

Soft-computing-based false alarm reduction for hierarchical data of intrusion detection system

International Journal of Distributed
Sensor Networks
2019, Vol. 15(10)
© The Author(s) 2019
DOI: 10.1177/1550147719883132
journals.sagepub.com/home/dsn


Parminder Singh¹ , Sujatha Krishnamoorthy², Anand Nayyar³,
Ashish Kr Luhach⁴  and Avinash Kaur¹

Abstract

A false alarm rate of online anomaly-based intrusion detection system is a crucial concern. It is challenging to implement in the real-world scenarios when these anomalies occur sporadically. The existing intrusion detection system has been developed to limit or decrease the false alarm rate. However, the state-of-the-art approaches are attack or algorithm specific, which is not generic. In this article, a soft-computing-based approach has been designed to reduce the false-positive rate for hierarchical data of anomaly-based intrusion detection system. The recurrent neural network model is applied to classify the data set of intrusion detection system and normal instances for various subclasses. The designed approach is more practical, reason being, it does not require any assumption or knowledge of the data set structure. Experimental evaluation is conducted on various attacks on KDDCup'99 and NSL-KDD data sets. The proposed method enhances the intrusion detection systems that can work with data with dependent and independent features. Furthermore, this approach is also beneficial for real-life scenarios with a low occurrence of attacks.

Keywords

Anomaly detection, intrusion detection system, hierarchal data, soft computing, classification

Date received: 20 June 2019; accepted: 28 August 2019

Handling Editor: Iftikhar Ahmad

Introduction

The rapid development of network systems has a big threat from the intrusions. Intrusion detection systems (IDS)^{1,2} are widely used to mitigate the various types of attacks. Broadly, IDS can be classified into three categories: network-based intrusion detection systems (NIDSs), distributed intrusion detection systems (DIDSs), and host-based intrusion detection systems (HIDSs). The NIDS's objective is to defend against the threats related to network, HIDS's aim is to figure out the local system anomalies, and DIDS is responsible for improving the performance based on IDS agents' information.

The detection methods for these IDSs are of three types: signature-based detection, anomaly-based detection, and hybrid detection. An anomaly-based IDS³ can

figure out abnormal network/system behavior from the comparison of normal profile with the current system. If deviation is found beyond a certain threshold, then the event is declared as abnormal. A signature-based IDS⁴ identifies the attack by comparing the stored signatures with the current incoming event. Here, the

¹Lovely Professional University, Phagwara, India

²Department of Computer Science, Wenzhou-Kean University, Wenzhou, China

³Duy Tan University, Da Nang, Vietnam

⁴The Papua New Guinea University of Technology, Lae, Papua New Guinea

Corresponding author:

Sujatha Krishnamoorthy, Department of Computer Science, Wenzhou-Kean University, Wenzhou 325060, Zhejiang, China.
Email: sujatha@wku.edu.cn



signature is a description of some features for a known attack. An alarm would be triggered by the absolute match. The combination of signature-based detection and anomaly-based detection is called a hybrid IDS.^{5,6}

The IDS effectiveness can be calculated from the probability of a positive detection on actual anomaly occurrence. In the application domain, the IDS effectiveness is considered from its capability to lower down the false-positive rate (FPR) instead of increasing the true-positive rate (TPR).⁷ Therefore, the crucial challenge is to lower down FPR with a minimum decrement in TPR to maintain the detection quality at a practical level.

Generally, the normal class is comprised of various disjoint subclasses. The application protocol (Simple Network Management Protocol (SNMP), File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), etc.), transport protocol (User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Internet Control Message Protocol (ICMP), etc.), and other network communication traffic have various subclasses. The accounting of subclasses can increase the performance of IDS.^{8,9} After that the subclasses are decomposed into more specific subclasses, which is mentioned in various studies.^{10,11} The user-defined multi-level hierarchy has the potential to maximize the accuracy of the anomaly detection system. These research studies are efficient to improve the anomaly detection system for multi-level subclass hierarchy. However, the limitation of these methods is of their practical implementation because of certain reasons.

- *Users required expert knowledge.* The users need to explore the hierarchal data structure to define the subclasses. They need to know the working of IDS thoroughly, which needs training and regular interactive session.
- *Specific for certain IDS.* Most of the IDS are applicable for specific domain only. It is critical to choose the most relevant IDS where the system is comprised of different domains.

So the fundamental issue is to minimize the FPR of IDS in practical solutions. In this study, we develop an autonomous system called soft-computing-based anomaly detection (SCAD) for hierarchical data.

SCAD generates fundamentally distinct “points of view” for ordinary information to which are compared with test cases. With contextual anomalies in comparison to the whole data set as ordinary, but an anomaly provided some context,¹² we have an especially reverse notion that we called contextual inlier. In this regard, we have a special focus on the contextual anomalies. Some test cases are normal but appear to be anomalous to few information subclasses. The principle of our technique is that only after comparing an anomaly to

every subclass, it should be declared normal or anomalous. SCAD takes all subclasses into account, which strengthens the proposed technique in efficient anomaly detection. Our contribution is as follows:

1. We designed a general-purpose IDS to improve the FPR, where the data could be represented with a hierarchical structure.
2. SCAD is more practical than most of the previously developed techniques which benefit from a hierarchical framework due to the following reasons:
 - (a) It is a generic approach for an anomaly detection system which is designed to work with various black-box methods.
 - (b) The exploration of hierarchy algorithm is fully automated and the user does not need to be domain-specific or to create an algorithm for the hierarchy.
3. We evaluated the SCAD-RNN and existing IDSs capacity to increase the efficiency with benchmarking data sets.

Related work

Recent IDS^{13–16} extracts the characteristics from payload packets to train the one-class classifiers which are capable of identifying abnormal network traffic. Mirsky et al.¹⁵ suggested Kitsune, the devised model fetch the implicit traffic on the network at runtime based on contextual characteristics with a tiny storage footprint to build one-class auto-encoders automatically. They demonstrate that Kitsune can almost tackle the problem well, but offline or batch IDSs are even better in some instances. The extracting functions for the IDSs suggested by Nguyen et al.¹³ and Duessel et al.¹⁴ constitute typical octets and enable syntactic connect to the communication protocol for incorporation. Here, the author extracted functional vector and used in both the IDS with the one-class support vector machine (OCSVM) kernel (radial basis function (RBF)) and other IDSs^{9,15} in a similar way. Ying et al.¹⁷ presented an IDS for cloaking attack with clock skew-based IDS as a solution for controller area network.

