The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 2 (2016) 103–116

www.elsevier.com/locate/icte

FCAAIS: Anomaly based network intrusion detection through feature correlation analysis and association impact scale[☆]

V. Jyothsna[*], V.V. Rama Prasad

Sree Vidyanikethan Engineering College A. Rangampet, Tirupati, India

Received 2 March 2016; received in revised form 6 August 2016; accepted 8 August 2016
Available online 26 August 2016

Abstract

Due to the sensitivity of the information required to detect network intrusions efficiently, collecting huge amounts of network transactions is inevitable and the volume and details of network transactions available in recent years have been high. The meta-heuristic anomaly based assessment is vital in an exploratory analysis of intrusion related network transaction data. In order to forecast and deliver predictions about intrusion possibility from the available details of the attributes involved in network transaction. In this regard, a meta-heuristic assessment model called the feature correlation analysis and association impact scale is explored to estimate the degree of intrusion scope threshold from the optimal features of network transaction data available for training. With the motivation gained from the model called "network intrusion detection by feature association impact scale" that was explored in our earlier work, a novel and improved meta-heuristic assessment strategy for intrusion prediction is derived. In this strategy, linear canonical correlation for feature optimization is used and feature association impact scale is explored from the selected optimal features. The experimental result indicates that the feature correlation has a significant impact towards minimizing the computational and time complexity of measuring the feature association impact scale.

Keywords: Intrusion detection; Feature reduction; Correlation analysis; Association impact scale

1. Introduction

Intrusion Detection Systems (IDSs) are mostly of two types: misuse IDSs and anomaly IDSs. A misuse IDS identifies intrusions based on parameters of known attacks and system weaknesses. It however does not recognize new or unfamiliar kinds of attacks. An anomaly IDS is based on the parameters of normal behavior and uses them for identifying any action that strays considerably from normal behavior. The mechanism of misuse intrusion detection trains on the existing patterns of intrusion and matches the data considered for examination with previous patterns to identify intrusions whereas, anomaly intrusion detection is based on identifying patterns from the examination data of normal usage.

IDSs that are efficient in nature are usually developed utilizing data mining techniques owing to their excellent performance of detecting intrusions and capability of generalization. However the process of implementing and installing such systems is complicated in nature. The inherent complications of the systems could be organized into separate problem sets based on the parameters of accuracy, competence, and usability. A key problem associated with IDSs built using data mining techniques, and mostly with those techniques based on anomaly detection is that they show a higher percent of false positive occurrences compared to the previous detection techniques based on hand-crafted signature. Hence, processing of audit data and detection of intrusions on-line are difficult for these techniques. Furthermore, compared to existing methodologies these techniques require vast training data and great complexity is associated with the learning process of the system. The key idea of the approach is to apply heuristic technique to anomaly based intrusion detection. The heuristic approach tends to define a scale that helps to assess the significance of the network

[*] Corresponding author.
E-mail addresses: jyothsna1684@gmail.com (V. Jyothsna), vvramaprasad@rediffmail.com (V.V. Rama Prasad).

Peer review under responsibility of The Korean Institute of Communications Information Sciences.

[☆] This paper is part of a special issue entitled ICT Convergence in the Internet of Things (IoT) guest edited by Yacine Ghamri-Doudane, Yeong Min Jang, Daeyoung Kim, Hossam Hassanein and JaeSeung Song.

http://dx.doi.org/10.1016/j.icte.2016.08.003

transaction. The process devised requires feature extraction, dimensionality reduction for reducing the features extracted and feature selection. Feature extraction involves using all the features with transformation which comprises a combination of all the initial features. During feature selection, features are selected according to the classification criteria.

2. Related work

Eduardo DelaHoz et al. [1] proposed a classification approach that combines statistical techniques and self-organizing maps for detecting the anomalies in the network. Principal component analysis (PCA) and Fisher's discriminant ratio are used for feature selection and noise removal and probabilistic self-organizing maps are used to classify the network transactions as normal or anomalous. Ujwala Ravale et al. [2] proposed a hybrid technique that combines data mining approaches. The K-means clustering algorithm is used to decrease the number of attributes associated with each data point and the Radial basis function (RBF) kernel of support vector machine (SVM) is used for classification. Gaikward et al. [3] proposed a machine learning approach for implementing the IDS. The genetic algorithm is used to reduce the dimensions in the feature set and the partial decision tree is used as a base classifier to implement the IDS. Sunil Nilkanth Pawar et al. [4] proposed a genetic algorithm based network IDS with variable length chromosomes. A chromosome with relevant features is used for rule generation. An effective fitness function is used to define the fitness of each rule. Each chromosome has one or more rules for efficient detection of anomalies. Fangjun Kuang et al. [5] proposed a novel SVM model by combining kernel PCA (KPCA) with improved chaotic particle swarm optimization. KPCA is applied as a preprocessor of SVM to reduce the dimension of feature vectors and shorten training time and improved chaotic particle swarm optimization is proposed to estimate whether the action is normal or intrusion. Iftikhar Ahmad et al. [6] proposed an approach that used PCA for feature subset selection that is based on eigenvalues. Instead of using a traditional approach of selecting features with the highest eigenvalues such as PCA, the authors applied genetic principal components to select the subset of features and SVM for classification. Chun Guo et al. [7] proposed a hybrid learning method, named distance sum-based SVM (DSSVM), for modeling an effective IDS. In DSSVM, the distance sum based on the correlation between each data sample and the cluster centers feature dimensions in the data set is obtained and SVM is used as a classifier. Saurabh Mukherjee et al. [8] proposed a feature vitality based reduction method to identify important features used to detect the anomalies in the selection system, and applied the naive Bayes classifier to detect the anomalies in the IDS.

3. Data set description

The data set developed by Lee and Stolfo et al. [9], KDD-99 is an extensively used data set and is commonly selected for the evaluation of anomaly detection. The data generated from the Intrusion Detection Evaluation DARPA program 1998 was used to build the original KDD-99 data set [10,11] that comprises close to 4,900,000 unique connection vectors, where every connection vector consists of 41 features of which 34 are continuous features and 7 are discrete features. The NSL-KDD [12,13] data set is a polished version of its predecessor KDD-99 data set. As the NSL-KDD data set comprises a huge quantity of data, for experimental purpose sample data from the Kddcup.data_10_percent.gz is taken for the purpose of training. The NSL-KDD data set considered for training is 10% of the main data set equaling 494,020 connection vectors and labeled either as normal or as attack. The activities that show variations with respect to 'normal network behavior' are considered not 'normal' and labeled as attacks [14] and the records corresponding to normal behavior are labeled as normal. The attacks simulated in our experiments belong to any of the four types [15] described below:

1. Denial of service attack (DOS): The DOS attack is a type of attack where an attacker blocks access to valid users by consuming the resources of computer or memory making the system unable to handle valid requests. Examples of DOS attacks are many such as 'teardrop,' 'neptune,' 'ping of death (pod),' 'mail bomb', 'back', 'smurf' and 'land'.
2. Users-to-root attack (U2R): The root attack is a type of attack where the attacker gains access to a valid user account in the system and based on existing system weaknesses acquires access to the systems root component. There are several types of U2R attacks such as 'load-module', 'buffer overflow', 'rootkit', 'perl'.
3. Remote-to-local attack (R2L): The remote-to-local attack is a type of attack where an attacker without an account, accesses locally a legitimate user account based on existing machine vulnerabilities. R2L attacks types are 'phf', 'warezmaster', 'warezclient', 'spy', 'imap', 'ftp_write', 'multihop' and 'guess_passwd'.
4. Probing attack (PROBE): The probing attack is an attack type where an attacker evades the security and collects data on the computers in the network. The PROBE attacks types are 'nmap', 'satan', 'ipsweep' and 'portsweep'.

