ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 7-11 June 2020, Dublin, Ireland.
ISBN: 978-1-7281-5089-5, DOI:10.1109/ICC40277.2020.9149250

Machine Learning and Multi-dimension Features based Adaptive Intrusion Detection in ICN

Zhihao Li*, Jun Wu*, Shahid Mumtaz†, A-E M. Taha‡ ,Saba Al-Rubaye§ and Antonios Tsourdos§
* School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
† Universidade de Aveiro, Aveiro, Portugal
‡ Electrical Engineering Department Alfaisal University P.O. Box 5092 Riyadh 11533 KSA
§ School of Aerospace, Transport and Manufacturing, Cranfield University, 2717 Cranfield,
Central Bedfordshire United Kingdom of Great Britain and Northern Ireland
junwuhn@sjtu.edu.cn

***Abstract*—As a new network architecture, Information-Centric Networks (ICN) has great advantages in content distribution and can better meet our needs. But it faced with many threats unavoidably. There are four types of attack in ICN: naming related attacks, routing related attacks, caching related attacks and miscellaneous attacks. These attacks will undermine the availability of ICN, the confidentiality and privacy of data. In addition, routers store a large amount of content for the users' request, and it is necessary to protect these intermediate nodes. Since the styles of content stored in nodes are not the same, using a unified set of intrusion detection rules simply will cause a large number of false positives and false negatives. Therefore, every node should perform intrusion detection according to its own characteristics. In this paper, we propose an intrusion detection mechanism to alert for abnormal packets. We introduce a extensive solution using machine learning for attacks in ICN. Moreover, the nodes in this scheme can adapt to the external environment and intelligently detect packets. Simulation on the machine learning algorithm involved prove that the algorithm is effective and suitable for network packets.**

***Index Terms*—ICN, machine learning, defense, intrusion detection**

I. Introduction

ICN is a promising research for the next generation networks [1]. The TCP/IP network architecture has approached the limit, and the ICN is considered to be a new network architecture that can better meet the users' demands for information transmission. ICN can realize the separation of content and address, and the built-in cache of the network. Thus, it can better meet the needs of large-scale network content distribution, mobile content access, and network traffic balance. Now ICN system has been widely used in Internet of Things, 5G [2] and other network structures.

At the same time, security is the emphasis for ICN. As shown in Fig.1, ICN architecture is faced with numerous threats such as DDoS, spoofing and sniffing. But ICN is different from TCP/IP network, so that the traditional isolation and defense mechanism of the TCP/IP network architecture is not applicable to the ICN needing a new defense mechanism to ensure information security [4].

Besides, the nodes are in different external environment. If a set of unified rules is used in all nodes, a large number of false positives and false negatives would be caused. But setting the rules independently based on the characteristic of each node would be a very complicated task.

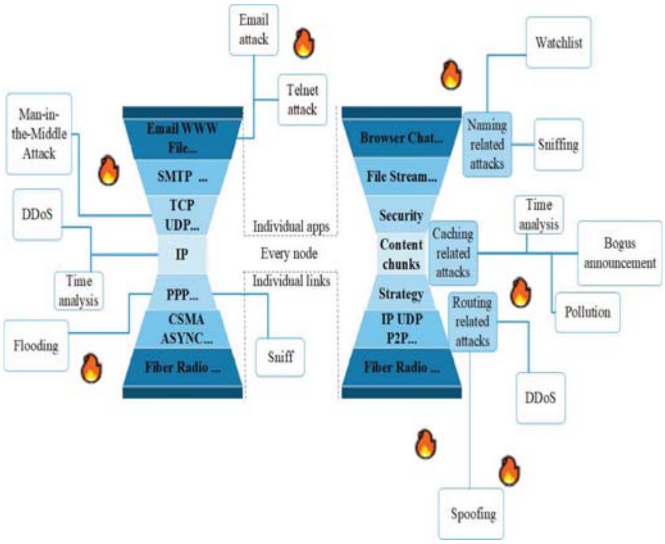

Fig. 1. The threats ICN faced

We designed an algorithm to let the nodes adapt to the environment and get a method to identify anomalous packets. On the other hand, the content-oriented ICN architecture means that there are many intermediate nodes storing and analyzing content. So its content needs to be detected. Besides, ICN nodes also receive packets that do not belong to the ICN [5]. Nodes need to be wary of these packets. We fully considered the positive role played by the ICN packet's header in the intrusion detection scheme. And most of the attacks are characterized by network traffic. So We also need to monitor the network state. These can be the basis for us to judge whether to accept the packet.

