Hindawi
Security and Communication Networks
Volume 2020, Article ID 8860418, 12 pages
https://doi.org/10.1155/2020/8860418

Research Article

Packet-Based Intrusion Detection Using Bayesian Topic Models in Mobile Edge Computing

Xuefei Cao,[1] **Yulong Fu**,[1] **and Bo Chen**[2]

[1]*School of Cyber Engineering, Xidian University, No. 2 South Taibai Rd., Xi'an 710071, China*
[2]*National Lab of Radar Signal Process, Xidian University, No. 2 South Taibai Rd., Xi'an 710071, China*

Correspondence should be addressed to Xuefei Cao; xfcao@xidian.edu.cn

Received 6 May 2020; Revised 6 June 2020; Accepted 7 July 2020; Published 25 August 2020

Academic Editor: Xiaolong Xu

In this paper, a network intrusion detection system is proposed using Bayesian topic model latent Dirichlet allocation (LDA) for mobile edge computing (MEC). The method employs tcpdump packets and extracts multiple features from the packet headers. The tcpdump packets are transferred into documents based on the features. A topic model is trained using only attack-free traffic in order to learn the behavior patterns of normal traffic. Then, the test traffic is analyzed against the learned behavior patterns to measure the extent to which the test traffic resembles the normal traffic. A threshold is defined in the training phase as the minimum likelihood of a host. In the test phase, when a host's test traffic has a likelihood lower than the host's threshold, the traffic is labeled as an intrusion. The intrusion detection system is validated using DARPA 1999 dataset. Experiment shows that our method is suitable to protect the security of MEC.

1. Introduction

Mobile edge computing (MEC) has become the main feature of 5G communications [1]. During the development of MEC, researchers have always been keeping a focus on security issues. The security issues in MEC include application layer security, network layer security, data security, and node security. Intrusion detection systems (IDSs) protect the network layer security for MEC and have been an important component in it [2]. There are two methods to detect intrusions in general, i.e., signature-based method and anomaly-based method. The signature-based method predefines the patterns of intrusions and matches the network traffic against the patterns to raise detection alarms. While this method has low false alarm rate, it gives less than satisfactory results in detecting new types of attacks beyond the predefined patterns. The anomaly-based method establishes the normal behavior patterns for network traffic and if the pattern is accurate and extensive enough, any behavior different from the former would be regarded as an intrusion. The anomaly-based method has the ability to detect the "zero day exposure" attacks, and requires no prior knowledge of attacks. This makes the anomaly-based method superior to the signature-based method. Given the large amount of data and the diversity in services in MEC, the anomaly-based method proposes an attractive choice for MEC [3–5]. The main challenge of anomaly-based detection is how to establish an accurate and efficient behavior pattern using the normal network traffic.

There are two methods to realize the anomaly-based IDS, i.e., host-based method and network-based method [6]. In the host-based method, the network traffic to and from a single host is put together, and the host is analyzed according to the traffic. An independent behavior pattern would be established for the host's traffic. In the network-based method, however, the network traffic of all the hosts in the network is analyzed as a whole. Different hosts usually devote to different tasks, such as e-mail delivery and web page proxy, and they have different behavior patterns. Therefore, a host-based method will yield a more accurate behavior pattern compared with the network-based method [7].

LDA (latent Dirichlet allocation) is proposed by Blei et al. [8]. LDA views a document as a mixture of a series of

probabilistic topics, and each topic is a collection of related words. A document is generated by first selecting several topics and then selecting words from each topic [9]. Given a collection of documents, one can deduce the topics covered by the corpus using LDA. For example, given 5000 documents which cover different topics, LDA is able to identify what these topics are from the documents. After running LDA on the 5000 documents, one can obtain a description of these topics by providing the words used with high frequency in each topic. For one topic, the LDA model could output the words used in it as film, show, music, movie, actor, play, musical, best, and so on; for another topic, LDA could output the frequently used words as million, tax, program, budget, billion, federal, year, spending, and so on. LDA does not generate the topic name, only the words used, but we know what the topic is about by looking at the words. In the above examples, the name of the first topic could be "arts," and the name of the second topic could be "budget." Because of LDA's ability to extract topics included in a large document corpus, it could be used for text categorization, document modeling, and collaborative filtering. Furthermore, we could apply it to analyze the network traffic which is also huge in volume. The resulted topics of network traffic could be viewed as a behavior pattern of network activities. If we only provide the normal traffic to LDA, then it could generate a behavior pattern of normal traffic. Given a new session of network traffic, if it deviates from the normal behavior pattern, it is likely to be an intrusion.

Based on the above idea, in this paper, we study the problem of intrusion detection in MEC using the LDA model. The challenge is how to analyze network traffic with LDA, i.e., how to turn the traffic into "documents," and how to define the "words" in network traffic so that the resulted "topics" could represent the behavior pattern of normal network activities. We propose a method to draw analogue between network traffic and documents. A comprehensive set of network features is abstracted from tcpdump packet headers, and the network traffic is turned into documents based on these features. We also propose a method to analyze network traffic using the LDA model for intrusion detection. Our method is testified on the widely used network traffic dataset DARPA 1999, and according to the experiment results, our method could detect the intrusions effectively in MEC.

We list the main contributions of our research:

(i) We explore the usage of LDA in the anomaly-based intrusion detection systems in MEC. As far as we know, this is the first intact work of applying LDA to the intrusion detections.

(ii) We propose the method of transforming network traffic into "documents" which are required by LDA. We propose to use packets in network traffic analysis. We select 16 feature fields and use the unique values in each feature fields as "words." We also propose a method to build vocabulary list. Based on this setting of words and vocabulary list, we are able to turn network traffic into documents and process the network traffic with LDA.

(iii) We present a way to detect intrusions using LDA. A host-based method is employed. LDA is used to analyze normal traffic of a host and extract the behavior pattern of the host. Then, the host's likelihood to the behavior pattern is computed. The lowest likelihood is used as a threshold. For a new traffic, if the likelihood is lower than the threshold, it is classified as an intrusion.

(iv) We validate our method in the widely tested dataset and compare the result to the result of existing scheme. According to the comparison results, our method could detect the network intrusion with a higher detection rate.

The remainder of this article is organized as follows: Section 2 discusses the state-of-the-art research results in the field, Section 3 introduces the LDA model, Section 4 proposes our method, and Section 5 describes the experiment using our method while Section 6 concludes the paper.

