FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 4 No. 4, December, 2020, pp 454 - 459
DOI: https://doi.org/10.33003/fjs-2020-0404-502

MODELLING OF AN INTRUSION DETECTION SYSTEM USING C4.5 MACHINE LEARNING ALGORITHM

[*]OLANREWAJU Oyenike Mary, ECHOBU Faith Oluwatosin, MOGAJI Abubakar

Department of Computer Science & Information Technology, Federal University Dutsin-Ma, Katsin-Ma State - Nigeria

[*]Corresponding Authors Email: oolanrewaju@fudutsinma.edu.ng

ABSTRACT

The increasing growth of wireless networking and new mobile computing devices has caused boundaries between trusted and malicious users to be blurred. The shift in security priorities from the network perimeter to information protection and user resources security is an open area for research which is concerned with the protection of user information's confidentiality, integrity and availability. Intrusion detection systems are programs or software applications embedded in sophisticated devices to monitor the activities on networks or systems for security, policy or protocol violation or malicious activities detection. In this work, an intrusion detection model was proposed using C4.5 algorithm which was implemented with WEKA tool and RAPID MINER. The model showed good performance when trained and tested with validation techniques. Implementation of the proposed model was conducted on the Network Security Laboratory Knowledge Discovery in Databases (NSL-KDD) dataset, an improved version of KDD 99 dataset, which showed that the proposed model approach has an average detection rate of 99.62% and reduced false alarm rate of 0.38%.

Keywords: Intrusion Detection, Model, NSL-KDD, Wireless Network

INTRODUCTION

Intrusion Detection Systems (IDS) are tools used to monitor and analyse events occurring on computer systems or networks for traces of possible illegal action. Among others, these could include threat of violation of computer or network security policies and possible security attacks whether they are malicious or not. If found malicious, necessary actions to recover the system are then taken.

Usually, intrusion is aimed at collecting information from an organization such as the structure of their internal networks, software systems, tools or utilities, then find a way into the internal network and launch attacks (Nayak and Rao, 2014). Intrusions are most often than not initiated by attackers outside the organisation. Sometimes however, intrusions can be initiated by authorized persons within the organisation (Malicious insider) misusing their authorization by accessing data or information that are beyond their area of authorization.

The fast rise of various network proliferations, data transfer rates and unpredictable internet usage have led to a record of more malicious activities and cyber-attacks. Increase in cyber-security attacks, has made it important to develop more reliable, effective, and self-monitoring intrusion detection systems that can carry out their activities without human interactions so as to reduce attacks on susceptible systems.

With machine learning, computers are able to learn and improve from experience without explicit programming. The process of learning begins with observation of patterns in data and thereby making better predictions based on the samples provided. The main aim is to allow the system learn based on instances given to it and adjust actions accordingly.

REVIEW OF RELATED WORKS

Crosby, Hester, & Niki (2011) used a location-aware trust based system to detect and isolate compromised nodes in wireless sensor networks. Reputation and trust were first developed amongst devices so that each device on a wireless sensor network could determine whether other devices had been compromised or not. If compromised, appropriate corrective actions were taken using negative information sharing and independent trust-based decision making.

Dhopte and Tarapore (2012) worked with fuzzy class association rule mining based on genetic algorithm to design an intrusion detection system which proved to be more accurate when compared with the use of Cross Industry Standard Process Data Mining (CRISP-DM) Approach. The system which could be applied to both misuse and anomaly intrusion detection used the KDD '99 cup dataset to train and test the system while fuzzy set theory was combined with association rule mining algorithm to extract the rules with attributes of continuous value. Their proposed system was able to avoid sharp boundary problem, achieve increased detection rate and detection accuracy.

Balaji and Kumar (2013) in their work used DARPA dataset to train and test a proposed hybrid model based on fuzzy logic and data mining, which could detect both misuse and anomaly attacks. The proposed system aimed at reducing the amount of data retained for processing and improve detection rates using techniques like Kuok fuzzy data mining algorithm technique which is a modified version of APRIORI algorithm, for implementing fuzzy rules. Fuzzy inference engine was applied using mamdani inference mechanism with three variable inputs for faster decision making. The proposed model however has issues with packet processing, and very high false positive rates.

Neural network approach is a commonly used technique in machine learning for resolving a complex practical problems and has been successfully applied to detection of anomalies in intrusion detection systems. However, the main drawbacks of neural network based intrusion detection system are low detection precision, particularly for low frequent attack such as user to root and remote to local with weaker detection stability.

An anomaly detection using an outlier detection approach was presented by Jabez and Muthukumar (2015). In their work, the features of a dataset culled from the internet were extracted from the data packet and forwarded to the IDS to the distance or deviation between the extracted features and the trained model. If the outlier value is greater than the specified threshold, a false alarm was generated.

Liu and Zhu (2019) made use of back propagation network, an iterative gradient algorithm for solving minimum mean square errors between the actual output and the expected output of the feed-forward network, for their proposed intrusion detection and alarm correlation system. The BP network is a multi-layer mapping network that transmits backwards and corrects errors. Their proposed system was modelled using multi-layer forward network which makes uses of nine neurons in the input layer and only one neuron in the output layer because there were only two output results which is either normal or attack.

MATERIALS AND METHODS

DATASET

NSL-KDD dataset is an improvement on the KDD’99 data developed at the Network Security Lab of the University of New Brunswick. The dataset eliminated redundancies of the previous west to get rid of biased classification results. Although several versions of this dataset exist with varying number of instances, there are always 42 attributes. The training dataset is known as KDDTrain+_20Percent has a total of 25192 instances while the test data set, KDDTest+_20Percent, has 22544 instances.

MACHINE LEARNING TOOLS

a) Waikato Environment for Knowledge Analysis (WEKA) is a suite of machine learning tools with a collection of visualization tools and algorithms for data analysis and predictive modelling (Holmes, Donkin, & Witten, 2002).

b) Rapid Miner is a versatile data science software that provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive analytics. Asides business and commercial applications, it is also used for research, education, training, rapid prototyping, and application development and supports all steps of the machine learning process including data preparation, results visualization, model validation and optimization (Hofmann and Klinkenberg, 2014).

c) C4.5 Algorithm is often referred to as a statistical classifier and builds decision trees from a set of training data using the concept of information entropy. The training data is a set $S_i = S_1, S_{2,} S_3 \dots$ of already classified samples. Each sample$S_i$consists of a p-dimensional vector$x_j = x_{1,i}, x_{2,i,} x_{3,i}, \dots, x_{p,i}$ where the$x_j$represents attribute values or features of the sample, as well as the class in which $S_i$falls.

At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy) and the attribute with the highest normalized information gain is chosen to make the decision.

THE PROPOSED INTRUSION DETECTION SYSTEM

The algorithm for the proposed IDS is presented below:

```
i.    Select an attribute from a set of training instances;
ii.   Select an initial subset of the training instances;
iii.  Build a decision tree using the selected attribute and the subset of instance;
iv.   Test the accuracy of the constructed tree with the remaining training instances
v.    If all instances are correctly classified – stop
vi.   If an instance(s) is incorrectly classified, add it to the initial subset and construct a new tree;
vii.  Iterate until a tree is built that correctly classifies all instances OR a tree is built from the entire training
      set
```

The conceptual view of the proposed system is represented in Figure 1. The major components are Network model for traffic generation, intrusion detection system and the prediction component.

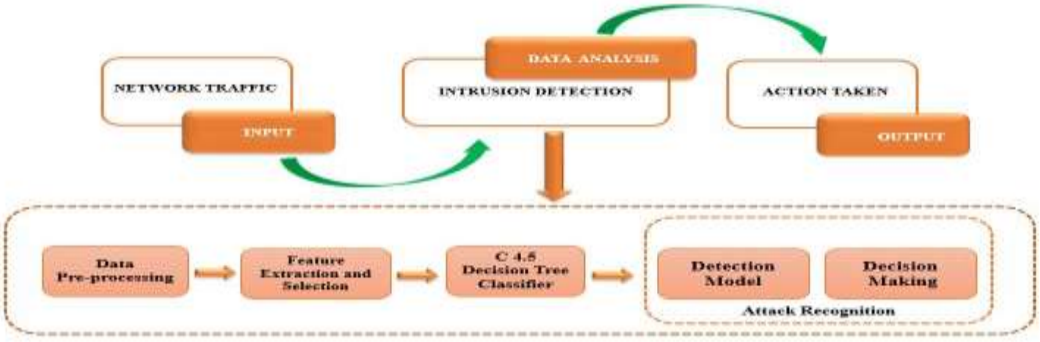


Figure 1: Conceptual Design of the Proposed Model

Figure 2 describes the structure of the system components and the relationship among these components. This relationship describes the interconnection, communication and interaction as regards data flow, processes and interfaces within the system.

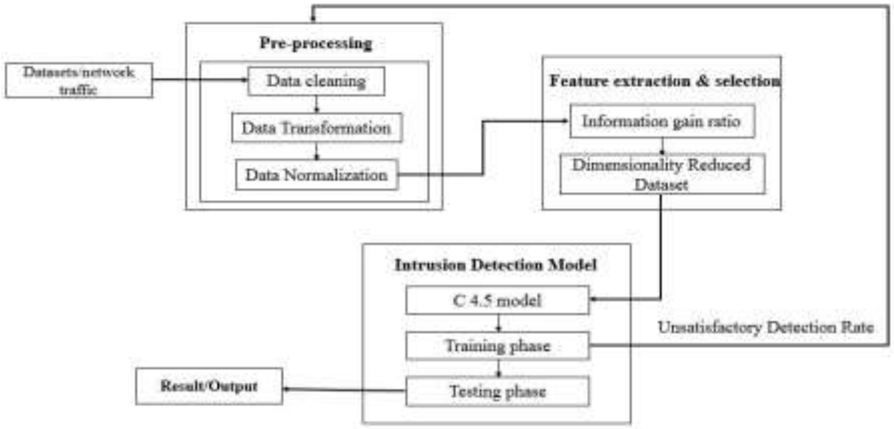


Figure 2: The Architectural Design of the Proposed Intrusion Detection System

Five (5) classifications of the NSL-KDD dataset were used as shown in Table1. Denial of Service (DoS), Unauthorized Access to Local Supervisor Privileges (User to Root (U2R)), Remote to Local (R2L), Probe and Normal (this is the category of network traffics that are free of attack from both insider (internal factor), or outsider (external factor)).

Table1: Summary of 20% of NSL-KDD Dataset

Dataset	Records	DoS Class	U2R Class	R2L Class	Probe Class	Normal Class
KDD Train+ 20%	25192	9234	11	209	2289	13449
		36.65%	0.04%	0.83%	9.09%	53.39%
KDD Test+ 20%	22544	7458	200	2754	2421	9711
		33.08%	0.89%	12.22%	10.74%	43.08%

There are forty-two (42) features or attributes of the NSL-KDD dataset but the following twenty (20) are mostly used considered key:

Table 2: Top Twenty (20) Most Important Features of NSL-KDD Dataset

Rank	Attribute	Attribute number in the KDD set	Description
1	Service	3	Different types of services provided such as http, ftp, smtp, telnet and others
2	Dst_bytes	6	The number of bytes accepted in one connection
3	Dst_host_diff_srv_rate	35	Percentage of connections that exist for different services among connections in dst_host_count
4	Diff_srv_rate	30	Percentage of connections that exist for different services among connections in count
5	Flag	4	Connection status. Possible statuses are SF, S0, S1, S2, S3, OTH, REJ, RSTO, RSTOSO, SH, RSTRH, SHR
6	Dst_host_serror_rate	38	Percentage of connections that activated the flag s0, s1, s2 or s3 among connections in dst_host_count
7	Dst_host_srv_count	33	Total connections to specific destination port
8	Same_srv_rate	29	Percentage of connections that exist for the same service among connections in count
9	Count	23	Sum of connections to specific destinations
10	Dst_host_same_srv_rate	34	Percentage of connections to the same service
11	Dst_host_srv_serror_rate	39	Percentage of connections that activated the flag
12	Serror_rate	25	Percentage of connections that activated the flag s0, s1, s2 and s3 among connections in count
13	Src_bytes	5	The number of bytes sent in one connection
14	Dst_host_srv_dff_host_rate	37	Percentage of connections to various destinations
15	Srv_serror_rate	26	Percentage of connections that activated the flag s0, s1, s2 and s3 among connections in srv_count
16	Dst_host_same_src_port_rate	36	Percentage of connections to the same source port
17	Logged_in	12	If login value is correct then assign 1 else 0
18	Dst_host_Count	32	Total connections to specific IP
19	Hot	10	Total number of hot connections
20	Dst_host_rerror_rate	40	Percentage of connections that activated the flag among connections in dst_host_count

Performance Evaluation Metrics

Accuracy and efficiency of the proposed model was measured using the following metrics:

a) Kappa Statistics is a measure of relationship between classification and true classes. The possible values for Kappa lies between interval such as [-1, 1]. It usually falls between the range of 0 and 1. The value of attribute which is 1 means perfect agreement. A Kappa with negative value indicates worse agreement than that expected by chance;

b) Relative Absolute Error is used to measure how close a classification is to the eventual outcome;

c) Root Mean Square Error is used to measure a model's accuracy;

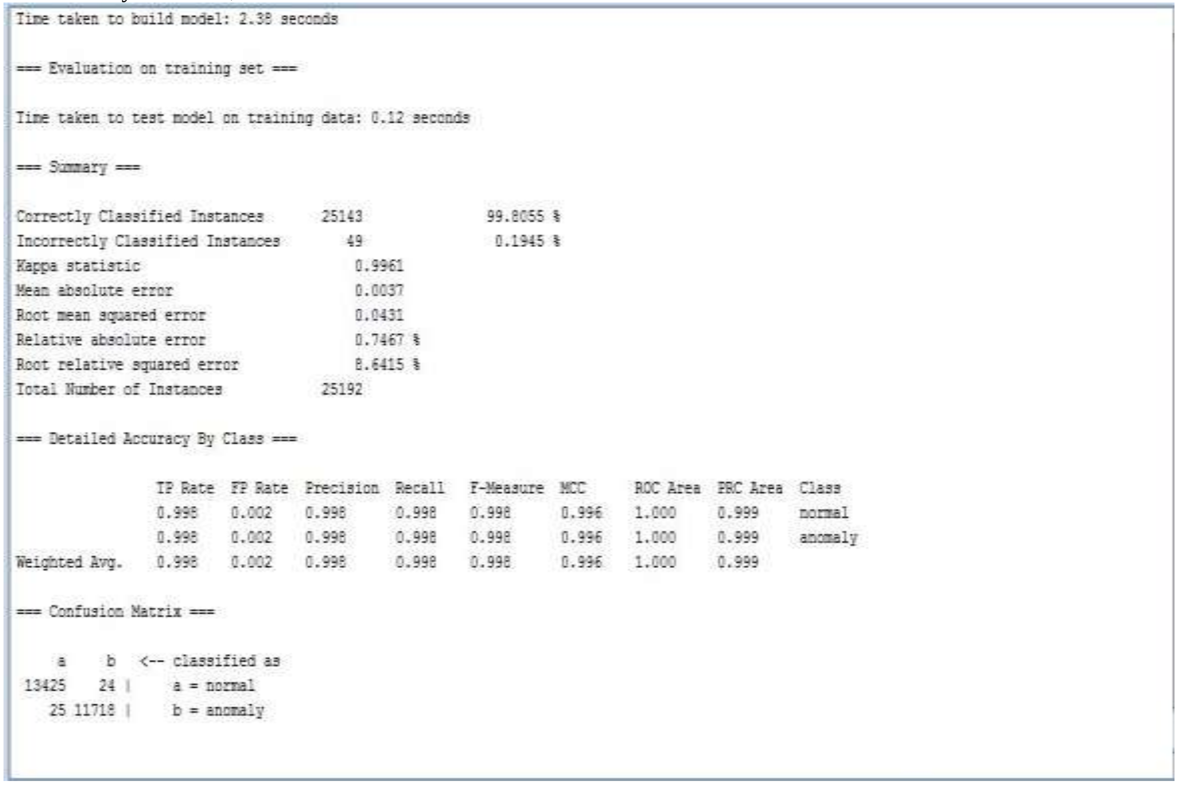
d) Root Relative Square Error is the average of the actual values;

e) Confusion Matrix or Contingency Table is the representation of the number of instances correctly or incorrectly classified;

f) Receiver Operating Characteristics (ROC) curve is used to compare two classification models. A ROC curve for a given model shows the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR).

RESULT PRESENTATION AND INTERPRETATION

The results of the evaluation metrics used during training of the model is shown in Figure 3. 99.8 percent instances were correctly classified. The summary results for both training and testing of the model is presented in Table 4.



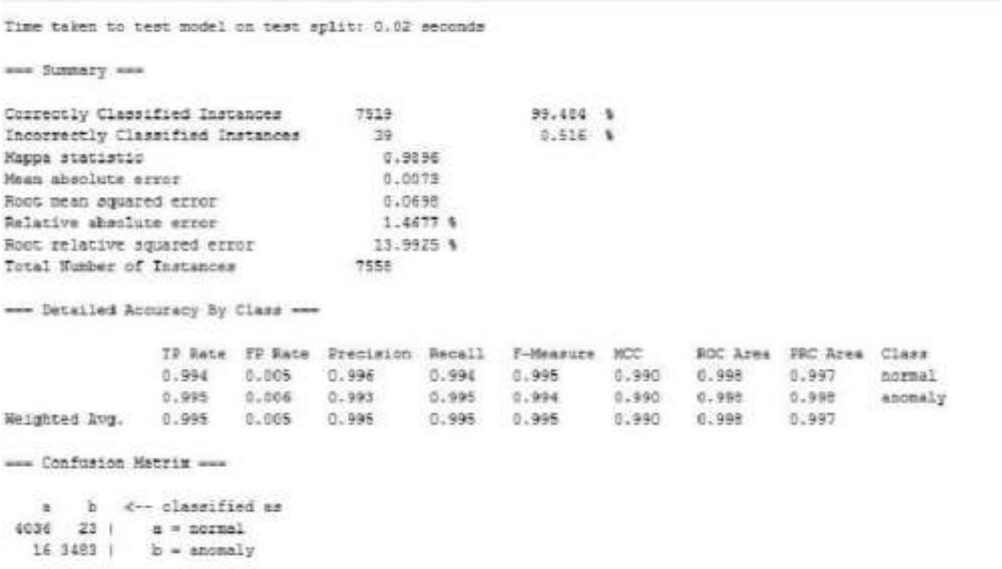
```
Time taken to build model: 2.38 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.12 seconds

=== Summary ===

Correctly Classified Instances       25143               99.8055 %
Incorrectly Classified Instances        49                0.1945 %
Kappa statistic                          0.9961
Mean absolute error                      0.0037
Root mean squared error                  0.0431
Relative absolute error                  0.7467 %
Root relative squared error              8.6415 %
Total Number of Instances            25192

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.002    0.998      0.998    0.998      0.996    1.000     0.999     normal
                 0.998    0.002    0.998      0.998    0.998      0.996    1.000     0.999     anomaly
Weighted Avg.    0.998    0.002    0.998      0.998    0.998      0.996    1.000     0.999

=== Confusion Matrix ===

     a     b   <-- classified as
 13425    24 |     a = normal
    25 11718 |     b = anomaly
```

Figure 3: Training Set Result in WEKA

Table 3: Confusion Matrix of Training Set Classification

Classification	A	B
Normal = A	13425	24
Anomaly = B	25	11718

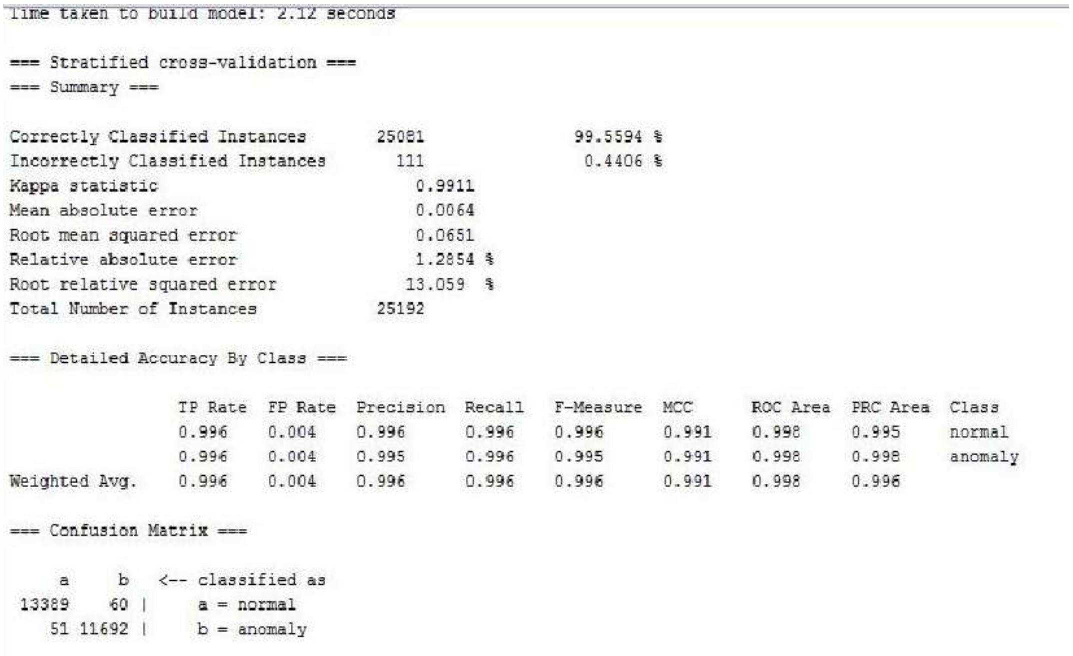
The test results include: test split (Figure 4) and cross validation result (Figure 5). The test split has about 99.5% correctly classified instances as displayed in figure 4.



```
Time taken to test model on test split: 0.02 seconds

=== Summary ===

Correctly Classified Instances        7519               99.484  %
Incorrectly Classified Instances        39                0.516  %
Kappa statistic                          0.9896
Mean absolute error                      0.0073
Root mean squared error                  0.0698
Relative absolute error                  1.4677 %
Root relative squared error             13.9925 %
Total Number of Instances             7558

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.994    0.005    0.996      0.994    0.995      0.990    0.998     0.997     normal
                 0.995    0.006    0.993      0.995    0.994      0.990    0.998     0.998     anomaly
Weighted Avg.    0.995    0.005    0.995      0.995    0.995      0.990    0.998     0.997

=== Confusion Matrix ===

    a    b   <-- classified as
 4036   23 |    a = normal
   16 3483 |    b = anomaly
```

Figure 4: Test split result

The cross validation has 99.56% correctly classified instances as displayed in Figure 5. The compiled summary results were presented in Table 4 for both training and testing of the system.



```
Time taken to build model: 2.12 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       25081               99.5594 %
Incorrectly Classified Instances       111                0.4406 %
Kappa statistic                          0.9911
Mean absolute error                      0.0064
Root mean squared error                  0.0651
Relative absolute error                  1.2854 %
Root relative squared error             13.059  %
Total Number of Instances            25192

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.996    0.004    0.996      0.996    0.996      0.991    0.998     0.995     normal
                 0.996    0.004    0.995      0.996    0.995      0.991    0.998     0.998     anomaly
Weighted Avg.    0.996    0.004    0.996      0.996    0.996      0.991    0.998     0.996

=== Confusion Matrix ===

     a     b   <-- classified as
 13389    60 |     a = normal
    51 11692 |     b = anomaly
```

Figure 5: Cross validation result

From the summary table of results, the kappa statistics value is about 0.99 for all, this value is close to 1 which is the maximum expected. The mean absolute and root mean square error values are also very minimal for both the training and testing results as expected.

Table 4: Evaluation Summary for both Training and Testing Set

Evaluation On Training Set			Evaluation On Testing Results	
S/N	Evaluation Metrics	Results	Test Split	Cross Validation
1.	Kappa Statistics	0.9961	0.9896	0.9911
2.	Mean Absolute Error	0.0037	0.0073	0.0064
3.	Root Mean Squared Error	0.0431	0.0698	0.0651
4.	Relative Absolute Error	0.7467%	1.4677	1.2854
5.	Root Relative Squared Error	8.6415%	13.9925	13.059
6	Correctly classified	99.81%	99.48%	99.56%

CONCLUSION

In this paper, the design, implementation and validation of the C4.5 based Intrusion Detection System model were conducted. While training and testing the model, it was concluded that classification and detection rate obtained from all training and test results is about 99% and ROC Area of about 0.9. The model showed great accuracy in detecting intrusion in comparison with other models.

REFERENCES

Balaji, S., & Kumar, K. B. (2013). A New Intrusion Detection System in Data mMining & Fuzzy Logic. *International Journal of Modern Engineering Research Vol. 3, Issue. 6*, 3425-3428. ISSN: 2249-6645.

Crosby, G. V., Hester, L., & Niki, P. (2011). Location-aware, Trust-based Detection and Isolation of Compromised Nodes in Wireless Sensor Networks. *International Journal of Network Security, Vol.12, No.2.*, 107-117.

Dhopte, S., & Tarapore, N. (2012). Design of Intrusion Detection System using Fuzzy Class-Association Rule Mining based on Genetic Algorithm. *International Journal of Computer Applications. Volume 53– No.14* , 20-27. ISSN: 0975 – 8887.

Hofmann, M., & Klinkenberg, R. (2014). *Rapid Miner: Data Mining Use Cases and Business Analytics Applications.* Boca Raton: CRC Press.

Holmes, G., Donkin, A., & Witten, I. (2002). WEKA: a machine learning workbench. *IEEE Xplore.* Brisbane: IEEE.

Jabez, J., & Muthukumar, B. (2015). Intrusion Detection System (IDS): Anomaly Detection using Outlier Detection Approach. *International Conference on Intelligent Computing, Communication & Convergence. Procedia Computer Science 48* (pp. 338 – 346). India: Elsevier B.V.

Liu, Y., & Zhu, L. (2019). A new intrusion detection and alarm correlation technology based on neural network. *EURASIP Journal on Wireless Communications and Networking*.

Nayak, U., & Rao, U. H. (2014). *The InfoSec Handbook. An Introduction to Information Security.* Apress. ISBN: 978-1-4302-6383-8.

Patil, U., Gunjal, R., Gadhe, A., Kulkarni, R., & Mandlik, S. (2016). Network Intrusion Detection & Prevention System using Fuzzy Logic and Genetic Algorithm. *International Journal of Innovative Research in Science and Engineering. Vol 2(3)*, 276-283. ISSN: 2454-9665.

Quinlan, R. J. (1993). *C4.5: programs for machine learning.* San Francisco: Morgan Kaufmann Publishers Inc.