SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 4, ISSUE 2
ISSN 2054-7390

Fuzzy Rough Classification Models for Network Intrusion Detection

Ashalata Panigrahi[1] and Manas Ranjan Patra[2]
Department of Computer Science, Berhampur University, Berhampur, India
[1]ashalata.panigrahi@yahoo.com; [2]mrpatra12@gmail.com

ABSTRACT

In recent years advancements in communication technology have led to a wide range of Internet services. While an overwhelming number of Internet users have shown interest in such services, incidences of cyber-attacks by miscreants have thwarted their dependence on electronically-accessible services. In order to deal with this alarming situation intrusion detection systems (IDS) have emerged as a potential solution to analyse network activities of users and report attempts of possible intrusions**.** Building an intrusion detection system is a complex and challenging task. This requires analysis of network data from several dimensions so as to develop a pragmatic system to handle different forms of intrusive behaviour of attackers. In this paper, we propose a hybrid intrusion detection approach which combines techniques based on both fuzzy and rough set theories to classify network data as normal and anomalous. Our approach comprises of two phases; in the first phase the most relevant features are extracted using a set of rank and search based methods; and in the second phase we classify the reduced dataset as normal or anomalous using five different classifiers, namely, Fuzzy Nearest Neighbour, Fuzzy-Rough Nearest Neighbour, Fuzzy-Rough Ownership NN, Vaguely Quantified Nearest Neighbours, and Ordered Weighted Average Nearest Neighbours. Experimental results show that the proposed hybrid approach has the ability to achieve high intrusion detection rate and low false alarm

Keywords: FNN, Fuzzy-Rough NN, FRONN, VQNN, OWANN.

1 Introduction

The last decade has witnessed an unprecedented expansion in Internet connectivity which has led to a plethora of internet based services catering to a wide range of user groups. This has evoked security concerns for protecting personal and sensitive data from misuse. As more and more number of users get connected to internet, the window of opportunity for malicious users to fiddle with user data becomes lucrative. Network security deals with the confidentiality, integrity, availability and protection of data as well as computing resources. Different approaches have been adopted to implement a range of security measures such as authentication, cryptography, firewalls, antivirus, spywares, Virtual Private Network, and intrusion detection systems (IDS) but none of them is capable of providing complete security. Malicious users constantly look for ways to by-pass the security features, and many-a-times succeed in accessing important network resources. As a result developing flexible and adaptive security systems is a major challenge. In this context, IDSs are becoming important tools to ensure network security where IDSs

DOI: 10.14738/tmlai.42.1882
Publication Date: 15th Mar, 2016
URL: http://dx.doi.org/10.14738/tmlai.42.1882

are deployed to dynamically monitor all incoming and outgoing network activities taking place in a system and distinguish between legitimate and anomalous network users. Hybrid IDS are dynamic defensive systems, capable of adapting to dynamically changing traffic patterns and try to detect varieties of network attacks.

2 Related Work

Classification techniques are being used to build predictive models in different application domains. Network intrusion detection is one such area which extensively uses different classifiers to build predictive models to distinguish between intrusions and normal connection requests in a network setup. Several works have been reported utilizing different classification techniques to analyse intrusion data and build prediction models with the sole objective of enhancing intrusion detection accuracy and lowering false alarms.

Gong, S [5] has proposed a feature selection approach based on Genetic Quantum Particle Swarm Optimization (GQPSO) for network intrusion detection wherein selection and variation of genetic algorithm with QPSO algorithm have been combined to reduce redundant and irrelevant features. Experimental results show that the GQPSO algorithm performs better than PSO and QPSO algorithms in terms of detection rate and speed of classification. Hoque et al**.** [6] have implemented an Intrusion Detection System by applying genetic algorithms to efficiently detect various types of network intrusive activities. To measure the efficiency of their system they used the standard KDD 99 intrusion detection benchmark dataset and obtained realistic detection rate. But their performance of detection rate was poor and they failed to reduce the false positive rate. Zhou et al. [7] presented a hierarchical neuro-fuzzy inference intrusion detection system (HFIS). In their proposed system, principal component analysis neural network was used to reduce the input data space. An enhanced fuzzy c-means clustering algorithm was applied to create and extract fuzzy rules. The adaptive neural fuzzy inference system was utilized repeatedly in their model. At last, the system was optimized by genetic algorithm. The main advantages of the HFIS model are its capability to perform not only misuse detection but also anomaly detection. Moreover, their method has higher speed and better performance.

Tong et al. [8] have proposed a hybrid IDS based on RBF/Elman neural network wherein the RBF neural network is employed as a real time pattern classifier while Elman neural network is employed to restore the memory of past events. Mohamadi H [9] proposed Simulated Annealing (SA) based fuzzy system to develop an Intrusion Detection System (IDS). The use of SA in IDS is an attempt to effectively explore and exploit the large search space associated with intrusion detection classification problem. Experiments were carried out on 10% of KDD Cup99 dataset of UCI KDD archive. Due to the imbalanced records in the dataset a subset of the dataset was used as training and testing sets (20752 randomly generated samples) and normalized between 0.0 and 1.0. Initial set of fuzzy if-then rules was generated and initial temperature was set as 100. The fitness of the rule was evaluated by number of correctly classified training patterns. The results showed that average accuracy rate obtained was varying from 94% to 99% with the number of rules ranging from 50 to 100. This approach was compared with the different baseline classifiers including pruning C4.5, Naïve Bayes, K-NN, SVM and multi-objective genetic fuzzy IDS. The results showed that the proposed approach obtained highest accuracy (92.89%), better precision, lowest classification cost (0.2093), F-measure, recall than other classifiers. In our previous work [10] we have proposed a hybrid classification model based on evolutionary computation based techniques. The result

shows that AIRS1 classifier with best first search feature selection gives highest accuracy and AIRS2 classifier with Gain Ratio feature selection gives lowest false alarm rate.

3 Proposed Hybrid Intrusion Detection Model

The aim of this work is to build a high performance hybrid intrusion detection model that can achieve low false alarm rate and high detection rate. The model comprises of two levels as depicted in figure 1.

Level-1consists of feature selection methods to extract the most relevant features from the intrusion dataset which can contribute to the classification process. This is achieved by identifying the irrelevant and redundant information in the intrusion dataset and discarding them from the dataset. Four different rank methods namely, Gain Ratio, Relief-F, One-R, Symmetrical Uncertainty and three different search methods namely, Best First, Greedy Stepwise, Rank Search have been applied for selection of relevant attributes. At Level-2 the reduced data obtained from Level-1 is classified using five classification techniques namely, Fuzzy Nearest Neighbour, Fuzzy-Rough Nearest Neighbour, Fuzzy-Rough Ownership NN, Vaguely Quantified Nearest Neighbours, and Ordered Weighted Average Nearest Neighbours. The NSL-KDD dataset has been used for building and validating the model. Further, 10-Fold cross-validation has been employed for analysis of detection rate, accuracy, false alarm rate, and fitness value.

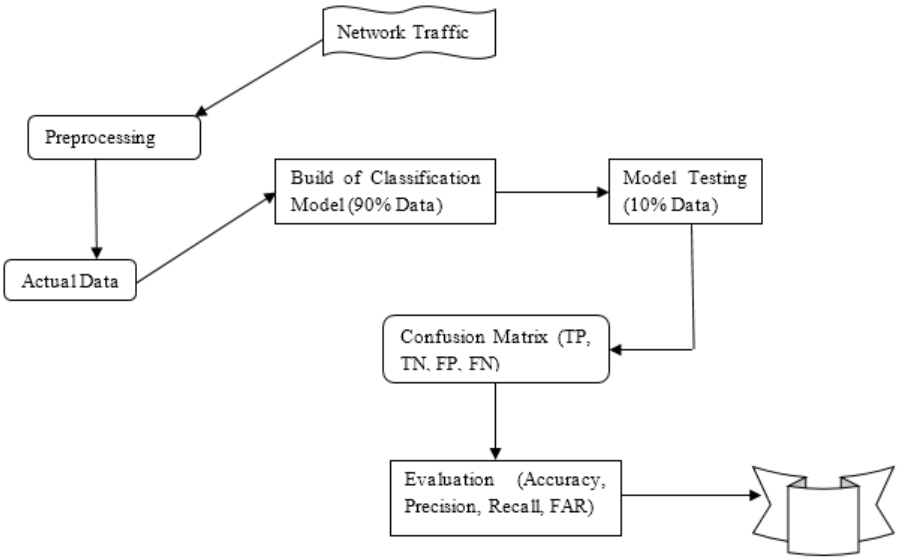

Figure. 1 System diagram for Hybrid IDS

4 Methodology

4.1 Hybridization of Rough Set and Fuzzy Set

Fuzzy Set

A fuzzy set [4] in X is an $X \rightarrow [0, 1]$ mapping, while a fuzzy relation in X is a fuzzy set in $X \times X$. For all y in X, the R-forest of y is the fuzzy set R_y is defined by

$$R_y(x) = R(x,y) \quad (1)$$

For all x in X, if R is reflexive and symmetric fuzzy relation, that is

$$R(x,x) = 1 \quad (2)$$

$$R(x,y) = R(y,x) \quad (3)$$

holds for all x and y in X, then R is called a "fuzzy tolerance ratio".

Rough Set

Rough Set Theory is a mathematical tool to deal with imprecise and insufficient knowledge [3]. In rough set theory, membership is not the primary concept unlike fuzzy sets. It deals with inconsistency, uncertainty, and incompleteness by imposing an upper and a lower approximation to set membership. The advantage of rough set theory is that it does not require any preliminary or additional information about data, like probability in statistics or grade of membership/value of possibility in fuzzy set theory.

Let (X, A) be an information system where X is the universe of discourse and A is a non-empty finite set of attributes such that a: $X \rightarrow V_a$ for every a Є A. The set V_a is called the "value set of a". Given B ⊆A there is an associated equivalence relation R_B:

$$R_B = \{ (x,y) \in X^2 \mid \forall a \in B, a(x) = a(y) \} \quad (4)$$

If (x,y) Є R_B, then x and y are indiscernible by attributes from B. The equivalence classes of the B-indiscernibility relation are denoted by $[x]_B$.

Let A be a subset X. A can be approximated using the information contained within B by constructing the B-lower and B-upper approximations of A.

$$R_B{\downarrow}A = \{ x \in X \mid [x]_B \text{ subset } A \} \quad (5)$$

$$R_B{\uparrow}A = \{ x \in X \mid [x]_B \cap A \neq \emptyset \} \quad (6)$$

The tuple ($R_B{\downarrow}A$, $R_B{\uparrow}A$) is called a rough set.

Fuzzy-Rough Set Theory

Hybridizing fuzzy rough set theory is focused mainly on fuzzifying the formulas for lower and upper approximations [2]. Given a fuzzy tolerance relation R and a fuzzy set A in X, the lower and upper approximation of A by R can be defined as:

$$(R{\downarrow}A)(x) = \inf_{y \in X} I(R(x,y),A(y)) \quad (7)$$

$$(R{\uparrow}A)(x) = \sup_{y \in X} T(R(x,y),A(y)) \quad (8)$$

Here *I* is an implicator and *T*is a t-norm.

4.2 Fuzzy Nearest Neighbour Classification

The Fuzzy Nearest Neighbour (FNN) algorithm [11] was introduced to classify test objects based on their similarity to a given number K of neighbours, and these neighbours' membership degree to (crisp or fuzzy) class labels. For the purpose of (FNN), the extent *C'(y)* to which an unclassified object y belongs to a class *C*is computed as:

$$C'(y) = \sum_{x \in N} R(x,y)\, C(x) \quad (9)$$

where N is the set of object y's K nearest neighbours, and *R(x,y)* is the [0,1]-valued similarity of *x* and *y*.

The Fuzzy K-Nearest Neighbour Algorithm

FNN (X, C_D, y, K)
Input: X: the training data set; C_D: the set of decision classes;
y: the objects to be classified; K: the number of nearest neighbours
1. begin
2. N ← get Nearest Neighbours (y, K)
3 for each C ∈ C_D do
4. C' (y) = $\sum_{x \in N} R(x,y)\, C(x)$
5. end
6. end
Output: arg max (C' (y))

4.3 Fuzzy-Rough Nearest Neighbour Classification

In Fuzzy-Rough Nearest Neighbour (FRNN) algorithm the nearest neighbours are used to construct the fuzzy lower and upper approximations of decision classes, and test instances are classified based on their membership to these approximations. FRNN algorithm combines fuzzy-rough approximation with the classical FNN approach [12]. The rationale behind the algorithm is that the lower and upper approximation of a decision class, calculated by means of the nearest neighbours of a test object y, provides good clues to predict the membership of the test object to that class. The algorithm is dependent on the choice of a fuzzy tolerance relation R. Given the set of conditional attributes A, the fuzzy tolerance relation R is defined by

$$R(x,y) = \min_{a \in A} R_a(x,y) \tag{10}$$

in which $R_a(x,y)$ is the degree to which objects *x* and *y* are similar for attribute a. Here we choose

$$R_a(x,y) = 1 - \frac{|a(x) - a(y)|}{|a_{max} - a_{min}|} \tag{11}$$

If (R↓C)(y) is high , it reflects that all of y's neighbours belong to C. A high value of (R↑C) means that at least one neighbour belongs to that class.

The Fuzzy Rough Nearest Neighbour Algorithm:

FRNN ***(X, C_D, y)***
Input: X: the training data set; C_D, the set of decision classes;
y: the objects to be classified;
1.begin
2. N ← get Nearest Neighbours (y, K)
3. τ← 0 , Class ← Ø
4. for each C ∈ C_D do
5. . if((R↓C)(y) + (R↑C)(y)) / 2 ≥ τ then
6. Class ← C
7. τ ← ((R↓C)(y) + (R↑C)(y)) / 2
8. endif
9 . end
10. end

Output Class

4.4 Fuzzy-Rough Ownership NN Classification

A fuzzy-Rough ownership is an attempt to handle both "fuzzy uncertainty" caused by overlapping classes and "rough uncertainty" caused by insufficient knowledge [12]. All training objects influence the ownership function. The algorithm does not use fuzzy lower or upper approximations to determine class membership. The fuzzy-rough ownership function τ_c of class *C* for an object *y* is defined as,

$$\tau_c(y) = \sum_{x \in X} \frac{R(x,y)C(x)}{|X|} \quad (12)$$

where the fuzzy relation R is determined by

$$R(x,y) = \exp(-\sum_{a \in A} Ka(a(y) - a(x))2/(m - 1)) \quad (13)$$

where m controls the weighting of the similarity and K_a is a parameter that decides the bandwidth of the membership and K_a is defined as

$$K_a = \frac{|X|}{2 \sum_{x \in X} \| a(y) - a(x) \|2/(m-1)} \quad (14)$$

$\tau_c(y)$ is interpreted as the confidence with which y can be classified to class C.

FROWNN (X, A, C_D, y)
Input: X the training data set; A the set of conditional features;
C_D the set of decision classes; y the object to be classified.

1. begin
2. for each a ∈ A do
3. $K_a = \frac{|X|}{2 \sum_{x \in X} \| a(y) - a(x) \|2/(m-1)}$
4. end
5. N ← I x I
6. for each C ∈ C_D do $\tau_c(y) = 0$
7. for each x ∈ N do
8. $d = \sum_{a \in A} Ka\ (a(y) - a(x))^2$
9. for each C ∈ C_D do
10. $\tau_c(y) += C(x).\ \exp(-d^{1/(m-1)}) / I N I$
11. end
12. end
13. end
14. end

Output $\underset{C \in C_D}{\text{argmax}} \tau_c(y)$

4.5 Vaguely Quantified Nearest Neighbours Classification

VQNN [12] depends only on the summation of the similarities of each class. It uses the linguistic quantifiers "most" and "some". Given a couple (Q_u, Q_l) of fuzzy quantifiers that represent "most" and "some" respectively, the lower and upper approximation of C. VQNN assigns a class to a target instance y as follows:

i. Determine NN, the K nearest neighbours of *y*.
ii. Assign y to the class *C* for which $(R{\downarrow}^{Qu} C)(y) + (R{\uparrow}^{Ql} C)(y)$ is maximal.

The upper and lower approximation of Vaguely Quantified rough sets are defined as

$$((R{\downarrow}^{Qu} C)(y)) = Q_u(\frac{\sum_{x \in X} \min(Rx,y),C(x)}{\sum_{x \in X} R(x,y)}) \quad (15)$$

$$((R \uparrow^{Q_l} C)(y)) = Q_l\left(\frac{\sum_{x \in X} \min(Rx,y), C(x)}{\sum_{x \in X} R(x,y)}\right) \quad (16)$$

The operators Q_u and Q_l are fuzzy quantifiers that represent most and some respectively. They are increasing [0, 1] → [0, 1] mapping such that

$$Q_u(1) = Q_l(1)=1 \text{ and } Q_u(0) = Q_l(0)=0$$

This classifier based on rough set theory is capable of handling noise data.

4.6 Ordered Weighted Average Nearest Neighbours Classification

The OWA operator [13] models an aggregation process in which a sequence A of n scalar values are ordered decreasingly and then weighted according to their ordered position by a weighting vector W = {w1, w2, ……,wp}. The OWA_W operator aggregates p values A = { a1, a2, ……., ap } as follows:

$$OWA_W(a1, a2, \ldots\ldots, ap) = \sum_{i=1}^{p} wi\ bi \quad (17)$$

where bi= aj if aj is the i-th largest value in A = {a1, a2, ……., ap}.

The weights W are associated with ordered positions. The higher values in {a1, a2, ……,ap} are assigned to the first weights in W and the lower values are associated with the last weights in W.

Let R be a fuzzy relation in X and A a fuzzy set in X = { x1, x2, ……., xn}. Let Ţ be a t-norm and I, a fuzzy implication. The OWA-based lower and upper approximation of A under R with weight vectors W_l and W_u are defined as

$$(R \downarrow W_l A)(y) = OW A_{Wi}(I(R(x_i, y), A(x_i))) \quad (18)$$

$$(R \uparrow W_u A)(y) = OW A_{Wu}(I(R(x_i, y), A(x_i))) \quad (19)$$

5 Experimental Setup

5.1 NSL-KDD Dataset

NSL- KDD is a dataset proposed by Tavallace et al. [14] which is a reduced version of the original KDD'99 dataset. NSL-KDD consists of same features as KDD'99 training dataset but has the following advantages over the original KDD'99dataset.

a) The training set does not include redundant records.
b) The test set has no duplicate records.
c) The number of records in the training and test set is reasonable, which makes it affordable to run experiments on the complete set without the need to randomly select a small portion. Consequently, the evaluation of results reported by different researchers can be comparable.

The data set consists of 41 feature attributes out of which 38 are numeric and 3 are symbolic. Total number of records in the data set is 125973 out of which 67343 are normal and 58630 are attacks. The dataset contains different attack types that could be classified into four main categories namely, Denial of Service (DOS), Remote to Local (R2L), User to Root (U2R), and Probing

The percentage distribution of records under each category of attack is provided in Table 1 and figure 2.

Table 1 Data Distribution of NSL-KDD Dataset

Class	Number of Records	% of occurrence
Normal	67343	53.48%
DOS	45927	36.45%
R2L	995	0.78%
Probes	11656	9.25%
U2R	52	0.04%
Total	125973	100%

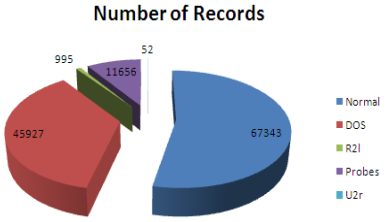

Figure. 2 Distribution of Records

5.2 Feature Selection

In order to build a high performance IDS, selection of the most relevant features present in the intrusion dataset is an important research challenge. Feature selection can be defined as a process that chooses a minimum subset of M features from the original set of N features, so that the feature space is optimally reduced according to certain evaluation criteria. As the dimensionality of a domain expands, the number of features N increases. Finding the best feature subset is usually intractable [15].

Feature selection improves classification performance by searching for the subset of features, which best classifies the training data. In case of high dimensional feature space, some of the features may be redundant or irrelevant. Removing these redundant or irrelevant features is very important as they may deteriorate the performance of classifiers. Feature selection involves finding a subset of features from the dataset, thereby decreasing the size of the original dataset in order to improve prediction accuracy of the classifier [16]. Now, we present the feature selection techniques that we have applied for reducing the NSL-KDD dataset with the most desirable features which can improve the performance of the classifiers.

Gain Ratio

The information gain measure prefers to select features having a large number of values. The extension of information gain is known as gain ratio [17] and is based on ranking which attempts to overcome any bias. It applies a kind of normalization to information gain using a "split information" value. The split information value represents the potential information generated by splitting the training dataset D into v partitions, corresponding to v outcomes on attribute A

$$\text{SplitInfo}_A(D) = -\sum_{j=1}^{v} |Dj| / |D| * \log_2(|D_j| / |D| \quad (20)$$

This value represents the potential information generated by splitting the training dataset D into v partitions corresponding to the v outcomes of a test on attribute A.

The gain ratio is defined as

$$GainRatio(A) = Gain(A) / SplitInfo(A) \quad (21)$$

The feature with the maximum gain ratio is selected as the splitting attribute.

One-R

One-R (short for One Rule) algorithm proposed by Holte [18] is a simple classification algorithm that generates a one-level decision tree expressed in the form of a set of rules all of which test one particular feature. It is capable of generating good rules for characterizing the structure in data.. One-R can handle missing values and numeric features. The One-R algorithm generates rules and tests a single feature at a time and a branch for every value of that feature. For every branch, the class with the best classification is selected.

Relief-F

Relief-F feature selection method is one of the most successful algorithms for assessing the quality of features due to its simplicity and effectiveness. Relief-F can handle noise and multiclass datasets [19]. Relief-F feature evaluation [20] evaluates the worth of a feature by repeatedly sampling an instance and considering the value of the given feature for the nearest instance of the similar and different classes. This feature evaluation assigns a weight to each feature based on the ability of the feature to distinguish among the classes, and then selects those features whose weights exceed a user-defined threshold as relevant features. The three basic steps of Relief-F feature evaluator technique are:

- Calculate the nearest miss and nearest hit
- Calculate the weight of a feature
- Return a ranked list of features or the top K features according to a given threshold

The function diff (Feature, Instance1, Instance2) computes the difference between the values of a feature for two different instances. For discrete attributes the difference is either 1 (the values are different) or 0 (the values are the same), whereas for continuous features the difference is the actual difference normalized to the interval [0, 1]. Higher the value of m (the number of instance sampled), the more reliable is Relief-F's estimate.

Symmetrical Uncertainty

Symmetrical uncertainty technique [17] is symmetric in nature and it reduces the number of comparisons required. It is not influenced by multi-valued features and its values are normalized to the range [0, 1]. This technique consists of two phases to select the most informative features to target classes from the original feature space. In the first phase (lines 1-5 in the algorithm), irrelevant features with poor prediction ability to target a class are removed. In the second phase (lines 7-12 in the algorithm) redundant features that are inter-correlated with one or more of other features are eliminated.

Given a dataset with a number of input features and a target class, the algorithm first calculates the mutual information between features and class. The algorithm then ranks the features in descending order according to their degrees of association to the target class. Once the input features are ranked, those terms whose information measures are greater than zero are kept; which means the removed features are totally irrelevant to target class and the remaining ones are predictive. Next, it starts by calculating the inter-correlated strengths of each pair of features. The total amount of mutual information for each feature is acquired by adding all mutual information measures together that relate to the feature.

Best First Search

Best First Search (BFS) [21] uses classifier evaluation model to estimate the merits of features. The feature with high merit values are considered as potential features and thus selected for classification. Best first moves through the search space by making local changes to the current feature subset. It searches the space of feature subsets by augmenting with a backtracking facility. Given enough time, a best first search will explore the entire search space, thus it is common to use a stopping criterion. It may start with an empty set of features and search forward, or start with the full set of features and search backward, or start at any point and search in both the directions.

Greedy Stepwise Search

Greedy Stepwise search [21] performs a greedy forward or backward search through the space of feature subsets. It may start with no / all features or from an arbitrary point in the space and stops when addition/ deletion of any feature results in decrease in evaluation. This can also produce a ranked list of features by traversing the space from one side to other and recording the order in which features are selected.

Rank Search

This uses a subset evaluator to rank all features. If a subset evaluator is specified, then a forward selection search is used to generate a ranked list. Next, from the ranked list of features a subset of best feature set is selected. Table 4 enlists the features selected after application of each of the above feature selection technique.

Table 4 Selected Attributes after Feature Selection

Feature Selection Method	No. of Features Selected	Feature Names
Gain Ratio	10	Flag, Src_bytes, Dst_bytes,Logged_in, Serror_rate, Srv_serror_rate, Same_srv_rate, Diff_srv_rate, Dst_host_serror_rate, Dst_host__srv_serror_rate.
One-R	14	Service,Flag, Src_bytes, Dst_bytes, Count,Serror_rate, Srv_serror_rate, Same_srv_rate, Diff_srv_rate, Dst_host_srv_count, Dst_host_same_srv_rate, Dst_host_diff_srv_rate, Dst_host_serror_rate, Dst_host__srv_serror_rate.
Relief Attribute Evaluator	12	Protocol_type, Service, Flag, Count,Same_srv_rate, Dst_host_count, Dst_host_srv_count, Dst_host_same_srv_rate, Dst_host_diff_srv_rate, Dst_host_same_srv_port_rate, Dst_host_serror_rate, Dst_host_rerror_rate
Symmetrical Uncertain Attribute Evaluator	16	Service, Flag, Src_bytes, Dst_bytes, Logged_in, Count, Serror_rate, Srv_serror_rate, Same_srv_rate, Diff_srv_rate, Dst_host_srv_count, Dst_host_same_srv_rate, Dst_host_diff_srv_rate, Dst_host_srv_diff_host_rate, Dst_host_serror_rate, Dst_host__srv_serror_rate.
Best First Search	13	Duration, Service, Src_bytes, Dst_bytes, Logged_in, Count,Ser_rate,Dst_h_co, Ds_ho_sr, Ds_Rate, Ds_d_h_rt, Ds_h_r, Ds_hrr.
Rank Search	13	Service, Flag, Src_bytes, Dst_bytes, Logged_in, Root_shell, Serror_rate, Srv_serror_rate, Same_srv_rate, Diff_srv_rate, Dst_host_srv_diff_host_rate, Dst_host_serror_rate, Dst_host__srv_serror_rate
Greedy Stepwise	11	Service,Flag, Src_bytes, Dst_bytes, Logged_in, Root_shell, Srv_serror_rate,Same_srv_rate, Diff_srv_rate, Dst_host_srv_diff_host_rate, Dst_host_serror_rate,

Cross Validation

Cross validation calculates the accuracy of the model by separating the data into two different populations, a training set and a testing set. In k-fold cross-validation [17] the dataset is randomly partitioned into n mutually exclusive folds, T_1, T_2, …..,T_n each of approximately equal size. Training and testing are performed n times. Each training set consists of (n – 1)/n th of the dataset and the remaining 1/n th is used as test data In 10-fold cross validation, a given dataset is partitioned into 10 subsets. Out of these 10 subsets, 9 subsets are used to perform a training fold and a single subset is retained as testing data. This cross-validation process is then repeated 10 times (the number of folds). The 10 sets of results are then aggregated by averaging to produce a single model estimation. The advantage of 10-fold cross validation over random sub-sampling is that all objects are used for both training and testing, and each object is used for testing only once per fold.

Confusion Matrix

The confusion matrix is a table with two rows and two columns that reports the number of False Positive, False Negative, True Positive, True Negative. The confusion matrix maintains the information about actual and predicted classes. An IDS is evaluated by its ability to make accurate prediction of attacks. Intrusion detection systems mainly discriminate between two classes, attack class (abnormal data), and normal class (normal data). While classifying the attacks and normal access behaviour of users, there can be four possibilities as depicted in Table 5 such as True Positives, False Positives, True Negatives, and False Negatives.

Table.5 IDS Confusion matrix

		Predicted Class	
		Negative Class (Normal)	Positive Class (Attack)
Actual Class	Negative Class(Normal)	True Negative (TN)	False Positive (FP)
	Positive Class (Attack)	False Negative (FN)	True Positive (TP)

The accuracy, detection rate, precision, F-value, false alarm rate, fitness value are calculated as follows

Accuracy measure the probability that the algorithm can correctly predict positive and negative examples and is given by:

$$\text{Accuracy} = \frac{TP+TN}{TN+TP+FN+FP}$$

$$\text{Detection Rate or Recall} = \frac{TP}{TP+FN}$$

Precision is a measure of the accuracy provided that a specific class has been predicted and it is calculated as:

$$\text{Precision} = \frac{TP}{TP+FP}$$

F- Value is the harmonic mean of Precision and Recall which measures the quality of classification which is given by:

$$\text{F - Value} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

False Alarm Rate is defined as the number of normal instances incorrectly labelled as intrusion divided by the total number of normal instances and is given by:

$$\text{False Alarm Rate} = \frac{FP}{TN+FP}$$

$$\text{Fitness Value} = \frac{TP}{TP+FP} * \frac{TN}{TN+FP}$$

6 Results and Discussion

Here, we study the effectiveness of the hybrid intrusion detection model that uses five classification techniques, viz., Fuzzy Nearest Neighbour, Fuzzy-Rough Nearest Neighbour, Fuzzy-Rough Ownership NN, Vaguely Quantified Nearest Neighbours, Ordered Weighted Average Nearest Neighbour along with different feature selection methods. The performance of different combinations of classifiers and feature selection methods are evaluated on the basis of accuracy, detection rate, precision, F-value, false alarm rate, fitness value, and error rate. The results are summarized in Table 6 and Table 7.

Table 6 Comparison of Accuracy, Detection rate, precision, F-value, false alarm rate, fitness value, of five classification techniques using Ranking Attribute Reduction methods

Attribute Reduction Method	Test Mode	Classifier Techniques	Accuracy in %	Detection Rate in %	Precision in %	F-Value in %	False Alarm Rate in %	Fitness Value in %
One-R	10-Fold Cross-Validation	Fuzzy NN	99.2427	99.0159	99.3548	99.185	0.5598	98.4614
		Fuzzy Rough NN	98.9712	98.9749	98.8165	98.8956	1.0320	97.9534
		Fuzzy Ownership NN	**99.4292**	**99.5037**	**99.3139**	**99.4087**	**0.5986**	**98.908**
		VQNN	98.8998	98.7105	98.9231	98.8167	0.9355	97.7871
		OWANN	98.9109	98.7395	98.9184	98.8288	0.934	97.8113
Relief-F	10-Fold Cross-Validation	Fuzzy NN	89.4414	88.7054	88.6193	88.6623	9.9179	79.9077
		Fuzzy Rough NN	99.4753	99.2734	99.5979	99.4354	0.3489	98.9269
		Fuzzy Ownership NN	99.2856	99.34	99.1269	99.2334	0.7618	98.5832
		VQNN	**99.4792**	**99.3809**	**99.4996**	**99.4402**	**0.4351**	**98.9485**
		OWANN	99.4507	99.3553	99.4638	99.409	0.3267	98.892
SU	10-Fold Cross-Validation	Fuzzy NN	99.2935	99.0141	99.3888	99.2011	0.833	98.4892
		Fuzzy Rough NN	99.3499	99.3399	99.2637	99.3018	0.6415	98.7026
		Fuzzy Ownership NN	**99.542**	**99.5173**	**99.5411**	**99.5292**	**0.3996**	**99.1192**
		VQNN	99.2252	99.2922	99.0455	99.1681	0.833	98.465
		OWANN	99.207	99.2819	99.0168	99.1492	0.8583	98.4298
Gain Ratio	10-Fold Cross-Validation	Fuzzy NN	96.8898	98.0641	95.3831	96.705	4.1326	94.0115
		Fuzzy Rough NN	98.941	98.2398	99.4784	98.8528	0.4484	97.7992
		Fuzzy Ownership NN	**99.1609**	**98.6798**	**99.556**	**99.1199**	**0.3832**	**98.3016**
		VQNN	98.9387	98.3575	99.3556	98.8569	0.5554	97.8112
		OWANN	98.9315	98.3592	99.3385	98.8464	0.5702	97.7984

It is observed that Fuzzy ownership nearest neighbour classification technique with symmetrical uncertainty feature selection yields better accuracy and low false alarm rate than other classification techniques. A comparison of classifiers with respect to accuracy, recall / detection rate, and false alarm rate is presented in figures 3, 4, and 5 respectively using rank based feature selection.

Table7 Comparison of Accuracy, Detection rate, precision, F-value, false alarm rate, fitness value of five classification techniques using Searching Attribute Reduction methods

Attribute Reduction Method	Test Mode	Classifier Techniques	Accuracy in %	Detection Rate in %	Precision in %	F-Value in %	False Alarm Rate in %	Fitness Value in %
Best First Search	10-Fold Cross-Validation	Fuzzy NN	99.5594	99.6401	99.4654	99.5526	0.5108	99.1311
		Fuzzy Rough NN	99.5142	99.49	99.4322	99.461	0.4648	99.276
		Fuzzy Ownership NN	**99.5729**	**99.606**	**99.533**	**99.5694**	**0.4071**	**99.2006**
		VQNN	99.3761	99.3485	99.3112	99.3299	0.5999	98.7524
		OWANN	99.3403	99.3263	99.2569	99.2916	0.6473	98.6833
Greedy Stepwise	10-Fold Cross-Validation	Fuzzy NN	95.0473	92.0706	97.1388	94.5352	2.361	89.8967
		Fuzzy Rough NN	99.615	99.4849	99.6872	99.5859	0.2717	99.2145
		Fuzzy Ownership NN	**99.6356**	**99.6145**	**99.6451**	**99.6288**	**0.309**	**99.3067**
		VQNN	99.438	99.3246	99.4671	99.3958	0.4633	98.8644
		OWANN	99.4221	99.316	99.4416	99.3775	0.4856	98.8337
Rank Search	10-Fold Cross-Validation	Fuzzy NN	95.0648	92.0808	96.2335	94.1114	2.3373	89.9286
		Fuzzy Rough NN	**99.6594**	**99.5156**	**99.718**	**99.6167**	**0.245**	**99.0285**
		Fuzzy Ownership NN	99.634	99.6026	99.6536	99.6281	0.3016	99.3018
		VQNN	99.4546	99.3399	99.4876	99.4138	0.4455	98.8973
		OWANN	99.4372	99.3314	99.4586	99.3963	0.4707	98.8638

Here, it is observed that Fuzzy-Rough nearest neighbour classification technique with rank search feature selection method provides better accuracy and low false alarm rate compared to other classification techniques.

On analysing the performance of different classifiers in combination with different ranking and search methods, it is found that Fuzzy-Rough nearest neighbour classification technique with rank search method performs much better in comparison to all other combinations.

Further, we have compared our results with some of the important results reported by other researchers, which is presented in Table 8. It is observed that there is significant improvement in terms of detection rate and false alarm rate. This shows the efficacy of our approach.

Table 8 Comparison of results between the proposed approach with that of the existing ones

Author	Dataset	Feature Selection Method	Classifier Techniques	Detection Rate	False Alarm Rate
Li et al.(2007) [22]	KDD Cup 99	Chi Squared Attribute Evaluator	Transductive Confidence Machines for K-Nearest Neighbour {TCM-KNN)	99.6%	0.1%
Kavitha et al. (2012) [23]	KDD Cup 99	Best First Search	Fuzzy Rule based Intrusion Detection (FRID)	95.47%	10.63%
			Intuitionistic Fuzzy Rule based Intrusion detection (IFRID)	97.86%	5.03%
			Emerging Neutrosophic Logic Classifier Rule based Intrusion Detection (ENLCID)	99.02%	3.19%
Chen et al. (2009) [24]	KDD Cup 99	Rough Set	Support Vector Machine (SVM)	86.72%	13.27%
Sindhu et al. (2012) [25]	KDD Cup 99	Wrapper Approach	Neurotree	98.38%	Not Provided
Sadek et al. (2013) [26]	NSL-KDD	Rough Set	Neural Network with Indicator Variable (NNIV)	96.7%	3.0%
Our Hybrid Approach	**NSL-KDD**	**Greedy Stepwise Search**	**Fuzzy Ownership NN**	**99.6145%**	**0.309%**

7 Conclusions

Building effective intrusion detection models is a challenging task. One of the approaches widely tried out is to classify user behaviour and raise alarms on detecting any anomalous behaviour. Keeping this in view several classifiers have been used but none of the classifier alone is capable of producing acceptable performance. Therefore, work has begun to design hybrid classifiers to improve upon the performance of IDS. The present research is a step forward in this direction where a hybrid model has been proposed with the help of five classifiers and two different categories of feature selection methods. The performances of the classifiers have been evaluated on the basis of accuracy, detection rate, false alarm rate, fitness value, etc. It is observed that the Fuzzy-Rough nearest neighbour classification technique with rank search method performs better in terms of detection rate and reduced false alarms than its counterparts. This observation can certainly help IDS developers in achieving greater accuracy and reducing false alarms. In future, we shall explore application of other hybrid approaches to further improve upon the detection rate and even classify specific attack types.

REFERENCES

[1] Abraham A, Thomas JP, Chebrolu S (2005) Feature deduction and ensemble design of intrusion detection systems. Computers & Security 295-307. doi: 10.1016/j.cose.2004.09.008

[2] Dubois D, Prade H (1992) Putting rough sets and fuzzy sets together. In: Huang S (ed) Intelligent Decision Support, Springer, Netherlands, pp.203-232 .

[3] Pawlak Z (1991) Rough sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishing. Springer, Netherlands

[4] Zadeh L (1965) Fuzzy sets. Information and Control 338-353. doi: 10.1016/S0019-9958(65)90241-X.

[5] Gong, S. (2011) Feature Selection Method for Network Intrusion Based on GQPSO Attribute Reduction, International Conference on Multimedia Technology (ICMT), 6365 – 6368. doi: 10.1109/ICMT.2011.6003117

[6] Hoque MS, Mukit MA, Bikas MAN (2012) An Implementation of Intrusion Detection System using Genetic Algorithm, International Journal of Network Security and Its Applications (IJNSA), .109-120. doi: 10.5121/ijnsa.2012.4208

[7] Zhou YP, Fang JA. (2009) Intrusion Detection Model Based on Hierarchical Fuzzy Inference System". Second IEEE International Conference on Information and Computing Science. IEEE Computer Society, 144–147. doi: 10.1109/ICIC.2009.145

[8] Tong X, Wang Z, Yu H (2009) A research using hybrid RBF/Elman neural network for intrusion detection system secure model. Computer Physics Communications 1795-1801. doi: 10.1016/j.cpc.2009.05.004

[9] Mohamadi H, Habibi J, Abadeh MS (2008) Misuse intrusion detection using a Fuzzy-Meta-heuristic approach. In Proceedings of 2nd Asia Intl. Conference on modeling and simulation 439-444. doi: 10.1109/AMS.2008.128

[10] Panigrahi **A,** Patra, MR (2015) An Evolutionary Computation based Classification Model for Network Intrusion Detection, International Conference on Distributed Computing and Internet Technology (ICDCIT-2015) 318-324. doi: 10.1007/978-3-319-14977-6-31

[11] Killer JM, Gray MR, Givens JA (1985) A Fuzzy K-Nearest Neighbour Algorithm. Systems Man and Cybernet 580-585. doi: 10.1109/TSMC.1985.6313426

[12] Jesen R, Cornelis, C (2008) A New Approach to Fuzzy-Rough Nearest Neighbour Classification. Rough sets and current trends of computing 310-319. doi: 10.1007/978-3-540-88425-5_32

[13] Yager, RR (1988) On ordered weighted averaging aggregation operators in multicriteria decision making, Systems, Man and Cybernetics 183-190. doi: 10.1109/21.87068

[14] Tavallaee M, Bagheri E, Lu W, Ghorbani A (2009) A detailed analysis of the KDD CUP 99 data set. Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defence Applications 1-6. doi: 10.1109/CISDA.2009.5356528

[15] Kohavi R, John GH (1997) Wrapper for feature subset selection . Artificial Intelligence 273-324. doi: 10.1016/S004-3702(97)00043-X

[16] Koller D, Sahami M (1995) Toward optimal feature selection. In Proceeding of International Conference on Machine Learning.284-92.

[17] Han J, Kamber M (2006) Data Mining Concepts and Techniques, 2nd edn, Morgan Kaufmann, San Francisco.

[18] Holte RC (1993) Very simple classification rules perform well on most commonly used datasets. Machine Learning 63-90. doi: 10.1023/A:1022631118932

[19] Kononenko I (1994) Estimating attributes: Analysis and extension of relief. In Proceedings of the Seventh European Conference on Machine Learning. 171-182. doi: 10.1007/3-540-57868-4_57

[20] Marko RS, Igor K (2003) Theoretical and empirical analysis of relief and relief. Machine Learning Journal 23-69. doi: 10.1023/A:1025667309714.

[21] Rich E, Knight K (1991) Artificial Intelligence. McGraw-Hill, 2nd Edition, New-York

[22] Li Y, Guo L (2007) An Active Learning based TCM-KNN Algorithm for Supervised Network Intrusion Detection. Computers & Security. 459-467. doi: 10.1016/j.cose.2007.10.002

[23] Kavitha B, Karthikeyan S, Maybell PS (2012) An Ensemble Design of Intrusion Detection System for Handling Uncertainty using Neutrosophic Logic Classifier. Knowledge-Based Systems 88-96. doi: 10.1016/l.knosys.2011.12.004

[24] Chen RC, Cheng KF, Hsieh CF (2009) Using Rough Set and Support Vector Machine for Network Intrusion Detection. International Journal of Network Security and its Applications (IJNSA) 1-13.

[25] Sindhu SSS, Geetha S, Kannan A (2012) Decision Tree based Light Weight Intrusion Detection using a Wrapper Approach. Expert System with Applications 129-141. doi: 10.1016/j.eswa.2011.06.013.

[26] Rowayda A, Sadek M, Soliman S, Elsayed HS (2013) Effective Anomaly Intrusion Detection System based on Neural Network with Indicator Variable and Rough set Reduction. International Journal of Computer Science Issues (IJCSI) 227-233.