iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** V **Month of publication:** May 2018

DOI: http://doi.org/10.22214/ijraset.2018.5049

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue V, May 2018- Available at www.ijraset.com

Protocol-Specific Intrusion Detection System using KNN Classifier

N. Sameera[1], Prof. M. Shashi[2]
[1]*Research Scholar,* [2]*Professor Dept of CSSE, Andhra University, Visakhapatnam, INDIA*

Abstract: ***Information Security Analytics is evolving as a big trend in recent years. Lots of existing knowledge is not enough to understand it completely. In order to explore it more and to sharpen research work on it, first of all, user has to understand the importance of security. The target is on analytics which is the process of taking raw data and by processing the data and produces meaningful information through which one can derive patterns. Information security requires enthusiastic people who are interested in taking security challenges against continually evolving attacks, as opportunities to excel in the field of security. In this digital world, attacker's strategy keeps changing. They even can make use of defenders actions as a step to build a new attack. This paper proposes a new approach of applying called "protocol-specific Intrusion Detection System Using KNN Classifier" for identifying an abnormal/anomalous transaction, which leads to an attack. This can be done by submitting each observation to the model, which is already trained with some training data based on the protocol of the transaction.***
Key Words: KNN, training data, validation data, testing data, confusion matrix

I. INTRODUCTION

Internet dependency has increased gradually in such a way that it has become part of our lives. Rapid growth in the process of digitization of data leads to the problem of maintaining large volumes of generated data. Because of un-patched vulnerabilities in the security mechanisms, there are chances that our data can be attacked by intruders [1]. This indicates a big threat to the confidentiality, availability and integrity of the user's information. Attacker enters the system pretending like a normal user to launch an attack by different ways, which is called as a cyber-attack or intrusion. The process of firing an attack always involves some prior task of observing victims environment to identify vulnerabilities and breach through to capture various components of the target system [2]. In order to protect the system from cyber-attacks or intrusions which may happen at any time, anywhere, intensive monitoring of the system activity is essential to prevent or detect and recover from such attacks. BInformation security analytics involves development of automated methods and tools for defending a system from attackers and are referred to as intrusion detection systems [3]. Intrusion detection systems often borrow extensive data analysis and mining techniques for classification, clustering etc. security related data [14]. Data in the real world is very tough and complex and requires strategic procedures to draw conclusions. Data Analytics supports the process of examining complex structured data and its behavior there by extracting patterns useful for decision making. Behavior of the data and patterns are extracted and analyzed step by step. Data analytics involves the process of collecting, preprocessing and transforming the data to extract patterns from it either through supervised or unsupervised learning depending on the availability of class label for known set of examples [4]. Intrusion detection systems apply supervised learning methods for detecting known attacks based on their signatures while they apply unsupervised learning methods for identifying unknown or zero-day attacks due to unavailability of signatures as they aim to exploit zero-day vulnerabilities [15]. This paper focuses on analyzing the traffic packets of NSL_KDD dataset described in terms of numeric attributes to classify them in to a set of known attacks. Since most of the attributes are numeric, K-nearest neighbor classifier is found to be suitable [5]. Though it is one of the most popular algorithms for performing classification on data sets with numeric attributes, the authors propose a variant named "Protocol-Specific Intrusion Detection System Using KNN Classifier" for appropriately handling an important attribute named "protocol type" which is a multi-valued categorical attribute. The rest of the paper is organized as follows. Section 2 gives the motivation of authors towards this approach. Session 3 presents the KNN classification. Section 4 describes about data set used. Section 5 tells about feature selection and preprocessing. Session 6 presents the approach of protocol specific KNN classifier. Section 7 describes about results and Session 8 presents conclusions.