In the NSL-KDD data set the protocols taken into account are TCP, UDP, and ICMP.

4. Data set preprocessing

The network transactions set contains 42 features with values of type continuous and categorical. To facilitate the optimization process, these values should be numeric and categorical. Henceforth, initially all alphanumeric values have to be converted to numeric values and then the continuous values need to be converted to categorical.

4.1. The procedure to represent the alphanumeric values as numeric values and continuous values as categorical values

- Consider each feature with alphanumeric values and then list all possible unique values and list them with an incremental index that begins at 1.
- Replace the values with their appropriate indexes.

Table 1
Binary representation of the association between T and V.

	val_1	val_2	val_3	val_4	val_5	val_6	val_7	val_8	
tvs_1	1	0	0	0	0	1	0	1	(val_1, val_6, val_8)
tvs_2	0	1	0	0	1	1	0	1	$(val_2, val_5, val_6, val_8)$
tvs_3	1	1	1	0	0	0	1	0	$(val_1, val_2, val_3, val_7)$
tvs_4	0	0	0	0	0	0	1	0	(val_7)
$tvs5$	0	0	0	1	0	1	1	1	$(val_4, val_6, val_7, val_8)$
$tvs6$	1	1	1	1	0	0	1	0	$(val_1, val_2, val_3, val_4, val_7)$

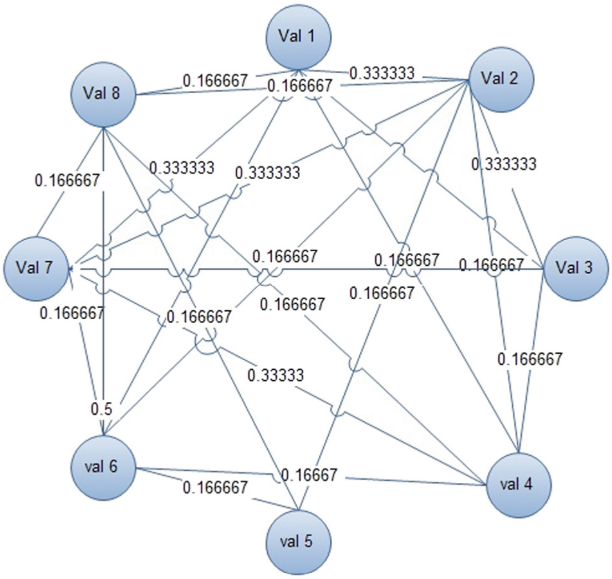

Fig. 1. An example weighted graph of categorical values set of count 8.

- Consider each feature with continuous values, and then partition them into a set of ranges with min and max values, such that the transactions are distributed evenly through all these ranges.

4.2. Feature optimization for anomaly based intrusion detection

The preprocessed set of network transactions are partitioned based on their labels, such that normal transactions are one set, DOS attack transactions are another set and so on.

Consider each feature values set $f_i v(NTS)$ in the resultant normal transactions set (NTS) and their coverage percentage as $f_i v = \{f_i(v_1, c_1),\ f_i(v_2, c_2),\ f_i(v_3, c_3),\ f_i(v_4, c_4), \ldots\ldots, f_i(v_j, c_j)\}$.

Then the feature optimization for each attack A_k can be performed as described in the following steps:

- Consider transactions set $ts(A_k)$ representing attack type A_k (as an example, consider the attack called DOS).
- For each feature $f_i(A_k)$, consider all values as a set $f_i v(A_k)$. Create an empty set $\overline{f_i v}$ of size $|f_i v(A_k)|$, and fill it with values from $f_i v$ according to their coverage percentage such that $|f_i v(A_k)| \cong |\overline{f_i v}|$. Here $|f_i v(A_k)|$ represents the size of the feature values set of $f_i(A_k)$.
- This process is recommended to prepare the feature values vector $\overline{f_i v}$ of the NTS, such that $\overline{f_i v}$ is compatible to '$f_i v(A_k)$' towards size and that also represents the coverage ratio of the values in $f_i v(NTS)$.
- This process should be applied for all feature values set in network transactions of attack A_k.
- Find the canonical correlation between $f_i v(A_k)$ and $\overline{f_i v}$. If the resultant canonical correlation is less than the given threshold or zero, then feature $f_i(A_k)$ can be considered as optimal towards assessing the scale of intrusion scope.

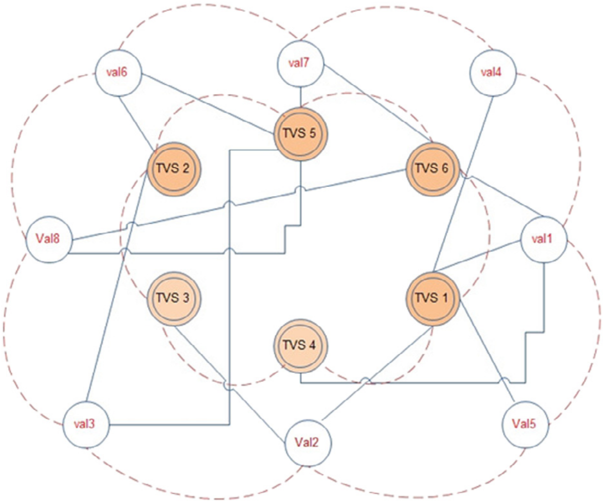

Fig. 2. Duplex graph between $STVS$ and V.

According to the procedure explained in the above steps, the optimal features of the specific attack A_k can be identified.

4.3. Canonical correlation analysis

Two multidimensional data sets X and Y are considered and linear relationships between the data sets are established with the auto covariance and cross-covariance matrices of the second order with standard statistical technique based CCA. The technique is based on finding two bases, one each for data sets X and Y, where the matrix of cross-correlation becomes diagonal, and correlations of the diagonal are maximized.

The parameters used for implementing the canonical correlations are studied [16,17], where, X and Y data vectors should be of equal number; however data vectors $x \in X$ and $y \in Y$ may have varying dimensions assuming the mean is zero. The canonical correlations computation is solved using the equations of eigenvector.

$$C_{xx}^{-1} C_{xy} C_{yy}^{-1} C_{yx} w_x = \rho^2 w_x$$
$$C_{yy}^{-1} C_{yx} C_{xx}^{-1} C_{xy} w_y = \rho^2 w_y \qquad (1)$$

