is to learn the behavior of legitimate users, and then detect anomalous behavior when such pattern changes. In our proposed method, we use three features (i.e. Popularity, Standard Deviation, Locality). The input features we considered in our proposed model are defined as follows:

i. Popularity: is the ratio of the number of requests for content "*C*" to the number of requests for all contents in a router "R_x". The popularity of content "*C*" is determined using this parameter.
ii. Standard Deviation: of the time intervals occurring between every two consecutive Interest messages for content "*C*" which helps to determine the status of requests distribution for a specific content. The standard deviation of the uniform distributions is close zero, while for non-uniform distributions e.g. Zipf –like distribution are not close to zero.
iii. Locality is the request distribution for content "*C*" throughout all the interfaces of router "R_x". The locality feature helps to define the locality-wise content request distribution. Locality of the contents requested from many interfaces decreases and close to zero, while for the contents, only requested from some specific interfaces are not close to zero.

There is a huge difference in the behavior of cache pollution attack network traffic and legitimate traffic, which is clearly depicted from its statistical behavior. In this subsection, we present MCA approach to describe these statistical properties. Triangle area is employed by this approach for extraction of the correlation between the features. The details are given in the following.

B. Multivariate Correlation Analysis

Given an arbitrary data set $X = \{x_1, x_2, x_3, \ldots, x_n\}$ where $x_i = [f_1^i\ f_2^i\ f_3^i \ldots f_m^i]^{\mathrm{T}}$, $(1 \le i \le n)$ represents the ith traffic record of the m-dimensional array. In our proposed method, we use three features (i.e. popularity, standard deviation, and locality), for which x_i vector becomes $[f_1^i\ f_2^i\ f_3^i]^{\mathrm{T}}$. We apply the triangle-area concept to extricate the correlation between jth and kth input features in the vector x_i. Data is transformed to obtain triangle between two input features. The vector x_i is firstly projected on the $(j,k)th$ 2D Euclidean subspace as $y_{i,j,k} = [\varepsilon_j\ \varepsilon_k]^T$ and $x_i = [f_j^i\ f_k^i]^{\mathrm{T}}$ where $(1 \le i \le n,\ 1 \le j \le 3,\ j \ne k)$. $\varepsilon_j = \left[e_{j,1}\ e_{j,2}\ e_{j,3}\right]^T$ and $\varepsilon_k = \left[e_{k,1}\ e_{k,2}\ e_{k,3}\right]^T$ are vectors having all elements equal to zero, excluding (j, j)th and (k, k)th elements that are equal to one in ε_j and ε_k respectively. The $y_{i,j,k}$ can be expressed as a two dimensional column vector

that can also be defined as a point on Cartesian coordinate system with coordinate $(f_j^i\ f_k^i)$ in the (j,k)th 2D Euclidean subspace. A triangle $\Delta f_j^i O f_k^i$ is formed on the Cartesian coordinate by the origin "O" and projected points $(f_j^i\ f_k^i)$ on the j-axis and k-axis of the coordinate. Area $Tr_{j,k}^i$ of the triangle $\Delta f_j^i O f_k^i$ is defined as

$$Tr_{j,k}^i = (\|(f_j^i, 0) - (0,0)\| \times (\|(0, f_k^i) - (0,0)\|)/2 \quad (1)$$

Where $1 \le i \le n,\ 1 \le j \le 3,\ j \ne k$. The $Tr_{j,k}^i$ defined in (1) can be simplified as its value becomes equal to one half of the product of the absolute values of the terms f_j^i and f_k^i. Hence, the transformation of data may be evicted, replacing (1) by $Tr_{j,k}^i = (|f_j^i| \times |f_k^i|)/2$.