II. MOTIVATION

NSL_KDD is heavily imbalanced data set to attack instances [17].Feature values and distribution of class label of instances in NSL-KDD data set are specific to protocol [6]. Some features need to be ignored based on the protocol as they include null information at the same time, the class which is more significant in one protocol may be less significant in another as shown in Fig.1. This skewed

distributions of classes and uneven distribution of feature values motivates authors to move towards Protocol-Specific Intrusion Detection System which partitions the data according to the protocol of the data and perform classification by only considering informative features and significant classes.

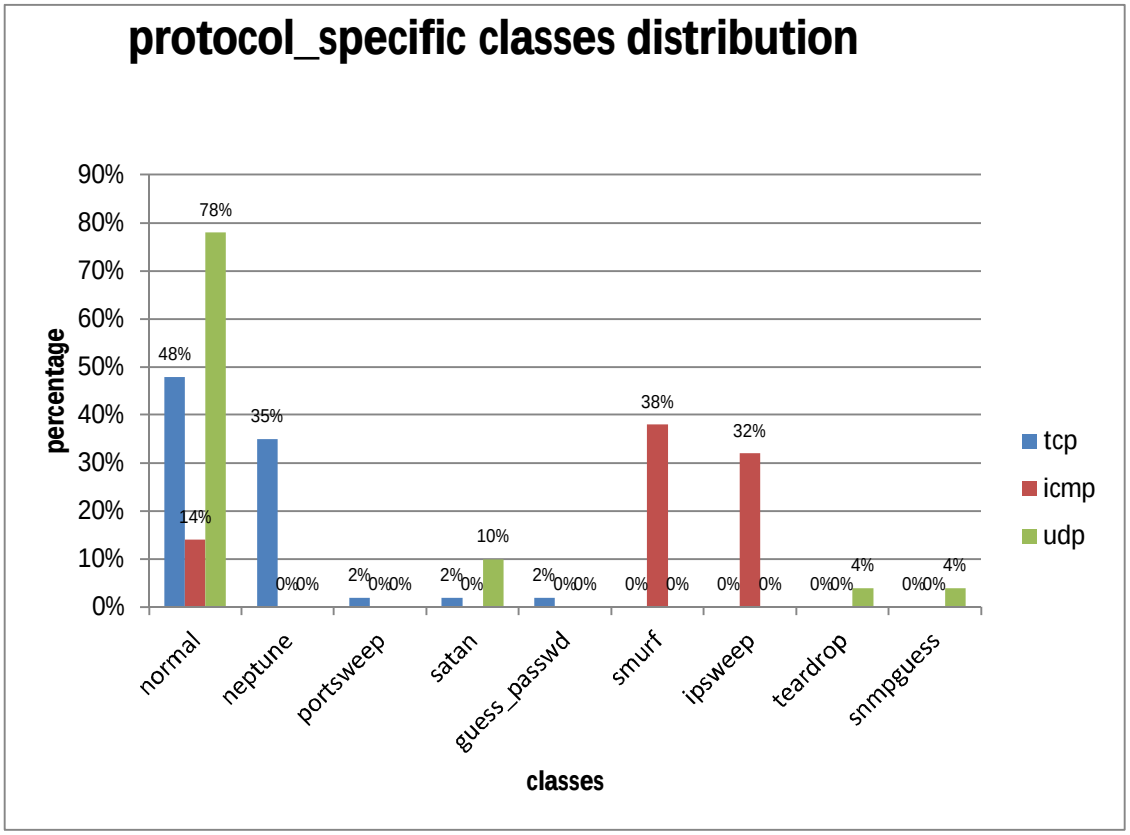


Fig. 1: Class-wise distribution of instances belonging to different protocols

III. K-NEAREST NEIGHBOR CLASSIFICATION MODEL

Supervised learning is defined in general as the process of learning about the data under the guidance of a supervisor. During the process of learning, labeled train data which consisting of set of training examples is observed and analyzed. This approach finally produces an inferred function which can be used for mapping new examples. KNN is an algorithm for classification of data under supervised learning approach [7]. Classification is the process of identifying and classifying the data in to corresponding classes. KNN classifies the things by performing analytics on past/historical data available. It uses different distance metrics like Euclidian, Manhattans, Murkowski. These metrics are described in Fig.2. Having a set of known class labels, for each new data item X, this algorithm takes its nearest K data items and estimate the closeness by using distance metrics. Finally assigns item X to the class of its closest neighbor [8]. This algorithm increases its efficiency by increasing the value of K. Overall, the K-nearest neighbor is a very effective classifier. Within R the "class" package offers support for K-nearest neighbor.