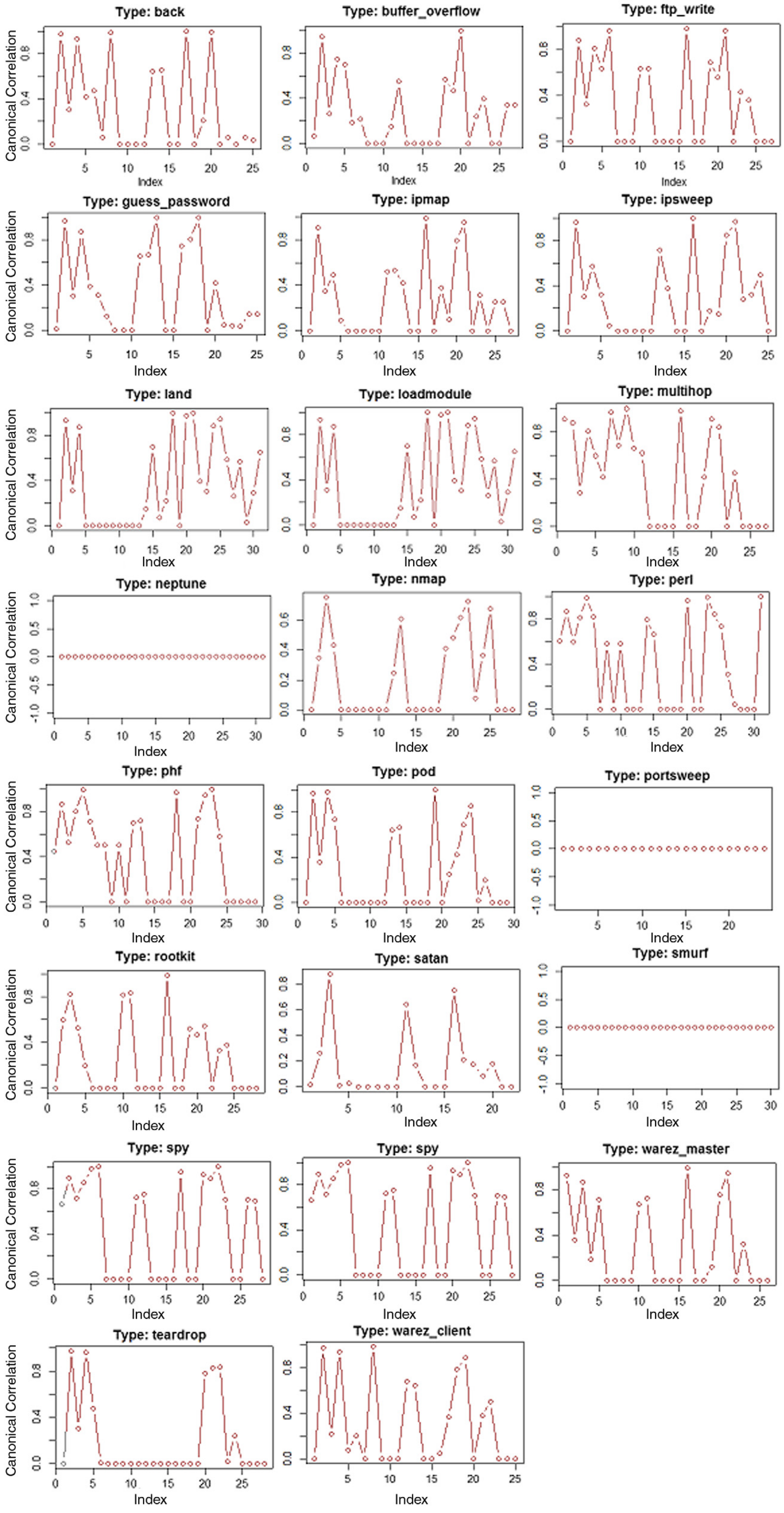

Fig. 3. Canonical correlation of different labels towards normal data.

Here, $C_{yx} = E\{yx^T\}$ where ρ^2 or eigenvalues are the square of canonical correlations and w_x and w_x or the eigenvectors are normalized CCA basis vectors. The solutions to the equations are equivalent to non-zero value whose numbers are equivalent to x and y a vector with lesser dimensional value is considered.

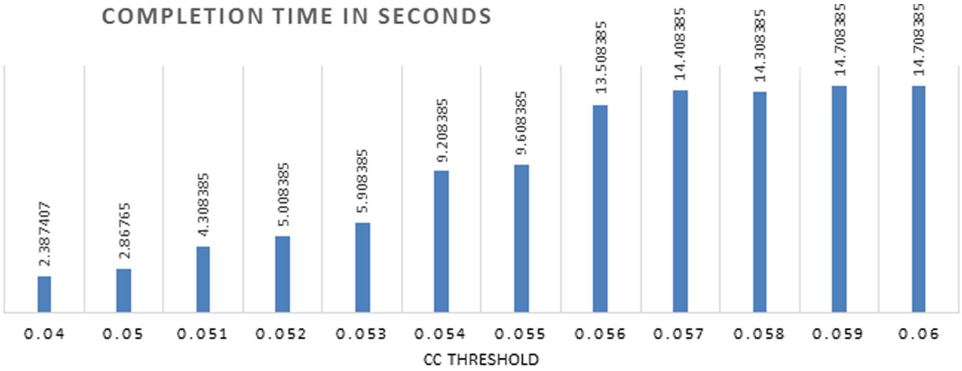

Fig. 4. Completion time of FCAAIS under divergent canonical correlation thresholds.

Table 2
The matrix representation of the edge weights in duplex graph (values rounded to a near value).

0.34	0	0	0	0	0.67	0	0.67
0	0.52	0	0	0.52	0.84	0	0.84
1	1	1	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0.67	0	0.84	0.67	0.84
1.17	1.17	1.17	0.84	0	0	1.34	0

Canonical correlations C_{xx} and C_{yy} are both converted to unit matrices. As $C_{yx} = C_{xy}^T$, Eq. (1) is converted to,

$$C_{xy}C_{xy}^T w_x = \rho^2 w_x$$
$$C_{yx}C_{yx}^T w_y = \rho^2 w_y. \tag{2}$$

These equations depicting the singular value decomposition [18,19] of the cross-covariance matrix C_{xy}

$$C_{xy} = U\Sigma V^T = \sum_{i=1}^{L} \rho_i u_i v_i^T. \tag{3}$$

Here U and V represent orthogonal square matrices ($U^T U = I$, $\mathrm{V}^T V = I$) comprising u_i and v_i representing singular vectors. In this approach, the singular vectors considered are w_{xi} and w_{yi} that represent basis vectors delivering canonical correlations. Matrices U and V and the subsequent u_i and v_i singular vectors dimensionalities usually vary according to the varied dimensions of x and y data vectors.

$$\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \tag{4}$$

The pseudo diagonal matrix includes diagonal matrix D comprising singular values equal to non-zero and attached with zero matrices which makes the matrix Σ compatible with various dimensions of x any y. The non-zero singular values are basically the nonzero canonical correlations whose number is less than any of x and y data vectors dimensions if C_{xy} or the cross-covariance matrix has full rank.

5. Intrusion detection by feature association impact scale

The approach of measuring feature association support (*fas*) metric proposed initially considers the network transaction records of the given training set and feature categorical values used in those network transactions as two independent sets and further builds a duplex graph between these two.

Table 3
Transpose representation of the matrix.

0.34	0	1	0	0	1.17
0	0.52	1	0	0	1.17
0	0	1	0	0	1.17
0	0	0	0	0.67	0.84
0	0.52	0	0	0	0
0.67	0.84	0	0	0.84	0
0	0	1	0	0.67	1.34
0.67	0.84	0	0	0.84	0

5.1. Assumptions

The features $\{f1, f2, f3, \ldots\ldots, fn \forall f_i = \{f_i v_1, f_i v_2, \ldots.. f_i v_m\}\}$, are categorical values and optimal to a specific attack A_k, which are selected through canonical correlation analysis applied on the set of network transactions $T(A_k)$. Here $T(A_k)$ is the set of network transaction records of specific attack A_k of the given training set such that

$$T = \{t_1, t_2, t_3, \ldots\ldots t_n \forall t_i = \{val(f_1), val(f_2), \ldots..val(f_i), val(f_{i+1}), \ldots..val(f_n)\}\}.$$

The set of categorical values of features belonging to each network transaction will be considered as transaction value set *tvs*, and all transaction value sets are referred as '*STVS*'.

In above description $val(f_i)$ can be defined as $val(f_i) \in \{f_i v_1, f_i v_2 \ldots. f_i v_m\}$; hereafter, the term feature refers the current categorical value of the feature. The two features '$val(f_i)$' and '$val(f_j)$', '$val(f_i)$' connected with '$val(f_j)$' if and only if $(val(f_i), val(f_j)) \in tvs_k$.

5.2. Process

To explore the process by an example, consider the total number of divergent values of features as 8 and represented as a set $V = \{val_1, val_2, \ldots.val_8\}$ and $|T|$ as 6, here, $|T|$ is size of the network transaction records. In Table 1 and Fig. 2, each element $\{val_1, val_2, \ldots.val_8\}$ can be $f_i v_j$ such that $\{f_i v_j \exists i \in [1, 2, \ldots\ldots n] \wedge j \in [1, 2, \ldots..m]\}$.

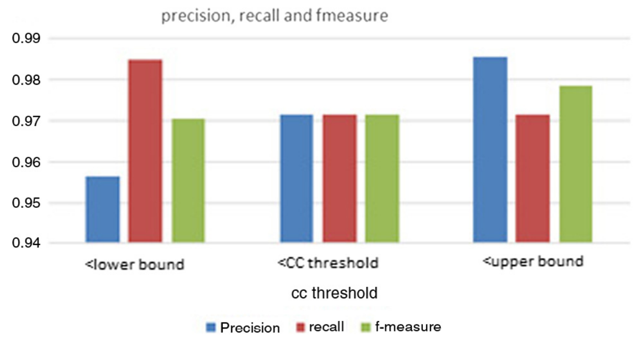

Fig. 5. Performance analysis of the prediction accuracy of FCAAIS under divergent canonical correlation threshold value.

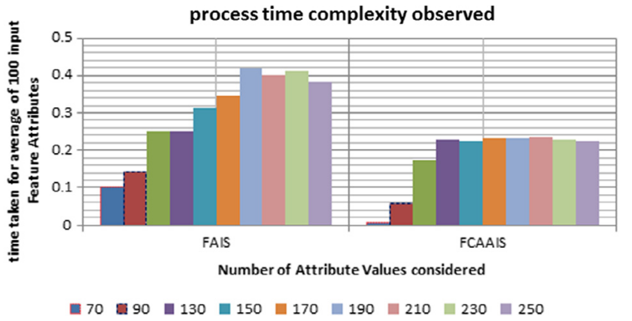

Fig. 6. Process time complexity optimization observed for FCAAIS over FAIS.

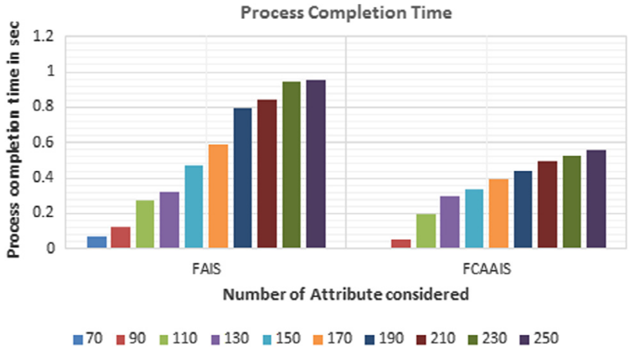

Fig. 7. Process completion time optimization observed for FCAAIS over FAIS.

Table 4
Canonical correlation of the fields of PROBE category under divergent labels against normal data.

	Attack type		
	IPSWEEP	NMAP	PORTSWEEP
Duration	0	0	0
protocol_type	0.965589	1	0
Service	0.304748	0.346288	0
Flag	0.575125	0.755122	0
src_bytes	0.323796	0.433333	0
dst_bytes	0.049496	0	0
Land	1	1	1
wrong_fragment	1	1	1
urgent	1	1	1
Hot	0	0	0
num_failed_logins	1	1	1
logged_in	1	0	0
num_compromised	1	1	1
root_shell	0	0	0
su_attempted	1	1	1
num_root	1	1	1
num_file_creations	0	0	0
num_shells	1	1	1
num_access_files	0	0	0
num_outbound_cmds	1	1	1
is_host_login	1	1	1
is_guest_login	0	0	0
count	0.717419	0.249187	0
srv_count	0.379226	0.608943	0
serror_rate	0	0	0
srv_serror_rate	0	0	0
rerror_rate	1	0	1
srv_rerror_rate	1	0	1
same_srv_rate	0.9999	1	1
diff_srv_rate	0	0	0
srv_diff_host_rate	0.177373	0.413273	0
dst_host_count	0.150618	0.483868	0
dst_host_srv_count	0.846976	0.618582	0
dst_host_same_srv_rate	0.967212	0.725555	0
dst_host_diff_srv_rate	0.280452	0.077617	1
dst_host_same_src_port_rate	0.324168	0.36744	0
dst_host_srv_diff_host_rate	0.497923	0.680034	0
dst_host_serror_rate	0	0	0
dst_host_srv_serror_rate	0	0	0
dst_host_rerror_rate	1	0	1
dst_host_srv_rerror_rate	1	1	1

In the process of detecting the association of each feature categorical value $f_i v_j$ referred as val_k with network transaction records, initially build a duplex graph between transaction value sets $STVS$ and the feature categorical values V (Fig. 3).

The formation of a duplex graph is considered as the graph relations are bipartite, and edges are formed between features and transaction value sets. Each relation in this graph indicates the role of a feature towards a network transaction. An edge between a transaction value set tvs and a feature f is possible if and only if that feature f is the part of tvs. This can be represented as $e_{tvs \leftarrow f} \exists f \in tvs$.

In the weighted undirected graph (Fig. 1) representing a weighted graph with values of features as vertices and edges between the values of features. An edge between any two features $val(f_1), val(f_2)$ will be weighted as follows:

$$\begin{aligned} &ctvs = 0; \\ &foreach\ \{tvs \forall tvs \in STVS\} \\ &ctvs+ = \{1 \forall (val(f_1), val(f_2)) \subseteq tvs\}. \end{aligned} \tag{5}$$

In the above equation, $ctvs$ indicates the count of transactions, which contains both features $val(f_1), val(f_2)$. Then the edge weight between features $val(f_1)$ and $val(f_2)$ can be measured as follows.

$$w(val(f_1) \leftrightarrow val(f_2)) = \frac{ctvs}{|STVS|} \tag{6}$$

In the process of building a weighted graph we consider that an edge between any two features exists if and only if $ctvs \geq 1$.

In the duplex graph (Fig. 2), the dotted line represents that the connected elements belong to the same level of the duplex

Table 5
Optimal features of PROBE Category (less than the mean of the CC value).

IPSWEEP

Duration	0
Hot	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
serror_rate	0
srv_serror_rate	0
diff_srv_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_bytes	0.049496
dst_host_count	0.150618
srv_diff_host_rate	0.177373
dst_host_diff_srv_rate	0.280452
Service	0.304748
src_bytes	0.323796
dst_host_same_src_port_rate	0.324168
srv_count	0.379226
dst_host_srv_diff_host_rate	0.497923

NMAP

Duration	0
dst_bytes	0
Hot	0
logged_in	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
dst_host_diff_srv_rate	0.077617
Count	0.249187
Service	0.346288
dst_host_same_src_port_rate	0.36744
srv_diff_host_rate	0.413273
src_bytes	0.433333

PORTSWEEP

duration	0
protocol_type	0
Service	0
Flag	0
src_bytes	0
dst_bytes	0
Hot	0
logged_in	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
Count	0
srv_count	0
serror_rate	0
srv_serror_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_count	0
dst_host_srv_count	0
dst_host_same_srv_rate	0
dst_host_same_src_port_rate	0
dst_host_srv_diff_host_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0

Table 6
Canonical correlation of the fields of DOS category under divergent labels against normal data.

Attack category: DOS						
Attack type	Back	Land	Neptune	POD	Smurf	Teardrop
Duration	0	0	0	0	0	0
protocol_type	0.973671824	0.932007	0	0.971684	0	0.972549
Service	0.304189382	0.31321	0	0.359382	0	0.303898
Flag	0.930959282	0.872872	0	0.983801	0	0.96171
src_bytes	0.418457014	0	0	0.735404	0	0.479463
dst_bytes	0.474984388	0	0	0	0	0.007766
Land	1	0	1	1	1	1
wrong_fragment	1	1	1	1	1	0
Urgent	1	1	1	1	1	1
Hot	0.06047447	0	0	0	0	0
num_failed_logins	1	1	1	1	1	1
logged_in	0.98836241	0	0	0	0	0
num_compromised	1	1	1	1	1	1
root_shell	0	0	0	0	0	0
su_attempted	1	1	1	1	1	1
num_root	1	1	1	1	1	1
num_file_creations	0	0	0	0	0	0
num_shells	1	1	1	1	1	1
num_access_files	0	0	0	0	0	0
num_outbound_cmds	1	1	1	1	1	1
is_host_login	1	1	1	1	1	1
is_guest_login	0	0	0	0	0	0
Count	0.646676093	0.151103	0	0.639281	0	1
srv_count	0.659146826	0.697969	0	0.662611	0	1
serror_rate	0	0.07359	0	0	0	0
srv_serror_rate	0	0.218218	0	0	0	0
rerror_rate	1	1	0	0	0	0
srv_rerror_rate	1	0	0	0	0	0
same_srv_rate	0.999733481	0.974774	0	0.999527	0	1
diff_srv_rate	0	0.996815	0	0	0	0
srv_diff_host_rate	0.20949181	0.394435	0	0.249303	0	0
dst_host_count	1	0.303341	0	0.422084	0	0.780417
dst_host_srv_count	1	0.883352	0	0.686862	0	0.828144

(*continued on next page*)

Table 6 (*continued*)

Attack category: DOS						
Attack type	Back	Land	Neptune	POD	Smurf	Teardrop
dst_host_same_srv_rate	0.994114036	0.942695	0	0.852949	0	0.833876
dst_host_diff_srv_rate	0	0.586353	0	0.0161	0	0.014048
dst_host_same_src_port_rate	0.057398167	0.264927	0	0.198461	0	0.236948
dst_host_srv_diff_host_rate	0	0.570761	0	1	0	0
dst_host_serror_rate	0.056962968	0.028009	0	0	0	0
dst_host_srv_serror_rate	0.036273813	0.29177	0	0	0	0
dst_host_rerror_rate	1	0.650791	0	0	0	0
dst_host_srv_rerror_rate	1	1	0	0	1	1

Table 7
Optimal features of DOS category (less than the mean of the CC value).

BACK

Duration	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
serror_rate	0
srv_serror_rate	0
diff_srv_rate	0
dst_host_diff_srv_rate	0
dst_host_srv_diff_host_rate	0
dst_host_srv_serror_rate	0.0362738
dst_host_serror_rate	0.056963
dst_host_same_src_port_rate	0.0573982
Hot	0.0604745
srv_diff_host_rate	0.2094918
Service	0.3041894
src_bytes	0.418457
dst_bytes	0.4749844

LAND

duration	0
src_bytes	0
dst_bytes	0
Land	0
Hot	0
logged_in	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
srv_rerror_rate	0
dst_host_serror_rate	0.028009
serror_rate	0.07359
Count	0.151103
srv_serror_rate	0.218218
dst_host_same_src_port_rate	0.264927
dst_host_srv_serror_rate	0.29177
dst_host_count	0.303341
Service	0.31321
srv_diff_host_rate	0.394435

POD

duration	0
dst_bytes	0
hot	0
logged_in	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
dst_host_srv_rerror_rate	0
dst_host_diff_srv_rate	0.0161
dst_host_same_src_port_rate	0.198461
srv_diff_host_rate	0.249303
service	0.359382
dst_host_count	0.422084

NEPTUNE

Duration	0
protocol_type	0
Service	0
Flag	0
src_bytes	0
dst_bytes	0
Hot	0
logged_in	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
Count	0
srv_count	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
same_srv_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_count	0
dst_host_srv_count	0
dst_host_same_srv_rate	0
dst_host_diff_srv_rate	0
dst_host_same_src_port_rate	0
dst_host_srv_diff_host_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
dst_host_srv_rerror_rate	0

SMURF

Duration	0
protocol_type	0
Service	0
Flag	0
src_bytes	0
dst_bytes	0
Hot	0
logged_in	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
Count	0
srv_count	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
same_srv_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_count	0
dst_host_srv_count	0
dst_host_same_srv_rate	0
dst_host_diff_srv_rate	0
dst_host_same_src_port_rate	0
dst_host_srv_diff_host_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0

TEARDROP

duration	0
wrong_fragment	0
hot	0
logged_in	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_srv_diff_host_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
dst_bytes	0.007766
dst_host_diff_srv_rate	0.014048
dst_host_same_src_port_rate	0.236948
service	0.303898

Table 8
Canonical correlation of the fields of U2R category under divergent labels against normal data.

Attack category: U2R				
Attack type	Buffer_overflow	Load module	Perl	Rootkit
Duration	0.06468	0	0.603172	0
Protocol_type	0.947758	0.932007	0.870388	1
Service	0.26415	0.31321	0.598528	0.599632
Flag	0.750583	0.872872	0.816497	0.822192
src_bytes	0.701439	0	0.988761	0.52679
dst_bytes	0.183606	0	0.820712	0.193833
Land	1	0	1	1
wrong_fragment	1	1	1	1
urgent	1	1	1	1
Hot	0.214286	0	0	0
num_failed_logins	1	1	1	1
logged_in	1	0	1	1
num_compromised	1	1	1	1
root_shell	1	0	0.57735	1
su_attempted	1	1	1	1
num_root	1	1	0	1
num_file_creations	0	0	0.57735	0
num_shells	1	1	0	1
num_access_files	0	0	0	0
num_outbound_cmds	1	1	1	1
is_host_login	1	1	1	1
is_guest_login	0	0	0	0
Count	0.150811	0.151103	0.796662	0.816236
srv_count	0.548471	0.697969	0.66472	0.834614
serror_rate	0	0.07359	0	0
srv_serror_rate	0	0.218218	0	0
rerror_rate	0	1	0	0
srv_rerror_rate	1	0	0	0
same_srv_rate	1	0.974774	0.96225	0.987763
diff_srv_rate	0	0.996815	0	0
srv_diff_host_rate	0	0.394435	0	0
dst_host_count	0.570181	0.303341	0.993878	0.521379
dst_host_srv_count	0.463404	0.883352	0.843733	0.466736
dst_host_same_srv_rate	0.999838	0.942695	0.735718	0.545537
dst_host_diff_srv_rate	0	0.586353	0.312348	0
dst_host_same_src_port_rate	0.239158	0.264927	0.042818	0.331178
dst_host_srv_diff_host_rate	0.392318	0.570761	0	0.377964
dst_host_serror_rate	0	0.028009	0	0
dst_host_srv_serror_rate	0	0.29177	0	0
dst_h ost_rerror_rate	0.342997	0.650791	1	0
dst_host_srv_rerror_rate	1	1	0	0

graph, and the solid line indicates the relation between a feature value and transaction value set. If a feature categorical value $f_i v_j$ referred as val_1 exists in tvs_1 then the weight of the connection between val_1 and tvs_1 will be the sum of the weights of the edges between val_1 and each feature categorical value $\{f_i v_j \exists f_i v_j \in tvs_1\}$ of tvs_1 that is defined in the weighted graph.

Further, a matrix A will be formed that represents the edge weights of the duplex graph between transaction value sets and feature categorical values. Then A', the transpose of matrix A is obtained.

Consider *STVS* as a database and depict it as a duplex graph without loss of information. Let $STVS = \{tvs_1, tvs_2, \dots, tvs_6\}$ be a list of transaction value sets and $V = \{val_1, val_2, \dots .val_8\}$ be the corresponding set of feature categorical values. Then, clearly *STVS* is equivalent to the duplex-graph $DG = (STVS, V, E)$.

Here, $E = \{(tvs_i, val_j) : val_j \in tvs_i, tvs_i \in STVS, val_j \in V\}$.

Assuming transaction value sets of the given duplex graph as pivots and the feature categorical values as pure prerogatives, the pivot and prerogative values can be measured as follows:

The matrix representation of transaction value sets and feature connections is represented as matrix 'A' (see Table 2). The value represents the sum of edge weights between a prerogative feature categorical value and other feature categorical values of the target pivot, which is a transaction value set.

If a feature categorical value val_1 exists in transaction value set tvs_1 then the weight of the connection between val_1 and tvs_1 will be the sum of the weights of the edges between val_1 and

Table 9
Optimal features of U2R category (less than the mean of the CC value).

BUFFER_OVERFLOW

num_file_creations	0.00
num_access_files	0.00
is_guest_login	0.00
serror_rate	0.00
srv_serror_rate	0.00
rerror_rate	0.00
diff_srv_rate	0.00
srv_diff_host_rate	0.00
dst_host_diff_srv_rate	0.00
dst_host_serror_rate	0.00
dst_host_srv_serror_rate	0.00
Duration	0.06
Count	0.15
dst_bytes	0.18
Hot	0.21
dst_host_same_src_port_rate	0.24
Service	0.26
dst_host_rerror_rate	0.34
dst_host_srv_diff_host_rate	0.39
dst_host_srv_count	0.46

LOAD MODULE

duration	0
src_bytes	0
dst_bytes	0
Land	0
Hot	0
logged_in	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
srv_rerror_rate	0
dst_host_serror_rate	0.028009
serror_rate	0.07359
Count	0.151103
srv_serror_rate	0.218218
dst_host_same_src_port_rate	0.264927
dst_host_srv_serror_rate	0.29177
dst_host_count	0.303341
Service	0.31321
srv_diff_host_rate	0.394435

ROOTKIT

duration	0
Hot	0
num_file_creations	0
num_access_files	0
is_guest_login	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_diff_srv_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
dst_host_srv_rerror_rate	0
dst_bytes	0.193833
dst_host_same_src_port_rate	0.331178
dst_host_srv_diff_host_rate	0.377964
dst_host_srv_count	0.466736

PERL

Hot	0
num_root	0
num_shells	0
num_access_files	0
is_guest_login	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_srv_diff_host_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_srv_rerror_rate	0
dst_host_same_src_port_rate	0.042818
dst_host_diff_srv_rate	0.312348

each feature categorical value $\{val_i \exists val_i \in tvs_1\}$ of tvs_1. These weights are the edge weights represented in weighted graph WG. The transpose of the matrix A is A' (see Table 3).

Find prerogative weights by aggregating each row of A' (a matrix v that represents the prerogative weights will be formed).

Now find the pivot weights through matrix multiplication between $A \; and \; v$.

$$u = A \times v. \tag{7}$$

Then the fas of feature categorical value $f_i v_j$ can be measured as follows

$$fas(f_i v_j) = \frac{\sum_{k=1}^{|STVS|} \{u(tvs_k) : (f_i v_j \rightarrow tvs_k) \neq 0\}}{\sum_{k=1}^{|STVS|} u(tvs_k)}. \tag{8}$$

Then the fas between feature categorical values $f_i v_j$ and $f_{i'} v_{j'}$ can be measured as follows

$$fas(f_i v_j \leftrightarrow f_{i'} v_{j'}) = \frac{\sum_{k=1}^{|STVS|} \{u(tvs_k) \exists (f_i v_j, f_{i'} v_{j'}) \subset tvs_k\}}{\sum_{k=1}^{|STVS|} u(tvs_k)}. \tag{9}$$

In the above equation descriptions, the $|STVS|$ represents total number of transaction value sets.

The feature association impact scale (FAIS) of each transaction value set tvs_i can be measured as follows:

$$fais(tvs_i) = 1 - \frac{\sum_{j=1}^{m} \{fas(\{val_j \exists val_j \in V\}) : (val_j \subset tvs_i)\}}{|tvs_i|}. \tag{10}$$

The FAIS threshold $faist$ can be found as follows:

$$faist = \frac{\sum_{i=1}^{|STVS|} fais(tvs_i)}{|STVS|}. \tag{11}$$

Table 10
Canonical correlation of the fields of R2L category under divergent labels against normal data.

Attack category: R2L

Attack type	FTP_WRITE	GUESS_PASSWORD	IPMAP	MULTIHOP	PHF	SATAN	SPY	WAREZ_CLIENT	WAREZ_MASTER
Duration	0	0.011374	0	0.9092	0.447214	0.014181	0.663675	0	1
protocol_type	0.883883	0.967297	0.903696	0.87831	0.866025	1	0.894427	0.973524	0.92966968
Service	0.321815	0.303572	0.346826	0.287029	0.527506	0.266065	0.720078	0.215949	0.354636861
Flag	0.810931	0.869756	1	0.805823	0.802955	0.882091	0.857493	0.942495	0.869318288
src_bytes	0.637815	0.388506	0.493244	0.597609	0.989973	0.010764	0.978258	0.075232	0.18312681
dst_bytes	0.967743	0.309517	0.094107	0.415869	0.71357	0.02411	0.999118	0.203639	0.715331926
Land	1	1	1	1	1	1	1	1	1
wrong_fragment	1	1	1	1	1	1	1	1	1
Urgent	1	1	1	1	1	1	1	1	1
Hot	0	0.125	0	0.970269	0.5	0	0	0	0
num_failed_logins	1	1	1	1	1	1	1	1	1
logged_in	1	1	1	1	1	1	1	0.988661	1
num_compromised	1	1	1	1	1	1	1	1	1
root_shell	0	0	0	1	0.5	0	0	0	0
su_attempted	1	1	1	1	1	1	1	1	1
num_root	1	1	1	1	1	1	1	1	1
num_file_creations	0	0	0	0.685994	0	0	0	0	0
num_shells	1	1	1	1	1	1	1	1	1
num_access_files	1	0	0	1	0.5	0	1	0	0
num_outbound_cmds	1	1	1	1	1	1	1	1	1
is_host_login	1	1	1	1	1	1	1	1	1
is_guest_login	1	1	0	1	0	0	0	1	1
Count	0.636775	0.656092	0.522655	0.66571	0.698297	0.641479	0.723356	0.678245	0.670388708
srv_count	0.636243	0.666086	0.535262	0.620293	0.717775	0.167174	0.754606	0.644706	0.72429978
serror_rate	0	1	1	0	0	0	0	0	0
srv_serror_rate	0	1	0.421042	0	0	0	0	0	0
rerror_rate	0	1	0	0	0	1	0	1	0
srv_rerror_rate	0	1	0	0	0	1	0	1	0
same_srv_rate	0.984798	0.99765	0.989762	0.982708	0.970725	1	0.948683	1	0.993812023
diff_srv_rate	0	0	0	0	0	1	0	1	0
srv_diff_host_rate	0	0	0.378165	0	0	0	0	0.045682	0
dst_host_count	0.692029	0.745901	0.100371	0.418296	0.737716	0.755822	0.928125	0.366917	0.120146782
dst_host_srv_count	0.559688	0.805651	0.795927	0.912457	0.945191	0.210034	0.892864	0.783955	0.75331486
dst_host_same_srv_rate	0.968427	0.997687	0.956324	0.841431	0.998623	0.171688	0.996833	0.887309	0.94792623
dst_host_diff_srv_rate	0	0	0	0	0.57735	0.085977	0.707107	0	0
dst_host_same_src_port_rate	0.43049	0.416211	0.315129	0.452633	0	0.179867	0	0.375513	0.32159335
dst_host_srv_diff_host_rate	0.359734	0.044368	0	0	0	0	0	0.50144	0
dst_host_serror_rate	0	0.041338	0.254121	0	0	1	0.707107	1	1
dst_host_srv_serror_rate	0	0.032949	0.254121	0	0	0	0.695795	0	0
dst_host_rerror_rate	0	0.143514	0	0	0	1	0	0	0
dst_host_srv_rerror_rate	0.342997	0.143514	0	1	0	1	1	0	0

In the above equation $|STVS|$ indicates the total number of transaction value sets

The standard deviation of the *fais* of each transaction value set needs to be measured further, to estimate the low, medium and high ranges of *faist*. The mathematical notation of estimating standard deviation is as follows:

$$sdv_{faist} = \sqrt{\frac{\left(\sum_{i=1}^{|STVS|} (fais(tvs_i) - faist)^2\right)}{(|STVS| - 1)}}. \tag{12}$$

The FAIS range can be explored as follows:

The lower threshold of *faist* range is

$$faist_l = faist - sdv_{faist}. \tag{13}$$

The higher threshold of *faist* range is

$$faist_h = faist + sdv_{faist}. \tag{14}$$

A network transaction *nt* can be said as safe if and only if $fais(nt) < faist_l$.

A network transaction *nt* can be said as suspected to be an intrusion if and only if $fais(nt) \geq faist_l \&\& fais(nt) < faist_h$. The network transaction *nt* can be confirmed as intrusion if $fais(nt) \geq faist_h$.

6. Results

The canonical correlation is applied on the IDS data set explored in Section 4.3, which is carried out by using expression language R. The data preprocessing is performed using JAVA.

Table 11
Optimal features of R2L category (less than the mean of the CC value).

FTP_WRITE	
Duration	0
Hot	0
root_shell	0
num_file_creations	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_diff_srv_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
Service	0.321815
dst_host_srv_rerror_rate	0.342997
dst_host_srv_diff_host_rate	0.359734
dst_host_same_src_port_rate	0.43049

IMAP	
Duration	0
Hot	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
dst_host_diff_srv_rate	0
dst_host_srv_diff_host_rate	0
dst_host_rerror_rate	0
dst_host_srv_rerror_rate	0
dst_bytes	0.094107
dst_host_count	0.100371
dst_host_serror_rate	0.254121
dst_host_srv_serror_rate	0.254121
dst_host_same_src_port_rate	0.315129
Service	0.346826
srv_diff_host_rate	0.378165
srv_serror_rate	0.421042
src_bytes	0.493244

PHF	
num_file_creations	0
is_guest_login	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_same_src_port_rate	0
dst_host_srv_diff_host_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
dst_host_srv_rerror_rate	0
duration	0.447214
hot	0.5
root_shell	0.5
num_access_files	0.5
service	0.527506

SATAN	
hot	0
root_shell	0
num_file_creations	0
num_access_files	0
is_guest_login	0
serror_rate	0
srv_serror_rate	0
srv_diff_host_rate	0
dst_host_srv_diff_host_rate	0
dst_host_srv_serror_rate	0
src_bytes	0.010764
duration	0.014181
dst_bytes	0.02411
dst_host_diff_srv_rate	0.085977
srv_count	0.167174
dst_host_same_srv_rate	0.171688
dst_host_same_src_port_rate	0.179867
dst_host_srv_count	0.210034
service	0.266065

GUESS_PASSWORD	
root_shell	0
num_file_creations	0
num_access_files	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_diff_srv_rate	0
Duration	0.011374
dst_host_srv_serror_rate	0.032949
dst_host_serror_rate	0.041338
dst_host_srv_diff_host_rate	0.044368
Hot	0.125
dst_host_rerror_rate	0.143514
dst_host_srv_rerror_rate	0.143514
Service	0.303572
dst_bytes	0.309517
src_bytes	0.388506
dst_host_same_src_port_rate	0.416211

MULTIHOP	
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_diff_srv_rate	0
dst_host_srv_diff_host_rate	0
dst_host_serror_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
Service	0.287029
dst_bytes	0.415869
dst_host_count	0.418296
dst_host_same_src_port_rate	0.452633
src_bytes	0.597609
srv_count	0.620293

WAREZ_CLIENT	
duration	0
hot	0
root_shell	0
num_file_creations	0
num_access_files	0
serror_rate	0
srv_serror_rate	0
dst_host_diff_srv_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
dst_host_srv_rerror_rate	0
srv_diff_host_rate	0.045682
src_bytes	0.075232
dst_bytes	0.203639
service	0.215949
dst_host_count	0.366917
dst_host_same_src_port_rate	0.375513
dst_host_srv_diff_host_rate	0.50144

WAREZ-MASTER	
Hot	0
root_shell	0
num_file_creations	0
num_access_files	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_diff_srv_rate	0
dst_host_srv_diff_host_rate	0
dst_host_srv_serror_rate	0
dst_host_rerror_rate	0
dst_host_srv_rerror_rate	0
dst_host_count	0.1201468
src_bytes	0.1831268
dst_host_same_src_port_rate	0.3215934
Service	0.3546369

SPY	
Hot	0
root_shell	0
num_file_creations	0
is_guest_login	0
serror_rate	0
srv_serror_rate	0
rerror_rate	0
srv_rerror_rate	0
diff_srv_rate	0
srv_diff_host_rate	0
dst_host_same_src_port_rate	0
dst_host_srv_diff_host_rate	0
dst_host_rerror_rate	0

The canonical correlation of different fields under divergent labels is explored in Fig. 3 and Tables 4–11 (see Appendix A).

Total records tested were 25% of actual data set (20% of selected test records are of each category called Probe, DOS, U2R, R2L and Normal records) and CC threshold is considered in the range of lower bound (the difference between CC average and standard deviation of CC) and upper bound (the sum of CC average and standard deviation of CC).

Total number of records found to be normal are 19.8% of test data records (false negative are 4.7% (intruded transactions are falsely claimed as normal) and true negatives are 15.1% (normal transactions truly claimed as normal)).

Table 12
Precision, recall and F-measure values found from the results of the empirical analysis.

	Precision	Recall	F-measure
Less than the lower bound of CC threshold	0.98	0.99	0.984974619
Less than the CC threshold	0.985	0.985	0.985
Less than the upper bound of CC threshold	0.989	0.987	0.987998988

Total number of records found to be intruded are 80.2% (intruded transactions claimed truly as intruded are 75.3% of test data records (true positives) and normal records are claimed falsely as intruded are 4.9% of test data records (false positives)).

As per the results explored, the proposed model is accurate to the level of 90.4%. The failure percentage is 9.6%.

The experiments are also conducted on the same data set using model called Anomaly based network intrusion detection through assessing feature association impact scale, and the explored results indicate that this model is also scalable and robust towards forecasting the intrusion scope of a network transaction (observed detection accuracy is approx. 91%), but the major obstacle observed in earlier model compared to the proposed model is process complexity. This is due to the number of features opted to assess the scale is drastically minimized in the proposed model than our earlier model and at the same time the model proposed preserves its accuracy towards intrusion detection (see Figs. 6 and 7).

As per these results, the accuracy of the proposal of optimizing feature set using canonical correlation (FCAAIS) minimizes the process complexity of the FAIS.

The observed time complexity is scalable since the completion time is not compliment to the ratio of features count, which is due to higher CC threshold (see Fig. 4).

Hence, it is obvious to conclude that the applying canonical correlation towards optimized attribute selection is significant to boost the FAIS model (see Table 12).

The intrusion detection accuracy (the percentage of valid predictions by the proposed method) is used as the main performance measure. In addition to accuracy, we used precision, recall, and F-measure to measure the performance (see Fig. 5).

7. Conclusion

It is desirable for anomaly based IDSs to achieve high classification accuracy, and meanwhile reduce the complexity of the rules that are extracted from training data. Owing to the fact that the accuracy and interpretability are often contradictory in the optimization process, we proposed canonical correlation analysis to optimize the features towards detecting the intrusions. The selection of optimal features simplifies the process of FAIS which is used in our earlier research article. The experiments were done using a benchmark data set. The results demonstrate the canonical correlation analysis is promising and significant to select optimal attributes of the network transactions used for training. Furthermore the proposed model minimized the process complexity and completion time and retains the maximal prediction accuracy.

Appendix

See Tables 4–11.

References

[1] Eduardo DelaHoz, EmiroDeLaHoz AndrésOrtiz, JulioOrtega and BeatrizPrieto, PCA filtering and probabilistic SOM for network intrusion detection, in: Special Issue: Advances in Computational Intelligence in Elsevier—Neurocomputing, vol. 164, 2015, pp. 71–81.

[2] Ujwala Ravale, Nilesh Marathe, Puja Padiya, Feature selection based hybrid anomaly intrusion detection system using K means and RBF kernel function, in: Proceeding of International Conference on Advanced Computing Technologies and Applications, ICACTA-2015, in: Procedia Computer Science, vol. 45, Elsevier, 2015, pp. 428–435.

[3] D.P. Gaikward, Ravindra c Thool, Intrusion detection system using bagging with partial decision tree base classifier, in: Proceeding of International Conference on Advanced in Computing, Communication and Control, ICAC3'15, in: Procedia Computer Science, vol. 49, Elsevier, 2015, pp. 92–98.

[4] Sunil Nilkanth Pawar1, Rajankumar Sadashivrao Bichkar, Genetic algorithm with variable length chromosomes for network intrusion detection, Int. J. Autom. Comput. 12 (3) (2015) 337–342.

[5] Fangjun Kuang, Siyang Zhang, Zhong Jin, Weihong Xu, A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection, Soft Computing 19 (2015) 1187–1199.

[6] Iftikhar Ahmad, Muhammad Hussain, Abdullah Alghamdi, Abdulhameed Alelaiwi, Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components, Neural Comput. 24 (2014) 1671–1682.

[7] Chun Guo, Yajian Zhou, Yuan Ping, Zhongkun Zhang, Guole Liu, Yixian Yang, A distance sum-based hybrid method for intrusion detection, in: Appl. Intell., vol. 40, Springer, 2014, pp. 178–188.

[8] Saurabh Mukherjee, Neelam Sharma, Intrusion detection using naive bayes classifier with feature reduction, in: Proceedings in 2nd International Conference on Computer, Communication, Control and Information Technology, C3IT-2012, in: Procedia Technology, vol. 4, Elsevier, 2012, pp. 119–128.

[9] W. Lee, S. Stolfo, A framework for constructing features and models for intrusion detection systems, ACM Trans. Inf. Syst. Secur. 3 (4) (2000) 227–261.

[10] KDD data set, 1999, http://kdd.ics.uci.edu/databases/-kddcup99/kddcup99.html.

[11] Mohbad Tavallaee, Ebrahim Bagheri, Wei Lu, Ali A Ghorbani, A detailed analysis of the KDD cup 99 data set, in: Proceedings of IEEE Symposium on computational Intelligence in security and defence applications, CISDA 2009, pp. 53–58.

[12] S. Revathi, Dr. A. Malathi, A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection, Int. J. Eng. Res. Technol. (IJERT) (ISSN: 2278-0181) 2 (12) (2013).

[13] L. Dhanabal, Dr. S.P. Shantharajah, A study on NSL-KDD dataset for intrusion detection system based on classification algorithms, Int. J. Adv. Res. Comput. Commun. Eng. 4 (6) (2015).

[14] http://nsl.cs.unb.ca/NSL-KDD/.

[15] Preeti Aggarwala, Sudhir Kumar Sharmab, Analysis of KDD dataset attributes—class wise for intrusion detection, in: Proceedings of 3rd International Conference on Recent Trends in Computing 2015, ICRTC-2015, in: Procedia Computer Science, vol. 7, Elsevier, 2015, pp. 842–851.

[16] Hardoon David R, Sandor Szedmak, John Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural Comput. 16 (2004) 2639–2664.

[17] M. Borga, Canonical Correlation: A Tutorial, Linkoping University, Linkoping, Sweden, 2001, p. 12.
Available at http://www.imt.liu.se/magnus/cca/tutorial/.

[18] S. Akaho, A Kernel Method for Canonical Correlation Analysis, International Meeting of Psychometric Society, IMPS2001, 2001.

[19] A. Hyvarinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, 2001.