Hindawi
Security and Communication Networks
Volume 2019, Article ID 5306395, 11 pages
https://doi.org/10.1155/2019/5306395

*Research Article*

# Applying Catastrophe Theory for Network Anomaly Detection in Cloud Computing Traffic

**Leila Khatibzadeh**,[1] **Zarrintaj Bornaee**,[1] **and Abbas Ghaemi Bafghi**[2]

[1]*Electrical Engineering and Information Technology Department, Iranian Research Organization for Science and Technology (IROST), Tehran 3353136846, Iran*
[2]*Computer Department, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran*

Correspondence should be addressed to Zarrintaj Bornaee; bornaei@irost.org

Received 18 August 2018; Revised 25 December 2018; Accepted 27 February 2019; Published 2 May 2019

Guest Editor: Yuan Yuan



In spite of the tangible advantages of cloud computing, it is still vulnerable to potential attacks and threats. In light of this, security has turned into one of the main concerns in the adoption of cloud computing. Therefore, an anomaly detection method plays an important role in providing a high protection level for network security. One of the challenges in anomaly detection, which has not been seriously considered in the literature, is applying the dynamic nature of cloud traffic in its prediction while maintaining an acceptable level of accuracy besides reducing the computational cost. On the other hand, to overcome the issue of additional training time, introducing a high-speed algorithm is essential. In this paper, a network traffic anomaly detection model grounded in Catastrophe Theory is proposed. This theory is effective in depicting sudden change processes of the network due to the dynamic nature of the cloud. Exponential Moving Average (EMA) is applied for the state variable in sliding window to better show the dynamicity of cloud network traffic. Entropy is used as one of the control variables in catastrophe theory to analyze the distribution of traffic features. Our work is compared with Wei Xiong et al.'s Catastrophe Theory and achieved a maximum improvement in the percentage of Detection Rate in week 4 Wednesday (7.83%) and a 0.31% reduction in False Positive Rate in week 5 Monday. Additional accuracy parameters are checked and the impact of sliding window size in sensitivity and specificity is considered.

## 1. Introduction

Nowadays cloud computing is the fastest-growing distributed computational platform in domains such as industries and research communities. In general, connected resources through various distributed networks form the cloud [1]. The network is a pivotal part of the cloud which provides quality of service, namely, ensuring the time constraints. Without it, integrations of various computation and storage resources are impossible [2]. It fulfills two important roles in the cloud environment: interacting with user application for connecting to the appropriate resource and sending back the output to the users [3]. Therefore, the importance of cloud networks has led to attacks on such networks by intruders via malicious attacks which will affect user applications and cloud resources causing a delay in the execution process within the overall cloud computing application [4].

Characterizing and monitoring network traffic, specifically resulting from the outburst in traffic arising from the massive number of cloud tenants that are connected to the internet, is becoming a more complex task. Nowadays, the fast-rising networks duplication, data transfer speed, and unpredictable internet usage have added further anomaly problems [5]. This challenge is even greater in cloud computing environments because its traffic may undergo sudden changes, and the elastic and scalable nature of cloud may easily be confused with traffic anomalies and lead to improper network management [6].

In a traditional network, the nodes are fixed, whilst in the cloud, the nodes are likely to move from one physical machine to others [7]. In the scene of cloud computing, traditional intrusion detection methods lack practicality [8]. The anomaly detection system used in a traditional network cannot be applied to such systems because of the dynamic

nature of logical resources [7]. Traditional network traffic predictors are often modelled on large historical databases. These databases are used for training algorithms. This may not be suitable for such highly unstable environments, where the interaction between past and current values might change quickly over time [9]. In such an environment, coping with the novelty of attacks in such situations is difficult due to various constraints like the unavailability of cloud networks, abundance of network links and devices, network virtualization, unpredictability of the network data, high bandwidth, fast moving network data, dissimilarity, and multitenancy, which lead intruders to exploit cloud networks with different attacks [10]. One of the challenges in predicting network traffic in the cloud is to minimize the computational cost besides maintaining an acceptable levels of accuracy. This issue is not clear while many of the current prediction models are unable to maintain a low computational complexity and dealing with a high degree of workload information over a short span of time [9]. These prediction models form the basis of the anomaly detection algorithm. In past decades, the majority of studies on anomaly detection systems have applied various soft computing, data mining and machine learning approaches for designing anomaly detection systems. Nevertheless, these systems are still inaccurate and involve more computational complexity [7].

Due to these challenges, we present another dynamic method based on Xiong et al.'s Catastrophe Theory [4] to detect network anomalies in the cloud environment. The reason why this theory was chosen in cloud platform is the distribution of processes and their sudden changes in the cloud environment which leads to the malfunctioning of anomaly instead of the dynamics associated factor; entropy is used as a disorderness factor due to its speed and light computing power. Because of the dynamic nature of cloud network traffic, the exponential moving average is introduced to diminish the impact of weights through time. To evaluate the performance of our approach, our methods are validated on the standard Defense Advanced Research Projects Agency data sets and our results are compared with Xiong et al.'s catastrophe model. The results depict that our approach based on modified Catastrophe Theory is effective in the processing time of randomness calculation and Detection Rate to detect cloud network anomalies. Additional accuracy parameters are calculated and compared with Xiong et al.'s.

The rest of the paper is organized as follows. Section 2 covers some of the most outstanding related work. In Section 3, the main conceptual theory of the article and our proposed method with its applying methodology are presented as the materials and methods section. Then we indicate some experimental results and validate the performance of our methods in Section 4. The final section concludes the paper and suggests directions for future work.

## 2. Related Works

Detecting anomaly, particularly in a safety critical system, is of considerable importance to mitigate any system failures in the future [17]. In some systems, such failures could lead to tremendous environmental catastrophes. Anomaly detection methods make use of a wide range of techniques based on statistics, classification, clustering, nearest neighbor search, and information theory [18]. Network anomaly detection is a source of difficulty due to the dynamic nature of network traffic. In another study, they classified solutions based on techniques taken from statistics, data mining, and machine learning [19].