2. Related Works

Intrusion detection systems could be divided into three broad categories according to the types of network traffic in use.

One form of network traffic in use is system calls. Forrest et al. pioneered in proposing using the traces of system calls to detect the possible intrusions [10]. They trained an n-gram model ($n = 3$ to 6) over the normal system calls for a given host and looked in the test data for the trace differences. Liao and Vemuri introduced the text categorization techniques to Forrest's method [11]. They employed the k-nearest neighbor classifiers to count the system call frequency to describe the normal program behavior. Then, each process is converted into a vector and the similarity between processes is calculated using the text categorization technique. To determine whether a process is normal or not, they chose k neighbors with the nearest similarity and compared the process with the k neighbors. Ding et al. used sematic analysis of system calls to extract static behavior from executable programs [12]. The static behavior is defined as the sequences of system calls and is used to detect the malicious codes. A method of deriving system call sequences is presented, and an n-gram model is used to extract the features from the system calls. Creech et al. used system calls by applying a semantic analysis to kernel level system calls and derived a new feature to classify the network activities as normal or intrusion [13]. Maggi et al. proposed a host-based intrusion detection system using system call arguments and sequences [14]. They defined a set of anomaly detection models for the individual parameters of the call and added a stage of clustering in order to better fit models to arguments. The model is complemented with Markov models to capture the correlation between system calls.

Another form of audit data is TCP/IP connection descriptions which include the summarization of high-level interactions between hosts such as session duration, type of service, number of failed login attempts, status of guest log in, and so on. Many systems first reconstruct raw network

data into connections and extract connection features before carrying out detection techniques. MADAMID [15], Bro [16], and EMERALD [17] are systems of this kind. These systems analyze the TCP/IP connections to abstract the behavior patterns of normal traffic and then detect the intrusions based on the behavior patterns. Stolfo et al. participated in the 1998 DARPA Lincoln Lab intrusion detection evaluation program [18]. Their project proposed an intrusion detection system and applied it to the DARPA 1998 dataset. They abstracted TCP/IP connections from DARPA 1998 tcpdump packets and then applied the data mining technique to the TCP/IP connections to obtain different features. They built specialized models using these features. The outputs of the models were the rules with which a classifier was trained to make final classification to a new connection. To remove the burden of transforming tcpdump packets into TCP/IP connections, the KDD99 dataset [19] was proposed. It is a revised version of the DARPA 1998 dataset [20] in which raw network traffic was summarized into TCP/IP connections where each connection is expressed by a set of network features. Various machine learning techniques have been applied to the dataset [19] and shown their effectiveness, for example, Naive Bayesian [21–24], nearest neighbor [25–29], neural networks [30–33], and fuzzy logic [34, 35].

The third form of network traffic in use is tcpdump packets. In some attacks, certain packet feature fields or payloads always employ less common values in order to launch successful attacks. Therefore, by analyzing the values of certain packet feature field, one can construct an effective intrusion detection system. One example is the firewall system: it secures the system by blocking the packets to certain ports or hosts. Recent research studies propose an improved method by building sophisticated models and combining more packet features to gain better detection results. The research in this line was first started by Mahoney, who proposed PHAD (packet header anomaly detector) by modeling more than 30 packet features and computed the abnormal score over the selected features. Attacks were detected based on the abnormal scores of a packet [36]. NETAD is an improvement of PHAD which is also proposed by Mahoney [37]. NETAD deleted the most notable packets including a connection's beginning and ending packets, and then it abstracted features from the first 48 bytes of a packet and modeled the protocol behavior accordingly. Scheme in [38] also used tcpdump packets and applied genetic algorithm to tcpdump packets. Reference [39] used similar packet feature fields as PHAD does [36], and it constructed a network behavior model for every protocol adopted in the traffic. Yassin et al. proposed a host-based PHAD [40]. They scored the packet features and performed the division of normal and abnormal using linear regression and Cohen's d measurement. Hareesh et al. [41] detected network attacks and worms by analyzing the packet header and payload. The research generated histogram for different IP header values, TCP flags, and payload. The histogram was used to represent the number of flows associated to a feature in a certain time. Then, data mining was employed to establish the normal behavior pattern given these histograms. Manandhar and Aung [42] analyzed the tcpdump packets but with a session-based method involving more packets to detect the high-level attacks.

There have been continuous efforts to apply the LDA model to the analysis of network traffic and cyber data. Cramer and Carin[43] studied the patterns of the network traffic in a corporation environment using LDA. They discovered the pattern differences in network usage between daytime and nighttime. Ferragut et al. [44] proposed several constructions of anomaly detectors in LDA's framework and noticed several abnormalities in a laboratory network. Huang et al. proposed an idea to analyze network traffic using LDA [45]. They suggested that network event can be regarded as "vocabulary," and a collection of events a user has done in a given time can be regarded as "document." They showed the possibility to detect network intrusions using the LDA model, but no detailed scheme was given. Steinhauer et al. proposed an anomaly detection system using LDA for telecommunication system [46]. They discussed the possibility of introducing LDA to the analysis of telecommunication network traffic. It turned out that the topics learned by LDA conformed to the telecommunication activities. But the proposal depended heavily on telecommunication experts to explain the result of the LDA model. Lee et al. proposed LARGen, an LDA-based automatic rule generation tool for signature-based intrusion detection systems [47]. They used LDA to analyze the network traffic and extracted the key content and signatures of malicious traffic. Then, IDS rules were built upon the signatures. They tested their method on some real network traffic.

3. Topic Model

Latent Dirichlet allocation is a statistical Bayesian topic model which could be used to infer the latent semantics of a set of documents. The LDA model is constructed under a basic assumption that the observed documents are yielded with a set of topics which are the probabilistic distributions over words. Each document is generated by first selecting the topics for the document and then selecting words from every topic [48].

The notations used in LDA are defined as follows:

(i) The vocabulary is a vector V, which is a collection of all the words used in the corpus. The length of the vocabulary is denoted as $|V|$. LDA treats a document using the concept of *bag-of-words*, i.e., a document is expressed using the predefined vocabulary and the times each word in the vocabulary appearing in the document; however, the order of words in the document is not considered.

(ii) There are D documents in all in the corpus. The d-th document in the corpus is expressed by $\mathbf{w}_d$ with $1 \le d \le D$. $\mathbf{w}_d$ is a vector of the size $1 \times |V|$, and each element in $\mathbf{w}_d$ is the times a word in the vocabulary appears in the document.