The above IDSs supposed that information can be processed separately. The distinct information in the documents may be linked, which can provide an effective context for anomaly detection as per their relative order of occurrence.¹⁸ Therefore, sequential IDSs are another significant category of IDSs. IDS operating on byte sequences in a packet or on packet continuation instead separate function vectors. Recently, this type of IDS has become more important because of the ever-increasing abundance of sequential information in numerous real-life scenarios. Gupta et al.¹⁹ analyzed

that large-scale and Internet processing requirements usually make it more demanding to analyze anomalies than in non-sequential information. In Wang et al.,²⁰ Swarnkar and Hubballi,²¹ and Wang and Stolfo,²² the sequential IDSs are suggested for the detection of anomalous sequences containing subsections of which their inherent frequency is unexpected. The latest is an IDS known as Rangegram,²³ which effectively produces a Normality model within ordinary sequences of the high-order n -grams with a maximum and minimum range of frequency. In a test sequence, the author analyzed that the n -grams increased from the normal range in case network intrusion.

The sequential IDSs like Tian et al.,²³ Michlovský et al.,²⁴ and Haddadi²⁵ analyzed based on the sequences they contain to identify ordinary sequences with much difference. These IDSs use a version of the SSK²⁶ to implicitly map sequence into a large function space where distances between sequences are equivalent. For instance, in a semi-supervised situation, Tian et al.²³ applied the SSK along with OCSVM for intrusion detection in computer scheme that is described in abnormal system subsequences. Because of their capacity to combine various base detectors with a view to optimizing the bias-variance trade, anomaly detection ensembles have grown popular in latest years.^{13,18,27,28} The IDSs suggested in Perdisci et al.²⁸ and Nguyen et al.¹³ have been designed with the support of a set of OCSVM basic detectors, trained with various function subsets to obtain distinct information representations and thus to improve detection precision. Likewise, auto-encoder detectors in the Kitsune¹⁵ set model distinct function subsets of ordinary network packets in their auto-encoders. In this research, the mentioned technique is the detection of anomalies, even though the base detectors of our ensemble are not represented by distinct types of ordinary information, as opposed to the ensembles outlined in literatures,^{13,15,28,29} and are driven by the hierarchy. Furthermore, our approach can be implemented automatically to any soft-computing technique and does not just concern a particular recurrent neural network (RNN) type. The user can use any soft-computing-based technique with the first stage of the proposed SCAD model.

The existing techniques applied the single hierarchical system to enhance intrusion detection based on anomalies. In order to enhance detection precisely under a controlled framework, Peddabachigari et al.³⁰ employed the support vector machines (SVMs), decision trees (DT), and a fusion classifier of DT-SVM. The group of three incorporated into a hierarchy of classifier. Kim et al.⁹ designed a hybrid IDS by integrating misuse detection with anomaly detection. The author broke down ordinary information into a hierarchical subclass tree and OCSVM is created for each subclass. These techniques need to attack information

labeled. However, it is challenging to break down the ordinary data into subclasses only on account of their resemblance to recognized attack classes. However, our strategy requires no marked information and records a more natural decay based exclusively on the similarity between ordinary data cases. In combination with the unchecked clustering algorithms, Xiang et al.³¹ have created a hierarchical classification³² process that improves the initial training label as per original data set composition. However, user assumptions required the clustering algorithm needs to define the user assumptions for the distribution detail of data. The initialization of the number of classes required is a costly numerical optimization in the convergence process. The hierarchical exploration is conducted based on correlations clustering (CC).³³ As CC has been designed to autonomously discover the maximum amount of classes, it automatically decomposes the ordinary class into subclasses by our hierarchical clustering algorithm. The method defined in this process is not a hierarchical approach for classification since every hierarchical node is not being taken as a decision-maker compared with route node;³² indeed, instances which are ultimately identified as anomalies by our method will have been passed through and checked at all nodes. Although, the cloud-based techniques are designed for load balancing by enhancing bat algorithm, Luhach and colleagues^{34,35,36} presented an effective framework based on service-oriented architecture (SoA) for e-commerce applications and Internet-of-things (IoT) applications. The benefits of hierarchical anomaly detection are also reported by Robinson et al.¹¹ The anomaly detection process is the sum of unsupervised learning and discovery of hierarchical data in the training set. Similar to the proposed approach, this model enables the user to select anomalies with a wide variety of definitions.

The structure is meant for sequential data processing which can limit the new anomalies exploration. As opposed, the sequential as well as non-sequential information can be processed by our structure. Therefore, the proposed model is indifferent to the information type and can be helpful in every field of abnormal identification where the information is hierarchical, as long as it is an equally similar example, graph data can also be processed on it.²⁷ In addition, the strategy of Robinson and other parties need consumers who are not necessarily known to enter semantic hierarchy, such as moment (e.g. days, months and years) or place hierarchy (e.g. state, city, town), particularly in a dynamic situation. Of course, this limits their usability for customers who have expert knowledge of the hierarchical interactions in the information and the method of identification of anomalies. Therefore, the customer should provide the connections between information characteristics, ontology, and the interaction between

anomaly detectors to detect any anomalies. Instead, we do not require customers to be domain specialists in our system. However, if consumers have a helpful knowledge of the data set, they can transmit in the form of pairwise features. Our approach has strong evidence to become more practical and applicable to the existing black-box-based IDS.

Deep learning, a machine learning branch, has become increasingly common in the latest years. The performance of a deep learning approach is far better than traditional approaches in intrusion detection. In Javaid et al.,³⁷ the authors detect the anomalies on the basis of the neural network, and the experimental results demonstrate that deep learning can be used to detect anomalies in networks. In Tang et al.,³⁸ the authors applied the self-taught learning (STL) with deep learning and propose in an NIDS. The method is shown to be more efficient when comparing its performance with those found in past research. However, the authors focused on the capacity of deep learning to reduce features. For prior-training, it primarily utilizes deep learning techniques and conducts classification through the traditional supervisory model. Applying the deep learning technique to directly conduct classification is not prevalent, and very few articles are researched on multi-class classification. RNNs are regarded to be reduced-size neural networks, according to Sheikhan et al.³⁹ The author recommended documenting a three-layer RNN architecture with the input of 41 characteristics and output of four categories of intrusion for IDS based on misuse. However, layer nodes are partially connected, features of high-dimension did not study for deep learning for hierarchical data with binary classification. Deep learning techniques have flourished rapidly with the ongoing growth computing power and big data. It has been commonly used in multiple fields. RNN used for intrusion detection via deep learning method presented in this document followed similar thinking. We use the hierarchical exploration and RNN-based model for classification based on static and dynamic training rather than pre-training only. In addition, NSL-KDD data set has used with the separate training set and testing set to assess their performance in identifying binary and multi-class network intrusions and the comparison performed with Naive Bayesian, J48, artificial neural network (ANN), SVM, Random Forest, and other machine learning techniques mentioned in previous studies.