Therefore, we propose to use the clustering algorithm to perform intrusion detection in ICN nodes. The clustering algorithm is a statistical analysis method for studying classification problems. We use clustering algorithm to learn the features of historical data in ICN nodes and filter out the problematic packets. Our contributions are as follows:

- For ICN, we propose a node intrusion detection mechanism. Anomalous data packets are alerted for by machine learning.
- For the ICN packets, we select 31 valid features for machine learning. We use k-means clustering algorithm to cluster packets received by ICN nodes. And then nodes are alerted for packets with excessive particle distances.
- We enable the ICN nodes to adapt to the environment and intelligently classify packets.

The paper is organized as follows. Section II describes related researches. Internal structure in the nodes and features extraction in Section III. In Section IV, we introduce the algorithm that can be applied to ICN node intrusion detection mechanism. The mathematical model simulation is provided in Section V. Finally, we draw our conclusion in Section VI.

II. Related Work

Research on ICN security is still in beginning. An overview of attacks to the ICN architecture are provided in [6]. These attacks have been classified, which provides the reference for the extraction of packet features. Tourani [7] combined existing access control methods to discuss some security issues and privacy issues in ICN architecture. These studies indicate that there are some security issues in the ICN architecture, but they do not involve specific solutions to these security issues.

Hamdane, Balkis, S. G. E. Fatmi, and A. Serhrouchni [8] proposed a powerful security model through modifying the naming system. Through studying the semantic features of ICN, an intelligent algorithm in defense fog nodes was proposed in [9] by detecting implicit knowledge and semantic relations in packet names and content with context communication content and knowledge graph. It established a content-based intelligent defense mechanism. Kondo, Daishi [10] proposed a simple filtering technology based on Web URL statistics to alert the abnormal ICN name. These studies have proposed specific solutions to improve the security of the ICN network architecture, and have considered the semantic threat of ICN packets.

A classification-based intrusion detection system was proposed in [12], which affirmed the existence of the difference between malicious packets and normal packets. M.M. Lisehroodi, Z. Muda, W. Yassin [11] used neural network MLP and K-means clustering algorithm for intrusion detection system. But this defense mechanism was only suitable for the traditional TCP/IP network architecture. None of these studies involved intrusion detection in ICN architecture.

ICN architecture is different from TCP/IP network architecture, so we need to re-select relevant features for machine learning. Due to the difference in ICN nodes, it is not comprehensive enough to characterize packets only through semantic analysis or ICN naming system. Therefore, it is necessary to use the header features of the ICN packets, the network traffic features and the content features as the detection basis.

III. Internal structure in the nodes and features extraction

Nodes in the ICN can receive interest packets and data packets. When an interest packet arrives at the ICN node, the node searches for the content requested through matching the prefix in CS table. If it is found, the data packet is immediately returned and the interest packet is discarded. If not, it checks whether the interest packet already sent by the other node in PIT. If it is found in PIT, add the transfer face of this interest packet to the existing PIT. If not, the node find the longest matching prefix in FIB to determine the path to forward the interest packet. If it is found, the node creates an entry for the transfer face of this interest packet in PIT and forwards the interest packet. If not, the interest packet is discarded.

When the node receives the data packet, it try to find the entry in CS table. If the entry is existing, this packet would be discarded. if not, it searches for the entry in PIT. If it was made, the data packet is forwarded to the corresponding face and cached in CS. If not, the data packet is discarded.