(iii) The corpus is $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_D\}$.

(iv) LDA assumes that the corpus $\mathbf{w}$ is generated by K latent topics. θ_d follows a Dirichlet distribution

with prior parameter α, and it denotes the topic distribution for the d-th document. θ_d is a row vector with K columns (a $1 \times K$ vector) with all elements in θ_d adding up to 1 and with the i-th element in θ_d representing the portion of the i-th topic in $\mathbf{w}_d$.

(v) $\theta = \{\theta_1, \theta_2, \ldots, \theta_D\}$ is a $D \times K$ matrix and denotes the topic distribution of $\mathbf{w}$.

(vi) A topic is presented by ϕ_i, which is a $|V| \times 1$ vector denoting the word distribution over vocabulary of the i-th topic. ϕ_i also follows a Dirichlet distribution with prior parameter β. All elements of ϕ_i add up to 1.

(vii) K topics $\{\phi_1, \phi_2, \ldots, \phi_K\}$ are used for $\mathbf{w}$. We define Φ as the topic distribution of the corpus $\mathbf{w}$. $\Phi = \{\phi_1 \phi_2 \cdots \phi_K\}$ is a $|V| \times K$ matrix.

(viii) A word of a document is expressed by $w_{d,n}$ where $1 \le d \le D$, $1 \le n \le N_d$. N_d means that there are N_d words in the document $\mathbf{w}_d$. $z_{d,n}$ is the topic label for the n-th word in the d-th document $w_{d,n}$ and $z_{d,n} \in \{1, \ldots, K\}$. $z_{d,n}$ follows a multinomial distribution with θ_d as its prior parameter.

Table 1 provides a summarization of the notations.

Figure 1 shows the generative process of LDA.

In the figure, α is the hyperparameter (prior parameter) for θ and β is the hyperparameter for Φ. For each document in a corpus, a topic distribution θ_d is drawn based the hyperparameter α, and this process is repeated for D times. For each word in a document, a topic label is drawn based on the topic distribution, and the draw of a topic label $z_{d,n}$ is repeated N_d times for every word in the d-th document.

The distributions of variables in Figure 1 are as follows:

$$\begin{aligned} \theta_d &\sim \mathrm{Dir}(\alpha), \\ z_{d,n} &\sim \mathrm{Multinomial}(\theta_d), \\ \phi_i &\sim \mathrm{Dir}(\beta), \\ w_{d,n} &\sim \mathrm{Multinomial}\left(\phi_{z_{d,n}}\right). \end{aligned} \tag{1}$$

In the distributions above, " $\sim$ " means "follows," $\mathrm{Dir}(x)$ means a Dirichlet distribution with hyperparameter x, and $\mathrm{Multinomial}(y)$ means a multinomial distribution with hyperparameter y.

Table 2 summarizes the parameters used in the LDA model.

$\mathbf{w}$ is an observable variable and is shown in gray in Figure 1; θ, Φ, and $\mathbf{z}$ are latent variables shown in white. LDA's goal is to infer θ and Φ. Variational Bayes and Gibbs sampling are two effective methods to do the inference [8, 9]. Gibbs sampling is a typical method to do the inference of hierarchical Dirichlet structures, and it is able to calculate the exact conditional posterior; in this paper, we use Gibbs sampling instead of variational Bayes. The performance of Gibbs sampling is slightly better than that of the variational Bayes but the speed is a little slower.

Gibbs sampling works on the idea that the topic label of a particular word is determined by the topic labels of all the other words in the corpus. Given the observed corpus $\mathbf{w}$, Gibbs sampling first calculates $\mathbf{z}$ according to the conditional posterior of $\mathbf{z}$ and then calculates θ and Φ according to the distribution of $\mathbf{z}$. We describe this process as follows:

(i) Draw θ_d for $d = \{1, \ldots, D\}$ using Dirichlet distribution with hyperparameter α. Draw ϕ_i for $i = \{1, \ldots, K\}$ using Dirichlet distribution with hyperparameter β. These are the initial values of θ and Φ.

(ii) Compute the conditional posterior of $z_{d,n}$ given the word $w_{d,n}$. The conditional posterior $p(z_{d,n} = j \mid w_{d,n}, \alpha, \beta)$ equals to $\phi_{w_{d,n},j} \times \theta_{d,j} \times c$ where c is a coefficient.

(iii) Draw $z_{d,n}$ according to the conditional posterior $p(z_{d,n} \mid w_{d,n}, \alpha, \beta)$.

(iv) Update the distribution of $z_{d,n}$ for every word in the d-th document accordingly.

(v) Calculate the conditional posterior of θ_d according to the Dirichlet distribution, but the hyperparameter should consider the distribution of $z_{d,n}$ in the d-th document.

(vi) After all documents have been processed, calculate the conditional posterior of ϕ_i according to the Dirichlet distribution, but the hyperparameter should consider the distribution of $z_{d,n} = i$ in $\mathbf{w}$.

The inferences of $z_{d,n}$, θ_d, and ϕ_i are as follows:

$$\begin{aligned} p\left(z_{d,n} = j \mid w_{d,n}, \alpha, \beta\right) &\propto \theta_{dj} \cdot \phi_{w_{d,n} j}, \\ p\left(\theta_d \mid \mathbf{w}_d, \alpha, \beta\right) &= \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma\left(\alpha_i\right)} \cdot \prod_{i=1}^{K} \theta_{di}^{n(d,i)+\alpha_i-1}, \\ p\left(\phi_i \mid \mathbf{w}, \alpha, \beta\right) &= \frac{\Gamma\left(\sum_{j=1}^{V} \beta_j\right)}{\prod_{j=1}^{V} \Gamma\left(\beta_j\right)} \cdot \prod_{j=1}^{V} \phi_{ij}^{n(\cdot,i,j)+\beta_j-1}, \end{aligned} \tag{2}$$

where $\Gamma(n) = (n-1)!$, $n(\cdot, i, j)$ is the number of word j assigned to topic i in the corpus, and $n(d, i)$ is the number of words in document d assigned to topic i.

4. Our Method

In this section, we propose our packet-based intrusion detection method using LDA for MEC. We first summarize how to generate documents from tcpdump packets and then describe our method of intrusion detection using LDA in detail.

4.1. Data Preprocessing. We consider the problem of how to transfer network traffic into documents that LDA model can handle. This procedure should be carried out before we can use the LDA model.

