Intrusion Detection Rule Structure Generation Method for Mobile Ad Hoc Network

[1]M Lalli

[1]Assistant Professor, School of Computer Science, Engineering and Applications,
Bharathidasan University, Trichy, Tamilnadu, India.

Abstract : Mobile ad-hoc network security problems are the subject of in depth analysis. A group of mobile nodes area unit connected to a set wired backbone. In MANET, the node themselves implement the network management in a very cooperative fashion. The entire nodes area unit accountable to create a constellation that is dynamically, modification it and conjointly the absence of any clear network boundaries. It tends to project a completely unique intrusion detection model for mobile ad-hoc network victimization. CP-KNN (Conformal Prediction K-Nearest Neighbor) algorithmic rule is to classify the audit knowledge for anomaly detection. The non-conformity score worth is employed to cut back the classification period of time for multi-level iteration. It is effectively notice anomalies with high true positive rate, low false positive rate and high confidence that the progressive of assorted anomaly detection ways. Additionally, it is interfered by "noisy" knowledge (unclean data), the projected technique is strong, effective and conjointly it retains its smart detection performance and to avoid the abnormal activity.

IndexTerms **- Mobile Ad Hoc Network, Genetic Algorithm, Conformal Prediction for K-Nearest Neighbor (CP-KNN), Relative Reduct.**

I. INTRODUCTION

Intrusion detection system is a device, typically another separate computer, which monitor various legitimate accesses or the system abuse their privileges [1] was used to identify malicious or suspicious events. Mobile ad-hoc network is a self-configuring infrastructure less network. It is free to move independently in any direction, and will therefore change its link to its other devices frequently. The network communication was become more complex but the potential malicious outsiders who have somehow passed the screens of security controls and access controls. Prevention is although necessary, but it is not a complete computer security control; detection during an incident copes with harm that cannot be prevented in advance. But the IDS are a sensor, like a smoke detector, that raises an alarm if specific things occur.

Intrusion detection system perform various functions such as monitoring users and system activity, managing audit trails and highlighting user violation of policy or normal activity , installing and operating traps to record information about intruders but no one IDS performs all of these functions. According to this issue, many IDS techniques are used to detect various malicious activities effectively in MANETs [2].

The two general types of intrusion detection systems are classified into signature based and anomaly based IDS. 1) Signature based detection: Signature based model perform simple pattern matching and report situations that match a pattern corresponding to a known attack type. 2) Anomaly based detection: Anomaly based model was build to perform the acceptable behavior and flag exceptions are used to find out the abnormal activity. The real activity is compared against a known suspicious area. 3) Hybrid of both anomaly and misuse detection: This model is a combination of signature and anomaly based to effectively find out the malicious activity. IDS model needs some of the characteristics, fault tolerance, imperviousness to subversion, scalability, adaptability, minimal overhead, configurability, Denial of service.

Due to this lack of security controls in mobile ad-hoc networks, we must picket not only against normal attacks such as denial of service, but also against selfish attacks and other malicious attacks. Intrusion prevention can be used as a first line of defense, but this system was not sufficient to prevent it directly [3]. Intrusion detection system was used as a mechanism for representing promising security failures in the system. This is simple way to classify in order to decide whether some observed traffic data is ''normal'' or ''abnormal''. The classification objective is to minimize the probability of error and to diminish the time period to get precise classification rate.

This research work proposed a novel intrusion detection model for mobile ad-hoc network using CP-KNN (Conformal Prediction K-Nearest Neighbor) algorithm to classify the audit data for anomaly detection. It has to calculate the non-conformity score value which is used to reduce the classification time period for multi-level iteration. It is effectively detect anomalies with high true positive rate and low false positive rate of various anomaly detection methods. The proposed method is robust, effective and also it retains its good detection performance after employ the feature selection to avoid anomalous activity.

II. RELATED WORKS

Today, Intrusion detection is a mature field in mobile ad-hoc network security. Many papers focus particularly on systems based classification algorithms. The amount of work to be reported for classification-based intrusion detection in mobile ad-hoc networks is very less, but it is extensively used for wired networks.

Zhang and Lee [4] proposed the first (high-level) Intrusion Detection System (IDS) approach specific for mobile ad hoc networks. A distributed and cooperative anomaly-based IDS provides an efficient design of IDS in wireless ad hoc networks. Anomaly detection approach was based up on various routing updates on MAC layer and mobile application layer.

Huang and Lee [5] discussed about cluster based IDS which utilize a set of statistical features which was get by the routing tables and it was classified by decision tree induction algorithm C 4.5 to detect the behavior as "normal" versus "abnormal". Abdel-Fattah et al. [6] proposed the Conformal Predictor k-nearest neighbor and the Distance based an Outlier Detection (CPDOD) algorithm which was used to detect various types of malicious activities in mobile ad-hoc networks.

Deng et al. [7] proposed two distributed intrusion detection approaches, based on hierarchical and distributed architecture respectively. The intrusion detection approach used a Support Vector Machine (SVM) classification algorithm. Using network layer,

so many sets of parameters are used in distributed intrusion detection approach and that will be suggested in hierarchical distributed approach for prominent solutions.

Liu et al. [8] proposed a completely distributed anomaly detection approach. They used MAC layer data to profile the behavior of mobile nodes and then applied cross-feature analysis [9] on feature vectors constructed from the training data. The cooperative and distributed IDS utilize the various data's from MAC layer, routing values in application layers and also which was coupled with a Bayesian classifier was proposed by Bose et al [10]. Cabrera et al. [11] proposed C 4.5 training multiple classifiers, which was used to evaluate them for two types of attacks.

The true positive (TP) and false positive (FP) detection rate was demonstrated by using K-Nearest Neighbor algorithm and one-class SVM, which was proposed by authors in [12] for unsupervised anomaly detection. Especially, the performance of one class SVM algorithm was compared to the traditional supervised anomaly detection methods. But the TP and FP results are not accuracy.(98% for TP and 10% for FP [8]).

Yang Li [13] proposed TCM-KNN (Transductive Confidence Machines for K-Nearest Neighbors) machine learning algorithm which has been successfully applied to pattern recognition, fault detection and outlier detection.

The accurate value of KNN was ruined by the presence of noisy data, unrelated features and the feature scales. It is not a consistent one. More research work has been put into selecting or scaling the features to improve classification. In this research work, the proposed algorithm was used to detect anomalies with high true positive rate and low false positive rate effectively.

III. PROPOSED METHODOLOGY

3.1 Conformal Prediction for K-Nearest Neighbor (CP-KNN)

The Conformal Prediction for k-nearest neighbor (CP-KNN) algorithm was used to calculate the resemblance between new individuals and other samples in the class using the K-nearest neighbor method. We have find out the non-conformity score values for each sample and also these values are applied to find the transductive confidence. To estimate that the new samples are belongs to this particular class with p-values. The objective is, nonconformity score value is corresponds to the uncertainty of the point which was measured with respect to all other classified samples of a class with higher uncertainty, and higher nonconformity scores values. Using the Euclidean distances between points the CP-KNN nonconformity score is computed.

- The CP-KNN nonconformity score was deliberate using the Euclidean distances between points of the network parameters.
- Let, D_i^y as a sort sequence of the Euclidean distance of the point i from other points with the same classification y. The distance between i and the j[th] shortest samples in the sequences is D_{ij}^y .
- Similarly, D_i^{-y} is the distance between the sample i from the other sample with different classification, then D_{ij}^{-y} as the remoteness between i and j[th] shortest samples in the same sequence.
- α is an individual nonconformity score value which was assign to every sample The nonconformity score value of the sample i with classification y is α_{ij}.

$$\alpha_{ij} = \frac{\sum_{i=1}^{k} D_{ij}^{y}}{\sum_{i=1}^{k} D_{ij}^{-y}} \qquad (1)$$

Therefore, the quantify value of nonconformity is the ratio of the sum of the k nearest distances from the same class (y) to the sum of the k nearest distances from all other classes (-y). There are many classes in the feature space. According to the classes, the nonconformity score for the fitness of the query sample is based to the class y with respect to all others classes in the features space. Finally, the nonconformity score of a sample was increased. When the sum of the k nearest distances from the points of the related class was higher and also the sum of the k nearest distances from the other classes was smaller.

In intrusion detection, the nonconformity score was used to measure the peculiarity of an activity i belongs to the normal class y with respect to the anomalous class -y. CP-KNN algorithm used to computes the nonconformity score of m training samples in class y and arrange their nonconformity score values in sliding order. Based on the Equation (1), the algorithm can also be calculating the nonconformity score of the latest query sample v if it is classified as normal class y. Then, the p-value of the query point was compute using Equation (2), where the nonconformity score of the new unknown sample was defined as v.

$$P(\alpha_v) = \frac{\#\{i=(1,\ldots m):\ \alpha_i \geq \alpha_v\}}{m+1} \qquad (2)$$

The entire training points are independent random samples. The strength of the indication against v was belongs to class y, where i is the number of class members with larger nonconformity score values. If the value is larger than the p-value then it shows how likely the query point is to be classified as y, by referring to the distribution of all points in the same class. The smaller the p-value the more improbable query point was belongs to class y.

3.2 Conformal Prediction for K-Nearest Neighbor (CP-KNN) Classification Algorithm

In this research work, the proposed CP-KNN which is used to classify the audit data for anomaly detection. Conformal prediction is to determine the precise levels of confidence in new predictions. The error probability is defined as ∈. The error probability and the method both are combine together and make a point prediction of a label y. It produces a set of labels that typically containing the point prediction is also contains y with probability 1-∈. Conformal prediction is a method for producing point predictions such as nearest neighbor method. In this method K depends upon the data; a larger values of *k* will reduce the effect of a noise on the classification, but it will make the boundaries between classes with less distinct. The class which was predicted to be the nearby class of the training sample (*k* = 1) is a nearest neighbor The CPKNN classification process is shown in Figure 1.

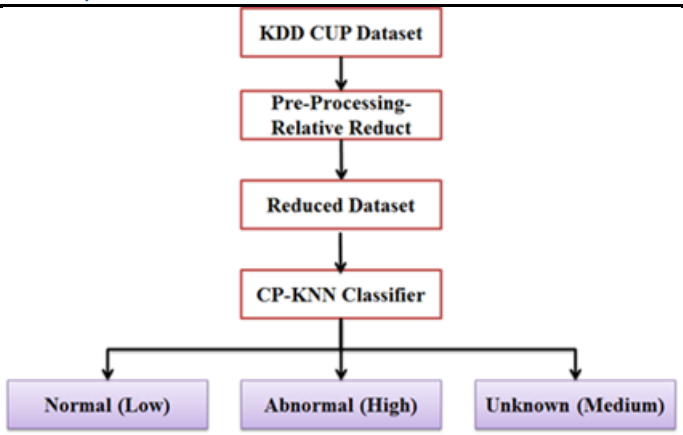

Figure 1: Framework for the proposed CP-KNN

The classification process consists of three stages such as network data collection, preprocessing and CP-KNN classification. The data collection stage provides the data for various network activities and also the number of features needs to be selected to represent the ad-hoc network activity which can be used to detect various attacks. The initial subset features selection is part of the preprocessing, which is implemented by feature selection algorithm. Different feature selection corresponds to different type of attacks. Depends upon the CP-KNN classification rate we have to detect ‘‘normal’’ versus ‘‘abnormal’’ behaviors. This will give higher performance when we removed the irrelevant features. The overview of CP-KNN design architecture is shown in Figure 2.

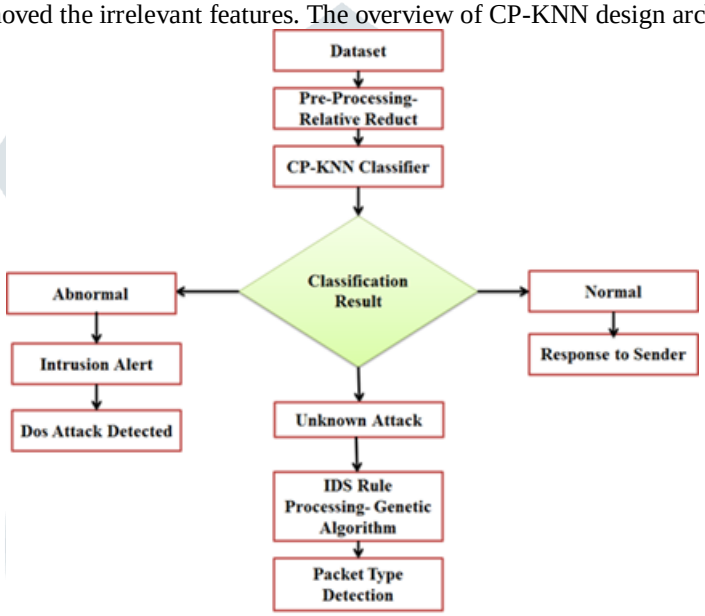

Figure 2: Design Architecture of proposed CP-KNN with Genetic Algorithm

3.3.2.1 Relative Reduct

Relative Reduct Algorithm is the most understood calculation for feature selection utilizing Rough sets [14][15]. This is an incremental methodology; where it begins with a void set and in every stride a feature is added to the Reduct, in such way that dependency quantifies increments. The methodology stops when the dependency measure of the arrangement of elements being considered is equivalent to the dependency measure utilizing all the conditional features. The calculation endeavors to figure a reduct without comprehensively producing every single conceivable subset [15]. Its pseudo code calculation is given underneath:

Input: Original Dataset, D the set of all conditional features; S- the set of decision features, a reduct is defined as Q subset.

Step 1: Begin
Step 2: Initialize Q as Empty set (is represented by {})
Step 3: R ← Q
Step 4: ∀χ ∈ (D - Q)
Step 5: if $\gamma\_(Q \cup \{ x \}) (S) > \gamma q (S)$
Step 6: R ← Q ∪ { x }
Step 7: Q ← R
Step 8: Until $\gamma q (S) = \gamma d (S)$
Step 9: Return Q
Step 10: End

Output: A Reduct Dataset

The Reduct Relative algorithm endeavors to ascertain a reduct without completely creating every single possible subset. It begins off with a vacant set and includes turn, each one in turn, those features that outcome in the best increment in the rough set dependency metric, until this delivers its most extreme conceivable quality for the dataset.

3.3.2.2 CP-KNN Algorithm

Step 1: for i=1 to m d
Step 2: $\sum D_i^y \sum D_i^{-y}$ and store
Step 3: end for
Step 4: Calculate CP(α) for all m and store
Step 5: for i=1 to r do

Step 6: $\sum dist(t) \geq t$
Step 7: for j=1 to c do
Step 8: for $\sum t$ classified as j do
Step 9: if $D_{tk}^{j} > dist(t) \rightarrow \alpha$ at a
Step 10: point t
Step 11: end for
Step 12:$\alpha \rightarrow \alpha_{new}\ classified\ as\ j$
Step 13: $p \rightarrow p_{new}\ classified\ as\ j$
Step 14: end for

3.3.3 Intrusion Detection System Rule Generation by using Genetic Algorithm

Genetic Algorithm is an intelligent probabilistic search algorithm which can be applied to a variety of combinational optimization problems. Theoretical foundations of Genetic Algorithm were initially developed by Holland in 1970's. The inspiration of GA is based on the evolutionary process of biological organisms in nature. During the course of evolution, natural population evolves according to the principle of natural selection and survival of the fittest. Individuals who are easily adaptable to all environmental conditions and have higher fitness are more likely to reproduce and generate offspring while lower fitness individuals are eliminated from population [16]. The combination of good characteristics from highly adaptive ancestors may produce even more fit offspring. In this way, species evolve more and more to become well adapted on environment.

A Genetic Algorithm stimulates these processes by taking an initial population of individuals and applying GA operators in each generation. Each individual is encoded as a chromosome which is a solution to the problem. A chromosome is a collection of genes, means an individual is made up of genes. The fitness of each individual is calculated by objective function. Highly fit individuals are given chances for reproduction, in crossover procedure. Mutation is optional for changing some of genes in individual to avoid duplicity. This evolution, selection, crossover process repeated until the condition is fulfilled.

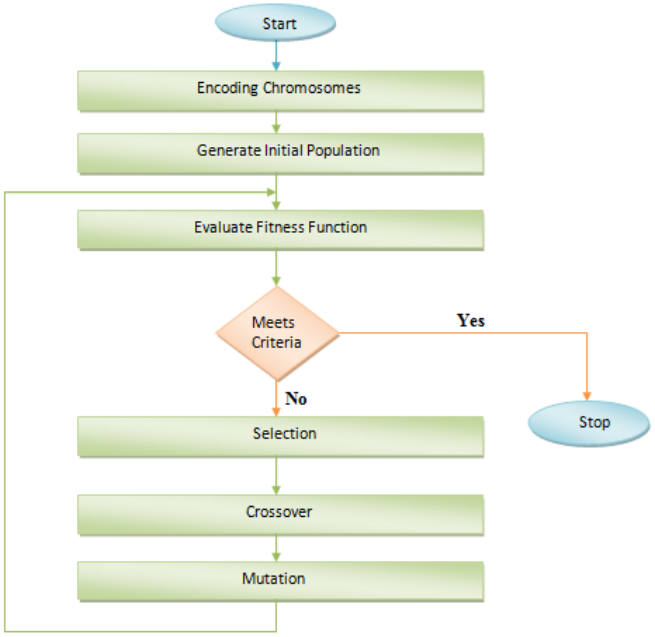

Figure 3: Flow of Genetic Algorithm

3.3.3.1 Encoding of a Chromosome

The chromosome should be encoded in such a way that it must represent information about the solution. The most commonly used way to encode chromosome is in binary string. Each bit represents some information about solution. Every chromosome is a collection of genes where each gene represents each bit of chromosome.

Table 1: Representation of the Chromosome

Chromosome 1	1110001101011010
Chromosome 2	1101010101110100

Pseudo Code of Genetic Algorithm

```
BEGIN
        Initialize population with random candidate Solutions
        Evaluate each candidate
        Repeat until (Termination condition is satisfied)
        DO
                Select parents
                Recombine pair of parents
                Mutate the resulting offspring
                Evaluate new candidate
                Select individuals for next generation
        END DO
END
```

Here the use of genetic algorithm is divided into two main sections: the pre-calculation section and also the detection section. In that pre-calculation section, collection of chromosome is created by set of training data. That chromosome sets are utilized in the next section for the purpose of comparison. The primary steps in pre-calculation as

Algorithm 1: Initialize chromosomes for evaluation
Input: Reduced Dataset (for training)
Output: A collection of chromosomes
Step 1: Range = 0.125
Step 2: For every training information
Step 3: If it has any nearest neighbor chromosome within Range
Step 4: Combine it with the adjacent chromosome
Step 5: Else
Step 6: Generate a new chromosome
Step 7: End if
Step 8: End for

3.3.3.2 Genetic Algorithm Operators

The basic operations used in Genetic Algorithm are selection, crossover and mutation. Performance of Genetic Algorithm is dependent on these operators. Selection and crossover affects more on performance while mutation impact is light.

- **Selection**: In selection, chromosomes are given a probability of being selected that is directly proportional to their fitness. Higher the fitness, more the chances for generating offspring. For example, if we have four chromosomes of certain fitness and only two are allowed in next generation then chromosomes with highest fitness are allowed to mate to generate new offspring's (in table 2) only chromosome 1 and chromosome 3 are allowed for crossover because they have higher fitness than Chromosome 2 and Chromosome 4.

Table 2: Individuals with Their Fitness

Individual	Encoding	Fitness Value
Chromosome 1	1110001101011010	0.6
Chromosome 2	1101010101110100	0.5
Chromosome 3	1010110101111001	0.7
Chromosome 4	1011011011010101	0.2

- **Crossover:** Crossover selects genes from parent chromosome and generates a new offspring [18]. We can select any crossover point. In example given below, we have two parents Parent 1 and Parent 2 and crossover point is 3rd bit. In the child's chromosome representation, we can see that Child 1 has 3 bits of Parent1 and 5 bits of Parent 2 .Similarly, Child2 has 3 bits of Parent 2 and 5 bits of Parent 1. Crossover point affects the performance of Genetic Algorithm.

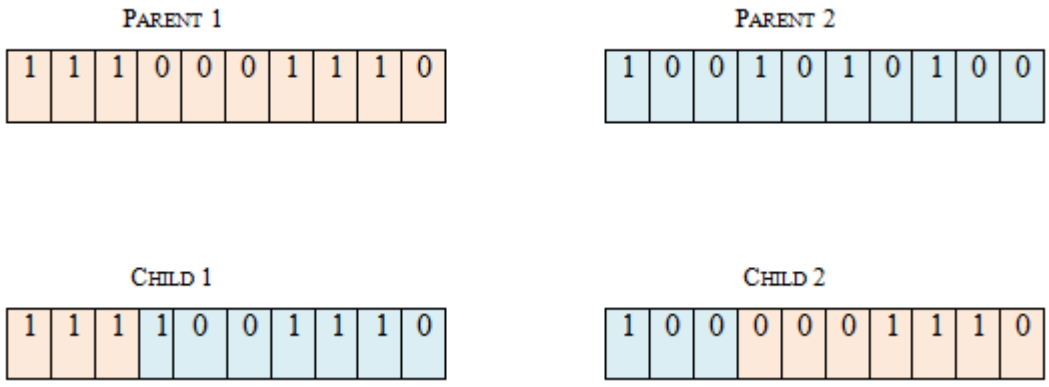

Figure 4: Crossover at 3rd point

- **Mutation**: After selection and crossover, we have a set of new individuals [17]. In order to ensure that all individuals are not exactly the same, mutation is done on individuals. Mutation is a slight change in encoding of chromosome. It just changes one or two bits in the chromosome. Random bit position is chosen for mutation, if bit is 1, set it to 0 and vice versa. In the example below, mutation is
  done on 5th bit position. Mutation rate should be 0.01 to 0.02.

Figure 5: Mutation at 5th point

There are three basic parameters of Genetic Algorithm namely, crossover probability, mutation probability and population size.

- **Crossover Probability**: Crossover Probability tells about the expectation of crossover to be performed. If there is 0% crossover probability then in next generation all individuals are exactly the same as parents. If there is 100% probability then all chromosomes are new in next generation means all parent undergo crossover.
- **Mutation Probability**: Mutation Probability tells about the change in chromosome. If mutation probability of a chromosome is 100% then all bits of chromosome have been changed and if it is 0% then no bit has been changed.

- **Population Size**: Population size tells us about the number of chromosomes in the population. If there is less number of chromosomes then possibility of crossover and mutation is less and if numbers of chromosomes are more, then possibility also increases but after a certain population, it does not affect the possibility and slows down the system.

The second one is a detection section, where a population is generated for a test data and going through some analysis process (selection, crossover, mutation) and also the type of test data is predicted. The pre-calculated set of chromosome is employed during this section and finds the fitness of each chromosome of entire population.

3.3.3.3 Designing Rules for Intrusion Detection

Every rule for intrusion detection is simple if-then clause. Suppose we have n features namely, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9…….an in a dataset and this dataset contains classes namely, c1, c2, c3…. cn. For rule designing we can use n features or subset of n features according to our requirement. Suppose all attributes contain numeric values and if we are using three features of this dataset then rule can be designed as:

(a_1 =1 && a_2 =2 && a_3=3) then c_1

where, a_1, a_2, a_3 are attributes of dataset and c_1 is class of the record.

3.3.3.4 Calculating Fitness Function of a Rule

To determine a fitness value of each rule, the following fitness functions can be used.

$$fitness = \frac{\alpha}{A} - \frac{\beta}{B} \quad (4.1)$$

$$fitness = w1 * support + w2 * Confidence \quad (4.2)$$

where, support=|A and B|/N and confidence =|A and B|/|A|

In fitness function (4.1), α is the number of correctly detected attacks, A is the total number of attacks in the training dataset, β is the number of normal connections incorrectly characterized as attacks, i.e. false-positives, and B is the total number of normal connections in the training dataset [19]. Scale of fitness values is [-1, 1], where -1 is the lowest and 1 the highest value. High detection rate and low rate of false-positives result in a high fitness value. On the other side, low detection rate and high rate of false-positives result in a low fitness value.

In fitness function (2) of each rule , where N is the total number of network connections in the training dataset, |A| stands for the number of connections matching the condition A, and |A and B| stands for the number of connections that matches the rule if A then B . The weights w1and w2 are used to control the balance between the two terms of a rule [20].

Algorithm 2: Envisage the data or intrusion type (using Genetic Algorithm)

Input: Review reduced dataset (for testing), Pre-calculated set of chromosomes.
Output: Various type of information
Step 1: Set the overall population
Step 2: Crossover Rate equal to the range of 0.15, Mutation Rate equal to the range of 0.35.
Step 3: While the number of generation is not attained
Step 4: For every chromosome in overall population
Step 5: For every pre-calculated chromosome
Step 6: Get the fitness
Step 7: End for
Step 8: Assign optimum fitness. The fitness of that chromosome
Step 9: End for
Step 10: Eradicate some chromosomes with worse fitness
Step 11: Apply crossover to the chosen pair of chromosomes of the population.
Step 12: Apply mutation to each and every chromosome of the population.
Step 13: End while

IV. EXPERIMENTAL RESULT AND DISCUSSION

The following table 3 represents the result obtained in the pre-processing step by using Relative Reduct algorithm as the feature selection technique. From the 41 features only 16 features are obtained by Relative Reduct algorithm.

Table 3: **Number of features obtained by using Relative Reduct and Information Gain Feature Selection techniques in the Pre-processing step**

Feature Index	Relative Reduct	Information Gain Technique
1	Duration	Duration
2	Service	protocol_type
3	protocol_type	Service
4	Flag	Flag
5	src_bytes	src_bytes
6	dst_bytes	dst_bytes
7	Same_srv_rate	Count
8	Srv_Count	srv_count
9	Dst_host_diff_Serv_rate	serror_rate
10	Dst_host_srv_error_rate	srv_serror_rate
11	Wrong_fragement	same_srv_rate
12	dst_host_same_srv_rate	diff_srv_rate
13	dst_host_diff_srv_rate	srv_diff_host_rate
14	dst_host_srv_serror_rate	dst_host_count
15	srv_diff_host_rate	dst_host_srv_count
16	dst_host_rerror_rate	dst_host_same_srv_rate

17		dst_host_diff_srv_rate
18		dst_host_srv_serror_rate
19		dst_host_rerror_rate
20		dst_host_srv_rerror_rate

The proposed CP-KNN is reducing the classification time period for multi-level iteration. The dissimilar features are removed to get the higher performance. It is interfered by "noisy" data (unclean data), the proposed method is robust, effective and also it retains its good detection performance after employ the feature selection to avoid anomalous activity. The Classifier Rate (P) is calculated as,

Classifier Rate (P) = (Number of data classified / Total Number of Data)* Iteration Level

The Iteration level is defined as the classifier classified the data with number of various levels.

Table 4: Performance Analysis of the Proposed CP-KNN and KNN algorithm

Performance Metrics	**Relative Reduct + Proposed CP-KNN**	**Relative Reduct+ Existing KNN Algorithm**
Correctly Classified Instance	97	95.0
Kappa Statistic	0.68	0.50
Mean Absolute Error	0.21	0.25
Root Mean Squared Error	0.31	0.40
Relative Absolute Error	45.50	50.75
Root Relative Absolute Error	75.25	85.62
True Positive Rate	0.81	0.69
False Positive Rate	0.21	0.30
Precision	0.81	0.63
Recall	0.81	0.69
Receiver Operating Characteristic Curve	0.84	0.73

Table 5: Classifier rate for CP-KNN and KNN based on number of iterations

Number of Iterations	**Classifier Rate (in %)**	
	Proposed CP-KNN	KNN
1`	99.8	98
2	99.0	97.5
3	98.9	96.9
4	98.4	96.4
5	97.8	95.8
6	97.4	95.4
7	97	95

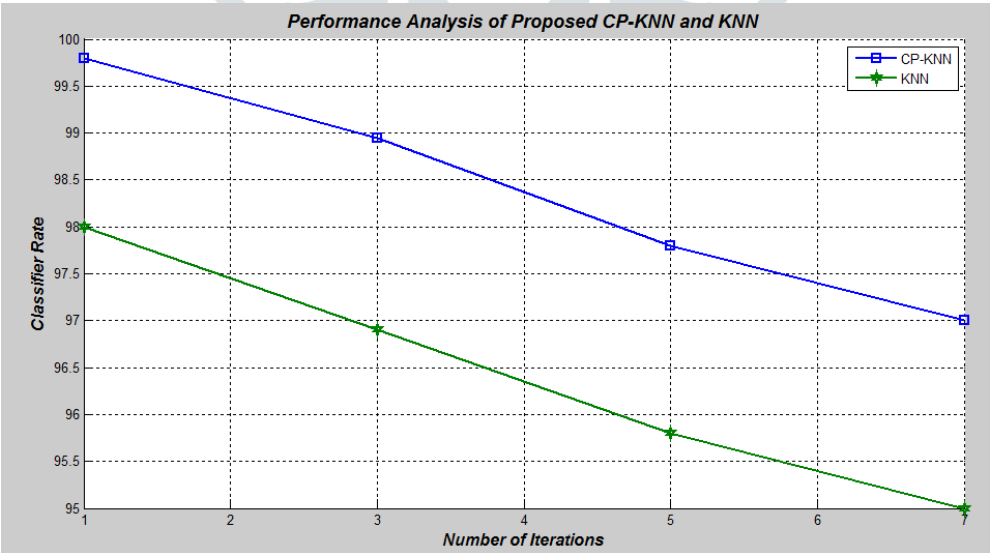

Figure 6: Classifier Rate for CP-KNN vs KNN

Table 6: Execution Time of proposed CP-KNN and KNN on Number of iterations

Number of Iterations	Execution time (in ms)	
	Proposed CP-KNN	KNN
2	0.01	0.2
3	0.1	0.4
4	0.2	0.5
5	0.3	0.7
6	0.37	0.81
7	0.4	0.82

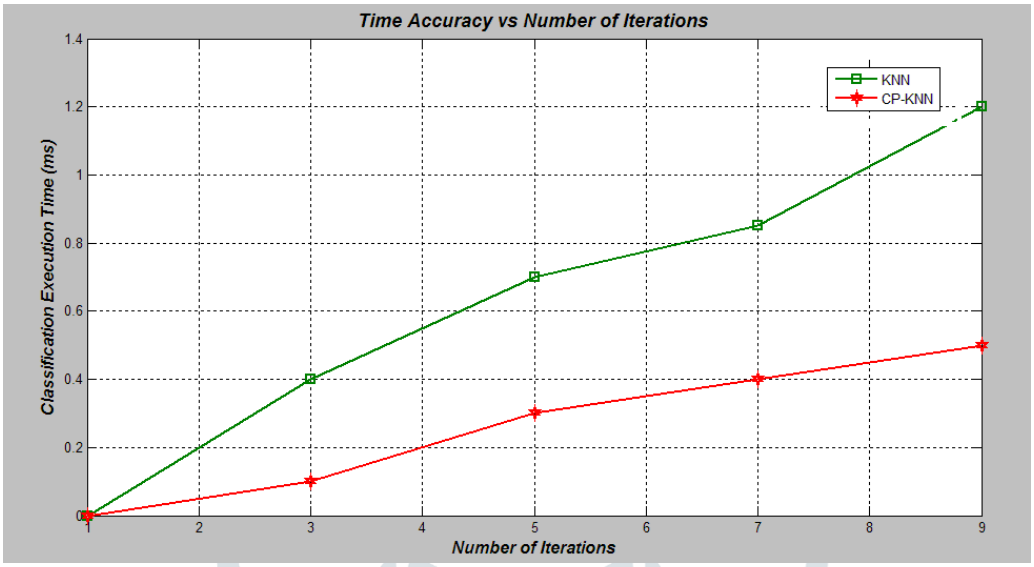

Figure 7: Time Accuracy Graph for CP-KNN and KNN

In Figure 7, the graph is based on the classifier that makes the time consumption for its classification at multi-level iteration. In that experiment, the accuracy for K-Nearest Neighbor is 90% and Conformal Prediction K-Nearest Neighbor accuracy is 94%.

V. CONCLUSION

Due to this vulnerability of ad hoc networks, the intrusion anticipation measures such as encryption and authentication. They are used to reduce the various types of intrusions, however they cannot be totally eliminate them. For this reason, researchers necessitate intrusion detection; it will be act as a frontline of security in mobile ad- hoc networks. In this research work, a novel intrusion detection model for mobile ad-hoc network using CP-KNN (Conformal Prediction K-Nearest Neighbor) algorithm was proposed to classify the packet data for anomaly detection. The nonconformity score value is used to reduce the classification time period for multi-level iteration. The proposed work was effectively detecting various anomalies with high true positive rate, low false positive rate and high confidence rate. In addition it is interfered by "noisy" data (unclean data), the proposed method is robust, effective and also it retains its good detection performance to avoid anomalous activity.

REFERENCES

[1] Abarna Sri.R, Lalli. M, "NIDS in Manet Using KMCA" International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 9, 2013.

[2] Farhan Abdel-Fattah, Zulkhairi Md. Dahalin, Shaidah Jusoh, "Distributed and Cooperative Hierarchical Intrusion Detection on MANETs", International Journal of Computer Applications, Volume 12-No.5, December 2010.

[3] Aikaterini Mitrokotsa, Christos Dimitrakakis, "Intrusion Detection in MANET using Classification algorithms: The effects of cost and model selection", Elsevier- Ad Hoc Networks 2012.

[4] Y. Zhang, W. Lee, Y. Huang, Intrusion detection techniques for mobile wireless networks, Wireless Networks 9 (5) (2003) 545–556.

[5] Y. Huang, W. Lee, A cooperative Intrusion detection system for ad hoc networks, in: Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03), Fairfax, VA, USA, 2003, pp. 135–147.

[6] F. Abdel-Fattah, Z. Md. Dahalin, S. Jusoh, Dynamic intrusion detection method for mobile ad hoc networks using CPDOD algorithm, International Journal of Computer Applications, vol. 2, Published by the Foundation of Computer Science, 2010. pp. 22–29.

[7] H. Deng, Q. Zeng, D.P. Agrawal, SVM-based intrusion detection system for wireless ad hoc networks, in: Proceedings of the 58th IEEE Vehicular Technology Conference (VTC'03), vol. 3, Orlando, FL, USA, 6–9 October 2003, pp. 2147–2151.

[8] Y. Liu, Y. Li, H. Man, Mac layer anomaly detection in ad hoc networks, in: Proceedings of the 6th Annual IEEE SMC Information Assurance Workshop (IAW '05), West Point, NY, USA, 15–17 June 2005, pp. 402–409.

[9] Y. Huang, W. Fan, W. Lee, P. Yu, Cross-feature analysis for detecting ad-hoc routing anomalies, in: Proceedings of the 23rd International Conference on Distributed Computing Systems, Rhode Island, USA, 2003, p. 478.

[10] S. Bose, S. Bharathimurugan, A. Kannan, Multi-layer intergraded anomaly intrusion detection for mobile ad hoc networks, in: Proceedings of the IEEE International Conference on Signal Processing, Communications and Networking (ICSCN 2007), February 2007, pp. 360–365.

[11] J.B.D. Cabrera, C. Gutiérrez, R.K. Mehra, Ensemble methods for anomaly detection and distributed intrusion detection in mobile adhoc networks, In Information Fusion 9 (2008) 96–119.

[12] D. Djenouri, O. Mahmoudi, M. Bouamama, D.L. Llewellyn-Jones, M. Merabti, On securing manet routing protocol against control packet dropping, in: Proceedings of the IEEE International Conference on Pervasive Services (ICPS '07), Istanbul, Turkey, 15-20 July 2007, pp. 100–108.
[13] Yang Li, Binxing Fang, Li Guo,You Chen, "Network Anomaly Detection Based on TCM KNN Algorithm", ASIACCS'07, March 20–22, 2007, Singapore.
[14] Xiuyi Jia, Lin Shang, Bing Zhou, Yiyu Yao, "Generalized Attribute Reduct in Rough Set Theory", Three-way Decisions and Granular Computing, Knowledge Based System- Elsevier, Volume 91, January 2016, pp. 204-218.
[15] Jun Wang, Jiaxu Peng, Ou Liu, "A Classification Approach for Less Popular Webpages based on Latent Semantic Analysis and Rough Set Model", Expert Systems with Applications - Elsevier, Volume 42, Issue 1, January 2015, pp. 642-648.
[16] Beasley, John E., and Paul C. Chu. "A genetic algorithm for the set covering problem." European Journal of Operational Research 94, no. 2 (1996): 392- 404.
[17] Tutorial available on http://geneticalgorithms.aidepot.com/Tutorial/Overview.html.
[18] Tutorial available on http://www.obitko.com/tutorials/geneticalgorithms/operators.php.
[19] Hashemi, V. Moraveji, Z. Muda, and W. Yassin. "Improving Intrusion Detection Using Genetic Algorithm." Information Technology Journal 12, no.5 (2013).
[20] Lu, Wei, and Issa Traore. "Detecting new forms of network intrusion using genetic programming." Computational Intelligence 20, no. 3 (2004): 475-494