International Journal of Computer Applications Technology and Research
Volume 13–Issue 04, 18 - 25, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1304.1003

Evaluation of various ML (Machine Learning) algorithms to detect attacks in Mobile Ad hoc Networks (MANETs) via AODV

N. Kanimozhi
Research Scholar,
Department of Computer Science, H.H The Rajah's College, Pudukkotai – 622 001.

Dr. S. Hari Ganesh
Assistant Professor,
Department of Computer Science, H.H The Rajah's College, Pudukkotai – 622 001.

Dr. B. Karthikeyan
Assistant Professor,
Department of Computer Science, Bishop Heber College, Trichy – 620 017.

Abstract: Infrastructure-less networks pose significant challenges in data security and time management, particularly within Mobile Ad-Hoc Networks (MANETs), where unstable mobile nodes can disrupt routing and affect Quality of Service (QoS) metrics. While numerous solutions exist for addressing these challenges, many of them introduce increased Overhead (OH) and Normalized Routing Load (NRL) to the MANET, which is unacceptable given the time-sensitive nature of communication sessions in MANETs. Exceeding processing or transmission times can lead to issues like Link Break (LB). The author proposes a solution to these challenges within the IDS-ATiC-AODV framework (Improved Data Security - Avoiding Time Complexity Ad-Hoc On-Demand Distance Vector). This IDS-ATiC AODV framework addresses five distinct qualitative and quantitative QoS issues using two primary algorithms. This study evaluates various Machine Learning algorithms from different clusters, leveraging the Infrastructure-Less Knowledge Measure Source Dataset (Infra-Less KMS Dataset) for analysis. This involves assessing the accuracy of each Machine Learning algorithm across ten different challenges using the Infra-Less KMS Dataset.

Keywords: MANET, Over Head, Normalized Routing Load, IDS-ATiC AODV, Infra-Less KMS Dataset

1. INTRODUCTION

The Mobile Node Network holds immense importance for the future due to its independence from existing infrastructure, making it the sole solution for military and emergency operations, particularly during natural disasters. Implementing MANETs and network routing presents significant challenges, both qualitatively and quantitatively. Addressing these challenges is crucial for MANETs to become indispensable networks. This research has already addressed Quality of Service (QoS) issues and introduced the IDS-ATiC-AODV algorithm group, capable of resolving various problems like.

- Black Hole Attach (BH)
- Grey Hole Attack (GH)
- Link Break (LB)
- End-to-End Time Delay (EETD)
- Data Theft (DT)(IDS)
- Data Change (DC)(IDS)

When a rule-based situation arises in the network during communication or at an opportune moment, IDS-ITiC promptly responds to mitigate the situation. Occasionally, this rule-based scenario provides pertinent information about ongoing attacks. However, there are instances where such attack-related information is unavailable. In such cases, the solution necessitates the execution of all relevant algorithms, leading to unnecessary processing. This redundancy often prolongs execution time, a scenario that commonly occurs.

MANET functions as a temporary network, establishing connections when a node (the source) seeks communication with its destination. For instance, if a node has a radio transmission range of 500 meters and its intermediate nodes travel at a speed of around 10 meters per second, the network availability from an intermediate node to the source would be less than 2 milliseconds. Thus, for the source node to maintain connectivity, it must complete its communication within this timeframe to avoid potential Link Break (LB) occurrences. Should communication exceed this window, the source node initiates a discovery process to establish a new route.

It is imperative that the solution algorithm does not escalate node overhead (OH) and load (NRL) along the transmission path by enlarging packet size, a responsibility entrusted to the solution algorithms. However, achieving this in practice proves challenging. Each sub-algorithm addressing specific issues necessitates the incorporation of certain parameters to monitor the situation effectively. These parameters are essential for the solution algorithm to identify occurring attacks and determine the appropriate algorithms to execute.

Given these constraints, reducing execution time is practically unfeasible. Therefore, this research aims to develop an automated solution. Specifically, the focus is on predicting attacks that occur at specific times. To achieve this, the author intends to integrate Machine Learning (ML) algorithms for attack prediction. Incorporating ML algorithms requires training them with a dataset, thus necessitating the acquisition of such data for this research.

The research has developed the Infra-Less KMS Dataset for evaluation purposes. Utilizing this dataset, the study will assess various categories of ML algorithms and subsequently recommend the most suitable algorithm for the IDS-ATiC-AODV routing protocol.