In addition to Content Store (CS), Pending Interest Table (PIT) and Forwarding Information Base (FIB), now we need to add the features extraction layer and the machine learning layer for classifying packets received in ICN nodes.

A. Internal structure in the nodes using intrusion detection

When the ICN node receives a packet, the node first checks whether it is an interest packet or a data packet. header features extraction module extracts the packet header features. At the same time, the network monitor collects current network features. content features extraction module parses the packet to extract the content features.

All extracted features need to be preprocessed in features aggregation module and stored with the correspondence of the original packet in the database for training model or detecting packets. In machine learning module, the standardized and normalized feature vectors are summarized as machine learning input to train model. In detection phase, packet will be classified and delivered to packets processing layer.

In packets processing layer, exception management determines how to process the packet classified into abnormal packets according to the predefined policy. Other packets will be processed normally.

The processing of packets in the nodes is in Fig.2.

B. Feature Extraction

ICN nodes need to extract the features of packets as the basis for detection. Firstly, we must know which features of packets are valid. How to find out these features? we need to understand the anomalies existing in ICN.

There are four security requirements in ICN architecture: confidentiality, integrity, availability and privacy [6]. We regard the packets destroying these security requirements as abnormal packets. There are naming related attacks, routing related attacks, caching related attacks and miscellaneous attacks in ICN.

TABLE I
FEATURE LIST

No.	Features	Types	No.	Features	Types
1	packet_type	discrete	17	sum_num_samenode_diffpacket	continuous
2	interestlifetime	continuous	18	sum_num_discard	continuous
3	hoplimit	continuous	19	sum_percent_interest	continuous
4	contenttype	discrete	20	sum_num_fail	continuous
5	freshnessperiod	continuous	21	meaningless_load_size	continuous
6	time_num_same_packets	continuous	22	content_type	discrete
7	time_num_same_node	continuous	23	num_fail	continuous
8	time_num_discard	continuous	24	loggged_in	discrete
9	time_num_samepacket_diffnode	continuous	25	root_shell	discrete
10	time_num_diffpacket_samenode	continuous	26	su_attempted	discrete
11	time_num_samenode_samepacket	continuous	27	num_root	continuous
12	time_num_fail	continuous	28	num_file_creations	continuous
13	time_percent_interest	continuous	29	num_shells	continuous
14	Sum_num_node	continuous	30	num_access_files	continuous
15	Sum_num_packet	continuous	31	num_sensitive	continuous
16	Sum_num_samepacket_samenode	continuous			

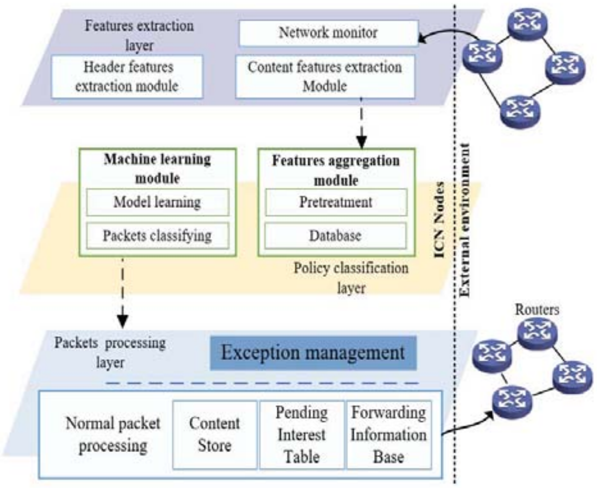

Fig. 2. Internal structure of nodes in our scheme

Routing related attacks can be classified primarily into distributed denial of service (DDoS) and spoofing attacks. The most typical attacks damaging availability are Distributed Denial of Service attacks. The attacks' goals are making the target have no resource to handle other normal packets. The features associated with DDoS attacks are primarily gathering in network traffic. We need to find out the features related to the consumption of network bandwidth resource. In general, the features about repetitively transmitting packets meet this condition.

Caching related attacks can be classified into time analysis, bogus announcements, and cache pollution attacks. Features related to these attacks are gathering in the content and semantics.