The network traffic we use is tcpdump packets because they help us to turn network traffic into documents easily. To turn the network traffic into documents, we should first construct the vocabulary list. We use a host-based intrusion

Table 1: Notations used.

Notation	Description
D	Number of documents
K	Number of topics
N_d	Number of words in the d-th document
V	Vocabulary list
$\mathbf{w}$	The corpus of documents
$\mathbf{w}_d$	The d-th document
$w_{d,n}$	The n-th word in the d-th document
θ	A $D \times K$ matrix, the topic distribution of the whole corpus
θ_i	A $1 \times K$ vector, the topic distribution of the i-th document
Φ	A $\lvert V \rvert \times K$ matrix; column i denotes the topic-word distribution of the i-th topic
ϕ_i	A $\lvert V \rvert \times 1$ vector, the word distribution of the i-th topic
$z_{d,n}$	The topic label of $w_{d,n}$, $z_{d,n} \in \{1, \ldots, K\}$

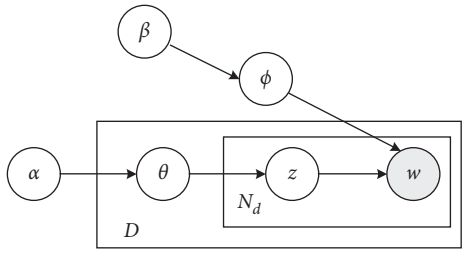

Figure 1: Illustration of the LDA model.

detection method, so we build an independent vocabulary list for each host. A tcpdump packet is composed of packet header and packet payload, and it could be grasped and analyzed by network sniff tools such as Wireshark [49]. In the packet header of a tcpdump packet, many feature fields are well defined such as MAC address, IP address, TCP service port, and so on. The format of packet header and feature fields is defined by corresponding IETF specifications. For example, RFC 791 [50] defines the format of IP header and RFC 793 [51] defines the TCP header. For most packets, the first packet header is the Ethernet header, followed by an IP header. According to the IETF specification RFC 791 [50], the IP header has a length of 20 bytes, and the 13th to the 16th byte is the source address sending out the packet. Since the tcpdump packets have such a well-defined format, we can make use of it. The values in the feature fields of a packet header can be treated as words, and the vocabulary list is the collection of all possible feature values. To define the vocabulary list suitable for intrusion detections, we select 16 feature fields from the packet header. According to available research result, these feature fields are used widely in IDS [36] and are shown in Table 3. In Table 3, the content in the bracket is the abbreviation name of the feature field. 16 feature fields are selected for a packet, and thus 16 words are generated from one packet. Each word is a combination of the feature field abbreviation name and the feature value.

Here, we give an example. A tcpdump packet is shown in Figure 2 using Wireshark. The packet is 79 B in length. The first 6 bytes indicate the Ethernet destination address, and the 7–12 bytes indicate the Ethernet source address. Thus, we can have two words EDST_00000c and ESRC_006097 according to the definitions in Table 3. The Ethernet header is followed by an IP header, the 16th byte is the type of service (0x10), and 17th-18th bytes are the IP packet length (0x0041 in hex and 65 in decimal), and thus we have two words TOS_10 and IL_65. Using the same method, we can abstract 16 words from the packet in all, and they are EDST_00000c, ESRC_006097, TOS_10, IL_65, FF_4000, TTL_40, SRC_192.168.1.30, DST_192.168.0.20, SP_21, TF_18, TU_0, TC_ffff, TO_Null, UC_Null, IC_Null, and PS_79.

However, to generate vocabulary list, we need further considerations. Since in our scheme, we use only normal traffic in the generation of vocabulary list, it may not cover all the feature values. Attacks usually employ feature values which are not covered in the normal traffic. To deal with these values, 16 extra words are added to vocabulary list to cover the features which do not appear in normal traffic, but could appear in attacks (or the test phase). For each feature field, we add an extra value, and this extra value is expressed by the combination of the field's abbreviation name and "others." For example, for the IP source field, the extra value is SRC_others. We use SRC_others to cover all the IP source addresses which do not appear in the normal traffic, but appear in the test phase. Therefore, the resulting vocabulary is all the unique feature values appearing in the traffic adding the 16 "_others" values. Let $F_i\, i \in \{1, 2, \ldots, 16\}$ denote the number of different features appearing in the i-th feature field; then, $\lvert V \rvert = F_1 + F_2 + \cdots + F_{16} + 16$ is the length of the vocabulary list.

Given the vocabulary list, we could transfer the traffic into documents. We view the tcpdump traffic in a given time length, for example, five minutes, as a document. We count which words are used in the document and count the times a word is used. A document is expressed as the times of words in vocabulary appearing in the document.

We also list the meaning of notations when we turn network traffic into documents in Table 4.

4.2. Intrusion Detection Using LDA. Given the documents transformed, we use an anomaly-based method to detect intrusions using LDA. Since LDA is able to extract the latent semantics of a corpus, we use it to abstract the latent behavior structure of network traffic. We train the LDA model

Table 2: Definitions of parameters.

Parameter	Definition
θ	Topic distribution of documents; follows a Dirichlet distribution with hyperparameter α
α	Hyperparameter of θ
Φ	Topic distribution over vocabulary; follows a Dirichlet distribution with hyperparameter β
β	Hyperparameter of Φ
z	Topic label of a word; follows a multinomial distribution with hyperparameter θ

Table 3: Feature list.

Packet layer	Features extracted
Ethernet layer	Higher 3 bytes of MAC source (ESRC), higher 3 bytes of MAC destination (EDST)
IP layer	IP length (IL), type of service (TOS), fragment flags (FF), time to live (TTL), IP source (SRC), IP destination (DST)
Transport/control layer	TCP flag (TF), TCP checksum (TC), TCP URG pointer (TU), TCP option (TO), UDP checksum (UC), ICMP checksum (IC), Service port (SP)
Others	Packet size (PS)

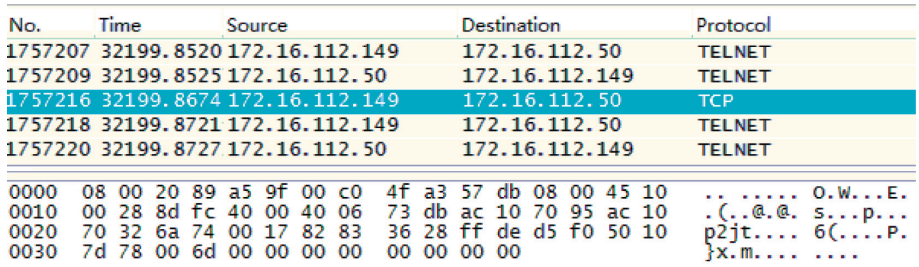

Figure 2: Example of a tcpdump packet.

