A Course Project Report on

Team ID & Title:
CS2021W1036

Diverse Intrusion Detection & Prevention Systems

Submitted
as part of **CSE4003-Cyber Security**
by


20BCE0461
Anish Desai


20BCE0462
Rishikesh Suresh Kumar


20BCE0457
Ayush Kumar


20BDS0288
Aparajita Senapati


20BCE2798
Devashish Chaudhary

To


Dr M Rajasekhara Babu

School of Computer Science and Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

April 2022

Index

Abstract

An Intrusion Detection System (IDS) is a monitoring system that detects suspicious activities and generates alerts when they are detected. An Intrusion Prevention System (IPS) is a form of network security that not only detects malicious threats but is a step ahead and goes on to prevent/block the threats. A network intrusion detection and prevention system (NIDPS) is crucial for network security because it enables you to detect and respond to malicious traffic. Intrusion detection systems primarily use two key intrusion detection methods: signature-based intrusion detection and anomaly-based intrusion detection. Signature-based intrusion detection is designed to detect possible threats by comparing given network traffic and log data to existing attack patterns. Anomaly-based intrusion detection is designed to pinpoint unknown attacks, such as new malware. The intrusion event whose pattern or signature already exists in the network can be easily identified but Signature-based IDS cannot recognize new or previously unknown threats. Even the Anomaly-based detection system which is being thought of as an improvement over signature-based detection system cannot be used for real-time data traffic. Though the multi-feature data clustering optimization models boasts of improving the accuracy of detection techniques, it still resulted in a maximum accuracy of about 97.8% which is not very efficient in a real-time environment.

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XGBoost, the algorithm used in the implementation of our proposed IDS method makes use of CPU cache to calculate and store the gradients which makes the computation faster. XGBoost always gives more importance to functional space when reducing the cost of a model, while the other existing models gives more preference to hyperparameters to optimize the model, which makes our model computationally cost effective. Our proposed IPS model, at the heart of which lies our proposed ML-based IDS, is a flow-based intrusion prevention system. This model can be deployed in a real-time environment and prevent DoS attacks using its flow-based mechanisms. It is an improvement over the existing IPS models which use packet-based approach rather than flow-based, making it almost impossible to detect DoS and other flow-based attacks. Our proposed model has achieved an impressive accuracy rate of 99.81%. Besides this, the implementation of the approach proposed in this paper results in a relatively computational cost-effective IDPS.

Keywords:

Intrusion Detection Systems, Intrusion Prevention Systems, Signature-based Intrusion Detection, Anomaly-based Intrusion Detection, Malicious, XGBoost, DoS attacks, Probe attacks, R2L attacks, U2R attacks, Packet-based Mechanism, Flow-based Mechanism.

Chapter 1
INTRODUCTION

With rapid increase in Industrialization and Modernization of businesses and organizations, the dependency on network structures has increased exponentially. The data stored across networks is majorly confidential in nature, easy to diagnose and rebuild. Numerous large-scale attacks have been reported since past few years. In these circumstances, the protection of networks has become even more essential. Any cybersecurity breach can have major, unwanted and long-lasting consequences, impacting the organizations and possibly the state of economy and security of any region.

The fast-track developments seen in other technologies such as AI/ML have enabled us to search for alternative and robust security solutions. In this paper, we have used Machine Learning and Network Communication-based Technologies to propose solutions to Network Intrusions. Machine Learning algorithm is the basis of our proposed IDS – train and build model, test it and detect attacks. Network-based mechanisms and integration of IDS forms the basis of our proposed IPS – the IDPS. The dataset used in this paper is the latest version of NSL-KDD, the data for which has been gathered from Massachusetts Institute of Technology - Lincoln labs and simulates a typical US Air Force Local Area Network (LAN). The structure of the paper is as follows: Section 2 states our problem statement and the objectives of this research, Section 3 is a survey about the existing models and the gaps found in them, Section 4 informs about the software requirements and the Sections 5, 6 and 7 showcases the working algorithm with flow-charts and block-diagrams of our model. Section 9 concludes our paper.

IMPORTANCE OF IDEA:

- Network security has become essential more than ever. A robust accurate mechanism to detect and prevent various kinds of malicious attacks over networks is the need of the hour.
- IDPS is the mechanism that can deal with such problems over network structures. Security of the networks is directly related to the Performance of the IDPS model.
- An improvement in IDPS model using advanced new-age technologies provides solution to such threats.
- There is constant evolution in the ways a network structure is attacked, thus not only the performance of the IDPS model is necessary, but also the spectrum of attacks that it can cover.

The proposed IDPS model ensures that the gaps found in existing models are filled and a secure and reliable approach is obtained.

CHAPTER 2

PROBLEM STATEMENT & OBJECTIVES

2.1. Problem Statement:

To develop a cost-effective and accurate IDS using XGBoost and To propose a two-level diverse IPS.