There are numerous methods for detecting anomalies, some of which will be reviewed in the following section. Bhat et al. introduced the virtual machine monitor anomaly detection model on cloud virtual machines using the machine learning approach. In the first part of their framework, the Naive Bayes (NB) Tree algorithm is applied to classify network connections into intrusion and normal data based on a labeled training dataset. The dataset aids the construction of classification patterns. In the second part of anomaly detection, a hybrid approach of NB Tree and Random forest algorithm is applied based on the likeness of connection features [11]. The method outperforms the traditional Naive Bayes in terms of detection accuracy, error rate, and misclassification cost. In a similar study, Fu et al. proposed another machine learning approach in which one-class and two-class Support Vector Machines are used in cloud computing. The first class formed abnormality score while the second class is retrained for new records appearance [12]. It does not need a prior failure history and is self-adaptive through learning from observed failure events, but the accuracy of failure detection is not fool proof [20].

Zhao and Jin proposed an automated approach to intrusion detection for keeping sufficient performance and reducing execution environment dependence in a VM-based environment. They presented a dynamic graph structure to monitor the dynamic changes in the environment. Based on this structure, a Hidden Markov Model (HMM) strategy for detecting abnormality using frequent system call sequences was considered to automatically and efficiently identify attacks and intrusions. An automated mining algorithm, which is called AGAS, was proposed to generate frequent system call sequences. The AGAS algorithm utilizes related probabilities to identify frequent sequences instead of setting a user-defined threshold on relevant sequences. According to the execution state, the detection performance is adaptively tuned every period. But if the system behavior changes intensively, the overhead of the dynamic graph might be increased and the benefit of their approach will diminish [13].

Jabez et al. proposed a new approach for detecting intrusions within computer networks which is called Outlier Detection [5]. Their training model is made up of big datasets with a distributed environment which is quite similar to the cloud. The performance of their method is superior to other existing machine learning approaches and can significantly detect all anomaly data in computer networks. In another study of this kind, Xinlong Zhao et al. proposed a new intrusion detection method based on improved K-means. The method is designed to fit the characteristics and security requirements of cloud computing. It includes a clustering algorithm and a method for distributed intrusion detection modelled on it. The new method can detect known attacks

in addition to anomaly attacks in the cloud computing environment. The result of a simulation test proves that this improved method can decrease the false positive and false negative rate and speed up the intrusion detection process [8]. Pandeeswari and Kumar also used the clustering method and proposed an anomaly detection system at the hypervisor layer which employs a hybrid algorithm (FCM-ANN), a combination of the Fuzzy C-Means clustering algorithm and Artificial Neural Network, in order to improve the accuracy of the detection system. FCM-ANN can automatically capture new attack patterns, so there is no necessity to manually update the database. Compared with the Naïve Bayes classifier and Classic ANN algorithm, their system can detect anomalies with a high detection accuracy and a low false alarm rate. Low frequent attacks can also be detected. It outperforms the Naïve Bayes classifier and Classic ANN [14]. Later, in 2016, they proposed an anomaly detection system named hypervisor detector in the cloud environment. It is designed with ANFIS (integration of fuzzy systems with adaptation and learning proficiencies of neural network called adaptive neurofuzzy inference system) in order to detect anomalies in the cloud network. The ANFIS used the back propagation gradient descent technique in combination with the least square scheme for the purpose of training and testing the system. The comparison results with other IDS using ANN, Naive Bayes, and a hybrid approach that combined Naïve Bayes and random forest indicate that the proposed model is both efficient and effective in discovering the anomalies in the cloud environment and is ideal for detecting anomalies with high detection accuracy and a minimum false negative rate. By employing hypervisor detector in very large datasets, it is possible to attain the best performance in the cloud environment, but the complexity of the algorithm makes its implementation difficult for such an environment [7].

Xiong et al. proposed two anomaly detection models based on catastrophe theory in network traffic [21, 22]. After that, in 2014 they proposed an intrusion detection system to detect network traffic anomaly based on synergetic neural networks and the catastrophe theory to reduce security risks in the cloud network. The results show high detection and low false alarm rates [4]. They adopted impulsive neural networks in addition to fast compression algorithms to examine network traffic anomaly in the cloud computing environment. They subsequently proposed the idea of network-based intrusion detection on the cloud platform but did not provide a clearer definition of the anomaly [8]. Approaches based on pattern recognition, such as first part of Xiong et al. which is related to neural networks, involve a severe disadvantage: they learn the training patterns but lack the ability to make generalizations, so the model may give inaccurate results for unknown patterns [9]. In the second part of Xiong et al.'s paper, the sudden change process of the network is shown by the catastrophe potential function. An index referred to as catastrophe distance is presented to assess deviations from normal behavior in detecting network anomalies.

In some cases, detecting network anomalies is performed based on traffic prediction. Network traffic prediction has received a great deal of attention for facilitating monitoring and managing computer networks [6]. In this field, most research efforts are focused on classical methods strongly based on historical data. Predicting network traffic is pertinent to many management applications such as resource allocation, admission control, and congestion control [10]. The major issue with these models is the computational overhead in relation to the size of the input data [9], which could be more intense in cloud computing due to its volatile and extensive environment. Yuehui Chen et al. use genetic programming to build a Flexible Neural Tree (FNT) for predicting network traffic online [15]. This approach was employed for a better comprehension of the main features of traffic data. Moreover, the proposed method is able to predict short time scale traffic measurements and can reproduce statistical features related to real traffic measurements. However, it needs initial input which is dependent on the characteristics of data that is being evaluated to achieve proper results [10]. Moayedi and Shirazi proposed a different model for predicting network traffic and detecting anomalies based on Autoregressive Integrated Moving Average (ARIMA). In order to isolate anomalies from normal traffic variation, they decompose the data flow. The authors try to predict anomalies independently from normal traffic. They evaluated their work with synthetic data, which depends on large historic data [16]. Although these works allow online traffic prediction, due to their dependence on large historical data for training the algorithms, they are inappropriate in the cloud environment [6]. To address this issue, Dalmazo et al. proposed a dynamic window size methodology for traffic prediction. The size of the window is related to the amount of traffic for traffic prediction and varies according to bounded historical data by using network traffic features such as short-range dependence [6]. They estimated network traffic through a statistical method based on a Poisson process. Their mechanism can be applied to determine the scope of data to be analyzed for any traffic prediction approach.

