Indian Institute of Information Techonology, Kalyani

IDS Using Deep Learning

Under the supervision of

Dr. SK Hafizul Islam

May 15, 2023

Team Members

Purvansh Sonthalia (586)
Somesh Kumar (602)
Vemana Joshua Immanuel (620)

Recap of Previous Works

We Have explored various domains in IDS

- IDS in Smart meter.
- IDS in CAN Bus
- NIDS

IDS in Smart Meters

unrolling

- We found that the current studies are using ML based model which are not very efficient.
- We have tried to implement it using Deep learning.
- But we faced the problem that it was not able to detect new kinds of attact.
- To overcome this we need to train our network but we dont have enough data set to do it.
- We need good resources to train deep learning models

IDS in CAN Bus

- We used supervised learning to detect anomaly in a car
- We got accuracy of 98% using Decision Tree
- Due to small dataset we could not extend this idea.

NIDS

- A NIDS is a type of security system that monitors network traffic for suspicious or malicious activity.
- NIDS can help organizations to detect and respond to cyber attacks, protect sensitive data.
- NIDS can be deployed at various points in the network architecture.

What we did this Semester?

- Literature survey
- Model comparison
- Model Tolerance to noise
- Conclusion

Introduction

- Studied various research papers
- Learned about various type of attack in network
- Analysed advantages and disadvantages of various models
- Understood the importance of Feature selection, Feature extraction and Feature engineering .
- Further extended the observation of models for noisy data.

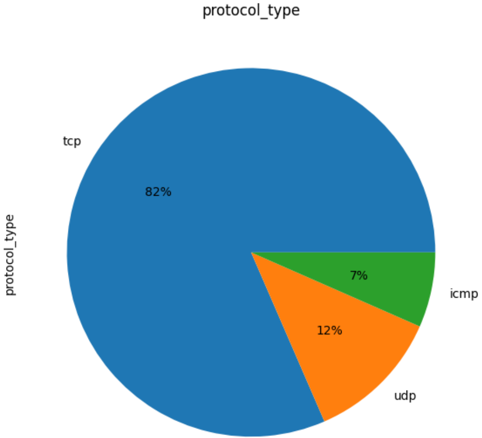
Using standalone Algorithm

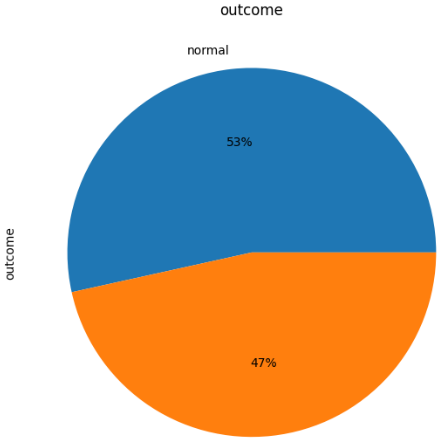
- In [2] ,worked on wireless NIDS and used various classification algorithm, found that Random forest with 32 features gives the best result with accuracy of 99.64
- In [1], worked on UNSW-NB 15 dataset and used classification algorithm such as SVM, NB, DT, RF and found that random forest gives the best accuracy of 97.49%

Using combination of 2 or more algorithms

- In [3], it used Combine regression tree and random forest on UNSW-NB 15 data set and it gave an accuracy of 87.76
- In [5] [4], it used Naive Bayes and Support vector machine (SVM) on NSL KDD and CICIDS 2017 data set and found that the accuracy was 93.36%, 92.56% respectively.

NSL KDD Dataset





Models Used

- Logistic Regression
- Naive Bayes
- Support vector machine
- Decision Tree
- Random Forest
- Artificial Neural Network
- Random Forest using PCA

Logistic Regression Model (Base Line Model)

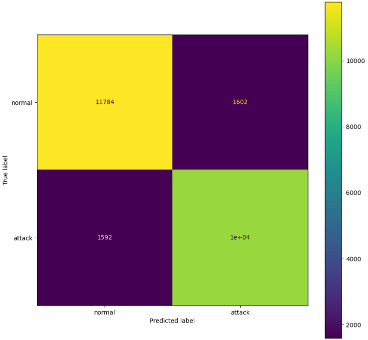


Figure: Confusion matrix of Logistic Regression Model

Left: Both false positive and true negative are high

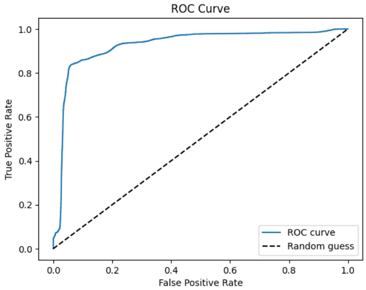


Figure: ROC of Logistic Regression

Naive Bayes Model (Base Line Model)