Naming related attacks can be classified into watchlist attacks and sniffing attacks. Watchlist attackers monitor the interest packets by controlling ICN nodes, and then they control the returning data packets. miscellaneous attacks can be classified into packet mistreatment, breaching signer's key and unauthorized access attacks. The attackers need to invade the ICN node in order to control the node and send malicious packets. For these attacks, the features are mainly gathering in the content of packets. And the attackers may obtain the management right of the ICN node through other portals. We also need to pay attention to the features of the header, the sensitive information and the semantics involved authorized operation in the packets. The header features related to the interest packets and the data packets header fields.

The features we extracted is in Table.1.

IV. K-means Clustering Algorithm in IDS of ICN

We characterize the packets with 31 features. The features of packet can form 31-dimensional feature vector used as a sample point. Every sample point can be regarded as a point in R^{31} space. We use Euclidean distance of points to measure the similarity between packets.

For the system, we first need to train the model. The training data for this model is the feature vectors of received packets. Then we need to set the radius parameter to limit the acceptance range. Finally, the clusters of training data are output. Accordingly we can get the range for accepting packets.

After training the model, we can use this model to detect the subsequent packets. We extracted 31 features in a packet, including 25 continuous features and 6 discrete features. We represent the feature as X_{ij} $(1 \leq i \leq n, 1 \leq j \leq 31)$.

The steps to train the model are as follows:

(1) Collect packets features as training data

(2) Standardize and normalize the training data X$[X_1, X_2...X_n]^T$. X_i is the feature vector of the packet, including 31 features X$_{ij}$. $X_i^{'}$ is the Standardized vector and $X_i^{''}$ is the normalized vector. $X^{''}$ is the set of vectors standardized and normalized.

$$X_i = [X_{i1} X_{i2} ... X_{i31}] \tag{1}$$

$$X_i{}' = \left[X_{i1}{}' X_{i2}{}' ... X_{i31}{}'\right] \tag{2}$$

$$X_i{}'' = \left[X_{i1}{}'' X_{i2}{}'' ... X_{i31}{}''\right] \tag{3}$$

$$X_{ij}{}' = \frac{X_{ij} - AVG_j}{ATAD_j} \tag{4}$$

$$AVG_j = \frac{1}{n}\left(X_{1j} + X_{2j} + ... + X_{nj}\right) \tag{5}$$

$$STAD_j = \frac{1}{n}\sum\nolimits_{i=0}^{n} \left|X_{ij} - AVG_j\right| \tag{6}$$

$$X_{ij}{}'' = \frac{X_{ij}{}' - X_{ij}{}'_{\min}}{X_{ij}{}'_{\max} - X_{ij}{}'_{\min}} \tag{7}$$

$$X'' = \begin{pmatrix} X_{11}{}'' & \cdots & X_{131}{}'' \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{n31}{}'' \end{pmatrix} \tag{8}$$

(3) Get the first vector X$_1$ as the center in cluster C$_1$

(4) Get the subsequent vector in the set X and calculate the distance between the vector and the centers of the existing cluster (C$^c_m enter$).

$$dist(X_k, C_m^{center}) = \sqrt{\sum\nolimits_{i=1}^{31}\left(X_{ki}{}'' - C_{mi}^{center''}\right)}^{2} \tag{9}$$

(5) If the distance is less than or equal to the radius parameter, classify it into the cluster (C$_m$), update the center of the cluster (C$^c_m enter$), and add members of the cluster.

$$C_{\mathrm{m}}^{center} = argmin\left\{\sum_{X_i \in C_m} dist(C_{\mathrm{m}}^{center}, X_i)\right\} \tag{10}$$

(6) If the distance is greater than the radius parameter, use this vector as the new cluster center.

(7) Input vectors until all data ends.

(8) Obtain the clustering result of the training data, input the threshold.

(9) If the ratio of the number of members in the cluster to all vectors is less than the threshold, this cluster is repealed.

