Hybrid Intrusion Detection System using Machine Learning Algorithms

Priyesha Jethi[1], Souhardyya Biswas[1], Saanchi Gangwani[1], Vanshikha Singh[1], Dr. Thandeeswaran R.[2]

[1]Student, School of Information Technology & Engineering (SITE), Vellore Institute of Technology, Vellore, Tamilnadu, India – 632014.
[2]Associate Professor Grade 1, School of Information Technology & Engineering (SITE), Vellore Institute of Technology, Vellore, Tamilnadu, India – 632014.

---***---

Abstract - *Data security in a networked computer system has become a key issue in today's world. Hackers and malevolent users are discovering new methods of network penetration as network traffic grows. An intrusion detection system (IDS) is being developed to address this issue, which will detect attacks in a computer network. The importance of information security and data analysis systems for Machine learning has recently altered due to the massive amounts of data and their incremental increase. An intrusion detection system (IDS) monitors and analyses data in order to detect any incursion into a system or network. The high volume, variety, and speed of data created in the network has made standard data analysis approaches for detecting attacks is extremely difficult. IDS use machine learning techniques to ensure that data analysis is accurate and efficient. To classify data in our research, we'll use a hybrid model of logistic regression, support vector machines, and the Random Forest Classifier and detect various types of intrusions like, DDoS, U2R, R2L, Backdoor, Injection, Password, XSS, M*

***Key Words*: Stacking Classifier, SVM, Random Forest, Logistic Regression, IPS**

1.INTRODUCTION

In this research, we have built a hybrid Intrusion Detection System using a combination of three Machine learning algorithms which are Logistic Regression, Support Vector Machine and Random Forest algorithms to classify and detect various types of intrusions like, DDoS, ransomware, Backdoor, Injection, Password, etc. Initially, we have pre-processed the dataset and split it into testing and training features. Then, we created a hybrid model and implemented it using the concept of stacking classifiers.

1.1 Machine Learning Algorithms used

Logistic Regression

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes. Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems.

Support Vector Machines

Support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or VC theory proposed by Vapnik (1982, 1995) and Chervonenkis (1974). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Random Forest Algorithm

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

Stacking Classifier

Stacking is an ensemble learning technique to combine multiple classification models via a meta-classifier. The individual classification models are trained based on the complete training set; then, the meta-classifier is fitted based on the outputs -- meta-features -- of the individual classification models in the ensemble. The meta-classifier can either be trained on the predicted class labels or probabilities from the ensemble. In this project we stacked the three ML models and created the hybrid model.

2. PAST WORK

Before starting our research, we conducted a literature survey which is as follows:

Reehan et al. [1] presented a paper where they proposed a discriminative component determination and interruption grouping dependent on SPLR for IDS. The SPLR is a method created for information examination and preparing through inadequate regularized streamlining that chooses a little subset from the first element factors to show the information with the end goal of characterization. A straight SPLR model expects to choose the discriminative components from the store of datasets and learns the coefficients of the direct classifier. Contrasted with the element choice methodologies, similar to channel (positioning) and covering strategies that are different from the element choice and order issues, SPLR can join highlight choice and grouping into a brought together structure. The analyses in this correspondence show that the proposed strategy has preferable execution over the vast majority of the notable strategies utilized for interruption recognition.

Deeman Yousif Mahmood [2] proposed a plan for a theory approach for planning a precise model for IDS as far as High Detection rates and execution with keeping a Low False Alarm rate by utilizing Logistic Model Trees (LMT). On high Dimensional datasets, this might prompt the model being Over-fit on the preparation set, which means exaggerating the exactness of expectations on the preparation set and along these lines the model will be unable to anticipate precise outcomes on the test set.

Alzahrania et al. [3] introduced a study where they inspected the significant and discriminative components, to perceive the different assaults by applying the Structural Sparse Logistic Regression (SSPLR) and Support Vector Machine (SVMs) methods. The SVMs are standard ML-based strategies, which give a sensible performance. Also, the scanty demonstrating (SSPLR) is considered as the high-level technique for information assessment and preparing through regularization. The primary scanty displaying can be utilized while selecting the unmistakable provisions or the gathering of discriminative components from the archive of the information collection to decide the coefficient of the straight classifier, where, earlier data of the element's construction can be planned on different sparsity-instigating regularizations. The tests and conversation show that the proposed procedures have further developed execution contrasted with the most cutting-edge strategies, utilized for the Intrusion Detection System (IDS).