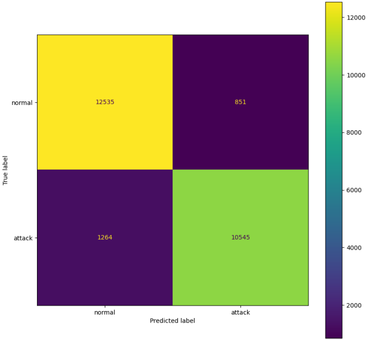


Figure: Confusion matrix of Naive Bayes

Left: Relatively Better than Logistic regression

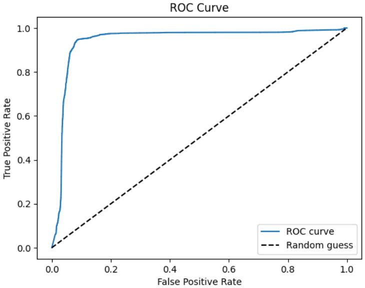


Figure: ROC of Naive Bayes

SVM Model

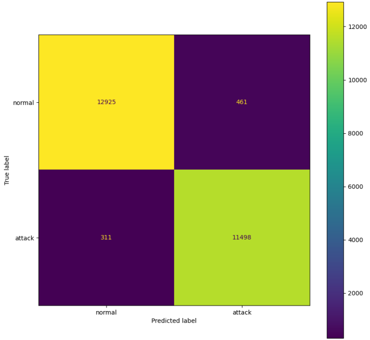


Figure: Confusion matrix of SVM

Left: Relatively Better than Base Line Models but takes more time

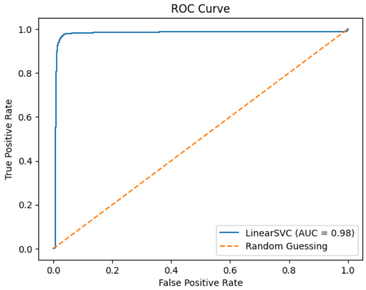


Figure: ROC of SVM

Decision Tree Model

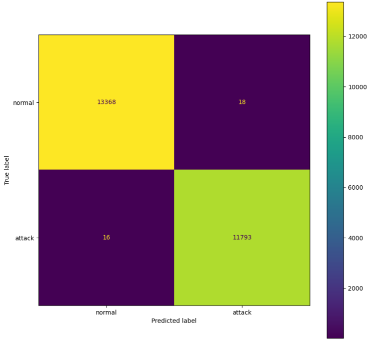


Figure: Confusion matrix of Decision Tree

Left: Accuracy is high and less training time

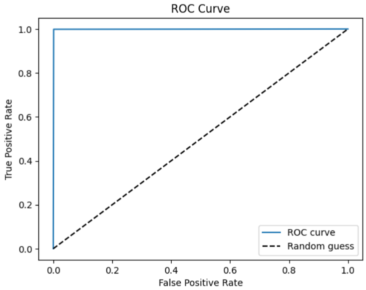


Figure: ROC of Decision Tree

Decision Tree Model

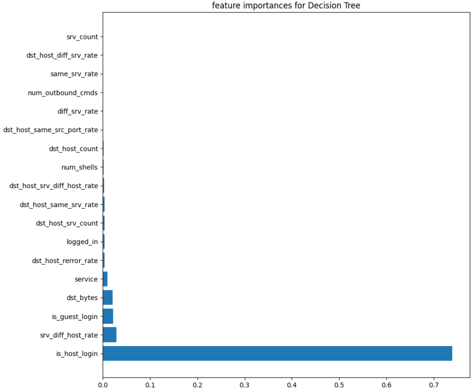


Figure: Feature Importance of Decision

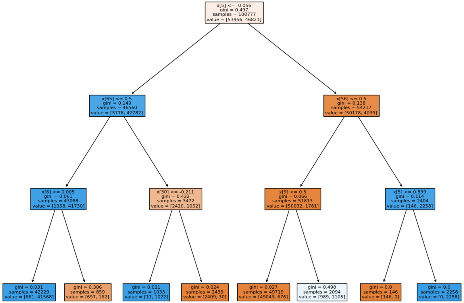


Figure: Decision Tree

Random Forest Model

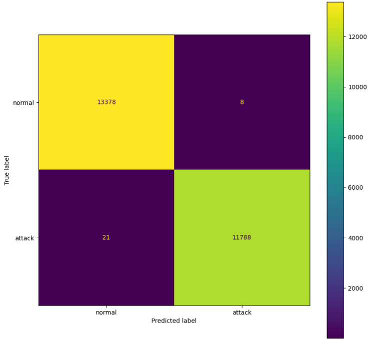


Figure: Confusion matrix of Random Forest

Left: Relatively similar performance to Dicision Tree

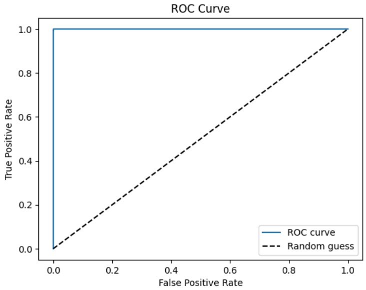


Figure: ROC of Random Forest

Random Forest

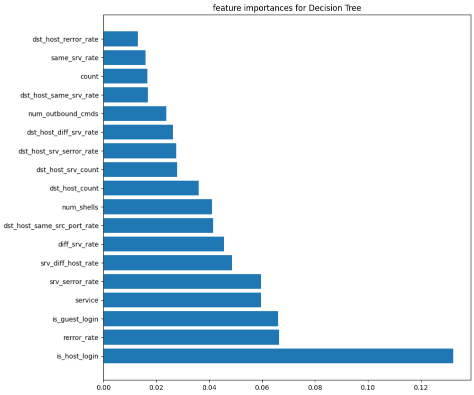


Figure: Feature Importance of Random

Left: Every Feature is given more importance than Decision Tree

Using Neural Network

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_10 (Dense)            (None, 64)                7872