To make a complete analysis, a triangle areas map (TAM) is constructed by computing triangle areas for all possible combinations of any two features in x_i vector. All the obtained triangle areas are ordered in the map (TAM) according to their indexes, e.g. the $Tr_{j,k}^i$ is located on the jth row and the kth column of the TAM having a size of 3×3. We only concern with the correlation between two well-defined features, so values of the main diagonal elements on are set to zeros ($Tr_{j,k}^i = 0,\ if\ j = k$). For nondiagonal elements $Tr_{j,k}^i$ and $Tr_{k,j}^i$ for $j \ne k$ represent the area of the same triangle which are equal. Thus, the TAM is a symmetric matrix having elements of zero values on the main diagonal. Therefore, correlations in vector x_i can be represented efficiently and appropriately by any of the two triangles upper and lower. In proposed method, we consider lower triangle, denoted as follows:

$$TAM_{lower}^i = [Tr_{2,1}^i\ Tr_{3,1}^i\ Tr_{3,2}^i]^T$$

C. Detection Technique

In this subsection, we introduce a threshold-based anomaly detection mechanism used in our method. For this, first we generate a normal profile using exclusively legitimate traffic records. For a new coming traffic record, a comparison is made with generated normal profile. If the similarity difference is greater than the predefined threshold, the incoming record is presumed as a malicious. It must be clear that normal profile and threshold have noticeable impact on the performance of a threshold-based detector.

1) Normal Profile Generation

Assume that we have a set of g legitimate traffic records, for which $X^{normal} = \{x_1^{normal}, x_2^{normal}, .., x_g^{normal}\}$. The TAB-MCA is applied to all g traffic records and the obtained set of the triangle areas map is denoted by $X_{TAM_{lower}}^{normal} = \{TAM_{lower}^{normal,1}, TAM_{lower}^{normal,2}, .., TAM_{lower}^{normal,g}\}$.

Mahalonabis distance is applied to compute the dissimilarity between all g traffic records. The precedence of Mahalonabis Distance over others like Manhattan Distance and Euclidean Distance is, it computes distance between two multivariate data objects by exploiting the correlations between features and removing the dependency on the measurements during the process of calculation.

Algorithm 1: Normal Profile Generation

Input: Training Data and $X_{TAM_{lower}}^{normal}$ for g elements

Output: NP

1. $\overline{TAM_{lower}^{normal}} \leftarrow \frac{1}{g}\sum_{i=1}^{g} TAM_{lower}^{normal,i}$
2. Compute covariance matrix C for $X_{TAM_{lower}}^{normal}$ using (5)
3. **for** i = 1 to g **do**
4. Compute $MD^{normal,i} \leftarrow MD(TAM_{lower}^{normal,i}, \overline{TAM_{lower}^{nor}})$ using (3)
5. **end for**
6. $\mu \leftarrow \frac{1}{g}\sum_{i=1}^{g} MD^{normal,i}$
7. $\sigma \leftarrow \sqrt{\left(\frac{1}{g-1}\sum_{i=1}^{g}(MD^{normal,i} - \mu)\right)^2}$
8. $NP \leftarrow (\mathrm{N}(\mu, \sigma^2), \overline{TAM_{lower}^{normal,i}}, Cov)$
9. **return** NP

Algorithm 1 shows Normal Profile (NP) generation. NP is built using density calculation of the MDs between individual legitimate training traffic record ($TAM_{lower}^{normal,i}$) and the mean of the g legitimate training traffic records ($\overline{TAM_{lower}^{normal,i}}$). MD is evaluated as follows:

$$MD^{normal,i} = \sqrt{\frac{\left(TAM_{lower}^{normal,i} - \overline{TAM_{lower}^{normal}}\right)^T \left(TAM_{lower}^{normal,i} - \overline{TAM_{lower}^{norma}}\right)}{Cov}} \quad (3)$$

$$MD^{observed} = \sqrt{\frac{\left(TAM_{lower}^{observed} - \overline{TAM_{lower}^{normal}}\right)^T \left(TAM_{lower}^{observed} - \overline{TAM_{lower}^{normal}}\right)}{Cov}} \quad (4)$$

The covariance matrix used in (3) and (4) can be obtained from the following equation:

$$Cov = \begin{bmatrix} \sigma(Tr_{2,1}^{normal}, Tr_{2,1}^{normal}) & \sigma(Tr_{2,1}^{norma}, Tr_{3,1}^{normal}) & \sigma(Tr_{2,1}^{normal}, Tr_{3,2}^{normal}) \\ \sigma(Tr_{3,1}^{norma}, Tr_{2,1}^{normal}) & \sigma(Tr_{3,1}^{normal}, Tr_{3,1}^{normal}) & \sigma(Tr_{3,1}^{normal}, Tr_{3,2}^{normal}) \\ \sigma(Tr_{3,2}^{normal}, Tr_{2,1}^{normal}) & \sigma(Tr_{3,2}^{normal}, Tr_{3,1}^{normal}) & \sigma(Tr_{3,2}^{normal}, Tr_{3,2}^{normal}) \end{bmatrix} \quad (5)$$

The covariance between two elements of the TAM_{lower}^i for normal traffic records is given below:

$$\sigma(Tr_{j,k}^{normal}, Tr_{p,q}^{normal}) = \frac{1}{g-1}\sum_{i=1}^{g}(Tr_{j,k}^{normal,i} - \mu_{Tr_{j,k}^{normal}}) \times \left(T_{p,q}^{norma,i} - \mu_{Tr_{p,q}^{normal}}\right) \quad (6)$$

The mean of the (j,k)th and (p,q)th elements of TAMs over total g legitimate traffic records are given as $\mu_{Tr_{j,k}^{norma}} = \frac{1}{g}\sum_{i=1}^{g} Tr_{j,k}^{norma,i}$ and $\mu_{Tr_{p,q}^{normal}} = \frac{1}{g}\sum_{i=1}^{g} Tr_{p,q}^{normal,i}$ respectively.

The distribution of the MDs is characterized by two parameters, mean (μ) and standard deviation (σ) of the MDs. Finally, the obtained distribution $\mathrm{N}(\mu, \sigma^2)$, $\overline{TAM_{lower}^{norma,i}}$ and Cov of the normal training traffic records are stored in the normal profile (NP) for attack detection.

2) Threshold Selection

The threshold is used to distinguish between legitimate traffic records and attack traffic records and is defined as follows:

$$Threshold = \mu \pm \alpha * \sigma \quad (7)$$

where μ and σ are the mean and standard deviation of the legitimate/normal training traffic records respectively, and α is a constant and can be selected in a specific range which can make decision with a certain level of confidence varying.

3) Attack Detection

As shown in algorithm 2, to detect cache pollution attacks, first we need to generate lower triangle ($TAM_{lower}^{observe}$) for an observed traffic record using the proposed TAB-MCA approach. The next step is to compute MD between the $TAM_{lower}^{observed}$ and the $\overline{TAM_{lower}^{normal,\iota}}$ stored in normal profile NP using (4). If the obtained MD for the observed traffic record $MD^{observed}$ is in the range of the threshold value, it is considered as a legitimate traffic, else it is considered as an attack.

Algorithm 2: Attack Detection

Input: Observed traffic record $x^{observed}$, $NP \leftarrow (\mathrm{N}(\mu, \sigma^2), \overline{TAM_{lower}^{normal,\iota}}, C)$ and constant parameter "α"

Output: Content's Nature (Normal or Attack)

1. Generate $TAM_{lower}^{observed}$ for the observed traffic record
2. Compute $MD^{observed} \leftarrow MD(TAM_{lower}^{observed}, \overline{TAM_{lower}^{normal}})$ using (4)
3. if $(\mu - \alpha * \sigma) \leq MD^{observed} \leq (\mu + \alpha * \sigma)$ then
4. return Normal
5. else
6. return Attack

D. Feature Normalization

Data normalization in machine learning is in important feature and known as features scaling. It is needed when we are dealing with the data in different scales that can cause problems like degradation in the detection performance. In our method, the above problem occur when the dissimilarity between attack and normal profile is close to that between legitimate user and normal profile.

Different methods are used for data normalization (i.e. decimal scaling, min-max normalization and z-score normalization). Proposed method uses statistical normalization technique presented by W. Wang et al. [6] that has been proven one of the best distance-based detection technique and outperforming other normalization techniques. To perform statistical normalization, mean scale and its statistical distribution of the attribute values are used.

Considering the same arbitrary data set $X = \{x_1, x_2, x_3, \ldots, x_n\}$ given in Section 3, the statistical normalization is as follows: The normalized value of feature f_j^i is given as $F_j^i = (f_j^i - \overline{f_j})/\sigma_{f_j^i}$, where $\overline{f_j} = \frac{1}{n}\sum_{i=1}^{n} f_j^i$ defines the mean of the feature f_j^i and $\sigma_{f_j^i} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_j^i - \overline{f_j})^2}$ defines the standard deviation of the feature f_j^i . Where $x_i = [F_1^i\ F_2^i\ F_3^i \ldots F_m^i]^{\mathrm{T}}$, $(1 \leq i \leq n)$ represents the ith traffic record of the normalized m-dimensional array. In our method, features are normalized in batch manner. After putting our detection system online, real-time features normalization can be achieved by using incremental learning [7]. Mean can be updated by using the formula $\overline{f_j} = \overline{f_j} + \frac{x_{n+1} - \overline{f_j}}{n+1}$.

V. EXPERIMENTAL RESULTS

This section depict the simulation environment, considered topology and experimental results of the simulations. We show through simulation that our devised TAB-CA outperform compared to pre-existing methods in many aspect i.e. accuracy, overhead cost, etc. Proposed countermeasure is checked for considered topology: XC, depicted in Fig. 2. The proposed TAB-MCA is implemented in each router of in the topology.

A. Simulation Environment

In this paper, we evaluate cache pollution attacks in Named Data Networking and the proposed countermeasure via simulations. Our simulation contingent upon open-source NS-3 ndnSIM [26] package, established at UCLA as an adjunct to NDN project. Training data was firstly simulated in to generate normal profile which is further used to get threshold values. These thresholds values are then used in ndnSIM coding to distinguish attacks from normal users.

TABLE I. SIMULATION PARAMETERS

Parameters	Symbols	Unit	Size
Ethernet link data rate	DR	Mbps	1.
Link delay	DT	ms	10
Queue length (max packets)	QL	-	10
Number of routers	NR	-	9
Number of end nodes	NN	-	9
Number of attackers	NA	-	[1,2,3]
Data size	DS	Byte	1024
Zipf parameter	α	-	0.
Zipf: total numbe of contents	NC	-	100
Simulation time	t	sec	20

The simulations are evaluated over Xie-Complex (XC) topology depicted in Fig. 2. Different symbols are used for different nodes in topology. Table.1 indicates the detail of the different parameters set of simulation for the experiment.

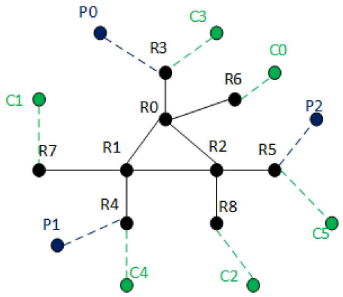

Fig. 2. Considered netwrok topology.

The legitimate consumers or we can say honest consumers mostly follow Zipf-like pattern while attacker mostly follow uniform pattern, but not all the time as we also consider smart attackers which pretend to be legitimate users. We use LRU as a cache replacement method when cache is full. We are using two types of data sets, one is training data set that contains all legitimate users and there is no attack used to generate normal profile while another one is testing data set, which is used to check the performance of the technique against malicious actions. We also set different ratio of attackers to legitimate users to check the performance of our proposed TAB-MCA method.

B. Performance Metrics

In order to assist the performance of the formulated strategy, the four metrics incorporated are described as following:

False Positive Error (FPE), False Negative Error (FNE), False Positive Error Ratio (FPER), False Negative Error Ratio (FNER), Average Number of Invalid Packets in Each Router, and Overhead of Invalid Cached Data. A brief explanation to each metric is given below:

1) False Positive Error (FPE): In the corresponding invented method, a threshold value is employed to distinguish between useable and non-useable contents; however, the analysis may be wrong sometimes due to error. Royle et al. [27] and Ortiz et al. [28] defined the term FPE that indicates that the provided criterion has been met when in reality it is still not adequate. This metric helps to measure the quality of valid packets that are ignored by the router during simulation due to error, and it is supposed that the applied condition actively detects the existence of invalid data packets.

2) False Negative Error (FNE): The term FNE indicates that the stated condition has not been satisfied but in fact it has been satisfied (Royle et al. [27], Ortiz et al. [28]). This metric helps to measure the numeric quantity of invalid content packets cached by router during simulation due to error, and it is supposed that the chosen condition readily detects valid data packets. Caching invalid content packets increase time consumption and reduce the cache-hit ratio of the compliant contents.

3) False Positive Error Ratio (FPER): FPER is a term related to FPE. FPE measures the numeric quantity of the useable content packets dropped by router due to error while FPER measures the ratio of the dismissed valid data packets to the entire number of the data packets that are received by the routers. This metric computes the occurrence of incorrect positive error in adopted methodology throughout simulation hours, therefore, the percentages of valid data packets that are dismissed is reasonable while employing this methodology.

4) False Negative Error Ratio (FNER): The metric FNER is related to FNE metric, which designate the condition is not met while in reality it was adequate. FNE measures the numeric quantity of the invalid data packets cached via router due to error while FNER measures the ratio of the accepted valid data packets to the entire number of the compliant data packets that are received by the routers. This metric computes the proportion of the incorrect negative residuals that the employed strategy encounters throughout the simulation, and it is justifying the percentages of invalid contents cached in the developed methodology.

C. Simulation Results

The simulation results considering the aforesaid performance metrics are explained in this subsection.

1) Calculating Error

This section shows different types of errors (i.e. FPE, FPER, FNE and FNER) occurred in simulation. First of all, we run our simulation for 60 seconds by setting the parameters as shown in Table. 1. The first 20 seconds of our simulation, we used as a training data set to train our algorithm and generate normal profile and the remaining 40 seconds we used as a testing data set. Fig. 3 (a) shows the FPER in training data set and Fig. 3 (b) shows FPER in testing data set. In both graphs, FPER is decreasing with increasing the value of α. Increase in α increase the threshold interval according to equation (7). The lesser value of the FPER will result greater efficiency but the problem is that it will increase the value of FNER. So to set the equilibrium and to get the best result, the value of the α needs to select carefully.

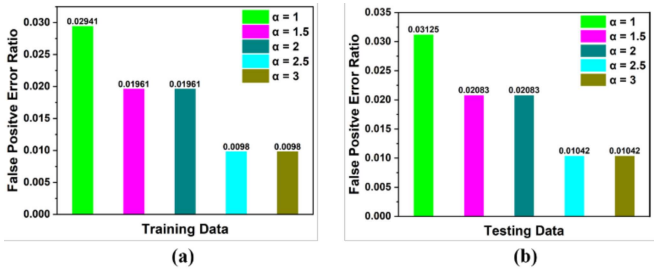

Fig. 3. Statistical Results on (a) Training Data Set (b) Testing Data Set.

a) Results and Analysis on Original Data

Fig. 4 (a) and (b) shows FPE and FPER on original/raw data respectively. All values are very small, which shows that proposed has better efficiency. Fig. 4 shows FPE and FPER for different ratios of attackers to legitimate users. The overall result is very satisfactory.

Similarly, Fig. 5 (a) and (b) shows FNE and FNER on original data for different malicious to legitimate users ratios respectively. The values for both values a little big as compare to FPE and FPER and the reason is that proposed method is dealing with smart attackers, which behaves like legitimate users and hard to find.

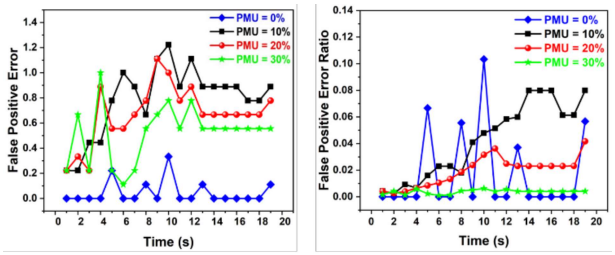

Fig. 4. For Original/Raw Data (a) False Positive Error (b) False Positive Error Ratio

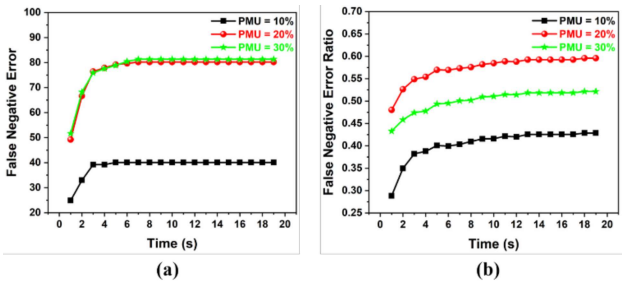

Fig. 5. For Original/Raw Data (a) False Negative Error (b) False Negative Error Ratio

b) Results and Analysis on Normalized Data

This section discusses the same errors as we discussed in the above section but the only difference is that the data used in this section is firstly normalized by statistical normalization method discussed in section 4.4. Fig. 6 (a) and (b) shows FPE and FPER and Fig. 7 (a) and (b) shows FNE and FNER for different ratios of malicious to legitimate users respectively.