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Fig. 2: K-nearest neighbor distance metrics.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue V, May 2018- Available at www.ijraset.com

The only input that can be supplied to the K-nearest neighbor is K, which indicates the number of closest training observations. The required class of the observation can be decided based on the majority of the votes gives by its K neighbors. Consider the Fig.3. for getting clear view of the K-nearest neighbor functionality [7].

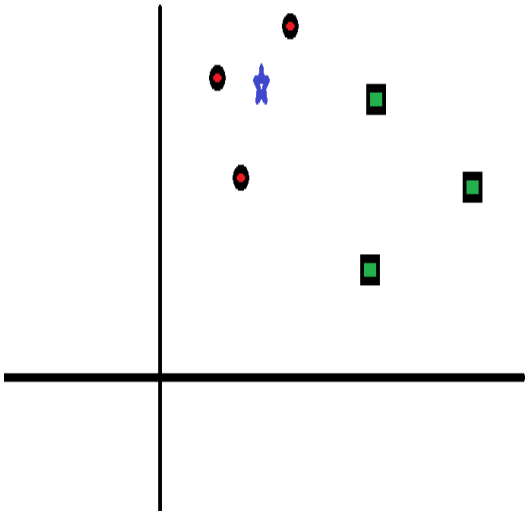


Fig.3: scenario1

Here, in this sample training data only two groups are there one is represented by circle and other is represented by square. Observation marked as star is the new incoming one, which needs to be classified in to the either of the groups. Considering K value is very important in KNN classification and it is better to consider an odd K value. For suppose take k=3,that means we need to consider 3 neighbors and finally consider majority class label as shown in the Fig.4 . Since in this case all the three nearest neighbors to the star are belongs to the circle only. So from this it can be clearly concluded that the class of star is circle.

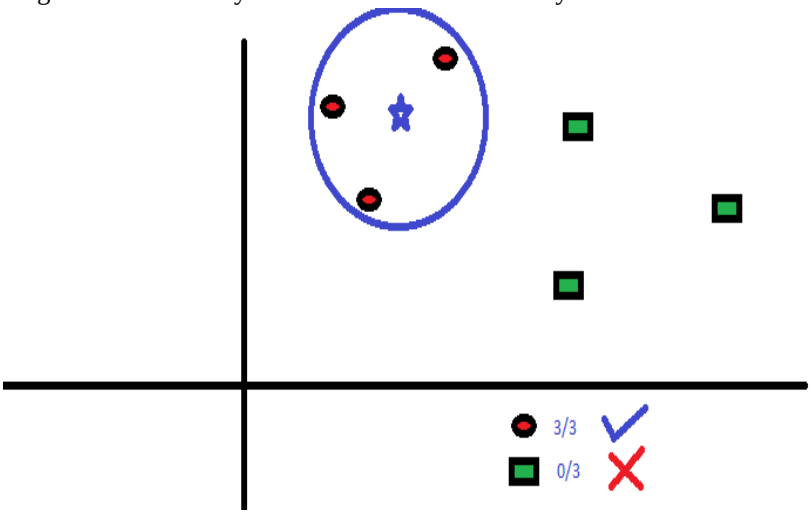


Fig.4: senario2

IV. DATA SET

The dataset we have taken for testing our new method is "NSL-KDD" which is a refined version of traditional KDD99 data set. This data set contains information about different attributes of data items that leads to different attacks [6]. These attacks are grouped in to four groups. Probe, U2R, R2L and DOD. Classes under these four groups and their frequency of occurrence are shown in Table1. Probe attack includes the process of observing the target for identifying vulnerabilities [13]. U2R(user to root) attack includes all the ways of an intruder, who is trying to gain the administrative access of the target.R2L (remote to local) and DOD (Denial of service) attack blocks the availability of the resources by making them busy, so that the resources are unavailable to the actual needy [9]. This class distribution is also pictorially represented in the Fig: 5.