Table 4: Meaning of notations in network traffic.

Notation	Description
D	The number of five-minute tcpdump sessions
K	The number of normal behavior patterns
$\mathbf{w}_d$	The d-th five-minute tcpdump session
V	Unique features values in the traffic
$w_{d,n}$	The n-th feature in the d-th session (abstracted from the $\lceil n/16 \rceil$-th packet in the session)

with only normal traffic. After running LDA on the training traffic which contains only normal traffic, we can automatically obtain the inference of latent variables θ and Φ. Since our method uses only normal traffic, Φ in fact describes what correct behaviors look like. It summarizes the features that should be included in a normal traffic behavior. For example, a topic-word distribution ϕ_i for a host 172.16.112.100 is TO_null, FF_0000, SP_53, and DST_172.16.112.20, and thus the normal traffic pattern for the host 172.16.112.100 related to ϕ_i could be the connection with the host 172.16.112.20 using UDP protocol (service port 53). The topic distribution of a document θ_i is the behavior structure distribution of a given session of network traffic. It describes what kind of behaviors are included in this network traffic.

To raise the intrusion alarm, we employ the document likelihood. It can be explained as the degree of how much a document looks like the normal behavior structure. We use the lowest likelihood of a host in the training phase as the threshold. A test document is labeled as abnormal and an alarm is raised if the likelihood of the test documents is lower than the threshold. In our method, every host has its own threshold, and the threshold is the minimum of the likelihood of all the host's training documents.

The likelihood of a document is computed using the following equation:

$$\text{lik}_d = \frac{1}{N_d}\prod_{n=1}^{N_d}\sum_{j=1}^{K} p\left(w_{d,n} \mid z_{d,n} = j, \phi_j\right) \cdot p\left(z_{d,n} = j \mid \theta_d\right)$$
$$= \frac{1}{N_d}\prod_{n=1}^{N_d}\sum_{j=1}^{K} \phi_{w_{d,n},j}\theta_{dj}. \quad (3)$$

To sum up, our method comprises four modules.

(i) Vocabulary list of a host is built based on the host's attack-free tcpdump packets data during a long enough time. Each packet is denoted by 16 features, and the anomalies of each feature field are collected. The vocabulary is the collection of all the anomalies in the 16 feature fields plus 1 extra word for each feature field.

(ii) Traffic is separated by host. A host's traffic is divided into segments, and each segment contains five minutes of tcpdump packets. A segment is transformed into a document by calculating which features are used in the segment and how many times every feature is used.

(iii) Train a LDA model for every host using the host's attack-free network traffic (training traffic) to compute Φ of the host. Use equation (3) to compute the likelihood of every training document of the host with Φ and θ_d. Set the minimal likelihood as the host's threshold.

(iv) In the test phase, according to Φ computed in the training phase and θ_d computed in the test phase, we compute the likelihood of every test document. The test document will be labeled as an attack if its likelihood is lower than the threshold.

5. Experimental Results

We implement our packet-based intrusion detection method using LDA described in Section 4 in this section. We describe the dataset used, data preprocess procedure, the training phase, the test phase, and the results.

5.1. Dataset Description. The network traffic used in this session is DARPA 1999 dataset of MIT Lincoln Laboratory which was prepared for 1999 DARPA intrusion detection evaluation program [52]. It is one of the most popular experimental datasets for network intrusion detection systems. Although it has many limitations such as the simplicity of the attacks, inaccuracy in the information, and so on, it is still used as the benchmark of many IDSs and provides a baseline to compare the performance of different IDSs.

The DARPA 1999 dataset provides a standard set of extensively gathered audit data, which comprises rich types of intrusions simulated in a military network environment. In the dataset, there are three weeks of training data and two weeks of test data. In the three weeks of training dataset, different types of data are provided, including the tcpdump data and audit data. There is no attack in Week 1 and Week 3 training traffic, and in Week 2 training traffic, there are attacks whose information is provided by the dataset. There are two weeks of test traffic, in which 201 attacks are provided and the attacks cover all four attack categories of 56 different types. The four attack categories include DOS, denial-of-service, e.g., Neptune; R2L, remote-to-local, unauthorized access from a remote machine to local machine, e.g., guessing password; U2R, user-to-root, unauthorized access to local superuser (root) privileges, e.g., eject; and probing, illegal scanning of service port, e.g., ipsweep. The ground truth of all the attacks in the test datasets is provided in an individual file.

In our experiment, we use the third week's 8-day tcpdump traffic (Mar 15–Mar 19 with three extra days) in the training phase. We use the inside.tcpdump data which are the data collected in the internal network. In the test phase, we use two weeks of test data (Mar 29–Apr 2 and Apr 5–Apr 9). Also, the inside tcpdump data are employed.

5.2. Data Preprocess Procedure. DARPA 1999 dataset is the traffic of an military network including multiple hosts. Our intrusion detection system is host-based; thus, in the preprocess of the data, we first divide the traffic according to the host addresses. 18 hosts produced the training and test traffic; therefore, we divide the network traffic according to the hosts. Figure 3 illustrates how the network traffic is divided in the training traffic, and every column in the figure represents a host. The test traffic is also divided using the same way.

For every host, we generate its own vocabulary list. The host's vocabulary list is generated using the host's training dataset. There are 18 vocabulary lists in all. The vocabulary is generated using the method described in Section 4.1. Take the vocabulary generation for the host 172.16.112.50 (Pascal) as an example. Table 5 illustrates the unique features employed by Pascal in the training phase. Table 6 illustrates the resulted vocabulary of Pascal. The size of vocabulary list for Pascal is 2788.

To convert the network traffic into documents, we divide the traffic of each host into five-minute sessions. The first packet's arriving time in the session is at most 300 seconds earlier than that of the last packet in the session. Such a time slot is chosen because we want to make the time slot large enough to cover a whole attack. Then, we calculate how many times every word is used in the session, and the session is turned into a document.