Table 1 presents these approaches and groups them according to their methods outlining their advantages and disadvantages.

According to works related to anomaly detection, Xiong et al. [4] used fast compression algorithms to examine network traffic anomaly in the cloud computing environment. They pay much attention to the sudden change process of network traffic which other works have not addressed. This is why we have chosen to use this theory in the present study.

## 3. Materials and Methods

Cloud computing provides a dynamic environment with complex network traffic behavior and without having linear trend pattern; therefore, high degree polynomials are required to fit the network traffic baseline [9]. The dynamic nature of cloud network traffic flow depends on equilibrium changes determined by primary factors. In normal network traffic, when no anomalies occur, the network state is referred to as the normal state of equilibrium. When anomalies occur, the network state will transform from a normal equilibrium state to an abnormal one driven by abnormal factors. The change process of the network traffic is regarded to be

Table 1: Comparison of anomaly detection approaches via their methods.

| Methods | Author Name | Techniques | Pros and Cons |
|---|---|---|---|
| Classification | Bhat et al. [11]<br>Fu et al. [12]<br>Feng Zhao and Hai Jin [13] | NB Tree & Random Forest<br>One Class and Two Class Support Vector Machines<br>Hidden Markov Model & mining algorithm | Through powerful algorithms, the method can distinguish between different class instances but they depend on labels. This is often impossible to achieve. |
| Nearest neighbor | Jabez et al. [5] | Outlier Detection | They are commonly used because they are unsupervised and do not require any data distribution, but for unsupervised techniques, if the normal data instances lack close enough neighbors or the anomalies have close enough neighbors, the technique fails to label them and computing complexity is, therefore, a challenge. |
| Clustering | Xinlong Zhao et al. [8]<br>Pandeeswari and Kumar [14] | K-means Clustering<br>Fuzzy C-means Clustering & Artificial Neural Network | The method is relatively faster than distance-based methods and they could reduce the computational complexity during the process of detecting intrusions in large datasets, but in smaller datasets, they may not provide accurate insights at the desired level of detail and dynamic updating of profiles is time consuming. |
| Statistical | Kumar and Pandeeswari [7]<br>Xiong et al. [4] | Fuzzy system & Neural Network<br>Neural Network & Catastrophe theory | A statistically-justifiable solution for detecting anomalies can be yielded from these methods, if the assumptions considering the data distribution hold true and the confidence interval for the anomaly score can be applied for decision making as additional information, but they are dependent on the assumption that the data is generated via a specific distribution. |
| Prediction | Yuehui Chen et al. [15]<br>Moayedi and Shirazi [16]<br>Dalmazo et al. [6] | Flexible Neural Tree<br>Autoregressive Integrated Moving Average<br>Poisson Moving Average | In some cases, predicting anomalies is done independently from normal traffic, but they almost depend on large historic data. |

transient and catastrophic [4]. As a result, the Catastrophe Theory is used to better display fluctuations in network traffic. But as mentioned in the previous section, Xiong et al. did not provide a clearer definition of anomalies which leads to not clearly defining network traffic behavior. In this article, in order to better describe traffic behavior, EMA is used.

In the following sections, we will first explain the details of this theory and then present the proposed method based on it.

*3.1. Catastrophe Theory.* Catastrophe Theory is a method for describing the evolution of forms in nature and is one of the main branches of dynamic systems. It was invented by Rene Thom in the 1960s [23] who explained the philosophy behind the theory in his 1972 book *Structural Stability and Morphogenesis*. Catastrophe Theory is specifically applicable to situations where gradually changing forces produce sudden effects. The applications of catastrophe theory in classical physics (or more generally in any subject governed by a minimization principle) provide us with a better understanding of what diverse models have in common [24]. The theory is derived from a branch of mathematics (topology) concerned with the properties of surfaces in many dimensions. Topology is involved because smooth surfaces of equilibrium describe the underlying forces in nature: when the equilibrium breaks down, catastrophe occurs. The strength of the model derived from catastrophe theory is that it can account for the bimodal distribution of probabilities [25]. "The line that marks the edges of the pleat in the behavior surface, when the top and bottom sheets fold over to form the middle sheet, is called the fold curve. When it is projected back onto the plane of the control surface, the result is a cusp-shaped curve" [26]. For this reason, the model is called the cusp catastrophe. It is one of the simplest of the seven elementary catastrophes, and so far, it has been the most productive [27]. In this paper, the cusp or Riemann-Hugoniot [28] catastrophe model is used to reveal the network traffic anomaly in the cloud.

In a sense, elementary catastrophe theory is a generalization of theorems about critical points or singularities of real-valued functions of $n$ real variables to one about parameterized families of such functions. Catastrophe theory analyzes degenerated critical points of the potential function. The critical points satisfy the condition that the first derivative and higher derivatives of the potential function are zero [4]. These are called the germs of the catastrophe geometries. The degeneracy of the critical points can be unfolded by expanding the potential function as a Taylor series in small perturbations of the parameters. What Thom has done is to determine conditions on derivations of $f$ which ensure the existence and which give a local normal form for a stable unfolding of $f$. He also shows that whenever the parameter space or control space has dimension $\leq 5$, there is a finite classification of stable unfolding [28].

The potential function $F(x)$ of the cusp catastrophe model is shown in [24]

$$F(x) = x^4 + aux^2 + bvx \tag{1}$$

where $x$ is a state variable, $u$, $v$ are control variables, and $a$, $b$ are the coefficients [22]. The equilibrium surface has the equation $F'(x) = 0$ and $G(x, u, v)$ which can be achieved by

$$4x^3 + 2aux + bv = 0. \tag{2}$$

The normal to the surface is vertical when $F''(x) = 0$ and the singularity set of the cusp catastrophe is achieved by

$$6x^2 + au = 0. \tag{3}$$

The bifurcation set (cusp) is the critical image of the projection $(u,v,x) \longrightarrow (u,v)$ from the equilibrium surface onto the control space. The equation difference set $G(x, u, v)$ of the cusp is as follows:

$$8a^3u^3 + 27b^2v^2 = 0 \tag{4}$$

It is obtained by eliminating $x$ from (2) and (3) for the fold curve.

