AI IN SECURITY
ASSINGMENT 1

NETWORK INTRUSION DETECTION

GROUP MEMBERS

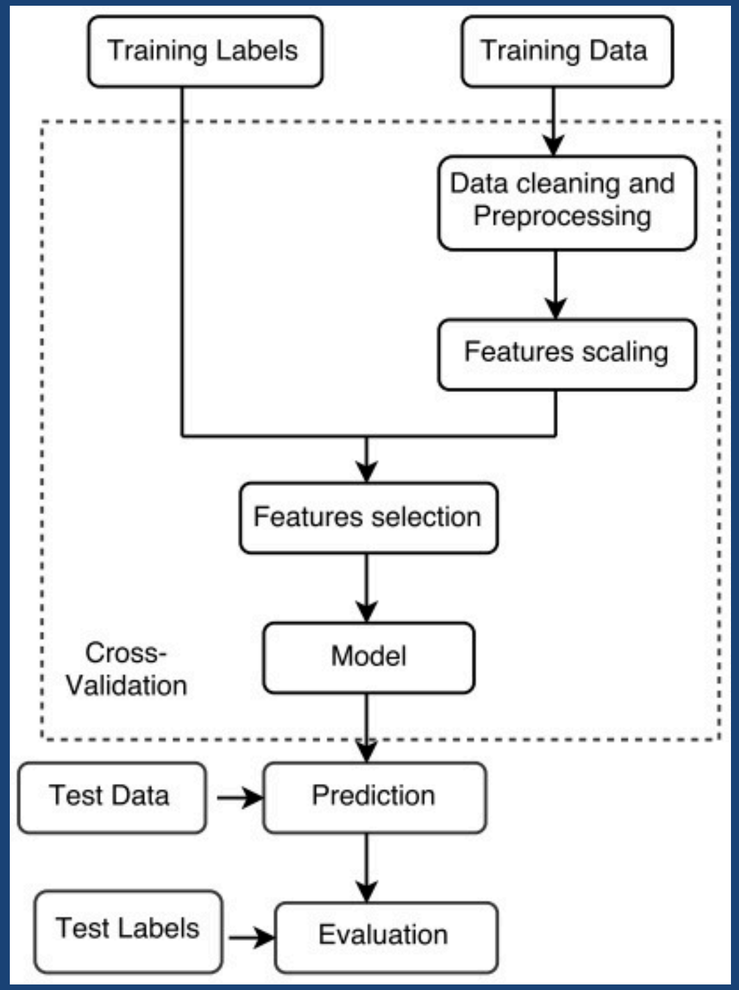
Ahmed Ghazal - 2205004
Ziad Abdelgwad - 2205021
Abdallah Hegazy - 2205228

TABLE OF CONTENT

INTRODUCTION

IDS (Intrusion Detection System) is a network security technology designed to detect unauthorized access or abnormal activities within a network or system. IDS monitors network traffic or system activities for malicious behavior and alerts the system administrators when a potential threat is identified.

INTRODUCTION CONT.



UNDERSTAND DATA

The NSL-KDD dataset is a refined benchmark for evaluating intrusion detection systems (IDS), addressing issues of redundancy and duplication present in its predecessor, the KDD 1999 dataset. It includes network traffic data—both normal and malicious—represented by 41 features divided into four categories:

1. Basic Features: Derived from TCP/IP connections (e.g., protocol type, duration).
2. Time-based Traffic Features: Analyze traffic over two-second intervals.
3. Content Features: Assess packet payloads for deeper insights.
4. Host-based Traffic Features: Focus on long-duration attacks.

UNDERSTAND DATA

Instances are classified into five categories:

1.- Normal traffic.
2.- Denial of Service (DoS).
3.- Probe.
4.- Remote to Local (R2L)
5.- User to Root (U2R).

DoS	back (956), land(18), neptune(41,214), pod(201), smurf(2,646), teardrop(892)	45,927
Probe	satan(3,633), ipsweep(3,599), nmap(1,493), portsweep (2,931)	11,656
R2L	guess_passwd(53), ftp_write(8), imap(658), phf(4), multihop(7), warezmaster(20), warezclient(890), spy(2)	1,642
U2R	buffer_overflow(30), loadmodule(9), rootkit(10), perl(3)	52
Grand Total	59,277	

CHALLENGES

- High Dimensionality
- Handling Large Dataset

PREPROCESSING

- Data Cleaning and Processing
  - Remove Duplicate and missing values.
- Preprocessing
  - Add labels to dataset.
  - Transform Non-Numerical Data --> Numerical Data.
  - Divide dataset into four categories depend on attack types.
  - Data Scaling.
- Feature Selection
  - Eliminate the redundant and irrelevant data.

```
[ ]  df_categorical_values_enc=df_categorical_values.apply(LabelEncoder().fit_transform)
     print(df_categorical_values_enc.head())
     # test set
     testdf_categorical_values_enc=testdf_categorical_values.apply(LabelEncoder().fit_transform)
```

```
   protocol_type  service  flag
0              1       20     9
1              2       44     9
2              1       49     5
3              1       24     9
4              1       24     9
```