Now we get the clusters of acceptable packet, We need to set the maximum radius (R$_m^{threshold}$) of every cluster to limit the range of acceptable packets.

$$R_m^{threshold} = \max\left\{dist\left(X_k, C_{\mathrm{m}}^{center}\right)\right\}, X_k \in C_m \tag{11}$$

The steps to detect the packets are as follows:

(1) Get the feature vector Y$[Y_1, Y_2...Y_n]^T$ through the feature collection module.

(2) Standardize and normalize the feature vector Y.

(3) Calculate the distance between the feature vector and every cluster center C$_m$

If dist(Y,C$_m^{center}$) $\leq$ R$_m^{threshold}$, accept it.

If dist(Y,C$_m^{center}$) $>$ R$_m^{threshold}$, the packet is processed according to the predefined exception processing policy.

The model training algorithm is display in Algorithm.1.

Algorithm 1 Model training algorithm

```
Input: Feature vector set $X$, Radius $r$, Threshold $ratio$
for $j = 1 \to 31$ do
    $AVG[j] = \frac{1}{n}\sum_{i=1}^{n} X[i][j]$
end for
for $j = 1 \to 31$ do
    $STAD[j] = \frac{1}{n}\sum_{i=1}^{n} |X[i][j] - AVG[j]|$
end for
$Min = \min\{(x[i][j] - AVG[j])/STAD[j]\}$
for $i = 1 \to n$ do
    for $j = 1 \to 31$ do
        $StandX[i][j] = (X[i][j] - AVG[j])/STAD[j]$
        $NorX[i][j] = (StandX[i][j] - Min)/(Max - Min)$
    end for
end for
$num \leftarrow 1$
$c[num++] \leftarrow NorX[1][1]$
for $i = 1 \to n$ do
    for $j = 1 \to 31$ do
        for $cluster = 1 \to num$ do
            if $dist[cluster][i][j] > r$ then
                $c[num++] = NorX$
            else
                $clu[cluster] = clu[cluster] \cup \{NorX\}$
            end if
        end for
    end for
end for
for $i = 1 \to num$ do
    if $clu[num] \geq ratio$ then
        Model gets the set $clu[num]$ , its center and its maximum radius.
    end if
end for
Output: Model
```

V. Simulation and Analysis

In this section, we used Matlab to simulate the mathematical model for testing K-means clustering algorithm applied to the detection of network packets. In Fig.3, we simulated 1500 31-dimensional feature vectors imitating the characteristics of network packets and trained them to be used as model for packets detection. We reduced the dimension of simulation result to 3D space for display.

As shown in Fig.3, the simulated vector set successfully forms the clusters. We use this model to detect packets. When the feature vector of the packet falls inside the sphere, the node

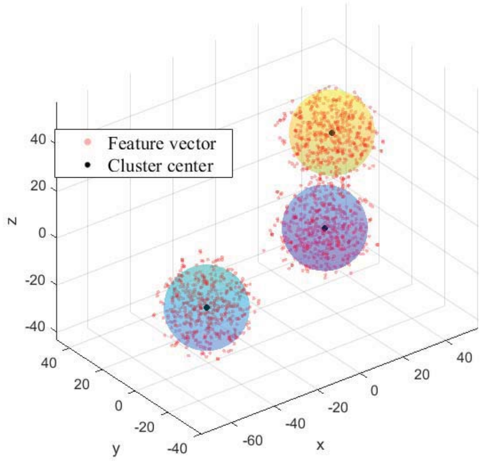

Fig. 3. Simulation of IDS in ICN using packet-like feature vectors

accepts the packet. When the distance between the feature vector and the center of the clusters exceeds the predefined maximum radius which means the gap between the packet and the normal packet is too large, we process it in other ways.

In addition, we selected 1500 random 31-dimensional vectors as the input of K-means algorithm. As shown in Fig.4, after dimension reduction, the sample points appear randomly in the 3D space, and the cluster cannot be formed effectively, so the model training fails. In order to avoid a lot of false positives, we can only expand the maximum radius. But at this time, this model can not detect packet undoubtedly.

This shows that network packets have great similarity so that we can use clustering algorithm for intrusion detection in ICN.

