**Title of the Paper
(16 pt, Bold, Times New Roman Font, Title Case Style)**

**Author1, Author2 and Author3(10pt bold Times New Roman Font)**

*1,2 Ph.D., Students, School of Computer and IT, Taylor’s University, Malaysia.
3Assistnat Professor, School of Computer and IT, SEGi University, Malaysia.
(10pt, Italic, Times New Roman Font)*

 **[All fonts of the Document in 10pt, Times new Roman font. ]**

**Abstract (bold)**

Machine learning (ML) is used for network intrusion detection because of its prediction ability after training with relevant data. ML provides a good method to detect new and unknown attacks. There are many types of attacks in network intrusion: however, this paper

**Keywords: (bold)** Classification, Distributed Denial of Service,

**I. INTRODUCTION**

**(Section must type in uppercase 10pt, bold)**

In network intrusion detection system (NIDS) research, there are three types of detection approaches misused or signature-

**II. RELATED WORK**

**(Section must type in uppercase 10pt, bold)**

Extensive research has been done on intrusion detection system

Averaged one dependence estimators (AODE) can provide good accuracy when performing binary classification. In addition, their training and testing times has been found to be relatively fast compared to other popular algorithms [7].



**Fig. 1.** Classification of ML algorithms

(abject figure in single column if large)

Lazy learning methodologies have been studied to overcome the limitation of eager learning methods. It has been argued that eager learning methods contribute to losess in performance efficiency when trying to generalise training data prior to receiving queries. This causes an unnecessary use of computational overhead. To overcome this, the use a heuristic weight-based indexing was proposed with a lazy method to overcome the high search complexity that is normally associated with lazy methods [9]. This study used kNN with cross validation as its lazy method.

**Table 1.** Summary of the performance of ML algorithms

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Reference** | **[9]** | **[8]** | **[10]** | **[4]** | **[7]** | **[11]** |
| Algorithm | FS |  |  | CCA(small) | CCA(big) | LDA(small) | LDA(big) |  |  | CFS | PCA | IGR | MRMR |
| kNN | 92.30 | 91.07 |  |  |  |  |  |  | 97.65 | 98.87 | 99.07 | 98.05 |
| SVM |  | 91.19 |  |  |  |  | 98.76 |  | 76.61 | 96.78 | 94.39 | 88.93 |
| NB |  |  | 58.44 | 59.55 | 57.93 | 55.50 |  | 75.73 | 82.66 | 89.91 | 90.26 | 87.56 |
| BN |  |  |  |  |  |  |  | 92.70 |  |  |  |  |
| DT |  |  |  |  |  |  |  |  | 98.99 | 98.95 | 97.83 | 98.78 |
| NN |  |  |  |  |  |  |  |  | 83.80 | 97.50 | 97.70 | 94.60 |

 (abject figure in single column if large)

**III. METHOD**

**(Section must be bold & type in uppercase 10pt, bold)**

This paper compared the performances of eight ML algorithms using the CICIDS2017 dataset. Steps taken are as shown in Fig. 2. ML algorithms were used to build Model, as in the diagram. Three important performance metrices [20] were used for the comparison; namely accuracy, true positive rate and false alarm rate.

**III.I Dataset**

**(Sub-section must be bold and type in title case 10pt, bold)**

We used the CICIDS2017 dataset which was created by the Canadian Institute for Cybersecurity for network security and

hardware specifications of the laptop were Intel i5 CPU, 8Gb RAM, and Intel HD Graphics 3000.

***III.II.I Pre-processing***

***(sub-sub-section must be bold & type in italic 10pt, bold)***

Raw data normally include many imperfections such as

***1) Missing Data***

Missing data include empty values or values not compatible with the data format. For example, features with numerical formats must consist of numbers only, and cannot include any

***2) Feature Selection***

All the features that have min, max, mean and std (standarad deviation) values were removed except the mean value of the

Accuracy: $\frac{TN+TP}{TN+TP+FP+FN}$ (1)

***(Equation number align right-side)***

**IV. RESULT**

We used various ML algorithms, and their results are shown in Table 3.

**Table 2.** Confusion matrix table

|  |  |  |
| --- | --- | --- |
|  |  | Predicted Class |
|  |  | Negative (Normal) | Positive (Attack) |
| Actual Class | Negative (Normal) | True Negative (TN) | False Positive (FP) |
| Positive (Attack) | False Negative (FN) | True Positive (TP) |

**V. DISCUSSION**

It appears that the dataset used, CICIDS2017, is well suited for DT algorithm and its derivatives, such as BT and random forest. Random forest produced the best result with the

**VI. CONCLUSION**

This paper aims to compare nine supervised algorithms’ performance towards DDoS intrusion. DDoS attack will result

**REFERENCES**

1. Nisioti A, Mylonas A, Yoo PD, Member S, Katos V. From Intrusion Detection to Attacker Attribution : A Comprehensive Survey of Unsupervised Methods. IEEE Commun Surv Tutorials. 2018;PP(c):1.
2. Hafizah S, Ariffin S, Muazzah N, Latiff A, Khairi MHH, Ariffin SHS, et al. A Review of Anomaly Detection Techniques and Distributed Denial of Service (DDoS) on Software Defined Network (SDN). Technol Appl Sci Res [Internet]. 2018;8(2):2724–30.