DETECTION APPROACH

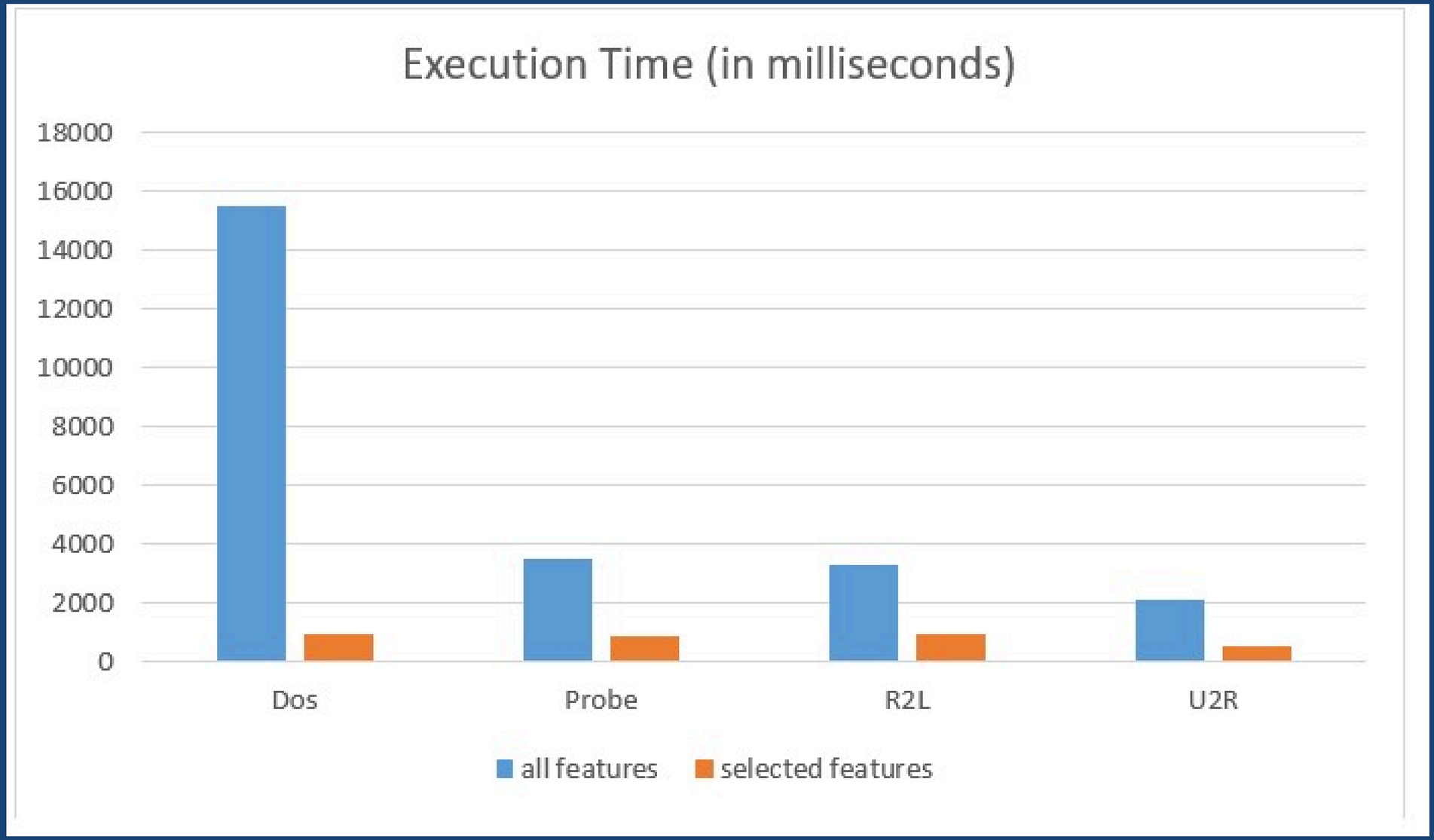
A decision tree model for network intrusion detection partitions the data based on information gain until instances in each leaf node represent uniform classifications (e.g., normal or anomaly traffic). The tree consists of decision nodes and terminal leaves. At each decision node, a test is applied to the network traffic data, with discrete outcomes directing traffic to different branches. Starting from the root, the algorithm recursively evaluates nodes until reaching a leaf, where the output corresponds to the classification (e.g., normal or intrusive) or a numeric value representing the severity of the intrusion. Each leaf node defines a region of network traffic that shares similar patterns, either benign or malicious.

RESULT EVALUTION (BEFORE FEAUTRE SELECTION)

Accuracy	Precision	Recall	F-Score	#Feautres	Class
99.66	99.505	99.71	99.61	41	Dos
99.57	99.04	98.84	98.94	41	Probe
97.03	95.83	95.59	95.71	41	R2L
99.64	99.66	99.61	99.65	41	U2R

RESULT EVALUTION (AFTER FEAUTRE SELECTION)

Accuracy	Precision	Recall	F-Score	#Feautres	Class
99.90	99.69	99.79	99.74	12	Dos
99.80	99.37	99.37	99.37	15	Probe
99.88	97.40	97.41	97.40	13	R2L
99.95	99.70	99.69	99.70	11	U2R


Execution Time (in milliseconds)
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Dos
Probe
R2L
U2R
all features
selected features

THANK YOU