IADIS International Journal on Computer Science and Information Systems
Vol. 12, No. 1, pp. 26-44
ISSN: 1646-3692

# A FEATURE CLUSTERING BASED DIMENSIONALITY REDUCTION FOR INTRUSION DETECTION (FCBDR)

Gunupudi Rajesh Kumar[1], NimmalaMangathayaru[1] and Gugulothu Narsimha[2]
[1]*Department of Information Technology, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India*
[2]*Computer Science and Engineering, JNTUH College of Engineering, Jagityal, India*

**ABSTRACT**

This work discusses the approach for intrusion detection and classification by devising a membership function, inspired from Yung, Jung, & Shie-Jue (2014) and used in this work to carry the dimensionality reduction of processes present in the training set in evolutionary approach. The reduced process representation may then be used to perform classification and prediction for detecting intrusion. It is seen that the reduced representation of processes retains the system call distribution of the initial process. Experiment results show the proposed approach is better compared to existing approaches and helps in effective identification of U2R and R2L attacks.

**KEYWORDS**

Classifier, Malicious, Intrusion, System Call, Fuzzy Feature

## 1. INTRODUCTION

With the rapid growth in applications such as on-line banking, e-commerce, business transactions, online monitoring systems, ensuring security and privacy is once again one of the important challenges in today's scenario. As the technology is growing day by day, the cybercrimes are also rising with accelerated growth. Researchers are proposing several methodologies to detect intrusion detection attacks, but these attacks are taking dynamic forms by changing their approach and states. Even today, intrusion detection is challenging as there is no single ideal solution to address this. In this paper an attempt is made to address this gap. We propose an efficient fuzzy feature clustering method that uses different text processing techniques in data mining.

Now a days the pace in the process of development of new intrusion detection techniques is slightly lagging behind the pace of introduction to new attacks. Intrusion detection techniques based on the detection methodology, are broadly classified into two categories such as signature based or misuse based intrusion detection techniques and anomaly based intrusion detection techniques. The signature based intrusion detection techniques are of no use when attacks are taking different shape and forms. Till late 1990s these signature based techniques seem to be working fine. But, as the internet usage is growing for different purposes and applications, the types and forms of these attacks are also taking revolutionary changes. This once again turned as a major challenge for researchers which led to the evolution of anomaly based intrusion detection techniques. The anomaly based intrusion detection system works on the principle of identification and prediction of packet patterns or bit sequence which causes a compromise in security. These anomaly based intrusion detection techniques looks for the probability of genuinity of the detected pattern. This method makes use of fuzzy, probability, statistical based approaches for the detection of the threat. The computerization of each and every field that includes robotics in medicine, pharmacy, science, physics, paramedical, space, education, communication, teaching and learning etc. deals with different types of data. Any algorithm proves to be efficient in detecting intrusion detection threats will not be performing better while dealing with different type of data. Over all it is proved in many cases that not all major intrusion detection techniques deal with multiple types of data and hence there is a lot of scope for research in this direction. Intrusion detection systems based on the analyzation of activity generally divided into two major groups such as network based intrusion detection and host based intrusion detection techniques. Network based intrusion detection (NIDS) includes Firewalls, Antivirus servers, and deployment of different security appliances such as UTMs (Unified Threat Management).

The host based intrusion detection systems works at the host level. NIDs operate in line with the live network traffic and have active capabilities such as detection and prevention of malicious packets from the network. NIDs are responsible for safeguarding the IT infrastructure from external threats where as the host based intrusion detection techniques (HIDS) ensure the safety within the organizations network i.e. behind the firewall. HIDS includes Antivirus, Internet Security, and anti-spyware services to be working at host level. The higher the detection of true positives, the performance of the IDS will be better. The IDS should comprise multiple layers of security in its architecture. The properties (Anderson, 2001; Nitin, Mattord, & Verma, 2008) of ideal IDS should be as follows:

1. Ability to isolate noise in data in the network makes an intrusion detection system efficient and long standing.
2. The number of false alarms should be as minimum as possible. Higher true positive rate does not make it more dependent rather the low rate of false alarms makes the system more robust.
3. Constant up-gradation of the malicious signatures, their packet sequences, and bit patterns makes the intrusion detection more novel and fresh. Failing to up-gradation of malicious data makes it more vulnerable to newer attacks.
4. Especially in the case of signature based intrusion detection systems the signature database need to be updated immediately, otherwise in this short updating gap the IDS becomes a vulnerable.
5. The IDS cannot do anything with respect to authentication process. In the case of weak authentication process the IDS may become helpless. The architects of IDS should keep this in their mind before commissioning IDS.

6. As the IDS needs to analyze the network traffic, it should not take too much of time so that the packet becomes expired. This delay should not be the burden for the IDS.
7. The IDS architects should keep in their mind that the data if it is encrypted, the IDS cannot understand whether it is legitimate or malicious.

In this paper we introduce a novel method for intrusion detection that involves fuzzy membership function as similarity measure inspired from Jiang, Liou, & Lee (2011), which is being tested with existing datasets such as DARPA, KDD Cup 99. As the KDD Cup 99 is having duplicate instances we propose to use NSL-KDD Cup 99 (Mahbod, Ebrahim, Wei, & Ghorbani, 2009; Preeti,& Sudhir, 2015) dataset which is free of duplicate instances. In our previous study (Gunupudi, Mangathayaru, & Narsimha, 2015; Rajesh, Mangathayaru, & Narsimha, 2017; Gunupudi, Mangathayaru, & Narsimha, 2016) the intrusion detection using text processing with Gaussian based similarity measure is introduced with a case study, which demonstrates how process system call feature matrix can be used with the proposed measure to demonstrate the process of intrusion detection. Literature review is outlined in section-2. Section 3 presents the proposed dimensionality reduction using fuzzy membership function. Section 4 presents the proposed algorithm. Section 5 discusses results obtained in the experimentation.

## 2. LITERATURE REVIEW

Design of Intrusion detection using text processing techniques proposed by Rawat and his team in Alok, Arun, & Kuldip (2007), Sanjay, Gulati, & Arun, (1998); Sanjay, Gulati, & Arun (2006) used Binary Cosine Metric as a similarity measure for their experimentation and demonstrated over the DARPA Dataset. Our previous publications present a novel frame work for intrusion detection which involves preprocessing, clustering and classification using novel Gaussian based measure and prediction techniques for intrusion detection and makes use of CANN approach (Wei-Chao, Shih, & Chih, 2015). Vangipuram, Kumar, & Janaki (2016) presents recent techniques which suits to temporal data makes a novel contribution in recent times. Asif, Shadi, & KaziSakib (2016) proposes web data amalgamation for security engineering which performs digital forensic investigation of open source cloud. There were many similarity measures that were proposed for dimensionality reduction. Starting form (Sanjay, Gulati, & Arun, 2005; Hai, Slobodan, & Franke, 2010) some of the proposed methods include SVM Wrapper, Markov Blanket, CART, GeFS methods.

We have proposed new Gaussian based similarity measure which proves to be better measure than the existing ones. Our previous study is tested upon DARPA 98 and KDD Cup 99 datasets and obtained positive results. Using DARPA98 dataset, we have constructed process system call feature matrix and performed the dimensionality reduction using CANN (Wei et al., 2015) approach using Gaussian based similarity measure. For dimensionality reduction we have used the k-Means clustering algorithm with new similarity measure. We used the same measure in kNN algorithm for prediction process. Both the binary process system call feature matrix and frequency based process system call feature matrix (Gunupudi et al., 2016; Gunupudi et al., 2015) are used for experiments.KDD Cup 99 dataset is different from DARPA 98 data set which includes different parameters instead of the system call pattern. In DARPA 98 dataset we have constructed process system call feature matrix as preprocessing and in KDD Cup 98 dataset the preprocessing involves in replacing the nominal

values to numeric values. We got better results after the dimensionality reduction from 60 features to 5 features when we tested the same with different number of features whereas in KDD cup 99 dataset we have obtained better results with 10 dimensions from 42 dimensions.

Tamer, Elkilani, & Abdul-Kader (2015) suggests different metaheuristic techniques in order to generate anomaly detectors. The experimentation is done through NSL KDD cup 99 dataset a modified version of the KDD Cup 99 dataset and achieved accuracy of 96.1%. Aljawarneh et al. (2011), Shadi (2011), Aljawarneh et al. (2016), Shadi A Aljawarneh, Raja A Moftah, & Abdelsalam M Maatuk (2016); Aram, Riccardo, WouterJoosen, & Walden, (2012) describes that there is no one solution which suits to different environments to protect from intrusion detection. A novel approach based on software development life cycle principles is being proposed. Yan, Harvey, & Robin (2011) proposes methodologies to organize the information in project repositories using semantic templates.

JuAn, Hao, MinzheGuo, & Min (2009) proposes an industry standard for vulnerability, the common vulnerabilities and exposures, the common vulnerability scoring system and a vulnerability scoring system designed to provide an public and standardized method for rating of software vulnerabilities. Nam, Tung, HoanAnh, & Nguyen (2010) have developed SecureSync an automatic tool to identify recurring software vulnerabilities. Aram et al. (2012) present vulnerability prediction or a novel approach that influences on the analysis of raw source code as text, in the place of cooked features. Alexander, Alexey, Andrey, Valentin, & Igor Shakhalov (2015) proposes a conceptual model for analyzing and synthesis of controls for secure software development, which allows the programmers to select considerable controls for developing secure software. Waseem, Aron, Yevgeniy, & Koutsoukos (2015) proposes an efficient heuristic algorithm for evaluation of attacks on various networks. Authors also propose design of scheduling schemes for IDS and hence the overall lifetime of the network is maximized.

Bela, Piroska & Kiss (2016) introduces a new framework for designing of RDIDS, a resilient distributed intrusion detection systems. This framework assesses the risk propagation level and will assign a rank that describes critical communication flows. Depending on the rank a shortest path is established to avoid delay in communication. Andre (2015) proposes overview of automotive cyber security challenges and solution approaches. Michael, & Richter (1993) gives application-oriented articles, a broad scope of theoretical, surveys and discussions in the field of pattern recognition, computational statistics analysis, databases, DNA and genome analysis with respect to the internet. Subhashini & Kumar (2010) proposes different similarity measures and document clustering encourages the researchers to use these similarity measures for intrusion detection using text processing techniques. Thaksen & Pravin (2015) proposes deep packet inspection for intrusion detection and with better accuracy and time complexity over existing measures which reduces load over the security appliances to great extent. Muneer Bani Yassein & Shadi Aljawarneh (2017); Aljawarneh, S., Yassein, M.B., & Talafha, W.A. (2017) discusses elastic trickle timer algorithm for IoT for reducing complexity and speedingup of requests. Shadi A Aljawarneh & Muneer O Bani Yassein (2016); Shadi A Aljawarneh, Federica Cena, & Abdelsalam Maatuk (2016); Shadi A. Aljawarneh, Radhakrishna Vangipuram, Veereswara Kumar Puligadda, & Janaki Vinjamuri (2017) proposes various algorithms for discovery of temporal patterns and trends.

Radhakrishna et al. (2014); Vangipuram et al. (2015); Vangipuram et al. (2016) proposes a novel Gaussian based similarity measure for clustering software components and documents and for mining temporal patterns. UjwalaRavale, NileshMarathe, & Padiya (2015) introduced

a feature selection method which uses a K-Means and radial basis kernel function (RBS) over the KDD Cup 99 Dataset.

Byung & Kim (2005) develops the IDS which combines Least Squares Support Vector Machine classier with on-line feature extraction method. Yu-Xin Ding, Min Xiao, & Ai-Wu Liu (2009) presents a signature generation module is based on a variant of Apriori algorithm. Bela et al. (2016) designed algorithm that accounts for the possibility that detection devices may be compromised or fails. ShahidRazaa, Wallgrena, & Voigta (2013) discuss intrusion in context of IoT. AimadKarkouch, HajarMousannif, & Moatassime (2016) aims at highlighting the data quality (DQ) in IoT. Julien, Mazhelis, Xiang, & SasuTarkoma (2016) evaluates, both proprietary and open-source IoT platforms, on the basis of their performance to meet the expectations of different IoT users. David, Jairo, & Ray (2016) presents a futuristic survey and analyzes existing routing protocols and systems to secure routing communications in IoT. Jiang et al. (2016) discusses identification of intrusions with user profiling, robustness of behavioral characteristics, to keep user profile secret, falsify input data to fool the intrusion detection system. Sarah et al. (2016) demonstrates the combination of a one- class SVM with deep learning and develops deep learning auto encoder. Another technique Particle Swam Optimization (PSO) which is robust (Seyed et al., 2015) used to improve the performance of MCLP Classifier and tested over KDD Cup 99 dataset. Yangsun Lee et al. (2016) proposes a new secure system configuration for smart virtual machine for IoT services and proposes a secure compiler for C/C++ programmers. As the data evolved into big data (JanezKranjc et al., 2017), it is becoming more significant how this big data is stored and processed. Wilson, Capretz, & Bittencourt (2016) proposes a system named CEPsim which can used to evaluate and analyze the performance of the data with low latency, complex event processing (CEP) and stream processing (SP)that suits to cloud computing environment. Zheng, Jun, Athanasios, & Yang (2015) presented a future direction for dealing massiveIoT data that is generating from 24 billion devices by 2020 ensuring trustworthy data fusion and mining. Other relatedworks were discussed in Yung et al. (2014) to BuketYksel, AlptekinKp, & Znurzkasap, (2017). Shadi A. Aljawarneh, Ali Alawneh, & Reem Jaradat (2016) discusses cloud security engineering in early stages of SDLC. Shadi Aljawarneh, Bassam Alshargabi, Sofyan MA Hayajneh, & Ayad T Imam (2015); Shadi A. Aljawarneh (2016); Shadi Aljawarneh (2011); Shadi A Aljawarneh, Raja A Moftah, & Abdelsalam M Maatuk (2016) discusses security measures to be taken care in the banking domain.

## 3. PROPOSED DIMENSIONALITY REDUCTION METHOD USING FUZZY MEMBERSHIP FUNCTION

A software process is a function of system calls invoked when a software application is initiated. The main objective of this research is to introduce a novel fuzzy membership function for addressing dimensionality reduction of system process. We use the literal ’P’ to indicate the system process set, ’S’ to indicate system calls set and ’D’ for decision class label set. i.e ’P’, ’S’ and ’D’ are vectors with dimensions denoted by ’p’,’s’ and’d’ respectively. The process vector (P), system call vector (S) and decision class vector (D) are defined using the equations (1) to (3)

$$P = [P]_{1X\,p} = [\,P(1), P(2), P(3)\ldots\ldots P(p)\,] \quad (1)$$

$$S = [S]_{1Xs} = [\,S(1), S(2), S(3)\ldots\ldots S(s)\,] \quad (2)$$

$$D = [D]_{1Xd} = [\,D(1), D(2), D(3)\ldots\ldots D(d)\,] \quad (3)$$

Given, a process-system call matrix representation, $[P\ S]_{pxs}$ with set of decision class labels as normal or abnormal, the goal is to transform the $[P\ S]_{pxs}$matrix with dimensionality 's' into a reduced process system call representation $[P\ S]_{pxr}$ whose dimensionality is 'r'. For achieving this, we use the concept of process pattern motivated from the work (Yung-Shen Lin et al., 2014) and (Jung-Yi Jiang, 2011). A process pattern is a vector of posteriori probabilities computed for each software process w.r.t decision class label. It is of dimension, |d| equal to total number of class label in the set 'D'.Given a matrix representation of process-system call, denoted as [PS]. The notation, P[i,j] denotes, $i^{th}$ process and $j^{th}$ system call element value in the matrix. In similar lines, S[i,j] denotes $j^{th}$system call and ith process element value in the matrix. The notation P [i,j] refers to row values and the notation, S[j,i] represents column values of process-system call matrix representation.

## 3.1 Posteriori Probability of Software System Call w.r.t Decision Class

Given S(j), the probability that the jth system call denoted by S(j) belongs to a decision class, D(d) is given by the equation 4.

$$Pr\left(\frac{D^{(d)}}{S^{(j)}}\right) = \frac{S(1,j) * \mathrm{M}_d + S(2,j) * \mathrm{M}_d + \cdots + S(p,j) * \mathrm{M}_d}{S(1,j) + S(2,j) + \cdots + S(p,j)} \quad (4)$$

Where, $M^d$ is defined as equation 5.

$$M_d = \begin{cases} 1 \ : \ S(i,j) \ \textit{belongs to class label} \ D(d) \\ 0 \ : \ S(i,j) \ \textit{does not belongs to class label} \ D(d) \end{cases} \quad (5)$$

In equations (4) and (5), the variables i and j vary from 1 to p and 1 to s respectively. '$M_d$ ' denotes the membership value of system call w.r.t a given decision class label. If the system call appears in a process and this system call belongs to decision class label say,'d', then the value of parameter '$M_d$' , is assigned to be 1, otherwise it is assigned a value equal to 0. We denote Pr(D(j)/S(j)) as C (j, d) for convenience.

## 3.2 System Call Pattern Vector

The system call pattern vector for jth system call is denoted as C (i), and is represented as the vector of probability values which are computed for $j^{th}$ system call w.r.t each decision class in the set, D. Formally, we denoted it using equation (6)

$$C\ (j) =< C\ (j,1), C\ (j,2) \ldots C\ (j,d\ -\ 1), C\ (j,d) > \quad (6)$$

The system call pattern for j[th] system call is hence given by equation (7)

$$C(j) =< Pr\left(\frac{D(1)}{S(j)}\right),\ Pr\left(\frac{D(2)}{S(j)}\right), \quad \ldots\ldots\ Pr\left(\frac{D(d)}{S(j)}\right) > \quad (7)$$

## 3.3 System Call Pattern Similarity Function

The fuzzy membership function for any two system calls, C(i) and C(j) with corresponding system call pattern probabilities, denoted by p( i, d) and p( j, d) is defined by the equation 8.

$$\mu_{G(P^i,P^j)} = 0.5\ * \left(1 + \exp\left(-\frac{p^i - p^j}{s^i}\right)^2\right) \quad (8)$$

Where $p^{i,d}$ and $p^{j,d}$ are the probability values of i[th] and j[th] system calls w.r.t decision class, d and $\sigma^i$ is the initial chosen deviation.

## 3.4 Membership Function

Equation (8) represents the membership function for a single decision class label, 'd', which is used to compute similarity degree for a given decision class label between two system calls with probabilities as $p^{i,d}$ and $p^{j,d}$ . If the decision class is a vector of class labels with dimension equal to |d|, then the membership function is given by the equation (9)

$$\mu_{G(C(i),C(j))} = \prod_{i=1}^{i=|d|} \mu_{G(C(i),C(j))} = \mu^{\,1}_{(C(i),C(j))_*} \mu^{\,2}_{(C(i),C(j))_{*\ldots.*}} \mu^{\,d}_{(C(i),C(j))} \quad (9)$$

The value obtained from equation (9) lies between 0 and 1. Equation (10) is used to compute membership value of a system call pattern to g[th] cluster with mean vector denoted by 'M'

$$\mu_{G(C(i),M(g))} = \prod_{i=1}^{i=|d|} \mu_{G(C(i),M(g))} = \mu^{\,1}_{(C(i),M(g))_*} \mu^{\,2}_{(C(i),M(g))_{*\ldots.*}} \mu^{\,d}_{(C(i),M(g))} \quad (10)$$

# 4. ALGORITHM FOR DIMENSIONALITY REDUCTION

Algorithm for Dimensionality Reduction is as follows:

***Step-1: Read threshold, global vector and class labels.***

Consider the process-system call matrix, [PS] with decision label, and from this obtain the system call pattern vectors computing posteriori probabilities of system call w.r.t every decision class label using the equations (1) to (4).

***Step-2: Create first cluster.***

Generate first cluster, H1 and place the first system call pattern vector, V1 in this cluster. Choose the initial cluster deviation value preferably not zero and not exceeding 1. The mean of this newly created cluster is the system call pattern vector, V1.

***Step-3: Generate Clusters.***

Choose each pattern vector at a time and then compute the membership value of this pattern with the existing cluster using the membership function. If the membership value obtained satisfies the similarity condition, add the pattern to the cluster. Otherwise generate a new cluster and place this pattern in that cluster. The newly generated cluster mean shall be the pattern vector which failed with the existing cluster. In this process, when a system call pattern is tested for computing membership value to existing clusters, if more than one cluster satisfies the similarity constraint, then the best choice for moving this pattern shall be the cluster to which the current system call pattern membership value is the maximum.

***Step-4: Update the mean and deviation of generated clusters***

After all patterns are clustered using evolutionary approach, update the mean and deviation for these clusters. The mean for the final clusters is the mean computed considering all patterns within the generated cluster. The new deviation is the sum of assumed deviation and the deviation obtained by considering the patterns within clusters.

***Step-5: Obtain the membership values of system call patterns to the generated clusters***

Consider each system call pattern and compute the membership value of the system call pattern to each generated cluster. The pattern is now assigned to the cluster to which the membership value is the maximum.

***Step-6: Obtain the mapping matrix***

From step-5, obtain the system call pattern vs. Cluster matrix. This will be the optimal transformation matrix for dimensionality reduction. Let C denote clusters and S denotes the system call pattern, then the notation [SXC] denotes the system call vs. cluster matrix.

## 5. CASE STUDY

The process-system call information with decision class is given in Table 1. The system call pattern vectors are given in Table 2. They are computed for every system call. The dimensionality is two as there are only two decision labels normal and abnormal. Table 3 shows membership value obtained between patterns, C (1) and C (2). The deviation and mean of three generated clusters are listed in Table 4. The transformation matrix generated is given in Table 5 and Table 6 gives the final reduced process matrix in frequency form. The distribution of dimensions w.r.t original process matrix in Table 1 and Table 6 remains same. Table 7 shows the reduced representation of process with class label.

Table 1. Process –System Call Representation

| PROCESS | | System Calls | | | | | | | | | | Decision |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SC1 | SC2 | SC3 | SC4 | SC5 | SC6 | SC7 | SC8 | SC9 | SC10 | |
| | Process-1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | NORMAL |
| | Process-2 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 1 | 0 | 0 | NORMAL |
| | Process-3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | NORMAL |
| | Process-4 | 0 | 0 | 1 | 0 | 2 | 1 | 2 | 1 | 0 | 1 | NORMAL |
| | Process-5 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | ABNORMAL |
| | Process-6 | 2 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | ABNORMAL |
| | Process-7 | 3 | 2 | 1 | 3 | 0 | 1 | 0 | 1 | 1 | 0 | ABNORMAL |
| | Process-8 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | ABNORMAL |
| | Process-9 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ABNORMAL |

Table 2. System Call Pattern

| System Call | System Call Pattern |
|---|---|
| SC1 | C(1) =<0.0,1.0> |
| SC2 | C(2)=<0.2,0.8> |
| SC3 | C(3)=<0.2,0.8> |
| SC4 | C(4)=<0.0,1.0> |
| SC5 | C(5)=<1.0,0.0> |
| SC6 | C(6)=<0.5,0.5> |
| SC7 | C(7)=<1.0,0.0> |
| SC8 | C(8)=<0.67,0.33> |
| SC9 | C(9)=<0.0,1.0> |
| SC10 | C(10)=<1.0,0.0> |

Table 3. Computation of System Call Pattern

| Pattern1 | Pattern2 | Membership value proposed | Membership value[46] |
|---|---|---|---|
| C(1)=<0.0,1.0> | C(2)=<0.2,0.8> | 0.85761 | 0.7262 |

Table 4. System Call Clusters

| | Cluster Mean | | Cluster Deviation | |
|---|---|---|---|---|
| Cluster-G1 | 0.08 | 0.92 | 0.609545 | 0.609545 |
| Cluster-G2 | 1 | 0 | 0.5 | 0.5 |
| Cluster-G3 | 0.585 | 0.415 | 0.620208 | 0.620208 |

Table 5. System Call Pattern Membership to Clusters in Binary Form

| | Transformation matrix | | |
|---|---|---|---|
| | Cluster-G1 | Cluster-G2 | Cluster-G3 |
| C(1) | 1 | 0 | 0 |
| C(2) | 1 | 0 | 0 |
| C(3) | 1 | 0 | 0 |
| C(4) | 1 | 0 | 0 |
| C(5) | 0 | 1 | 0 |
| C(6) | 0 | 0 | 1 |
| C(7) | 0 | 1 | 0 |
| C(8) | 0 | 0 | 1 |
| C(9) | 1 | 0 | 0 |
| C(10) | 0 | 1 | 0 |

Table 6. Reduced Process Matrix in Frequency Form

| | Transformation matrix | | |
|---|---|---|---|
| | Cluster -G1 | Cluster -G2 | Cluster -G3 |
| Process-1 | 1 | 2 | 1 |
| Process-2 | 0 | 1 | 3 |
| Process-3 | 0 | 1 | 0 |
| Process-4 | 1 | 5 | 2 |
| Process-5 | 2 | 0 | 1 |
| Process-6 | 5 | 0 | 1 |
| Process-7 | 10 | 0 | 2 |
| Process-8 | 3 | 0 | 1 |
| Process-9 | 4 | 0 | 0 |

Table 7. Reduced Representation with Class Labels

| | Reduced Process Matrix | | | Class |
|---|---|---|---|---|
| | D1 | D2 | D3 | |
| Process-1 | 1 | 2 | 1 | NORMAL |
| Process-2 | 0 | 1 | 3 | NORMAL |
| Process-3 | 0 | 1 | 0 | NORMAL |
| Process-4 | 1 | 5 | 2 | NORMAL |
| Process-5 | 2 | 0 | 1 | ABNORMAL |
| Process-6 | 5 | 0 | 1 | ABNORMAL |
| Process-7 | 10 | 0 | 2 | ABNORMAL |
| Process-8 | 3 | 0 | 1 | ABNORMAL |
| Process-9 | 4 | 0 | 0 | ABNORMAL |

## 6. RESULTS AND DISCUSSIONS

For the experimentation the NSL-KDD Cup 99 dataset was considered. In the previous works by (Lin et al., 2015) the experimentation is unable to detect the R2L and U2R attacks properly. With the proposed function, we have observed that the accuracy of the intrusion detection is greatly improved. The figure 1 shows the attributes of the NSL KDD Cup dataset.

| | | | |
|---|---|---|---|
| duration | logged_in | count | dst_host_same_srv_rate |
| protocol_type | num_compromised | srv_count | dst_host_diff_srv_rate |
| service | root_shell | serror_rate | dst_host_same_src_port_rate |
| flag | su_attempted | srv_serror_rate | dst_host_srv_diff_host_rate |
| src_bytes | num_root | rerror_rate | dst_host_serror_rate |
| dst_bytes | num_file_creations | srv_rerror_rate | dst_host_srv_serror_rate |
| land | num_shells | same_srv_rate | dst_host_rerror_rate |
| wrong_fragment | num_access_files | diff_srv_rate | dst_host_srv_rerror_rate |
| urgent | num_outbound_cmds | srv_diff_host_rate | |
| hot | is_host_login | dst_host_count | |
| num_failed_logins | is_guest_login | dst_host_srv_count | |

Figure 1. 41Attributes of the NSL KDD Cup Dataset

Table 8. Detection Accuracy for Correctly Classified instances for KDD Cup 99 Dataset

| No of Features after DR | Threshold chosen | Detection Accuracy – Correctly Classified | | | |
|---|---|---|---|---|---|
| | | kNN (k=1) | kNN (k=3) | kNN (k=5) | J.48 |
| 32 | 0.9995 | 99.1547 % | 98.9667 % | 98.7505 % | 98.5465 % |
| 34 | 0.9999 | 99.2047 % | 99.0111 % | 98.786 % | 98.5626 % |
| 19 | - | 99.1697 % | 99.0681 % | 98.9641 % | 99.5277 % |
| 2 | 0.9995 | 92.9882 % | 92.3142 % | 91.8371 % | 90.0066 % |
| 7 | 0.99995 | 93.3875 % | 93.093 % | 92.9167 % | 91.1179 % |
| 41 | - | 99.6896 % | 99.5975 % | 99.5356 % | 99.7865 % |

In the Table 8, the experimentation is done for the various combinations such as with threshold values chosen as 0.9995, 0.9999 over the different number of dimensions obtained after the running the dimensionality reduction using proposed approach. The accuracies for correctly classified is mentioned in the table.

Table 9. Confusion Matrix of KNN for Threshold = 0.9995, k=5 after Dimensionality Reduction Using Proposed Approach

| | Normal | DoS | Probe | R2L | U2R | Accuracies |
|---|---|---|---|---|---|---|
| Normal | 65831 | 148 | 284 | 41 | 1 | 99.50% |
| DoS | 100 | 44811 | 266 | 0 | 0 | 98.70% |
| Probe | 163 | 450 | 10854 | 0 | 0 | 95.20% |
| R2L | 54 | 1 | 1 | 916 | 0 | 95.60% |
| U2R | 39 | 0 | 0 | 1 | 12 | 92.30% |

Table 10. Confusion Matrix CANN over the 6-Dimensional Dataset

| Predicted | | | | | | Accuracy (%) |
|---|---|---|---|---|---|---|
| | Normal | Probe | DoS | U2R | R2L | |
| *Actual* | | | | | | |
| Normal | 97,275 | 0 | 2 | 0 | 0 | 99.99 |
| Probe | 0 | 4106 | 1 | 0 | 0 | 99.98 |
| Dos | 2 | 0 | 391,456 | 0 | 0 | 99.99 |
| U2R | 52 | 0 | 0 | 0 | 0 | 0 |
| R2L | 1126 | 0 | 0 | 0 | 0 | 0 |

Table 11. Confusion Matrix CANN over the 19-Dimensional Dataset

| Predicted | | | | | | Accuracy (%) |
|---|---|---|---|---|---|---|
| | Normal | Probe | DoS | U2R | R2L | |
| *Actual* | | | | | | |
| Normal | 94,398 | 221 | 2130 | 35 | 493 | 97.04 |
| Probe | 201 | 3598 | 306 | 1 | 1 | 87.61 |
| Dos | 1076 | 177 | 390,190 | 8 | 7 | 99.68 |
| U2R | 36 | 1 | 11 | 2 | 2 | 3.85 |
| R2L | 471 | 1 | 10 | 2 | 642 | 57.02 |

It can be observed from the above tables 10 and 11; the U2R and R2L attacks could not be identified by the (Lin et al., 2015) using CANN approach. In this paper we made a significant attempt by proposing novel approach for dimensionality reduction using different accuracies which is detecting the U2R and R2L attacks. Table 9 presents using the proposed approach a sharp improvement of the U2R and R2L attacks over CANN approach.

Table 12. Confusion Matrix of KNN for k=1 over NSL-KDD 99 after Dimensionality Reduction for Threshold = 0.9995 with 3 Classes

| | Normal | DoS | Probe | R2L | U2R | Accuracies |
|---|---|---|---|---|---|---|
| Normal | 63885 | 1284 | 245 | 1888 | 41 | 96.10% |
| DoS | 976 | 43774 | 28 | 1149 | 0 | 94.00% |
| Probe | 1400 | 1488 | 33 | 8734 | 1 | 74.00% |
| R2L | 206 | 19 | 736 | 33 | 1 | 70.60% |
| U2R | 40 | 0 | 0 | 1 | 11 | 20.40% |

From the tables 12, 13 and 14 it can be observed that after the dimensionality reduction to 3 dimensions from 41 attributes, the R2L and U2R detection is reduced when compared to Table 9. It indicates that number of the attributes after the dimensionality reduction is greatly influences the detection rate. That is why we get different results for different thresholds which directly decide the number of the dimensions after the DR process.

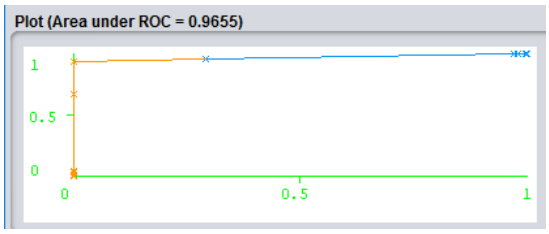
Table 13. Confusion Matrix of KNN for k=3 over NSL-KDD 99 after Dimensionality Reduction for Threshold = 0.9995 with 3 Classes

| | Normal | DoS | Probe | R2L | U2R | Accuracies |
|---|---|---|---|---|---|---|
| Normal | 63963 | 1185 | 233 | 1949 | 13 | 94.60% |
| DoS | 1253 | 43754 | 29 | 891 | 0 | 93.90% |
| Probe | 2130 | 1656 | 17 | 7853 | 0 | 73.20% |
| R2L | 230 | 17 | 719 | 29 | 0 | 72.00% |
| U2R | 50 | 0 | 0 | 0 | 2 | 13.30% |

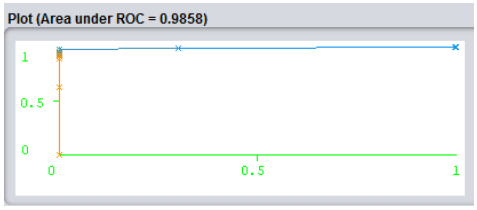
Table 14. Confusion Matrix of KNN for k=5 over NSL-KDD 99 after Dimensionality Reduction for Threshold = 0.9995 with 3 Classes

| | Normal | DoS | Probe | R2L | U2R | Accuracies |
|---|---|---|---|---|---|---|
| Normal | 63680 | 1504 | 237 | 1919 | 3 | 94.30% |
| DoS | 1246 | 43876 | 31 | 774 | 0 | 92.90% |
| Probe | 2359 | 1840 | 36 | 7421 | 0 | 73.10% |
| R2L | 228 | 20 | 713 | 34 | 0 | 70.10% |
| U2R | 51 | 1 | 0 | 0 | 0 | 0.00% |

ROC of R2L over NSL KDD Data with 41 Attributes before Dimensionality Reduction using KNN for k=1 Accuracy = 94.3 %




ROC of R2L using KNN for k=5 for 33 attributes with threshold = 0.9995, Accuracy = 95.6 %




ROC of R2L using KNN for k=5 for 35 attributes with threshold = 0.9999 , Accuracy = 95.6 %

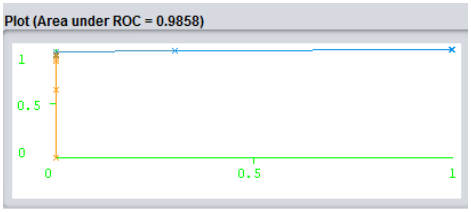



Figure 2. ROC Curves of R2L over NSL KDD Data with 41 Attributes Before and After DR for k=5

Figures 2 and 3 shows ROC curves for U2R and R2L attacks which is a significant improvement over previous works i.e CANN approach, Lin et al (2015). For R2L attack, area under ROC before DR is 0.9655 as seen in Figure 2. The area under ROC curve after dimensionality reduction is 0.9858 for thresholds 0.9995 and 0.9999. This value is shown in ROC plot of figure 2.

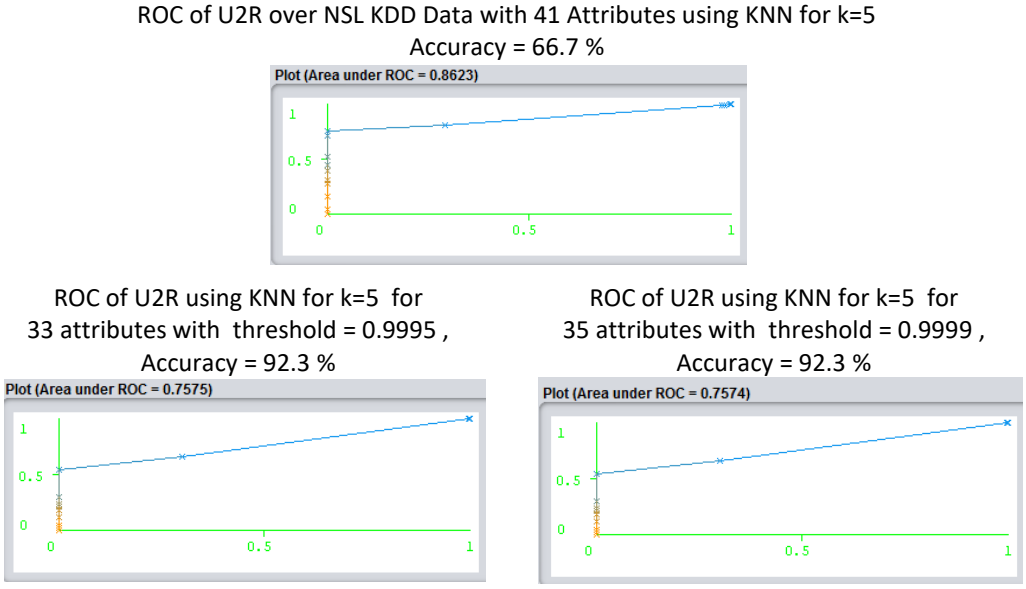



Figure 3. ROC Curves of U2Rover NSL KDD Data with 41 Attributes Before and After DR for k=5

For U2R attack, area under ROC before DR is 0.8623 as seen in Figure 3. The area under ROC curve after dimensionality reduction is 0.757 for thresholds 0.9995 and 0.9999. This value is shown in ROC plot of figure 3.

From experimental results recorded in figure 2, it is seen that the R2L attack accuracy achieved is 94.3% using kNN classifier for k=1. For k=5, with thresholds 0.9995 and 0.9999, feature dimensions are 33 and 35 respectively and recorded R2L attack accuracy is 95.6%. It can be easily deduced from figure 3, that U2R accuracy achieved considering all 41 attributes with our dimensionality reduction is 66.7% using kNN classifier for k=5. Using our proposed method, FCBDR with reduced dimensions (33), the classifier accuracies are 92.3% for both cases for a chosen threshold equal to 0.9995 and 0.9998 respectively.

## 7. CONCLUSIONS

Intrusion detection and data mining are implicitly related research areas. The computational complexity of intrusion detection systems and algorithms adopted depends on handling the dimensionality problem without losing the original flavor of information. Recently, the research on intrusion detection systems is into addressing challenges of handling dimensionality. This work concentrates on defining a membership function which is suitable to handle the high dimensionality problem by applying text data mining techniques. The dimensionality reduction is carried out by retaining the original feature distribution. The reduced dimensional representation may be used to perform classification, clustering and prediction. A case study discussed outlines the working model with example. The ROC curve and experimental results of U2R and R2L attacks show that our proposed approach is giving better classification when compared to others.

# REFERENCES


Aimad, K., Hajar, M., Hassan A. M., & Thomas N. (2016). Data quality in internet of things: A state-of-the-art survey.*Journal of Network and Computer Applications*, *73*, 57-81.

Alexander B., Alexey M., Andrey F., Valentin T., & Igor S. (2015). Synthesis of secure software development controls.*Proceedings of the 8th International Conference on Security of Information and Networks*. 93-97.doi:10.1145/2799979.2799998

Aljawarneh, S. (2011). Cloud security engineering: avoiding security threats the right way. *International Journal of Cloud Applications and Computing,1*(2), 64-70.

Aljawarneh, S. A., Moftah, R. A., & Maatuk, A. M. (2016). Investigations of automatic methods for detecting the polymorphic worm signatures. *Future Generation Computer Systems*, *60*, 67-77.

Aljawarneh, S., Yassein, M.B., & Talafha, W.A. (2017). A resource-efficient encryption algorithm for multimedia big data. *Multimedia Tools and Applications*, 1–22. doi:10.1007/s11042-016-4333-y

Alok Sharma, Arun K. Pujari, & Kuldip K. Paliwal. (2007). Intrusion detection using text processing techniques with a kernel based similarity measure. *Computers & Security*,*26*(78), 488-495.

Anderson, Ross. (2001). *Security Engineering: A Guide to Building Dependable Distributed Systems*. New York: John Wiley & Sons. 3873-88.

Andre Weimerskirch. (2015). An overview ofautomotive cybersecurity: challenges and solution approaches.*Proceedings of the 5th International Workshop on Trustworthy Embedded Devices*, 53-53.doi:10.1145/2808414.2808423

Aram Hovsepyan, Riccardo Scandariato, WouterJoose, & James Walden. (2012). Software vulnerability prediction using text analysis techniques.*Proceedings of the 4th international workshop on security measurements and metrics*, ACM, New York, USA. 7-10. doi:10.1145/2372225.2372230

Asif Imran, Shadi Aljawarneh, & KaziSakib. (2016). Web data amalgamation for security engineering: digital forensic investigation of open source cloud. *Journal of Universal Computer Science ,22*(4), 494-520.

Bela Genge, Piroska Haller, & Istvan Kiss. (2016). A framework for designing resilient distributed intrusion detection systems for critical infrastructures. *International Journal of Critical Infrastructure Protection, 15*, 3-11.

Buket Yüksel, Alptekin Küpçü, & Öznur Özkasap. (2017). Research issues for privacy and security of electronic health services, *Future Generation Computer Systems, 68*, 1-13. doi:10.1016/j.future.2016.08.011

Byung-joo Kim, & Il-kon Kim. (2005, July). Kernel based intrusion detection system.ACIS International Conference on Computer and Information Science, 13-18.doi:10.1109/ICIS.2005.78

David Airehrour, Jairo Gutierrez, & Sayan Kumar Ray. (2016).Secure routing for internet of things: A survey. *Elsevier, Journal of Network and Computer Applications. 66*, 198-213.

Faisal Alkhateeb, Zain Al-Abdeen Al-Fakhry, Eslam Al Maghayreh, Ahmad T. Al-Taani & Shadi Aljawarneh. (2015). Integration of wireless technologies in smart university campus environment: framework architecture. *International Journal of Information and Communication Technology Education, 11*(1), 60-74.

Gangin Lee, & Unil Yun. (2017). A new efficient approach for mining uncertain frequent patterns using minimum data structure without false positives. *Future Generation Computer Systems, 68*, 89-110.

Gunupudi, R. K., Mangathayaru, N., &Narsimha, G. (2015). A novel similarity measure for intrusion detection using gaussian function. *Revista Tecnica De La Facultad De Ingenieria Universidad Del Zulia, 39*(2), 173-183.

Gunupudi, R. K., Mangathayaru, N., & Narsimha, G. (2016). An approach for intrusion detection using novel gaussian based kernel function. *Journal of Universal Computer Science, 22*(4), 589-604

Gunupudi, R. K., Mangathayaru, N., & Narsimha, G. (2016). Intrusion detection a text mining based approach.*Special issue on Computing Applications and Data Mining International Journal of Computer Science and Information Security (IJCSIS),14*, 76-88.

Gunupudi, R. K., Mangathayaru, N., & Narsimha, G. (2015). Intrusion detection using text processing techniques: a recent survey.*Proceedings of the International Conference on Engineering & MIS*, ACM, New York, USA.doi:10.1145/2832987.2833067

Hai, T. N., Slobodan P., & Katrin F. (2010). A comparison of feature-selection methods for intrusion detection. *Proceedings of the 5th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security*, 242-255

Janez Kranjc, Roman Orač, Vid Podpečan, Nada Lavrač, & Marko, Robnik-Šikonja. (2017). ClowdFlows: Online workflows for distributed big data mining, *Future Generation Computer Systems, 68*, 38-58.

Jian Peng, Kim-Kwang Raymond Choo, & Helen Ashman. (2016). User profiling in intrusion detection: A review. *Journal of Network and Computer Applications*, *72*, 14-27.

Jiang, J. Y, Liou, R. J, & Lee, S. J. (2010). A fuzzy self-constructing feature clustering algorithm for text classification. IEEE Transactions on Knowledge and Data Engineering, 23(3), 335-349. doi: 10.1109/TKDE.2010.122

JuAn Wang, Hao Wang, MinzheGuo, & Min Xia. (2009). Security metrics for software systems. *Proceedings of the 47th Annual Southeast Regional Conference*, ACM, Clemson, South Carolina. Article No. 47.doi:10.1145/1566445.1566509

Julien Mineraud, Oleksiy Mazhelis, Xiang Su, & SasuTarkoma. (2016). A gap analysis of Internet-of-Things platforms, *Elsevier, Computer Communications, 89-90*, 5-16.

Mahbod, Tavallaee, Ebrahim, Bagheri, Wei, Lu, & Ali, A. Ghorbani. (2009). A detailed analysis of the KDD CUP 99 data set. *Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications*. IEEE Press, Piscataway, NJ, USA, 53-58.Abstract retrieved from http://dl.acm.org/citation.cfm?id=1736489

Michael,& M. Richter. (1993). Classification and learning of similarity measures, information and classification.(Eds.), Dr. Otto Opitz, Dr. Berthold Lausen, Dr. Rüdiger Klar, *Classification, Data Analysis and Knowledge Organization*, 323-334. Springer International Publishing AG. Part of Springer Nature.

Michael, E. Whitman, & Herbert, J. Mattord. (2008). Principles of information security. Course Technology, Cengage Learning.

Muneer Bani Yassein, & Shadi Aljawarneh. (2017). A new elastic trickle timer algorithm for Internet of Things. Journal of Network and Computer Applications. doi:10.1016/j.jnca.2017.01.024

Nam, H. Pham, Tung Thanh Nguyen, HoanAnh Nguyen, & Tien N. Nguyen. (2010). Detection of recurring software vulnerabilities.*Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*. ACM, New York, NY, USA, 447-456. doi:10.1145/1858996.1859089

Preeti Aggarwal, & Sudhir Kumar Sharma. (2015). Analysis of KDD dataset attributes - class wise for intrusion detection, *Procedia Computer Science, 57*, 842-851. doi:10.1016/j.procs.2015.07.490

Rajesh Kumar Gunupudi, Mangathayaru Nimmala,Narsimha Gugulothu, & Suresh Reddy, Gali. (2017). CLAPP: A self constructing feature clustering approach for anomaly detection, *Future Generation Computer Systems*. doi:10.1016/j.future.2016.12.040

S. Aljawarneh, V. Radhakrishna, P. V. Kumar & V. Janaki, (2016). A similarity measure for temporal pattern discovery in time series data generated by IoT. Proceedings of the International Conference on Engineering and MIS, 1-4. doi: 10.1109/ICEMIS.2016.7745355

Sanjay Rawat, Arun K. Pujari, Gulati, V. P. (2006). On the use of singular value decomposition for a fast intrusion detection system. *Electronic Notes in Theoretical Computer Science, 142*(3), 215-228.

Sanjay Rawat, Gulati, V. P.,Arun, K. Pujari, & Vemuri, V. Rao. (1998). Intrusion detection using text processing techniques with a binary-weighted cosine metric.*Proceedings of International Conference on Digital Libraries*, 89-98.Retrieved from http://web.cs.ucdavis.edu/~vemuri/papers/rawat-vemuri.pdf

Sanjay Rawat, Gulati, V. P, & Arun K. Pujari. (2005). A fast host-based intrusion detection system using rough set theory. *Transactions on Rough Sets IV, 3700*,144-161.

Sarah, M. Erfani, SutharshanRajasegarar, Shanika Karunasekera, & Christopher Leckie. (2016). High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, *Pattern Recognition, 58*, 121-134.

SeyedMojtaba, Hosseini Bamakan, Behnam Amiri, & Yong Shi. (2015). A new intrusion detection approach using PSO based multiple criteria linear programming, *Procedia Computer Science, 55*, 231- 237. doi:10.1016/j.procs.2015.07.040

Shadi Aljawarneh. (2011). Cloud Security Engineering: Avoiding security threats the right way. *International Journal of Cloud Applications and Computing (IJCAC). 1*(2), 64-70.

Shadi Aljawarneh, Bassam Alshargabi, Sofyan MA Hayajneh, & Ayad T Imam. (2015). Integration of e-learning and cloud computing platform through software engineering. *Recent Patents on Computer Science, 8*(2), 100-105.

Shadi A Aljawarneh, Federica Cena, & Abdelsalam Maatuk. (2016). Advanced research on software security design and applications. *Journal of Universal Computer Science, 22*(4), 453-458.

Shadi A Aljawarneh,& Muneer O Bani Yassein. (2016). A conceptual security framework for cloud computing issues. *International Journal of Intelligent Information Technologies (IJIIT), 12*(2), 12-24.

Shadi A Aljawarneh, Raja A Moftah, & Abdelsalam M Maatuk. (2016). Investigations of automatic methods for detecting the polymorphic worms signatures. *Future Generation Computer Systems, 60*, 67–77.

Shadi Aljawarneh, Ali Alawneh, & Reem Jaradat. (2016). Cloud security engineering: Early stages of SDLC, Future Generation Computer Systems, doi: 10.1016/j.future.2016.10.005

Shadi Aljawarneh. (2016). Online banking security measures and data protection. IGI Global. doi:10.4018/978-1-5225-0864-9

Shadi Aljawarneh. (2011). A web engineering security methodology for e-learning systems, *Network Security, 2011*(3), 12-15.doi:10.1016/S1353-4858(11)70026-5

Shadi A. Aljawarneh, Radhakrishna Vangipuram, Veereswara Kumar Puligadda, & Janaki Vinjamuri. (2017). G-SPAMINE: An approach to discover temporal association patterns and trends in internet of things. Future Generation Computer Systems. doi: 10.1016/j.future.2017.01.013

ShahidRazaa, Linus Wallgrena, & Thiemo Voigta. (2013). SVELTE: Real-time intrusion detection in the Internet of Things, *Ad Hoc Networks, 11*(8), 2661-2674.

Subhashini, R., & Kumar, V. J. S. (2010). Evaluating the performance of similarity measures used in document clustering and information retrieval. *Proceedings of the First International Conference on Integrated Intelligent Computing*, 27-31.doi:10.1109/ICIIC.2010.42

Tamer F. Ghanem, Wail S. Elkilani, & Hatem M. Abdul-Kader. (2015). A hybrid approach for efficient anomaly detection using metaheuristic methods. *Journal of Advanced Research, 6*(4), 609-619.

Thaksen J Parvat, &Pravin Chandra. (2015). A Novel approach to deep packet inspection for intrusion detection, *Procedia Computer Science,45*, 506-13.

UjwalaRavale, NileshMarathe, & Puja Padiya. (2015). Feature selection based hybrid anomaly intrusion detection system using k means and rbf kernel function.*Procedia Computer Science, 45*,428- 435.

Radhakrishna Vangipuram, Kumar, P.V., & Janaki, V. (2016). An efficient approach to find similar temporal association patterns performing only single database scan. *Revista Tecnica De La Facultad De Ingenieria Universidad Del Zulia,39*(1), 241-255.

Radhakrishna Vangipuram, Kumar, P.V., & Janaki, V. (2015). A novel approach for mining similarity profiled temporal association patterns,*Revista Tecnica De La Facultad De Ingenieria Universidad Del Zulia, 38(3),* 80-93.

Vangipuram Radhakrishna, Chintakindi Srinivas, & Guru Rao, C.V. (2014). A modified Gaussian similarity measure for clustering software components and documents. *Proceedings of the International Conference on Information Systems and Design of Communication,* Lisbon, Portugal. doi:10.1145/2618168.2618184

Vangipuram Radhakrishna, P. V. Kumar, & V. Janaki. (2016). Mining outlier temporal association patterns.*Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies.*1-6.doi:/10.1145/2905055.2905320

Vangipuram Radhakrishna, Kumar, P.V., & V.Janaki. (2016). A novel similar temporal system call pattern mining for efficient intrusion detection. *Journal of Universal Computer Science, 22*(4), 475-493

Waseem, Abbas, Aron, Laszka, Yevgeniy Vorobeychik, & Xenofon, Koutsoukos. (2015). Scheduling intrusion detection systems in resource-bounded cyber-physical systems. *Proceedings of the First ACM Workshop on Cyber Physical Systems Security and/or PrivaCy*, Denver, Colorado, USA, 55-66. doi:10.1145/2808705

Wei-Chao, Lin, Shih-Wen, Ke, & Chih-Fong, Tsai. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledgebase Systems, 78*, 13-21.

Wilson, A. Higashino., Miriam, A.M., Capretz., & Luiz F. Bittencourt. (2016). CEP- Sim: Modeling and simulation of complex event processing systems in cloud environments. *Future Generation Computer Systems*, *65*, 1221-39.

Yan Wu, Harvey Siy, & Robin Gandhi. (2011*).* Empirical results on the study of software vulnerabilities (NIER track).*Proceedings of the 33rd International Conference on Software Engineering* ACM, New York, NY, USA, 964-967.doi:10.1145/1985793.1985960

YangSun Lee, JunhoJeong, & Yunsik Son. (2016). Design and implementation of the secure compiler and virtual machine for developing secure IoT services. *Future Generation Computer Systems.*doi:10.1016/j.future.2016.03.014

Yung-Shen Lin, Jung-Yi Jiang, & Shie-Jue Lee. (2014). A similarity measure for text classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*,*26*(7), 1575-1590.

Yu-Xin Ding, Min Xiao, & Ai-Wu Liu. (2009). Research and implementation on snort-based hybrid intrusion detection system*. Proceedings of International Conference on Machine Learning and Cybernetics,* Baoding, China. doi:10.1109/ICMLC.2009.5212282

Zheng, Yan, Jun, Liu, & Athanasios, V. Vasilakos, Laurence T. Yang. (2015). Trust worthy data fusion and mining in Internet of Things. *Future Generation Computer Systems, 49*, 45-46.