2.2. Objectives:

1 To study various types of intrusion detection and prevention systems and their applications.

2 To develop a procedure to identify various modern attacks, generate alerts and prevent them as part of intrusion detection and prevention systems.

3 To design an intrusion detection system using XGBoost algorithm, which can detect various attacks like DoS, Probe, R2L, U2R.

4 To propose a software-solution IPS model which can prevent a wide range of attacks, including DDoS attacks.

5 To analyse the performance of our proposed model over accurately identifying the malicious attacks and computation cost.

CHAPTER 3
LITERATURE REVIEW

3.1. Existing models/methods/algorithms

Paper **[1]** explains in detail about Signature-based IDS methods, their types such as clustering-based approach, classification-based approach, decision tree-based approach, etc., their descriptions and advantages.

According to the paper mentioned above, Signature-based IDS model consists of four components:

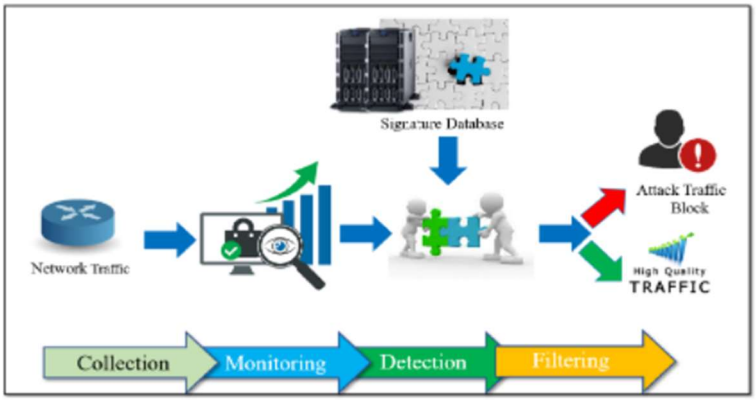

Figure 1: Methodology used in Signature based IDS

The **first component** involves collecting the network packets and then analysing them.
The **second component** immediately drops those packets and checks whether or not they correspond to the block table rules. Packets that have no autonomous element manager and have autonomous coordinator to those rules are forwarded to warning clustering module and it generates warning for suspicious packets.
The **third component** blocks the packets that are suspicious and sends warning to other IDSs.
The **fourth component** gathers warnings and makes packet decisions.

The approach proposed in the paper **[2]** consists of a two-stage architecture based on machine learning algorithms. In the first stage, the IDS uses K-Means to detect attacks and the second stage uses supervised learning to classify such attacks and eliminate the number of false positives.

3.1. Existing models/methods/algorithms

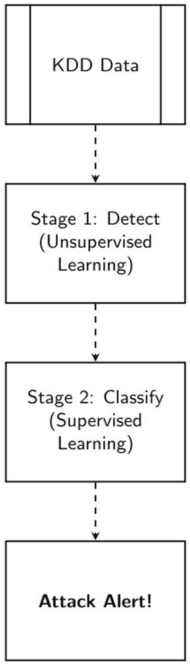

Fig. 2 Intrusion detection system architecture

The **first stage** uses a K-Means model to cluster the data and detect an attack. Clustering in this phase simply classifies the data into two categories: “attack” connections or “normal” connections. The design objective for this stage is to simply detect if there is an attack or not.

In the **second stage**, the objective is to lower the rate of false positives and improve accuracy. By classifying these attacks into a certain category, we increase the confidence that an attack is detected and appropriately classified. For this, four supervised-learning algorithms have been used:

J48 is a decision tree-based algorithm which classifies data. This algorithm builds decision trees by using information entropy and is based on C4.5 decision tree.

Random forest uses an ensemble learning method to combine decision trees.

Adaptive Boosting is another ensemble of machine learning algorithm.

Naive Bayes is another algorithm selected for use in the second stage of the intrusion detection system. This algorithm uses a probabilistic classification and is based on Bayes theorem.

It is proposed in the paper **[3]**, an industrial network intrusion detection algorithm based on **multi-feature data clustering optimization model**, where the weighted distances and security coefficients of data are classified based on the priority threshold of data attribute feature for each node in the network, given that the data modules in the industrial network environment are diverse and easy to diagnose, restore and rebuild.

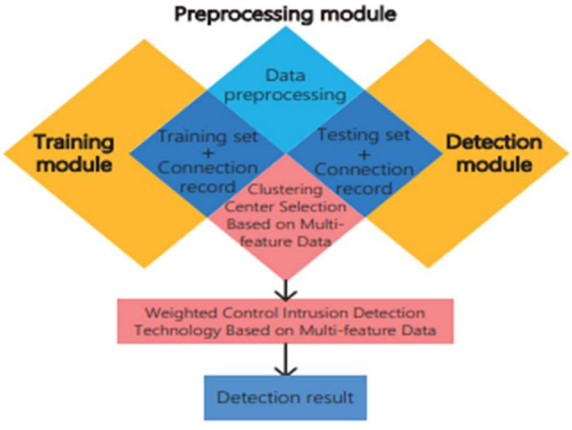

Fig. 4. Chart of multi-feature data detection

3.1. Existing models/methods/algorithms

The proposed algorithm mainly focuses on the extraction and pre-processing of useful data attribute features from the intrusion data. The results are analysed based on previous experience and actual situation. The clustering analysis algorithm used in industrial network intrusion detection is to use a specific artificial intelligence algorithm for data clustering analysis. Based on the distribution of data objects in each cluster, the cluster can be marked as normal or abnormal. After that, the cluster centre, radius, and the average security coefficient can be used to generate the detection rules.

