A Proposed Hybrid Model for Intrusion Detection System using AI and PCA

*Himadri Shekhar Giri

*Master of Technology in information security, MAKAUT

Abstract:- "A Proposed Hybrid Model For Intrusion Detection System Using AI and PCA" research is used in implementing the hybrid model in Intrusion detection system. As the dependency on cyber wold is increasing rapidly, the chance of data compromise is also increasing rapidly. It has two parts signature-based or pattern based detection and anomaly-based or ai based detection. In signature-based detection, the system works with security patches and known attacks. In the case of unknown data, signature-based detection is not a perfect method of detection. Anomaly-based intrusion detection system is nothing but the implementation of artificial intelligence, those are used to compare the accuracy of prediction. The main aim of the research is to find out the best accuracy along with the implementation of the Hybrid detection model. Data Analysis and Principal Component Analysis is the instinctive part of this research to comprehend the data sets properly. Along with the implementation of an Anomaly-based intrusion detection system, a hybrid model also has been proposed for the best way of intrusion detection. The hybrid model is a combined implementation of signature and anomaly-based detection.

Keywords:- *Machine Learning, Deep Learning, Intrusion detection system, cyber security, Principle Component Analysis.*

I. INTRODUCTION

Nowadays, every working field is going to introduce a digital system to their organization. As much as the dependency on the digital system will increase, the chance of data being compromised through that digital system will also increase rapidly.To protect the data compromised cyber security plays an important role. There is no such countermeasure is present which can defend against different cyber-attacks. The increment in cyber threats is a big concern nowadays.

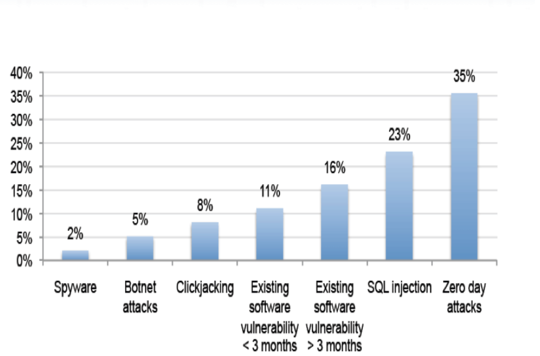


Fig. 1: Stats about zero-day attacks [25]

The development of cyber security for protection is always a difficult task to do. Most of the solutions that are present are signature-based detection. Signature-based detection method needs human intervention to supervise and update the signature patch. This signature-based detection failed to identify the new types of attack or newly generated malware. Signature-based detection system relays on the signature database. To solve these kinds of problems Artificial Intelligence (AI) is introduced along with machine learning algorithms [1].

A. Concept of Intrusion Detection System (IDS)

A software application that can monitor network traffic for suspicious activities and issues alerts when such activities are identified. It can scan the network as well as the system for unauthorized activities or policy breaking. Unauthorized activities or privacy violations are reported to the administrator and it is also collected inside the system information and event management system (SIEM).

We need to tune the intrusion detection system (IDS) a in such way that it can properly recognize normal traffic behaviors as compared to malicious activities.

B. Signature Based IDS system

In this type of IDS system, it is always best to use this technique for the known threats. It consists of both preprogram lists of known threats and their indicators of compromise (IOC). IOC precedes perfectly the malicious network attack, file hashes, malicious domain, known byte sequences, etc. Signature-based has its database with some known threats. It scans the packets traversing through the network. It also compares those packets to the database for any suspicious activities.

C. Anomaly Based IDS system

An anomaly-based IDS system can alert the unknown malfunction. It uses machine learning to train the detection system and predict according to that. After finding the malicious activities it will generate IOC to trigger an alert.

D. Importance of IDS system

There is a firewall to protect all systems but no firewall is foolproof or impenetrable. Attackers can implement different ways or procedures to trap the defense of any system. Many attacks help all the malware to get the user credentials. These user credentials help them to get access to networks or data. An intrusion detection system always helps the security system to identify any malicious traffic present in the network.

The main responsibility of an IDS system is to notify the administrator when any unauthorized or malicious activity is taking place. IDS system not only scans the whole network but also monitors all the data traversing between systems within the network. IDS generate alerts when it finds some malicious activity or known threats are detected. This system has a huge role to block attacks that can be performed by the intrusion or network.

II. LITERATURE REVIEW

Liu et al. [3] done a survey and proposes a taxonomy of IDS. They take the proposed taxonomy system as a baseline and explain how to solve key IDS issues with artificial intelligence.

Wake et al. [4] has proposed an IDS system with principal component analysis (PCA). Results obtained to state that the proposed approach works more efficiently in terms of accuracy as compared to other AI techniques.