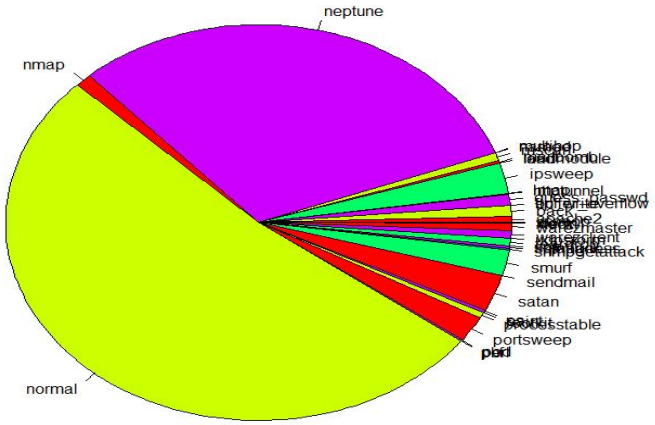


Fig.5: distribution of classes in nsl_kdd data set.

From the pie chart it is observed that almost half observations are labeled as normal class. After that next majority class is Neptune. All the remaining classes are very less in number. This unequal distribution may affect the predictions. In order avoid this we have taken stratified sampling from the data and considered this sampled data as the data given to the model.Table.2 gives the list of features present in the dataset.

V. FEATURE SELECTION AND PREPROCESSING

Variables describing the data set are called features. Features of dataset plays an active part in identifying the class label. After choosing the dataset, the next important step one has to take is selecting significant features which contribute much to the dataset. This process is called "feature selection". Based on the domain expertise one has carefully decided which feature is significant and which is not. Feature selection is an iterative process [6]**.** If the constructed model is not doing good with the selected features we can backtrack to the feature selection process. Out of 43 features, as a step of preprocessing the data, we have eliminated some of the features which are having all zeros, as zero valued fields contribute nothing. Those eliminated fields are 7,9,14,15,18,20 and 21.And features numbered 3, 4 are also eliminated as they are non- numeric fields having more levels. Later we have divided the pre-processed data in to modules with respect to the prototype, in which each module data is again pre-processed if necessary and partitioned for training and testing purposes [16]. From training data set we have considered only the observations with respect to the significant classes. Before going to build the model, this preprocessed data should be normalized.

Sno	Class	Frequency	Sno	Class	Frequency
1	apache2	737	21	xsnoop	4
2	imap	12	22	ftp_write	11
3	multihop	25	23	loadmodule	11
4	phf	6	24	nmap	1566
5	saint	319	25	processtable	685
6	spy	2	26	smurf	3108
7	worm	2	27	udpstorm	2
8	back	1300	28	xterm	13
9	ipsweep	3643	29	guess_passwd	1284
10	named	17	30	mailbomb	293
11	pod	221	31	normal	76967
12	satan	4360	32	ps	15
13	sqlattack	2	33	snmpgetattack	178
14	xlock	9	34	warezclient	890
15	buffer_overflow	50	35	httptunnel	133
16	land	22	36	mscan	996
17	neptune	45716	37	perl	5
18	portsweep	3070	38	rootkit	23
19	sendmail	14	39	snmpguess	331
20	teardrop	901	40	warezmaster	964

Table 1: classes in NSL-KDD data set

Sno	Class	Sno	Class
1	Duration	24	srv_count
2	protocol_type	25	serror_rate
3	service	26	srv_serror_rate
4	flag	27	rerror_rate
5	src_bytes	28	srv_rerror_rate
6	dst_bytes	29	same_srv_rate
7	land	30	diff_srv_rate
8	wrong_fragment	31	srv_diff_host_rate
9	urgent	32	dst_host_count
10	hot	33	dst_host_srv_count
11	num_failed_logins	34	dst_host_same_srv_rate
12	logged_in	35	dst_host_diff_srv_rate
13	num_compromised	36	dst_host_same_src_port_rate
14	root_shell	37	dst_host_srv_diff_host_rate
15	su_attempted	38	dst_host_serror_rate
16	num_root	39	dst_host_srv_serror_rate
17	num_file_creation	40	dst_host_rerror_rate
18	num_shells	41	dst_host_srv_rerror_rate
19	num_access_files	42	class
20	num_outbound_cmds	43	difficulty
21	is_host_login		
22	is_guest_login		
23	count		

Table 2: Attributes in NSL-KDD dataset

VI. PROTOCOL SPECIFIC INTRUSION DETECTION SYSTEM USING KNN CLASSIFIER:

In-order to improve the accuracy of the classification, authors proposes a new approach called "Protocol Specific Intrusion Detection System Using KNN Classifier" by following the principle of the divide-and-conquer. Algorithm of the proposed model is given in Table 3. The total data is divided into two parts. One part is used for implementing the approach and other art is used for testing the approach it is called testing data. KNN is only applicable for numeric data [10]. Based on the distribution of NSL-KDD dataset, It is found that categorical fields like protocol plays major role in classification. So, to consider protocol authors move towards protocol specific approach. Dataset used for implementing the approach is partitioned based on its protocol. By observing the dataset, it is found that features and class labels are very specific with respect to the protocol. According to this authors ignore some features based on the protocol and also ignore less significant classes by only considering the most significant ones. This preprocessed data along with the validation data is supplied to the KNN classifier. K value which indicates number of neighborhoods, influences the accuracy of results. The best K value is noted for each protocol partition by performing validation. This best K value should be supplied at the time of testing in order to get the best accuracy.

Algorithm:

Step 1: Remove non-numeric features having more levels and also remove features with null information.

Step 2: Convert all features with 2 distinct values into binary.

Step 3: Partition the dataset into two parts in 80:20 ratio and use 80% data (part1) for building and validating the models and use 20% (part2) data for testing the performance of the approach.

Step 4: Partition the total data (part1) into 3 modules based on the protocol.

Step 5: On each protocol based partition, define a model in terms of relevant features, plausible class labels and the Best_K by repeating the following steps

(a) Divide the partition into 60:40 ratios to generate training set and validation set.

(b) Identify significantly frequent classes from the training set as plausible class labels and screen the remaining instances.

(c) Generate a training set sample with almost equal representation from each of the plausible classes either by over sampling or by under sampling the training instances.

(d) Remove features with null information and normalize the relevant features of training and validation data by fitting into the scale of [0, 1] .

(e) Record the class labels of training and validation data separately.

(f)Apply KNN by supplying normalized training and validation data sets along with the class labels of the training data as detailed below for each K:

(i) For each instance of the validation data find its distance to all instances of the training set sample and identify closest K training examples referred to as its neighbors.

(ii) Label the validation instance with the majority class label among its neighbors and record it as the predicted class label for the validation instance.

(iii) Find the classification accuracy by comparing predicted class labels with the actual class labels of the validation set for a given K value.

(g) Repeat the steps (i) to (iii) for different K values within the plausible range and find the K that results in the maximum accuracy as the Best_K.

Step 6: Integration of protocol-specific models to classify test cases/unseen cases :

(a)For each instance of the test data, based on the type of the protocol, invoke corresponding model along with its Best_K to predict class label of the test instance.

(b)To estimate the performance of the protocol specific KNN classifier model compare the predicted labels with the actual labels of the test cases and count the number of true-positives true-negatives, false positives and false negatives.

(c) Estimate the accuracy and loss

Table 3: protocol specific Intrusion Detection System Using KNN classifier

VII.EVALUATION METHODS

Several techniques have been developed to evaluate the model with in supervised algorithms. Based on results obtained by running the approach on the test data one can have some predictions about the results [11]. To determine how well the approach did, one need to look at the proportion of correctly predicted results on the test data. We can calculate the accuracy of the result by considering number of correct predictions means true positives and true negatives and then dividing that by the total number of predictions. For this we need to construct the confusion matrix by using cross table. Accuracy = (TP+TN) / (Total-Predictions) Loss estimation can be obtained by considering cost of ignoring the intrusion, cost of miss-classification and cost of false intrusion by assigning proper weightage. "Cost of ignoring the intrusion" can be obtained by adding all false positives and high weightage is given to it as it leads to the dangerous step if the actual intrusion is ignored. Next high weightage is given to the cost of misclassification which indicates false positives and false negatives. And no weightage is assigned to the cost of false intrusion which indicates false negatives.Results are ultimate output of the approach which decides the importance of using that particular approach. By applying the KNN classifier to the nsl_kdd data set we will get the validation results at different K values shown in fig. 6,7,8. By observing validation results for each protocol-module the best K-value for each module is decided [12]. By using this best K-value testing is done on the test set of NSL-KDD data which results in best accuracy. Coincidentally, here the best K-Value for the three protocol modules is seems to be 3 as observed from bellow graphs. By submitting this best K-value to the protocol-specific KNN classifier, testing accuracy is as good as 96% and the loss is as low as 0.4

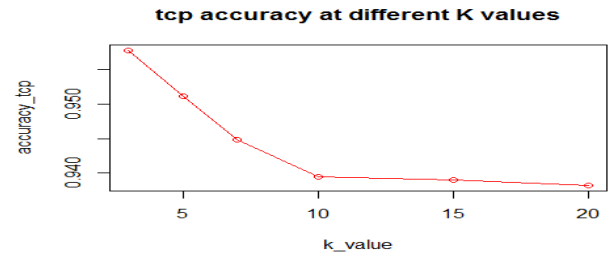


Fig 6: Validation results of TCP module

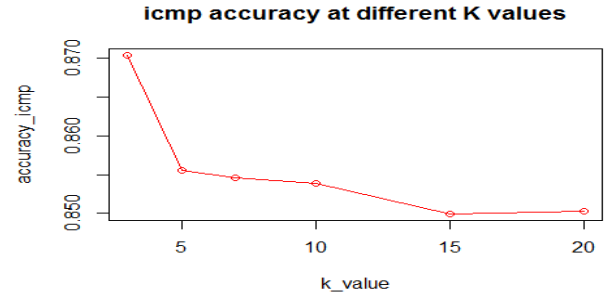


Fig 7: Validation results of ICMP module

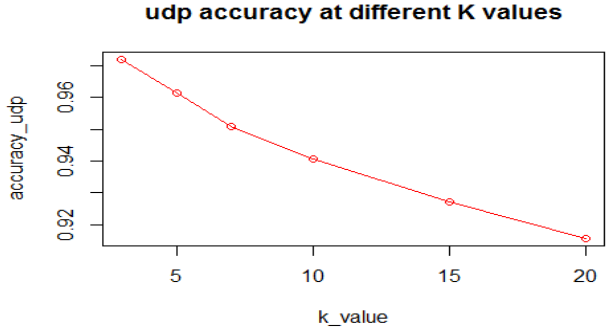


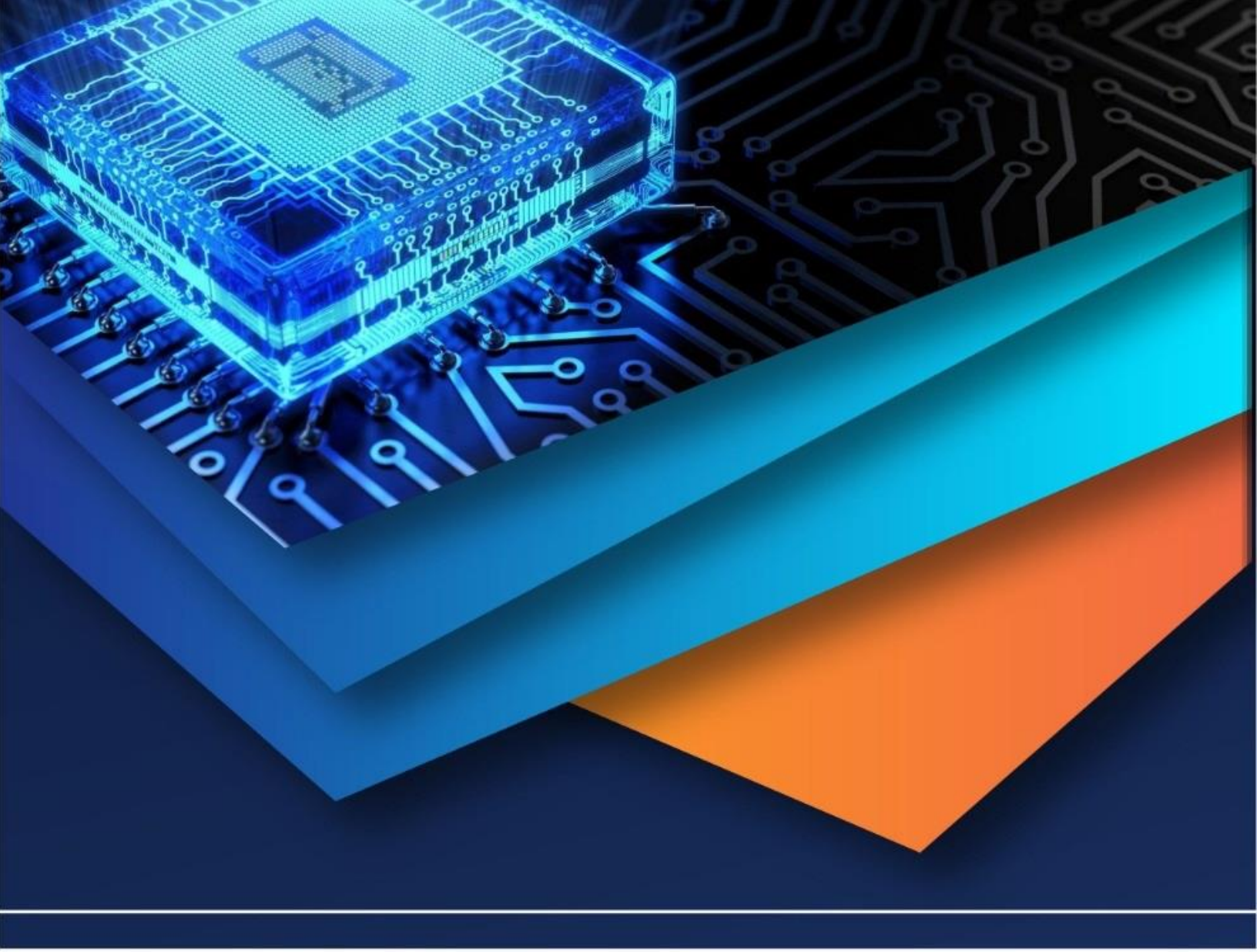
Fig 8: Validation results of UDP module

VIII. CONCLUSION

In this paper, authors propose a novel approach for detecting intrusions by protocol specific KNN classifier. Since most of the attacks are significant with respect to protocol, the focus of this approach is in considering the important categorical attribute called protocol. Data is partitioned according to its protocol. Separate model is built on the protocol specific data by only considering the predominant class labels from each partition and by ignoring the null informative features. The testing data can be supplied to this intrusion detection system which invokes the corresponding models to generate class labels of instances. Results show that our approach is good at detecting existence of the attacks and is feasible in improving the performance of attack detection process. The accuracy is as good as 96%. In this way, the proposed "Protocol-specific intrusion detection system using KNN classifier" gives best results for detection of intrusions.

REFERENCES

[1] George. S. Oreku, Fredrick J.Mtenzi "Cybercrime: Concerns, Challenges and Opportunities" "Information Fusion for Cyber-Security Analytics", Studies in Computational Intelligence 691, Springer International Publishing, 2017.
[2] Jay Jacobs, Bob Rudis , "Data-Driven Security: Analysis, Visualization and Dashboards"
[3] Leandros A. Maglaras,, Ki-Hyung Kim, Helge Janicke, Mohamed Amine Ferrag, Stylianos Rallis, Pavlina Fragkou, Athanasios Maglarasf, Tiago J. Cruz, "Cyber security of critical infrastructures " Available online at www.sciencedirect.com, ICT Express 4 (2018) 42–45.
[4] Setiawan*, Supeno Djanali, Tohari Ahmad, "A Study on Intrusion Detection Using Centroid-Based Classification" 4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia
[5] Buczak, Member, IEEE, and Erhan Guven, Member, IEEE, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection" IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 18, NO. 2, SECOND QUARTER 2016.
[6] Frans Hendrik Botes, Louise Leenen and Retha De La Harpe, "Ant Colony Induced Decision Trees for Intrusion Detection" conference paper june 2017. Available:
[7] Ahmed I. Saleh1 · FatmaM. Talaat1 · LabibM. Labib, ''A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers'' © Springer Science+Business Media B.V. 2017
[8] https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering [Last accessed on April 27,2018
[9] Amar Meryem, Douzi Samira, El Ouahidi Bouabid, Lemoudden Mouad, "A novel approach in detecting intrusions using NSLKDD database and MapReduce programming" . The 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017). Available online at www.sciencedirect.com , Procedia Computer Science 110 (2017) 230–235 .
[10] HONGXING MA, JIANPING GOU, (Member, IEEE), XILI WANG1, JIA KE, & SHAONING ZENG, "Sparse Coefficient-Based k-Nearest Neighbor Classification" September 6, 2017.
[11] Shichao Zhang, Senior Member, IEEE, Xuelong Li, Fellow, IEEE, Ming Zong, Xiaofeng Zhu, and Ruili Wang, "Efficient kNN Classification With Different Numbers of Nearest Neighbors" ,IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 29, NO. 5, MAY 2018.
[12] Camille Cobb, Samuel Sudar, Nicholas Reiter, Richard Anderson, Franziska Roesner, Tadayoshi Kohno, "Computer security for data collection technologies", ICTD Lab and Computer Security & Privacy Research Lab Computer Science & Engineering, University of Washington, USA. Contents lists available at ScienceDirect Development Engineering.
[13] F. C. Freiling, T. Holz, and G. Wicherski, ``Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks," Lect. Notes Comput. Sci., vol. 10, no. 2, pp. 319_335, 2005.
[14] WEIZHI MENG, (Member, IEEE), ELMAR WOLFGANG TISCHHAUSER, QINGJU WANG, YU WANG, & JINGUANG HAN, (Member, IEEE), "When Intrusion Detection Meets Blockchain Technology: A Review" March 15, 201
[15] Lee, W., Stolfo, S. J., Chan, P. K., Eskin, E., Fan, W., Miller, M., ... & Zhang, J. (2001). Real time data mining-based intrusion detection. In DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings (Vol. 1, pp. 89-100). IEEE.
[16] Muttaqien, I. Z., & Ahmad, T. (2016, December). Increasing performance of IDS by selecting and transforming features. In Communication, Networks, and Satellite (COMNETSAT), 2016 IEEE International Conference on (pp. 85-90). IEEE.
[17] Özgür, A., & Erdem, H. (2016). A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ PrePrints, 4, e1954v1.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY