IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Similarity Based Feature Transformation for Network Anomaly Detection

Arun Nagaraja[1], (MEMBER, IEEE), B.Uma[2], Khalaf Khatatneh[3], V.Radhakrishna[4], (MEMBER, IEEE), N.Rajasekhar[5], (MEMBER, IEEE), V.Sravan kiran[6]

[1]Department of Information Science and Engineering Jain University, Bangalore, India
[2]Department of Information Science and Engineering, Malnad College of Engineering, Hassan, India
[3]Department of Computer Science, Al Balqa Applied University, As-Salt, Jordan
[4,6]Department of Information Technology, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India
[5]Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India

Corresponding author: N.Rajasekhar (e-mail: drnrajasekhar@gmail.com, n.rajasekhar@ieee.org)

ABSTRACT The fundamental objective behind any network intrusion detection system is to automate the detection process whenever intrusions occur in the network. The problem of the network anomaly detection is to determine, if the network incoming traffic is legitimate (or) anomalous. Automated detection systems designed to identify incoming anomalous traffic patterns usually apply widely used machine learning techniques. However, irrespective of any system model which is developed to identify anomalous traffic, all these models requires comparing anomalous and normal traffic patterns. Such comparisons implicitly depend on the ability of the underlying machine learning model to gauge the similarity between a known legitimate observation and the target. The efficiency of any network anomalous detection system depends on the use of distance (or) similarity measures and how they are actually applied. A novel distance function which can be applied to determine the similarity between two conditional feature pattern vectors is an important contribution of present research. Feature dimensionality is another important issue for any machine learning algorithm. In the present work, feature reduction is achieved using the proposed feature transformation technique. However, our approach for feature transformation uses the proposed gaussian distance function to achieve dimensionality reduction to represent the original input dataset in the new transformation space. We have also proposed new computation expressions for determining equivalent deviation and threshold in gaussian space. Experiments are performed on KDD and NSL-KDD datasets by considering widely applied classifier algorithms in various state-of-art research contributions. For performance validation of machine learning models, k-fold cross validation is applied by setting k to 10 through considering evaluation parameters such as accuracy, precision and recall. Experiment results have proved that our approach for anomaly detection that applies the proposed feature transformation technique proved comparatively better to detection methods CANN, GARUDA, and UTTAMA addressed in the recent research literature.

INDEX TERMS Similarity function, Feature clustering, Intrusion, Conditional feature pattern vector, Anomaly detection

I. INTRODUCTION

The fundamental purpose of any network anomaly detection system is to precisely and methodically detect diverse types of malicious traffic patterns that may not be detected by conventional firewall systems. Designing a potential and powerful intrusion detection system has three essential challenges. These three challenges are i) Addressing the high dimensionality problem of input observations ii) Applying the appropriate machine learning technique which does not suffer from issues such as overfitting and underfitting and iii) Choosing the appropriate distance measure (or similarity measure) to gauge the similarity between any two network observations.

Feature selection [1], Feature representation [19][20] and dimensionality reduction approaches [21][22] [23] have been studied and extensively addressed in many research contributions related to text classification, data fusion, image fusion, medical data classification and various machine

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2975716, IEEE Access

learning and data mining applications. Feature reduction techniques are also applied for the design of intrusion detection systems (IDS) [19][20] in the literature. Several studies are also carried on how to choose a right classifier and apply it for building efficient network intrusion detection [1]. The performance of NIDS is implicitly related to the choice of distance measures [18][19] that are applied by IDS for reaching a decision, if an incoming observation is actually normal (or) an abnormal one. A relatively little effort is made by researchers to devise new distance functions [19][20] that can be applied by NIDS for efficient intrusion and anomaly detection.

The recent studies such as CANN [23], CLAPP [20], and UTTAMA [24] have applied feature reduction techniques to improve accuracy and detection rates of IDS. The distance measure applied by CANN is the Euclidean distance function. CLAPP, UTTAMA approaches have applied membership functions for the learning process. However these studies did not propose novel similarity measures for carrying unsupervised feature learning and supervised learning tasks. Although CANN [23] has reduced time consumed by classifiers, the detection accuracies of U2R and R2L classes have not been so promising. For example, detection accuracies of U2R and R2L classes are almost zero for CANN. Although CLAPP and UTTAMA have attempted to improve detection accuracies of U2R and R2L attack classes, but these approaches were just limited to applying membership functions. Fundamentally, the contribution addressed in our paper is mainly motivated from all these studies.

The fundamental objective of the present work is to address the challenge in detecting U2R and R2L attacks with higher accuracy, precision, recall rates by obtaining an equivalent representation of the original dataset through projecting it on to a new transformation space. Another important aim of the present study is to recommend a novel distance measure that can be used to perform similarity computations for feature clustering, feature representation, and supervised learning for efficient intrusion detection.

The organization of the paper is as follows. Section II summarizes the state of art research contributions which are the main basis of motivation for the proposed work. Section-III describes the proposed approach and algorithms for feature transformation and supervised learning; Section-IV outlines various experimental results obtained using both the proposed and existing methods. Finally, Section-V summarizes important findings and concludes the paper.

II. BACKGROUND AND MOTIVATION

The distance function introduced in this paper is motivated from several state-of-art research contributions that have proposed distance functions for text classification, temporal pattern mining, software component classification and medical data classification. Distance measures and similarity measures are widely applied in various data mining and machine learning algorithms that require distance (or) similarity computations to be carried as part of algorithm processing. One of the recent contributions that motivated the present contribution is the work by Jiang et.al [1]. In the study reported by Jiang [1], an approach for reducing dimensionality of feature vectors is suggested for text classification. For similarity computation between feature vectors, Jiang et.al [1] has proposed a fuzzy gaussian function which is applied to self-construct feature clusters. Another, important recent research contribution by Jiang et.al [2][3] is the gaussian text similarity measure for text classification. Similarity measure [2][3] proposed by Jiang takes into consideration, the effect of feature deviation on text features to best estimate, the similarity degree between text document vectors. The feature similarity function and text similarity functions designed in [1][2][3] are based on feature vectors that are non-binary. The feature similarity functions that are proposed in contributions [4][5][9] are based on binary representation of feature vectors. Another interesting similarity measure is the gaussian based temporal similarity measure proposed by Chen et.al [6][7] to uncover the similarity between temporal patterns on time-interval data. The text feature vector dimensionality problem is recently addressed by Suresh et.al [8] which is motivated from [1][2][3]. Motivated from the text similarity function [8], similarity functions for measuring software component similarity (which are based on determining binary feature vectors) are proposed by Vangipuram et.al [9]. Similarity measures to compute temporal similarity in Z-SPACE and gaussian space are proposed by Vangipuram et.al [10][11][12][13][14][15][16] and these measures require equivalent deviation and equivalent threshold values to be determined to compute similarity in new transformation space. Another contribution is the imputation measure MANTRA [17] suggested to find similarity between complete and incomplete medical records for medical data classification.

The efficacy of network intrusion anomaly detection algorithm banks on the use of an appropriate distance measures and similarity measures which are applied to compute the similarity of new incoming observations (not present in the training set) to the available observations in the trained knowledge base. A recent survey reported by Fahy et.al [18] proved that many research studies related to network intrusion anomaly detection have not documented the measures that are applied by machine learning algorithms in published research. Relatively less research literature is available on similarity measures applied in the research contributions that addressed the problem of network intrusion anomaly detection. A detailed study carried by Weller-Fahy et.al [18] provides us a complete overview of various similarity measures that are used within the field of NIAD (network intrusion anomaly detection) research. The fundamental idea behind network intrusion detection (or) the design of any NIDS (network intrusion detection system) is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2975716, IEEE Access

to automate the detection process whenever intrusions occur in the network. Thus, the problem of intrusion detection [18] may be viewed as a subproblem within NAD (network anomaly detection). Hence, the idea behind the network anomaly detection is to determine, if the network incoming traffic is legitimate (or) anomalous traffic. Automated detection systems that are designed to identify incoming anomalous traffic patterns usually apply widely used machine learning techniques such as supervised learning (or) un-supervised learning. However, irrespective of any system model which is developed to identify anomalous traffic, all these models require comparing anomalous and normal models [18][19]. Such comparisons implicitly depend on the ability of the underlying machine learning model to gauge the similarity between a known legitimate observation and the target. This means that efficiency of network anomalous detection system banks on the use of similarity measures and how they are actually applied.