In Paper **[4]**, the proposed method uses a two-level classifier structure - the first-stage classifier supports real-time classification, and the second-stage classifier supports accurate classification.

```
IF packet P is received THEN
    Consult firewall to find the matching policy for P.
    IF policy action is 'deny' THEN
      Drop P and RETURN
    END_IF
    Update bi-directional internal session states & bi-directional session stateful
    features.
    IF P is in the forward direction THEN
      Update forward internal session states and forward session stateful features.
    ELSE
      Update backward internal session states and backward session stateful features.
    END_IF
      Create derived session features and classify P using CPC.
    IF score < MCS THEN
      RETURN
    END_IF
    IF P is malicious THEN
      Add P session to blacklist of firewall.
    ELSE
      Add P session to whitelist of firewall.
    END_IF
ELSE IF expired session S is found THEN
    Create derived session features for S.
    Remove data for S from the session table.
    Classify S using TFC.
IF S is malicious THEN
    Notify the administrator and log the result.
END_IF
```

The entire proposed methodology is based on **Cumulative packet-based Classifier Detection** and the probability scores associated with CPC. The Minimum Cumulative Score (MCS) is considered the threshold value. If the Score of a packet is less than MCS, then it is dropped. If not, then it waits for IDS for the final decision – to blacklist or to whitelist the packet.

3.2. Gaps identified in existing literature

From the research in Paper **[1]**, we can infer that:

- The intrusion event whose pattern / signature already exists in the network can be easily identified but it cannot recognize new or previously unknown threats.
- The processor load for the device that analyses each signature depends on the database of the signature. The increasing the number of signatures that are searched for, the increasing the probability that more false positives will be found.
- **Signature-based IDS** are not optimal for attacks with self-modifying behaviour.
- Even the **Anomaly-based detection system** which is being thought of as an improvement over signature-based detection system cannot be used for real-time data traffic.

There are various models that have been analysed in Paper **[2]**, yet they have their own demerits:

- **J48** does not perform well once the feature with the most information gain of an attack falls into the wrong branch. J48 tree propagates that error for the rest of the leaves.
  If a J48 decision tree is not able to be configured properly, it results in a large tree and the algorithm denigrates pretty easy. If J48 outputs a complex tree it gives a poor performance and requires high computational power. J48 and decisions trees in general have limits when dealing with continuous data, or decisions which require more than one output per attribute.
- **Random Forest** relates to the same issues described in J48. Another disadvantage of random forest is the fact that it is hard to interpret it. In addition, a careful analysis is needed in deciding its configuration parameters (e.g numbers of trees) according to the data-set used, otherwise the performance accuracy will suffer.
- When we injected additional unknown attacks or outlier packages, **AdaBoost** suffered in their classification. This algorithm is not the most optimal solution for our problem, this is also often the case for other complex classification problems.
- The accuracy performance for **Naive Bayes** is the lowest among the algorithms, suggesting that the independence assumption is not a strong characteristic of the attacks found in our data. This means that Naive Bayes is an algorithm that does not perform well in data-sets where features are not independent of each other.

3.2. Gaps identified in existing literature

The model proposed in Paper **[3]** have some basic limitations which are too obvious to ignore:

- Though the method used in this paper boasts of improving the accuracy of detection techniques and reducing the rate of false positives, it still resulted in a maximum **accuracy** of about 97.8% which is still less when needed in real-time environment.
- This approach can **reduce** the storage of intrusion detection system.
- The approach proposed in this paper uses SVM-based and clustering analysis-based approach which itself has many disadvantages and **computational cost** is also high.
- The proposed method will waste plenty of time to recognize the classification label of massive data, as quoted in the research paper itself.

Coming to the Intrusion Prevention Model proposed in Paper **[4]**:

- The IPS proposed in this paper is a packet-based IPS. This Intrusion Prevention System focusses on **packet-wise detection** and analysis, excluding the flow-based analysis which is crucial to prevent especially DoS and DDoS attacks.
- With an increase in the number of packets, the detection time takes longer and increases the **probability of error**.