Anomaly detection in hierarchical data

The data sets with meaningful hierarchical structure can be used to minimize the FPR in the IDS. Here, the SCAD for IDS is presented for hierarchical data. This process is accomplished in two steps: (1) explore the

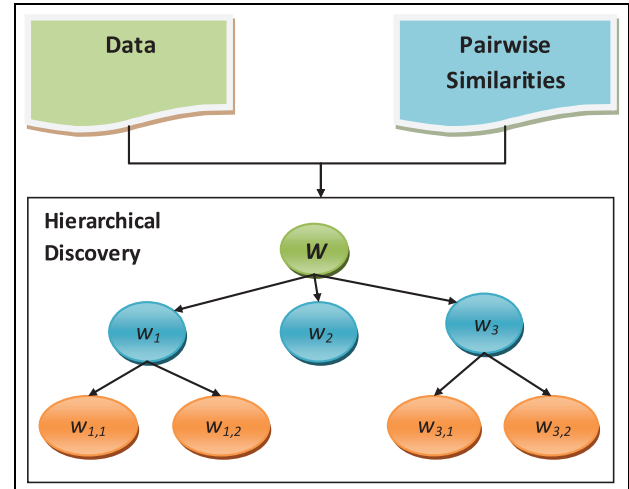


Figure 1. Exploration of level of hierarchical structure.

hierarchical structure of data and (2) deep-learning-based anomaly detection to reduce the FPR in a discovered hierarchical structure.

Exploration of hierarchical structure

In this stage, the normal data W is considered for training to explore the hierarchy structure of the data set (see Figure 1). To explore the level of the hierarchy, divisive hierarchical clustering technique⁴⁰ has been applied.

Since no prior information is available for the data, we conduct the CC³³ to classify the number of subsets in the data set $W = y_1, y_2, \dots, y_n$. The CC does not need to provide the number of clusters, rather it needs to provide the correlation values between the instances. The positive correlation between y_i and y_j indicates that they should assign the same subset, otherwise attempt to assign a different subset.

We employed the settings of anomaly detection as semi-supervised learning;¹⁸ in this, normal data instances W are provided as the training data set. “IS-A” hierarchy^{32,41} applied to unearth the hierarchy in the form of a tree structure.

In CC, we used supervised classification at the first stage to define the available labeled training objects. After that, we insert the unlabeled clusters in the base of the representative cluster on similarity calculation. The input to Algorithm 2 is a pairwise similarity calculated as per Algorithm 1.

The similarity of y_i and y_j is the sum of per attribute similarity for total attributes. It is calculated as per equation (1)

$$\text{sim}(y_i, y_j) = \sum_{j=1}^d s(y_i, y_j) \quad (1)$$

Algorithm 1. The pseudocode for pairwise similarity matrix

```

1.  $\forall i, j: \text{sim}[i, j] \leftarrow 0$ 
2. for each  $w$  in  $W$  do
3.   for  $j = 1$  to  $d$  do // as per equation (1)
4.      $\text{total} \leftarrow \text{sum}(y_i, y_j)$ 
5.      $\text{sim}(y_i, y_j) \leftarrow \text{total}$ 
6.   for each  $y_i, y_j$  do // as per equation (2)
7.     if  $\text{sum}(y_i, y_j) > \text{sum}(y_m, y_n)$  then
8.        $\text{sim}(y_i, y_j) \leftarrow 1$ 
9.     else
10.       $\text{sim}(y_i, y_j) \leftarrow 0$ 

```

The pairwise similarity matrix values are calculated with equation (1) and input to Algorithm 2. But, while deciding the similarity function, we follow equation (2), y_i and y_j should follow the same subclass more than y_m and y_n are believed to belong to the same subclass. The user needs to decide the similarity function on the basis of their confidence in correct implication

$$\text{sim}(y_i, y_j) = \begin{cases} 1, & \text{if } s(y_i, y_j) > s(y_m, y_n) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The objective of CC is to achieve the maximum pairwise agreement with the calculation of the highest value of agreement objective function $AGM_{CM^w}(x)$

$$\sum_{CM_{ij} \in CM^{w+}} |CM_{ij}| \cdot x_{ij} + \sum_{CM_{ij} \in CM^{w-}} |CM_{ij}| \cdot (1 - x_{ij}) \quad (3)$$

where CM^{w+} is a positive correlation and CM^{w-} represents the negative correlation values in CM_{ij} . $AGM_{CM^w}(x)$ function is a set where $x = \{x_{ij}\}_{i \neq j \wedge i, j = 1, 2, \dots, l}$ of rate for

$$x_{ij} = \begin{cases} 1, & y_i \wedge y_j \text{ are assigned to the same subset} \\ 0, & y_i \wedge y_j \text{ are assigned to different subset} \end{cases} \quad (4)$$

The assignment of x for $AGM_{CM^w}(x)$ function performs the portioning of data set w . The CC method choose the maximum value of $AGM_{CM^w}(x)$ for optimal portioning. In this study, we adopt the heuristics mentioned in Gal-Oz et al.⁴² because it has the complexity $O(l^2)$ in worst case and it is a greedy algorithm. In practice, this is more efficient than AutoClass algorithm⁴³ and K-means algorithm.⁴⁴

Considering the set W and similarity matrix SM^w , Algorithm 3 used to design the hierarchy λ . In the first step, W is input to the root node in a hierarchical structure with learned detection model of IDS. Algorithm 2 is used to evaluate correlation matrix CM^w from similarity matrix SM^w . Algorithm 3 is a recursive invocation and obtained the sub-hierarchy at λ_i^1 . The sub-hierarchies are isolated from each other, this process

Algorithm 2. The pseudocode to calculate correlation matrix.

```

1. Input:  $SM$  — a  $n \times n$  matrix of similarity values, a data set  $W = y_1, y_2, \dots, y_n$ 
2. Output: a  $n \times n$  correlation matrix  $CM$  for every pair of  $W$  data set.
3. procedure Calculate  $CM$ 
4.   Matrix  $CM[n, n] \leftarrow \text{null}$ 
5.   Vector  $V[n] \leftarrow \text{null}$ 
6.   loop:
7.     for  $i = 1$  to  $ndo$ 
8.        $V_i \leftarrow \frac{\sum_{j=1, j \neq i}^n S_{ij}}{n-1}$ 
9.     loop:
10.    for  $i = 1$  to  $n$  do
11.      for  $j = 1$  to  $n$  do
12.         $CM_{ij} \leftarrow S_{ij} - \frac{V_i + V_j}{2}$ 
13.    return  $CM$ 