An important contribution to NIDS research literature is the contribution by Aljawarneh et.al [19] in which a distance function is introduced to perform feature clustering. These feature clusters are used to achieve dimensionality reduction. Fuzzy membership functions are proposed by Gunupudi et.al [20][21] for feature clustering. These membership functions which are proposed by [20][21][22] are applied to obtain the similarity between feature pattern vectors for anomaly detection. An intrusion detection system, namely CANN proposed by Lin et.al [23], is the recent state of art contribution that combines the cluster center information with the nearest neighbor information to define a new distance which is one dimensional. Although CANN aims at addressing time efficiency and space efficiency, the accuracies of U2R and R2L attacks are not favorable. For example, using CANN [23] and choosing KNN classifier with k =1, attack accuracies of U2R and R2L classes on KDD dataset with 19 attributes are obtained as 3.85% and 57.02% . Also, from experiment analysis [23], the accuracies of KNN (K=1) for U2R and R2L classes on KDD dataset with 19 attributes are 17.31% and 91.74%. Similarly, for SVM classifier (degree 2), accuracies of U2R and R2L attack classes are 61.54% and 78.95% respectively. The overall accuracy obtained using CANN (K=1) is 99.46% and this value is slightly lesser than KNN (K=1) which is 99.89%. Thus, the challenge in design of new intrusion detection techniques, approaches and algorithms is to essentially aim at improving the accuracies of the low frequency classes such as U2R and R2L classes in KDD dataset. Another recent contribution that has proposed an approach for anomaly detection is UTTAMA [24]. UTTAMA applied a fuzzy membership function for similarity computation and feature transformation. The overall accuracy of UTTAMA on KDD dataset with 19 attributes is 99.89% when J48 classifier is applied for classification. When compared to CANN (K=1), UTTAMA (J48) has achieved better accuracies for low frequency attack classes. Aljawarneh et.al [25] applied feature selection on NSL-KDD dataset. An accuracy of 99.7% is reported on NSL-KDD dataset. A recent survey on intrusion detection techniques discussed various issues in designing an efficient intrusion system and some of the state of art contributions [18][26].

A machine learning approach, PAREEKSHA is proposed by Nagaraja et.al [27] for intrusion detection. The membership function has its basis from contribution [1][2]. On similar lines, [28] also proposed a membership function for detection of low frequency attacks. Network intrusion detection is a challenging task and it further becomes much more challenging for the machine learning algorithms when low frequency observations have to be detected with higher accuracies through overcoming challenges such as over fitting and under fitting. Many times, classifier algorithms employed to detect low frequency attacks do not perform well. This is because of the lesser number of instances in the dataset for those classes. Cross fold validation is usually applied to evaluate classifier performance and validation of machine learning models. Thus, improving classifier accuracies of low frequency classes is an important challenge that mandates an immediate attention from researchers. Conditional probability [1] can be used to derive hidden information and knowledge between features and dataset class labels. Such information may later be used to carry un-supervised learning [19][24].

The present research contribution is motivated from all the above discussed state of art contributions. It has been observed that, there is a scope for devising new similarity and distance functions that can be applied by detection systems to achieve better classification and detection rates. The next section describes the proposed method which is based on feature clustering for feature transformation.

III. PROPOSED METHOD

This section outlines the proposed method for the anomaly detection. Our approach extends the recent contribution by Vangipuram et.al [19] by proposing a new distance function which also considers feature distribution to determine the similarity between observations. Also, novel computation expressions to obtain the equivalent deviation and threshold values in gaussian space are proposed. The computed deviation value is used in similarity function to carry similarity computation. The basic idea is to represent dataset in new dimensionality space for improving classification and detection rates. Our method involves three tasks to be carried as outlined in [19]. They are (i) Feature clustering which is based on the use of the proposed gaussian based distance function (ii) Dimensionality reduction by feature reduction (iii) Applying the machine learning algorithm which uses the first two outcomes. Algorithms for these three tasks are outlined below.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2975716, IEEE Access

A. Feature clustering based on proposed gaussian distance function

Algorithm A1: feature clustering based on proposed Gaussian distance function

Input : User threshold, Observation matrix with m-dimensions
Output : Soft Clusters

δ^U	:	User threshold
δ^f	:	Transformation threshold
σ^f	:	Standard deviation in transformation space
m	:	Dimensionality
f_i	:	i[th] feature
f_{i_c}	:	Probability of i[th] feature w.r.t c[th] decision label
$\overrightarrow{f_l}$	:	m-dimensionality feature pattern, $\overrightarrow{f_l}=<f_{i_1}, f_{i_2}, \ldots\ldots f_{i_m}>$
$\mathcal{F}_{dist}$	:	distance function between two feature patterns $\overrightarrow{f_l}$ and $\overrightarrow{f_J}$
σ^0	:	Initial standard deviation, m-dimensional vector
$\sigma^{(g)}$	:	Deviation, $<\sigma^{g1}, \sigma^{g2}, \sigma^{g3}, \ldots\ldots\ldots, \sigma^{gm}>$
$m^{(g)}$	:	mean of g[th] cluster, $<m^{g1}, m^{g2}, m^{g3}, \ldots\ldots\ldots, m^{gm}>$
i	:	iterative index variable, varies from 1 to m
g	:	number of clusters, initially g = 0
C_g	:	g[th] cluster

Begin

1. Read the allowable dissimilarity value, δ^U
2. Determine the initial deviation value, σ^f and transformation threshold value, δ^f using Eq. (8) and Eq. (11) respectively.
3. Choose the first feature pattern (say $\overrightarrow{f_1}$). Initially, g = 0. Generate the first cluster by placing the first feature pattern, say, $\overrightarrow{f_1}$ in this cluster. Set g = g+1. Call it C_g. Now, C_g contains only $\overrightarrow{f_1}$.
4. Initialize mean and deviation of generated cluster (initially for the first cluster and then for other generated clusters).
   4.1 Mean of the first cluster is an m-dimensional vector and is same as the first feature pattern. i.e. $\overrightarrow{m^{(g)}}=\overrightarrow{f_1}$
   4.2 Initial standard deviation of the new generated cluster, $\overrightarrow{\sigma^{(g)}}=<\sigma^{(f)}, \sigma^{(f)}, \sigma^{(f)}, \ldots\ldots\ldots m\ times>$
5. If no other feature pattern exists then go to step-10 else go to step-6.
6. Choose the feature pattern that is not yet clustered, say $\overrightarrow{f_p}$. Determine the distance between this feature pattern ($\overrightarrow{f_p}$) and mean of each existing cluster ($\overrightarrow{m^{(g)}}$) with the proposed distance measure. i.e compute $\mathcal{F}_{dist}(\overrightarrow{f_p}, \overrightarrow{m^{(g)}})$.
7. If ($\mathcal{F}_{dist}(\overrightarrow{f_p}, \overrightarrow{m^{(g)}}) \leq \delta^f$)
   Add $\overrightarrow{f_p}$ to existing cluster and go to step-8
   else
   Set g= g+1. Create a new cluster. Call it C_g and repeat the process in step-4.
8. Update mean vector of the cluster after adding the feature pattern to the cluster. The new mean shall be the average of the existing feature pattern.
9. Go to step-5.
10. At the halt of incremental clustering, 'g' clusters and their respective mean vectors are generated.
11. Compute the respective standard deviation vector for each of these generated clusters by considering only those feature patterns that exist in respective clusters.
12. Update the deviation of final clusters. Now, the final deviation vector of each generated cluster shall be sum of the initial chosen deviation and respective deviation computed in step-11.

End

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2975716, IEEE Access

B. Algorithm for Dimensionality Reduction Based on Feature Transformation (A2)

Algorithm (A2): Dimensionality reduction by Feature transformation

Input : Observation matrix with decision class label
Output : Observation matrix with reduced dimensionality

Begin

1. Read the input observation matrix with 'o' instances, 'f' features and 'd' class labels. Represent the above information in the form of a matrix. Call the matrix as instance-feature matrix (also called as observation -feature matrix) denoted by $[O\text{-}F]_{|o|x|f|}$.
2. Obtain feature pattern vector for every feature using procedure outlined in sub-section D.
   2.1 Given a class label, say 'd', the conditional probability that the feature, 'f' could belong to the class, 'd' must be computed for each decision class. (computed using eq. 5)
   2.2 The set of all values computed in step 2.1 for a given feature w.r.t each class label is represented as a vector called as feature pattern (or feature pattern vector)
3. Apply incremental clustering (A1) algorithm with feature pattern vectors as input to obtain feature clusters (say 'g' clusters). The mean and deviation of individual clusters are the corresponding cluster representatives.
4. Compute the distance, i.e $\mathcal{F}_{dist}$ between each feature pattern to every generated cluster. Represent these feature-cluster similarities or dissimilarities in the form of a matrix called the soft feature-cluster dissimilarity matrix, $[F\text{-}C]_{fxg}$. Alternately, soft matrix can represent similarity values [1].
5. Transform the original observation-feature matrix by multiplying matrices $[O\text{-}F]_{oxf}$ and $[F\text{-}C]_{fxg}$. The result is a soft matrix (or soft observation-cluster matrix) denoted by $[O\text{-}C]_{oxg}$.
6. Matrix $[O\text{-}C]_{oxg}$ obtained in step-5 is transformed representation of input observation matrix. A high value of allowable dissimilarity yields minimal clusters. A low dissimilarity threshold chosen would generate more clusters.
7. Output the reduced dimensionality matrix, $[O\text{-}C]_{oxg}$. This matrix would be used as input dataset to build classifier model.