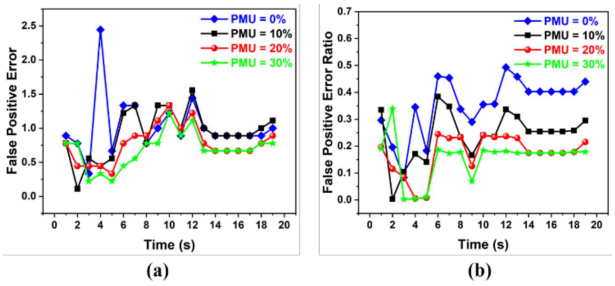

Fig. 6. For Normalized Data (a) False Positive Error (b) False Positive Error Ratio

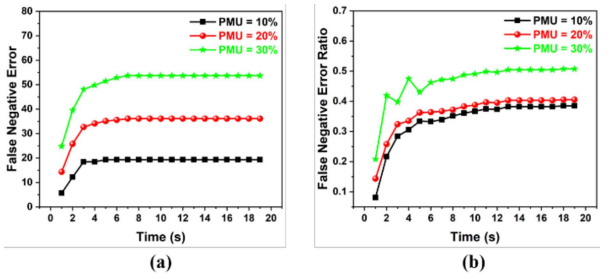

Fig. 7. For Normalized Data (a) False Negative Error (b) False Negative Error Ratio

2) Computational Complexity Analysis

This section comprised of our directed analysis upon the computational complexity and the tie consumption of our devised TAB-MCA system for attack detection.

As portrayed in section 4.2, that the triangle regions of all feasible combinations of whichever two distinguishing features in a traffic record be obliged to figured when dealing out our devised technique. Subsequently, every single traffic record has m features and for those features $\frac{m(m-1)}{2}$ triangle reg are engendered and are used to set up a TAM^{i}_{lower}. Therefore, the formulated technique has a computational complexity of $O(m)^2$. Contradictorily, as elucidated in section 4.3, the existed MD between the detected traffic record (i.e., the TAM^{i}_{lower}) and $\overline{TAM^{normal}_{lower}}$ of the particular normal profile needs to be figured in the detection system of our devised detection system to calculate the level of the heterogeneity between them. This computation sustains a complexity of $O(M)^2$, where $M = \frac{m(m-1)}{2}$ is the dimensions of TAM^{i}_{lower}. $O(M)^2$ can be written as $O(m)^4$. While considering the computational complexities of our formulated method and the detection process of our invented detection system, the inclusive computational complexity of the suggested detection system is $O(m)^2 + O(m)^4 = O(m)^4$. However, m is a fixed number, which is 3 in our case, hence the overall computational complexity is equal to $O(1)$.

VI. Conclusion and Future Work

This work has proposed TAB-MCA for cache pollution attacks detection. This detection system has two parts, the triangle-area-based MCA technique and the threshold-based anomaly detection technique. The TAB-MCA technique is used to extract hidden geometrical correlations between two well-defined features for all possible permutations. The threshold-based anomaly detection technique helps our model to be able to distinguish legitimate traffic records from attack traffic records without requiring prior knowledge. The evaluation has been performed for XC a well-known topology and the simulations gives satisfactory results.

Future work includes using of this technique for larger and more complex topologies accompanied by smarter attacks. We will test this technique using real-world data. We will also use this technique as a cache replacement method instead of caching decision method targeting both cache pollution attacks and cache poisoning attacks in NDN considering more features i.e. longevity, cache hit etc.

Acknowledgment

This work was supported by the CETC Joint Advanced Research Foundation (Grant No. 6141B08080101) and the Key R&D Plan of Anhui Province (Grant No. 202004a05020078).

References

[1] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A Survey of Green Information-Centric Networking: Research Issues and Challenges," *IEEE Communications Surveys & Tutorials,* vol. 17, pp. 1455-1472, 2015.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," presented at the

Proceedings of the 5th international conference on Emerging networking experiments and technologies, Rome, Italy, 2009.

[3] V. Sourlas, P. Flegkas, and L. Tassiulas, "A novel cache aware routing scheme for Information-Centric Networks," *Computer Networks,* vol. 59, pp. 44-61, 2014/02/11/ 2014

[4] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters*, et al.*, "Named data networking (NDN) project," 05/19 2012

[1] C. Fang, R. Yu, T. Huang, J. Liu, and J. Liu, "A Survey of Green Information-Centric Networking: Research Issues and Challenges," IEEE Communications Surveys & Tutorials, vol. 17, pp. 1-1, 07/01 2015.

[2] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, Networking named content, 2009.

[3] V. Sourlas, P. Flegkas, and L. Tassiulas, "A novel cache aware routing scheme for Information-Centric Networks," Computer Networks, vol. 59, 01/01 2013.

[4] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters, et al., "Named data networking (NDN) project," 05/19 20

[5] G. Carofiglio, M. Gallo, and L. Muscariello, "On the performance of bandwidth and storage sharing in information-centric networks," Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 57, pp. 3743-3758, 12/01 2013.

[6] Y. Kim and I. Yeom, "Performance analysis of in-network caching for content-centric networking," Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 57, pp. 2465-2482, 09/01 2013.

[7] H. Lee and A. Nakao, "User-assisted in-network caching in information-centric networking," Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 57, pp. 3142-3153 11/01 2013.

[8] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, Interest flooding attack and countermeasures in Named Data Networking, 2013.

[9] A. Compagno, M. Conti, P. Gasti, and G. Tsudik, "Poseidon: Mitigating Interest Flooding DDoS Attacks in Named Data Networking," 03/19 2013

[10] K. Wang, Z. Huachun, Y. Qin, J. Chen, and H. Zhang, Decoupling malicious Interests from Pending Interest Table to mitigate Interest Flooding Attacks, 2013.

[11] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang, "SNAMP: Secure namespace mapping to scale NDN forwarding," vol. 2015, pp. 281-286, 08/04 2015.

[12] B. Alzahrani, M. Reed, and V. Vassilakis, Enabling z-Filter updates for self-routing denial-of-service resistant capabilities, 2012.

[13] B. Alzahrani, V. Vassilakis, and M. Reed, Securing the forwarding plane in information centric networks, 2013.

[14] Z. Rezaiefar, J. Wang, and H. Oh, "A trust-based method for mitigating cache poisoning in Name Data Networking," Journal of Network and Computer Applications, vol. 104, 12/01

[15] D. Kim, S. Nam, J. Bi, and I. Yeom, Efficient Content Verification in Named Data Networking, 2015.

[16] C. Ghali, G. Tsudik, and E. Uzun, Needle in a Haystack: Mitigating Content Poisoning in Named-Data Networking, 2014.

[17] H. Park, I. Widjaja, and H. Lee, Detection of cache pollution attacks using randomness checks, 2012.

[18] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking," Proceedings - IEEE INFOCOM, pp. 2426-2434, 03/01 2

[19] M. Conti, P. Gasti, and M. Teoli, "A lightweight mechanism for detection of cache pollution attacks in Named Data Networking," Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 57, pp. 3178-3191, 11/01 2013.

[20] 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, 02/07 2

[21] 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 and Security, vol. 11, pp. 2077-2090, 2016.

[22] 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, pp. 1-1, 09/18

[23] L. Yao, Z. Fan, J. Deng, X. Fan, and G. Wu, "Detection and Defense of Cache Pollution Attacks Using Clustering in Named Data Networks," IEEE Transactions on Dependable and Secure Computing, vol. PP, pp. 1-1, 10/16 2018

[24] W. Wang, X. Zhang, S. Gombault, and S. J. Knapskog, "Attribute normalization in network intrusion detection," in 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks, 2009, pp. 448-453.

[25] D. E. Knuth, "Fundamental algorithms," 1973.

[26] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," University of California, Los Angeles, Tech. Rep, vol. 4, 2012.

[27] J. A. Royle and W. A. Link, "Generalized site occupancy models allowing for false positive and false negative errors," Ecology, vol. 87, pp. 835-841, 2006.

[28] M. Ortiz, L. Sarabia, A. Herrero, M. Sánchez, M. Sanz, M. Rueda, et al., "Capability of detection of an analytical method evaluating false positive and false negative (ISO 11843) with partial least squares," Chemometrics and intelligent laboratory systems, vol. 69, pp. 21-33, 2003.