A Report

On

LAYERED APPROACH USING CONDITIONAL RANDOM FIELDS FOR INTRUSION DETECTION

by

Manan Dublish

BITS ID: - 2022MT12008

Dissertation work carried out at

Defence Services,
Jhansi
(Indian Army)

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE,PILANI

VIDYA VIHAR, PILANI, RAJASTHAN - 333031.

April 2024

LAYERED APPROACH USING CONDITIONAL RANDOM FIELDS FOR INTRUSION DETECTION

BITS ZG628T: Dissertation
by

Manan Dublish

BITS ID:- 2022MT12008

Dissertation work carried out at

Defence Services, Jhansi

Submitted in partial fulfilment of **M.Tech (Software Systems)** degree programme

Under the Supervision of **Lt Col Manjunath Bilur**

Defence Services, Jhansi

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE,PILANI

VIDYA VIHAR, PILANI, RAJASTHAN - 333031.

April 2024

Certificate

This is to certify that the Dissertation titled “Layered Approach using Conditional Random Fields for Intrusion Detection” and submitted by Col Manan Dublish having BITS ID : 2022MT12008 for the partial fulfilment of the requirements of M.Tech (Software Systems) degree of BITS embodies the bonafide work done by him under my supervision

Place : Jhansi

Date: 29 Apr 24

(Lt Col Manjunath Bilur)

ACKNOWLEDGEMENTS

I wish to register my profound gratitude to my supervisor, Lieutenant Colonel Manjunath Bilur, and Additional Examiner, Lieutenant Colonel Sushil Maurya for their guidance and course correction while undertaking this study. Their extensive experience of working in cyberspace-related billets in the Indian Army assisted me in understanding the nuance of the problem statement and recommending practical measures, which I have tried to include in the report.

I would also like to acknowledge with much appreciation the crucial role of Mr Hemant Rathore, BITS Pilani, Goa Campus, BITS faculty mentor, for providing deep insightand aspects for improvement in my study report. His vision as a learned scholar in thediscipline and comments during the evaluation of abstract and mid-sem reports helped me make the recommendations practical and implementable while working under the organization's constraints.

I would also like to convey my special thanks to Col Ajay Fuloria, Advisor Cyber security, Army Cyber Group for hisinputs during the conduct of my study.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI(RAJASTHAN)

WILP Division

Abstract Sheet

Organization: Indian Army **Location**: Jhansi

Duration: 5 Months **Date of Start**: Jan 24

Date of Submission: 30 April 24

Title of the Project: Layered Approach Using Conditional Random Fields for Intrusion Detection

Student ID No. 2022MT12008

Name of the student: Manan Dublish

Name (s) and Designation (s) of Supervisor and Additional Examiner:

Supervisor: Lieutenant Colonel Manjunath Bilur, M Tech
(IIT Mumbai) Lieutenant Colonel (Indian Army)

Additional Examiner: Lieutenant Colonel Sushil Maurya, M Tech (BITS Pilani)
Lieutenant Colonel (Indian Army)

Name of the Faculty mentor: - Mr Hemant Rathore, BITS Pilani, Goa Campus

Key Words: Layered Approach, Conditional Random Field, Hybrid Algorithm

Project Areas: INTRUSION DETECTION, CYBER SECURITY

ABSTRACT

A network invasion detection system must function effectively to handle high network traffic volumes and consistently identify hostile activity occurring within a network, among other problems. In this research, we use CRF and Layered Approach to tackle these two problems: Accuracy and Efficiency. We show that the Layered Approach may be implemented to achieve high efficiency and high accuracy in attack detection while employing Conditional Random Fields. One of the most important and difficult jobs for network administrators and security experts is intrusion detection. More complex penetration techniques are developed by attackers to overcome installed security systems as a result of increasingly sophisticated security technologies. Last but not least, our approach offers the benefit of having more levels, hence flexibility. Future studies can make use of our approach to extract features that can help with the creation of signatures for systems that rely on signatures. Implement Signature-based systems on the periphery of a network to screen out known and common assaults, and to identify novel attacks use anomaly and hybrid systems.

Signature of Student

Name: Lt Col Manan Dublish

Date: 30 Apr 2024

Place: Jhansi

Signature of Supervisor

Name: Lt Col Manjunath Bilur

Date: 30 Apr 2024

Place: Delhi

Table of Contents

List Of Figures

1. INTRODUCTION

Intrusion detection is defined as the skill of identifying improper, erroneous, or unusual activities. One of the most important and difficult jobs for network administrators and security experts these days is intrusion detection. More complex penetration techniques are developed by attackers to overcome installed security systems as a result of increasingly sophisticated security technologies. Therefore, by creating more dependable and effective intrusion detection systems, it is necessary to protect the networks from known vulnerabilities while also taking action to identify fresh but potential system threats. There are various prerequisites that come with every intrusion detection system. Its main goal is to identify the greatest number of attacks with the least amount of false alarms.

The aim of an intrusion detection system will not be served by an accurate system that is unable to manage high volumes of data and has a delayed decision-making process. We want a system that can handle massive volumes of data, quickly enough to make judgments in real time, detects the majority of threats, and generates very few false alarms. Around the 1980s, intrusion detection became popular following the seminal paper by Anderson. Intrusion detection systems are distinguished based on how they are deployed and the data they analyze. Additionally, depending on the attack detection technique, intrusion detection systems can be categorized as anomaly or signature. Certain patterns, or signatures, are extracted from previously identified assaults and used to train signature-based systems.

2. LITERATURE SURVEY

2.1 Intrusion Detection:

IDS are used to recognize unauthorized attempts to access, alter, or disable computers, usually across a network. These attempts could come in the form of various attacks (see the list of potential attacks in Appendix A). Attacks within correctly encrypted communication cannot be directly detected by IDS.

IDS can include sensors that sense security events, a console that monitors and controls the sensors, and a central engine that logs the events that the sensors record in a database and uses a set of rules to create alerts based on the security events it receives. All three parts are often found in a single appliance or device in basic IDS setups.

2.2 Wireless Sensor Network (WSN):

A wireless sensor network is a network of dispersed devices that work together to collaboratively monitor various physical and environmental factors at different locations, such as temperature, motion, sound, vibration, pressure, or pollution. The initial use was seen in military, specifically for battlefield monitoring. Nonetheless, a wide range of civilian application domains, such as traffic control, home automation, healthcare, and environment and habitat monitoring, currently employ wireless sensor networks.

Every node in a sensor network normally has one or more sensors, a microcontroller, a battery, and. It is possible for a single sensor node to be as small as a shoebox or as small as a grain of dust, yet actual tiny "motes" that can function have not yet been developed. In a similar vein, the price of sensor nodes varies greatly, from several hundred dollars to pennies, contingent upon the scale of the sensor network and the level of complexity that each sensor node must meet.

Constraints on sensor node size and cost translate into commensurate limitations on resources like memory, bandwidth, energy, and computing speed.

There are several significant rivals in this well-established business sector, including Cisco and Network Associates. It is true that IDS products themselves miss a lot of known assaults and generate a lot of false positives. IDS product development, however, is probably going to follow the same trajectory as previous antivirus software development. Each time the user created a new file, the original antivirus program sounded an alert. They are confident that the anti-virus program, which is currently operating, can detect every known infection.

3. PROPOSED SYSTEM

In order to create reliable and effective intrusion detection systems, the dual issues of accuracy and efficiency have been addressed in this paper. The results of our experiment, which are included as Appendix B, demonstrate the effectiveness of CRFs in raising the attack detection rate and lowering the FAR. Moreover, the time needed to train and test the model is greatly decreased by feature selection and the application of the Layered Approach. Future studies can make use of our approach to extract features that can help with the creation of signatures for systems that rely on signatures.

3.1 Layered Approach Using Conditional Random Fields For Network Intrusion Detection

One popular and often used technique for identifying malicious activity throughout a network is network monitoring. However, due to the high volume of network traffic, it might not be possible to monitor every event in real time, even in a moderately sized network. Because of this, pattern matching can only be done with attack signatures, which can only, at most, identify attacks that have already been discovered. When anomaly-based solutions are utilized for event analysis, audit data is dropped. Because of this, network monitoring frequently consists of examining just the audit data's summary statistics. Features of a single TCP session between two IP addresses or network-level characteristics like the load on the server, the number of incoming connections per unit time, and others may be included in the summary statistics. The KDD 1999 data set contains these statistics. Using the Layered CRF, precise anomalous intrusion detection systems that function well on fast networks may be constructed. The system offers the following benefits in particular:

1. When CRFs are used, the accuracy of attack detection for particular sub-systems increases.

2. Every subsystem in the system is trained to identify attacks that fall within a specific attack

class, giving the entire system broad attack detection coverage.

3. High-speed networks are capable of efficiently detecting attacks.

4. The system is less affected by noise.

3.1.1 Layered Approach For Intrusion Detection

Here is a detailed description of the Layer-based Intrusion Detection System (LIDS). The Airport Security paradigm, in which many security checks are carried out sequentially one after the other, serves as the inspiration for the LIDS. Like this paradigm, the LIDS is built on guaranteeing the CIA triad over a network. It is a sequential Layered Approach. Reducing computation and the total amount of time needed to identify anomalous events is the aim of employing a layered model. The amount of time needed to identify an invasive event is considerable, but it can be shortened by removing the overhead of communication between several layers. Making the layers independent and self-sufficient to thwart an attack without the need for a central decision-maker can help achieve this. In the LIDS framework, each layer is deployed sequentially after undergoing independent training. The four attack groups listed in the data set are represented by the four layers we define.

Then, using a limited number of pertinent features, each layer is independently trained. Certain features might exist in multiple levels to provide the layers their independence. In essence, the layers serve as filters that stop any unusual connection, therefore removing the need for additional processing at later layers and allowing for a prompt reaction to incursion.

Because of this, we use the LIDS and choose a subset of features for each layer instead of utilizing all 41 features. As a result, the system performs significantly better during both

the training and testing phases. There is frequently a trade-off between the system's accuracy and efficiency, and there are a number of ways to enhance its performance.

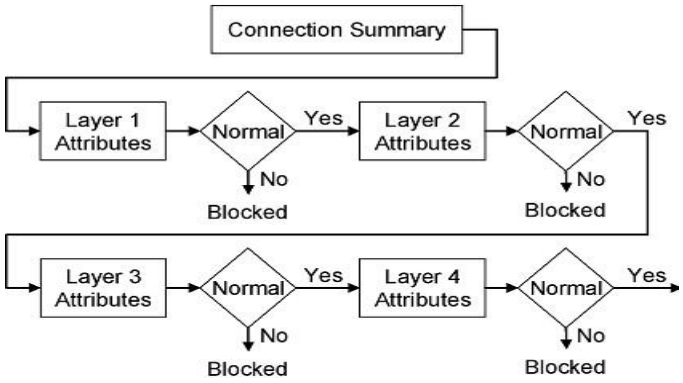


Fig 1: Integrating Layered Approach

3.1.2 Integrating Layered Approach With Conditional Random Field

By decreasing the amount of false alarms, CRFs can effectively increase the accuracy of attack detection, and the Layered Approach can be used to increase system efficiency as a whole. Therefore, integrating them to create a single system that is accurate in identifying assaults and effective in operation is a logical decision. Even though the CRFs are more costly, we use the Layered Approach to increase system performance in general. Our suggested approach, Layered CRFs, performs similarly to Decision Trees and Naive Bayes, and it has a greater accuracy rate for detecting attacks.

3.2 Feature Selection:

3.2.1 Probe Layer

The goal of the probe attacks is to obtain data from a source, usually outside the network, about the target network. Hence, while statistics like "number of file creations" and "number of files

accessed" are not expected to provide information for detecting probes, basic connection level features like "duration of connection" and "source bytes" are important.

3.2.2 DoS Layer

Traffic features like the "percentage of connections having same destination host and same service" and packet level features like the "source bytes" and "percentage of packets with errors" are important because they are used in DoS attacks to force the target to stop providing the service(s). It might not matter if a user is "logged in or not" in order to identify DoS attacks.

3.2.3 R2L Layer

Due to their involvement of both host and network level elements, R2L attacks are among the hardest to identify. In order to detect R2L assaults, we thus chose to use both host-level information like the "number of failed login attempts" and network-level features like the "duration of connection" and "service requested."

3.2.4 U2R Layer

Semantic details involved in U2R attacks are extremely hard to acquire in the early stages. These kinds of assaults typically target an application and are content-based. Therefore, we choose to ignore features like "protocol" and "source bytes" and instead focus on features like "number of file creations" and "number of shell prompts invoked" for U2R assaults.

3.3 Hybrid Machine Learning Algorithm combining Decision Tree, Naïve Bayes, and Random Forest

In the subsequent paragraphs, various algorithms have been individually analyzed with their pros & cons. Creating a hybrid machine learning algorithm that combines Decision Trees, Naive Bayes, and Random Forests involves integrating the strengths of each algorithm to potentially improve overall predictive performance. Here's a high-level overview of how to approach this.

3.3.1 Decision Tree

Popular machine learning algorithms called decision trees are well-liked for their interpretability, simplicity, and efficiency across a variety of applications. Using decision trees has the following benefits and drawbacks:

Pros:

(i) **Interpretability**: Decision trees give the decision-making process a concise, understandable depiction. They are useful for communicating forecasts to stakeholders who are not technically inclined since they may be graphically represented and emulate human decision-making.

(ii) **Managing Non-linearity and Interactions**: Without the need for feature engineering or transformation, decision trees are able to capture non-linear relationships and interactions between features.

(iii) **Handles Mixed Data Types**: Decision trees are adaptable to a wide range of dataset types since they can handle both categorical and numerical data.

(iv) **No Assumptions about Data Distribution**: Decision Trees are resistant to outliers and skewed distributions because, in contrast to some other algorithms (like linear regression), they do not make any assumptions about the distribution of the data.

