International Journal of Inventive Engineering and Sciences (IJIES)
ISSN: 2319-9598 (Online), Volume-10 Issue-2, February 2023

A Simple and Effective Intrusion Detection System for Manets

M V D S Krishna Murty, Lakshmi Rajamani

Abstract: ***This work proposes a simple and effective Intrusion Detection System (IDS) to classify different attacks in MANETs. IDS extracts four features for every traffic pattern and applies the Support Vector Machine algorithm to them for classification. Before using the feature extraction, the input traffic pattern is subjected to pre-processing, as it is composed of non-uniform features. IDS classifies the input traffic pattern into three classes: normal, blackhole, and wormhole. Finally, this work analyses the feasibility of machine learning algorithms for detecting security attacks in MANETs. For experimental validation, we have referred to a self-created dataset acquired from observations of the traffic patterns of nodes attacked by black holes and wormholes. Moreover, we have also validated the proposed method through the NSL-KDD dataset.***

Keywords: ***Intrusion Detection System, Preprocessing, Feature Extraction, Support Vector Machine, Self-Created Dataset.***

I. INTRODUCTION

Mobile Ad hoc Networks (MANETs) are one type of wireless network formed with mobile devices as nodes. Due to the nature of decentralized communication, MANETs have gained tremendous interest in different applications including emergency rescue operations, military operations, collaborative distributed computing, disaster management and some personal network applications [1] etc. Due to the unique characteristics of mobile nodes, several challenges exist in MANETs that need to be addressed. Among the several challenges, mobility is the most significant, and it has consequences for several other sub-challenges. Almost all the problems in MANETs are linked to the mobility of nodes. Among several sub-challenges, secure data exchange between mobile nodes is the major challenge. Due to the open network topology, distributed nature, and the absence of centralized administration in MANETs, the mobile nodes are susceptible to various attacks [2].

The impact of these attacks ranges from naïve passive eavesdropping to severe battery-draining attacks [3]. Primarily, the attackers focus on the resources of mobile nodes, such as battery power, bandwidth, and data. Among the several security attacks in MANETs, the blackhole attack [4] and wormhole attack [5] are the two major attacks which cause severe damages to the network. These two attacks are dynamic and vary based on several network parameters. Hence, the identification of mobile nodes that were attacked with blackhole and wormhole attacks is much difficult. Recently, the use of artificial intelligence has increased in various applications. Compared to static algorithms, which cannot provide any prior information about attacks to mobile nodes, machine learning algorithms that train the nodes can facilitate the proper and accurate detection of attacks. A mobile node trained with the attack's characteristics can easily identify the attacked or compromised neighbour node. Hence, our research has been motivated by these issues and focused on developing an effective Intrusion Detection System to address these problems to some extent. This paper presents a simple and effective Intrusion Detection System for classifying mobile nodes into three categories: standard, blackhole, and wormhole. The overall system is composed of three phases: pre-processing, feature extraction, and classification. At the initial phase, the input data is normalised and transformed into a uniform format, as the raw data collected from MANETs is non-uniform in nature. For feature extraction, we have employed four statistical features: mean, variance, maximum, and minimum. After feature extraction, we applied principal component analysis for dimensionality reduction. Finally, the Support Vector Machine (SVM) algorithm was used for classification.

The remaining paper is organized as follows; Section II explores the literature survey on IDS methods. Section III examines the details of the proposed approach in more depth. Section IV presents the details of the experimental analysis, and Section V concludes the paper.

II. LITERATURE SURVEY

IDS primarily operates based on the principles of machine learning and information processing. In IDS, the mobile node initially learns about the characteristics of different attacks through machine learning algorithms, and it becomes ready to identify the attack if it occurs. IDS works on the features of network traffic (data packets and control packets).

For an incoming traffic pattern, the node analyses its features and matches those features with the features by which it was

Manuscript received on 31 January 2023 | Revised Manuscript received on 06 February 2023 | Manuscript Accepted on 15 February 2023 | Manuscript published on 28 February 2023.

*Correspondence Author

M V D S Krishna Murty*, Research Scholar, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Hyderabad (Telangana), India. E-mail: mkrishnamurty@gmail.com, ORCID ID: https://orcid.org/0000-0002-4705-3818

Dr. Lakshmi Rajamani, Professor and Head (Retd), Department of Computer Science and Engineering, Osmania University, Hyderabad (Telangana), India. E-mail: drlakshmiraja@gmail.com

trained. If the features of the incoming packet match the attack features, then the incoming traffic is declared as an attacked packet or a packet coming from an attacked node.

The earlier IDS works [6-9] are employed on fixed network traffic which was acquired in specific situations. In that, researchers installed a monitoring unit in a networked system and analysed the flow of traffic. Based on the analysis, they formulated standard datasets and made them publicly available. KDD-CUP 99 is a dataset of this kind that researchers generally use. In this section, we explore the details of several earlier IDS mechanisms.

Ji et al. [10] proposed the IDS model in three steps: feature selection, visual analysis and classification. Under feature selection, this method employed a signal processing technique, specifically the multi-level discrete wavelet transform (MDWT). Next, for visual analysis, iPCA is employed, and finally, for classification, the SVM algorithm is used. The NSL-KDD dataset is used to validate the IDS model developed. However, the data connections related to data traffic won't have any significance in terms of high and low frequencies. Ambusaidi et al. [11] have developed filter based feature selection mechanism called as Flexible Mutual Information based Feature Selection (FMIFS) to select optimal features for data traffic connections.

FMIFS employs mutual information (MI) to determine the mutual dependency between features, and based on the obtained MI values, duplicate features are eliminated. The duplicate features are those that contribute less to the class, as well as neighbour features. Least Squares SVM (LS-SVM) is employed for classification, and the simulations are conducted on the three datasets, such as KDD cup99, NS-KDD and Kyoto2006+. Fei Zhao et al. [12] proposed a new feature selection algorithm called Redundant Penalty between Features based on Mutual Information (RPFMI) to select optimal features.

The RPFMI considers three factors during feature selection: redundancy between features, the effect of selected features on classes, and their relationship with candidate features. Two datasets, such as KDD Cup 99 and Kyoto 2006+, are employed for experimental validation. The performance is measured through an accuracy measure. Jingping Song et al. [13, 14] proposed a Modified Mutual Information-based feature Selection (MMIFS) method for intrusion detection. After selecting features through MMIFS, they employed the C4.5 classifier for classification purposes.

For simulation purposes, they used the KDD Cup 99 dataset, and performance is measured through an accuracy metric. G. Farahani [15] proposed a new method called Cross-Correlation based feature selection (CCFS) and employed four classifiers for classification purposes. The four classifiers are, namely, K-nearest neighbour (KNN), Decision Tree (DT), Naïve Bayes (NB), and SVM. The primary purpose of CCFS is to achieve dimensionality reduction, thereby reducing the computational burden. For simulation purposes, they have considered four datasets: KDD Cup99, NSL-KDD, AWID, and CIC-IDS2017, and the performance is measured in terms of accuracy, recall, and precision.

Chun Meng et al. [16] proposed an improved version of K-means algorithm for intrusion detection in computer networks. Initially, the PCA algorithm is applied to reduce the dataset's dimensionality, and then outlier detection is used to eliminate outliers that significantly impact the final clustering results. The initial clustering centre is chosen based on distance, allowing for an optimal local solution. Then, K is used to obtain the final cluster centres.

Simulation is performed using the KDD Cup99 dataset, and performance is measured through the detection rate and false positive rate. Wang et al. [17] proposed an ensemble method for the anomaly based intrusion detection. This method combined two ML algorithms, namely Artificial Neural Networks (ANN) and Fuzzy Clustering (FC). FC is used for creating different training sets, and ANN is used for training the models developed. Finally, they applied the fuzzy aggregation module to find the average results of all models. Experiments are conducted through the KDD Cup 99 dataset, and performance is measured through detection stability and precision.

Hoz et al. [18] proposed an anomaly based model by hybridizing three algorithms namely Probabilistic Self Organizing Maps (PSOMs), Fisher Discriminant Ratio (FDR) and Principal Component Analysis (PCA). In their method, the FDR and PCA are used to discover feature selection by suppressing noise. PSOMs are designed to model the feature space and ensure perfect discrimination between regular and malicious connections. The detection capabilities are altered without repetitive training, but by changing the probable activation units.

III. PROPOSED APPROACH

The overall working mechanism of the proposed IDS is shown in Figure 1. Accordingly, the proposed mechanism initially preprocesses the input data and then extracts features. Finally, the obtained features are fed to a machine learning algorithm for classification. As an additional methodology, the obtained features are processed through principal component analysis (PCA) for dimensionality reduction.

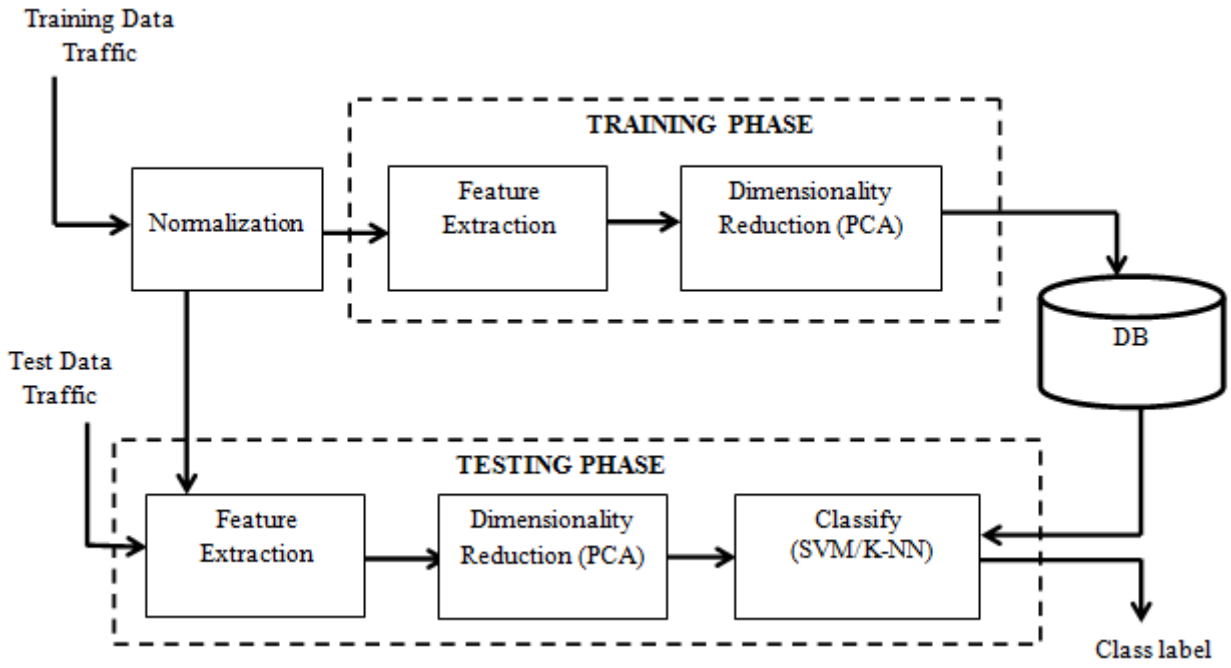

Figure 1: Overall Block Diagram of IDS Mechanism

3.1 Normalization

The features of a packet in raw form can be inconsistent, incomplete, redundant and noisy. In the obtained black hole and wormhole packet features, each packet is represented by a set of features, and these features are not in the same format. Some features are symbolic in nature, some features are numerical, and some features are in binary format. To process this dataset, all features must be in a uniform format. Hence, to sort out all these problems, data normalisation is required, and it varies from dataset to dataset. For demonstration purposes, here we employed a step-by-step normalization process as shown below-

Step 1: Consider the dataset X with size $M \times N$, where M is the total number of packets and N is the total number of features used to represent each packet.

Step 2: Fetch the features that need to be normalized

$$F_i = X(i); \quad (1)$$

Where F_i Denotes the i^{th} feature, which needs to be normalised. For our new dataset, we found that each packet is represented by 52 features, of which 31 are symbolic, eight are numeric, and the remaining are binary. Here, only symbolic features are processed for normalization. Thus, the value of i in Eq. (5.1) varies from 1 to 31.

Step 3: Determine the total number of occurrences of each feature by comparing it with the name already specified. The comparison followed by count is done as follows:

$$M_i = \sum_{i=1}^{L} strcmp(F_i, Feature\_X) \quad (2)$$

Where $Feature\_X$ represents the individual feature name and the F_i Represent the i^{th} feature in every row.

Step 4: Measure the probability of each feature given as

$$PF_i = \frac{M_i}{Length(X)} \quad (3)$$

Where M_i Is the total number of occurrences of feature i and $Length(X)$ Denotes the total size of the respective row.

Step 5: Replace the probability values of the i^{th} feature in their respective position in the row X.

For other datasets, if we observe incomplete connections, they are completed by adding zeros in sufficient numbers. Similarly, for datasets that contain connections with abnormal values, such as NaN and Infinity, these values are replaced with 0s.

3.2 Feature Extraction

For feature extraction, we measure four types of features: mean, standard deviation, maximum, and minimum. Each packet is represented with these four features. During the training phase, every packet is initially processed for block division, and then each block is processed to compute four features. The obtained features are then processed through PCA to extract only the principal components. For a given row X of size, 1×52It is divided into several overlapping blocks of size $1 \times w$. The mathematical calculations of the four features are described as follows:

A. Mean: The mean is measured as a ratio of the summation of features in the block to the total number of features in that block.

$$\mu = \frac{1}{w}\sum_{i=1}^{w} p_i \quad (4)$$

Where p_i Is the feature at the i^{th} position in the block of size $1 \times w$.

B. Maximum: For a given Block Bx of size $1 \times w$The maximum feature is calculated as

$$Mx = Max(B) \quad (5)$$

C. Minimum: For a given Block Bx of size $1 \times w$The minimum feature is calculated as

$$Mn = Min(B) \quad (6)$$

D. Standard deviation: Standard deviation explores the statistical distribution relative to the mean. Standard deviation can also be called as square root of variance. For a given Block Bx of size $1 \times w$ The standard deviation is calculated as

$$\sigma = \sqrt{\frac{1}{w \times w} \sum_{i=1}^{w \times w} (p_i - \mu)^2} \quad (7)$$

Hence, each packet is represented with B number of mean values, standard deviation values, maximum values and minimum values. For example, if the B value is 20, then each packet is represented with 20 mean values, 20 standard deviation values, 20 maximum values and 20 minimum values. Thus, the total number of features used to describe one packet is 80. For instance, if 300 packets are used for every class to train the system. In such a case, each class is represented by a total of 80 × 300 = 24,000 features, which is a considerable number. Hence, we have applied PCA to reduce the dimension with less information loss. Similarly, the PCA is also used to test the packet.

3.3 Classification

For classification purposes, we have employed the most popular SVM algorithm. The SVM increases the sample size such that it can separate them effectively. Hence, instead of a general trend towards dimensionality reduction, SVM follows an opposite process, increasing the size of features. The primary objective is to identify a hyperplane that encompasses the samples from a specific class. SVM employs kernel functions that postulate the linear and non-linear features, and hence it can construct a separating plane that is implicitly defined by the kernel function. Here, we have employed LIBSVM for classification purposes at the first stage.

IV. EXPERIMENTAL ANALYSIS

To demonstrate the effectiveness of the proposed IDS model, we conducted a comprehensive set of experiments on various datasets, and the performance is analysed for each dataset. Initially, we provide an overview of the datasets. Next, we explore the details of observed results and performance metrics. Finally, we evaluate the effectiveness of the proposed model by comparing the results obtained with those from existing methods.

4.1 Datasets and Settings

For simulation, we have considered two datasets: NSL-KDD and a self-created dataset. Initially, the details of the self-created dataset are explained, followed by an overview of the NSL-KDD dataset.

A. Self-created dataset

This is the real-time dataset that we acquired during our research on black hole and wormhole attacks in MANETs. The central theme behind this dataset is the observation of packet characteristics in the network in the presence of blackhole and wormhole attacks. As we used AODV for routing, the packet features are derived based on the attributes of the AODV protocol. In our research, we found that each packet is represented with 52 features. For dataset creation, we varied different network parameters and analysed the packets coming from blackhole-and wormhole-attacked nodes. Based on the analysis, we have identified various features that have a strong relationship with black hole and wormhole attacks. Additionally, we have also acquired features with typical characteristics, i.e., without any attack. On average, we have acquired a total of 12,354 standard packet traffic patterns, 6,128 blackhole-associated packet traffic patterns, and 6,355 wormhole-associated packet traffic patterns. In each class, we have employed 70% for training and 30% for testing. The details of the simulation data for the self-created dataset are shown in Table 1.

Table 1: Self-created dataset statistics

Class/Set		Total Traffic Patterns	Training (70%)	Testing (30%)
Normal		12,354	8648	3706
Attacks (12,483)	Blackhole	6128	4290	1838
	Wormhole	6355	4449	1906

B. NSL-KDD

The NSL-KDD is a revised version of the KDD Cup 99 dataset proposed by Tavallaee et al. This dataset is reconstructed to address several issues with the KDD Cup 99, including the large number of redundant records. To group the connections into five groups, the initial dataset was subjected to different classifiers, and each connection is labelled with the number of successful estimations. This dataset consists of five types of classes. They are Normal, DoS, U2R, R2L and Probe. Among these classes, the first one is non-attack, and the remaining four are attacks. Each connection of the NSL-KDD dataset consists of 41 features. Furthermore, the dataset comprises three distinct sets: KDDTrain+, KDDTest+, and KDDTest^{-21}. The initial set, i.e., KDDTrain+, comprises 125,973 connections, of which 67,343 are regular traffic connections and 58,630 are attack traffic connections. In the second set, i.e., KDDTest^{+}, the total number of traffic connections is 22,544, comprising 9,711 regular traffic connections and 12,833 attack traffic connections. Finally, in the KDDtest-21 set, the total number of connections present is 11,850, comprising 2,152 regular traffic connections and 9,698 attack traffic connections. We have conducted a cross-validation over the KDDTrain+ set and also considered a validation test using the KDDTest+ and KDDTrain-21 sets. The details of the number of connections present in these sets are demonstrated in the Table.2

Table 2 NSL-KDD Dataset Statistics

Class/Set		KDD Train^{+}	KDD Test^{+}	KDD Test^{-21}
Normal		67343	9711	2152
Attacks	DoS	45927	7458	4342
	U2R	52	200	200
	R2L	995	2754	2754
	Probe	11656	2421	2402
	Total	58630	12833	9698
Total		125973	22544	11850

Table. 3 Confusion Matrix of Self-Created Dataset

Actual/ Predicted	Normal	Blackhole	Wormhole	Total
Normal	**3335**	200	171	3706
Blackhole	300	**1360**	178	1838
Wormhole	372	200	**1334**	1906
Total	4007	1760	**1683**	7450

Table 4: Performance Metrics of Proposed Method for Self-Created Dataset

Class/Metric	DR (%)	PPV (%)	FNR (%)	FPR (%)	FAR (%)	F-Score (%)
Normal	89.9952	83.23	10.01	16.77	13.39	86.4852
Blackhole	73.9965	77.2741	26.012	22.7345	24.37	75.6042
Wormhole	69.9522	79.2652	30.0142	20.7415	25.371	74.3429

4.2 Results

Initially, we explained the results of our self-created dataset, followed by the details of NSL-KDD. For both datasets, we have trained the system with the specified number of patterns listed in the tables above. Once the training is completed, we start testing through the testing connections. After testing is completed, a confusion matrix is formulated based on the detected results. From that confusion matrix, we have measured performance using several key metrics.

Table 3 presents the confusion matrix of the results obtained by simulating the proposed model on a dataset created specifically for this purpose. For simulation purposes, we have considered 70% of the training data and 30% of the testing data. The original self-created dataset comprises 17,387 traffic connections in the training set and 7,000 traffic connections in the testing set. Here, we aimed to conduct a five-fold cross-validation mechanism; therefore, we considered only 70% of the data for training and 30% for testing. At every validation, we have removed some traffic connections from the past-trained and test sets and added new connections that were not used in earlier validations. In this way, we have conducted five-fold cross validations and the best results are shown in Table.3. Based on these values, the performance is measured through several performance metrics and they are shown in Table.4. From these values, we can observe that the maximum DR and PPV are observed for standard patterns. Since standard traffic patterns are significantly deviated from attack traffic patterns, the system can classify them effectively. In the remaining two classes, larger false positives are observed with regular traffic, as they exhibit significant deviation from individuals. The black hole packet characteristics are significantly different from those of wormhole packets; they experience more false positives in the normal category. For instance, we can see that the FP of normal is 300, while the FP of wormhole is only 178 under the blackhole category. Similarly, the FP of normal is 372 while the FP of blackhole is only 200 under wormhole. Based on these observations, we can say that the packet characteristics vary significantly when the attack on the nodes changes. In such a scenario, the common malicious node identification mechanism is not suitable. Every attack requires a specific detection mechanism, and only then can the MANET be protected from severe network attacks.

Furthermore, individual analysis is carried out by varying the kernels in the SVM algorithm. In this case study, we have modified the traffic patterns used for training and testing, as well as the kernels. For each fold, we have employed three kernels: Linear, Polynomial, and RBF, and performance is measured through the F-score. The results are shown in the following figures.

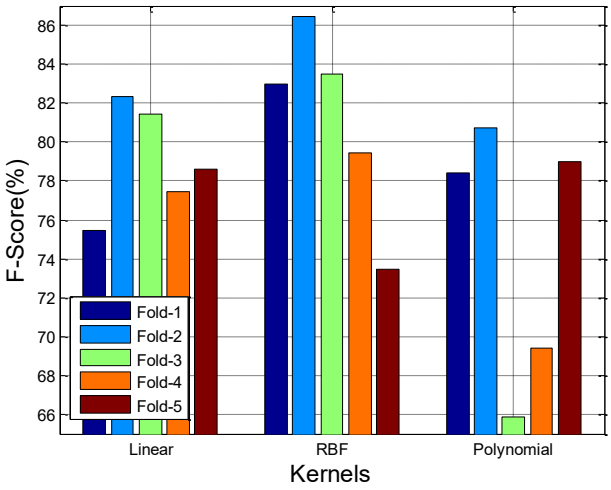

Figure 2: F-Score of Normal Class Under Different Kernels

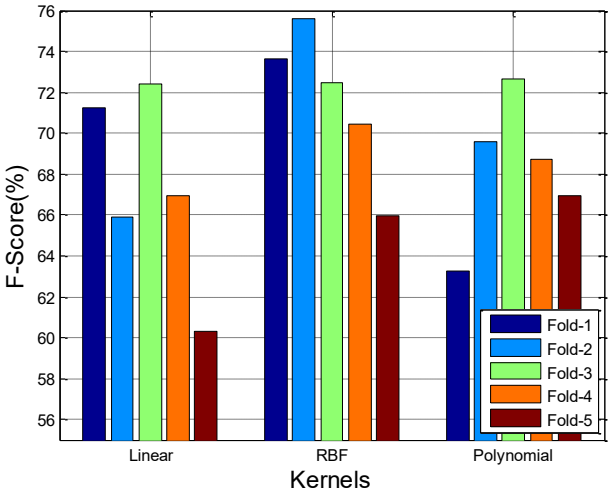

Figure. 3 F-score of Blackhole Attack under different kernels

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Figures 2, 3, and 4 show the F-core analysis for three different categories: standard, black hole, and wormhole attacks. The study is conducted using different kernels and across multiple folds. From the results, the maximum F-score is observed for the RBF kernel, at approximately 86.4842%, 75.6030%, and 74.3433% for normal, blackhole, and wormhole attacks, respectively. Furthermore, the average F-score of the regular class is observed to be 79.0620%, 81.1560%, and 74.6820% for the Linear, RBF, and Polynomial kernels, respectively. Next, the average F-score of the black hole attack class is observed to be 67.3640%, 71.6300%, and 68.2280% for the Linear, RBF, and Polynomial kernels, respectively. Finally, the average F-score of the wormhole attack class is observed to be 66.9460%, 70.4300%, and 66.6280% for linear, RBF, and polynomial kernels, respectively.

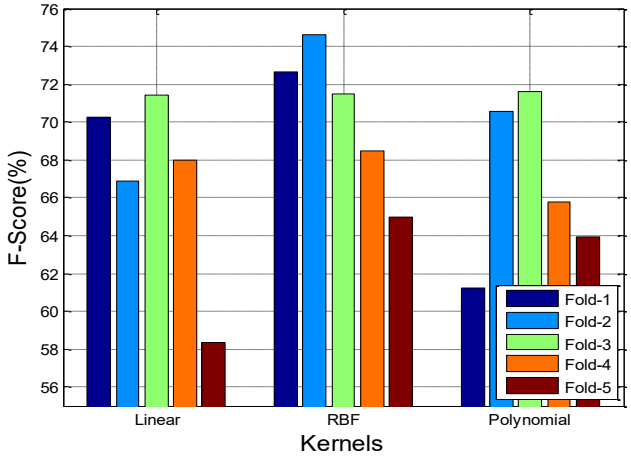

Figure 4: F-score of Wormhole Under Different Kernels

Table 5 shows the confusion matrix of the results obtained after simulating the NSL-KDD dataset. To construct this matrix, we have simulated the traffic connections of 75% of KDD Train+ and KDD Test+. This dataset is also subjected to five-fold cross-validation by exchanging the traffic connections of each class. At every validation, 25% of connections are replaced with new traffic connections in both training and testing sets. The selection of traffic connections is done randomly, and there is no specific criterion for this process. Based on the values shown in Table 5, the performance metrics are calculated and presented in Table 6.

Table 5: Confusion matrix of results from the simulation of KDD Test+ of NSL-KDD dataset

	Normal	DoS	U2R	R2L	probe	Total
Normal	**5826**	925	100	100	332	7283
Dos	705	**4205**	164	136	383	5593
U2R	19	10	**103**	11	7	150
R2L	241	160	200	**1342**	122	2065
Probe	165	105	115	87	**1343**	1815
Total	6956	5405	682	1676	2187	**16906**

Table 6: Performance Metrics of Proposed Method for KDD Test+ of NSL-KDD Dataset

Class/Metric	DR (%)	PPV (%)	FNR (%)	FPR (%)	FAR (%)	F-Score (%)
Normal	79.9900	83.7600	20.0100	16.2400	18.1300	81.8300
DoS	75.1800	77.8000	24.8200	22.2000	23.5100	76.4700
U2R	68.6700	15.1000	31.3300	84.9000	58.1200	24.7600
R2L	64.9900	80.0007	35.0100	19.9300	27.4747	71.7500
Probe	73.9900	61.4100	26.0100	38.5900	32.3000	67.1200

Based on the performance metrics, we can observe that the maximum recall and precision are achieved for the classification of the regular class. Furthermore, among the attack classes, major attacks such as DoS and Probe have achieved nearly equal performance, while minor attacks have shown slightly lower performance. From Table 5.7, the average DR is observed to be 72.5640%, while the average PPV, FPR, FNR, FAR, and F-score are observed to be 63.6141%, 27.4360%, 36.3720%, 31.9069%, and 64.3860%, respectively. Compared to earlier methods that worked on NSL-KDD datasets, the proposed method has not shown encouraging performance because it considered only basic features, which are unable to explore the in-depth discriminative characteristics of attacks. However, to provide support for our self-created dataset, we conducted this simulation.

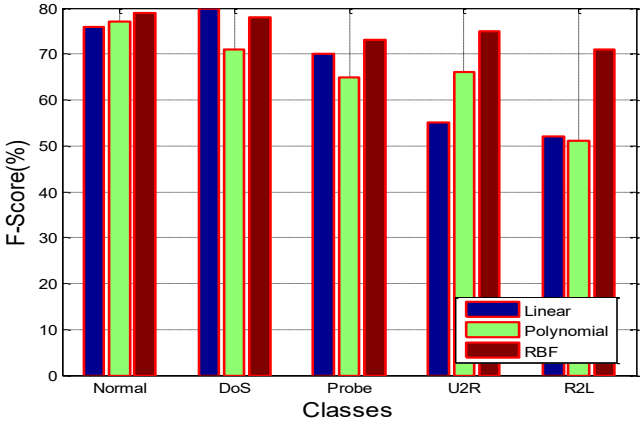

Figure 5: Average F-score of Uder Different SVM Kernels

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Figure 5 shows the comparison between different kernels based on mean F-scores. To perform this simulation, the SVM is employed with three sets of kernels: Linear, RBF, and Polynomial. From the results, we can see that the maximum F-score is achieved using the linear kernel for classifying DoS attacks. The average F-score for the Linear kernel is observed to be 66.6320%, while for the Polynomial and RBF kernels, it is observed to be 67.1240% and 75.2014%, respectively.

V. CONCLUSION

This work primarily aimed to analyse machine learning algorithms in the detection of various security attacks in MANETs. As there is no work employed for detecting malicious nodes in MANETs using machine learning algorithms, we were interested in conducting a simple analysis, which we carried out using packet features acquired from real-time data. The major novelty is the self-created dataset, which was acquired from data packets passing through MANETs. The accomplishment of information processing and machine learning algorithms over the self-created dataset revealed the possibility of deploying machine learning strategies in MANETs for identifying malicious nodes. Based on the simulation studies, we conclude that the IDS is effective for purely static data (created in the past by observing several network traffic), which may or may not work effectively on MANETs. As the mobile nodes are dynamic and the attacks are purely random in nature, a system trained with a constant set of features can't show encouraging detection.ion results

ACKNOWLEDGEMENT

The authors acknowledge the immense help received from the scholars whose articles are cited and included in the references of this paper. The authors are obliged to the authors/editors/publishers of all those articles, journals, and books from which the literature of this paper has been reviewed.

DECLARATION

Funding/ Grants/ Financial Support	No, I did not receive.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, the article does not require ethical approval or consent to participate, as it presents evidence that is not subject to interpretation.
Availability of Data and Material/ Data Access Statement	Not relevant.
Authors Contributions	Conceptualization, software, methodology, validation, formal analysis, investigation, M V D S Krishna Murty; writing-original draft preparation, M V D S Krishna Murty; writing-review and editing, M V D S Krishna Murty, Lakshmi Rajamani

REFERENCES

1. G. A. Walikar, R. C. Biradar, A survey on hybrid routing mechanisms in mobile ad hoc networks, *Journal of Network and Computer Applications*, 77 (2017) 48-63. [CrossRef]
2. G. Dhananjayan and J. Subbiah, "T2AR: trust-aware ad-hoc routing protocol for MANET", *SpringerPlus (2016) 5:995.* [CrossRef]
3. Serhani, N. Naja, and A. Jamali. QLAR: A Q-learning-based adaptive routing for MANETS. *In Computer Systems and Applications*, pages 1–7, 2017. [CrossRef]
4. Raja Mahmood RA, Khan AI, "A Survey on Detecting Black Hole Attack in AODV-based Mobile Ad Hoc Networks, *International Symposium on High Capacity Optical Networks and Enabling Technologies*, Dubai, United Arab Emirates, November 2007. [CrossRef]
5. YIH-CHUN HU, A. PERRIG, D.B. JOHNSON, "WORMHOLE ATTACKS IN WIRELESS NETWORKS", ***IEEE*** *JOURNAL ON SELECTED AREAS IN COMMUNICATIONS,* VOL. 24, ISSUE 2, FEB. 2006, PP.370-380. [CROSSREF]
6. H. J. Liao, C. H. Richard Lin, Y.C. Lin, K. Y. Tung, "Intrusion detection system: a comprehensive review", *J. Netw. Comput. Appl.* 36 (2013) 16–24. [CrossRef]
7. A. Murali, M. Rao, "A survey on intrusion detection approaches", *in: Proc. 1st Int. Conf. Inf. Commun. Technol. ICICT*, 2005, pp. 233–240.
8. C. Kolias, G. Kambourakis, M. Maragoudakis, "Swarm intelligence in intrusion detection: a survey", *Comput. Secur.* 30 (2011) 625–642. [CrossRef]
9. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications, 60*, 19–31 [CrossRef]
10. S.-Y. Ji, B. K. Jeong, S. Choi, D.H. Jeong, "A multi-level intrusion detection method for abnormal network behaviours", *J. Netw.Comput. Appl.* 62 (2016) 9–17. [CrossRef]
11. M.A. Ambusaidi, X. He, P. Nanda, Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm", *IEEE Trans. Comput.* 65 (2016) 2986–2998. [CrossRef]
12. Fei Zhao, Jiyong Zhao, Xinxin Niu, Shoushan Luo and Yang Xin, "A Filter Feature Selection Algorithm Based on Mutual Information for Intrusion Detection", Appl. Sci. 2018, 8, 1535; doi:10.3390/app8091535. [CrossRef]
13. Jingping Song, Zhiliang Zhu, Peter Scully, Chris Price, "Modified Mutual Information-based Feature Selection for Intrusion Detection Systems in Decision Tree Learning", Journal of Computers, Vol. 9, No. 7, July 2014, pp.1542-1546. [CrossRef]
14. Jingping Song, Zhiliang Zhu, and Chris Price, "Feature Grouping for Intrusion Detection Based on Mutual Information," Journal of Communications, vol. 9, no. 12, pp. 987-993, 2014.
15. G. Farahani, "Feature Selection Based on Cross-Correlation for the Intrusion Detection System", *Hindawi Security and Communication Networks*, Volume 2020, Article ID 8875404, 17 pages. [CrossRef]
16. Chun Meng, Yuanyuan Lv, Long You and Yuchen Yue, "Intrusion Detection Method Based on Improved K-Means Algorithm", *IOP Conf. Series: Journal of Physics: Conf. Series* 1302 (2019) 032011. [CrossRef]
17. G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6225 – 6232, 2010. [CrossRef]
18. E. De la Hoz, E. De La Hoz, A. Ortiz, J. Ortega, and B. Prieto, "PCA filtering and probabilistic SOM for network intrusion detection," *Neurocomputing*, vol. 164, pp. 71–81, 2015. [CrossRef]

AUTHORS PROFILE

M V D S Krishna Murty completed B.E. in CSE from the University of Madras, Chennai and M. Tech in CS from JNTU, Hyderabad. Currently, he is a research scholar in the Department of CSE at JNTUH, Hyderabad. His areas of interest are MANETs and Machine Learning. He has 21 years of teaching experience and 5.5 years of industrial experience as a software professional. He has presented technical papers in the area of MANETs and Data Science at international conferences held in India. Additionally, he has published in reputable international journals.

Dr. Lakshmi Rajamani obtained a Ph.D. in Computer Science (CSE) from Jadavpur University, Kolkata, and worked as a Professor and Head in the Department of CSE at OUCE, Osmania University, Hyderabad. Her areas of interest are Fuzzy Logic and Network Security. She has published several papers in reputable international journals. Also, she has presented technical papers in international conferences held in India and abroad.