CHAPTER 4

REQUIREMENTS

4.1. Software Requirements

S.No	Item	Versions	Spec	Vendor	Price	Description	Reference
1.	Jupyter Notebook	Jupyter Notebook 6.3.0	512 MB RAM + 1 GB of disk	Project Jupyter	Free-ware	Python-based user interface where users can work with an ordered list of input/output cells to achieve Python Web server related tasks and deposit code solutions.	Project Jupyter | Try Jupyter
2.	Python	Python 3.10.0	2 GB RAM + 1 GB of disk	Python Software Foundation	Free-ware	High-level, general-purpose programming language	Welcome to Python.org

3.	Scikit-learn	Scikit-learn 1.0.1	2 GB RAM + 27 MB of disk	The scikit-learn developers	Free-ware	Key library for the Python programming language that is typically used in machine learning projects. Scikit-learn is focused on machine learning tools including mathematical, statistical and general-purpose algorithms that form the basis for many machine learning technologies.	scikit-learn: machine learning in Python — scikit-learn 1.0.2 documentation

Table 4.1: Summary of Software Components

Justification for the software usage

One of the software that we have used is Jupyter Notebook. The main advantage of Jupyter Notebook is its modularity approach. We can run cell by cell to better get an understanding of what the code does. It is also very easy to host server side, which is useful for security purposes. A lot of data is sensitive and should be protected, and one of the steps toward that is no data is stored on local machines. A server-side Jupyter Notebook setup gives you that for free.

Python is an advanced programming language. It works with less code, doesn't demand from the users to put a lot of code and thus reduces the number of tasks involved. Python has many in-built libraries that eases the work and the time and space complexity of any algorithm. It is open-source and has a vast community of programmers using python for a long time. It is the most sought-after tool by Data Scientists, AI/ML experts and Graphic Designers.

In the scikit learn, the documentation of the task is done in a proper way. Scikit learn library is extensive and has well-defined API documentation that is accessible from their website is provided. This library covers almost all the mainstream algorithms for machine learning tasks. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. It provides all the modular implementation of the ML algorithms.

No financial costs were incurred in the making of this project as:

1. The project is still at a rudimentary stage.
2. Free-wares were used in the making of models.

CHAPTER 5
Design

The **Block Diagram** for the proposed Intrusion Detection System (XGBoost IDS) is as follows:

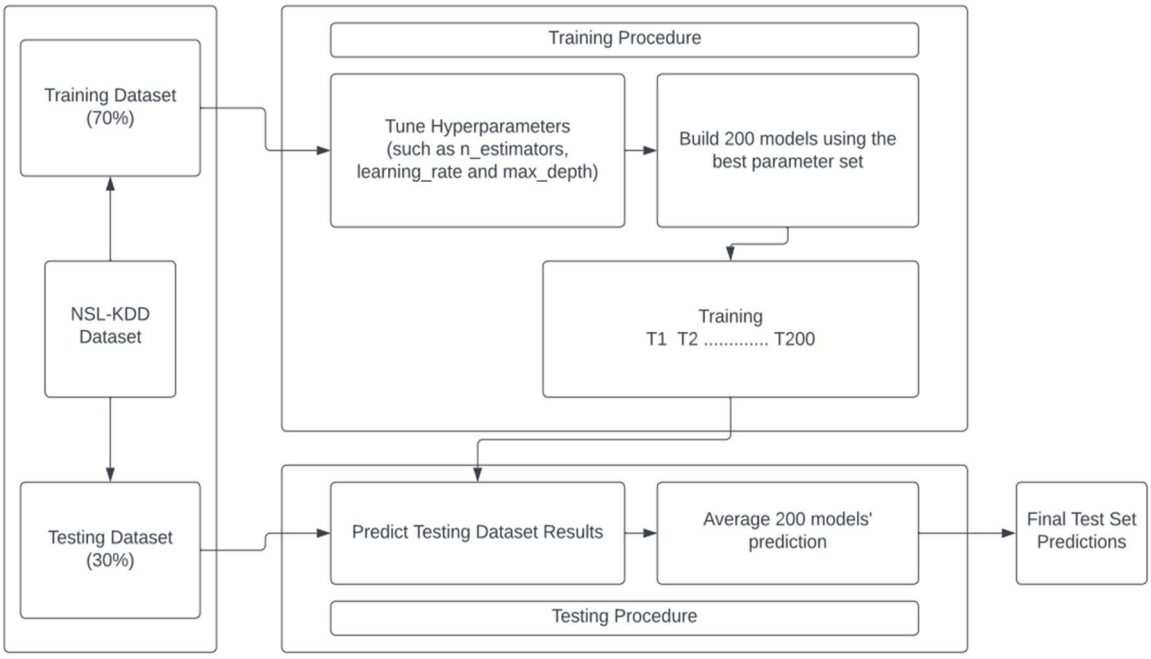

Figure 5.1: IDS Block Structure

The **NSL-KDD** dataset is split into two parts – 70% of the dataset for training the model and the rest 30% for testing the model. In the Training procedure, the training dataset is given as input and the hyperparameters for the XGBoost model is set to optimal standards. In the XGBoost Algorithm, the prediction model makes use of 200 decision trees to predict the results. The model is then given the testing dataset to analyse the working and its accuracy. The final results are thus obtained.