2. REVIEW OF LITERATURE

Elife Ozturk Kiyak et al. [1]: Our study introduces a machine learning model, termed high-level k-nearest neighbors (HLKNN), aimed at enhancing predictive analytics. This model enhances the classification capability of the KNN algorithm, commonly utilized across various machine learning domains. Given KNN's susceptibility to irrelevant data, its accuracy is often compromised by the quality of the training dataset. Experimental findings demonstrate that our developed model consistently outperforms KNN on well-known datasets, achieving average accuracy rates of 81.01% and 79.76%, respectively.

Xinhui Zhang et al. [2]: To achieve efficient online spatial data clustering, this research presents a DBSCAN extension algorithm integrating granular models tailored for online clustering. The proposed algorithm, structured into three layers via granular computing, constructs structural granules utilizing DBSCAN and GrC within the input space.

Thao-Trang Huynh-Cam et al. [3]: Experimental results indicate that the mean prediction accuracy rate of RF in 10-fold experiments was approximately 79.99%, DT achieved 74.59% by C5.0 algorithm and 80.00% by CART algorithm, and MLP reached 69.02%. CART surpasses C5.0, RF, and MLP algorithms.

Luca Scrucca [4]: This study introduces a novel approach to initializing the noise component in a Gaussian mixture model, offering an effective methodology for anomaly detection. The proposed approach involves an automatic procedure for selecting initial outlying observations to be used in the EM algorithm for Gaussian mixture models with a noise component. Specifically, noise initialization is based on comparing the contribution of each data point to the entropy of the Gaussian mixture with that arising from a uniform distribution over the hyper-rectangle enclosing the data.

Mohiuddin Ahmed et al. [5]: This paper addresses the popular k-means algorithm and its challenges regarding initialization and the handling of mixed feature types. Through critical analysis of existing literature and experimental assessments on benchmark datasets, the study reveals that each variant of the k-means algorithm is either application-specific or data-specific. Future research aims to develop a robust k-means algorithm capable of addressing both issues concurrently.

Nalindren Naicker et al. [6]: Results indicate that linear support vector machines outperform other methods in predicting student performance using student data. The algorithm suggests that parental education level does not influence student performance, whereas factors such as race, gender, and lunch have an impact. Future endeavors will explore ensemble methods of classical machine learning algorithms to enhance prediction accuracy.

Kumar S et al. [7]: Logistic regression proves to be a valuable tool in medical research for predicting binary outcomes and understanding the influence of predictor variables on patient health. By analyzing coefficients and odds ratios, clinicians can make informed decisions, personalize treatments, and advance medical knowledge, thereby facilitating evidence-based practice and revolutionizing patient care.

Jakub Horak et al. [8]: This study aims to develop bankruptcy prediction models and evaluate results obtained from classification methods, namely Support Vector Machines and artificial neural networks (multilayer perceptron artificial neural networks—MLP and radial basis function artificial neural networks—RBF).

3. MACHINE LEARNING ALGORITHMS

Machine learning algorithms come in a variety of forms, each intended to address a certain problem type and learn from data in a unique way. The following are a few major categories of machine learning algorithms:

1. Supervised Learning

Regression: Makes continuous result predictions.

Classification: Assigns a class or category to provided data.

2. Unsupervised Learning

Clustering is a type of unsupervised learning that puts comparable data points in a group. Reducing the amount of characteristics in the data without compromising its organisation is known as dimensionality reduction. Labelled and unlabelled data are combined for training in semi-supervised learning.

3. Reinforcement Learning:

This method of learning involves making mistakes, interacting with the environment, and getting feedback in the form of incentives or penalties.

4. Deep Learning:

Learns intricate patterns from vast volumes of data by utilising multi-layered neural networks.

5. Transfer Learning:

This method, which usually makes use of trained models, moves knowledge from one activity to another.

6. Ensemble Learning:

This technique, which uses Gradient Boosting or Random Forests, combines several models to increase performance.

7. Anomaly Detection:

Spots anomalies or odd trends in data.

These are only a few of the most common kinds of machine learning algorithms; there are many more particular algorithms with special traits and uses within each category.