(v) **Feature Importance**: Decision Trees offer a metric for feature importance, which can be used to determine which features are most pertinent for forecasting.

(vi) **Efficiency**: Decision trees can handle big datasets with ease and have a quick training period.

Cons:

(i) **Overfitting**: Decision trees can overfit, particularly if the dataset is noisy or the tree depth is not appropriately managed. This problem can be lessened by employing ensemble techniques like Random Forests, pruning, and defining the maximum depth.

(ii) **Instability**: Decision trees are unstable because even slight changes in the data might result in various tree architectures. Multiple trees are combined in ensemble approaches such as Random Forests to address this.

(iii) **Prejudice Toward Dominant Classes**: Decision Trees tend to prefer the majority class in datasets with unequal class distributions, which results in biased predictions. This bias can be reduced by employing strategies like employing ensemble methods or balancing class weights.

(iv) **Greedy Nature**: Decision Trees greedily split nodes, selecting the optimum split at each stage without taking the optimality of subsequent splits into account. This may result in less-than-ideal trees.

(v) **Sensitive to Small Variations**: Different tree architectures may result from decision trees' sensitivity to minute changes in the training set. Random Forests and other ensemble techniques can help lessen this sensitivity.

(vi) **Limited Expressiveness**: Compared to more intricate models like neural networks, which can capture extremely non-linear correlations in the data, decision trees are less expressive.

Notwithstanding these drawbacks, Decision Trees' simplicity, interpretability, and versatility make them an invaluable and popular machine learning algorithm. Decision trees are strong, simple to use, and capable of handling both category and numerical data. They are a decent place to start, but they are readily overfitted and sensitive to noisy data.

3.3.2 Naïve Bayes

Based on the Bayes theorem and the supposition of feature independence, this algorithm is a straightforward yet effective machine learning technique. Using Naive Bayes has the following benefits and drawbacks:

Pros:

(i) **Easy and Quick**: Naive Bayes scales well with big datasets and high-dimensional feature spaces and is computationally efficient. It's especially helpful for applications where speed is of the essence and real-time predictions.

(ii) **Simple to Implement and Interpret**: This algorithm is easy to use and comprehend, making it appropriate for novices as well as a starting point for comparing more intricate algorithms.

(iii) **Performs Well with High-Dimensional Data**: Naive Bayes is effective even in situations when there are a lot more features than cases. It works well for high-dimensional data applications like document categorization and text classification.

(iv) **Robust to Irrelevant Features**: Because Naive Bayes relies on feature independence, it is resistant to irrelevant features. This reduces the likelihood of overfitting brought on by the dimensionality curse.

(v) **Handles Missing Values**: Naive Bayes is resilient to incomplete datasets since it can handle missing values by simply disregarding them during training and prediction.

(vi) **Requires Less Training Data**: Naive Bayes is helpful in situations when data availability is constrained because it can still function well with tiny training datasets.

Cons:

(i) **Feature Independence Assumption**: The primary drawback of Naive Bayes is the robust feature independence assumption, which might not hold for a large number of real-world datasets. In real life, features frequently correlate, which results in less-than-ideal performance.

(ii) **Inaccurate Probability Estimation**: Naive Bayes frequently generates inaccurate probability estimates, particularly for uncommon or unobserved events. The term "zero-frequency problem" or "Laplace smoothing" refers to this problem.

(iii) **Sensitive to Feature Distribution**: If the features have a complicated dependency structure, Naive Bayes may not hold true in assuming that the features are conditionally independent given the class label.

(iv) **Restricted Expressiveness**: In comparison to more intricate models like decision trees or neural networks, Naive Bayes has a restricted expressiveness. It can have trouble capturing the complex relationships between the features in data.

(v) **Difficulty Managing Continuous Features**: Although Naive Bayes excels at handling categorical features, it could struggle to handle continuous features as well. To solve this problem, methods like binning or kernel density estimation might be applied.

(vi) **Unable to Learn Interactions between Features**: Naive Bayes is unable to learn interactions between features because it is predicated on the idea that features are independent of one another. This may hinder its performance on tasks where these interactions are crucial.

Naive Bayes is nevertheless a well-liked and often used algorithm in many different applications, particularly in text categorization, spam filtering, sentiment analysis, and recommendation systems, despite these drawbacks. It is an important instrument in the machine learning toolkit because of its ease of use, quickness, and efficacy with high-dimensional data. Based on Bayes' theorem, Naive Bayes is a probabilistic classifier that operates under the presumption that feature independence, which might not hold in many real-world scenarios.

3.3.3 Random Forest

By constructing several decision trees and combining their predictions, this algorithm—a potent ensemble learning technique—improves accuracy and robustness. The Random Forest algorithm has the following benefits and drawbacks:

Pros:

(i) **High Accuracy**: By averaging the outputs of several decision trees, Random Forest reduces the risk of overfitting when compared to individual decision trees, leading to generally very accurate predictions.

(ii) **Robustness to Overfitting**: By combining predictions from several decision trees trained on various subsets of the data and characteristics, Random Forest reduces overfitting. As a result, a more generic model with good performance on unknown data is produced.

(iii) **Handles High-Dimensional Data**: Random Forest works well with datasets that have a lot of features, which makes it appropriate for high-dimensional data like text, image, and genetic data.

(iv) **Feature Importance**: Random Forest offers a feature importance metric that aids in determining which characteristics have the most predictive power. The selection and interpretation of features can benefit from this knowledge.

(v) **Handles Missing Data**: Random Forest can deal with data that has missing values without the need for imputation. It just doesn't take into account missing values when building and predicting trees.

(vi) **Minimizes Variation**: Random Forest stabilizes the model's predictions and minimizes variation by mixing many decision trees, producing more dependable results.

Cons:

(i) **Complexity of Computation**: Random Forest can be computationally costly, particularly when there are a lot of trees and characteristics. Compared to simpler algorithms, Random Forest training and prediction may take longer.

(ii) **Lack of Interpretability**: Although Random Forest generates feature importance scores, it is harder to interpret due to its ensemble structure than it is for individual decision trees. It could be difficult to comprehend how traits and predictions relate to one another.

(iii) **Memory Consumption**: Random Forest necessitates the memory storage of several decision trees, which can be memory-intensive, particularly for big ensembles with deep trees.

(iv) **Black Box Model**: Since Random Forest conceals the underlying decision-making mechanism, it is referred to as a "black box" model. It could be challenging to comprehend how each tree adds to the ensemble's forecasts.

(v) **Biased Towards Majority Classes**: Random Forest may exhibit bias towards the majority class in datasets with uneven class distributions, resulting in inferior outcomes for minority classes. This problem can be solved using methods like class weighting or resampling.

(vi) **Regression Task Difficulty**: Random Forest works very well for classification tasks, but it might not work as well for regression tasks, particularly if there is a strongly non-linear

relationship between the features and the target variable.

Despite these drawbacks, Random Forest's outstanding performance across a variety of datasets and its capacity to manage challenging issues with high-dimensional data make it one of the most well-liked and often used machine learning methods. Several decision trees are constructed using the Random Forest ensemble learning technique, which then combines them to produce a prediction that is more reliable and accurate. When compared to individual decision trees, they minimize overfitting and work effectively with big datasets.

3.4 Here's a basic outline of how to combine these algorithms into a hybrid model:

3.4.1 **Feature Engineering**: Take care of missing values, encode categorical variables, preprocess your data, and scale numerical features as necessary. This stage guarantees that the format of your data is appropriate for modeling.

3.4.2 **Individual Models**: Use the training data to train each model independently. Optimize hyperparameter performance by fine-tuning them with methods like cross-validation.

3.4.3 **Group Formation**:

3.4.3.1 **Voting Classifier**: Utilizing a voting mechanism (such as "hard" or "soft"), aggregate the predictions from all three models. The final prediction is determined by either the average probability or the majority vote, respectively.

3.4.3.2 **Stacking**: To determine the optimal way to combine the outputs of the three base models, train a different model (meta-learner) using their predictions.

We are creating and developing a Voting Based Hybrid technique in the suggested method, which combines DT, NB, and RF.

3.4.4 **Model Evaluation**: Using the proper evaluation metrics (e.g., accuracy, precision, recall, F1-score, ROC AUC), assess the performance of your hybrid model.

3.4.5 **Fine-Tuning**: To further enhance performance, you could wish to investigate alternative ensemble methodologies or fine-tune hyperparameters based on how well your hybrid model performs.

3.4.6 **Deployment**: After you're happy with the performance, put your hybrid model into use in production to forecast fresh, untested data.

Thorough feature engineering, model selection, and hyperparameter tuning are essential to the hybrid model's performance. Try out various combinations and approaches to determine which model performs the best for a given situation.

3.5 HARDWARE & SOFTWARE SPECIFICATION:

3.5.1 Hardware Specification:

Processor : Pentium IV 2.8GHz.

RAM : 512 MB RAM.

Hard Disk : 40 GB.

Input device : Standard Keyboard and Mouse.

Output device : VGA and High-Resolution Monitor.

3.5.2 Software Specification:

Operating System: Windows XP

Language: JDK 1.5.

4. DESIGN ARCHITECHTURE

4.1 Data Flow Diagram:

The suggested data flow diagram for a Layered Approach and Conditional Random Fields is as depicted.

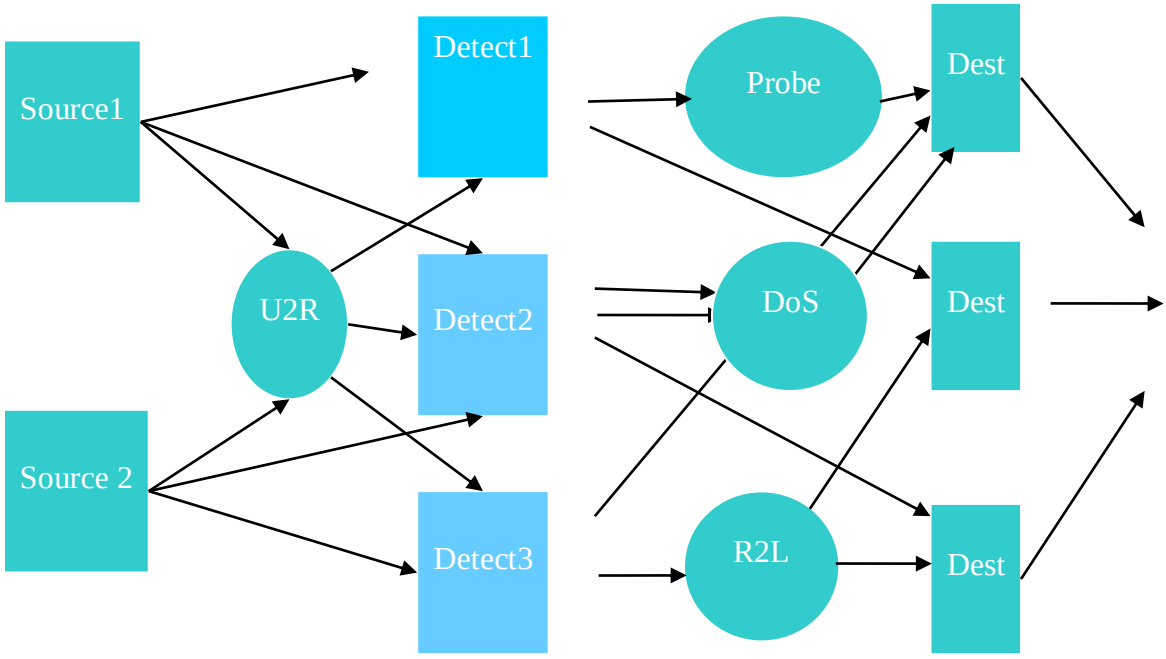


Fig 2 : Data Flow Diagram

4.2 Module Diagram:

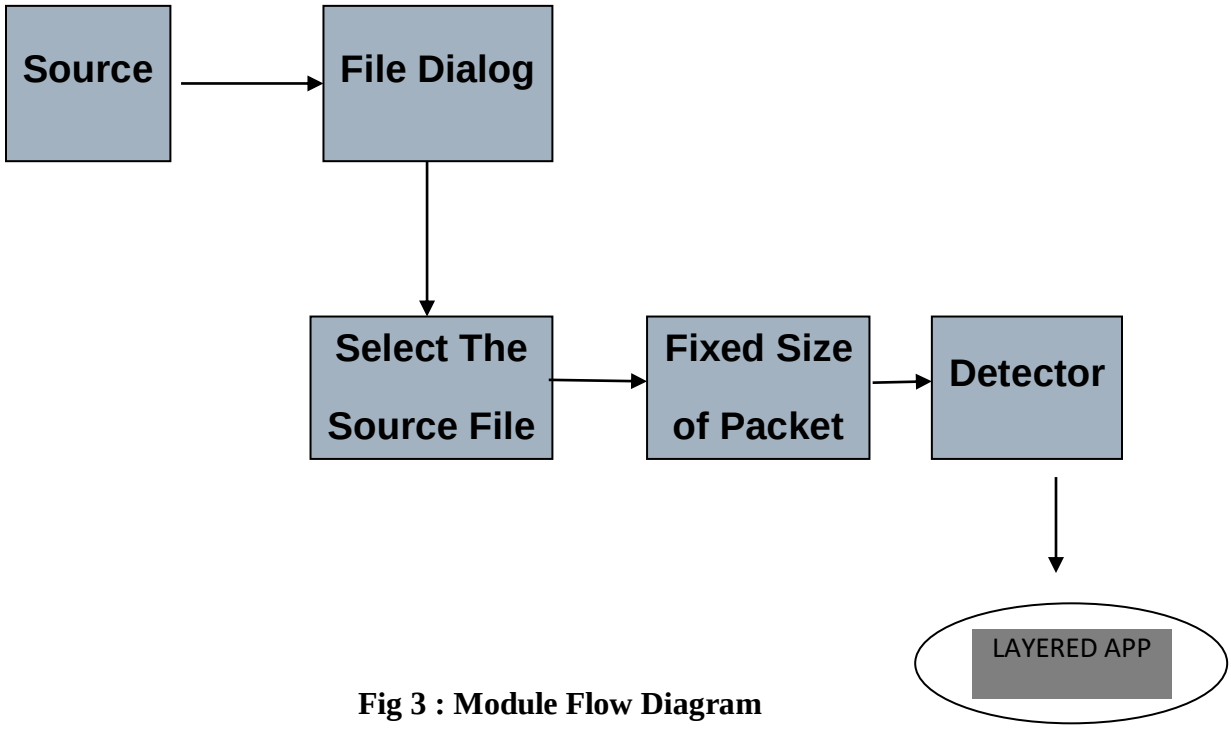


Fig 3 : Module Flow Diagram

4.3 Use Case Diagram:-

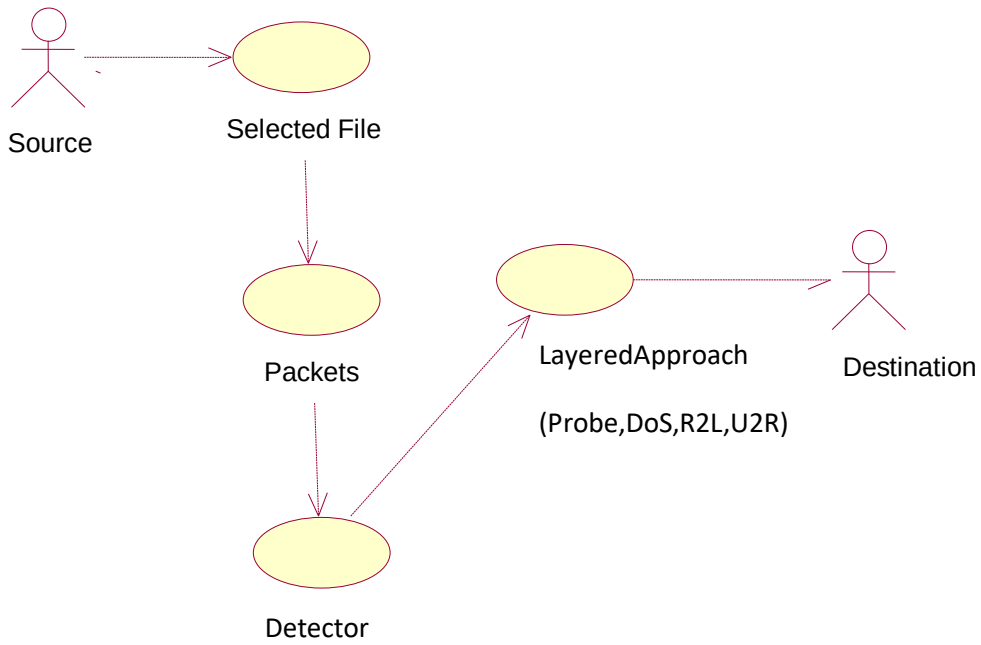


Fig 4 : Use Case Diagram

4.4 Sequence Diagram:-

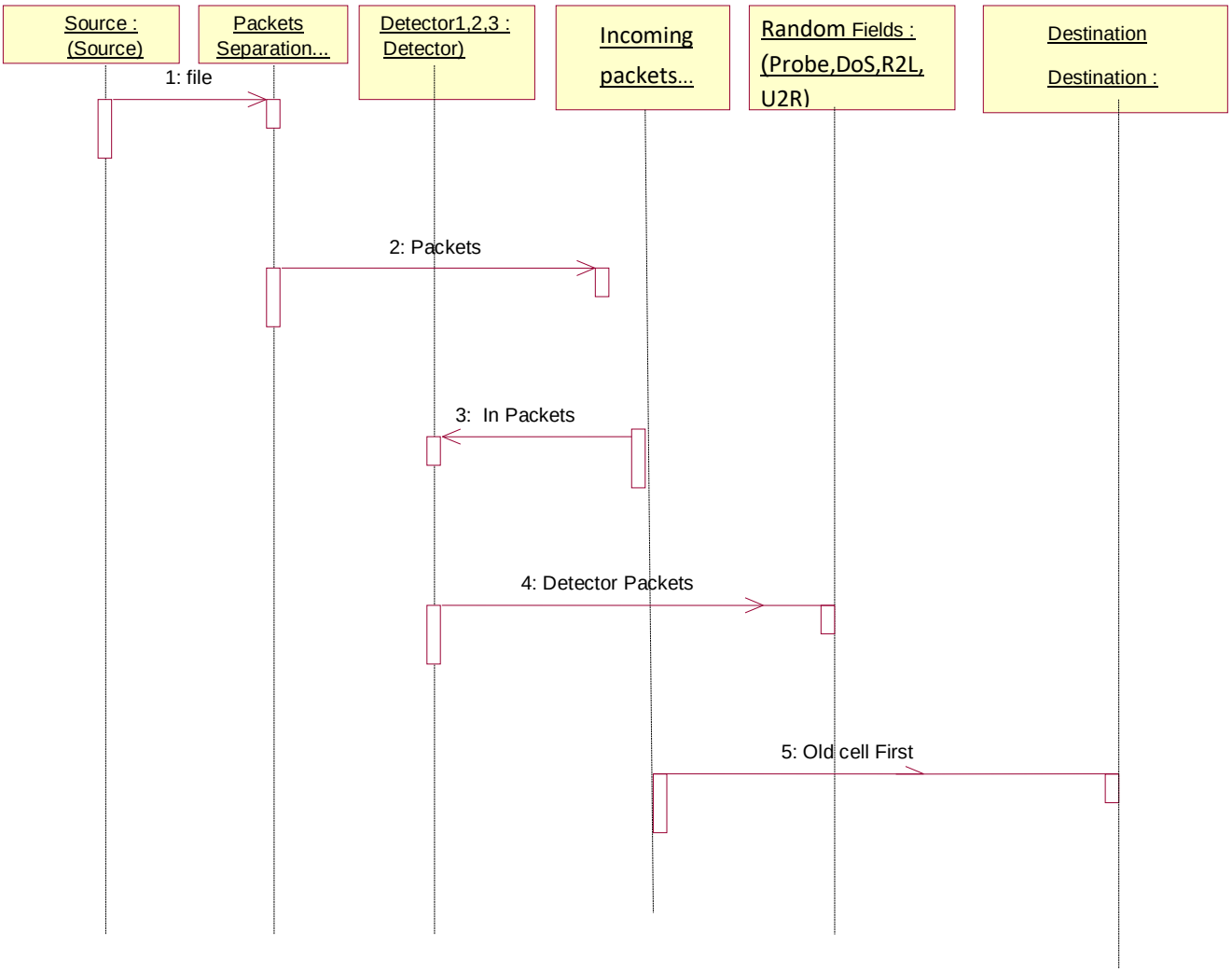


Fig 5 : Sequence Diagram

CONCLUSION:

To create reliable and effective intrusion detection systems, two issues of accuracy and efficiency have been addressed in this work. A low false alarm rate (FAR) is crucial for any intrusion detection system. Also, training and testing time is reduced by feature selection and the application of the Layered Approach. Techniques like CRFs can be quite successful in identifying attacks and outperform other techniques.

On contrasting this strategy with a few well-known techniques, we discovered, while many current intrusion detection techniques are unable to consistently identify R2L and U2R attacks, this hybrid system is capable of doing so with effectiveness and efficiency. Enhancing our system can assist in recognizing an attack as soon as it is identified at a certain layer, which speeds up the intrusion response process and lessens the effect of an assault. When applied to relational data, sequence analysis techniques like CRFs enable us to use the Layered Approach, as demonstrated in this study. This can be further expanded to multicore CPUs to enable pipelining of layers, which should yield very high performance.

ATTACK DESCRIPTION

Apache2

This attack exploits the inability of some versions of the Apache web server to handle very long HTTP requests. A typical attack contains multiple requests each with thousands of lines and looking something like this:

GET / HTTP/1.1

User-Agent: sioux

User-Agent: sioux

ARPpoison

An attacker who has compromised a host on the local network disrupts traffic by listening for "ARP-who-has" packets and sending forged replies. ARP (address resolution protocol) is used to resolve IP addresses to Ethernet addresses. Thus, the attacker disrupts traffic by misdirecting traffic at the data link layer

DoS attack

A *denial-of-service attack* or *distributed denial-of-service attack (DDoS attack)* is an attempt to make a computer resource unavailable to its intended users. Although the means to, motives for, and targets of a DoS attack may vary, it generally consists of the concerted, malevolent efforts of a person or persons to prevent an Internet site or service from functioning efficiently or at all,

temporarily or indefinitely by choking the network bandwidth, and/or consuming computing resources like memory and CPU.

Fragment overlap attack

A TCP/IP Fragmentation Attack is possible because IP allows packets to be broken down into fragments for more efficient transport across various media. The TCP packets (and its header) are carried in the IP packet. In this attack the second fragment contains incorrect offset. When packet is reconstructed, the port number will be overwritten

IPsweep

An IPsweep attack is a surveillance sweep to determine which hosts are listening on a network. This information is useful to an attacker in staging attacks and searching for vulnerable machines

Land

This is a Denial of service attack where a remote host is sent a UDP packet with the same source and destination

Mailbomb

This attack floods a user with thousands of junk emails. This type of attack can be detected by the fact that the SMTP “mail” command is lowercase. It is normally uppercase but not required to be

Neptune Floods the target machine with SYN requests on one or more ports, thus causing Denial of service

Phf attack

The Phf attack abuses a badly written CGI script to execute commands with the privilege level of the http server. Any CGI program which relies on the CGI function escape_shell_cmd() to prevent exploitation of shell-based library calls may be vulnerable to attack. In particular, this vulnerability is manifested by the "phf" program that is distributed with the example code for the Apache web server.

PoD

This attack, also known as "*ping of death*", crashes some older operating system by sending an oversize fragmented IP packet that reassembles to more than 65,535 bytes, the maximum allowed by the IP protocol. It is called "ping of death" because some older versions of Windows 95 could be used to launch the attack using "ping -l 65510"

Smurf

This is a distributed network flooding attack initiated by sending ICMP ECHO REQUEST packets to a broadcast address with the spoofed source address of the target. The target is then flooded with ECHO REPLY packets from every host on the broadcast address.

TCPreset

This attack listens for TCP SYN packets on a compromised host on the local network and immediately sends a spoofed RST (connection refused) packet, disrupting traffic.

Teardrop

This attack reboots the Linux host by sending a fragmented IP packet that cannot be reassembled because of a gap between the fragments.

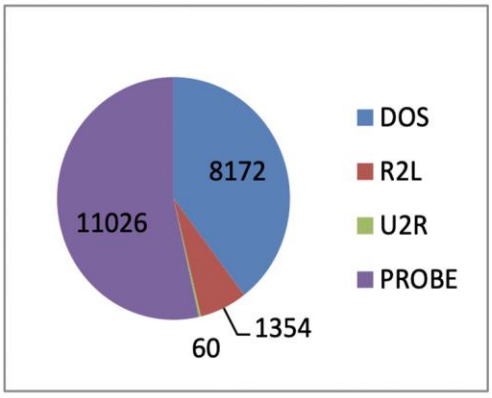
UDPstorm

An attacker floods the local network by setting up a loop between an *echo* server and a *Client machine* or another *echo* server by sending a UDP packet to one server with the spoofed source address of the other.

Exploratory Data Analysis

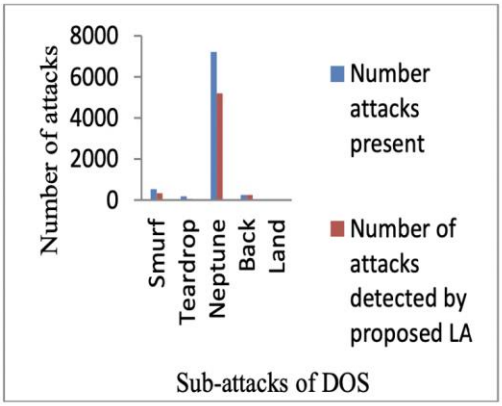
PERFORMANCE BENCHMARK

Name of attack	Number of attacks present in Input file	Number of attacks detected by proposed Layered Approach
DOS	8172	5824
R2L	1354	1192
U2R	60	36
PROBE	11026	9533



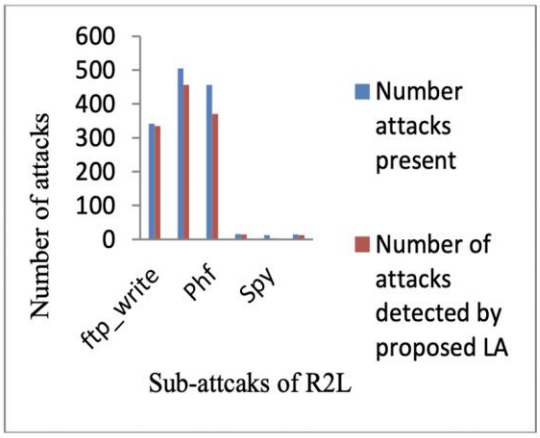
BENCHMARK - DoS ATTACK

Name of sub-attacks	Number of attacks present	Number of attacks detected by proposed LA
Smurf	527	340
Teardrop	183	49
Neptune	7212	5190
Back	249	245
Land	1	0



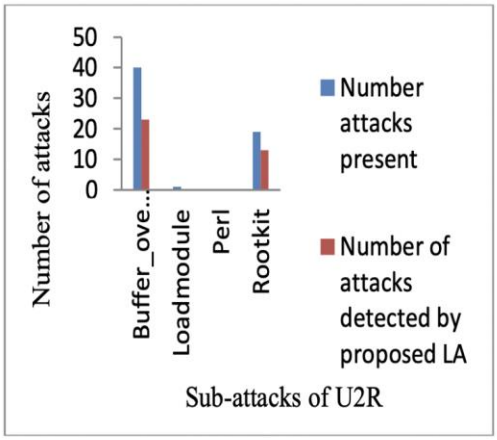
BENCHMARK - R2L ATTACK

Name of sub-attacks	Number of attacks present	Number of attacks detected by proposed LA
ftp_write	341	335
Warezclient	505	456
Phf	456	370
Multihop	16	15
Spy	13	3
Warezmaster	15	13



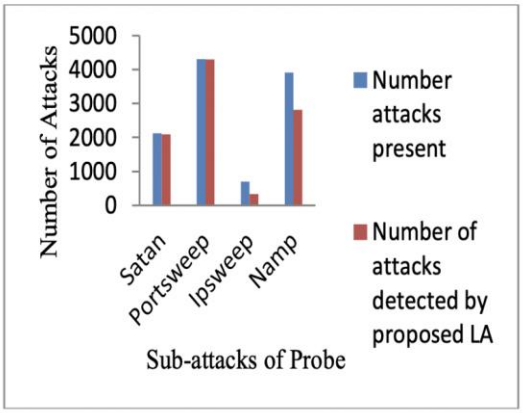
BENCHMARK - U2R ATTACK

Name of Sub-attacks	Number of attacks present	Number of attacks detected by proposed LA
Buffer_overflow	40	23
Loadmodule	1	0
Perl	0	0
Rootkit	19	13



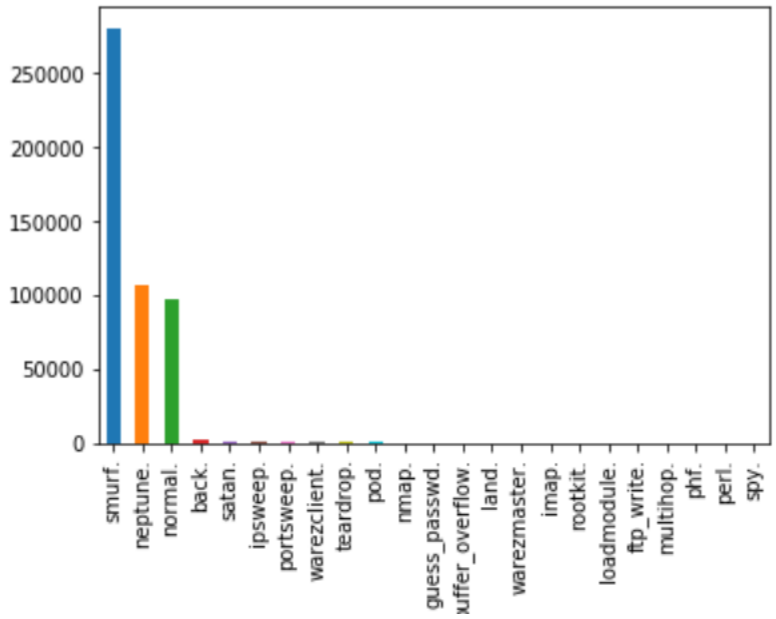
BENCHMARK - PROBE ATTACK

Name of sub-attacks	Number of attacks present	Number of attacks detected by proposed LA
Satan	2119	2089
Portsweep	4302	4293
Ipsweep	700	336
Namp	3905	2815



For this proposed work KDD cup data set will be considered, which has 50+ attributes and attack types.

Distribution of the target variable in the dataset.



Multinomial Naïve Baysian

Performance analysis:

```
precision    recall  f1-score   support

         dos       0.93      0.99      0.96    117444
      normal       0.99      0.55      0.71     29172
       probe       0.00      0.00      0.00      1264
         r2l       0.00      0.00      0.00       310
         u2r       0.00      0.53      0.00        17

   micro avg       0.90      0.90      0.90    148207
   macro avg       0.39      0.42      0.34    148207
weighted avg       0.93      0.90      0.90    148207

Accuracy Score: 0.8965973267119637
Classifier Training time = []
Classifier Prediction time = 0.060544490814208984
```

Decision Tree algorithm

```
precision    recall  f1-score   support

         dos       0.98      0.94      0.96    117444
      normal       0.95      0.89      0.92     29172
       probe       0.08      0.60      0.15      1264
         r2l       0.44      0.42      0.43       310
         u2r       0.58      0.65      0.61        17

   micro avg       0.92      0.92      0.92    148207
```

macro avg	0.61	0.70	0.61	148207
weighted avg	0.97	0.92	0.94	148207

Accuracy Score: 0.9240454229557309

Classifier Training time = [1.2965433597564697]

Classifier Prediction time = 0.1235041618347168

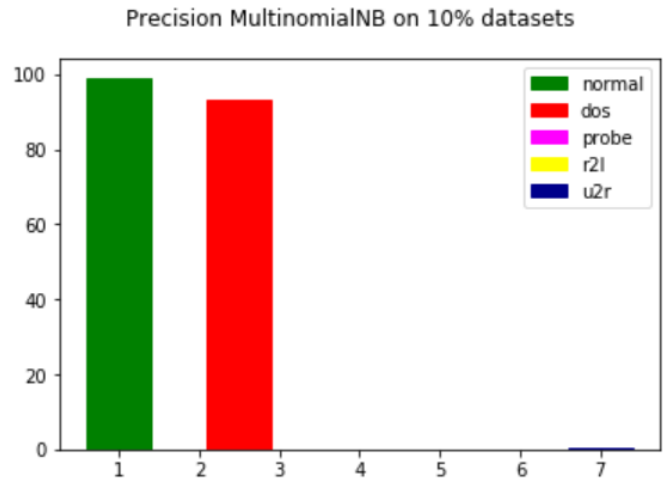
Random Forest

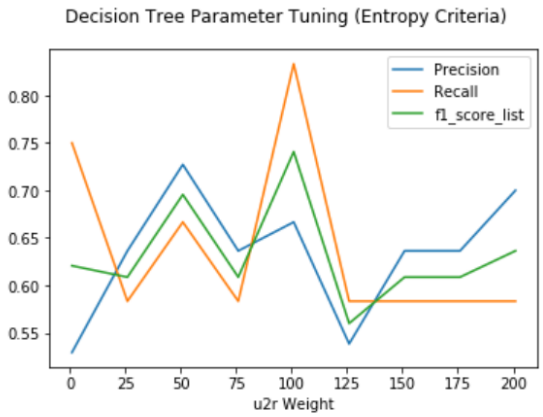
precision	recall	f1-score	support
dos	0.92	0.99	0.95	117444
normal	0.99	0.63	0.77	29172
probe	0.00	0.00	0.00	1264
r2l	0.00	0.00	0.00	310
u2r	0.00	0.00	0.00	17
micro avg	0.91	0.91	0.91	148207
macro avg	0.38	0.33	0.35	148207
weighted avg	0.92	0.91	0.91	148207

Accuracy Score: 0.9120351940191759

Classifier Training time = [1.2965433597564697, 1.621917963027954]

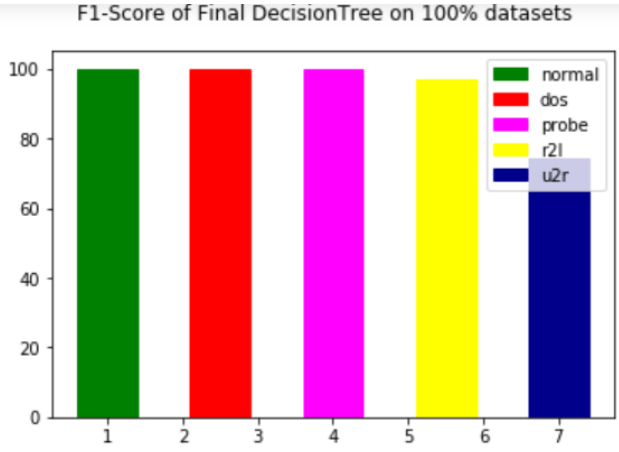
Classifier Prediction time = 0.05406546592712402

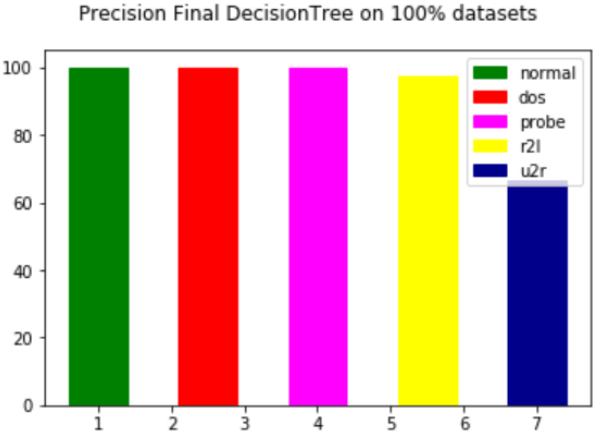


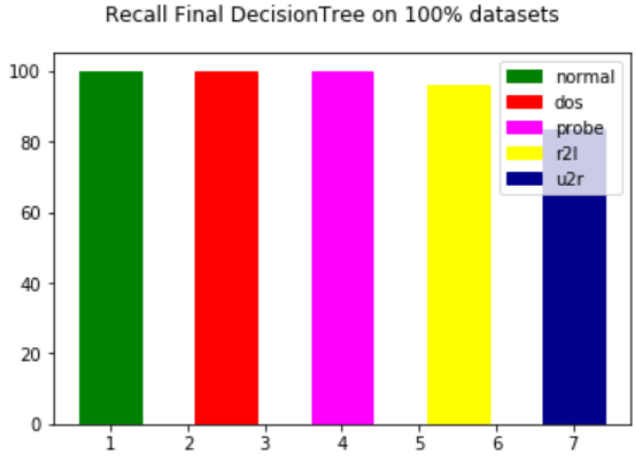


Recall MultinomialNB on 10% datasets

Decision Tree Parameter Tuning (GINI Criteria)







REFERENCES:

[1] Autonomous Agents for Intrusion Detection, http://www.cerias. purdue.edu/research/aafid/, 2010.

[2] CRF++: Yet Another CRF Toolkit, http://crfpp.sourceforge.net/, 2010.

[3] KDD Cup 1999 Intrusion Detection Data, http://kdd.ics.uci.edu/ databases/kddcup99/kddcup99.html, 2010.

[4] Overview of Attack Trends, http://www.cert.org/archive/pdf/ attack_trends.pdf, 2002.

[5] Probabilistic Agent Based Intrusion Detection, http://www.cse.sc. edu/research/isl/agentIDS.shtml, 2010.

[6] SANS Institute—Intrusion Detection FAQ, http://www.sans.org/ resources/idfaq/, 2010.

[7] T. Abraham, IDDM: Intrusion Detection Using Data Mining Techniques, http://www.dsto.defence./gov.au/publications/ 2345/DSTO-GD-0286.pdf, 2008.

[8] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. ACM SIGMOD, vol. 22, no. 2, pp. 207-216, 1993.

[9] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs. Decision Trees in Intrusion Detection Systems," Proc. ACM Symp. Applied Computing (SAC '04), pp. 420-424, 2004.

[10] J.P. Anderson, Computer Security Threat Monitoring and Surveillance, http://csrc.nist.gov/publications/history/ande80.pdf, 2010.

[11]. R.Coolen, "Intrusion Detection: Generics and State of the Art", RTO Technical Report 49, http://www.tno.nl/instit/fel/div2/resources/rto-tr-049-ids.pdf

[12]. J. P. Anderson, "Computer Security Threat Monitoring and Surveillance", Technical Report

April 1980, http://csrc.nist.gov/publications/history/ande80.pdf

[13]. Martin Roesch : "Snort Documents", http://www.snort.org/docs/

[14]. Net Optics, Inc. "White Paper: Deploying Network Taps with Intrusion Detection Systems",

http://www.netoptics.com/products/downloads.asp?PageID=150&Section=res

Checklist of items for the Final Dissertation Report

This checklist is to be duly completed, verified and signed by the student.

1.	**Is the final report neatly formatted with all the elements required for a technical Report?**	**Yes** / No
2.	Is the Cover page in proper format as given in Annexure A?	**Yes** / No
3.	Is the Title page (Inner cover page) in proper format?	**Yes** / No
4.	(a) Is the Certificate from the Supervisor in proper format? (b) Has it been signed by the Supervisor?	**Yes** / No **Yes** / No
5.	Is the Abstract included in the report properly written within one page? Have the technical keywords been specified properly?	**Yes** / No **Yes** / No
6.	Is the title of your report appropriate? **The title should be adequately descriptive, precise and must reflect scope of the actual work done.** Uncommon abbreviations / Acronyms should not be used in the title	**Yes** / No
7.	Have you included the List of abbreviations / Acronyms?	**Yes** / No
8.	Does the Report contain a summary of the literature survey?	**Yes** / No
9.	Does the Table of Contents include page numbers? (i). Are the Pages numbered properly? (Ch. 1 should start on Page # 1) (ii). Are the Figures numbered properly? (Figure Numbers and Figure Titles should be at the bottom of the figures) (iii). Are the Tables numbered properly? (Table Numbers and Table Titles should be at the top of the tables) (iv). Are the Captions for the Figures and Tables proper? (v). Are the Appendices numbered properly? Are their titles appropriate	**Yes** / No **Yes** / No **Yes** / No **Yes** / No **Yes** / No
10.	Is the conclusion of the Report based on discussion of the work?	**Yes** / No
11.	Are References or Bibliography given at the end of the Report? Have the References been cited properly inside the text of the Report? Are all the references cited in the body of the report	**Yes** / No **Yes** / No **Yes** / No
12.	Is the report format and content according to the guidelines? The report should not be a mere printout of a Power Point Presentation, or a user manual. Source code of software need not be included in the report.	**Yes** / No

Declaration by Student:

I certify that I have properly verified all the items in this checklist and ensure that the report is in the proper format as specified in the course handout.

Place: Jhansi **Signature of the Student**

Date: 30 Apr 2024 **Name: Manan Dublish**

ID No.: 2022MT12008