XGBoost is the heart of the IDS Architecture.

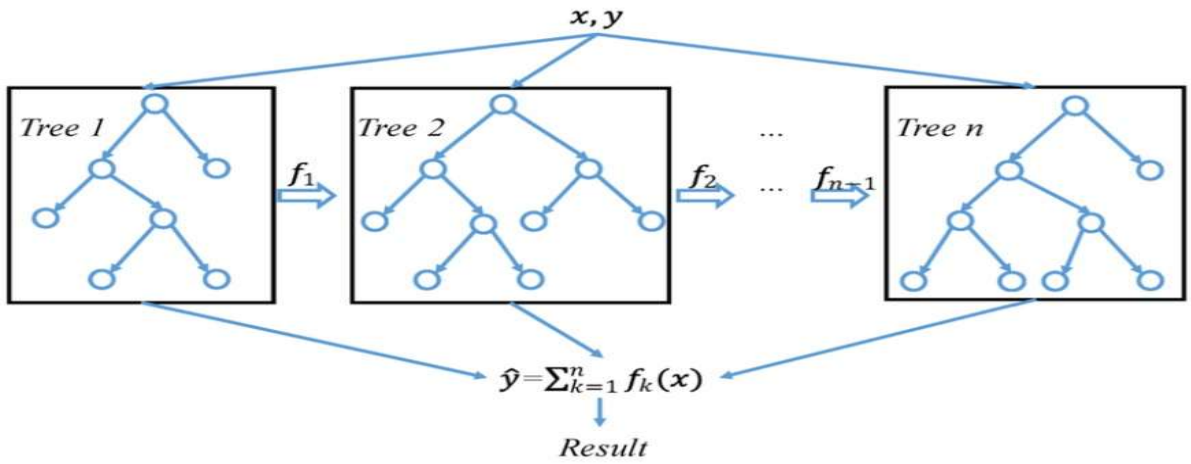

Figure 5.2: Basic Diagrammatic Architecture of XGBoost

The explanation below briefs about the architecture of the proposed IDS model and provides mathematical basis:

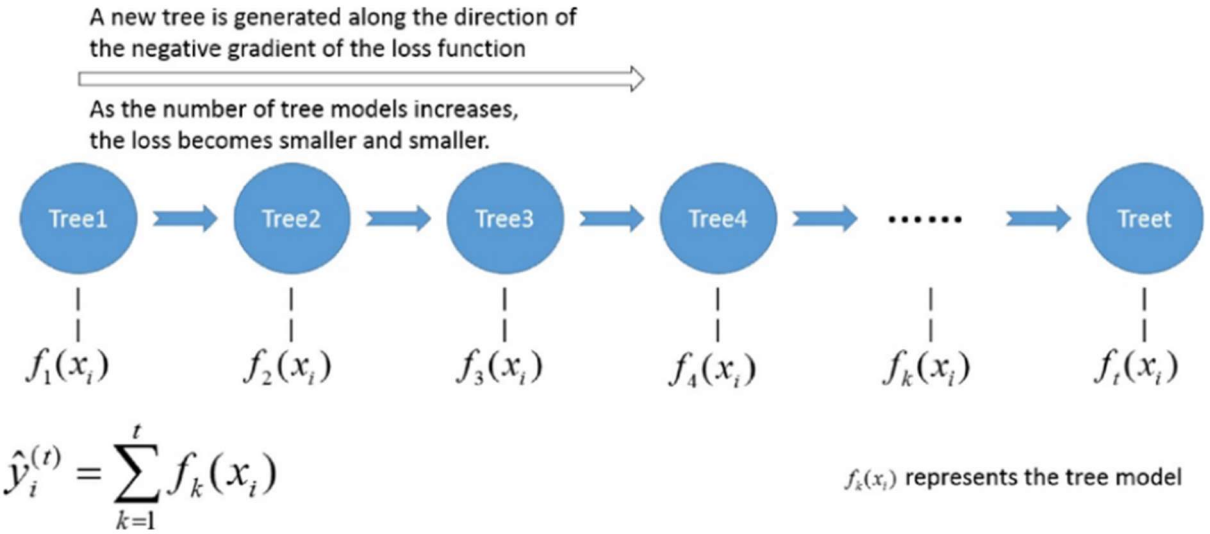

Figure 5.3: Mathematical Architecture of XGBoost

XGBoost is an optimized distributed gradient boosting system.
It is an iterative decision tree algorithm with multiple decision trees. Every tree is learning from the residuals of all previous trees. Rather than adopting most voting output results in Random Forest, the predicted output of XGBoost is the sum of all the results.

$$\hat{y}_i = \sum_{k=1}^{n} f_k(x_i), \quad f_k \in F$$

Figure 5.4: Output

Figure 5.4 gives the mathematical equation of the output calculated by XGBoost based on the features and the model decision trees. Here, F means the space of regression trees, f_k corresponds to a tree, so $f_k(x_i)$ is the result of tree k, and y_i is the predicted value of i^{th} instance x_i.

CHAPTER 6
IMPLEMENTATION

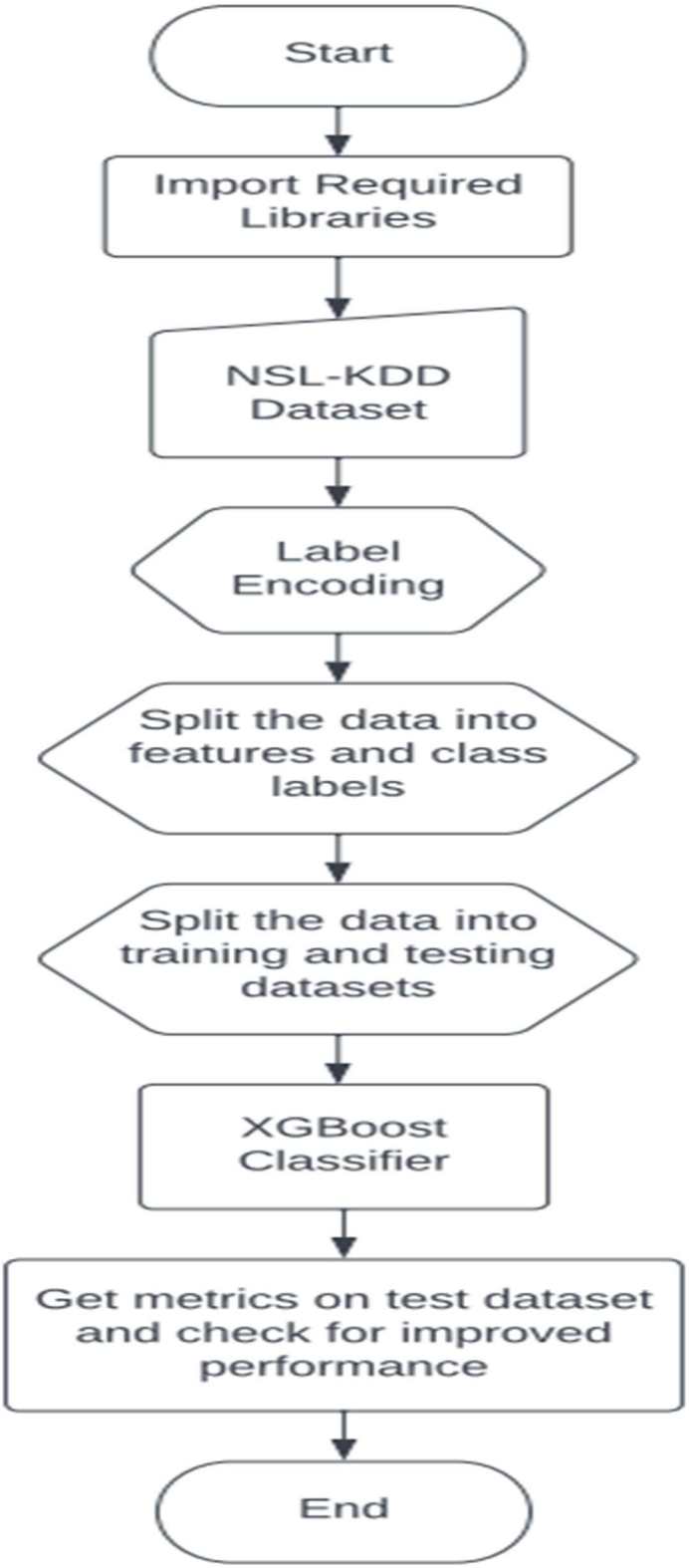

Figure 6.1: Flowchart of the proposed IDS model

Algorithm of the proposed IDS model:

1. Import Required Libraries.
2. Import the NSL-KDD dataset.
3. In the data pre-processing stage, use label encoder to convert the non-numeric columns to numerical factors.
4. Split the dataset into features and class labels.
5. Split the data into training and testing datasets in the ratio 70:30.
6. Apply XGBoost Classifier to train our model.
7. Get metrics on test dataset and check for improved performance.

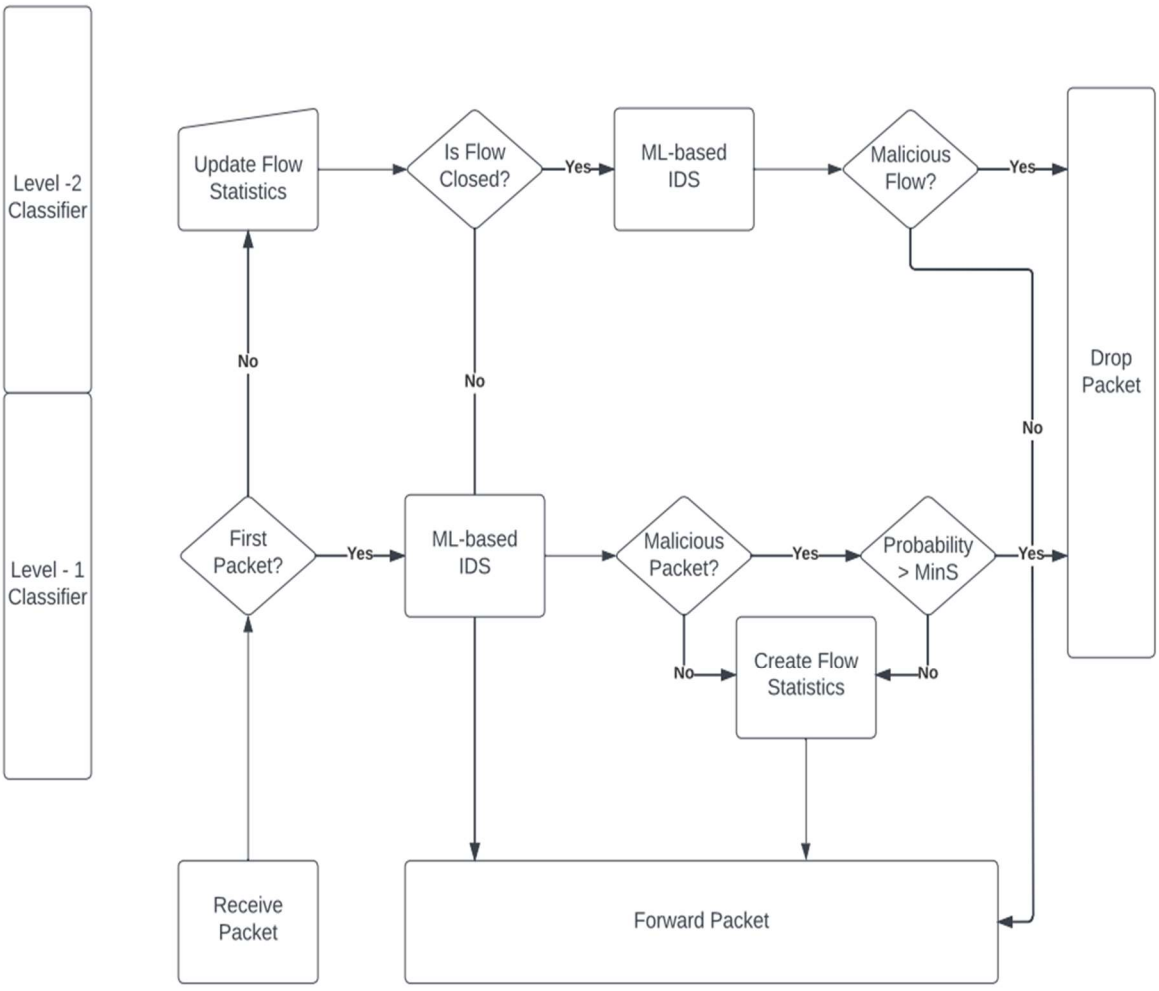

Figure 6.2: Flowchart of the proposed IPS model

Algorithm of the proposed IPS model:

```
Receive Packet
In Level 1 Packet-based Classifier
    IF Packet is First Packet of the Flow
        Use ML-based IDS
        IF Malicious Packet
            Use Naïve Bayes Probability Measures
            IF the score is more than the Minimum Probability Score
                Packet is dropped
            ELSE
                Packet added to the Flow-based Prevention Dataset
                Packet Forwarded
        ELSE
            Packet added to the Flow-based Prevention Dataset
            Packet Forwarded

In Level 2 Flow-based Classifier
    IF Packet is not First Packet of the Flow
        Update Flow Statistics
    IF Flow Closed
        Use IDS to detect Malicious Flow
            IF the flow is not malicious
                Forward
            ELSE
                Drop Packets and Block Flow
```

CHAPTER 7
RESULTS ANALYSIS

The basic parameter to test the efficiency of any classification model is Accuracy.

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy: %.2f%%" % (accuracy * 100.0))
```

```
Accuracy: 99.81%
```

Figure 7.1: Accuracy of the IDS model

Our model has outperformed several other IDS models with an improved performance of **99.81% accuracy**. Besides this, the **computational cost** has been drastically reduced due to the deployment of XGBoost Algorithm, as compared to other Ensemble learning algorithms, Hybrid Models and Multi-level Models which have near about similar accuracy.