```

Algorithm 3. The pseudocode for exploration of hierarchy

```

1. Input:  $SM$  — a  $n \times n$  matrix of similarity values, a data set  $W = y_1, y_2, \dots, y_n$ 
2. Output:  $\lambda$  — the hierarchy of  $W$ .
3. procedure ExploreHierarchy
4.    $\lambda^0 \cdot \text{dataset} \leftarrow W$ 
5.    $\lambda_1^0 \leftarrow \text{SCAD.fit}(W)$ 
6.    $CM^w \leftarrow \text{CalculateCM}(SM^w)$ 
7.    $P \leftarrow \text{CorrelationClustering}(CM^w)$ 
8.   for Each subset  $w_i \subset P$  do
9.      $SM^{w_i} \leftarrow \text{DeriveSubMatrix}(D_i)$ 
10.     $\lambda_i^1 \leftarrow \text{ExploreHierarchy}(w_i, SM^{w_i}, \text{SCAD})$ 
    return  $\lambda$ 

```

can be accomplished parallel which can increase the speed of design.

SCAD

The IDS is usually working in two phases. The IDSs which are working on *fit* and *predict* procedures are known as a “block box” technique. After exploring the hierarchy of data set, next, we applied the soft-computing-based technique to classify the test instances.

Numerical conversion and normalization. The NSL-KDD data set has a total 41 observations, out of which 3 are non-numeric features and 38 are numeric features. The proposed SCAD technique used RNN, which takes the numeric values. We must convert non-numeric values in the form of numeric values such as “service,” “protocol types,” and “flag.” The “protocol type” is of three types ICMP, UDP, and TCP, we encode them into binary form vector (0,0,1), (0,1,0), and (1,0,0), respectively.

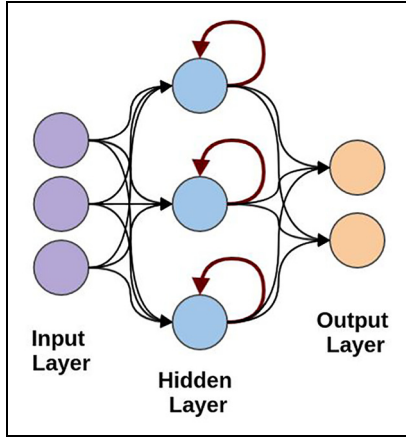


Figure 2. Recurrent neural networks.

Similarly, we convert the “flag” and “service” types in 122 and 11 attributes, respectively. Furthermore, the numerical values have a large scope in the data set, there is a large difference between the highest and lowest values. Furthermore, we normalize the values as per the following equation

$$y_i = \frac{y_i - \text{Min}}{\text{Max} - \text{Min}} \quad (5)$$

In recent neural networks, the most important research is to be done for inputs, output units, and concealed units. The RNN model has information from the entrants in one-way to the cache units essentially. The summary of information from the preceding time clock unit to the current time clock unit is shown in Figure 2. One-way information is carried out by RNN.

RNN. Hidden devices can be seen as the entire network storage, which remembers the end of data. We can find that the RNN will embody the profound learning when we unfold as shown in Figure 3. For monitored classification, learning an RNN strategy can be used as shown in Figure 2. The directional loop is implemented by current neural networks and remember the prior data to be applied on present output. This is the main difference in traditional and fuzzy neural network (FNN).

The previous output also has to do with the present output sequence, the nodes are no longer connected but concealed states has a link with each other. In addition to the output of the input layer as well as the output of the final concealed layer, we can look at hidden units as the entire network storage reminiscent of the end-to-end information. We can say that when we unfold RNN, it embodies profound learning. An approach to RNNs can be used for monitoring classification learning. Recent neural networks have initiated a directional

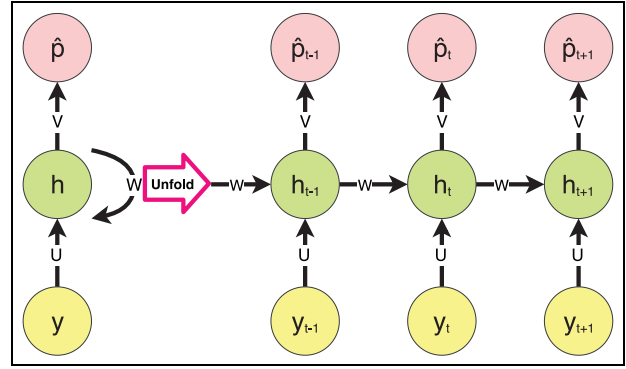


Figure 3. The unfolded recurrent neural network.

belt, it has an essential distinction from conventional feed-forward neural networks, able to record and apply previous information. This represents the key distinction. The output from the previous iteration is connected with the sequence of current output. Hidden layers between the nodes are not connected; otherwise, the hidden layer has connections. The SCAD-RNN model working is shown in Figure 4.

Methodology of SCAD-RNN model. It is evident that the SCAD-RNN model is the combination of two modules: forward propagation and back propagation. In forward propagation, the output value is calculated, and in back propagation, the output value is deployed to pass the residuals to update the weights, which is similar to the formation of ordinary neural network. The RNN model present in the methodology could further replace with other black-box methods.

As per Figure 3, we applied the unfolded RNN. To formalize the standard RNN, the training samples are $y_i (i = 1, 2, \dots, n)$, the hidden states sequence $h_i (i = 1, 2, \dots, n)$, and prediction sequence $\hat{p}_i (i = 1, 2, \dots, n)$. Furthermore, the input-to-hidden weight matrix is defined with W_{hy} , the hidden-to-hidden weight matrix is defined with W_{hh} , the hidden-to-output weight matrix is defined with W_{ph} , and biases are b_h and b_p vectors.⁴⁵ The *sigmoid* is defined as e , an activation function. The *SoftMax* function evolved in g classification function.

As per Figure 4 and Martens and Sutskever,⁴⁵ Algorithm 4 is a pseudocode for forward propagation. Algorithm 5 is a pseudocode for weight update.