End

C. Algorithm for Anomaly Detection

Algorithm (A3): *Anomaly Detection*

Input : Proposed Classifier, features, observations
Output : Classification label

Begin

1. Read the dataset.
2. Set allowable threshold value to indicate minimum error such 0.0001, 0.01 etc.
3. Partition the dataset into training and testing groups via k-fold cross validation by setting k to 10.
4. Preprocess both the training and testing groups. Preprocessing of the training group is achieved carrying steps 4.1 to 4.5.
   4.1 For every feature in the dataset, generate the feature pattern vectors.
   i.e Given a decision class, then the probability that considered feature may belong to decision class is to be computed.
   4.2 Run evolutionary clustering (A1) for a chosen dissimilarity constraint.
   4.3 Run dimensionality reduction algorithm (A2).
   4.4 Determine soft (or hard) feature-cluster matrix [1][19].
   4.5 Derive the observation matrix for new transformation space from the soft or hard matrix generated in step 4.4.
   4.6 Store the resulting observation matrix obtained.
5. Repeat the step 4.5 and step 4.6 for testing set.
6. Apply learning algorithms (such as J48, KNN) by considering transformation observation matrix along with class label.
7. Evaluate performance of the machine learning algorithm by considering parameters such as accuracy, precision, detection, false rates.

End

D. Computation of the Feature pattern vector

Suppose that $[O\text{-}F]_{|o|x|f|}$ symbolizes the equivalent observation matrix representation of the KDD dataset where O symbolizes observation set, F symbolizes attribute set of the dataset and D symbolize the decision class set. In this paper, we use |o| to represent the total observations and |f| to symbolize the total attributes present in the dataset. Further, |d| is used to symbolize the total number of classes in the KDD dataset. In our case, there are five classes. So, |d|=5.

Consider the equations (1) to (3) which represents observation set, attribute set and decision class label set respectively.

$$O = \{O_1, O_2, O_3, \ldots\ldots\ldots\ldots\ldots, O_o\} \quad (1)$$

$$F = \{F_1, F_2, F_3, \ldots\ldots\ldots\ldots\ldots\ldots, F_f\} \quad (2)$$

$$D = \{D_1, D_2, D_3, \ldots\ldots\ldots\ldots\ldots\ldots, D_d\} \quad (3)$$

Let F_i symbolize i[th] feature in the feature set, F and f_{io} symbolize, the value of feature f_i in o[th] observation. The representation $\overrightarrow{X_i}$ symbolizes feature pattern vector corresponding to any feature, F_i. Our approach requires computing feature pattern vector for every feature; F_i present in the feature set, F. As mentioned already, |d| symbolizes dimensionality of the feature pattern vector. We represent the feature pattern vector using equation (4) where X_{id} symbolizes the probability that feature, F_i belong to the class, D_d .

$$\overrightarrow{X_i} = (X_{i1}, X_{i2}, X_{i3}, \ldots\ldots\ldots\ldots\ldots, X_{id}) \quad (4)$$

The element value X_{id} in equation (4) can be obtained by applying equation (5)

$$X_{id} = \frac{\sum_{i=1}^{i=|o|} f_{ji} * \mathcal{M}_d^j}{\sum_{i=1}^{i=|o|} f_{ji}} \quad (5)$$

In equation (5), f_{ji} symbolizes j[th] feature value in the i[th] observation of the observation matrix. The value of $\mathcal{M}_d^j$ is 1 ; if the j[th] feature symbolized using F_j belongs to class label, D_d and $\mathcal{M}_d^j$ is equal to 0 ; if F_j do not belong to class label, D_d.

The next subsection gives the proposed distance function to find the similarity between any two feature conditional probability vectors.

E. PROPOSED DISTANCE FUNCTION

This subsection gives the computation expression of the proposed feature distance function that can be applied to determine the similarity between any two feature pattern vectors and input observations. The similarity condition for considering two feature pattern vectors as similar is stated below.

Similarity Condition: Given $\overrightarrow{\mathbf{X_P}}$ and $\overrightarrow{\mathbf{X_q}}$ are the two conditional probability vectors, $\overrightarrow{\mathbf{X_P}}$ and $\overrightarrow{\mathbf{X_q}}$ are similar, if and only if, the distance obtained using the distance function $\boldsymbol{\mathcal{F}_{dist}}(\overrightarrow{\mathbf{X_P}}, \overrightarrow{\mathbf{X_q}})$ satisfies the condition $\boldsymbol{\mathcal{F}_{dist}}(\overrightarrow{\mathbf{X_P}}, \overrightarrow{\mathbf{X_q}}) \leq \boldsymbol{\delta^U}$.

1) Proposed distance function

Suppose, $\overrightarrow{\boldsymbol{X_P}}$ and $\overrightarrow{\boldsymbol{X_q}}$ are any two conditional probability vectors (i.e feature pattern vectors) and let the notation $\boldsymbol{\delta^U}$ symbolize the distance threshold. Let $\boldsymbol{m}$ be the dimensionality of the probability vector. Now, $\overrightarrow{\boldsymbol{X_P}}$ and $\overrightarrow{\boldsymbol{X_q}}$ can be represented as $\overrightarrow{\boldsymbol{X_P}}$ = ($\boldsymbol{X_{P_1}}$, $\boldsymbol{X_{P_2}}$, $\boldsymbol{X_{P_3}}$, ……., $\boldsymbol{X_{P_m}}$) and $\overrightarrow{\boldsymbol{X_q}}$ = ($\boldsymbol{X_{q_1}}$, $\boldsymbol{X_{q_2}}$, $\boldsymbol{X_{q_3}}$, ……., $\boldsymbol{X_{q_m}}$). The element values of the form $\boldsymbol{X_{P_i}}$ and $\boldsymbol{X_{q_i}}$ in the probability vector represented by $\overrightarrow{\boldsymbol{X_P}}$ and $\overrightarrow{\boldsymbol{X_q}}$ is the posterior probability value such that $\boldsymbol{X_{P_i}} \in \{0,1\}$.

The distance between any two conditional probability vectors symbolized using $\overrightarrow{X_P}$ and $\overrightarrow{X_q}$ can be obtained by using the proposed distance function which is given by using equation (6), with $\alpha = 0.3679$.

$$\mathcal{F}_{dist}(\overrightarrow{X_P}, \overrightarrow{X_q}) = \frac{1 - \mu(\overrightarrow{X_P}, \overrightarrow{X_q})}{1 + \alpha} \quad (6)$$

where

$$\mu(\overrightarrow{X_P}, \overrightarrow{X_q}) = \frac{\sum_{i=1}^{i=m} exp^{-\left(\frac{X_{p_i} - X_{q_i}}{\sigma^f}\right)^2}}{\sum_{i=1}^{i=m} 1} \quad (7)$$

Eq.(7) represents the average fuzzy similarity value between $\overrightarrow{X_P}$, $\overrightarrow{X_q}$. The parameter 'σ^f' used in eq.(7) is the standard deviation value which can be obtained by applying Eq.(8).

The expression for computing deviation is given by Eq.(8),

$$\sigma^f = \frac{\delta^U}{\sqrt{ln_e\left(\frac{1}{abs(1 - (1+\alpha) * \delta^U)}\right)}} \quad (8)$$

where δ^U is the allowable dissimilarity chosen between 0 and 1 and $\alpha = 0.3679$.

2) Expression for gaussian distance threshold

We know that δ^U represents the distance threshold between vectors, $\overrightarrow{X_P}$ and $\overrightarrow{X_q}$ in euclidean space. Our approach requires computing the new deviation value for the gaussian space. The deviation for new tarnsformation space can be derived by considering single dimension vectors. In this case, for any given dimension (say, i[th] dimension), the distance between vectors X_{p_i} and X_{q_i} is given by Eq.(9)

$$\boldsymbol{\delta^U} = \sqrt{\left(\boldsymbol{X_{p_i}} - \boldsymbol{X_{q_i}}\right)^2} = |\boldsymbol{X_{p_i}} - \boldsymbol{X_{q_i}}| \quad (9)$$

Now, the distance between $\overrightarrow{X_P}$ and $\overrightarrow{X_q}$ using proposed distance function is given by Eq.(10)

$$\mathcal{F}_{dist}(\overrightarrow{X_P}, \overrightarrow{X_q}) = \frac{1 - e^{-\left(\frac{|X_{p_i} - X_{q_i}|}{\sigma^f}\right)^2}}{1 + \alpha} \quad (10)$$

Using Eqs.(9) and (10), the distance threshold for new transformation space is given by Eq.(11)

$$\delta^f = \frac{1 - e^{-\left(\frac{\delta^U}{\sigma^f}\right)^2}}{1 + \alpha} \quad (11)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2975716, IEEE Access

In Eq.(10) and Eq. (11), α is 0.3679.

F. DERIVATION OF PROPOSED FEATURE PATTERN SIMILARITY FUNCTION

Consider the two conditional probability feature pattern vectors $\boldsymbol{X_{P_i}}$ and $\boldsymbol{X_{q_i}}$ given by $\overrightarrow{\boldsymbol{X_P}}$ = ($\boldsymbol{X_{P_1}}$, $\boldsymbol{X_{P_2}}$, $\boldsymbol{X_{P_3}}$, ……., $\boldsymbol{X_{P_m}}$) and $\overrightarrow{\boldsymbol{X_q}}$ = ($\boldsymbol{X_{q_1}}$, $\boldsymbol{X_{q_2}}$, $\boldsymbol{X_{q_3}}$, ……., $\boldsymbol{X_{q_m}}$). The element values of the form $\boldsymbol{X_{P_i}}$ and $\boldsymbol{X_{q_i}}$ in the probability vector represented by $\overrightarrow{\boldsymbol{X_P}}$ and $\overrightarrow{\boldsymbol{X_q}}$ is the posterior probability value such that $\boldsymbol{X_{P_i}} \in \{0,1\}$.

The membership value of $\overrightarrow{\boldsymbol{X_P}}$ to $\overrightarrow{\boldsymbol{X_q}}$ for i[th] feature dimension, i.e $\overrightarrow{\boldsymbol{X_P}}$ = ($\boldsymbol{X_{P_i}}$) and $\overrightarrow{\boldsymbol{X_q}}$ = ($\boldsymbol{X_{q_i}}$) can be obtained by applying the basic gaussian membership function as given by equation (12)

$$\boldsymbol{\mu_i}\left(\overrightarrow{\boldsymbol{X_{pi}}},\ \overrightarrow{\boldsymbol{X_{qi}}}\right) = \boldsymbol{exp}^{-\left(\frac{X_{p_i}-X_{q_i}}{\sigma^i}\right)^2} \qquad (\mathbf{12})$$

The normalized membership value of feature pattern vector $\overrightarrow{\boldsymbol{X_P}}$ to $\overrightarrow{\boldsymbol{X_q}}$ by considering all the 'm' dimensions may be obtained using equation (13)

$$\mu\left(\overrightarrow{X_P},\ \overrightarrow{X_q}\right) = \frac{\sum_{i=1}^{i=m} \mu_i\left(\overrightarrow{X_{pi}},\ \overrightarrow{X_{qi}}\right)}{\sum_{i=1}^{i=m} 1} \qquad (13)$$

Substituting expression for $\mu_i\left(\overrightarrow{X_{pi}},\ \overrightarrow{X_{qi}}\right)$ represented by Eq.(12) in expression for normalized membership value represented by Eq.(13), we have the resulting expression for normalized membership value (also called as average membership value) given by Eq.(14)

$$\mu\left(\overrightarrow{X_P},\ \overrightarrow{X_q}\right) = \frac{\sum_{i=1}^{i=m} exp^{-\left(\frac{X_{p_i}-X_{q_i}}{\sigma^i}\right)^2}}{\sum_{i=1}^{i=m} 1} \qquad (14)$$

However, Eq.(14) cannot be considered as the similarity value as it defines the average membership value (or) average similarity between pattern vectors. So, similarity must be defined by some other function. To achieve this, we define the similarity function given by Eq.(15) to compute the similarity between $\overrightarrow{X_P}$ and $\overrightarrow{X_q}$ as

$$\mathcal{F}_{sim}\left(\overrightarrow{X_P}, \overrightarrow{X_q}\right) = \frac{\mu\left(\overrightarrow{X_P},\ \overrightarrow{X_q}\right) + \alpha}{1 + \alpha} \qquad (15)$$

Where α is any constant.

The value of α can be obtained by performing analytical analysis through analyzing for lowest and highest possible similarity values. Consider two cases to define the value of α namely i) worst case and ii) best case.

1) Best case

In the best case, the similarity between X_{P_i} and X_{q_i} is unity $\forall\, i : 1\ to\ m$. i.e $|X_{P_i} - X_{q_i}| = 0$. In this case, $\mu\left(\overrightarrow{X_P},\ \overrightarrow{X_q}\right)$ is computed as

$$\mu\left(\overrightarrow{X_P},\ \overrightarrow{X_q}\right) = \frac{1+1+1+\cdots .m\ times}{1+1+1+\cdots m\ times} = \frac{m}{m} = 1 \qquad (16)$$

So, the similarity between $\overrightarrow{X_P}$ and $\overrightarrow{X_q}$ is given by

$$\mathcal{F}_{sim}\left(\overrightarrow{X_P}, \overrightarrow{X_q}\right) = \frac{1 + \alpha}{1 + \alpha} = 1 \qquad (17)$$

2) Worst case

In the worst case, the similarity between X_{P_i} and X_{q_i} is exactly (or) almost equal to a zero, $\forall\, i : 1\ to\ m$. The distance is hence equal to unity (which is the maximum). i.e $|X_{p_i} - X_{q_i}| = 1$. The membership value of each X_{P_i} to $X_{q_i}$is given by $\mu_i\left(\overrightarrow{X_{pi}},\ \overrightarrow{X_{qi}}\right) = exp^{-\left(\frac{1}{\sigma^i}\right)^2}$. The maximum value of σ^i is unity. So, the value of $\mu\left(\overrightarrow{X_P},\ \overrightarrow{X_q}\right)$ is computed as $\mu_i\left(\overrightarrow{X_{pi}},\ \overrightarrow{X_{qi}}\right) = exp^{-\left(\frac{1}{1}\right)^2} = 0.3679$. From Eq.(15), the similarity between $\overrightarrow{X_P}$ and $\overrightarrow{X_q}$ is given by

$$\mathcal{F}_{sim}\left(\overrightarrow{X_P}, \overrightarrow{X_q}\right) = \frac{0.3679 + \alpha}{1 + \alpha} \qquad (18)$$

Since, the similarity in worst case is zero. This means that $0.3679 + \alpha = 0$. This gives α = -0.3679. Now consider the expression for similarity given by Eq.(15). Substituting the value of α = -0.3679 in Eq.(15), we have

$$\mathcal{F}_{sim}\left(\overrightarrow{X_P}, \overrightarrow{X_q}\right) = \frac{\mu\left(\overrightarrow{X_P},\ \overrightarrow{X_q}\right) - 0.3679}{0.6321} \qquad (19)$$

Rationalizing Numerator and Denominator of Eq. (19), We finally get

$$\mathcal{F}_{sim}\left(\overrightarrow{X_P}, \overrightarrow{X_q}\right) = \frac{\mu\left(\overrightarrow{X_P},\ \overrightarrow{X_q}\right) + 0.3679}{1 + 0.3679} \qquad (20)$$

Eq.(20), may be re-written as Eq.(21) Where α = 0.3679.

$$\mathcal{F}_{sim}\left(\overrightarrow{X_P}, \overrightarrow{X_q}\right) = \frac{\mu\left(\overrightarrow{X_P},\ \overrightarrow{X_q}\right) + \alpha}{1 + \alpha} \qquad (21)$$

We have $\mathcal{F}_{dist}\left(\overrightarrow{X_P}, \overrightarrow{X_q}\right) + \mathcal{F}_{sim}\left(\overrightarrow{X_P}, \overrightarrow{X_q}\right) = 1$. Using this relation, Eq.(22) gives the computation expression for the distance computation between $\overrightarrow{X_P}$ and $\overrightarrow{X_q}$

$$\mathcal{F}_{dist}\left(\overrightarrow{X_P}, \overrightarrow{X_q}\right) = \frac{1 - \mu\left(\overrightarrow{X_P},\ \overrightarrow{X_q}\right)}{1 + \alpha} \qquad (22)$$

Hence proved.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2975716, IEEE Access

V. EXPERIMENTAL EVALUATION

All the experiments discussed in this section are conducted on DELL INSPIRON 15 5000 series having 32 GB RAM with Intel CORE i5 7th generation CPU. For experimental analysis of the proposed machine learning method, we have considered the two widely used benchmark datasets. They are (i) KDD dataset which consist 41 and 19 attributes and (ii) NSL-KDD dataset which consist 41 attributes.

Feature transformation is one of the most important preprocessing techniques which can improve the classifiers overall performance [19][20]. By feature transformation technique, we mean that the attributes (or) features of the input dataset are projected on to another dimensionality space. The proposed feature transformation approach is based on generating feature clusters by considering the attribute belongingness to various class labels of the input dataset [1][19]. The generated clusters using the proposed feature transformation technique represents the dimensionality of the transformed input dataset. For example, in our approach, we first cluster the attributes of the dataset into a finite number of clusters. From these clusters, their representative features such as mean and deviation are obtained. Using these representative elements of clusters, a matrix called as soft transformation matrix is obtained. The soft transformation matrix gives the similarity of each feature to each of these clusters. This soft matrix is used to obtain dimensionality reduced input dataset (or) a matrix that is an equivalent representation of the original matrix. It must be noted that in the original form, each observation is a function of attributes whereas in the transformed representation, each observation is expressed in terms of feature clusters.

The transformed dataset is then applied as input for various classifiers such as i) Naïve bayes classifier, ii) BayesNet classifier, iii) SMO classifier, iv) J48 decision tree classifier and v) KNN (k-Nearest Neighbors) classifier by choosing k-fold cross validation resampling technique for evaluating performance of the machine learning model. The evaluation parameters considered for performance evaluation are a) Accuracy b) Precision c) Recall d) Correctly classified instances and also the overall weighted accuracy and precision. Subsection-A gives the experiment results obtained by considering the KDD dataset with 41 attributes.

A. KDD-Cup 99 Dataset with 41 attributes

In this subsection, we discuss the experiment results obtained by considering the equivalent dimensionality reduced input dataset which is the result of carrying feature transformation on the KDD-Cup dataset with 494021 observation instances defined over a feature set consisting 41 attributes. For all experiments, the similarity threshold is set to 0.9995 and initial deviation is set to 0.5. To evaluate the performance of the model, k-fold cross validation resampling technique is used by setting k value equal to 10. The result of feature transformation is 35 clusters. This means that each of the observations in the input dataset is now represented in terms of these 35 clusters. Experiments are conducted by considering classifiers such as i) Naïve bayes classifier, ii) BayesNet classifier, iii) SMO classifier, iv) J48 decision tree classifier and v) KNN (k-Nearest Neighbors) classifier. Figure 1 shows the J48 classifier confusion matrix which is obtained by considering the resulting dataset obtained after feature transformation. The classifier accuracies (in percentage) obtained is 99.97% for normal class and 99.99% for U2R, DoS, R2L and Probe classes. The percentage of correctly classified instances with J48 classifier is 99.967%.

	Normal	U2R	DoS	R2L	Probe	Accuracy
Normal	97231	7	9	12	19	0.9997
U2R	20	26	0	5	1	0.9999
DoS	12	0	391443	1	2	0.9999
R2L	41	5	0	1079	1	0.9999
Probe	25	0	1	1	4080	0.9999
Correctly classified instances						99.9672

FIGURE 1. Confusion matrix obtained for KDD dataset with proposed feature transformation approach and using J48 classifier .

A simple analysis of the confusion matrix shows that the percentage precision values for Normal class and attack classes (U2R, Dos, R2L, Probe) are 99.89%, 68.42%, 99.997%, 98.27%, 99.44% respectively. Similarly, the respective recall values of Normal, U2R, Dos, R2L, Probe classes are 99.95%, 50%, 99.99%, 95.82%, 99.34%. The accuracy, precision and recall values obtained for J48 classifier using the transformed input dataset representation shows that importance of the proposed approach.

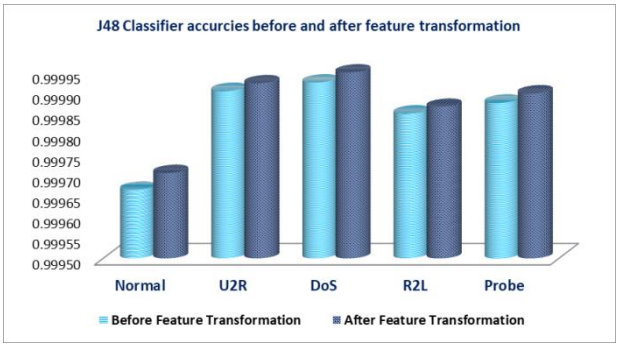

FIGURE 2. J48 Classifier accuracies before and after feature transformation using proposed approach and distance function.

Figure 2 depicts J48 classifier accuracies before and after proposed feature transformation technique. It is visible from Figure 2 that the accuracies of KDD classes (i.e Normal, U2R, DoS, R2L, and Probe) obtained by considering the transformed dataset after feature transformation have improved when compared to the accuracies obtained by using original dataset with 41 attributes (i.e without feature transformation). After feature transformation technique, the dimensionality is reduced to 35 attributes. This proves the importance of the proposed feature transformation technique. Thus, it is inferred from the above experiment that feature

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2975716, IEEE Access

transformation has shown improvement in accuracies of each class of KDD dataset.

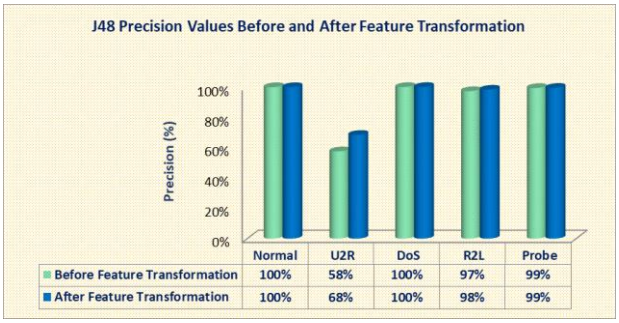

FIGURE 3. J48 Classifier Precision Values before and after feature transformation using proposed approach.

Figure 3 depicts the J48 classifier precision values before and after proposed feature transformation technique. It is visible from Figure 3 that the precision values of Normal, Dos and Probe classes before and after feature transformation are same. However, after feature transformation technique is applied, there is considerable improvement w.r.t U2R and R2L attack classes. The precision value for U2R attack class is improved from 58% to 68% and R2L attack class is improved from 97% to 98%. Thus, this experiment once gain infers the importance of the proposed approach.

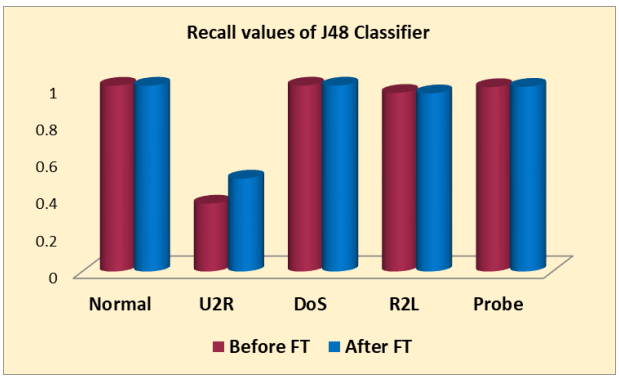

FIGURE 4. J48 Classifier Recall values before and after feature transformation using proposed approach for various classes.

Figure 4 depicts the J48 classifier percentage Recall values before and after proposed feature transformation technique. It is visible from Figure 4 that the precision values of Normal, Dos and Probe classes before and after feature transformation are same. However, after feature transformation technique is applied, there is a considerable improvement in four classes of KDD dataset except for R2L attack class. The recall value for R2L attack class is 95.8% after feature transformation whereas it is 96% before feature transformation is applied. For U2R attack class, the recall is increased from 36.53% to 50%. Hence, it is inferred that using proposed approach the accuracy, precision and recall values have all been better when compared to the values obtained on the KDD dataset without feature transformation for J48 Classifier.

	Normal	U2R	DoS	R2L	Probe	Accuracy
Normal	96992	5	49	134	98	0.9976
U2R	11	26	2	13	0	0.9999
DoS	272	0	391177	5	4	0.9993
R2L	170	2	9	945	0	0.9993
Probe	466	0	12	1	3628	0.9988
Correctly classified instances						0.99746

FIGURE 5. Confusion matrix obtained for KDD dataset with proposed feature transformation approach and using KNN (K=1) classifier.

Figure 5 shows the KNN (K=1) classifier confusion matrix which is obtained by considering the resulting dataset obtained after feature transformation. The percentage classifier accuracies obtained are 99.76% for normal class and 99.99% for U2R, 99.93% for DoS, R2L and 99.88% for Probe attack class. The percentage of correctly classified instances with J48 classifier is 99.75% for KNN with K=1 which is slightly lesser than J48 classifier results. Using proposed method, the precision value of U2R is improved from 68.08% to 78.78%. However, the U2R class accuracy remained same whether (or) not the feature transformation is applied. For other classes, there is no much difference in terms of classifier accuracies. It is also observed from the experiments that the recall value of low frequency classes U2R and R2L have slightly decreased for KNN even after feature transformation is applied. However, the overall performance of classifier in terms of correctly classified instances is nearly the same.

	Normal	U2R	DoS	R2L	Probe	Accuracy
Normal	96126	174	2	527	449	0.9962
U2R	8	43	0	0	1	0.9956
DoS	638	1986	387323	21	1490	0.9916
R2L	12	21	0	1091	2	0.9988
Probe	44	8	2	1	4052	0.9960
Correctly classified instances						0.9891

FIGURE 6. Confusion matrix obtained for KDD dataset with proposed feature transformation approach and using BayesNet classifier .

Figure 6 and Figure 7 respectively shows the Bayesnet classifier and Naïve Bayes confusion matrices which are obtained by considering the resulting KDD dataset (i.e 494021 instances with 35 dimensionality) obtained after feature transformation.

	Normal	U2R	DoS	R2L	Probe	Accuracy
Normal	74057	2982	9814	539	9886	0.947
U2R	6	42	1	2	1	0.992
DoS	2550	47	368158	9	20694	0.933
R2L	53	639	4	369	61	0.997
Probe	176	227	66	0	3638	0.937
Correctly classified instances						0.903

FIGURE 7. Confusion matrix obtained for KDD dataset with proposed feature transformation approach and using Naïve Bayes classifier.

Figure 8 shows the accuracy (ACC), precision (PREC) and recall (RECALL) values recorded from experiments for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2975716, IEEE Access

Bayesnet classifier before and after applying proposed feature transformation technique. From the experiment values, it is observed that the accuracy, precision and recall values are improved for Normal and Probe classes when feature transformation technique is applied. Similarly, improvement in accuracy and precision values for U2R attack class, accuracy and precision values for DoS class are seen with proposed approach. The precision value of the Bayesnet classifier has shown improvement in terms of R2L class. For all other cases, though there is no improvement in values of precision, recall and accuracy, but these values remained same both before and after feature transformation. Thus, it can be deduced that the Bayesnet classifier has seen improvement in terms of overall classifier performance. In general, it is observed from experiments conducted that classifiers performance achieved by applying the proposed feature transformation technique has been better when compared to performance achieved without feature transformation.

Before Feature Transformation - 41 dimensions			
KDD Classes	ACC	PREC	RECALL
Normal	0.9953	0.99243	0.983562573
U2R	0.9954	0.01855	0.826923077
DoS	0.9912	0.99999	0.988915797
R2L	0.9988	0.65644	0.968916519
Probe	0.9949	0.62246	0.986364743

After Feature Transformation - 35 dimensions			
KDD Classes	ACC	PREC	RECALL
Normal	**0.9962**	**0.99275**	**0.988157651**
U2R	**0.9956**	**0.01927**	0.826923077
DoS	**0.9916**	0.99999	**0.989436926**
R2L	0.9988	**0.66524**	0.968916519
Probe	**0.9960**	**0.67601**	**0.98660823**

FIGURE 8. Accuracy, Precision, Recall values for BayesNet Classifier with and without feature transformation technique.

This subsection discussed the classifiers performance before and after feature transformation by considering KDD dataset with 494021 observations with 41 attributes. The next subsections compares proposed approach to other recent approaches.

B. COMPARISON WITH UTTAMA and GARUDA

For all experiments discussed in this section, the similarity threshold is set to 0.9995 and initial deviation is set to 0.5 and 10-fold cross validation is considered to evaluate the model performance. Experiments are conducted to evaluate performance of proposed approach to UTTAMA [24] and GARUDA [19]. UTTAMA [24] proposed by Arun et.al is an evolutionary feature clustering approach for network intrusion anomaly detection which uses fuzzy membership function for similarity computations. It is motivated from contributions [1] [19] [20]. The performance of proposed approach is compared to UTTAMA by considering various classifier evaluation parameters such as precision, recall, and correctly classified instances.

Figure 9 shows the plot of percentage of correctly classified instances with the proposed approach and UTTAMA for KDD dataset with 494021 observations and 41 attributes. The overall accuracy of UTTAMA and proposed approaches are 99.982% and 99.99% respectively while the percentage of correctly classified instances is 99.952% and 99.97% for UTTAMA and proposed methods respectively. This proves that proposed method is better to UTTAMA.

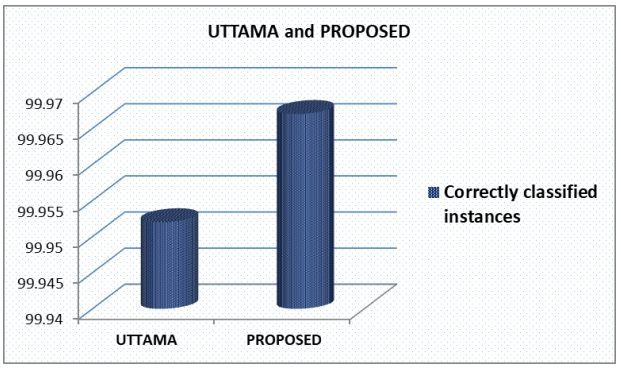

FIGURE 9 Percentage of Correctly classified instances for UTTAMA and PROPOSED methods

Figure 10 depicts the plot of weighted and class wise accuracies obtained using proposed approach and UTTAMA [24] for both the normal and attack classes of KDD dataset. It is observed that accuracies of both normal and attack classes using the proposed method are better when compared to UTTAMA. Experiment results obtained using the proposed approach for various classes are as follows: 99.97% for Normal, 99.99 % for U2R, 99.99% for DoS, 99.99% for R2L, and 99.99% for Probe which is comparatively very much better to UTTAMA. It is visible that U2R and R2L accuracies (i.e low frequency attack classes) are efficiently identified using proposed approach when compared to UTTAMA.

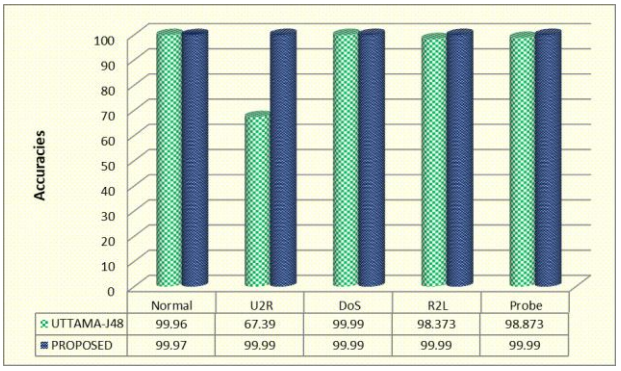

FIGURE 10 Percentage of Correctly classified instances for UTTAMA and PROPOSED methods

Figure 11 gives the plot of accuracies obtained using proposed approach and GARUDA for each class. In [19], a feature clustering technique for reducing the dimensionality

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2975716, IEEE Access

of the dataset is proposed which uses a distance function GARUDA. Here, we propose to use the proposed distance function for feature transformation instead of GARUDA. From experiments conducted, it is observed that a considerable improvement in terms of overall accuracy is recorded for Bayesnet, NaiveBayes and SMO classifiers. For J48 classifier with GARUDA, the accuracy is 99.82% whereas for the proposed approach, the accuracy is obtained as 99.97%. For KNN classifier with proposed approach, it is observed that the accuracy is 99.89% whereas for GARUDA with KNN, it is marginally higher value (99.91%). Overall the proposed approach has improved accuracies of classifiers when compared to feature transformation technique with distance function proposed in [19].

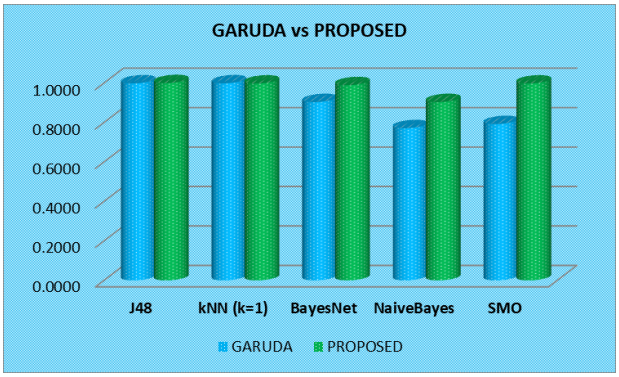

FIGURE 11. Accuracies for Feature transformation with GARUDA and PROPOSED distance function

An interesting observation is that when proposed distance function is used for feature clustering and feature transformation, for J48 classifier, the accuracies of U2R and R2L attack classes are 99.99%, 99.98% for UTTAMA and 99.99%, 99.99% for proposed method. However, considering the precision value for these two attack classes, it is observed that the precision values of U2R and R2L attack classes are 78.94%, 96.43% for UTTAMA whereas for the proposed approach, we have obtained a precision of 68.42%, 98.27% for proposed method. From overall perspective, the performance of the proposed approach seems better when compared to UTTAMA.

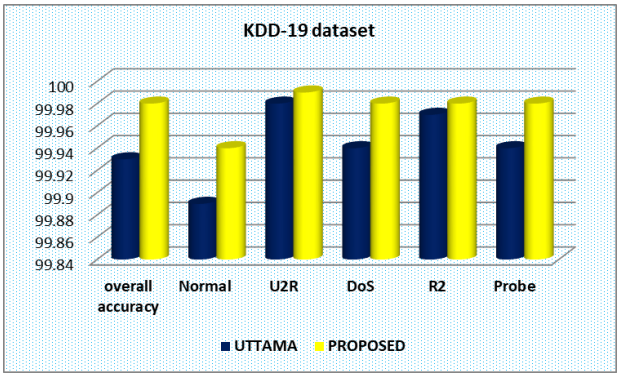

FIGURE 12. UTTAMA vs PROPOSED on KDD-19

Experiments are also conducted by considering KDD dataset with 19 attributes [19][24]. Figure 12 shows the plot of overall and Classwise accuracies for all five classes of KDD dataset by considering UTTAMA and proposed approaches. It is observed that for KDD dataset [23] with 19 attributes the accuracies of all classes and overall classifier accuracy using proposed approach have seen improvement. The overall accuracy of UTTAMA using J48 classifier is 99.89% whereas using our proposed approach it is 99.94%. Thus, the proposed approach has also achieved better accuracies on KDD dataset with 19 attributes.

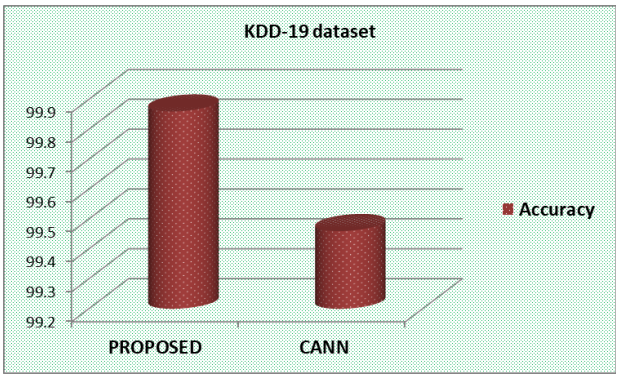

FIGURE 13. CANN [23] vs PROPOSED on KDD-19

Finally, the overall accuracy obtained using proposed approach and CANN [23] on KDD-19 dataset are compared. Using CANN approach for K=1[23], the overall accuracy achieved is 99.46% whereas the accuracy is 99.86% using our approach with K=1. Thus, it can be deduced that the classifier accuracy of proposed approach is improved when compared to CANN. All these experiment results prove that the proposed approach for network intrusion and anomaly detection is better when compared to intrusion detection approaches GARUDA, CANN, UTTAMA.

C. NSL-KDD Dataset with 41 attributes

Experiments are also conducted by considering NSL-KDD dataset with 41 attributes. For experiments discussed in this section, the similarity threshold is set to 0.9995 and initial deviation is set to 0.5 and 10-fold cross validation is considered to evaluate the model performance. The dimensionality of the dataset is 36 after feature transformation. Figure 14 and Figure 15 shows the confusion matrix obtained for J48 and KNN (K=1) classifier after performing feature transformation using proposed approach. The Classwise accuracy for each class is also shown in the last column of confusion matrix. From the confusion matrices of J48 and KNN shown in Figure 14 and Figure 15, it can be observed that the classifier accuracies for U2R and R2L attack classes are very much better.

For instance, using J48 classifier, the accuracy for U2R and R2L classes are obtained as 99.97% and 99.92% and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2975716, IEEE Access

corresponding U2R and R2L accuracy values for KNN classifier are 99.98% and 99.87% respectively. The precision, recall values for J48 and KNN classifiers are depicted in Figure 16 and Figure 17 respectively.

	Normal	DoS	R2L	Probe	U2R	Accuracy
Normal	67123	27	40	145	8	0.9960
DoS	22	45828	0	77	0	0.9981
R2L	60	0	929	3	3	0.9992
Probe	177	117	0	11362	0	0.9959
U2R	27	0	0	0	25	0.9997
Correctly classified instances						0.9944

FIGURE 14. J48 Classifier confusion matrix for NSL-KDD 41

	Normal	DoS	R2L	Probe	U2R	Accuracy
Normal	66951	66	90	226	10	0.9936
DoS	56	45840	0	31	0	0.9977
R2L	65	6	915	6	3	0.9987
Probe	285	133	0	11237	1	0.9946
U2R	14	0	0	1	37	0.9998
Correctly classified instances						0.99212

FIGURE 15. KNN (K=1) Confusion matrix for NSL-KDD 41

	ACC	PREC	RECALL
Normal	0.99598	0.99576	0.996733
DoS	0.99807	0.99687	0.997844
R2L	0.99916	0.95872	0.933668
Probe	0.99588	0.98058	0.974777
U2R	0.99970	0.69444	0.480769

FIGURE 16. Accuracy, Precision and Recall values for J48 classifier using proposed feature transformation

	ACC	PREC	RECALL
Normal	0.99355	0.99377	0.994179
DoS	0.99768	0.99555	0.998106
R2L	0.99865	0.91045	0.919598
Probe	0.99458	0.97705	0.964053
U2R	0.99977	0.72549	0.711538

FIGURE 17. Accuracy, Precision and Recall values for KNN classifier using proposed feature transformation

The F-score values can be computed from precision and recall values depicted in Figure 16 and Figure 17 for J48 and KNN classifiers. In our case, for J48 classifier, the F-Score for Normal, DoS, R2L, Probe and U2R classes is 0.9962, 0.9973, 0.9460, 0.9776, and 0.5682. Similarly, for KNN classifier, the F-Score values for Normal, DoS, R2L, Probe and U2R classes is 0.9939, 0.9968, 0.915, 0.9705 and 0.7184. Also, the ROC values obtained for J48 classifier for Normal, DoS, R2L, Probe and U2R classes are 0.997, 0.999, 0.98, 0.992 and 0.938. The respective ROC values for KNN classifier are 0.994, 0.998, 0.958, 0.981 and 0.841. The F-Score and ROC values for J48 and KNN classifiers prove the importance of the proposed approach.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have applied the proposed distance function for carrying feature clustering and to achieve feature transformation. Thus, dimensionality reduction is carried via feature transformation. The distance function proposed in this work is designed by considering the basic gaussian membership function. After achieving dimensionality reduction using proposed feature extraction technique, we have applied classifier algorithms for evaluating performance of the classifiers on the transformation datasets. Several experiments are conducted on KDD dataset with 41 and 19 attributes and the performance of classifiers is evaluated. Experiment analysis proved that the performance of the proposed approach is comparatively very much better and has achieved an improved performance interms of accuracy, precision and recall parameters. One of the significant findings and important outcomes of the proposed approach which is derived from the experiment results is that the accuracy and precision values of low frequency attack classes have substantially improved. This work is limited to proposing a new distance function and applying the proposed distance function for feature clustering and transformation so as to prove the importance of distance functions in machine learning model and also to show how a comparatively better performance may be achieved by classifiers, if an appropriate distance function is employed. Experiments are performed on KDD dataset with 41 and 19 attributes and NSL-KDD dataset with 41 attributes by considering several classifier algorithms. Classifier performance is evaluated in terms of accuracy, precision, recall and F-Score parameters. Experiment results and analysis proved that our approach for anomaly detection using proposed feature transformation technique proved to be better when compared to other detection methods that are addressed in the literature. As a future extension of the present work, we are currently studying the possibility of designing new decision tree based classifiers.

REFERENCES

[1] J. Jiang, R. Liou and S. Lee, "A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 3, pp. 335-349, March 2011. doi: 10.1109/TKDE.2010.122

[2] J. Jiang, W. Cheng, Y. Chiou and S. Lee, "A similarity measure for text processing," *2011 International Conference on Machine Learning and Cybernetics*, Guilin, 2011, pp. 1460-1465. doi: 10.1109/ICMLC.2011.6016998

[3] Y. Lin, J. Jiang and S. Lee, "A Similarity Measure for Text Classification and Clustering," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1575-1590, July 2014. doi: 10.1109/TKDE.2013.19

[4] G. SureshReddy, T. V. Rajinikanth, and A. Ananda Rao. 2014. Design and analysis of novel similarity measure for clustering and classification of high dimensional text documents. In Proceedings of the 15th International Conference on Computer Systems and Technologies (CompSysTech '14), Boris Rachev and Angel Smrikarov (Eds.). ACM, New York, NY, USA, 194-201. DOI=http://dx.doi.org/10.1145/2659532.2659615

[5] Chintakindi Srinivas, C. V. Guru Rao, and V. Radhakrishna. 2018. Feature Vector Based Component Clustering for Software Reuse. In Proceedings of the Fourth International Conference on Engineering & MIS 2018 (ICEMIS '18). ACM, New York, NY, USA, Article 39, 6 pages. DOI: https://doi.org/10.1145/3234698.3234737

[6] Y. C. Chen, W. C. Peng and S. Y. Lee, "Mining temporal patterns in interval-based data," *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, Helsinki, 2016, pp. 1506-1507.

[7] Y. C. Chen, W. C. Peng and S. Y. Lee, "Mining Temporal Patterns in Time Interval-Based Data," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 12, pp. 3318-3331,Dec.1,2015

[8] G. S. Reddy, "Dimensionality reduction approach for high dimensional text documents," *2016 International Conference on Engineering & MIS (ICEMIS)*, Agadir, 2016, pp. 1-6. doi: 10.1109/ICEMIS.2016.7745364

[9] Vangipuram Radhakrishna, Chintakindi Srinivas, and C. V. GuruRao. 2014. A modified Gaussian similarity measure for clustering software components and documents. In Proceedings of the International Conference on Information Systems and Design of Communication (ISDOC '14). ACM, New York, NY, USA, 99--104.

[10] Radhakrishna, V., Veereswara Kumar, P. & Janaki, V. SRIHASS - a similarity measure for discovery of hidden time profiled temporal associations. Multimed Tools Appl (2017). https://doi.org/10.1007/s11042-017-5185-9

[11] Shadi A. Aljawarneh, Radhakrishna Vangipuram, Veereswara Kumar Puligadda, Janaki Vinjamuri, G-SPAMINE: An approach to discover temporal association patterns and trends in internet of things, Future Generation Computer Systems, Volume 74, 2017, Pages 430-443, ISSN 0167-739X, https://doi.org/10.1016/j.future.2017.01.013

[12] Vangipuram Radhakrishna, Shadi A. Aljawarneh, P.V. Kumar, V. Janaki, A novel fuzzy similarity measure and prevalence estimation approach for similarity profiled temporal association pattern mining, Future Generation Computer Systems, Volume 83, 2018, Pages 582-595, ISSN 0167-739X, https://doi.org/10.1016/j.future.2017.03.016

[13] Vangipuram Radhakrishna, Shadi A. Aljawarneh, P.V. Kumar, V. Janaki, ASTRA - A Novel interest measure for unearthing latent temporal associations and trends through extending basic gaussian membership function, Multimedia Tools and Applications, Volume 78, Issue 4, Pages 4217-4265, ISSN 1573-7721, https://doi.org/10.1007/s11042-017-5280-y

[14] Vangipuram Radhakrishna, P. V. Kumar, Vinjamuri Janaki, Shadi. A. Aljawarneh, Juan A. Lara, Khalaf Khatatneh, Krishna Sudarsana—A Z-Space Interest Measure for Mining Similarity Profiled Temporal Association Patterns, Foundations of Science, Pages 1-22, ISSN 1572-8471, https://doi.org/10.1007/s10699-019-09590-y

[15] Shadi A. Aljawarneh, Vangipuram Radhakrishna, John William Atwood, Ultimate: Unearthing Latent Time Profiled Temporal Associations, Foundations of Science, Pages 1-25, ISSN 1572-8471, https://doi.org/10.1007/s10699-019-09594-8

[16] Vangipuram Radhakrishna, Shadi A. Aljawarneh, Puligadda Veereswara Kumar, Kim-Kwang Raymond Choo, A novel fuzzy gaussian-based dissimilarity measure for discovering similarity temporal association patterns, Volume 22, Issue 6, ISSN 1903-1919, https://doi.org/10.1007/s00500-016-2445-y

[17] Shadi Aljawarneh, Vangipuram Radhakrishna, and Gali Suresh Reddy. 2018. Mantra: a novel imputation measure for disease classification and prediction. In Proceedings of the First International Conference on Data Science, E-learning and Information Systems (DATA '18). ACM, New York, NY, USA, Article 25, 5 pages. DOI: https://doi.org/10.1145/3279996.3280021

[18] D. J. Weller-Fahy, B. J. Borghetti and A. A. Sodemann, "A Survey of Distance and Similarity Measures Used Within Network Intrusion Anomaly Detection," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 70-91, Firstquarter 2015. doi: 10.1109/COMST.2014.2336610

[19] Aljawarneh, S.A. & Vangipuram, R. J Supercomput (2018). https://doi.org/10.1007/s11227-018-2397-3

[20] Rajesh Kumar Gunupudi, Mangathayaru Nimmala, Narsimha Gugulothu, Suresh Reddy Gali, CLAPP: A self-constructing feature clustering approach for anomaly detection, Future Generation Computer Systems, Volume 74, 2017, Pages 417-429, ISSN 0167-739X, https://doi.org/10.1016/j.future.2016.12.040.

[21] G. R. Kumar, N. Mangathayaru and G. Narsimha, "Design of novel fuzzy distribution function for dimensionality reduction and intrusion detection," *2016 International Conference on Engineering & MIS (ICEMIS)*, Agadir, 2016, pp. 1-6. doi: 10.1109/ICEMIS.2016.7745346

[22] Gunupudi Rajesh Kumar, Nimmala Mangathayaru, Gugulothu Narsimha, and Aravind Cheruvu. 2018. Feature Clustering for Anomaly Detection Using Improved Fuzzy Membership Function. In Proceedings of the Fourth International Conference on Engineering & MIS 2018 (ICEMIS '18). Association for Computing Machinery, New York, NY, USA, Article 35, 1–9. DOI:https://doi.org/10.1145/3234698.3234733

[23] Wei-Chao Lin, Shih-Wen Ke, Chih-Fong Tsai, CANN: An intrusion detection system based on combining cluster centers and nearest neighbors, Knowledge-Based Systems, Volume 78, 2015, Pages 13-21, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2015.01.009.

[24] Nagaraja, A., Uma, B. & Gunupudi, R.. Found Sci (2019). https://doi.org/10.1007/s10699-019-09589-5

[25] Shadi Aljawarneh, Monther Aldwairi, Muneer Bani Yassein, Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model, Journal of Computational Science, Volume 25, 2018, Pages 152-160, ISSN 1877-7503, https://doi.org/10.1016/j.jocs.2017.03.006

[26] Nagaraja, A., & Satish Kumar, T. (2018). An extensive survey on intrusion detection- past, present, future. In Proceedings of the fourth international conference on Engineering & MIS 2018 (ICEMIS '18). New York: ACM. https://doi.org/10.1145/3234698.3234743.