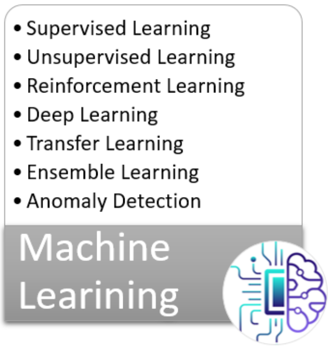

Figure 1: Different type of Machine Learning Algorithms

In this work, the author intends to utilize either supervised or unsupervised learning techniques.

Supervised learning involves training machines using meticulously labeled training data, allowing them to predict outputs based on this information. The labeled data indicates that certain input data is already associated with the correct output.

In supervised learning, the training data acts as a guide for the machines, instructing them on how to accurately predict outputs. This process entails providing both input and corresponding output data to the machine learning model. The objective of a supervised learning algorithm is to establish a mapping function that links the input variable (x) with the output variable (y).

In practical applications, supervised learning finds use in various domains such as risk assessment, image classification, fraud detection, and spam filtering.

During supervised learning, models are trained using labeled datasets, allowing them to familiarize themselves with different types of data. Following the completion of the training process, the model undergoes testing with a separate subset of the training data known as the test set, after which it generates predictions.

The operation of supervised learning can be comprehended through the following example and diagram:

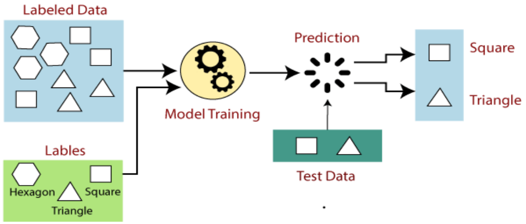

Figure 2: Working of Supervised Learning

Steps Involved in Supervised Learning:

1. Determine the type of training dataset required.
2. Collect or gather the labeled training data.
3. Divide the training dataset into subsets: training dataset, test dataset, and validation dataset.
4. Identify the input features within the training dataset, ensuring they provide sufficient information for accurate output prediction.
5. Select a suitable algorithm for the model, such as support vector machines, decision trees, etc.
6. Implement the algorithm on the training dataset, occasionally requiring validation sets for control parameters.
7. Assess the model's accuracy by evaluating its performance on the test set. A correct output prediction indicates model accuracy.

Types of Supervised Machine Learning Algorithms:

Supervised learning can be further categorized into two types of problems:

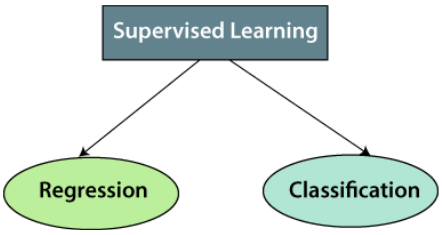

Figure 3: Major classification of Supervised Learning

1. Regression:

Regression algorithms are employed when a relationship exists between the input and output variables. They predict continuous variables, such as weather forecasting and market trends. Below are some prominent regression algorithms falling under supervised learning:

- Linear Regression
- Regression Trees
- Non-Linear Regression
- Bayesian Linear Regression
- Polynomial Regression

2. Classification:

Classification algorithms come into play when the output variable is categorical, defining two classes like Yes-No, Male-Female, True-False, etc. They are utilized in applications such as spam filtering. Popular classification algorithms in supervised learning include:

- Random Forest
- Decision Trees
- Logistic Regression
- Support Vector Machine

3.1. Unsupervised Machine Learning

Unsupervised learning is a machine learning approach where models are not provided with a labeled training dataset. Instead, these models autonomously uncover hidden patterns and insights within the given data. It's akin to how the human brain learns new concepts without explicit instruction.

In essence, unsupervised learning entails training models with unlabeled datasets and empowering them to glean insights without external guidance. Unlike supervised learning, where input data is paired with corresponding output data, unsupervised learning operates solely on input data.

The primary objective of unsupervised learning is to discern the underlying structure of a dataset, group similar data points together, and represent the dataset in a more compact format.

The operation of unsupervised learning can be illustrated through the following diagram:

Clustering: Clustering involves grouping objects into clusters based on their similarities, where objects within a cluster share more similarities with each other than with objects in other clusters. Cluster analysis identifies commonalities among data objects and organizes them based on the presence or absence of these commonalities.

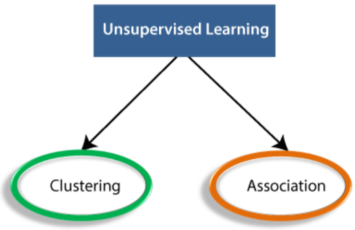

Figure 5: Major classification of Unsupervised Learning

Association: Association is an unsupervised learning technique used to discover relationships between variables within a large dataset. It identifies sets of items that frequently co-occur in the data, aiding in the formulation of effective marketing strategies. For instance, association rules can reveal patterns such as customers who purchase item X (e.g., bread) are also likely to buy item Y (e.g., butter/jam). Market Basket Analysis is a classic example of association rule usage.

Here is a compilation of some well-known unsupervised learning algorithms:

- K-means clustering
- K-nearest neighbors (KNN)
- Hierarchical clustering
- Anomaly detection
- Neural Networks
- Principle Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Apriori algorithm
- Singular value decomposition (SVD)

4. DATASET

The upcoming dataset to be utilized in this study is the Infra-Less KMS Dataset, comprising approximately 31 attributes. These attributes are employed to analyze eight distinct factors encompassing both Qualitative and Quantitative Quality of Service (QoS) metrics. These factors include:

- DoS & DDoS
- Data Change
- Data Theft
- Block Hole
- EETD
- PDR
- (BU) Bandwidth Utilization
- NC(Network Congestion)
- UN(Unbelievable Node)

The following list presents the attributes contained within the Infra-Less KMS Dataset

- Com_ID: An exclusive communication identifier structured as SourceIP-DestinationIP-SourcePort-DestinationPort-TransportProtocol.
- IP-Source: The source IP address of the communication.
- Service-Source: The source port number.
- IP-Destination: The destination IP address.
- Service-Destination: The destination port number.
- Pro-ID: Identification number for the transport layer protocol (e.g., TCP = 6, UDP = 17).
- T_Instant: Timestamp indicating the capture time of the packet, in DD/MM/YYYY HH:MM:SS format.
- Com_Session: Total duration of the communication.
- Log_Fail: Number of failed login attempts.
- Log_Success: Binary indicator (1 if successfully logged in, 0 otherwise).
- Operation_Access_Control: Number of operations on access control files.
- SH_No_Of_Connection: Number of connections to the same host as the current connection within the past 2 seconds.
- Cur_Con_Ser_No.Of_Connection: Number of connections to the same service as the current connection within the past two seconds.
- Per_of_Connection_Diff_Hosts: Percentage of connections to different hosts.
- No_of_Same_Connection: Count of connections with the same destination host and service.
- Per_of_Connection_Diff_Hosts_Same_Ser: Percentage of connections to the same service originating from different hosts.
- No_Of_FW_Pkts: Total number of packets in the forward direction.
- No_Of_BW_Pkts: Total number of packets in the backward direction.
- Len_FW_Pkts: Total bytes in the forward direction across all packets in the communication.
- Len_BW_Pkts: Total bytes in the backward direction across all packets in the communication.
- Len_FW_Pkt_Max: Maximum packet length in bytes in the forward direction.
- Len_FW_Pkt_Min: Minimum packet length in bytes in the forward direction.
- Len_FW_Pkt_Mean: Mean packet length in bytes in the forward direction.
- Len_FW_Pkt_SD: Standard deviation of packet lengths in bytes in the forward direction.
- Len_BW_Pkt_Max: Maximum packet length in bytes in the backward direction.
- Len_BW_Pkt_Min: Minimum packet length in bytes in the backward direction.

- Len_BW_Pkt_Mean: Mean packet length in bytes in the backward direction.
- Len_BW_Pkt_SD: Standard deviation of packet lengths in bytes in the backward direction.
- Com_BPS: Bytes per second in the communication.
- Com_PktPS: Packets per second in the communication.
- One_Hop Neighbour: Updated when neighboring changes occur, structured as Source node-one hop node.

The table below provides details of the dataset including attack information.

Table 1: Display the number of rows allocated for each attack

Total Number of Rows	2,11,010.00
DoS (Denial of Service) & DDoS (Distributed Denial of Service)	32,021.00
DC (Data Change)	10,165.00
DT (Data Theft)	12,010.00
BH (Block Hole)	35,202.00
EETD (End to End Time Delay)	50,151.00
PDR (Packet Delivery Ratio)	42,435.00
BU (Bandwidth Utilization)	9,682.00
NC(Network Congestion)	9,669.00
UN(Unbelievable Node)	9,675.00

5. PROPOSED EVALUATION

The following algorithms are intended for use in this evaluation:-

1. Gaussian Naïve Byes (GNB)
2. Logistic Regression (LR)
3. K-Nearest Neighbours (KNN)
4. Linear Support Vector (LSV)
5. Decision Tree (DT)
6. Random Forest (RF)
7. Support vector Machines (SVM)
8. Gaussian Mixture Model (GMM)
9. K-Means (KM)
10. DBSCAN

Test dataset = 80% =1, 68, 810 rows

Train dataset =20% =42, 200 rows

5.1. Data Pre-processing

In this data pre-processing, the following tasks were performed: handling missing data, removing duplicates, identifying and handling outliers, encoding categorical labels, analyzing correlations, selecting relevant features, scaling the data, and splitting it into appropriate subsets.

Missing Data finding is done by the following code

```
total = train.shape[0]
missing_columns = [col for col in train.columns if train[col].isnull().sum() > 0]
for col in missing_columns:
    null_count = train[col].isnull().sum()
    per = (null_count/total) * 100
    print(f"{col}: {null_count} ({round(per, 3)}%)")
```

Duplicates is found following code.

```
train.duplicated().sum()
```

5.2. Feature selection

The subsequent code is utilized for feature selection in random forest classifiers.

```
rfc = RandomForestClassifier()
rfe = RFE(rfc, n_features_to_select=10)
rfe = rfe.fit(X_train, Y_train)
feature_map = [(i, v) for i, v in itertools.zip_longest(rfe.get_support(), X_train.columns)]
selected_features = [v for i, v in feature_map if i==True]
```

The provided code serves as a sample for training a Gaussian Naive Bayes (GNB) model on the BH (Black Hole) dataset

```
from sklearn.naive_bayes import GaussianNB
NB = GaussianNB()
NB.fit(X_train,y_train)
predictions = NB.predict(X_train)
print_stats(predictions, X_train, y_train, "Gaussian Naive Bayes on the train set")
```

6. RESULT AND DISCUSSION

Attack prediction percentage is listed in the following table. In this work around ten different attacks were predicted by the use of ten different Machine Learning algorithms. For the prediction it uses Infra-Less KMS Dataset. This dataset has around 2 lack records. Following table shows prediction accuracy with different attacks and different ML algorithms.

The preceding figure illustrates the prediction accuracy for attacks using Gaussian Naïve Bayes. According to these results, the algorithm's prediction accuracy falls within the range of 90% to 93%. While this range suggests promising accuracy, further validation against real-world scenarios is necessary to confirm its predictive capability.

Moreover, Gaussian Naïve Bayes exhibits notably low prediction accuracy of 90% for Data Theft and Packet Delivery Ratio. Conversely, it demonstrates favorable prediction accuracy for Data Change and End-to-End Time Delay, as indicated by these results.

The following figure shows the prediction accuracy with Logistic Regression. This algorithm gives high accuracy with Packet Delivery Ratio, Data Theft and Unbelievable Node, it gives 96% accuracy and it give low prediction accuracy 94% in Data change

Packet Delivery Ratio, Data Theft, and Unliveable Node exhibit higher accuracy rates. Conversely, Logistic Regression demonstrates lower prediction accuracy for Data Change.

Table 2: Prediction Accuracy between different ML Models

Attacks	Accuracy (%)									
	GNB	LR	KNN	LSV	DT	RF	SVM	GMM	KM	DBSCAN
PDR	90	96	99	97	99	100	92	81	69	29
EETD	93	95	99	95	65	98	78	74	70	52
DoS	92	95	99	96	52	80	85	77	69	41
DDoS	92	95	99	96	53	80	85	77	69	42
DC	93	94	35	94	48	65	71	67	65	48
DT	90	96	37	97	52	73	94	86	73	51
BH	92	95	52	96	99	99	83	76	69	51
BU	92	95	98	96	67	100	83	77	69	37
NC	91	95	99	95	100	100	80	73	67	39
UN	92	96	99	96	100	100	86	79	72	45

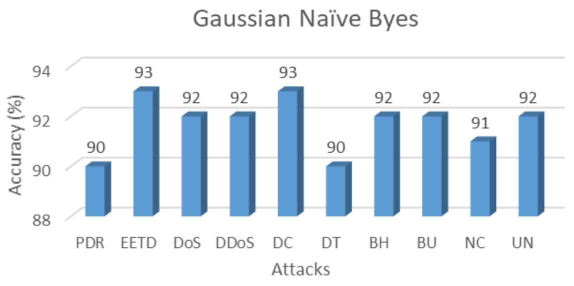

Figure 6: Prediction Accuracy between different attacks using Gaussian Naïve Byes

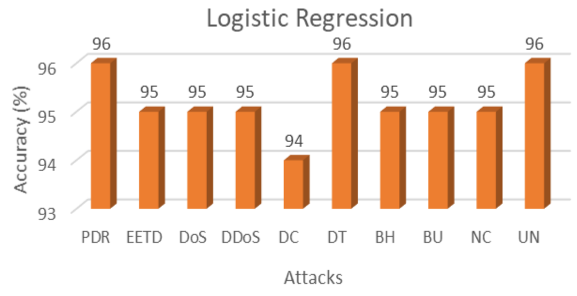

Figure 7: Prediction Accuracy between different attacks using Logistic Regression

The figure 8 above displays the prediction accuracy results for K-Nearest Neighbours, whereas the figure 9 below depicts the outcomes for Linear Support Vector. In comparison, Linear Support Vector yields moderate results.

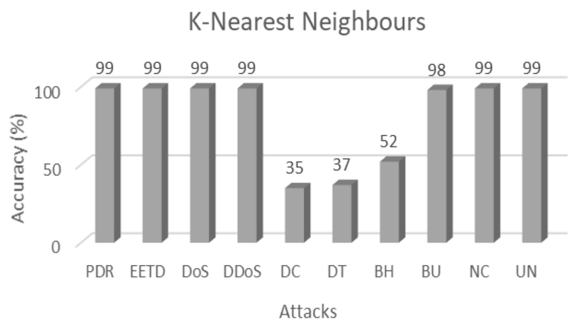

Figure 8: Prediction Accuracy between different attacks using K-Nearest Neighbors

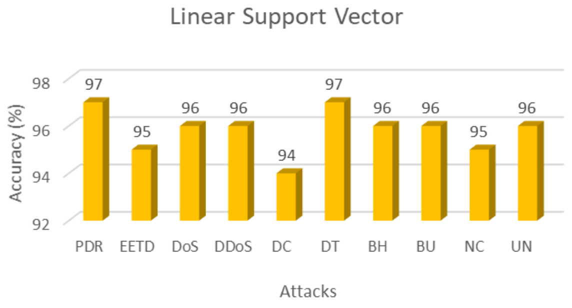

Figure 9: Prediction Accuracy between different attacks using Linear Support Vector

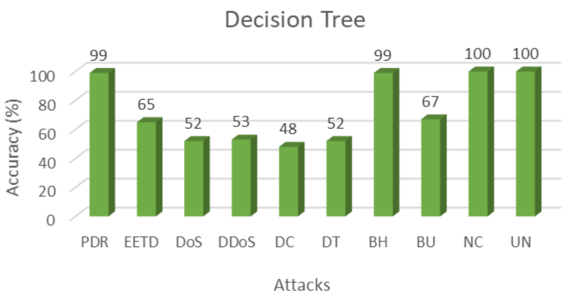

Figure 10: Prediction Accuracy between different attacks using Decision Tree

Decision Tree prediction accuracy is shown in the figure 10. Figure 11 shows Random Forest prediction accuracy. Random Forest gives moderate accuracy.

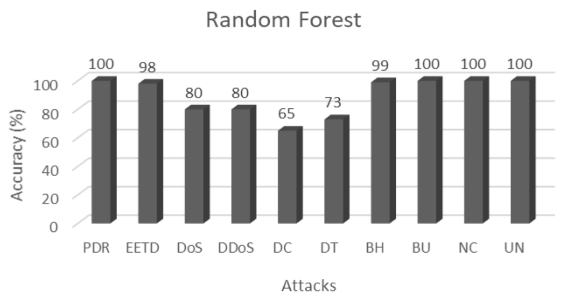

Figure 11: Prediction Accuracy between different attacks using Random Forest

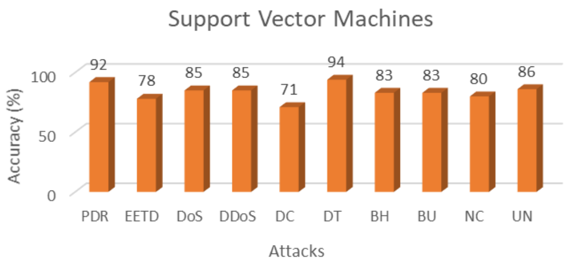

Figure 12: Prediction Accuracy between different attacks using Support Vector Machines

Figure 12 above indicates moderate prediction accuracy across all attacks, with rates ranging between 71% and 94%. This moderate level of accuracy, neither excessively high nor low, suggests its suitability for implementation in the IDS-ATiC-AODV routing protocol.

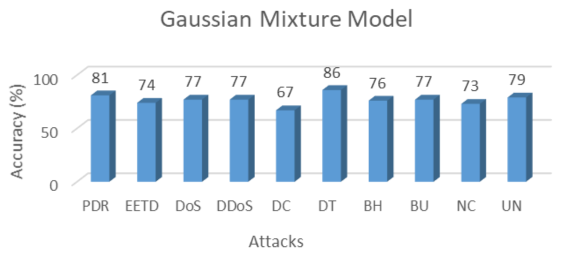

Figure 13: Prediction Accuracy between different attacks using Gaussian Mixture Model

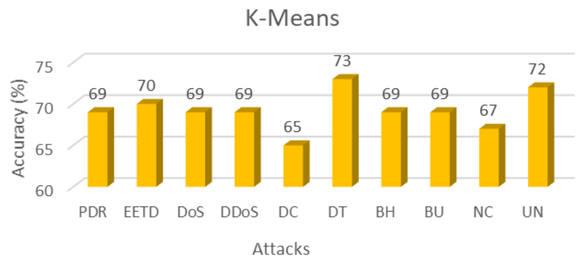

Figure 14: Prediction Accuracy between different attacks using K-Means

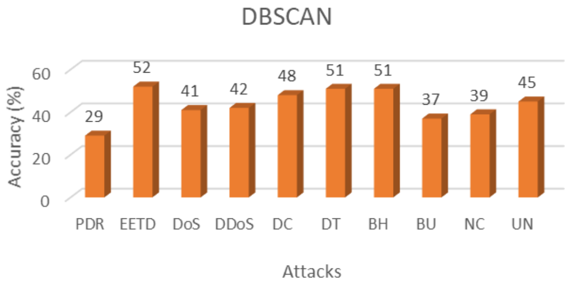

Figure 15: Prediction Accuracy between different attacks using DBSCAN

Figures 13, 14, and 15 depict the prediction accuracy for Gaussian Mixture Model, K-Means, and DBSCAN respectively. Notably, DBSCAN yields lower accuracy compared to Gaussian Mixture Model, which exhibits higher accuracy. However, the predictions from these three algorithms do not align with those from SVM.

7. CONCLUSION

Ten different ML algorithms accuracy prediction is evaluated with then different attacks by the use of Infra-Less KMS Dataset. This dataset has around 2 lack samples, in it 20% is used for training and remaining 80% of samples is used for testing. According to the evaluation some of the ML algorithms gives higher and lower prediction accuracy. But this research needs only on the moderate prediction accuracy giving ML Algorithms. So SVM is the moderate prediction accuracy provide ML algorithm. In the next work it will be using in the implementation with the IDS-ATiC-AODV routing protocol.

8. REFERENCES

[1] Elife Ozturk Kiyak, Bita Ghasemkhani and Derya Birant, "High-Level K-Nearest Neighbors (HLKNN): A Supervised Machine Learning Model for Classification Analysis",Electronics, 12, 3828, September 2023.

[2] Xinhui Zhang, Xun Shen, and Tinghui Ouyang, "Extension of DBSCAN in Online Clustering: An Approach Based on Three-Layer Granular Models", Appl. Sci., 12, 9402. September 2022.

[3] Thao-Trang Huynh-Cam, Long-Sheng Chen, and Huynh Le, "Using Decision Trees and Random Forest Algorithms to Predict and Determine Factors Contributing to First-Year University Students' Learning Performance", Algorithms, 14, 318. October 2021.

[4] Luca Scrucca, "Entropy-Based Anomaly Detection for Gaussian Mixture Modeling", Algorithms, 16, 195.April 2023.

[5] Mohiuddin Ahmed, Raihan Seraj and Syed Mohammed Shamsul Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation", Electronics, 9, 1295, August 2020.

[6] Nalindren Naicker, Timothy Adeliyi, and Jeanette Wing, "Linear Support Vector Machines for Prediction of Student Performance in School-Based Education", Hindawi Mathematical Problems in Engineering Volume 2020, Article ID 4761468,October 2020.

[7] Kumar S, Gota V., "Logistic regression in cancer research: A narrative review of the concept, analysis, and interpretation", Cancer Research, Statistics, and Treatment 2023;6:573-8,Dec-2023.

[8] Jakub Horak, Jaromir Vrbka and Petr Suler, "Support Vector Machine Methods and Artificial Neural Networks Used for the Development of Bankruptcy Prediction Models and their Comparison", Risk Financial Manag. 2020, 13, 60, March 2020.

[9] N. Kanimozhi, S. Hari Ganesh 2, B. Karthikeyan, "An Analysis of Machine Learning Solution for QoS and QoE in Network (Infrastructure Oriented and Less)", International Journal of Computer Sciences Engineering, Vol.11, Issue 5, pp.41-59, May 2023, ISSN: 2347-2693 (Online), PP 41-59

[10] N. Kanimozhi, S. Hari Ganesh 2, B. Karthikeyan, “Performance Analysis of MANET Routing Protocols”, International Journal of Computer Applications (0975 – 8887) Volume 185 – No. 50, December 2023, PP-44-50

[11] N. Kanimozhi, S. Hari Ganesh 2, B. Karthikeyan, "Irregularity Behaviour Detection - Ad-hoc On-Demand Distance Vector Routing Protocol (IBD - AODV): A Novel Method for Determining Unusual Behaviour in Mobile Ad-hoc Networks (MANET)", International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 11 Issue: 9, September 2023 pp 1098-1010.

[12] N. Kanimozhi, S. Hari Ganesh 2, B. Karthikeyan, "Minimizing End-To-End Time Delay in Mobile Ad-Hoc Network using Improved Grey Wolf Optimization Based Ad-Hoc On-demand Distance Vector Protocol (IGWO-AODV)", International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 11 Issue: 9, September 2023 pp 1111-1115.

[13] B.Karthikeyan, N. Kanimozhi and Dr.S.Hari Ganesh, “Analysis of Reactive AODV Routing Protocol for MANET”, IEEE Xplore (978-1-4799-2876-7), Oct 2014, pp. 264-267.

[14] B.Karthikeyan, N. Kanimozhi and Dr.S.Hari Ganesh, “Security and Time Complexity in AODV Routing Protocol”, International Journal of Applied Engineering Research (ISSN:0973-4562), Vol. 10, No.20,June 2015, pp.15542- 155546. – Scopus Indexed

.

[15] B. Karthikeyan, Dr.S.Hari Ganesh and N. Kanimozhi, “Encrypt - Security Improved Ad Hoc On Demand Distance Vector Routing Protocol (En-SIm AODV)”, ARPN Journal of Engineering and Applied Sciences (ISSN: 1819-6608), Vol. 11, No. 2, January 2016,pp. 1092-1096.

[16] B. Karthikeyan,Dr.S.Hari Ganesh and Dr. JG.R. Sathiaseelan, “ Optimal Time Bound Ad-Hoc On-demand Distance Vector Routing Protocol (OpTiB-AODV)”, International Journal of Computer Applications (ISSN:0975 – 8887), Vol. 140, No.6, April 2016,pp 7-11.

[17] B. Karthikeyan, Dr.S.Hari Ganesh, Dr. JG.R. Sathiaseelan and N. Kanimozhi , “High Level Security with Optimal Time Bound Ad-Hoc On-demand Distance Vector Routing Protocol (HiLeSec-OpTiB AODV)”,International Journal of Computer Science Engineering(E-ISSN: 2347-2693),Vol. 4, No. 4, April 2016, pp.156-164.