A single training pair (y_i, p_i) defined as $f(\theta) = LV (p_i : \hat{p}_i)$ in RNN has the association with objective function,⁴⁵ the distance function LV calculates the deviation from actual labels (p_i) to predicted labels (\hat{p}_i) . The learning rate is defined as b_e and current iteration

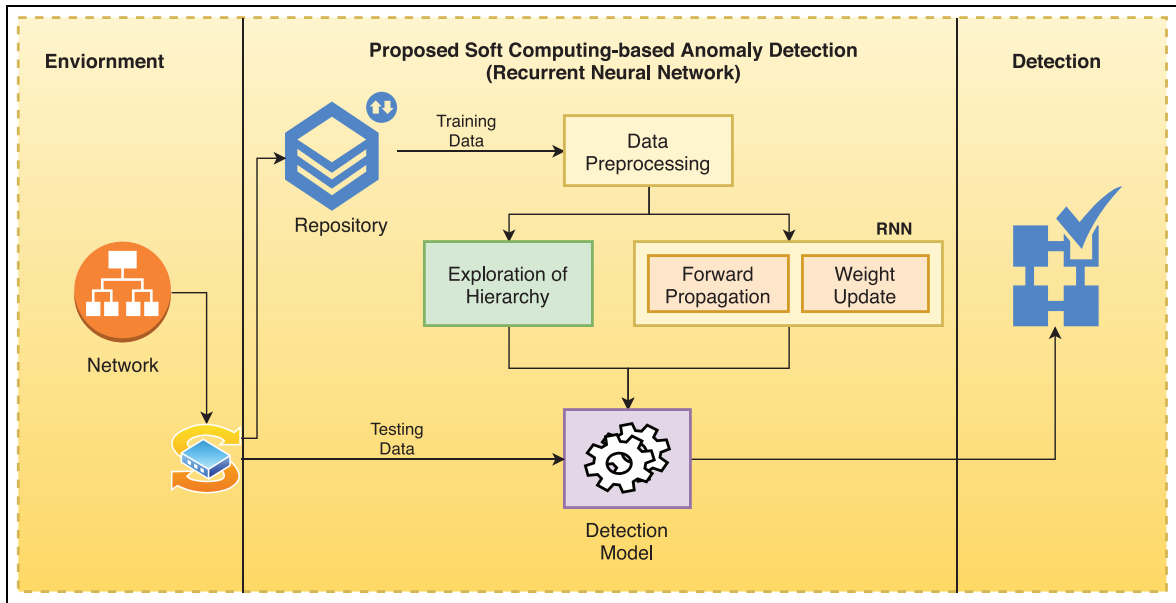


Figure 4. Proposed SCAD-RNN intrusion detection system.

Table 1. List of NSL-KDD and KDDCup'99 data set normal and attack classes.

Class	NSL-KDD and KDDCup'99 subclasses
Normal	Normal traffic
DoS	udpstorm, processtable, neptune, mailbomb, land, back, apache2, smurf, pod, teardrop
Probe	saint, mscan, nmap, ipsweep, portsweep, satan
R2L	guess_passwd, phf, xlock, worm, snmpgetattack, imap, snmpguess, sendmail, xsnoop, multihop, warezmaster, warezclient, ftp_write, named, spy
U2R	sqlattack, ps, xterm, rootkit, perl, loadmodule, httptunnel, buffer_overflow
Total	40 subclasses

DoS: Denial of Services; R2L: Root to Local; U2R: User to Root attack.

presented with k . The $p_i (i = 1, 2, \dots, m)$ defined the labels in sequence and k is defined as the present iterations.

The detection model is shown in Figure 4 is the combination of hierarchical exploration and RNN model. At this stage, the hierarchical data λ has been discovered. The test data t and λ are used in the detection model. This model checks the node's data set in the λ levels, where t is normal or not. Once t is found as an anomaly, the process performed recursively at each child node of the sub-tree. This process could run parallel or with GPU acceleration because of the isolation of data set in different nodes and levels to find the perfect subclass of the intrusion. The RNN model helps to discover the new attacks and grow/update the tree structure at runtime.

Data set

In the field of intrusion detection studies, the 2009 produced NSL-KDD^{46,47} data set is widely used. This is a

benchmarking data set used by most of the authors.^{48–50} The data set is vital in solving the inherent redundant records issue in KDDCup'99 data set. The most frequent record does not favor the classifier in training and testing data set. The KDDTrain⁺ covers the training data set and testing data set are KDDTest⁺ and KDDTest-21. Table 1 shows the different types of attacks and normal records in the data set. The KDDTest⁺ subset is designed named as KDDTest-21, which is more difficult to classify. The data set is classified in four categories based on the types of attacks: U2R (User to Root attack), R2L (Root to Local), Probe (Probing attack), and DoS (Denial of Services). The testing data contain some attack which is disappearing in the training set, which strengthens the testing process of IDS.

Evaluation

For the measurement of the results of the SCAD-RNN model, the largest performance indicator is accuracy,⁵¹

Algorithm 4. The pseudocode of forward propagation

```

1. Input: Data set  $W = y_1, y_2, \dots, y_n$ 
2. Output:  $\hat{p}_i$ 
3. procedure ForwardPropagation
4. loop:
5.   for  $j = 1:n$  do
6.      $t_j = W_{hyj} + W_{nhhi-1} + b_h$ 
7.      $h_j = \text{sigmoid}(t_j)$ 
8.      $s_j = W_{phhi} + b_x$ 
9.      $\hat{p}_j = \text{SoftMax}(s_j)$ 
10.  return  $\hat{p}_i$ 

```

used in our model. We implement the detection rate (DR) and FPR concerning the accuracy. The true positive (TP) corresponds to the correctly rejected documents, and it refers to the number of anomaly documents recognized as an anomaly. This is the equivalent of erroneously refused false positive (FP), which indicates the number of ordinary documents recognized as an anomaly. The true negative (TN) corresponds to the properly recognized ones and refers to the number of ordinary logs recognized as usual. The false negative (FN) corresponds to the wrongly admitted ones and it refers to the number of anomaly records recognized as usual. The confusion matrix is calculated for binary and multi-class classification. Our note is as follows:

Accuracy. It is calculated as a percentage of the total number of records versus classified TP and TN data. It can be calculated as per the following equation

$$Accuracy = \frac{TN + TP}{TP + FP + FN + TN} \quad (6)$$

TPR. It is similar to the DR, it is calculated as a total number of anomalies versus data identified correctly. It can be calculated as per the following equation

$$TPR = \frac{TP}{TP + FN} \quad (7)$$

FPR. It is the percentage of several rejected records versus normal records in the data set

$$FPR = \frac{FP}{TN + FP} \quad (8)$$

Thus, the objective of this study is to reduce the FPR while maintaining the TPR.

Algorithm 5. The pseudocode of weight update

```

1. Input:  $(p_i, \hat{p}_i)$  ( $i = 1, 2, \dots, n$ )
2. Initialize  $\theta = \{W_{hy}, W_{hh}, W_{ph}, b_h, b_p\}$ 
3. Output:  $\theta = \{W_{hy}, W_{hh}, W_{ph}, b_h, b_p\}$ 
4. procedure WeightUpdate
5. loop:
6.   for  $j = k:l$  do
7.      $LV(p_i, \hat{p}_i) \leftarrow \sum_i \sum_j p_{ij} \log(\hat{p}_{ij}) + (1 - p_{ij}) \log(1 - \hat{p}_{ij})$ 
        $\triangleright$  Calculation of cross entropy
       between label value and output value
8.      $\delta \leftarrow dL/d\theta_j$ 
9.      $\theta_j \leftarrow \theta_j \eta + \delta_j$ 
        $\triangleright$  Computation of partial
       derivative with respect to  $\theta_j$ 

```

Experiment results and discussion

We have used one of the latest and widest profound frameworks in this study⁵² for deep learning in Python. The experiment conducted in a private notebook with an Intel Core i5-3210M CPU @ 2.50 GHz, 4 GB of memory. Two experiments for binary category (Normal, Anomaly) and 5-category classification such as Normal, R2L, DoS, Probe, and U2R have been conducted for performance analysis of the SCAD-RNN model. Contrasting experiments are conceived simultaneously to compare with other machine learning techniques. We contrasted the output with a naive Bayesian, ANN, random forest, multi-layered perceptron, support vector machines, and other machine learning techniques in binary classifications, as stated in Tavallae et al.⁴⁶ and Ingre and Yadav.⁵³ Likewise, we analyze the SCAD-RNN model's multi-class classification using the data set of NSL-KDD. By comparison, in the five-category classification, we studied the J48, naive Bayesian, SVM, random forest, multi-layer perceptron, ANN and support vector machine, and other machine learning models performance for intrusion detection. Finally, with the traditional methods, we combine the efficiency of proposed SCAD-RNN model. In addition, we build a used data set referred to as Sheikhan et al.³⁹ and Yin et al.⁵⁴ and compare the output with the SCAD-RNN method of reduced size.

Binary classification

In data preprocessing, the 41-dimensional characteristics have been mapped to 122-dimensional characteristics. In binary classification, the SCAD-RNN model has two output nodes and 122 input nodes. The epoch count is 100. In order to train this better pattern, allow 240, 120, 80, 60, and 20 hidden nodes. The learning rate is set as 0.5, 0.1, and 0.001, and then comply with the NSL-KDD data set classifications precision. The experiment result indicates that the learning rate and hidden nodes are directly related to the accuracy of the model.

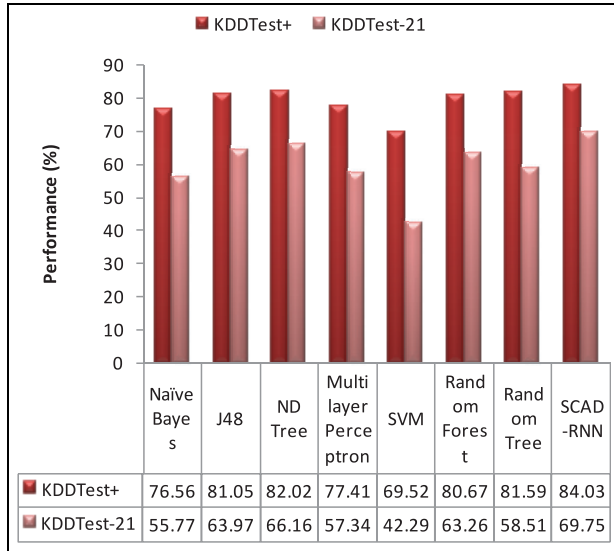


Figure 5. Binary classification comparison of SCAD-RNN and other models.

The KDDTest⁺ test set for the two-category experiment for classification. The confusion matrix is shown in Table 2. In this experiment, the proposed model gives higher efficiency with 0.1 learning rate and 80 hidden nodes. The results indicate that when 100 periods are provided for the KDDTest⁺ data set, SCAD-RNN operates at a good DR (84.03%). For KDDTest-21, we receive 69.75% performance.

The comparison has been conducted on various machine learning techniques such as Naive Bayesian, J48, Multi-layer Perceptron, Random Forest, Support Vector Machine, and other classification algorithms in Tavallae et al.,⁴⁶ and an algorithm in the ANN which is also 81.2% given in Ingre and Yadav.⁵³ All these findings are conducted on the same NSL-KDD benchmark data set. The proposed SCAD-RNN model is more efficient as compared to other binary classification models as shown in Figure 5.

Multi-class classification

It has been found from the experiment on KDDTest⁺ that the SCAD-RNN model has high accuracy as we set learning rate as 0.5 and hidden node as 80.

Comparison of proposed SCAD-RNN model with machine learning model such as Naive Bayesian, J48, Support Vector Machine, Multi-layer perceptron, and other with 10-layer cross-validation using Python libraries. The hierarchical model and RNN model uniformly discover the detection model. The combination of SCAD and RNN tested on the testing data. The performance of binary classification is better as compared to multi-class classification as shown in Figures 5 and 6.

Table 2. KDDTest⁺ confusion matrix for binary classification.

Actual class	Predicted Class	
	Normal	Anomaly
Normal	9467	244
Anomaly	3361	9472

Table 3. KDDTest⁺ confusion matrix for multi-class classification.

Actual class	Predicted Class				
	Normal	DoS	R2L	U2R	Probe
Normal	9412	80	2	6	211
DoS	1001	6257	113	0	87
R2L	2038	0	702	5	9
U2R	141	0	9	35	15
Probe	214	142	5	0	2060

DoS: Denial of Services; R2L: Root to Local; U2R: User to Root attack.

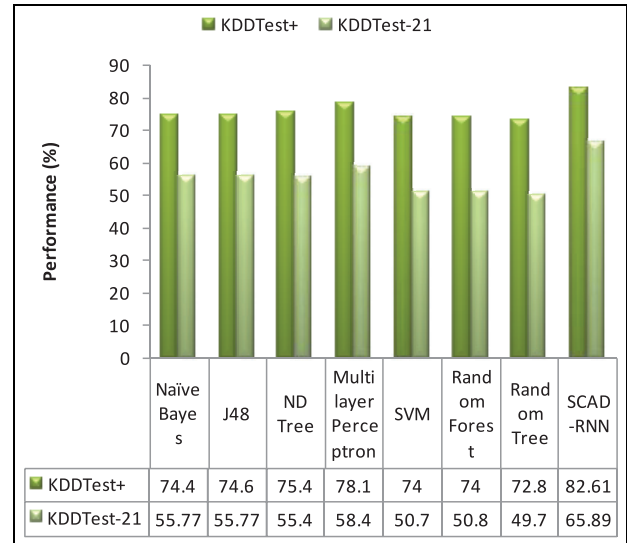


Figure 6. Multi-class classification comparison of SCAD-RNN and other models.