 dropout_8 (Dropout)         (None, 64)                0

 dense_11 (Dense)            (None, 128)               8320

 dropout_9 (Dropout)         (None, 128)               0

 dense_12 (Dense)            (None, 512)               66048

 dropout_10 (Dropout)        (None, 512)               0

 dense_13 (Dense)            (None, 128)               65664

 dropout_11 (Dropout)        (None, 128)               0

 dense_14 (Dense)            (None, 1)                 129

=================================================================
Total params: 148,033
Trainable params: 148,033
Non-trainable params: 0
```

Figure: Artificial Neural Network Model

Left: Activation function used is Relu and Sigmoid in the last layer

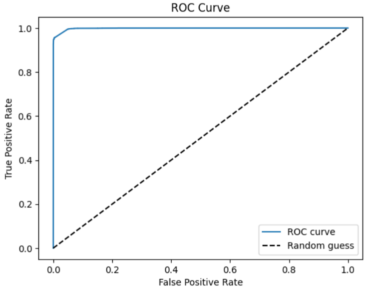


Figure: ROC of ANN

Neural Network Accuracy

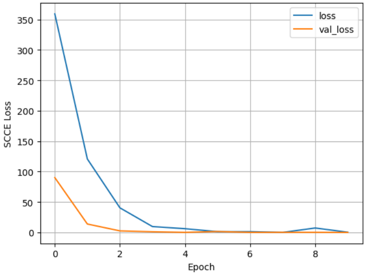


Figure: Loss v/s Epochs

Left: Accuracy of About 98% for 10 Epochs

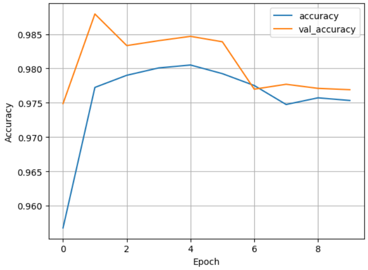


Figure: Accuracy v/s Epochs

Comparison among models

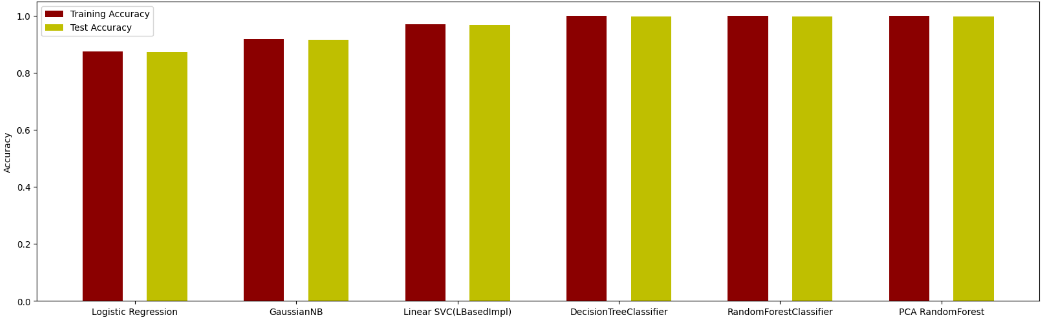
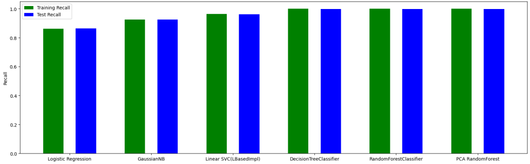
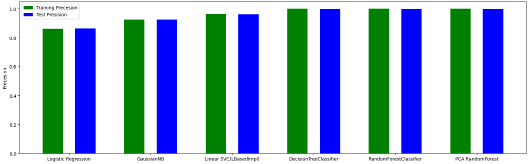


Figure: Training and Test Accuracy

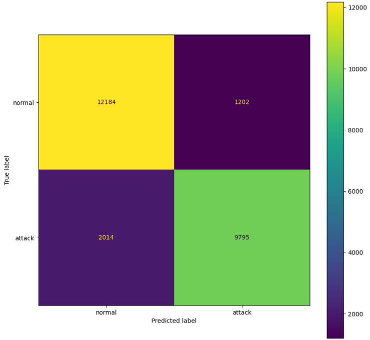
Precision and Recall of models



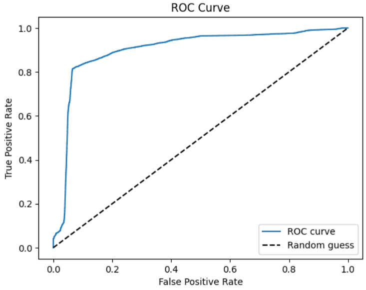


Data with noise level-1

Logistic Regression Model (Base Line Model)



Left: Accuracy decreases due to noisy data



Naive Bayes Model

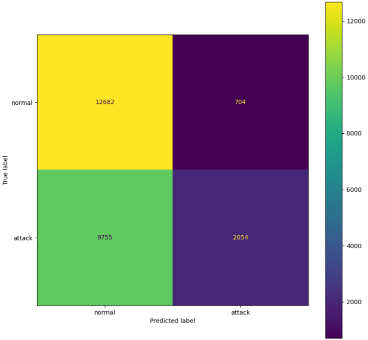


Figure: Confusion matrix of Naive Bayes

Left: Worst performance by Naive Bayes due to noise

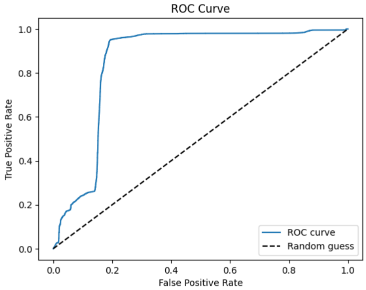


Figure: ROC of Naive Bayes

SVM Model

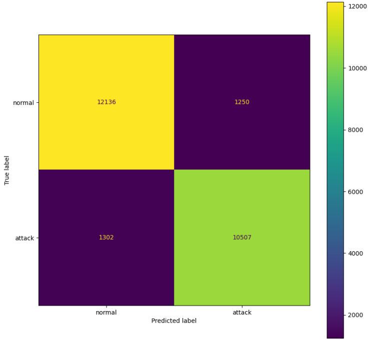


Figure: Confusion matrix of SVM

Left: A bit more better than previous two model

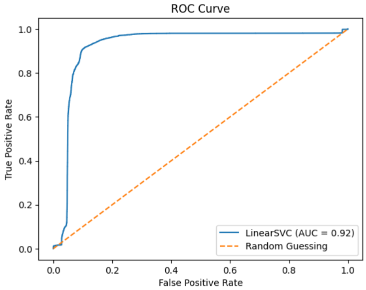


Figure: ROC of SVM

Decision Tree Model

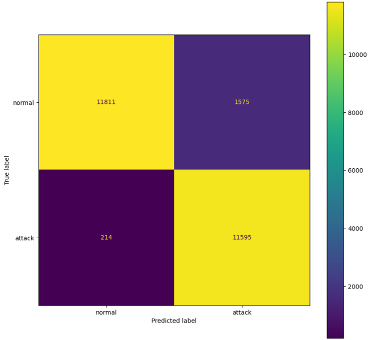


Figure: Confusion matrix of Decision Tree

Left: Predicts attack as normal in most cases due to noise

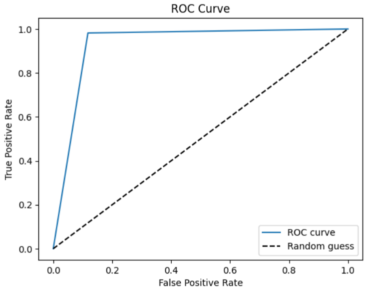


Figure: ROC of Decision Tree

Decision Tree Model

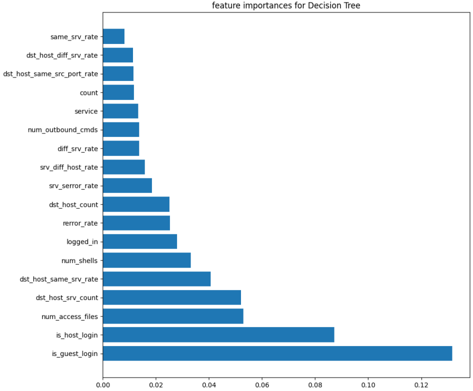


Figure: Feature Importance of Decision

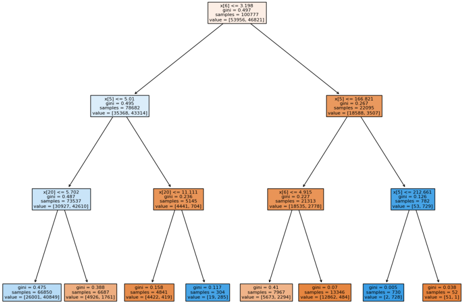


Figure: Decision Tree

Random Forest Model

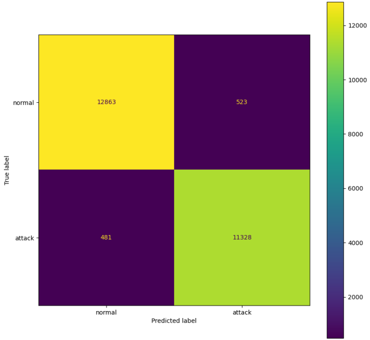


Figure: Confusion matrix of Random Forest

Left: Among all the above model it is more robust to noise

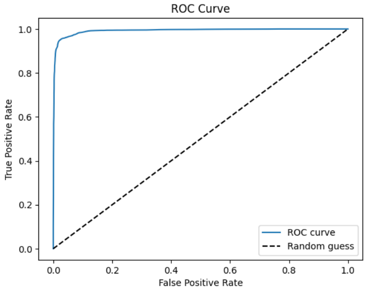


Figure: ROC of Random Forest

Using Neural Network

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_10 (Dense)            (None, 64)                7872

 dropout_8 (Dropout)         (None, 64)                0

 dense_11 (Dense)            (None, 128)               8320

 dropout_9 (Dropout)         (None, 128)               0

 dense_12 (Dense)            (None, 512)               66048

 dropout_10 (Dropout)        (None, 512)               0

 dense_13 (Dense)            (None, 128)               65664

 dropout_11 (Dropout)        (None, 128)               0

 dense_14 (Dense)            (None, 1)                 129

=================================================================
Total params: 148,033
Trainable params: 148,033
Non-trainable params: 0
```

Figure: Artificial Neural Network Model

Left: Activation function used this Relu and Sigmoid in last layer

Neural Network Accuracy

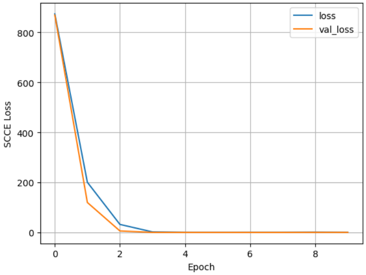


Figure: Loss v/s Epoch

Left: Accuracy decreases to 83% for 10 Epochs

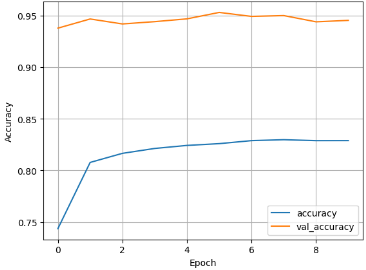


Figure: Accuracy v/s Epoch

Comparision among models

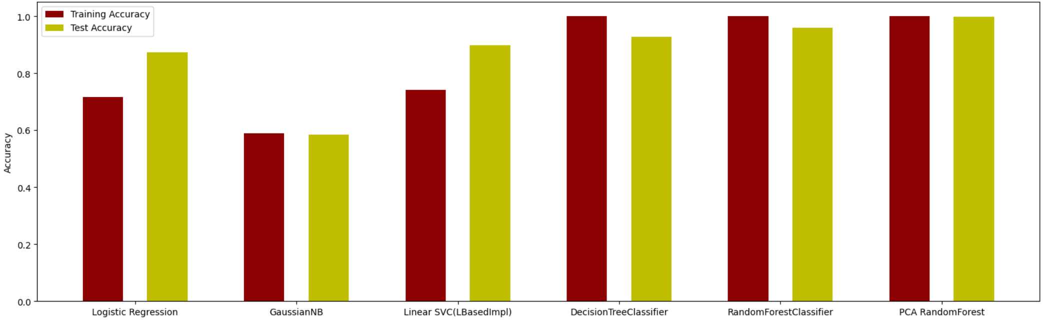
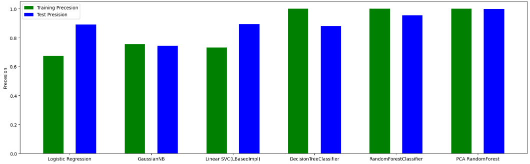
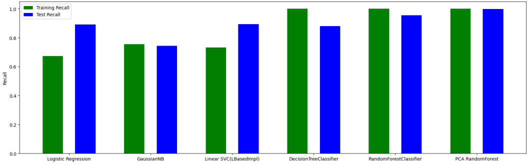


Figure: Training and Test Accuracy

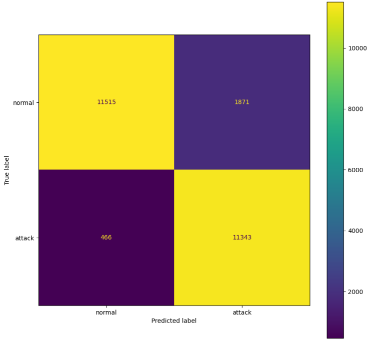
Precision and Recall of models



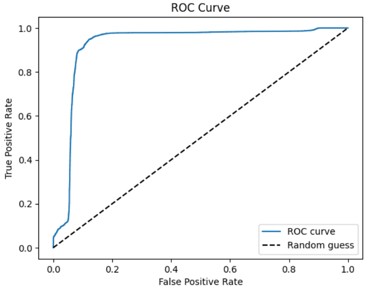


Data with noise level-2

Logistic Regression Model (Base Line Model)



Left: Accuracy decreases due to noisy data



Naive Bayes Model

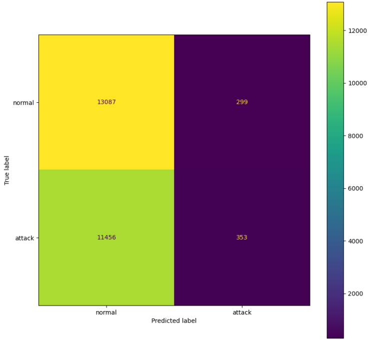


Figure: Confusion matrix of Naive Bayes

Left: Worst performance by Naive Bayes due to noise

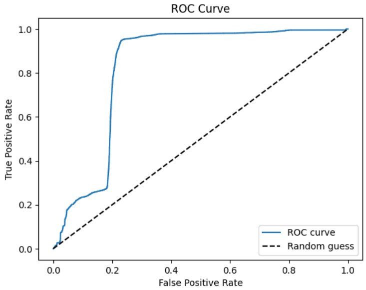


Figure: ROC of Naive Bayes

SVM Model

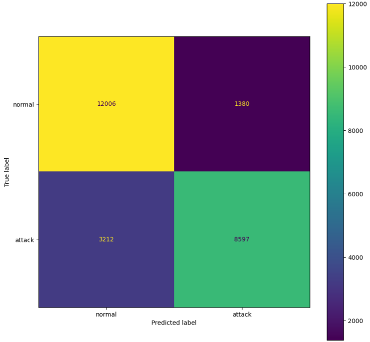


Figure: Confusion matrix of SVM

Left: SVM does not perform well when the data has more noise. It performs worst than logistic regression model.

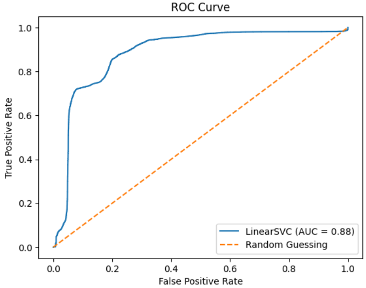


Figure: ROC of SVM

Decision Tree Model

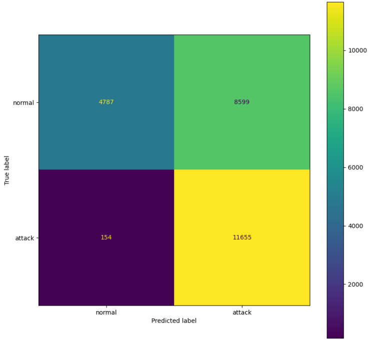


Figure: Confusion matrix of Decision Tree

Left: Predicts attack as normal in most cases due to noise. It performs worst than SVM.

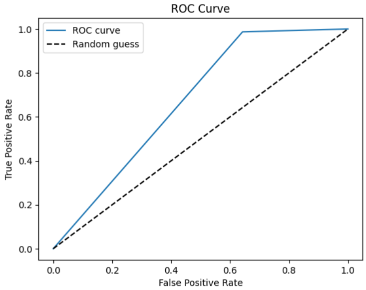
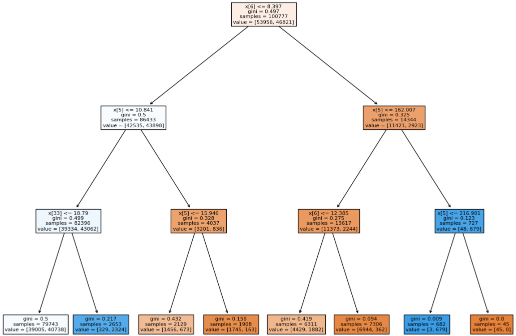


Figure: ROC of Decision Tree

Decision Tree Model

figure: Decision Tree



Random Forest Model

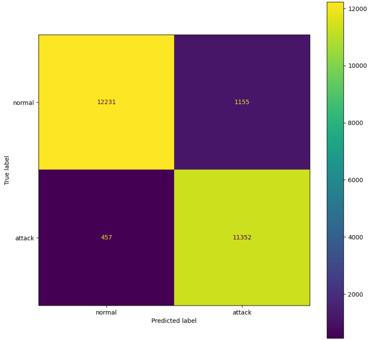


Figure: Confusion matrix of Random Forest

Left: Among all the above model it is more robust to noise

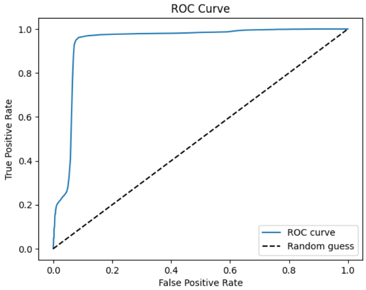


Figure: ROC of Random Forest

Using Neural Network

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_10 (Dense)            (None, 64)                7872

 dropout_8 (Dropout)         (None, 64)                0

 dense_11 (Dense)            (None, 128)               8320

 dropout_9 (Dropout)         (None, 128)               0

 dense_12 (Dense)            (None, 512)               66048

 dropout_10 (Dropout)        (None, 512)               0

 dense_13 (Dense)            (None, 128)               65664

 dropout_11 (Dropout)        (None, 128)               0

 dense_14 (Dense)            (None, 1)                 129

=================================================================
Total params: 148,033
Trainable params: 148,033
Non-trainable params: 0
```

Figure: Artificial Neural Network Model

Left: Activation function used is Relu and Sigmoid in the last layer

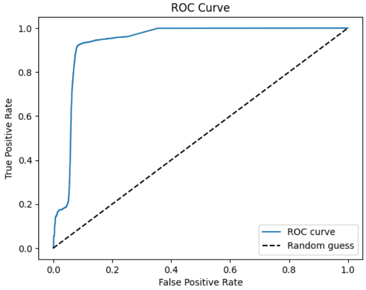


Figure: ROC of ANN

Neural Network Accuracy

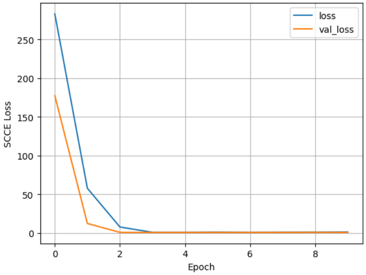


Figure: Loss v/s Epoch

Left: Accuracy decreases to about 67%.

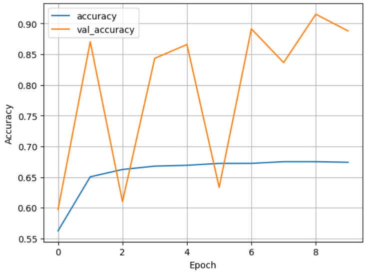


Figure: Accuracy v/s Epoch

Comparison among models

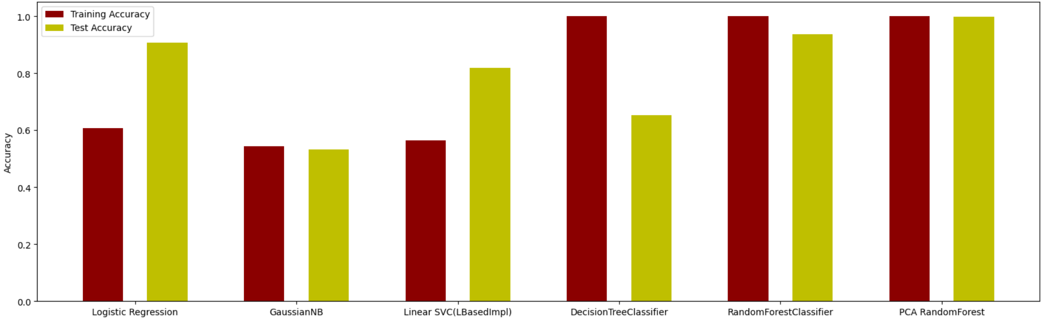
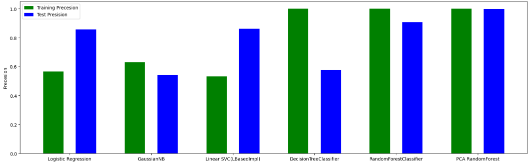
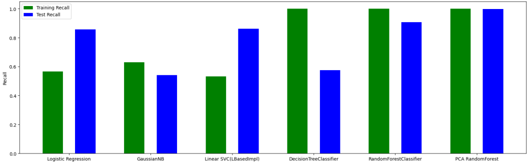


Figure: Training and Test Accuracy

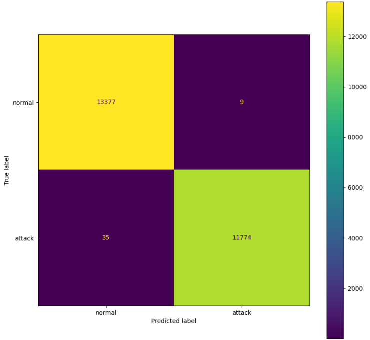
Precision and Recall of models

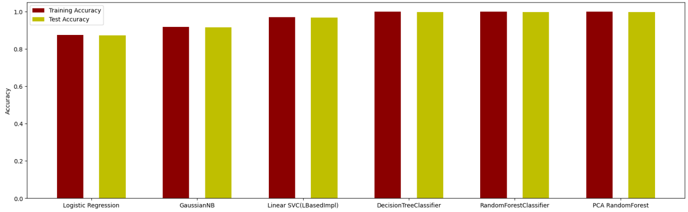




Proposed Model

PCA on Random forest for noiseless Data





PCA on Random forest for noise level 1

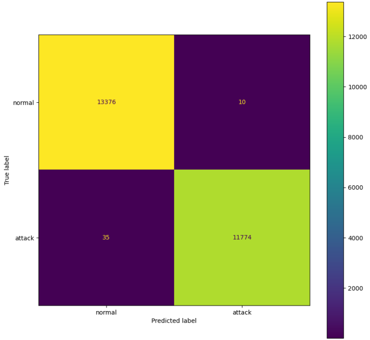


Figure: Confusion matrix after applying PCA

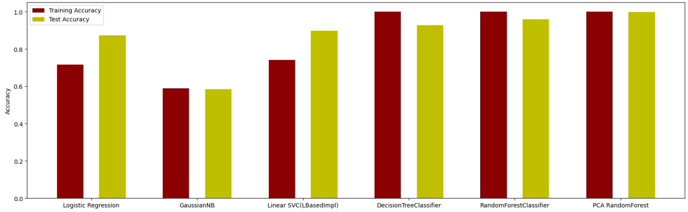


Figure: Training and Test Accuracy

PCA on Random forest for noise level 2

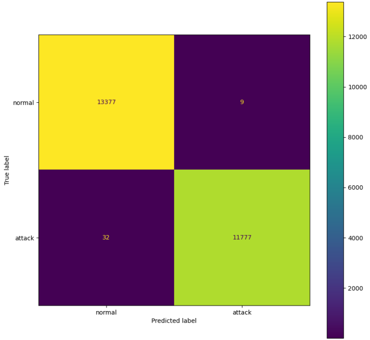


Figure: Confusion matrix after applying PCA

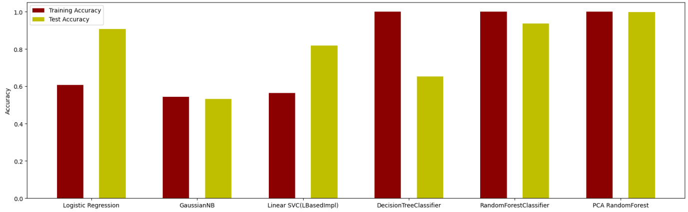


Figure: Training and Test Accuracy

Conclusion

In conclusion, we learned about the performance of various models and how robust they are to artifacts/noisy data. We used Logistic Regression as the baseline model for the project. We observe that SVM is very sensitive to noise while it performs better for noiseless data which is similar to the result of [4]. We found the idea of using Random forest [3] and we experimentally concluded that using Random forest with PCA give the better result. One of the reason is that the noise was distributed only on 20 features which was not the case before applying PCA. However, using ANN for more number of Epochs will definitely outperform other models but in reality any new intrusion must be trained immediately and be deployed in the environment. Hence, concluding that Random forest with PCA is and effective way to detect anomaly in noisy data.

References

[1] Razan Abdulhammed et al. "Effective features selection and machine learning classifiers for improved wireless intrusion detection". In: *2018 International symposium on networks, computers and communications (ISNCC)*. IEEE. 2018, pp. 1–6.

[2] Mustapha Belouch, Salah El Hadaj, and Mohamed Idhammad. "Performance evaluation of intrusion detection based on machine learning using Apache Spark". In: *Procedia Computer Science* 127 (2018), pp. 1–6.

[3] Karuna S Bhosale, Maria Nenova, and Georgi Iliev. "Data Mining Based Advanced Algorithm for Intrusion Detections in Communication Networks". In: *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*. IEEE. 2018, pp. 297–300.

[4] Zina Chkirbene et al. "Hybrid machine learning for network anomaly intrusion detection". In: *2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT)*. IEEE. 2020, pp. 163–170.

[5] Jie Gu and Shan Lu. "An effective intrusion detection approach using SVM with naıve Bayes feature embedding". In: *Computers & Security* 103 (2021), p. 102158.

[6] Kishor Kumar Gulla et al. "Machine learning based intrusion detection techniques". In: *Handbook of Computer Networks and Cyber Security: Principles and Paradigms* (2020), pp. 873–888.

[7] Alif Nur Iman and Tohari Ahmad. “Improving intrusion detection system by estimating parameters of random forest in Boruta”. In: *2020 International Conference on Smart Technology and Applications (ICoSTA)*. IEEE. 2020, pp. 1–6.

[8] Farrukh Aslam Khan et al. “A novel two-stage deep learning model for efficient network intrusion detection”. In: *IEEE Access* 7 (2019), pp. 30373–30385.

[9] Kazi Abu Taher, Billal Mohammed Yasin Jisan, and Md Mahbubur Rahman. “Network intrusion detection using supervised machine learning technique with feature selection”. In: *2019 International conference on robotics, electrical and signal processing techniques (ICREST)*. IEEE. 2019,

Thank you!