The equilibrium surface has equation $4x^3 + 2aux + bv = 0$ where $(u, v)$ are coordinates on the control space and the vertical coordinate $x$ is only state variable. As the control $(u, v)$ varies, the state $(u, v, x)$ will be forced to jump to the other sheet when it crosses the fold curve. The curve over the cusp is shown in Figure 1. The top surface is the equilibrium surface of the cusp catastrophe model, which is divided into the upper sheet and lower sheet [24]. When the state of the system transfers from the stable equilibrium state to another stable equilibrium state, there is a sudden jump between the stable states and then sudden change appears. The bottom one is the control space illustrated by the control variables $u$, $v$.

*3.2. Proposed Method.* In this part of the section, we propose our anomaly detection method based on the modified Catastrophe Theory and discuss our contributions based on Xiong et al.'s Catastrophe Theory to detect anomalies in cloud network traffic. The reason why we followed this theory could be due to their attention to sudden change process of cloud network traffic that most of the previous works did not pay much attention to. By studying this theory, we realized that it could be possible to generalize it to similar problems in detecting abnormalities in similar conditions.

How to extract state and control variables plays a significant role in the accurate analysis of the model. In our model, the Hurst index [29] reflects the degree of the self-similarity between the current and next state of the network traffic and was selected as $u$ like the one that Xiong et al. used [4]. Nevertheless, the majority of algorithms for the similarity-based detection of anomalies use a multivariate distance function and these functions are susceptible to the problem of dimensionality, which means they are unable to provide a reliable measurement of the similarity of high-dimensional data because of the data dispersion in high dimensional spaces [18]. In addition, such functions cannot localize the source of anomaly and detect the specific dimensions that cause anomalous patterns [18]. We use a damping coefficient to diminish the effect of similarity versus randomness in cloud network traffic. In the field of engineering, the damping

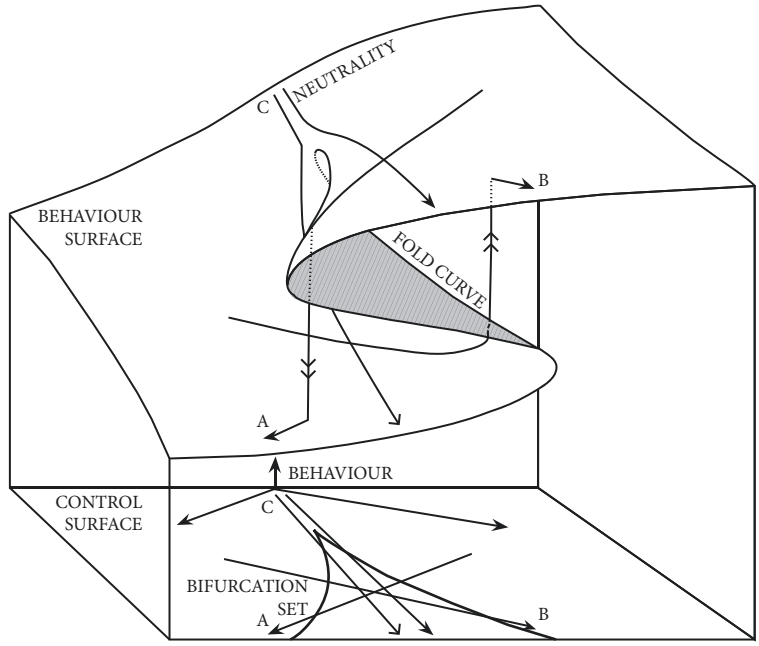


Figure 1: Cusp catastrophe model [27].

coefficient is a dimensionless measure which describes how oscillations decay after a disturbance [30]. Many systems show oscillatory behavior upon being disturbed from their position of static equilibrium [31]. Larger values of the damping coefficient or damping factor produce transient responses with a minor oscillatory nature, which is similar to the dynamic nature of cloud network traffic.

Entropy reflects the level of irregularities that occur or in other words, it is a measure of disorder, and we have selected it as control variable $v$. The entropy-based method needs little computing power and is fast enough for detecting anomalies [17]. These features are appropriate for our purpose in our case. We utilize the entropy concept for analyzing the randomness distribution of features in cloud network traffic.

The state variable $x$ (of the cusp catastrophe model) was taken as the volume of the network traffic. We use a sliding window to construct a vector set. Exponential Moving Average (EMA) is applied in constructing $x$ value due to the volume of traffic accumulated in each sliding window.

In [32], Frank Klinker defines a mathematical tool for the prediction of market trends. Specifically, he states that it is possible to use the Exponential Moving Average (EMA) in order to efficiently forecast network traffic with short historical data. In the case of EMA, weighting coefficients increase exponentially through the time in the sliding window. The weighing for the oldest values in each sliding window reduces exponentially and never reaches zero unlike most other moving averages, so this approach reacts faster to recent value changes. Similar to other techniques that make use of a moving average, EMA should strictly be used for data that does not involve seasonal behavior [9] and according to the dynamic nature of cloud traffic and because of online detection; we have chosen to use this kind of moving average instead of the others. The formula for calculating EMA is as follows:

$$EMA = \frac{\sum_{i=s}^{n+s} \left(exp^i * y_i\right)}{\sum_{i=s}^{n+s} \left(exp^i\right)}. \tag{5}$$

In this formula, $n$ is the size of time window and $y$ is the number of packets which are received in each $i$ seconds. Because of applying sliding window and various $x$ for each time window, the $s$ parameter which changes the basis of the formula indicates the start time of each time window and then the result reflected by $EMA$ replaces the $x$ variable for each time window.

After calculating the parameters $(x, u, v)$ for each sliding window in train and test data, we must calculate the catastrophe distance ($Dp$) between the observed point in test data and the bifurcation set $G(x, u, v)$. Assume one point ($P_t$) of the equilibrium surface $G(x, u, v)$ and one point of test data ($P_i$); the catastrophe distance between these two points, labelled as "$D_E(P_i,P_t)$, is computed by the Euclidean distance" [22] as shown in

$$D_p\left(P_i, G\left(x, u, v\right)\right) = \min_{P_t \epsilon G(x,u,v)} \left\{D_E\left(P_i, P_t\right)\right\}. \tag{6}$$

When the catastrophe distance $Dp$ is beyond a given threshold $\eta$, there is an anomaly which exists at the observing point in the test data.

## 4. Results and Discussion

This section presents the experimental results to evaluate the proposed anomaly detection method. The standard DARPA datasets used in our experiments are widely used in network intrusion detection, to name a few, Horng et al. [33], Shon

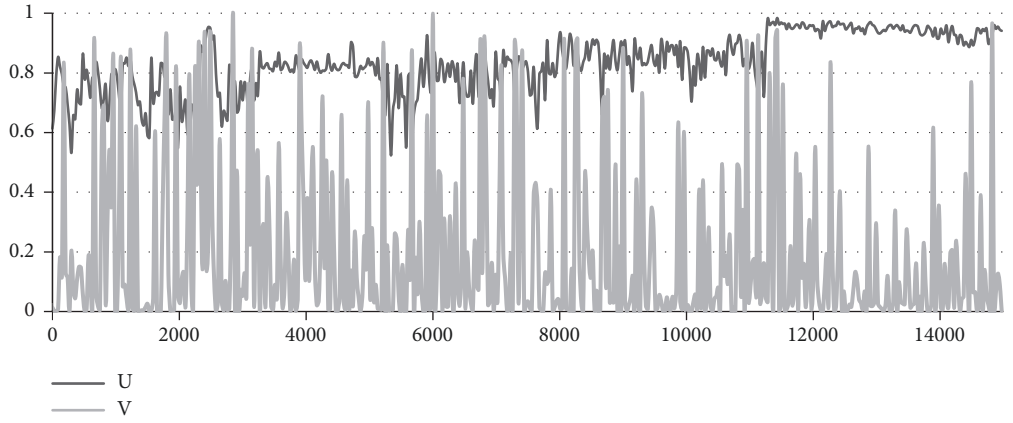


Figure 2: A trade-off between $u$ & $v$ control variables in week 5 Tuesday; the horizontal axis represents time.

and Moon [34], and Xiong et al. [4]. Although there are new intrusion datasets such as ISCX 2012, NSL-KDD 2013, and CIC 2017, DARPA dataset is used to compare the results with Xiong et al.'s paper [4]. It contains 5 weeks of data, whose 4 weeks are utilized in our implementation. Weeks 1 and 3 traffic data included no attack. Traffic data from these 2 weeks are used as train datasets. Weeks 4 and 5 traffic data included different types of attacks mixed with normal traffic and are used as test datasets. In this paper, we extract "the aggregated network traffic in packets per second from the tcpdump data files" [4] in the DARPA data set [35].

For all experiments, processing time, the number of false positives, and the number of true anomalies which can be detected are reported. According to attack file which was published with DARPA, false positive and true positive could be calculated. These experiments are processed at 8 Core Xenon Server with 2.19 GHz CPU frequency with 16 GB memory.

*4.1. Parameter Analysis.* We calculate each state and control variables based on weeks 1, 2, 4, and 5. As the input data, we produced a list of catastrophe distance before applying the algorithm for detecting anomalies. "The given parameters $p$, $\eta$ were chosen as $p = 30$ and $\eta = 0.85$" like Xiong et al.'s [4]. We change the threshold in 5 days of test data, and in 3 cases, the number 0.8 reached better results but because of the comparison, 0.85 is chosen as $\eta$.

We randomly choose control parameters from test files in weeks 4 and 5. Then compare their distribution achieved in each random interval, namely, in the first 15000 seconds of week 5 Tuesday which is depicted in Figure 2. Due to the dynamic nature of the cloud, the randomness control parameter is dominant, but as they proceed and time elapsed, the self-similarity parameter does not show any significant changes. To reduce the impact of self-similarity control factor, the damping coefficient has been used. The selection of the damping coefficient for such an application needs a trade-off between the maximum percentage of self-similarity and the time of the peak in which self-similarity occurs. A smaller damping coefficient reduces the time, but it enhances the maximum percent of similarities which is not desirable for cloud network traffic. The final choice of the damping coefficient is subjective. It has been Shinners' experience that "the damping coefficient range is usually selected between 0.4 and 0.7 for general cases" [36]. Experiments were performed to determine the amount of damping coefficient whose [0.6, 0.7] interval results the best and in most cases, 0.69 resulted better, so in this experiment $\zeta = 0.69$ was considered.

*4.2. Experimental and Comparisons.* Detection Rate and False Positive Rate are common metrics for assessing the effectiveness of an anomaly detection system. The Detection Rate (DR) is the number of correctly classified as normal packets divided by the total number of test data (or true negative plus false positive). The False Positive Rate (FPR) is defined as the total number of normal data, which was classified as anomalies wrongly, divided by the total number of normal data traffic (or true negative plus false positive) [37] as shown in the following, respectively:

$$DR = \frac{N_{detected}}{TN + FP} \tag{7}$$

$$FPR = \frac{NF_{detected}}{TN + FP}. \tag{8}$$

The detection results based on Xiong et al. are depicted in Figures 3 and 4 [4]. We implemented the Xiong et al. article with C# programming language and repeated the experiments. Then, our method is implemented and these improvements in DR and FPR are achieved; the average rates of these improvement percentages in two weeks are 2.24 and 0.069 promotion, respectively. The results show that, in most days, DR is improved, but FPR could not show the best results contrary to what we expected. It may be possible if we change the damping coefficient due to the trade-off between self-similarity percentages versus disorderness percentage.

The attack file, which was published by MIT Lincoln Laboratory [35], has two main parts for each attack; the first one is the ID of each attack that contains some subattacks with different start times and durations. In this comparison, we use each attack with its ID information and did not consider any subinformation which is distinct between each part of an

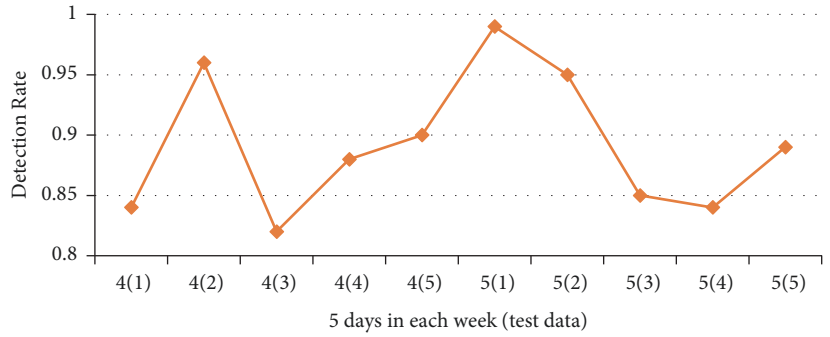


Figure 3: DR of each day in weeks 4 and 5 in Xiong et al.'s catastrophe theory [4].

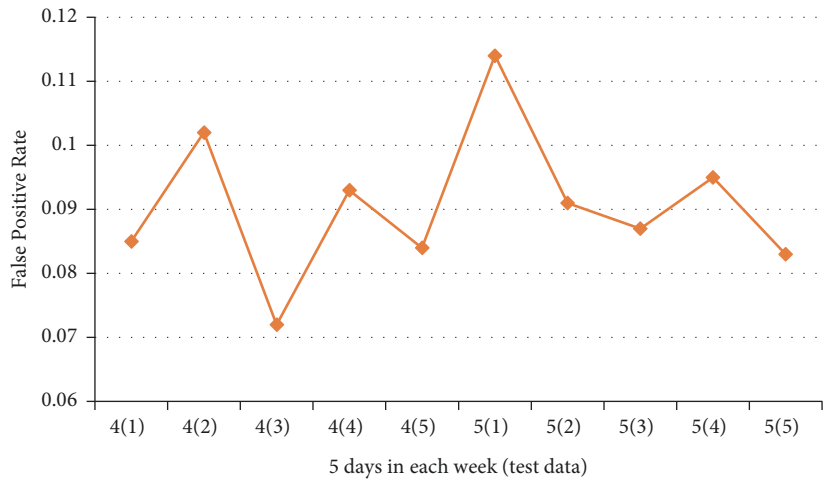


Figure 4: FPR of each day in weeks 4 and 5 in Xiong et al.'s catastrophe theory [4].

attack. In the second experiments, details are considered. Due to this change, the results are more accurate. For instance, in week 4 Thursday, the improvement percentage of DR is 7.83% in accordance with details implemented as Xiong et al.'s catastrophe theory. In the same day, the improvement percentage of FPR shows 0.042% reduction. But in week 5 Tuesday, we could not reach the ideal. In week 4 Friday and week 5 Wednesday, DR and FPR rates repeated exactly. The results of our implementation based on DR and FPR improvement percentage are indicated in Table 2. The result of the comparison on our model shows better precision.

How correctly an intrusion detection system works is measured by a metric referred to as accuracy. It measures the percentage of detection and failure as well as the number of false alarms produced [38]. We compared our model with Xiong et al.'s in some accuracy measures as follows.

(a) Misclassification rate estimates the probability of disagreement between the true and predicted cases by dividing the sum of FN and FP by the total number of pairs observed. In other words, misclassification rate is defined as [38] follows:

$$mis - classification\ rate = \frac{(FN + FP)}{(TP + FP + FN + TN)}. \quad (9)$$

The improvement percentage of misclassification rate between Xiong et al.'s model and our proposed model is indicated in Table 3.

(b) Precision is a measure of how a system identifies abnormality or normality. Precision is used to measure the exactness of the detection. We calculated this factor in our model and compared it with Xiong et al.'s, and the improvement percentage is shown in Table 3.

(c) The F-measure mixes the values of two previous measures (precision and recall). By considering only one metric for evaluation, F-measure is the most preferable. It is calculated as follows:

$$F - measure = \frac{2}{1/precision + 1/recall}. \quad (10)$$

Table 3 depicts the results of F-measure improvement in 2-week test data.

Considering processing time, one could notice that our method is more efficient than Xiong et al.'s in calculating randomness parameters. Comparisons of processing time in calculating $v$ control parameter for each day in train and test data are shown in Tables 4 and 5, respectively. In Table 4, for instance, in week1 Monday, the processing time with Xiong

Table 2: DR & FPR improvement percentage in each day of test data between Xiong et al. and proposed model.

| Weeks include test data► | W4(1) | W4(2) | W4(3) | W4(4) | W4(5) | W5(1) | W5(2) | W5(3) | W5(4) | W5(5) |
|---|---|---|---|---|---|---|---|---|---|---|
| DR Improvement Percentage | 0.15% | 7% | 3.2% | 7.83% | - | 4.29% | reduced | - | 0.033% | 0.30% |
| FPR Improvement Percentage | 0.025% | 0.03% | 0.031% | 0.042% | - | 0.31% | increased | - | 0.16% | 0.15% |

Table 3: Misclassification rate, precision, and F-measure improvement in each day of test data between Xiong et al. and proposed model.

| Weeks include test data► | W4(1) | W4(2) | W4(3) | W4(4) | W4(5) | W5(1) | W5(2) | W5(3) | W5(4) | W5(5) |
|---|---|---|---|---|---|---|---|---|---|---|
| Misclassification rate Improvement Percentage | 0.253% | 0.232% | 0.647% | 0.207% | 0 | 0.315% | reduced | 0 | 0.106% | 0.570% |
| Precision Improvement Percentage | 0.008% | 0.006% | 0.016% | 0.008% | 0 | 0.028% | reduced | 0 | 0.010% | 0.018% |
| F-measure Improvement Percentage | 0.04% | 0.018% | 0.104% | 0.027% | 0 | 0.019% | reduced | 0 | 0.017% | 0.064% |

Table 4: Comparison of processing time in calculating randomness for each day in train weeks.

| Weeks include train data► | W1(1) | W1(2) | W1(3) | W1(4) | W1(5) | W3(1) | W3(2) | W3(3) | W3(4) | W3(5) |
|---|---|---|---|---|---|---|---|---|---|---|
| Execution time of Xiong et al. Randomness algorithm [sec] | 0.091 | 0.068 | 0.083 | 0.083 | 0.105 | 0.087 | 0.098 | 0.097 | 0.094 | 0.091 |
| Execution time of our Randomness algorithm [sec] | 0.045 | 0.05 | 0.036 | 0.039 | 0.042 | 0.047 | 0.049 | 0.053 | 0.054 | 0.057 |
| Differences between two time [sec] | 0.046 | 0.018 | 0.047 | 0.044 | 0.063 | 0.04 | 0.049 | 0.044 | 0.04 | 0.034 |
| Improvement percentage (%) | 50.55 | 26.47 | 56.63 | 53.01 | 60.00 | 45.98 | 50.00 | 45.36 | 42.55 | 37.36 |

Table 5: Comparison of processing time in calculating randomness for each day in test weeks.

| Weeks include test data► | W4(1) | W4(2) | W4(3) | W4(4) | W4(5) | W5(1) | W5(2) | W5(3) | W5(4) | W5(5) |
|---|---|---|---|---|---|---|---|---|---|---|
| Execution time of Xiong et al. Randomness algorithm [sec] | 0.104 | 0.124 | 0.11 | 0.117 | 0.147 | 0.123 | 0.125 | 0.129 | 0.142 | 0.161 |
| Execution time of our Randomness algorithm [sec] | 0.084 | 0.094 | 0.079 | 0.074 | 0.078 | 0.084 | 0.087 | 0.089 | 0.094 | 0.097 |
| Differences between two time [sec] | 0.02 | 0.03 | 0.031 | 0.043 | 0.069 | 0.039 | 0.038 | 0.04 | 0.048 | 0.064 |
| Improvement percentage (%) | 19.23 | 24.19 | 28.18 | 36.75 | 46.94 | 31.71 | 30.40 | 31.01 | 33.80 | 39.75 |

et al. algorithm is 0.091 s but in our algorithm, the time is 0.045 s which improved 50%. In Table 5, for example in week 4 Friday, the processing time with Xiong et al. algorithm is 0.147 s but in our algorithm, the time is 0.078 s which improved approximately 47%. As implicated in Tables 4 and 5, our method is approximately two times faster than Xiong et al.'s algorithm in terms of efficiency.

*4.3. Sensitivity and Specificity Analysis.* As mentioned about sliding window size and because of the comparison, we select the same window size as Xiong et al.'s. On the other hand, applying different window sizes and studying their impacts on sensitivity have always been a question for us. So, other implementations based on different window size have been performed. TPR which is also known as sensitivity and TNR which is also called specificity are considered. The results of Thursday week 5 are indicated in Table 6.

Table 6: Sensitivity and Specificity in Thursday Week 5 with different sliding window size.

| Window Size► | 30 | 40 | 50 | 60 | 70 |
|---|---|---|---|---|---|
| TPR (Sensitivity) | 86 | 98 | 100 | 99 | 97 |
| TNR (Specificity) | 80 | 86 | 86 | 86 | 85 |

## 5. Conclusions and Future Work

Security threats from inside and outside the cloud make security a major challenge in the widespread cloud adoption. One of the main challenges of the cloud is the tremendous amount of network traffic and the diversity of cloud tenants which make controlling the traffic and preventing intrusion difficult to achieve. We should enhance the security of networks in cloud computing by applying intrusion detection

systems which are capable of detecting sudden changes in traffic as fast as possible.

In our work, another dynamical method based on catastrophe theory is presented to detect anomalies in a cloud network. A damping coefficient is introduced for controlling the effectiveness of self-similarity factor. Entropy is used as disorderness factor versus self-similarity in control parameters of cusp catastrophe theory. To reduce the impact of weights through the time because of the dynamic nature of cloud traffic, the exponential moving average is applied. To evaluate the performance of our approaches, we consider DARPA datasets and compare the results with Xiong et al. catastrophe model. The results depict that our approach grounded in modified catastrophe theory is more effective in DR, FPR, misclassification, and F-measure rates to detect cloud network anomalies. We indicate that our randomness method based on entropy is two times faster than what Xiong et al. preferred. Different size of sliding window is applied and the maximum sensitivity is caught in window size 50. We prefer to repeat the experiments with new window size and compared the results in near future. As a future work, we would like to analyze a trade-off between the accuracy we achieved and the speed of detection. Also, we would like to test the impact of distinguishing protocol types in DARPA datasets and consider the differences. One of the ways which better indicates the performance of our model is to implement in a real environment and review the results. This work will be done as future work.

## Data Availability

The DARPA datasets used to support the findings of this study were used in previously reported studies and supplied by Lincoln Laboratory and could be available in https://www.ll.mit.edu/r-d/datasets.

## Conflicts of Interest

We declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was supported by the Iranian Research Organization for Science and Technology (IROST), Ministry of Science, Research & Technology (MSRT). We thank technology and communication center of IROST who provided a server that greatly assisted the research.

## References

[1] S. Khan, A. Gani, A. W. Abdul Wahab et al., "Towards an applicability of current network forensics for cloud networks: a SWOT analysis," *IEEE Access*, vol. 4, pp. 9800–9820, 2016.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Information Technology Laboratory, 2011.

[3] C. Gong, J. Liu, Q. Zhang et al., "The characteristics of cloud computing," in *Proceedings of the 39th International Conference on Parallel Processing Workshops (ICPPW '10)*, pp. 275–279, San Diego, California, Calif, USA, 2010.

[4] W. Xiong, H. Hu, N. Xiong et al., "Anomaly secure detection methods by analyzing dynamic characteristics of the network traffic in cloud communications," *Information Sciences*, vol. 258, pp. 403–415, 2014.

[5] J. Jabez and B. Muthukumar, "Intrusion detection system (IDS): anomaly detection using outlier detection approach," *Procedia Computer Science*, vol. 48, pp. 338–346, 2015.

[6] B. L. Dalmazo, J. P. Vilela, and M. Curado, "Online traffic prediction in the cloud: a dynamic window approach," in *Proceedings of the International Conference on Future Internet of Things and Cloud (FiCloud '14)*, Barcelona, Spain, August 2014.

[7] P. G. Kumar and N. Pandeeswari, "Adaptive neuro-fuzzy-based anomaly detection system in cloud," *International Journal of Fuzzy Systems*, vol. 18, no. 3, 2016.

[8] X. Zhao and W. Zhang, "An anomaly intrusion detection method based on improved k-means of cloud computing," in *Proceedings of the 6th International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC '16)*, pp. 2519–2523, July 2016.

[9] B. L. Dalmazo, J. P. Vilela, and M. Curado, "Performance analysis of network traffic predictors in the cloud," *Journal of Network and Systems Management*, vol. 25, no. 2, pp. 290–320, 2017.

[10] B. L. Dalmazo, J. P. Vilela, and M. Curado, "Online traffic prediction in the cloud," *International Journal of Network Management*, vol. 26, no. 4, pp. 269–285, 2016.

[11] A. H. Bhat, S. Patra, and D. Jena, "Machine learning approach for intrusion detection on cloud virtual machines," *International Journal of Application or Innovation in Engineering and Management*, vol. 2, no. 6, 2013.

[12] S. Fu, J. Liu, and H. Pannu, "A hybrid anomaly detection framework in cloud computing using one-class and two-class support vector machines," in *Advanced Data Mining and Applications*, pp. 726–738, Springer, Berlin, Germany, 2012.

[13] F. Zhao and H. Jin, "Automated approach to intrusion detection in VM-based dynamic execution environment," *Computing and Informatics*, vol. 31, no. 2, 2012.

[14] N. Pandeeswari and G. Kumar, "Anomaly detection system in cloud environment using fuzzy clustering based ANN," *Mobile Networks and Applications*, vol. 21, no. 3, pp. 494–505, 2016.

[15] Y. Chen, B. Yang, and Q. Meng, "Small-time scale network traffic prediction based on flexible neural tree," *Applied Soft Computing*, vol. 12, no. 1, pp. 274–279, 2012.

[16] H. Z. Moayedi and M. Masnadi-Shirazi, "ARIMA model for network traffic prediction and anomaly detection," in *Proceedings of the International Symposium on Information Technology (ITSim '08)*, vol. 4, pp. 1–6, 2008.

[17] A. Waskita, H. Suhartanto, and L. T. Handoko, "A performance study of anomaly detection using entropy method," in *Proceedings of the International Conference on Computer, Control, Informatics and its Applications (IC3INA '16)*, October 2016.

[18] E. Menahem, A. Schclar, L. Rokach, and Y. Elovici, "XML-AD: detecting anomalous patterns in XML documents," *Information Sciences*, vol. 326, pp. 71–88, 2016.

[19] F. Iglesias and T. Zseby, "Analysis of network traffic features for anomaly detection," *Machine Learning*, vol. 101, no. 1-3, pp. 59–84, 2015.

[20] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, pp. 708–713, 2015.

[21] W. Xiong, N. Xiong, L. T. Yang et al., "Network traffic anomaly detection based on catastrophe theory," in *Proceedings of the IEEE Globecom Workshops*, pp. 2070–2074, January 2011.

[22] W. Xiong, N. Xiong, L. T. Yang, J. H. Park, H. Hu, and Q. Wang, "An anomaly-based detection in ubiquitous network using the equilibrium state of the catastrophe theory," *The Journal of Supercomputing*, vol. 64, no. 2, pp. 274–294, 2013.

[23] R. Thom, "Structural stability, catastrophe theory, and applied mathematics," *SIAM Review*, vol. 19, no. 2, pp. 189–201, 1977.

[24] J. W. Robbin, "Thom's catastrophe theory and zeeman's model of the stock market," *Chaos and Complexity Seminar*, 2013.

[25] H. J. Sussmann and R. S. Zahler, "Catastrophe theory as applied to the social and biological sciences: a critique," *Synthese*, vol. 37, no. 2, pp. 117–216, 1978.

[26] S. Qin, J. Jimmy Jiao, S. Wang, and H. Long, "A nonlinear catastrophe model of instability of planar-slip slope and chaotic dynamical mechanisms of its evolutionary process," *International Journal of Solids and Structures*, vol. 38, no. 44-45, pp. 8093–8109, 2001.

[27] E. C. Zeeman, "Catastrophe theory," *Scientific American*, vol. 234, no. 4, pp. 65–83, 1976.

[28] M. Golubitsky, "An introduction to catastrophe theory and its applications," *SIAM Review*, vol. 20, no. 2, pp. 352–387, 1978.

[29] X. Wang and B.-X. Fang, "An exploratory development on the Hurst parameter variety of network traffic abnormity signal," *Journal of Harbin Institute of Technology*, vol. 37, no. 8, pp. 1046–1049, 2005.

[30] D. G. Alciatore, *Introduction to Mechatronics and Measurement Systems*, McGraw Hill, New York, NY, USA, 4th edition, 2012.

[31] M. Rao and H. Qiu, *Process Control Engineering: A Textbook for Chemical, Mechanical and Electrical Engineers*, CRC Press, Boca Raton, Fla, USA, 1993.

[32] F. Klinker, "Exponential moving average versus moving exponential average," *Mathematische Semesterberichte*, vol. 58, no. 1, pp. 97–107, 2011.

[33] S.-J. Horng, M.-Y. Su, Y.-H. Chen et al., "A novel intrusion detection system based on hierarchical clustering and support vector machines," *Expert Systems with Applications*, vol. 38, no. 1, pp. 306–313, 2011.

[34] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Information Sciences*, vol. 177, no. 18, pp. 3799–3821, 2007.

[35] Lincoln Laboratory, "The 1999 DARPA intrusion detection datasets," 1999, https://www.ll.mit.edu/r-d/datasets.

[36] S. M. Shinners, *Modern Control System Theory and Design*, Wiley InterScience Publication, 2nd edition, 1992.

[37] B. L. Dalmazo, J. P. Vilela, P. Simoes, and M. Curado, "Expedite feature extraction for enhanced cloud anomaly detection," in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium, NOMS '16*, July 2016.

[38] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.