The SCAD-RNN confusion matrix for multi-class classification is shown in Table 3. The test demonstrates that for the KDDTest⁺ test set at 82.61% and for KDDTest-21 at 65.89% are better than those achieved using other machine learning models as mentioned in Figure 6. The proposed model performs better than the ANN algorithm,⁵³ which give the accuracy 79.9%. The FPR and TPR of normal and attacks are shown in Table 4.

Table 4. Multi-class classification evaluation metrics.

Type of class	TPR (%)	FPR (%)
Normal	96.82	0.03
DoS	83.90	0.16
Probe	25.49	0.74
R2L	17.5	0.83
U2R	85.09	0.14

DoS: Denial of Services; R2L: Root to Local; U2R: User to Root attack; TPR: true-positive rate; FPR: false-positive rate.

Performance of multi-class classification of SCAD-RNN with RNN of reduced size³⁹ with KDDCup'99 data set with same testing and training sets gives the following results given as follows. According to the experiment conducted, the SCAD-RNN model achieves the higher accuracy of 98.02% on the test set, which is higher than 94.1% accuracy achieved by Sheikhan et al.³⁹ The SCAD-RNN has a strong model with hierarchical exploration and soft-computing technique as compared to RNN with reduced size. The training time of the proposed model is higher, which can further reduce with GPU acceleration and parallel processing.

Conclusion and future scope

The proposed SCAD-RNN model has a powerful intrusion detection modeling capability and high precision in binary as well as multi-class classification. The hierarchy of data sets developed using CC. Furthermore, deep learning approach is applied to compare the test case with subclasses to decide whether the test case is anomaly or normal. Compared with traditionally classified techniques like naive Bayesian, J48, random forests, and SVM, the achievement is attained by greater accuracy rates and a small FPR, particularly under the KDDCup'99 and NSL-KDD data set classification. The model improves both the TPR for intrusion detection and the capacity to detect the intrusion class efficiently. In future, the potential research continues to reduce the time for training using GPU and parallel processing to prevent explosion and gradients removal studies to improve the classification efficiency of hierarchical discovery, long short-term memory (LSTM), and the bidirectional RNN intrusion sensing algorithm.



Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Parminder Singh  <https://orcid.org/0000-0002-0750-6309>
Ashish Kr Luhach  <https://orcid.org/0000-0001-8759-0290>

References

1. Scarfone K and Mell P. Guide to intrusion detection and prevention systems (IDPS). Technical report, National Institute of Standards and Technology, Gaithersburg, MD, 2012.
2. Peng LX, Xie DQ, Gao Y, et al. An immune-inspired adaptive automated intrusion response system model. *Int J Comput Int Sys* 2012; 5(5): 808–815.
3. Ghosh AK, Wanken J and Charron F. Detecting anomalous and unknown intrusions against programs. In: *Proceedings 14th annual computer security applications conference (Cat. No. 98Ex217)*, Phoenix, AZ, 7–11 December 1998, pp.259–267. New York: IEEE.
4. Ilgun K, Kemmerer RA and Porras PA. State transition analysis: a rule-based intrusion detection approach. *IEEE T Software Eng* 1995; 21: 181–199.
5. Tombini E, Debar H, Mé L, et al. A serial combination of anomaly and misuse IDSes applied to HTTP traffic. In: *20th annual computer security applications conference*, Tucson, AZ, 6–10 December 2004, pp.428–437. New York: IEEE.
6. Zhang J and Zulkernine M. A hybrid network intrusion detection technique using random forests. In: *First international conference on availability, reliability and security (ARES'06)*, Vienna, Austria, 20–22 April 2006, p.8. New York: IEEE.
7. Axelsson S. The base-rate fallacy and its implications for the difficulty of intrusion detection. In: *Proceedings of the 6th ACM conference on computer and communications security*, 1999, pp.1–7. New York: ACM, <http://www.raid-symposium.org/raid99/PAPERS/Axelsson.pdf>
8. Song J, Takakura H, Okabe Y, et al. Unsupervised anomaly detection based on clustering and multiple one-class SVM. *IEICE T Commun* 2009; 92(6): 1981–1990.
9. Kim G, Lee S and Kim S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Syst Appl* 2014; 41(4): 1690–1700.
10. Smith J, Nouredinov I, Craddock R, et al. Conformal anomaly detection of trajectories with a multi-class hierarchy. In: *International symposium on statistical learning and data sciences*, Egham, 20–23 April 2015, pp.281–290. New York: Springer.
11. Robinson J, Lonergan M, Singh L, et al. Shard: a framework for sequential, hierarchical anomaly ranking and detection. In: *Pacific-Asia conference on knowledge discovery and data mining*, Kuala Lumpur, Malaysia, 29 May–1 June 2012, pp.243–255. New York: Springer.