Besharati et al. [4] planned a framework dependent on AI strategies to identify have based interruptions in the cloud. The proposed framework has the four periods of pre-handling, highlight choice, preparing classifiers and testing the information, where for include determination an ideal arrangement of components is chosen for each class utilizing the Logistic Regression calculation. Interruption discovery datasets are generally lopsided, to such an extent that the quantity of tests in various classes has an enormous distinction with one another. As needs be, in this examination, combinational Bagging classifier is utilized to order every information class and the classifiers utilized are neural organization, choice tree and the LDA.

Asghar et al. [5] presented a study in which the classification model Logistic Regression was applied on the dataset containing various types of intrusions. The performance of the classification algorithm was evaluated and compared. The metrics used for evaluation were Specificity, Accuracy, Sensitivity and MCC which helps in evaluating the prediction framework to acquire the respective True Positive, false-positive rate for two different varieties of Logistic Regression. Using 10-fold hyperparameter tuning and Jack-Knife, the model had a Sensitivity of 99%, Specificity of 96%, Accuracy of 99% and MCC of 98%.

Hosseini et al. [6] presented a new hybrid method, which is based on a combination of MGA-SVM and HGS-PSO-ANN techniques for attack detection. In the proposed method called MGA-SVM-HGS-PSO-ANN, finally, a combination of MGA and SVM is used to obtain the required features, and then ANN is used as a classifier for attack detection. It is further trained with a HGS-PSO algorithm to achieve optimal weights. SVM requires extensive training time and performs well with properly preprocessed dataset i.e. SVM does not perform very well when the data set has more noise i.e. target classes are overlapping. The proposed method had an accuracy of 99.3% when trained and tested using the NSL-KDD dataset.

Wang et al. [7] proposed a new and effective intrusion detection framework (DT-EnSVM) combining both the ensemble learning and data transformation technique. More specifically, the framework works by combining SVM ensemble with feature augmentation. This intrusion detection framework mainly consists of three parts, that is, data split, new data formation, ensemble-based intrusion detection model. This model achieves a robust performance in terms of accuracy, detection rate and training speed when compared to other already existing models.

Borkar et al. [8] presented a work in which an efficient classifier with data mining concept is introduced for the detection of intrusions accurately with less time. It uses an acknowledgement-based method which has 2 stages of classification known as adaptive SVM. Using an optimization algorithm for cluster head selection, the time consumption is reduced and scalability is improved. This method achieves better detection rate and prediction accuracy of the different types of intrusions.

Fang et al. [9] presented a paper that proposes a machine learning method for intrusion detection. This technique can fully exploit the Elman neural network and the advantages of a robust Support Vector Machine algorithm based on noise data elimination, and then combine the two to solve the safety risks of intrusion detection for various information systems to ensure their safety. The Elman neural network intrusion detection system clusters the text of the network packet by the clustering algorithm. In the meantime, the Support Vector Machine algorithm-based neighbour classification intrusion detection can accomplish the element space weighting of the ideal classification face framework log. It can then eliminate the negative impact of noise data and eventually reduce the false alarm rate, last but not least, improve the detection accuracy.

Safaldin et al. [10] published a paper in which an enhanced intrusion detection system is proposed by using the modified binary grey wolf optimizer with support vector machine. Generally, the proposed method is divided into three main stages, the first stage is feature selection using modified grey wolf optimization, the second stage is the classification using SVM, and the third stage is the evaluation stage to improve the performance of the applied methods. Hybridization of the modified Grey Wolf Optimiser algorithm with other efficient feature selection could be conducted to increase the detection rate in the WSN environment.

3. METHODOLOGY

In this research, we have built a hybrid Intrusion Detection System using a combination of three Machine learning algorithms which are Logistic Regression, Support Vector Machine and Random Forest algorithms to classify and detect various types of intrusions like, DDoS, ransomware, Backdoor, Injection, Password, etc. Initially, we have pre-processed the dataset and split it into testing and training features. Then, we created a hybrid model and implemented it using the concept of stacking classifiers.

3.1 Dataset Used

The dataset used was: **IOT Sensors Dataset**

It contains Normal (35,000 rows), DDoS (5000 rows), Injection (5000), Password (5000 rows), Backdoor (5000 rows), Ransomware (2865 rows), XSS (866 rows), and Scanning (529 rows). The file presents the data of temperature measurements, pressure readings, and humidity readings of a weather sensor linked to the network.

- **Logistic Regression Implementation**

Accuracy= 67%

- **SVM Implementation**

Accuracy=86%

- **Random Forest Implementation**

Accuracy=92%

We will be using a hybrid model of these algorithms as a novelty and to improve the accuracy of the algorithm. This will be achieved by using stacking classifiers.

4. ARCHITECTURAL DIAGRAM

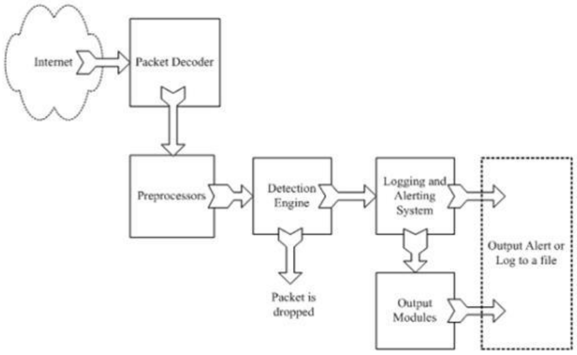


Fig -1: Block Diagram

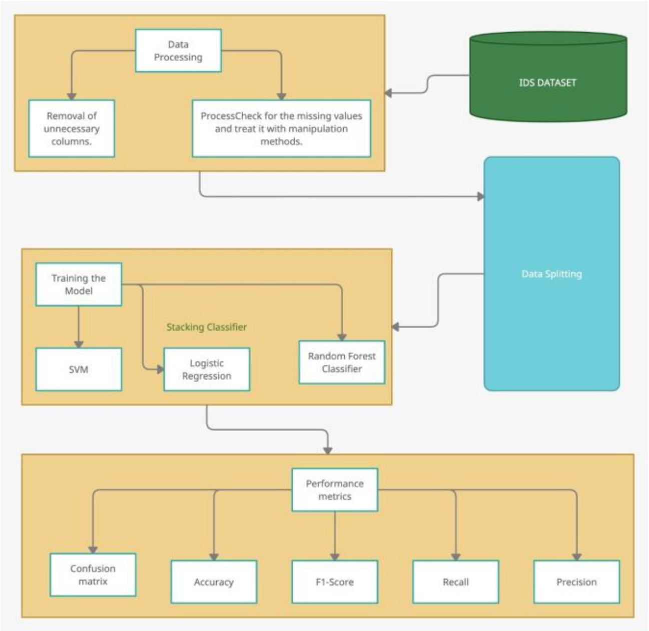


Fig -2: Architectural Diagram

5. IMPLEMENTATION

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from numpy import array
from numpy import argmax
from sklearn.preprocessing import LabelEncoder
```

data collection

```
from google.colab import files
```

```
uploaded = files.upload()
import io
df = pd.read_csv(io.BytesIO(uploaded['IDS.csv']))
df.head()
df.info()
```

data pre-processing

```
df.drop(columns=['date','time','ts','label'],inplace=True)
df.head()
df.type.value_counts()
label_type=LabelEncoder()
df['type']=label_type.fit_transform(df['type'])
df.type.value_counts()
type = {3:0, 4:4 ,2:2, 1:1, 0:3, 5:5, 6:6, 7:7}
df['type']=df['type'].map(type)
df.type.value_counts()
```

data splitting

```
from sklearn.model_selection import StratifiedShuffleSplit
split = StratifiedShuffleSplit(n_splits=1, test_size=0.25,
random_state=42)
for train_index, test_index in split.split(df, df['type']):
   train_set = df.loc[train_index]
   test_set = df.loc[test_index]
X_train =
train_set[['temperature','pressure','humidity']].values
Y_train = train_set[['type']].values
X_test = test_set[['temperature','pressure','humidity']].values
Y_test = test_set[['type']].values
```

Training and Testing of the hybrid model

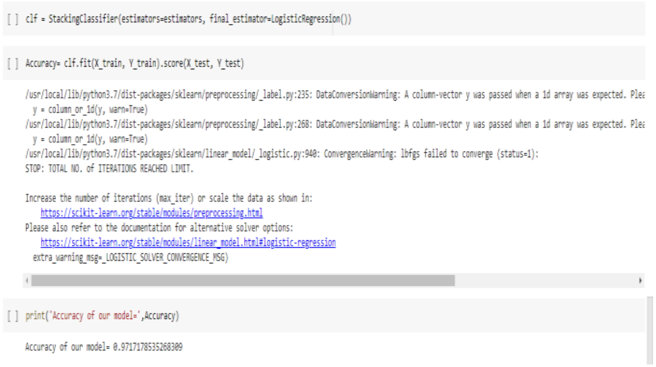
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.ensemble import StackingClassifier

estimators = [('rf', RandomForestClassifier(n_estimators=10,
random_state=42)),('svr', make_pipeline(StandardScaler(),
LinearSVC(random_state=42)))]
clf = StackingClassifier(estimators=estimators,

final_estimator=LogisticRegression())
Accuracy= clf.fit(X_train, Y_train).score(X_test, Y_test)
print('Accuracy of our model=',Accuracy)
```

6. RESULTS

The accuracy of the hybrid model is 97.17%. The hybrid model was developed using a stacking classifier which used a combination of logistic regression, random forest and support vector classifier.



```
[ ] clf = StackingClassifier(estimators=estimators, final_estimator=LogisticRegression())

[ ] Accuracy= clf.fit(X_train, Y_train).score(X_test, Y_test)

    /usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py:235: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Plea
      y = column_or_1d(y, warn=True)
    /usr/local/lib/python3.7/dist-packages/sklearn/preprocessing/_label.py:268: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Plea
      y = column_or_1d(y, warn=True)
    /usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:940: ConvergenceWarning: lbfgs failed to converge (status=1):
    STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

    Increase the number of iterations (max_iter) or scale the data as shown in:
        https://scikit-learn.org/stable/modules/preprocessing.html
    Please also refer to the documentation for alternative solver options:
        https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
      extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)

[ ] print('Accuracy of our model=',Accuracy)

    Accuracy of our model= 0.9717178535268309
```

Fig 3. Accuracy of the Hybrid Model

7. FUTURE SCORE

Recently, Cyber Physical Systems such as Data Acquisition Networks or SCADA and Unmanned Aerial Vehicles or UAV; these aforementioned networks are becoming intricate and complex. An Intrusion Detection System plays a vital role in such networks, where these can detect intruders by sniffing or inspecting the network traffic. The addition of Machine Learning increases the overall efficiency of the IDS and hence an extra edge to detect the cyber-attacks on the SCADA and UAV networks.

Intrusion Detection Systems based on Machine Learning may produce excellent results whilst detecting malicious cyber-attacks and intrusions. However, despite their growing popularity, these IDS systems are intricate and resource-intensive. They require high computational power, storage and high throughput. These factors pose major obstacles for these IDS systems to be established in real-time environments, due to high latencies. The Graphics Processing Units required for managing, deciphering and processing such massive datasets efficiently are pretty expensive. Hence the trade-off between performance and cost comes into play. In the future, cloud-based GPUs can be utilised (such as Google Cloud, providing NVIDIA K80, P100 etc.) to cut through this obstacle.

8. CONCLUSION

As stated above we built a hybrid Machine learning algorithm using Logistic Regression, Support Vector Machine and Random Forest algorithms to classify and detect various types of intrusions. For training and testing our algorithm we used a dataset containing various types of intrusions such as- a) DDos b) Backdoor c) Injection d) Password e) XSS f) Ransomware g) Scanning along with normal/standard values. The dataset used was a IOT sensors dataset. The accuracy of the hybrid model is 97.17%. However, when the same dataset is applied to the separate algorithms each of them has low accuracy compared to that of our hybrid model.

REFERENCES

[1] Reehan Ali Shah, Yuntao Qian, Dileep Kumar, Munwar Ali and Muhammad Bux Alvi, “Network Intrusion Detection through Discriminative Feature Selection by Using Sparse Logistic Regression”, MDPI Information (10 November 2017).

[2] Deeman Yousif Mahmood, “Classification Trees with Logistic Regression Functions for Network-Based Intrusion Detection System”, IOSR Journal of Computer Engineering (IOSR-JCE) Volume 19, Issue 3, Ver. IV (May - June 2017).

[3] Ahmed S. Alzahrani a, Reehan Ali Shah b , Yuntao Qian c , Munwar Ali, “A novel method for feature learning and network intrusion classification”, Alexanderia Engineering journal(2020).

[4] Elham Besharati1, Marjan Naderan1, Ehsan Namjoo , “LR-HIDS: logistic regression host-based intrusion detection system for cloud environments”, Journal of Ambient Intelligence and Humanized Computing(2018).

[5] Asghar Ali Shah, Nighat Usman, Jasir Waqar, Haseeb Saeed, “An Efficient Machine Learning Prediction based Model for Intrusion Detection”, International Conference on Innovative Computing (2019).

[6] Hosseini, S., & Zade, B. M. H. (2020). New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN. Computer Networks, 173, 107168.

[7] Gu, J., Wang, L., Wang, H., & Wang, S. (2019). A novel approach to intrusion detection using SVM ensemble with feature augmentation. Computers & Security, 86, 53-62.

[8] Borkar, G. M., Patil, L. H., Dalgade, D., & Hutke, A. (2019). A novel clustering approach and adaptive SVM classifier for intrusion detection in WSN. Sustainable Computing: Informatics and Systems, 23, 120-135.

[9] Fang, W., Tan, X., & Wilbur, D. (2020). Application of intrusion detection technology in network safety based on machine learning. Safety Science, 124, 104604.

[10] Safaldin, M., Otair, M., & Abualigah, L. (2021). Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. Journal of ambient intelligence and humanized computing, 12(2), 1559-1576.