```
import seaborn as sns
from sklearn.metrics import confusion_matrix
sns.set(rc={'figure.figsize':(4,4)})
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True)
#0 - anamoly
#1 - normal
```

```
<AxesSubplot:>
```

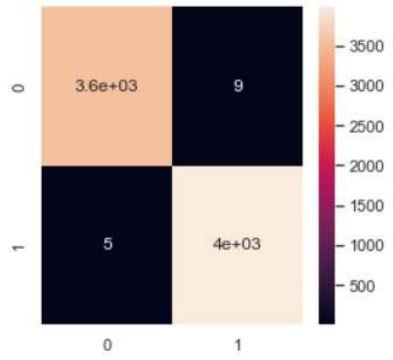

Figure 7.2: Confusion Matrix

This matrix is especially used to evaluate various classification models. This has four elements – true positives, false positives, false negatives and true negatives.
The **number of false positives and false negatives (5 & 9)** are negligible, thus proving the efficacy of our model.
Other various parameters such as precision and f1-score of the predictions have been obtained in Classification Report.

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
#0 - anamoly
#1 - normal
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      3561
           1       1.00      1.00      1.00      3997

    accuracy                           1.00      7558
   macro avg       1.00      1.00      1.00      7558
weighted avg       1.00      1.00      1.00      7558
```

Figure 7.3: Classification Report

The Precision and f1-score is **1.0** which implies that the model is pin-point precise in its predictions. This model has been trained using NSL-KDD dataset which features about 25000+ samples from network intrusions.

CHAPTER 8
Applicability category

The Intrusion Detection and Prevention Systems is a very essential component of any network structure. IDPS finds its applications across various sectors. Few of them are listed below:

1. With an ever-increasing amount of information in hospitals stored electronically, regulations and potential threats to that information have made Intrusion Detection Systems (IDS) a necessity. Although electronic information in hospitals helps enhance patient care, this same access introduces risks that must be addressed to ensure that this information is protected. Not only is this protection of PHI the right thing to do, various legislations regarding public health information make it mandatory.

2. An increasing urbanization of the society has propelled the use of internet networks across all the regions. IDPS acts as a robust security layer for all LANs and WANs along with firewall.

3. IDPS is also deployed in various places to protect not only the network or software but also the hardware associated with them. A Network intrusion or Software corruption event poses threat to the hardware, especially the Operating systems.

4. Various educational institutions use online portal to regulate various procedures and day-to-day functioning of the institution. The portal contains all the information about the students, staff and others associated. This makes it essential to have a diverse IDPS system in place to protect all the sensitive and general information alike.

5. Intrusion Detection and Prevention Systems can be extended to all those companies, institutions and organizations which have their data (especially confidential) stored in their local machines. There is always a persistent threat looming over such data by attackers.

Our proposed model sits perfectly with all the requirements by these industries and organizations – An improved accuracy of ML-based IDS, Perfectly Precise Predictions, Relatively Lesser Computational Cost wrt. existing models, and an Integrated IDPS consisting of Flow-based IPS with an ML-based IDS embedded in it. The proposed flow-based IPS approach is an impressive improvement over the existing packet-based models.

CHAPTER 9
Conclusions

We have thus proposed an improved approach to solve the problems looming over the networks. The Ensemble-learning based Intrusion Detection System detects a wider range of attacks with greater accuracy than most existing models and has relatively lesser computational costs. The XGBoost algorithm used has had its stable release in January 2022, proving its efficiency. It focusses on functional space and reducing cost without any kind of compromise in the accuracy of detections. The proposed model of Two-level Diverse Intrusion Prevention System (IPS) incorporates Flow-based Mechanism and IDS which makes it suitable in even real-time environment. Our models may not be able to process and detect all the malicious activities in a real-time network, besides the cost of hardware for such an infrastructure would be pretty high. Despite these limitations, our paper is of great significance for having integrated an ML-based IDS to a real-time flow-based IPS.

Chapter 10
REFERENCES

[1] Snehi, Jyoti. (2020). Diverse Methods for Signature based Intrusion Detection Schemes Adopted.

[2] Kaja, N., Shaout, A. & Ma, D. An intelligent intrusion detection system. Appl Intell 49, 3235–3247 (2019). https://doi.org/10.1007/s10489-019-01436-1

[3] W. Liang, K. -C. Li, J. Long, X. Kui and A. Y. Zomaya, "An Industrial Network Intrusion Detection Algorithm Based on Multifeature Data Clustering Optimization Model," in IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 2063-2071, March 2020, doi: 10.1109/TII.2019.2946791.

[4] Yeongje Uhm & Woonguil Pak, Real-Time Network Intrusion Prevention System Using Incremental Feature Generation (2021).

[5] G. Yedukondalu, G. H. Bindu, J. Pavan, G. Venkatesh and A. SaiTeja, "Intrusion Detection System Framework Using Machine Learning," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021, pp. 1224-1230, doi: 10.1109/ICIRCA51532.2021.9544717.

[6] Leon Reznik, "Intrusion Detection Systems," in Intelligent Security Systems: How Artificial Intelligence, Machine Learning and Data Science Work For and Against Computer Security , IEEE, 2022, pp.109-176, doi: 10.1002/9781119771579.ch3.