12. Goldstein M and Uchida S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS ONE* 2016; 11(4): e0152173.
13. Nguyen XN, Nguyen DT and Vu LH. POCAD: a novel payload-based one-class classifier for anomaly detection. In: *2016 3rd national foundation for science and technology development conference on information and computer science (NICS)*, Da Nang, Vietnam, 14–16 September 2016, pp.74–79. New York: IEEE.
14. Duessel P, Gehl C, Flegel U, et al. Detecting zero-day attacks using context-aware anomaly detection at the application-layer. *Int J Inf Secur* 2017; 16(5): 475–490.
15. Mirsky Y, Doitshman T, Elovici Y, et al. Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089, 2018.
16. Aloqaily M, Otoum S, Al Ridhawi I, et al. An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Netw* 2019; 90: 101842.
17. Ying X, Sagong SU, Clark A, et al. Shape of the cloak: formal analysis of clock skew-based intrusion detection system in controller area networks. *IEEE T Inf Foren Sec* 2019; 14(9): 2300–2314.
18. Aggarwal CC. *Outlier analysis* (Data mining). Cham: Springer, 2015, pp.237–263.
19. Gupta M, Gao J, Aggarwal C, et al. Outlier detection for temporal data. *Synth Lect Data Min Knowl Discov* 2014; 5(1): 1–129.
20. Wang K, Parekh JJ and Stolfo SJ. Anagram: a content anomaly detector resistant to mimicry attack. In: *International workshop on recent advances in intrusion detection*, Hamburg, 20–22 September 2006, pp.226–248. Berlin, Heidelberg: Springer.
21. Swarnkar M and Hubballi N. Rangegram: a novel payload based anomaly detection technique against web traffic. In: *2015 IEEE international conference on advanced networks and telecommunications systems (ANTS)*, Kolkata, India, 15–18 December 2015, pp.1–6. New York: IEEE.
22. Wang K and Stolfo SJ. Anomalous payload-based network intrusion detection. In: *International workshop on recent advances in intrusion detection*, Sophia Antipolis, 15–17 September 2004, pp.203–222. New York: Springer.
23. Tian S, Mu S and Yin C. Sequence-similarity kernels for SVMs to detect anomalies in system calls. *Neurocomputing* 2007; 70(4–6): 859–866.
24. Michlovský Z, Pang S, Kasabov N, et al. String kernel based SVM for internet security implementation. In: *International conference on neural information processing*, Bangkok, 1–5 December 2009, pp.530–539. Berlin, Heidelberg: Springer.
25. Haddadi F. Investigating a behaviour analysis-based early warning system to identify botnets using machine learning algorithms, 2016, <https://pdfs.semanticscholar.org/9da6/fe6ab1befbc09678ef6d94cc2b8c432e9ea1.pdf>
26. Lodhi H, Saunders C, Shawe-Taylor J, et al. Text classification using string kernels. *J Mach Learn Res* 2002; 2: 419–444.
27. Aggarwal CC. An introduction to outlier analysis. In: C Aggarwal. (ed.) *Outlier analysis*. Cham: Springer, 2017, pp.1–34.
28. Perdisci R, Ariu D, Fogla P, et al. McPAD: a multiple classifier system for accurate payload-based anomaly detection. *Comput Netw* 2009; 53(6): 864–881.
29. Yahalom R, Steren A, Nameri Y, et al. Improving the effectiveness of intrusion detection systems for hierarchical data. *Knowl-Based Syst* 2019; 168: 59–69.
30. Peddabachigari S, Abraham A, Grosan C, et al. Modeling intrusion detection system using hybrid intelligent systems. *J Netw Comput Appl* 2007; 30(1): 114–132.
31. Xiang C, Yong PC and Meng LS. Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. *Pattern Recogn Lett* 2008; 29(7): 918–924.
32. Silla CN and Freitas AA. A survey of hierarchical classification across different application domains. *Data Min Knowl Disc* 2011; 22(1–2): 31–72.
33. Bansal N, Blum A and Chawla S. Correlation clustering. *Mach Learn* 2004; 56(1–3): 89–113.
34. Sharma S, Luhach AK and Sinha SA. An optimal load balancing technique for cloud computing environment using bat algorithm. *Indian Journal of Science and Technology* 2016; 9(28), <http://www.indjst.org/index.php/indjst/article/view/98384/71764>
35. Luhach AK, Dwivedi SK and Jha C. Designing a logical security framework for e-commerce system based on SOA. arXiv preprint arXiv:1407.2423, 2014.
36. Batra I, Luhach AK and Pathak N. Research and analysis of lightweight cryptographic solutions for internet of things. In: *Proceedings of the second international conference on information and communication technology for competitive strategies*, Udaipur, India, 04–05 March 2016, p.23. New York: ACM.
37. Javaid A, Niyaz Q, Sun W, et al. A deep learning approach for network intrusion detection system. In: *Proceedings of the 9th EAI international conference on bio-inspired information and communications technologies (formerly BIONETICS)*, New York, 3–5 December 2016, pp.21–26. New York: ACM.
38. Tang TA, Mhamdi L, McLernon D, et al. Deep learning approach for network intrusion detection in software defined networking. In: *2016 international conference on wireless networks and mobile communications (WINCOM)*, Fez, Morocco, 26–29 October 2016, pp.258–263. New York: IEEE.
39. Sheikhan M, Jadidi Z and Farrokhi A. Intrusion detection using reduced-size RNN based on feature grouping. *Neural Comput Appl* 2012; 21(6): 1185–1190.
40. Kaufman L and Rousseeuw PJ. *Finding groups in data: an introduction to cluster analysis*, vol. 344. Hoboken, NJ: John Wiley & Sons, 2009.
41. Wu F, Zhang J and Honavar V. Learning classifiers using hierarchically structured class taxonomies. In: *International symposium on abstraction, reformulation, and approximation*, Airth Castle, 26–29 July 2005, pp.313–320. New York: Springer.
42. Gal-Oz N, Yahalom R and Gudes E. Identifying knots of trust in virtual communities. In: *IFIP international conference on trust management*, Copenhagen, 29 June–1 July 2011, pp.67–81. Berlin, Heidelberg: Springer.

43. Cheeseman P, Kelly J, Self M, et al. Autoclass: a Bayesian classification system. In: J Laird. (ed.) *Machine learning proceedings*. Amsterdam: Elsevier, 1988, pp.54–64.
44. Lloyd S. Least squares quantization in PCM. *IEEE T Inform Theory* 1982; 28(2): 129–137.
45. Martens J and Sutskever I. Learning recurrent neural networks with hessian-free optimization. In: *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp.1033–1040. Citeseer, http://www.icml-2011.org/papers/532_icmlpaper.pdf
46. Tavallaee M, Bagheri E, Lu W, et al. A detailed analysis of the KDD CUP 99 data set. In: *2009 IEEE symposium on computational intelligence for security and defense applications*, Ottawa, ON, Canada, 8–10 July 2009, pp.1–6. New York: IEEE.
47. Revathi S and Malathi A. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *Int J Eng Res Tech* 2013; 2(12): 1848–1853.
48. Paulauskas N and Auskalmis J. Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset. In: *2017 open conference of electrical, electronic and information sciences (eStream)*, Vilnius, Lithuania, 27 April 2017, pp.1–5. New York: IEEE.
49. Bhattacharjee PS, Fujail AKM and Begum SA. Intrusion detection system for NSL-KDD data set using vectorised fitness function in genetic algorithm. *Adv Comput Sci Tech* 2017; 10(2): 235–246.
50. Ashfaq RAR, Wang XZ, Huang JZ, et al. Fuzziness based semi-supervised learning approach for intrusion detection system. *Inform Sciences* 2017; 378: 484–497.
51. Baldi P, Brunak S, Chauvin Y, et al. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics* 2000; 16(5): 412–424.
52. Theano. Theano 1.0.0 documentation, 2017, <http://deeplearning.net/software/theano/>
53. Ingre B and Yadav A. Performance analysis of NSL-KDD dataset using ANN. In: *2015 international conference on signal processing and communication engineering systems*, Guntur, India, 2–3 January 2015, pp.92–96. New York: IEEE.
54. Yin C, Zhu Y, Fei J, et al. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access* 2017; 5: 21954–21961.