Aung et al. [5] used K-means and the Random Forest algorithm to classify instances. This model was verified using the KDD'99 dataset. Experimental results show that hybrid methods can support suitable detection rates and lower model training time than using a single algorithm.

Anton et al. [6] have analysed the industrial data with machine learning and time series-based anomaly detection algorithms to discover the attacks introduced to the data. Two different data sets are used, one Modbus-based gas pipeline control traffic and one OPC UA-based batch processing traffic. To detect attacks, two machine learning-based algorithms are used, namely SVM and Random Forest.

Farahnakian et al. [7] used a deep learning approach for intrusion detection systems. They have used Deep Auto-Encoder as one of the most well-known deep learning models. The proposed DAE model is trained in a greedy layer-wise fashion to avoid overfitting and local optima. The experimental result shows that our approach provides a substantial improvement over other deep learning-based approaches in terms of accuracy, detection rate, and false alarm rate.

Saranya et al. [8] have explored the comparative study of various ML algorithms used in IDS for several applications such as fog computing, big data, and smart city. They also have classified the intrusions using ML algorithms like Linear Discriminate analysis (LDA), classification and regression trees (CART), and Random Forest.

Sheikh et al. [9] have demonstrated the use of an optimized pattern-recognized algorithm to detect attacks. They have proposed an intrusion detection system methodology and designed architecture for the internet of things that makes use of this search algorithm to thwart various security breaches. Numerical results are also presented from the test conducted.

Khraisat et al. [1havehas proposed a novel assemble Hybrid intrusion detection system, combining a c5 decision tree and one-class support vector machine classifier. HIDS combines the advantages of both Signature-based IDS and anomaly-based IDS. This framework aims to detect both well-known and zero-day attacks with high detection accuracy and low false alarm rate.

Zaman et al. [11] have applied seven different machine learning techniques with information entropy calculation to Kyoto 2006+ dataset and evaluated the performance of these techniques.

Vinayakumar et al. [12] have applied a Deep learning model over the different datasets. They tried to find out a better data distribution among all types of different dimensions dataset.

Mighan et al. [13] have successfully tackled the problems of processing a vast amount of security-related data for the task of network intrusion detection. They employ Apache spark as a big data processing tool for processing a large size of traffic data. They combine the advantages of both Deep learning and machine learning methods.

Mushtaq et al. [29] have proposed a hybrid framework, where they have combined deep auto encoder with long short-term memory and Bi-directional long short-term memory. This hybrid system was implemented over the NSL-KDD dataset. They have calculated the f1 score, recall, accuracy, and flash alarm rate.

Yu et al. [30] have proposed an intrusion detection system for network security communication based on a multi-scale convolutional neural network. They have conducted experiments on a public dataset. In this paper, they have done a comparative study among AdaBoost, recurrent neural networks, and multi-scaled convolutional neural networks. They have concluded with a multi-scale convolutional neural network not only increases the accuracy of the model but also reduces the time complexity at the time of detecting error.

Soltani et al. [31] have proposed their detection system against content-based detection attacks like SQL injection, Cross-site scripting, and various virus. In this paper, they have proposed a framework, called a deep intrusion detection system (DID) that uses the pure content of traffic flow in addition to traffic metadata in the learning and detection phase of passive DNS IDS. They have implemented LSTM as a deep learning technique.

Bhati et al. [32] have implemented ensemble-based IDS using voting have been seen to outperform individual approaches. Since the Voting methodology can work around both, theoretically similar and different classifiers and produce a single classifier based on the majority characteristics, it proved to be better than the other ensemble-based techniques. In this paper, they have used two algorithms SVM and extra tree. This method has been implemented over the KDD Cup dataset.

Sajith et al. [33] have implemented an Adaptive neuro-fuzzy inference system (ANFIS) as a classifier for classifying network malicious. ANFIS show better performance because it joins the upsides of both ANN and fuzzy deduction frameworks including the capacity to catch the nonlinear design of interaction, variation ability, and fast learning limit.

Al-Enazi et al [34] has aimed to enhance the accuracy and the speed of intrusion detection system by using the feature selection method, reducing the dimension of the data and eliminating irrelevant features. They have used a classifier of the stacking method. They have introduced logistic regression as Meta a classifier and combined random forest, sequential minimal optimization, and naïve biased method.

Ahmed et al. [35] have first clarified the concept of an IDS and then provide the taxonomy based on the notable machine learning and deep learning techniques adopted in designing network-based IDS systems. Then, recent trends and advancements in ML and DL-based NIDS are provided in terms of the proposed methodology, evaluation metrics, and dataset selection.

III. METHODOLOGY

A. *Proposed Model*

The main problem with this project is implementing both the signature-based and anomaly-based detection system in a single frame. Because both the IDS system has different disadvantages. In the case of signature-based IDS, the main disadvantage is to detecting zero-day attacks properly. The disadvantage of anomaly-based IDS is the accuracy of the model. If the accuracy of the model is high then also for the same data time complexity is a big issue. To make a perfect IDS system we have to merge both the detection technique to overcome other disadvantages. In the proposed model image, we can see the implementation of both signature-based and anomaly-based ids. The first part of the image consists of the signature-based detection system. We have to provide a dataset (Train data) for this hybrid system.

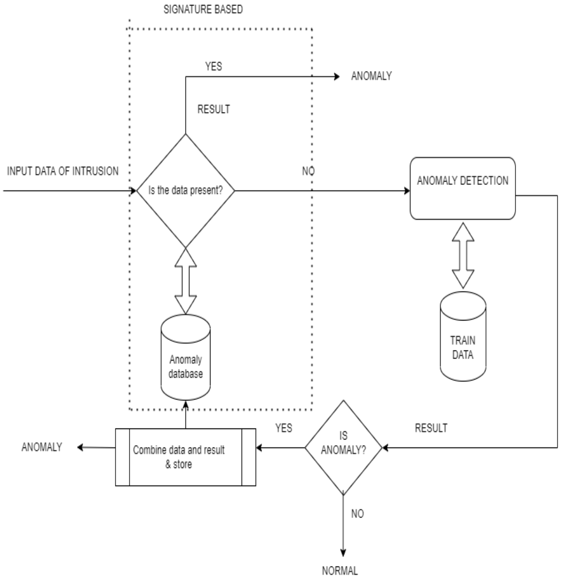


Fig. 2: Proposed Hybrid Model

B. *Signature-Based Detection Process*

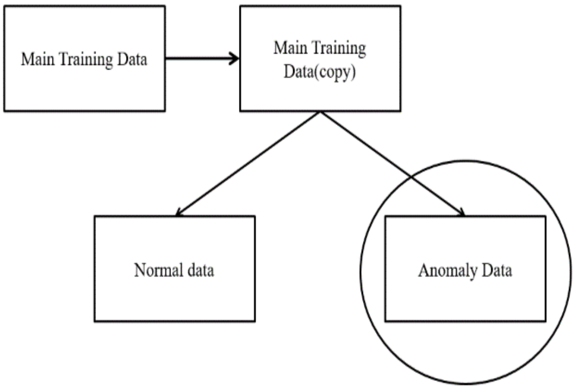


Fig. 3: Build a database

In the first step, we have to make a copy of the training database. Then we have to separate the copied train database into two parts, normal data and anomaly data. As our main aim is to detect anomaly data, we will select the anomaly database as a master database for the signature-based detection system.

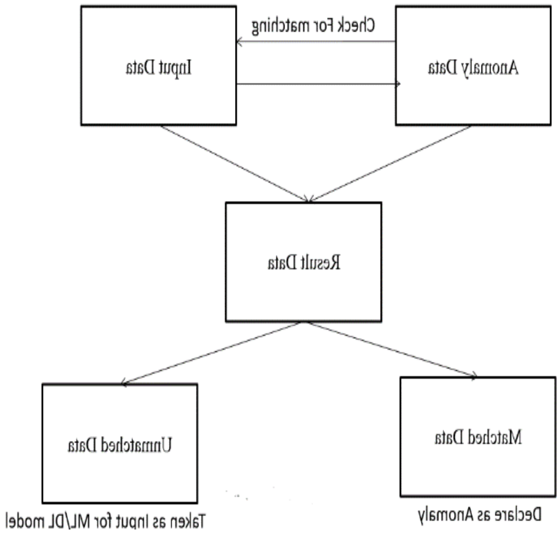


Fig. 4: Check for matching

Now we give input to the system. There is a condition checking, whether the data is present or not in the master (anomaly) database. The result will store in the result database. If the data is present in the database then it will say matched. If the data is not present in the database then unmatched. After getting the result data, we will separate the data into two parts concerning matched or unmatched data. The matched database will be declared as anomaly data and the unmatched database will be taken as input data for the machine learning or deep learning model in anomaly-based detection.

C. Anomaly-Based Detection Process

Initially, we have taken training data. That Training data is there to train the model. Then the model will predict whether the data is an anomaly or normal. According to the accuracy of the model, it will predict the result.

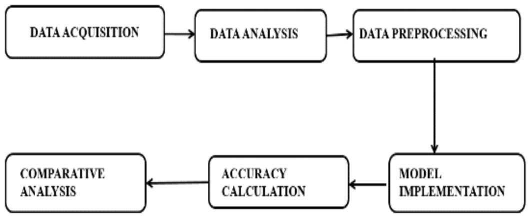


Fig. 5: Anomaly-based IDS

a) Data Acquisition

Dataset	size	features	No. of class
Data-1	25,193	42	2
Data-2	1,25,974	42	5

Table 1: Datasets description

Here we have used two datasets. Data-1 has 25,193 data and data-2 has 1, 25,974 data. Data-1 has two classifiers normal and anomaly. Data-2 has 5 different classifiers. Both the data are 42 features and both data are there in labeled and unlabeled format. As the data is present in both formats, we have an opportunity to perform supervised and unsupervised models on the same database. It also helps to compare the model and find the best model among them.

b) Data Preprocessing

The most important part before implementing any machine learning approach is data preprocessing. Some data preprocessing techniques we have used to make the data perfect for the implementation. Initially, we load the dataset in the CSV mode of the file. After that, we performed null value checking in the whole database. A function is used to check the whole data and whether the data consists of any null value or not. After checking, if the null value is there then drop that null value to make the data perfect. In the second step, now we are checking for any value that is duplicated or not in the database. After successfully dropping the duplicate value from the database we lastly performed a label encoder to convert all the string data to an integer value. This label encoder helps to convert string which is present in the database with integers. The Same string denotes the same integers.

c) Data Standardization

As a data preprocessing technique, we have used a standard scalar. In the dataset there are different kinds of variables are present. Those variables are measured at different scales. This does not contribute equally to the model at the time of fitting the data. As a result, this can create a bias in the dataset. To resolve it, we standardize the data before fitting it into the model.

The concept is standardized i.e. μ=0,σ=1 for any variable or feature x. Each column which consists of different features will be standardized using a standard scalar function in individual features.

$$Standerization: \quad Z = \frac{x-\mu}{\sigma}$$

$$With\,mean: \ \mu = \frac{1}{N}\sum_{i=1}^{N}(X_i)$$

And,

Standard Deviation: $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2}$

After implementing this, it removes the mean values and scales each feature or variable with unit variance. This process is executed individually feature-wise.

As we know the drawback of the standard scalar is that it can be influenced by outliers if they exist. So, before implementing it we have to remove outliers.

The most important part of classification is to make the data properly so that can predict results accurately. The data has to be unbiased in that case otherwise it can influence the result that is going to be predicted. We are plotting a graph concerning the target variable or class. This will define the number of targets of different classes. It will also help to check whether the data is biased or not. We areallowing a 60-40 ratio on an average in the binary class. Beyond this ratio, the data will be calculated as biased data in the dataset.

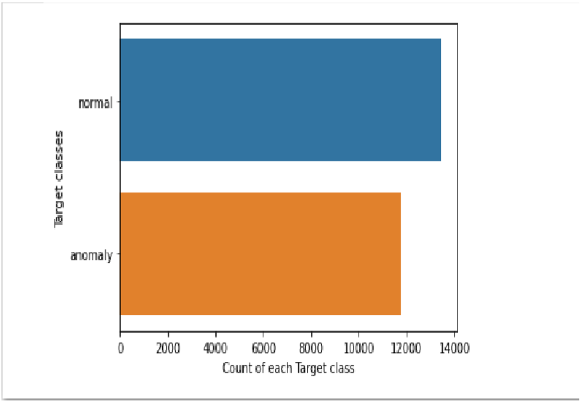


Fig. 6: Class count to check data biasness

d) Data analysis
Data analysis is the most important part to perform. After checking whether the data is biased or not. We have to eliminate the feature which is highly co-related to each other. To perform this operation, we have introduced a co-relation matrix. First, we print all the variables with an upper co-related value than 0.5. Then, to fine-tune the higher value we have taken the co-related threshold as 0.8. Now we check the data shape of the dataset. Set the threshold as 0.9 and drop those features, which have a higher co-related value than that. Now again check the new shape of the data after dropping the highly co-related features.

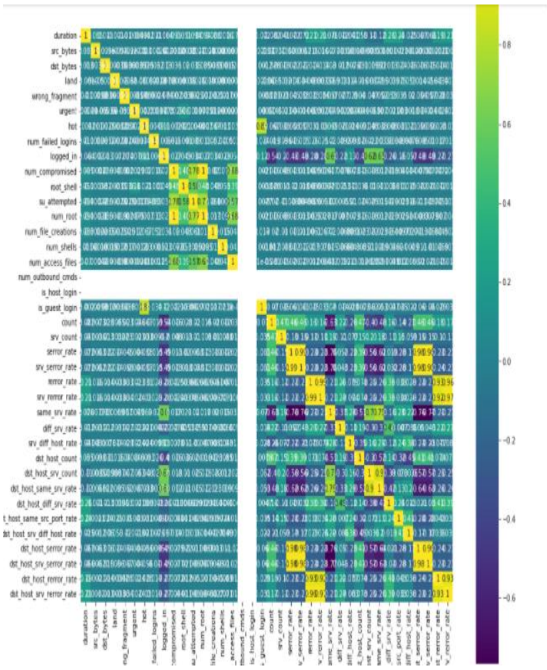
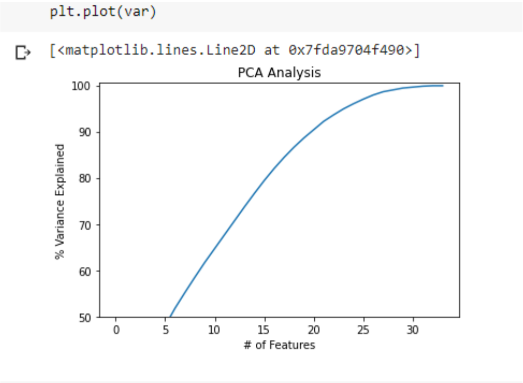


Fig. 7: Correlation matrix

e) PCA Component
Principle component analysis is one of the most important feature selection processes before implementing the machine learning model. Sometimes, working with lots of variables is difficult. It can create several problems. So many variables can create some danger like overfitting of data to the model, it can also violate the assumptions of the model. Sometimes the important variables might be non-priorities due to the involvement of so many variables. So we are not able to find the impact of the important variables on the result.

To resolve this problem, we have introduced this procedure named PCA components.

In this process, we are using the reduction of the dimension of the feature space. By using this we can reduce the chance of overfitting for the short term.



```
plt.plot(var)
```

```
[<matplotlib.lines.Line2D at 0x7fda9704f490>]
```

Fig. 8: PCA graph

From the above graph, we can understand that initially, the accuracy of the model increases as we increase the number of features. But after a certain point of time, the accuracy remains constant even after increasing the number of features. We have to take that point where the accuracy is constant. That point dedicates us that, this is the minimum number of features that are needed to get the highest accuracy. If we want to reduce the dimension of the feature then we can only take the minimum features by this process.

f) Feature Elimination
Here we will reduce the dimension by eliminating features from the dataset the features, that are important and have a higher information gain will remain the same. The features with low importance will be eliminated to reduce the dimension of the features. This feature selection method will be performed manually with the help of a graph. From the graph, we will manually select how many minimum features at least we need to get the best accuracy.

Eliminating features might include the simplicity and interpretability of the variables.

g) Feature Extraction
In this process, we are combining different kinds of features into a single new feature category. It will also reduce the dimension but not eliminate the features that are very much important for the dataset.

h) Why PCA used: -

- Not able to deal with all the variables present in the database but not that much important.
- It ensures that the variables are independent of each other.
- We want to make the variables more independent and less interpretable.

D. Need for Classification

The task that has to be performed by machine learning is to recognize objects as well as be able to separate them into categories. This is known as classification.

Classification implies utilizing input training data to predict the same type or same predetermined categories. As we are predicting whether a packet data is anomaly or normal, there will be two categories being separated by somerelated features into two groups. This separation is performed concerning similarities in features. Then it will behelpful for themachine to predict which data comes under which category or class. As the level or target data is separated into groups or classes then always it will be very useful to use classification in machine learning.

IV. RESULT AND ANALYSIS

A. *Result*

a) RANDOM FOREST

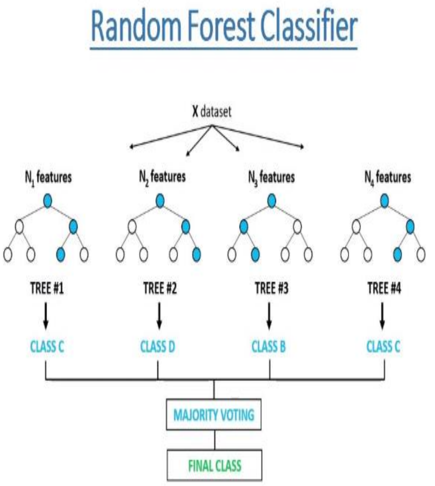


Fig. 9: Random forest[20]

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePostive + FalseNegetive}$$

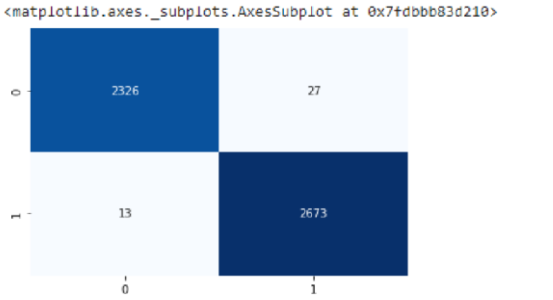
$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Macro F1 calculates the F1 separated by class but not using weights for the aggregation:

$$F1_{Class1} + F1_{Class2} + \cdots + F1_{ClassN}$$

Weighted F1 score calculates the F1 score for each class independently but when it adds them together uses a weight that depends on the number of true labels of each class:

$$F1_{Class1} * W_1 + F1_{Class2} * W_2 + \cdots + F1_{ClassN} * W_N$$



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdbbb83d210>
```

Fig. 10: - Confusion matrix of random Forest

Random Forest

Dataset	**Accuracy**	**Confusion Matrix**	**Precision**	**F1-score**	**MSE**
Data-1	**99.20%**	**99%**	**99.21%**	**99.20%**	**99.20%**
Data-2	**99.61%**	**100%**	**88.71%**	**85.27%**	**98.77%**

Table 2 Result of random forest

b) SVM (Support Vector Machine)

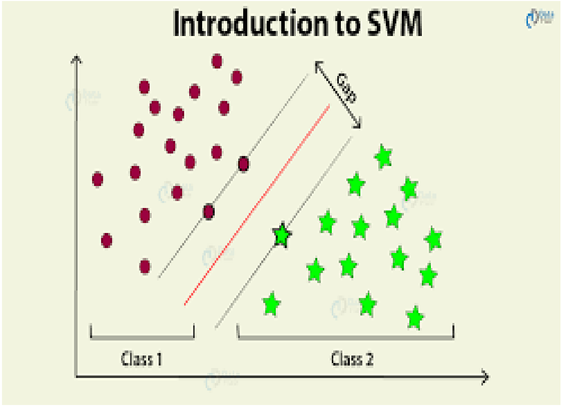


Fig. 11: SVM [21]

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePostive + FalseNegetive}$$

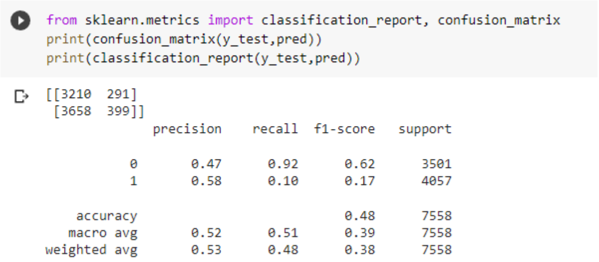
$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Macro F1 calculates the F1 separated by class but not using weights for the aggregation:

$$F1_{Class1} + F1_{Class2} + \cdots + F1_{ClassN}$$

Weighted F1 score *calculates the F1 score for each class independently but when it adds them together uses a weight that depends on the number of true labels of each class:*

$$F1_{Class1} * W_1 + F1_{Class2} * W_2 + \cdots + F1_{ClassN} * W_N$$



```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
[[3210  291]
 [3658  399]]
              precision    recall  f1-score   support

           0       0.47      0.92      0.62      3501
           1       0.58      0.10      0.17      4057

    accuracy                           0.48      7558
   macro avg       0.52      0.51      0.39      7558
weighted avg       0.53      0.48      0.38      7558
```

Fig. 12: SVM confusion matrix

SVM

Dataset	**Accuracy**	**Confusion Matrix**	**Precision**	**F1-score**	**MSE**
Data-1	**51.49%**	**51%**	**58.99**	**48.12 %**	**51.49 %**
Data-2	**59.80%**	**60%**	**31.42%**	**26.26 %**	**20.59 %**

Table 3: Result of SVM

c) AUTO ENCODER

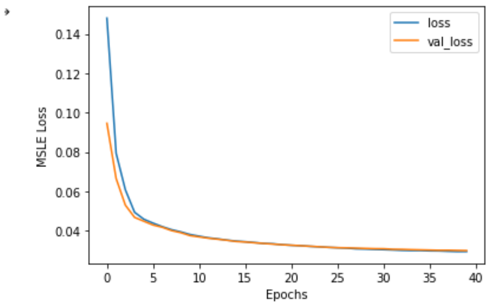


Fig. 13: Epochs vs MSLE loss

The above image represents a graphical image of Epochs vs MSLE loss. Blue line represents training loss and orange line represents validation loss. When training and validation loss will constant then we have to take that point as number of epochs.

d) Auto Encoder

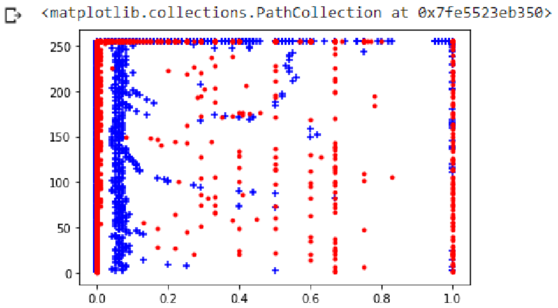
Dataset	**Accuracy**	**Confusion matrix**	**Precision**	**F1-score**	**MSE**
Data-1	**90.75%**	**91%**	**90.77%**	**90.73 %**	**90.75 %**
Data-2	**84.77%**	**85%**	**33.74%**	**35.45 %**	**57.01 %**

Table 4: Result of Autoencoder

B. Analysis

After getting all the accuracy from the different models with respect to the same data we have compared them. As already we can see random forest is providing better accuracy with respect to SVM and autoencoder. That's why we can say that random forest is the best model for this kind of dataset.

a) Why SVM can't perform better?



```
<matplotlib.collections.PathCollection at 0x7fe5523eb350>
```

Fig. 14: - Data distribution points

We have to find out the reason why SVM is not performing better with this dataset. In the above image as we can see this is the distribution of data with respect to two classes. We can see how much data is scattered in this image. In SVM hyper plane vector line tries to separate two classes with respect to their distribution. If this is the condition of the data distribution, then it is always difficult for the SVM hyperplane to separate two classes differently and also accurately.

b) Why Auto Encoder is not performing well?
First of all auto encoder is an unsupervised deep learning model. So it will not deal with labeled data. We have to deal with this model differently.

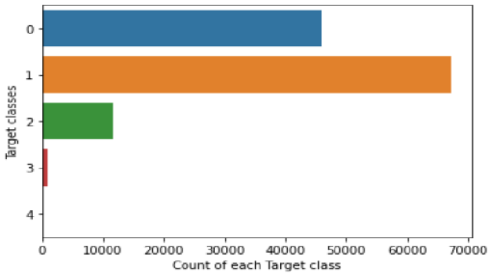


Fig. 15: Class distribution for multi classifiers

In the above, image we can see different classes are distributed in different ratios. Especially, if we can see the class ratio of class 3 and class 4 is very less in comparison to other classes. As the ratio of two classes is less so there is a high chance of data underfitting. As a result, when the model will try to predict the result, there is a high possibility to predict those data wrongly. That's why the accuracy of the model is less with respect to the random forest.

c) Why Random forest Best?
Whatever problem comes with different models will be resolved when we will deal with random forests.

* In random forest, it does not depend on the data distribution. It totally deals with features.

* In random forest main advantage is that there is no data over fitting or under fitting problem. That's why the random forest is working better.

V. CONCLUSION

In this thesis, the performance of machine learning algorithms and deep learning algorithms has been discussed in the case of cyber-attack detection or intrusion detection. The importance of cyber security nowadays is also discussed. Different types of cyber-attack are also explained. A hybrid model of IDS is also proposed and implemented. As we can see that the accuracy of the Random Forest classifier is highest than other algorithms like SVM and Auto encoder. In Random Forest, we are getting an accuracy of almost 99% in both datasets. The selected model for this anomaly-based detection is Random Forest. In a hybrid detection system, a signature-based detection part has been done to increase the accuracy of known data and reduce the time complexity. This overall project is not only increasing the accuracy of the model but also reducing the time complexity. We have implemented PCA to reduce the dimension of the data. We have also reduced the irrelevant features. At last, we are able to successfully implement the whole hybrid model that we have already proposed.

REFERENCES

[1.] Jordan MI, Mitchel TM (2015) Machine Learning: trends, perspectives and prospects. Science 349(6245):255-260

[2.] LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436

[3.] Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. Applied sciences, 9(20), 4396.

[4.] Waskle, S., Parashar, L., & Singh, U. (2020, July). Intrusion detection system using PCA with random forest approach. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 803-808). IEEE.

[5.] Aung, Y. Y., & Min, M. M. (2017, June). An analysis of random forest algorithm-based network intrusion detection system. In 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) (pp. 127-132). IEEE.

[6.] Anton, S. D. D., Sinha, S., &Schotten, H. D. (2019, September). Anomaly-based intrusion detection in industrial data with SVM and random forests. In 2019 International conference on software, telecommunications and computer networks (SoftCOM) (pp. 1-6). IEEE.

[7.] Farahnakian, F., &Heikkonen, J. (2018, February). A deep auto-encoder based approach for intrusion detection system. In 2018 20th International Conference on Advanced Communication Technology (ICACT) (pp. 178-183). IEEE.

[8.] Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. A. (2020). Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Computer Science, 171, 1251-1260.

[9.] Sheikh, T. U., Rahman, H., Al-Qahtani, H. S., Hazra, T. K., & Sheikh, N. U. (2019, October). Countermeasure of Attack Vectors using Signature-Based IDS in IoT Environments. In 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 1130-1136). IEEE.

[10.] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., &Alazab, A. (2019). A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. Electronics, 8(11), 1210.

[11.] Zaman, M., & Lung, C. H. (2018, April). Evaluation of machine learning techniques for network intrusion detection. In NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium (pp. 1-5). IEEE.

[12.] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525-41550.

[13.] Mighan, S. N., &Kahani, M. (2021). A novel scalable intrusion detection system based on deep learning. International Journal of Information Security, 20(3), 387-403.

[14.] Lippmann RP, Fried DJ, Graf I, Haines JW, Kendall KR, McClung D, Weber D, Webster SE, Wyschogrod D, Cunningham RK, Zissman MA (2000) Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation. In: Proceedings DARPA information survivability conference and exposition, DISCEX'00, vol 2. IEEE, pp 12–26

[15.] Kayacik H, Zincir-Heywood AN, Heywood MI (2005) Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets. In: Proceedings of the third annual conference on privacy, security and trust 2005, PST 2005, DBLP

[16.] Zhang J, Zulkernine M, Haque A (2008) Random-forests-based network intrusion detection systems. IEEE Trans Syst Man Cybern Part C Appl Rev 38(5):649–659

[17.] Li W (2004) Using genetic algorithm for network intrusion detection. In: Proceedings of the United States department of energy cyber security group, vol 1, pp 1–8

[18.] Kolias C, Kambourakis G, Maragoudakis M (2011) Swarm intelligence in intrusion detection: a survey. ComputSecur 30(8):625–642. https://doi.org/10.1016/j.cose.2011.08.009

[19.] Al-Subaie M, Zulkernine M (2006) Efficacy of hidden Markov models over neural networks in anomaly intrusion detection. In: 30th Annual international computer software and applications conference. COMPSAC 06.,vol 1, pp 325–332. ISSN 0730-3157

[20.] Upadhyay R, Pantiukhin D Application of convolutional neural network to intrusion type recognition. https://www.researchgate.net

[21.] Gao Ni et al (2014) An intrusion detection model based on deep belief networks. In: 2014 Second international conference on advanced cloud and big data (CBD). IEEE

[22.] Moradi M, Zulkernine M (2004) A neural network based system for intrusion detection and classification of attacks. In: Paper presented at the proceeding of the 2004 IEEE international conference on advances in intelligent systems Theory and applications. Luxembourg

[23.] Mukkamala S, Sung AH, Abraham A (2003) Intrusion detection using ensemble of soft computing paradigms. In: Third international conference onintelligent systems design and applications, intelligent systems design and applications, advances in soft computing. Springer, Germany, pp 239–48

[24.] Javaid A, Niyaz Q, Sun W, Alam M (2015) A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI international conference on bio-inspired information and communications technologies (formerly BIONETICS), New York, NY, USA, 3–5 Dec 2015, pp 21–26. They also used recurrent network to preserve the state full information of malware sequences

[25.] Jihyun K, Howon K (2015) Applying recurrent neural network to intrusion detection with hessian free optimization. In: Proc, WISA

[26.] Kim J, Kim J, Thu,HLT, Kim H (2016) Long short term memory recurrent neural network classifier for intrusion detection. In: 2016 International conference on platform technology and service (PlatCon), Jeju, pp 1-5. https://doi.org/10.1109/PlatCon.2016.7456805

[27.] Staudemeyer RC (2015) Applying long short-term memory recurrent neural networks to intrusion detection. S AfrComput J 56(1):136–154

[28.] Vinayan kumar Ravi, Soman Kp, Prabaharan Poornachandran, Akarsh Soman (2019) Application of Deep Learning Architecture for Cyber Security. ResearchGate 10.1007/978-3-030-16837-7_7

[29.] Mushtaq, E., Zameer, A., Umer, M., & Abbasi, A. A. (2022). A two-stage intrusion detection system with auto-encoder and LSTMs. *Applied Soft Computing*, *121*, 108768.

[30.] Yu, J., Ye, X., & Li, H. (2022). A high precision intrusion detection system for network security communication based on multi-scale convolutional neural network. *Future Generation Computer Systems*, *129*, 399-406.

[31.] Soltani, M., Siavoshani, M. J., & Jahangir, A. H. (2022). A content-based deep intrusion detection system. *International Journal of Information Security*, *21*(3), 547-562.

[32.] Bhati, N. S., & Khari, M. (2022). A new ensemble based approach for intrusion detection system using voting. *Journal of Intelligent & Fuzzy Systems*, *42*(2), 969-979.

[33.] Sajith, P. J., & Nagarajan, G. (2022). Network intrusion detection system using ANFIS classifier. *Soft Computing*, 1-10.

[34.] Al-Enazi, M., & El Khediri, S. (2022). Advanced Classification Techniques for Improving Networks' Intrusion Detection System Efficiency. *Journal of Applied Security Research*, *17*(2), 257-273.

[35.] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150.