To illustrate how a document is generated, we assume a simplified session of Pascal with 200 packets. The session's feature distributions are shown in Table 7. Based on the vocabulary list of Pascal, the resulted document is a vector with the size of 1×2788. In the vector, the {1, 2, 8, 9, 15, 20, 1348, 1350, 1359, 1363, 1365, 1394, 1396, 1425, 1442, 1452, 1455, 1458, 1461, 1465, 1466, 1467, 2788} digits are set as {100, 100, 100, 100, 200, 199, 1, 200, 200, 100, 100, 100, 100, 200, 200, 200, 200, 200, 200, 199, 1, 199, 1}, respectively. Note that this session should come from a test session because it contains IL_others, IC_others, and PS_others values.

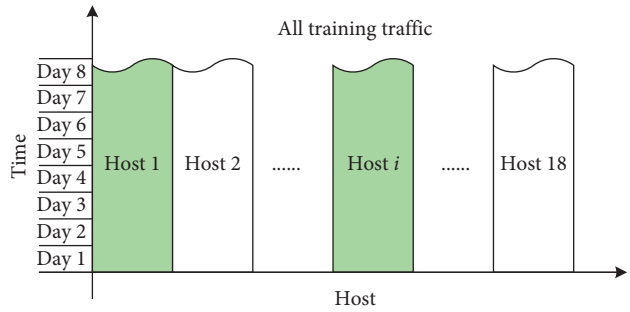

Figure 3: Example of data separation for training traffic.

Table 5: Feature anomalies of Pascal.

Field	Unique feature values	Anomaly #
EDST	0x0000c0, 0x00105a, 0x00107b, . . .	6
ESRC	0x0000c0, 0x00105a, 0x00107b, . . .	5
TOS	Null, 0x00, 0x08, 0x10	4
IL	Null, 38, 40–170, 172–217, . . ., 1489–1500	1329
FF	Null, 0x0000, 0x4000	3
TTL	Null, 60, 63, 64, 128, 254, 255	7
SRC	135.8.60.182, 135.13.216.191, 172.16.112.10 172.16.112.20, 172.16.112.50, . . .	29
DST	135.8.60.182, 135.13.216.191, 172.16.0.1 172.16.112.10, 172.16.112.20, 172.16.112.50, . . .	33
SP	Null, 20, 21, 22, 23, 25, 53, 79, 80, 113, 123, 514, 515, 6000, 6667, 8000	16
TF	Null, 0x02, 0x04, 0x10, . . .	9
TU	Null, 0	2
TC	Null, 0xffff	2
TO	Null, 0x020405b4	2
UC	Null, 0xffff	2
IC	Null, 0xffff	2
PS	60–184, 186–231, . . ., 1503–1514	1321

Table 6: Vocabulary list of Pascal.

Field	Words	Anomaly #
EDST	EDST_0000c0, EDST_00105a, EDST_00107b, . . ., EDST_others	7
ESRC	ESRC_0000c0, ESRC_00105a, ESRC_00107b, . . ., ESRC_others	6
TOS	TOS_Null, TOS_00, TOS_08, TOS_10, TOS_others	5
IL	IL_Null, IL_38, IL_40, IL_41, . . ., IL_1500, IL_others	1330
FF	FF_Null, FF_0000, FF_4000, FF_others	4
TTL	TTL_Null, TTL_60, TTL_63, TTL_64, . . ., TTL_255, TTL_others	8
SRC	SRC_135.8.60.182, SRC_135.13.216.191, SRC_172.16.112.10 SRC_172.16.112.20, . . ., SRC_others	30
DST	DST_135.8.60.182, DST_135.13.216.191, DST_172.16.0.1 DST_172.16.112.10, DST_172.16.112.20, . . ., DST_others	34
SP	SP_Null, SP_20, SP_21, SP_22, SP_23, SP_25, SP_53, . . ., SP_515, SP_6000, SP_6667, SP_8000, SP_others	17
TF	TF_Null, TF_02, TF_04, . . ., TF_others	10
TU	TU_Null, TU_0, TU_others	3
TC	TC_Null, TC_ffff, TC_others	3
TO	TO_Null, TO_020405b4, TO_others	3
UC	TC_Null, TC_ffff, TC_others	3
IC	IC_Null, IC_ffff, IC_others	3
PS	PS_60, PS_61, PS_62, . . ., PS_1513, PS_1514, PS_others	1322

Table 7: Example of a simplified session.

Field name	Feature value	Count number
EDST	0x0000c0	100
	0x00105a	100
ESRC	0x0000c0	100
	0x00105a	100
TOS	0x00	200
IL	38	199
	171	1
FF	0x0000	200
TTL	255	200
SRC	172.16.112.10	100
	172.16.112.50	100
DST	172.16.112.10	100
	172.16.112.50	100
SP	Null	200
TF	Null	200
TU	Null	200
TC	Null	200
TO	Null	200
UC	Null	200
IC	0xffff	199
	0xabcd	1
PS	60	199
	185	1

5.3. Training Phase. For each host, an independent LDA model is trained. This is because different hosts may be used for different purposes, for example, mail proxy and Internet server. As a result, the topic distributions could be totally different among hosts. The detection accuracy could be greatly improved if we train an individual LDA model for each host.

For all the hosts, the Dirichlet prior parameters α and β are set empirically to obtain good model quality. We have $\alpha = (10/K)$ and $\beta = 0.01$. According to [47], the number of topics used by LDA will have limited impact on the detection accuracy; thus, for most of hosts with a vocabulary size around 2000, we set $K = 150$ and set $K = 10$ for hosts 172.16.118.80 and 192.168.1.1 whose vocabulary sizes are around 100. Since too large a K will increase the running time, we do not choose K to be large. We use the training documents to train the LDA model. For different hosts, there are 1716 documents at most and 829 documents at least used in the training phase.

After the parameters of LDA have been set, we train the LDA model with the documents of a host and yield the topic-word distribution Φ and topic distribution θ of the host. Since only normal traffic is used in the training phase, Φ can be viewed as a normal behavior pattern of the host. The likelihood of each document is computed using equation (3), and the threshold of the host is set as the minimal likelihood.

5.4. Test Phase. We detect attacks in the test phase. In this phase, we run the LDA model still using the same parameter settings of α, β, and T. Φ is not inferred in this phase because the training phase has already computed Φ which is deemed as the normal behavior structure. The topic distribution θ of every test documents is inferred based on Φ computed in the training phase. The likelihood of each test document is computed using equation (3) according to Φ and the test document's topic distribution θ. It measures the extent to which the test document resembles the normal behavior structure. The document is identified as an attack if the likelihood is lower than the threshold. The host-based method is also used in the test phase.