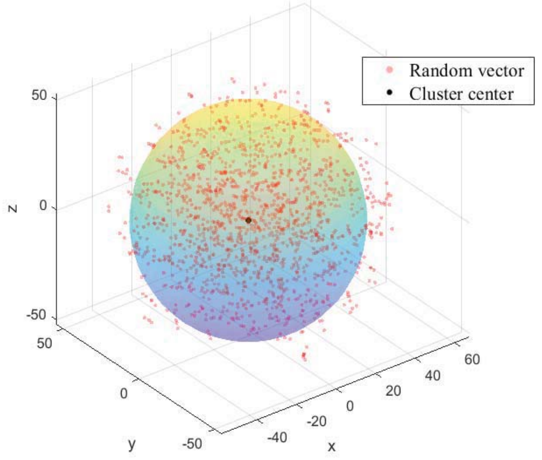

Fig. 4. Simulation in random vectors

VI. Conclusion

We use clustering algorithm to perform intrusion detection in ICN architecture. In ICN, we redefined the packet features that potentially characterize the credibility of the packet. And we simulated the mathematical model and proved that the algorithm can effectively form clusters. However, whether there is a better feature selection scheme remains to be further studied.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant 61831007, 61972255.

References

[1] Qi Li, Ravi Sandhu et al, “Mandatory Content Access Control for Privacy Protection in Information Centric Networks,” IEEE Transactions on Dependable and Secure Computing, vol. 14, no. 5, pp. 494-506, 2017

[2] L. Guo, M. Dong et al, “A Secure Mechanism for Big Data Collection in Large Scale Internet of Vehicle,” IEEE Internet of Things Journal, vol. 4, no. 2, pp. 601-610, 2017.

[3] J. Xu, K. Ota et al, “Real-time awareness scheduling for multi-media big data oriented in-memory computing,” IEEE Internet of Things Journal.

[4] Adel Djama,Badis Djamaa et al, “TCP/IP and ICN Networking Technologies for the Internet of Things: A Comparative Study,” 2019 International Conference on Networking and Advanced Systems (ICNAS)

[5] Chunmei Xia, Mingwei Xu et al, “A loss-based TCP design in ICN,” 2013 22nd Wireless and Optical Communication Conference

[6] Abdallah, Eslam, H. Hassanein et al, “A Survey of Security Attacks in Information-Centric Networking,” IEEE Communications Surveys & Tutorials (2015):1-1.

[7] Tourani, Reza et al, “Security, Privacy, and Access Control in Information-Centric Networking: A Survey,” IEEE Communications Surveys & Tutorials PP.99(2016):1-1.

[8] Hamdane, Balkis et al, “A novel name-based security mechanism for information-centric networking,” Wireless Communications & Networking Conference 2014.

[9] Yi Liu, Mianxiong Dong et al, “SCTD: Smart Reasoning Based Content Threat Defense in Semantics Knowledge Enhanced ICN,” ICC 2019.

[10] Kondo, Daishi et al, “Name anomaly detection for ICN,” IEEE International Symposium on Local & Metropolitan Area Networks IEEE, 2016.

[11] M.M. Lisehroodi, Z. Muda et al, “A HYBRID FRAMEWORK BASED ON NEURAL NETWORK MLP AND KMEANS CLUSTERING FOR INTRUSION DETECTION SYSTEM,” Proceedings of the 4th International Conference on Computing and Informatics ICOCI 2013 (p. Paper No. 020), pp. 305-311, 2013.

[12] Effendy, David Ahmad, K. Kusrini et al, “Classification of intrusion detection system (IDS) based on computer network,” 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE) IEEE, 2017.

Machine learning and multi-dimension features based adaptive intrusion detection in ICN

Li, Zhihao

2020-07-27

Li Z, Wu J, Mumtaz S, et al., (2020) Machine learning and multi-dimension features based adaptive intrusion detection in ICN. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 7-11 June 2020, Dublin, Ireland

https://doi.org/10.1109/ICC40277.2020.9149250