5.5. Detection Results. All the 18 hosts generate 24037 documents using our method. Of all the documents, 1041 documents are labeled as intrusions. 490 are false positives and 730 are true positives. There are 94 attacks detected because several documents may correspond to one same attack instance.

We compare the performance of our scheme with the performance of PHAD [36] in terms of their ability in detecting intrusions. The comparison result is listed in Table 8. Column 1 of Table 8 is all the attack instances contained in the DARPA 1999 dataset, column 2 is the number of intrusions detected by PHAD, and column 3 is the number of intrusions detected by our method.

5.6. Result Analysis. From the comparison result of Table 8, we can see that our method is superior to PHAD because it detects more types and more instances of attacks. The reason is that our method employs LDA to learn the behavior rule of network traffic. By using LDA, every feature is treated as an independent variable, and all the features are used fully. The behavior rule for the normal traffic is generated automatically. In the LDA model, a topic is a representation of all the normal features with different probability. Φ is the description of normal traffic behavior. A five-minute session of traffic, or a document, should be generated by normal topics if it is to be labeled as normal. As a result, if a document's likelihood computed by the normal behavior rule, or Φ, is lower than the threshold, there may be an attack.

However, in PHAD, the behavior rule for normal traffic is generated by adding up all anomaly values of each feature field, and then the sum is used to separate attacks from normal traffic. The behavior rule generated in this way depends heavily on a single variable, and it is too strong. The information accuracy and extensiveness presented by features are lost by this method. As a result, PHAD cannot detect as many attacks as our method can detect. The limit of our method is that it detects fewer probe attacks such as portsweep, queso, and ipsweep. The reason is that our method is host-based but PHAD is network-based, and the latter has advantage to detect the probe attacks. To fix the problem, we can increase the weight of certain features including port number and TCP flag. We will look into this in our future work.

Table 8: Detection comparison.

Attack	#	Ours	PHAD
anypw	1	1	0
apache2	3	3	2
arppoison	4	1	0
back	4	3	0
casesen	3	2	1
crashiis	8	1	1
dict	1	1	0
dosnuke	4	4	4
Eject	2	2	0
fdformat	3	1	0
ffbconfig	2	1	0
ftpwrite	2	1	0
guessftp	2	1	1
guesspop	1	1	0
guesstelnet	4	2	2
httptunnel	3	1	0
imap	2	2	0
illegalsniffer	2	1	1
ipsweep	7	3	4
mailbomb	4	3	2
mscan	1	1	1
named	3	3	1
ncftp	5	1	0
neptune	4	3	1
netbus	3	2	3
Total		45	24
netcat	4	4	1
ntfsdos	3	1	0
ntinfoscan	3	2	1
perl	4	1	0
phf	4	2	0
pod	4	4	4
portsweep	15	3	13
ppmarcro	3	0	1
processtable	4	3	1
ps	4	2	0
queso	4	0	3
satan	2	2	2
sechole	3	0	1
selfping	3	1	0
sendmail	2	1	1
smurf	5	5	5
snmpget	4	1	0
tcpreset	3	1	0
teardrop	3	3	3
udpstorm	2	2	2
warez	4	2	1
xlock	3	3	1
xsnoop	3	2	0
xterm1	3	1	0
yaga	4	3	0
Total		49	40

6. Conclusion

Using the topic model, we propose a network intrusion detection scheme in this paper. Our scheme proposes a way to analyze network traffic using the LDA model. Packet features are employed to turn network traffic into documents, and the LDA model is used to learn the normal traffic behavior. Experiments on standard dataset are carried out using our method, and the experiment results show the efficiency of our method in detecting network intrusions. Our method can build normal behavior rules automatically for network in advance and then protect network traffic. It is suitable to be used in the networks where there are multiple data formats and data origins, and thus it provides a way of security protection to mobile edge computing.

Data Availability

The DARPA 1999 dataset used to support the findings of this study is included within the article.

Disclosure

Part of this study was finished during the first author's work with Duke University.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank National Defense Pre-Research Plan for the 13th Five-Year Project no. 90407180012, National Science Foundation of China (no. 61771361), Scientific Plan Project of Shaanxi Province (no. 2020JQ-319), and Fundamental Research Funds for the Central Universities (no. JB181503) for funding.

References

[1] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[2] T. Dbouk, A. Mourad, H. Otrok, H. Tout, and C. Talhi, "A novel ad-hoc mobile edge cloud offering security services through intelligent resource-aware offloading," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1665–1680, 2019.

[3] A. Sperotto, M. Mandjes, R. Sadre, P.-T. de Boer, and A. Pras, "Autonomic parameter tuning of anomaly-based IDSs: an SSH case study," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 128–141, 2012.

[4] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi et al., "Adaptive computation offloading with edge for 5G-envisioned Internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[5] X. Xu, B. Shen, X. Yin, M. R. Khosravi, H. Wu et al., "Edge server quantification and placement for offloading social media services in industrial cognitive IoV," *IEEE Transactions on Industrial Informatics*, 2020.

[6] P.-F. Marteau, "Sequence covering for efficient host-based intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 994–1006, 2019.

[7] C. J. Fung, J. Zhang, and R. Boutaba, "Effective acquaintance management based on bayesian learning for distributed

intrusion detection networks," *IEEE Transactions on Network and Service Management*, vol. 9, no. 3, pp. 320–332, 2012.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[9] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. 1, pp. 5228–5235, 2004.

[10] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, IEEE, Los Alamitos, CA, USA, pp. 120–128, May 1996.

[11] Y. Liao and V. R. Vemuri, "Using text categorization techniques for intrusion detection," in *Proceedings of the 11th USENIX Security Symposium*, pp. 51–59, San Francisco, CA, USA, 2002.

[12] Y. Ding, X. Yuan, D. Zhou, L. Dong, and Z. An, "Feature representation and selection in malicious code detection methods based on static system calls," *Computers & Security*, vol. 30, no. 6-7, pp. 514–524, 2011.

[13] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," *IEEE Transactions on Computers*, vol. 63, no. 4, pp. 807–819, 2014.

[14] F. Maggi, M. Matteucci, and S. Zanero, "Detecting intrusions through system call sequence and argument analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 4, pp. 381–395, 2010.

[15] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 4, pp. 227–261, 2000.

[16] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23-24, pp. 2435–2463, 1999.

[17] P. A. Porras and P. G. Neumann, "Emerald: event monitoring enabling responses to anomalous live disturbances," in *Proceedings of the 20th National Information Systems Security Conference*, IEEE, Baltimore, MD, USA, pp. 353–365, 1997.

[18] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: results from the JAM project," in *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*, IEEE, Hilton Head, SC, USA, pp. 130–144, January 2000.

[19] "KDD cup 1999 data," 1999, https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[20] "DARPA intrusion detection evaluation dataset," 1998, https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset.

[21] Y. Wang, "A multinomial logistic regression modeling approach for anomaly intrusion detection," *Computers & Security*, vol. 24, no. 8, pp. 662–674, 2005.

[22] Y. Wang, I. Kim, G. Mbateng, and S.-Y. Ho, "A latent class modeling approach to detect network intrusion," *Computer Communications*, vol. 30, no. 1, pp. 93–100, 2006.

[23] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a hidden naïve Bayes multiclass classifier," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13492–13500, 2012.

[24] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, 2012.

[25] S. Jiang, X. Song, H. Wang, J.-J. Han, and Q.-H. Li, "A clustering-based method for unsupervised intrusion detections," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 802–810, 2006.

[26] Y. Li, B. Fang, l. Guo, and Y. Chen, "Network anomaly detection based on TCM-KNN algorithm," in *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security*, ACM, Singapore, pp. 1–19, 2007.

[27] C.-F. Tsai and C.-Y. Lin, "A triangle area based nearest neighbors approach to intrusion detection," *Pattern Recognition*, vol. 43, no. 1, pp. 222–229, 2010.

[28] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: an intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, pp. 13–21, 2015.

[29] A. I. Saleh, F. M. Talaat, and L. M. Labib, "A hybrid intrusion detection system (HIDS) based on prioritized *k*-nearest neighbors and optimized SVM classifiers," *Artificial Intelligence Review*, vol. 51, no. 3, pp. 403–443, 2019.

[30] G. Liu, Z. Yi, and S. Yang, "A hierarchical intrusion detection model based on the PCA neural networks," *Neurocomputing*, vol. 70, no. 79, pp. 1561–1568, 2007.

[31] A. N. Toosi and M. Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers," *Computer Communications*, vol. 30, no. 10, pp. 2201–2212, 2007.

[32] S. Rezvy, M. Petridis, A. Lasebae, and T. Zebin, "Intrusion detection and classification with autoencoded deep neural network," *Innovative Security Solutions for Information Technology and Communications*, pp. 142–156, Springer, Berlin, Germany, 2019.

[33] J. Kim, J. Kim, H. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Proceedings of the 2016 International Conference on Platform Technology and Service*, IEEE, Jeju, Republic of Korea, January 2016.

[34] C.-H. Tsang, S. Kwong, and H. Wang, "Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection," *Pattern Recognition*, vol. 40, no. 9, pp. 2373–2391, 2007.

[35] N. N. P. Mkuzangwe and F. V. Nelwamondo, "A Fuzzy logic based network intrusion detection system for predicting the TCP SYN flooding attack," *Intelligent Information and Database Systems*, pp. 14–22, Springer, Berlin, Germany, 2017.

[36] M. V. Mahoney and P. K. Chan, "Learning nonstationary models of normal network traffic for detecting novel attacks," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, Edmonton, Canada, pp. 376–385, July 2002.

[37] M. V. Mahoney, "Network traffic anomaly detection based on packet bytes," in *Proceedings of the 2003 ACM Symposium on Applied Computing*, ACM, Melbourne, FL, USA, pp. 346–350, 2003.

[38] T. Shon, X. Kovah, and J. Moon, "Applying genetic algorithm for classifying anomalous tcp/ip packets," *Neurocomputing*, vol. 69, no. 1618, pp. 2429–2433, 2006.

[39] S. B. Shamsuddin and M. E. Woodward, "Modeling protocol based packet header anomaly detector for network and host intrusion detection systems," in *Proceedings of the 6th International Conference on Cryptology and Network Security*, Springer, Singapore, pp. 209–227, 2007.

[40] W. Yassin, N. I. Udzir, A. Abdullah, M. T. Abdullah, Z. Muda, and H. Zulzalil, "Packet header anomaly detection using statistical analysis. International Joint Conference SOCO14-CISIS14-ICEUTE14," *Advances in Intelligent Systems and*

Computing, vol. 299, pp. 473–482, Springer, Berlin, Germany, 2014.

[41] I. Hareesh, S. Prasanna, M. Vijayalakshmi, and S. M. Shalinie, "Anomaly detection system based on analysis of packet header and payload histograms," in *Proceedings of the 2011 International Conference on Recent Trends in Information Technology (ICRTIT)*, IEEE, Chennai, India, pp. 412–416, June 2011.

[42] P. Manandhar and Z. Aung, "Towards practical anomaly-based intrusion detection by outlier mining on TCP Packets," *Database and Expert Systems Applications*, pp. 164–173, Springer, Berlin, Germany, 2014.

[43] C. Cramer and L. Carin, "Bayesian topic models for describing computer network behaviors," in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Prague, Czech Republic, 2011.

[44] E. M. Ferragut, D. M. Darmon, C. A. Shue, and S. Kelley, "Automatic construction of anomaly detectors from graphical models," in *Proceedings of the 2011 IEEE Symposium on Computational Intelligence in Cyber Security*, IEEE, Paris, France, 2011.

[45] J. Huang, Z. Kalbarczyk, and D. M. Nicol, "Knowledge discovery from big data for intrusion detection using LDA," in *Proceedings of the 2014 IEEE International Congress on Big Data*, IEEE, Washington, DC, USA, pp. 760-761, 2010.

[46] H. J. Steinhauer, T. Helldin, G. Mathiason, and A. Karlsson, "Topic modeling for anomaly detection in telecommunication networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2019, 2019.

[47] S. Lee, S. Kim, S. Lee et al., "LARGen: automatic signature generation for malwares using latent Dirichlet allocation," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 771–783, 2018.

[48] D. Mimno and D. Blei, "Bayesian checking for topic models," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, IEEE, Edinburgh, Scotland, pp. 227–237, 2011.

[49] Wireshark, "Go deep," 2020, https://www.wireshark.org/.

[50] RFC 791, "Internet protocol," 1981, https://tools.ietf.org/html/rfc791#page-11.

[51] RFC 793, "Transmision control protocol," 1981, https://www.ietf.org/rfc/rfc793.txt.

[52] "Darpa intrusion detection evaluation dataset," 1999, https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset.