MAHATMA GANDHI INSTITUTE OF TECHNOLOGY(A)
DEPARTMENT OF INFORMATION TECHNOLOGY

An Industry Oriented Mini Project (IT653PC)

On

HYBRID APPROACH FOR NETWORK INTRUSION DETECTION SYSTEM USING MACHINE LEARNING

BY

Patloori Durga-22261A1246

Suragani Revathi-22261A1253

Batch ID-IT-25-12

Internal Supervisor

Mrs.U.Chaitanya

IOMP Supervisor

Mrs.Ch.Sudha

OUTLINE

- ABSTRACT
- INTRODUCTION
- EXISTING SYSTEM
- PROPOSED SYSTEM
- APPLICATIONS
- REQUIREMENTS
- LITERATURE SURVEY
- PROBLEM STATEMENT
- OBJECTIVES
- MODULES DESCRIPTION
- ALGORITHM
- DESIGN ARCHITECTURE
- UML DIAGRAMS
- TEST CASES
- RESULTS
- CONCLUSION
- FUTURE SCOPE
- REFERENCES

ABSTRACT

- IDS is essential for detecting malicious activities and securing networks.
- Signature-based IDS effectively detects known threats but struggles with anomalies and novel attacks.
- Machine learning classifiers improve detection but have difficulty identifying rare attacks like R2L and U2R due to their similarity to normal behavior.
- The Double Layered Hybrid Approach (DLHA) integrates enhanced BLSTM to improve attack detection.
- PCA extracts key features, while BLSTM enhances temporal and sequential pattern recognition for rare attack detection.
- Layer 1 uses Naive Bayes for detecting DoS and Probe attacks; Layer 2 combines SVM and BLSTM to differentiate R2L and U2R from normal instances.

INTRODUCTION

- Intrusion Detection Systems (IDS) help secure networks by detecting and mitigating cyber threats.
- Signature-based IDS fails to detect new and anomalous threats.
- Machine learning improves detection but struggles with rare attacks like R2L and U2R due to imbalanced data.
- The Double Layered Hybrid Approach (DLHA) integrates BLSTM for better anomaly detection.
- Naïve Bayes classifies DoS and Probe attacks, while SVM and BLSTM differentiate R2L and U2R attacks from normal traffic.
- BLSTM enhances the learning of attack patterns, making IDS more effective against evolving cyber threats.

IDENTIFYING ATTACKS

DoS Attacks (Denial of Service)-

Back,land,neptune,pod,smurf,teardrop,apache2,udp storm,processtable,worm.

Probe Attacks (Probing and Scanning)-

Satan,ipsweep,nmap,portsweep,mscan,saint.

R2L (Remote-to-Local)-

ftp_write, guess_passwd, imap, multihop, etc.

U2R (User-to-Root)-

buffer_overflow, rootkit, loadmodule, perl, etc.

EXISTING SYSTEM:

Limited Detection of Evolving Threats

- Signature-based IDS fails to detect new and evolving attacks, while anomaly-based IDS has high false positives.
- ML models struggle with imbalanced datasets, leading to poor detection of rare attacks like R2L and U2R. Limitation: Ineffective against zero-day attacks and misclassifies rare threats.

High Computational Cost & Real-Time Inefficiency

- Deep learning models (CNN, LSTM, Hybrid Approaches) improve accuracy but require high processing power.
- Manual feature selection reduces adaptability, making real-time intrusion detection difficult. Limitation: Slow processing, scalability challenges, and high resource consumption.

PROPOSED SYSTEM

Optimized Feature Selection & Real-Time Efficiency

- Principal Component Analysis (PCA) extracts key features, improving detection speed and reducing noise.
- The system balances accuracy and computational efficiency, making it scalable for real-time intrusion detection. Benefit: Faster processing, adaptability to new threats, and efficient real-time performance.

Improved Rare Attack Detection with Hybrid Approach

- A two-layer classification model:
- Layer 1: Naive Bayes efficiently detects DoS and Probe attacks.
- Layer 2: SVM + BLSTM accurately identifies rare R2L and U2R attacks by learning sequential attack patterns. Benefit: Higher accuracy in detecting both common and rare attacks with reduced false positives.

APPLICATIONS

- Enterprise Network Security
- Cloud Security & Data Centers
- Government & Defense Networks
- Banking & Financial Institutions
- Healthcare Systems & IoT Security
- Smart Cities & Critical Infrastructure
- Educational Institutions & Research Labs

REQUIREMENTS

SOFTWARE:

Operating System:Windows 10/11

Programming Language:Python 3.x

Libraries and Frameworks:TensorFlow / PyTorch (for BLSTM),Scikit-learn (for PCA, SVM, and Naïve Bayes),NumPy & Pandas (for data handling),Matplotlib & Seaborn (for visualization)

Development Environment:VsCode

Additional Tools:NSL-KDD (data set)

HARDWARE:

- **Processor** - Intel Core i5/i7/i9
- **RAM** - Minimum: 8GB
- **Storage** - Minimum: 256GB SSD

LITERATURE SURVEY

S.NO	AUTHOR NAMES,YEAR OF PUBLICATION	JOURNAL OR CONFERENCE NAME AND PUBLISHER NAME	METHODOLOGY/ ALGORITHM / TECHNIQUES USED	MERITS	DEMERITS	RESEARCH GAPS
1.	Muhammad Sajid et al. (2024)	Journal of Cloud Computing, Springer	Developed a Hybrid Machine and Deep Learning Approach integrating CNN, LSTM, and ensemble learning	Enhanced anomaly detection with a focus on cloud based environments	CNN struggles with sequential dependencies in network traffic	Requires improvement in feature selection techniques for better performance
2.	Security and Communication Networks (2024)	Security and Communication Networks, Hindawi	Designed a Supervised Ensemble Stacking Model combining multiple ML algorithms for IDS	High detection accuracy across multiple attack type	Overfitting issue due to ensemble complexity	Needs validation on real - world datasets for better generalization

3.	S. Shi, D. Han, & M. Cui (2023)	Connection Science, Taylor & Francis	Introduced a Multimodal Hybrid Parallel IDS using ML and DL models for feature extraction and classification	improved performance in handling high dimensional network traffic data	High computational cost	Needs lightweight IDS solutions for real time deployment
4.	R. Jalili, S. Imani, & M. R. Aminzadeh (2021)	Proceedings of ACM SIN Conference, ACM Digital Library	Proposed a CNN LSTM-based approach for anomaly detection in Software Defined Networks (SDNs)	Effective in detecting unknown attacks and reducing false positives	High training time and resource consumption	Needs optimization techniques to improve efficiency
5.	M. Mohammadi, A. Navaras, & M. H. Amini (2021)	IEEE Xplore, IEEE	Proposed a Double Layered Hybrid Approach (DLHA) using Naïve Bayes, SVM for network intrusion detection	Effective in detecting DoS and Probe attacks with high accuracy	Struggles with class imbalance in R2L and U2R attacks	Requires further optimization for real time attack detection

PROBLEM STATEMENT

- Traditional IDS fail to detect rare and evolving attacks like R2L and U2R, resulting in high false negatives due to limited adaptability and reliance on signature-based methods.

- Existing machine learning approaches face challenges with data imbalance, while deep learning models are resource-intensive—highlighting the need for a hybrid, efficient, and accurate IDS for real-time cybersecurity.

OBJECTIVES

- Enhance Intrusion Detection Accuracy
- Optimize Feature Selection
- Reduce False Positives & Improve Adaptability

MODULES DESCRIPTION

Data Preprocessing & Feature Selection

- Uses PCA to extract relevant features and reduce noise.

Layer 1: Naive Bayes Classification

- Detects DoS and Probe attacks as an initial filtering step.

Layer 2: SVM + BLSTM for Rare Attack Detection

- SVM separates normal traffic from anomalies.
- BLSTM captures sequential attack patterns, improving R2L and U2R detection.

Intrusion Detection & Classification

- Labels traffic as normal or attack, minimizing false positives.

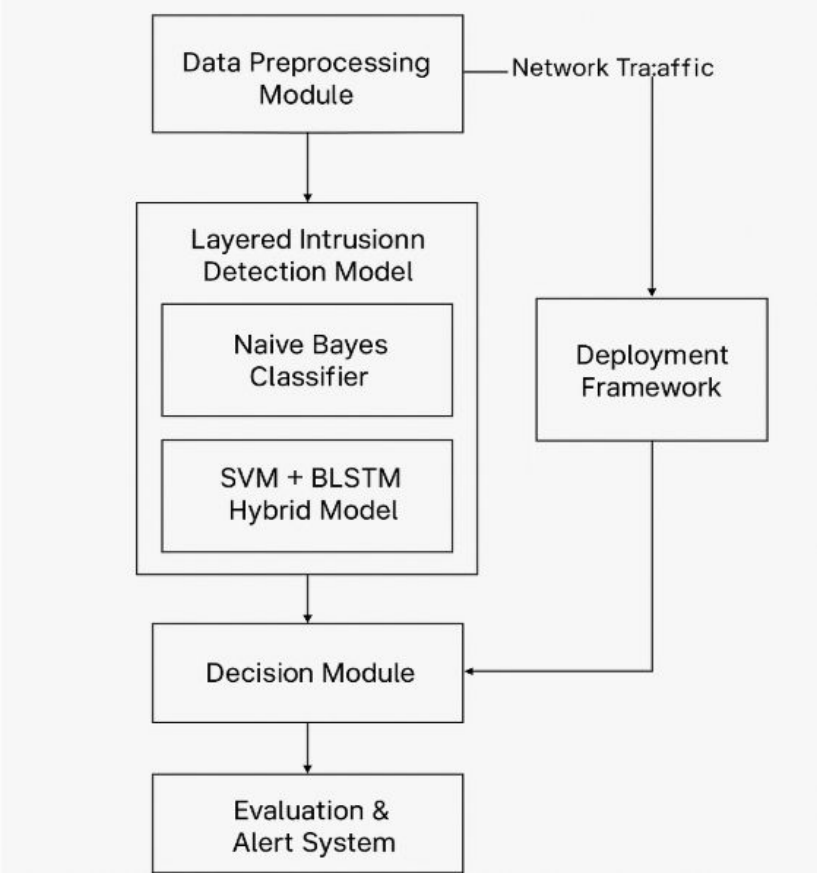
Performance Evaluation & Real-Time Implementation

- Measures accuracy, precision, recall, and F1-score for real-time security.

ALGORITHM

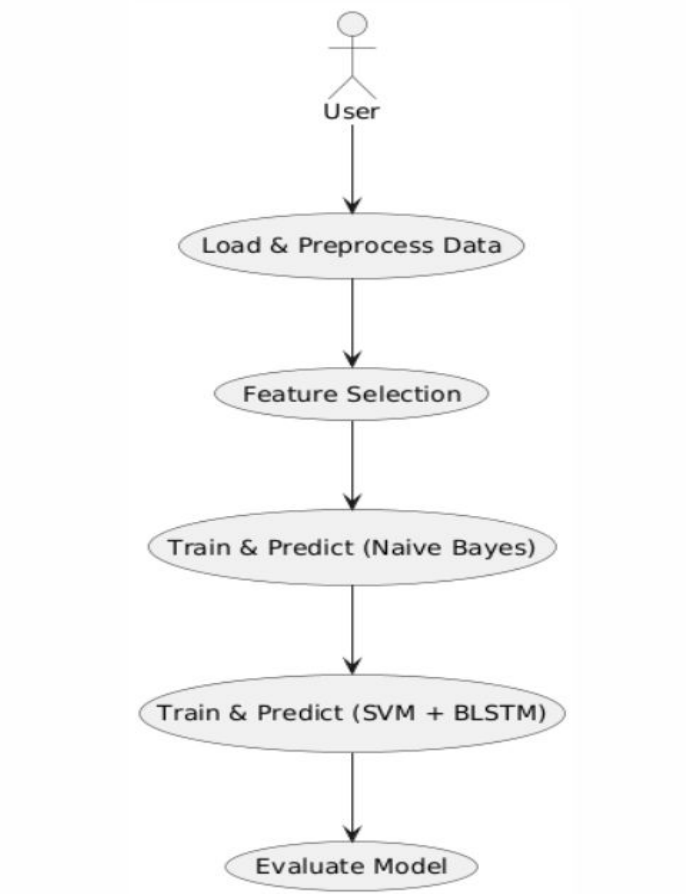
1. Data Preprocessing & Feature Selection
2. Layer 1 - Initial Classification (Naïve Bayes)
3. Layer 2 - Rare Attack Detection (SVM + BLSTM)
4. Intrusion Detection & Classification
5. Performance Evaluation & Real-Time Implementation

DESIGN ARCHITECTURE

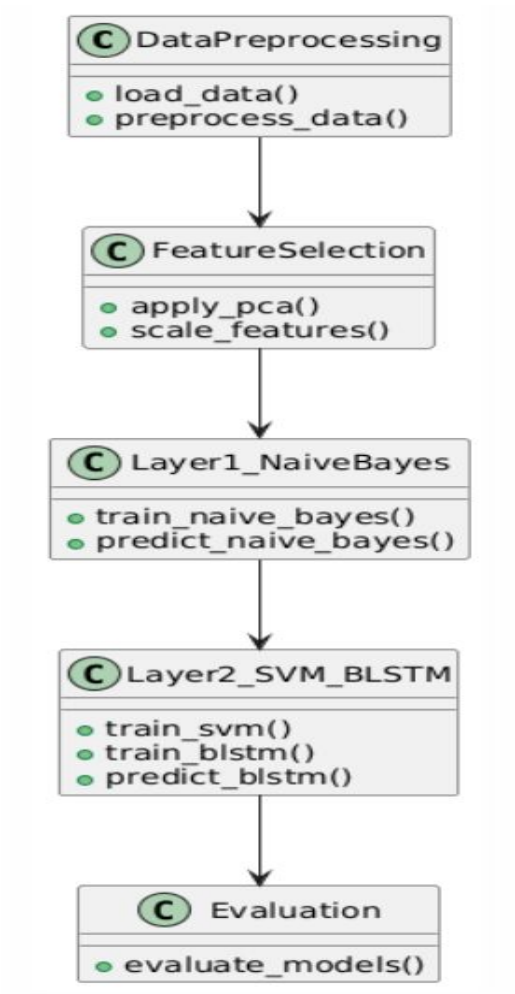


UML DIAGRAMS

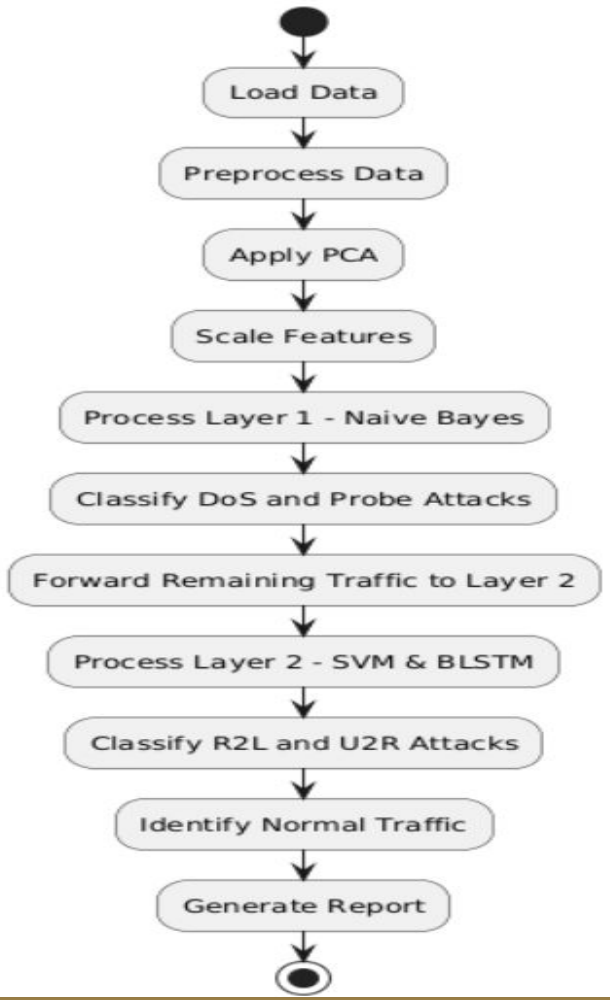
USE CASE DIAGRAM:



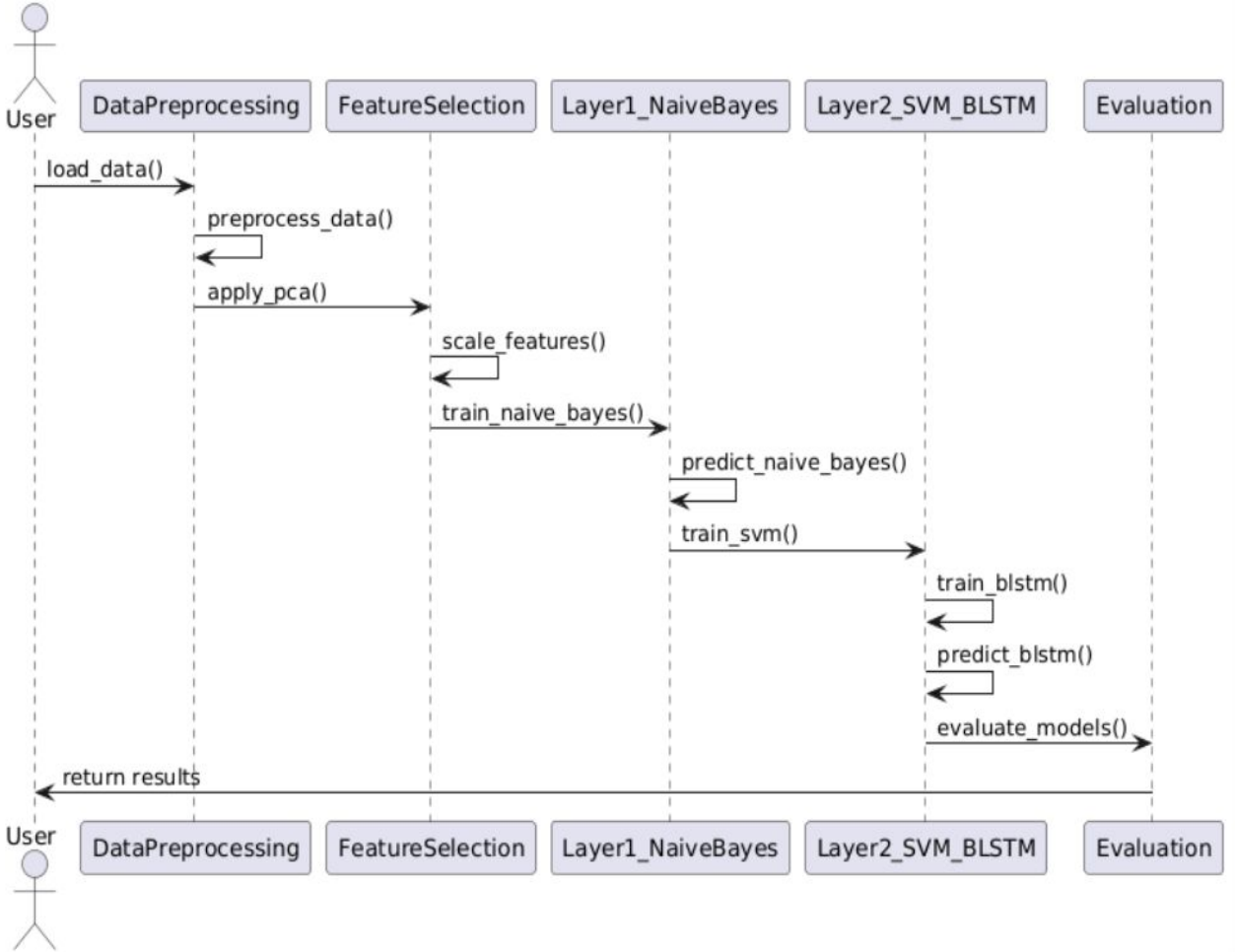
CLASS DIAGRAM:



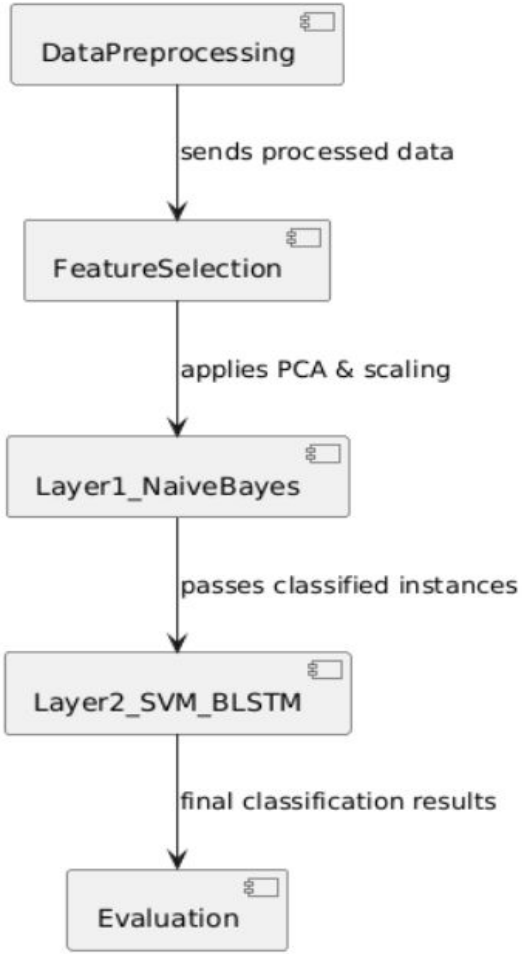
ACTIVITY DIAGRAM:



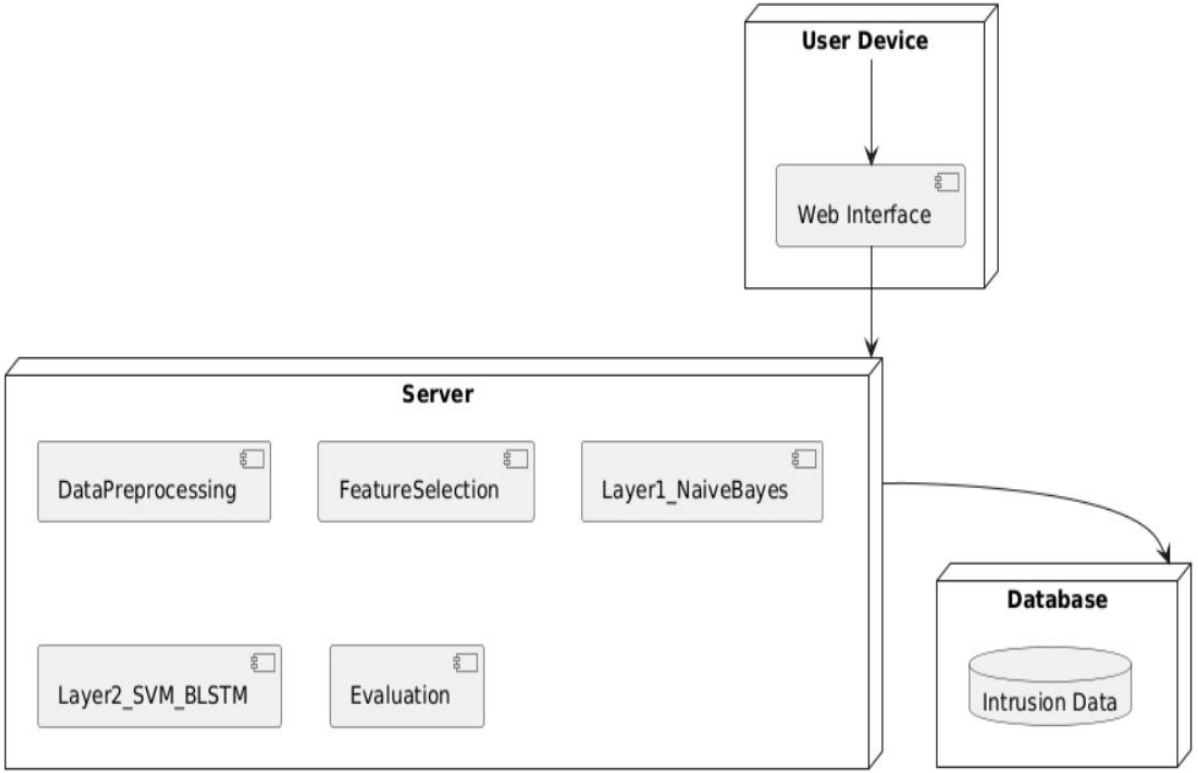
SEQUENCE DIAGRAM:



COMPONENT DIAGRAM:



DEPLOYMENT DIAGRAM:



TEST CASES:

Test Case ID	Model	Data Source	What's Tested	Pass Criteria
TC1	Naive Bayes (Layer 1)	30% hold-out split of X_pca/y from training set (X_test1, y_test1)	Layer 1 classification performance	Model returns predictions
TC2	SVM (Layer 2)	30% hold-out split of the *correctly* classified subset (X_test2, y_test2)	Layer 2 (SVM) classification on "easy" cases	No errors in prediction pipeline
TC3	BLSTM (Layer 2)	30% hold-out split of the same "easy" subset, used as validation during training	BLSTM training & validation accuracy	Loss decreases, accuracy improves per epoch

TC4	Naive Bayes (Test Data)	Entire external test file (KDDTest+.txt) preprocessed → X_test_all, y_test_all	Final NB performance on unseen data	Passes and gives confusion matrix
TC5	SVM (Test Data)	Same external test set (X_test_all, y_test_all)	Final SVM performance on unseen data	Passes and gives confusion matrix
TC6	BLSTM (Test Data)	Same external test set reshaped for BLSTM	Final BLSTM performance on unseen data	Passes and gives confusion matrix

RESULTS:

```
PROBLEMS 11  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                                    powershell

(myenv) PS C:\Users\cyber\OneDrive\Desktop\IOMP1> python p2.py
2025-06-12 23:27:20.460367: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-poin
t round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-06-12 23:27:21.723428: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-poin
t round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
   0  tcp ftp_data   SF  491   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  ...  0.00.5  150   25  0.17  0.03  0.17.1  0.00.6  0.00.7  0.00.8  0.05  0.00.9   normal  20
0  0  udp    other   SF  146    0    0    0    0    0    0    0    0  ...   0.000  255    1 0.000 0.600   0.880   0.000   0.000   0.000 0.000   0.000   normal  15
1  0  tcp  private   S0    0    0    0    0    0    0    0    0  ...   0.000  255   26 0.100 0.050   0.000   0.000   1.000   1.000 0.000   0.000  neptune  19
2  0  tcp     http   SF  232  8153    0    0    0    0    1    0  ...   0.000   30  255 1.000 0.000   0.030   0.040   0.030   0.010 0.000   0.010   normal  21
3  0  tcp     http   SF  199   420    0    0    0    0    1    0  ...   0.090  255  255 1.000 0.000   0.000   0.000   0.000   0.000 0.000   0.000   normal  21
4  0  tcp  private  REJ    0    0    0    0    0    0    0    0    0  ...   0.000  255   19 0.070 0.070   0.000   0.000   0.000   0.000 1.000   1.000  neptune  21

[5 rows x 43 columns]
   duration protocol_type  service flag  src_bytes  ...  dst_host_srv_serror_rate  dst_host_rerror_rate  dst_host_srv_rerror_rate   attack  level
0         0           udp    other   SF        146  ...                     0.000                 0.000                     0.000   normal     15
1         0           tcp  private   S0          0  ...                     1.000                 0.000                     0.000  neptune     19
2         0           tcp     http   SF        232  ...                     0.010                 0.000                     0.010   normal     21
3         0           tcp     http   SF        199  ...                     0.000                 0.000                     0.000   normal     21
4         0           tcp  private  REJ          0  ...                     0.000                 1.000                     1.000  neptune     21

[5 rows x 43 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 125972 entries, 0 to 125971
Data columns (total 43 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   duration                     125972 non-null  int64
 1   protocol_type                125972 non-null  object
 2   service                      125972 non-null  object
 3   flag                         125972 non-null  object
 4   src_bytes                    125972 non-null  int64
 5   dst_bytes                    125972 non-null  int64
 6   land                         125972 non-null  int64
 7   wrong_fragment               125972 non-null  int64
 8   urgent                       125972 non-null  int64
```

```
PROBLEMS 11  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

 8   urgent                       125972 non-null  int64
 9   hot                          125972 non-null  int64
 10  num_failed_logins            125972 non-null  int64
 11  logged_in                    125972 non-null  int64
 12  num_compromised              125972 non-null  int64
 13  root_shell                   125972 non-null  int64
 14  su_attempted                 125972 non-null  int64
 15  num_root                     125972 non-null  int64
 16  num_file_creations           125972 non-null  int64
 17  num_shells                   125972 non-null  int64
 18  num_access_files             125972 non-null  int64
 19  num_outbound_cmds            125972 non-null  int64
 20  is_host_login                125972 non-null  int64
 21  is_guest_login               125972 non-null  int64
 22  count                        125972 non-null  int64
 23  srv_count                    125972 non-null  int64
 24  serror_rate                  125972 non-null  float64
 25  srv_serror_rate              125972 non-null  float64
 26  rerror_rate                  125972 non-null  float64
 27  srv_rerror_rate              125972 non-null  float64
 28  same_srv_rate                125972 non-null  float64
 29  diff_srv_rate                125972 non-null  float64
 30  srv_diff_host_rate           125972 non-null  float64
 31  dst_host_count               125972 non-null  int64
 32  dst_host_srv_count           125972 non-null  int64
 33  dst_host_same_srv_rate       125972 non-null  float64
 34  dst_host_diff_srv_rate       125972 non-null  float64
 35  dst_host_same_src_port_rate  125972 non-null  float64
 36  dst_host_srv_diff_host_rate  125972 non-null  float64
 37  dst_host_serror_rate         125972 non-null  float64
 38  dst_host_srv_serror_rate     125972 non-null  float64
 39  dst_host_rerror_rate         125972 non-null  float64
 40  dst_host_srv_rerror_rate     125972 non-null  float64
 41  attack                       125972 non-null  object
 42  level                        125972 non-null  int64
```

```
 42  level                    125972 non-null  int64
dtypes: float64(15), int64(24), object(4)
memory usage: 41.3+ MB
None
```

	count	mean	std	min	25%	50%	75%	max
duration	125972.000	287.147	2604.526	0.000	0.000	0.000	0.000	42908.000
src_bytes	125972.000	45567.101	5870354.481	0.000	0.000	44.000	276.000	1379963888.000
dst_bytes	125972.000	19779.271	4021285.112	0.000	0.000	0.000	516.000	1309937401.000
land	125972.000	0.000	0.014	0.000	0.000	0.000	0.000	1.000
wrong_fragment	125972.000	0.023	0.254	0.000	0.000	0.000	0.000	3.000
urgent	125972.000	0.000	0.014	0.000	0.000	0.000	0.000	3.000
hot	125972.000	0.204	2.150	0.000	0.000	0.000	0.000	77.000
num_failed_logins	125972.000	0.001	0.045	0.000	0.000	0.000	0.000	5.000
logged_in	125972.000	0.396	0.489	0.000	0.000	0.000	1.000	1.000
num_compromised	125972.000	0.279	23.942	0.000	0.000	0.000	0.000	7479.000
root_shell	125972.000	0.001	0.037	0.000	0.000	0.000	0.000	1.000
su_attempted	125972.000	0.001	0.045	0.000	0.000	0.000	0.000	2.000
num_root	125972.000	0.302	24.400	0.000	0.000	0.000	0.000	7468.000
num_file_creations	125972.000	0.013	0.484	0.000	0.000	0.000	0.000	43.000
num_shells	125972.000	0.000	0.022	0.000	0.000	0.000	0.000	2.000
num_access_files	125972.000	0.004	0.099	0.000	0.000	0.000	0.000	9.000
num_outbound_cmds	125972.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
is_host_login	125972.000	0.000	0.003	0.000	0.000	0.000	0.000	1.000
is_guest_login	125972.000	0.009	0.097	0.000	0.000	0.000	0.000	1.000
count	125972.000	84.108	114.509	0.000	2.000	14.000	143.000	511.000
srv_count	125972.000	27.738	72.636	0.000	2.000	8.000	18.000	511.000
serror_rate	125972.000	0.284	0.446	0.000	0.000	0.000	1.000	1.000
srv_serror_rate	125972.000	0.282	0.447	0.000	0.000	0.000	1.000	1.000
rerror_rate	125972.000	0.120	0.320	0.000	0.000	0.000	0.000	1.000
srv_rerror_rate	125972.000	0.121	0.324	0.000	0.000	0.000	0.000	1.000
same_srv_rate	125972.000	0.661	0.440	0.000	0.090	1.000	1.000	1.000
diff_srv_rate	125972.000	0.063	0.180	0.000	0.000	0.000	0.060	1.000
srv_diff_host_rate	125972.000	0.097	0.260	0.000	0.000	0.000	0.000	1.000
dst_host_count	125972.000	182.149	99.207	0.000	82.000	255.000	255.000	255.000
dst_host_srv_count	125972.000	115.654	110.703	0.000	10.000	63.000	255.000	255.000
dst_host_same_srv_rate	125972.000	0.521	0.449	0.000	0.050	0.510	1.000	1.000
dst_host_diff_srv_rate	125972.000	0.083	0.189	0.000	0.000	0.020	0.070	1.000
dst_host_same_src_port_rate	125972.000	0.148	0.309	0.000	0.000	0.000	0.060	1.000
dst_host_srv_diff_host_rate	125972.000	0.033	0.113	0.000	0.000	0.000	0.020	1.000
dst_host_serror_rate	125972.000	0.284	0.445	0.000	0.000	0.000	1.000	1.000
dst_host_srv_serror_rate	125972.000	0.278	0.446	0.000	0.000	0.000	1.000	1.000
dst_host_rerror_rate	125972.000	0.119	0.307	0.000	0.000	0.000	0.000	1.000
dst_host_srv_rerror_rate	125972.000	0.120	0.319	0.000	0.000	0.000	0.000	1.000
level	125972.000	19.504	2.292	0.000	18.000	20.000	21.000	21.000

duration	0
protocol_type	0
service	0
flag	0
src_bytes	0
dst_bytes	0
land	0
wrong_fragment	0
urgent	0
hot	0
num_failed_logins	0
logged_in	0
num_compromised	0
root_shell	0
su_attempted	0
num_root	0
num_file_creations	0
num_shells	0
num_access_files	0
num_outbound_cmds	0
is_host_login	0
is_guest_login	0
count	0
srv_count	0
serror_rate	0

PROBLEMS 11 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
dst_host_same_srv_rate         0
dst_host_diff_srv_rate         0
dst_host_same_src_port_rate    0
dst_host_srv_diff_host_rate    0
dst_host_serror_rate           0
dst_host_srv_serror_rate       0
dst_host_rerror_rate           0
dst_host_srv_rerror_rate       0
attack                         0
level                          0
dtype: int64
Column: protocol_type
------------------------------
Unique Values (3): ['udp' 'tcp' 'icmp']

Value Counts:
protocol_type
tcp     102688
udp      14993
icmp      8291
Name: count, dtype: int64
====================================

Column: service
------------------------------
Unique Values (70): ['other' 'private' 'http' 'remote_job' 'ftp_data' 'name' 'netbios_ns'
 'eco_i' 'mtp' 'telnet' 'finger' 'domain_u' 'supdup' 'uucp_path' 'Z39_50'
 'smtp' 'csnet_ns' 'uucp' 'netbios_dgm' 'urp_i' 'auth' 'domain' 'ftp'
 'bgp' 'ldap' 'ecr_i' 'gopher' 'vmnet' 'systat' 'http_443' 'efs' 'whois'
 'imap4' 'iso_tsap' 'echo' 'klogin' 'link' 'sunrpc' 'login' 'kshell'
 'sql_net' 'time' 'hostnames' 'exec' 'ntp_u' 'discard' 'nntp' 'courier'
 'ctf' 'ssh' 'daytime' 'shell' 'netstat' 'pop_3' 'nnsp' 'IRC' 'pop_2'
 'printer' 'tim_i' 'pm_dump' 'red_i' 'netbios_ssn' 'rje' 'X11' 'urh_i'
 'http_8001' 'aol' 'http_2784' 'tftp_u' 'harvest']
```

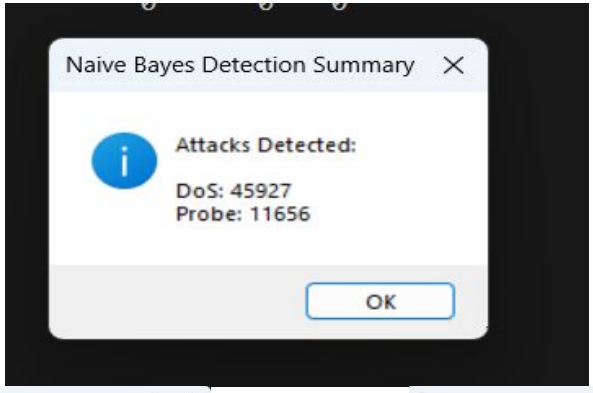
```
Value Counts:
service
http         40338
private      21853
domain_u      9043
smtp          7313
ftp_data      6859
              ...
tftp_u           3
http_8001        2
aol              2
harvest          2
http_2784        1
Name: count, Length: 70, dtype: int64
====================================

Column: flag
------------------------------
Unique Values (11): ['SF' 'S0' 'REJ' 'RSTR' 'SH' 'RSTO' 'S1' 'RSTOS0' 'S3' 'S2' 'OTH']

Value Counts:
flag
SF        74944
S0        34851
REJ       11233
RSTR       2421
RSTO       1562
S1          365
SH          271
S2          127
RSTOS0      103
S3           49
OTH          46
Name: count, dtype: int64
```

```
Column: attack
------------------------------
Unique Values (23): ['normal' 'neptune' 'warezclient' 'ipsweep' 'portsweep' 'teardrop' 'nmap'
 'satan' 'smurf' 'pod' 'back' 'guess_passwd' 'ftp_write' 'multihop'
 'rootkit' 'buffer_overflow' 'imap' 'warezmaster' 'phf' 'land'
 'loadmodule' 'spy' 'perl']

Value Counts:
attack
normal             67342
neptune            41214
satan               3633
ipsweep             3599
portsweep           2931
smurf               2646
nmap                1493
back                 956
teardrop             892
warezclient          890
pod                  201
guess_passwd          53
buffer_overflow       30
warezmaster           20
land                  18
imap                  11
rootkit               10
loadmodule             9
ftp_write              8
multihop               7
phf                    4
perl                   3
spy                    2
Name: count, dtype: int64
```

```
🔹 Features (X):
   duration  src_bytes  dst_bytes  land  wrong_fragment  urgent  hot  num_failed_logins  ...  flag_RSTOS0  flag_RSTR  flag_S0  flag_S1  flag_S2  flag_S3  flag_SF  fla
g_SH
0         0        146          0     0               0       0    0                  0  ...        False      False    False    False    False    False     True    F
alse
1         0          0          0     0               0       0    0                  0  ...        False      False     True    False    False    False    False    F
alse
2         0        232       8153     0               0       0    0                  0  ...        False      False    False    False    False    False     True    F
alse
3         0        199        420     0               0       0    0                  0  ...        False      False    False    False    False    False     True    F
alse
4         0          0          0     0               0       0    0                  0  ...        False      False    False    False    False    False    False    F
alse

[5 rows x 124 columns]

🔹 Encoded Labels (y_encoded):
[1 0 1 1 0 0 0 0 0 0]

🔹 Original Labels (y):
0    normal
1       dos
2    normal
3    normal
4       dos
Name: label, dtype: object

🔹 Label Mappings:
0 -> dos
1 -> normal
2 -> probe
3 -> r2l
4 -> u2r
```

```
Total number of DoS attacks identified: 45927
Total number of Probe attacks identified: 11656
              precision    recall  f1-score   support

         dos       0.95      0.42      0.58     13941
      normal       0.68      0.95      0.79     20014
       probe       0.40      0.32      0.36      3526
         r2l       0.39      0.95      0.55       291
         u2r       0.11      0.40      0.17        20

    accuracy                           0.70     37792
   macro avg       0.51      0.61      0.49     37792
weighted avg       0.75      0.70      0.67     37792

              precision    recall  f1-score   support

         dos       0.99      1.00      1.00      1765
      normal       1.00      1.00      1.00      5707
       probe       1.00      0.97      0.99       350
         r2l       0.94      0.96      0.95        79
         u2r       1.00      1.00      1.00         1

    accuracy                           1.00      7902
   macro avg       0.99      0.99      0.99      7902
weighted avg       1.00      1.00      1.00      7902

2025-06-12 23:27:50.824513: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to us
critical operations.
To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 AVX_VNNI FMA, in other operations, rebuild TensorFlow wi
Epoch 1/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 4s 5ms/step - accuracy: 0.8468 - loss: 1.1221 - val_accuracy: 0.9984 - val_loss: 0.0148
Epoch 2/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.9986 - loss: 0.0109 - val_accuracy: 0.9991 - val_loss: 0.0046
Epoch 3/10
```

```
289/289 ━━━━━━━━━━━━━━━━━━━━ 4s 5ms/step - accuracy: 0.8468 - loss: 1.1221 - val_accuracy: 0.9984 - val_loss: 0.0148
Epoch 2/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.9986 - loss: 0.0109 - val_accuracy: 0.9991 - val_loss: 0.0046
Epoch 3/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.9997 - loss: 0.0030 - val_accuracy: 0.9990 - val_loss: 0.0033
Epoch 4/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.9996 - loss: 0.0017 - val_accuracy: 0.9987 - val_loss: 0.0030
Epoch 5/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.9997 - loss: 9.2889e-04 - val_accuracy: 0.9991 - val_loss: 0.0029
Epoch 6/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.9999 - loss: 0.0011 - val_accuracy: 0.9989 - val_loss: 0.0030
Epoch 7/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.9998 - loss: 7.0879e-04 - val_accuracy: 0.9989 - val_loss: 0.0029
Epoch 8/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.9999 - loss: 4.6672e-04 - val_accuracy: 0.9985 - val_loss: 0.0043
Epoch 9/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.9996 - loss: 6.2580e-04 - val_accuracy: 0.9992 - val_loss: 0.0023
Epoch 10/10
289/289 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.9999 - loss: 3.7507e-04 - val_accuracy: 0.9992 - val_loss: 0.0022
247/247 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step

✅ Total number of R2L attacks identified in dataset: 995
✅ Total number of U2R attacks identified in dataset: 52
              precision    recall  f1-score   support

         dos       1.00      1.00      1.00      1765
      normal       1.00      1.00      1.00      5707
       probe       1.00      1.00      1.00       350
         r2l       0.97      0.96      0.97        79
         u2r       1.00      1.00      1.00         1

    accuracy                           1.00      7902
   macro avg       0.99      0.99      0.99      7902
weighted avg       1.00      1.00      1.00      7902
```


Naive Bayes Detection Summary
Attacks Detected:
DoS: 45927
Probe: 11656
OK

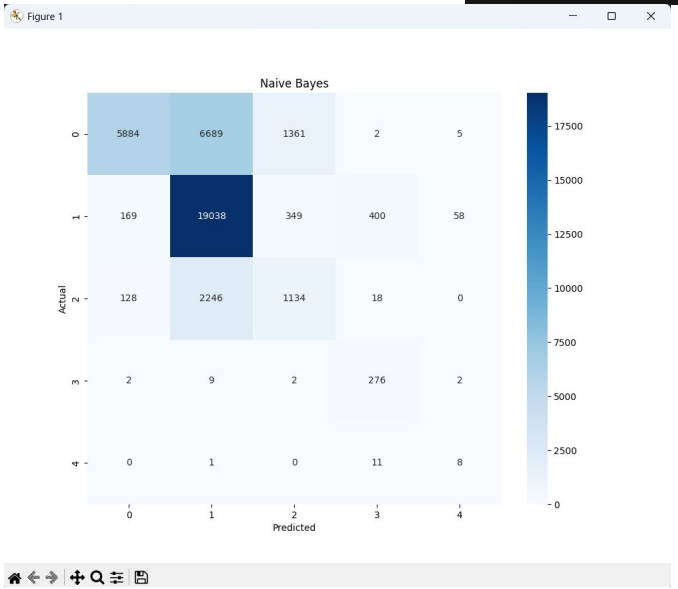


Figure 1
Naive Bayes
Actual
Predicted
0
1
2
3
4
5884
6689
1361
2
5
169
19038
349
400
58
128
2246
1134
18
0
2
9
2
276
2
0
1
0
11
8
17500
15000
12500
10000
7500
5000
2500
0

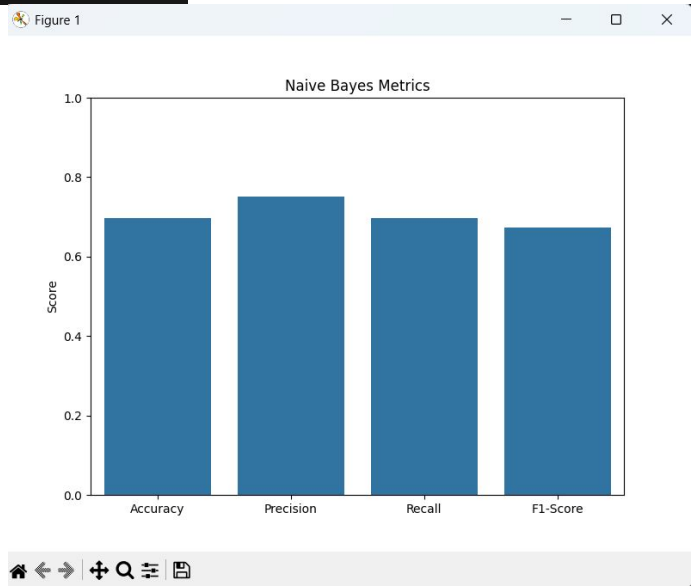


Figure 1
Naive Bayes Metrics
Score
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Precision
Recall
F1-Score

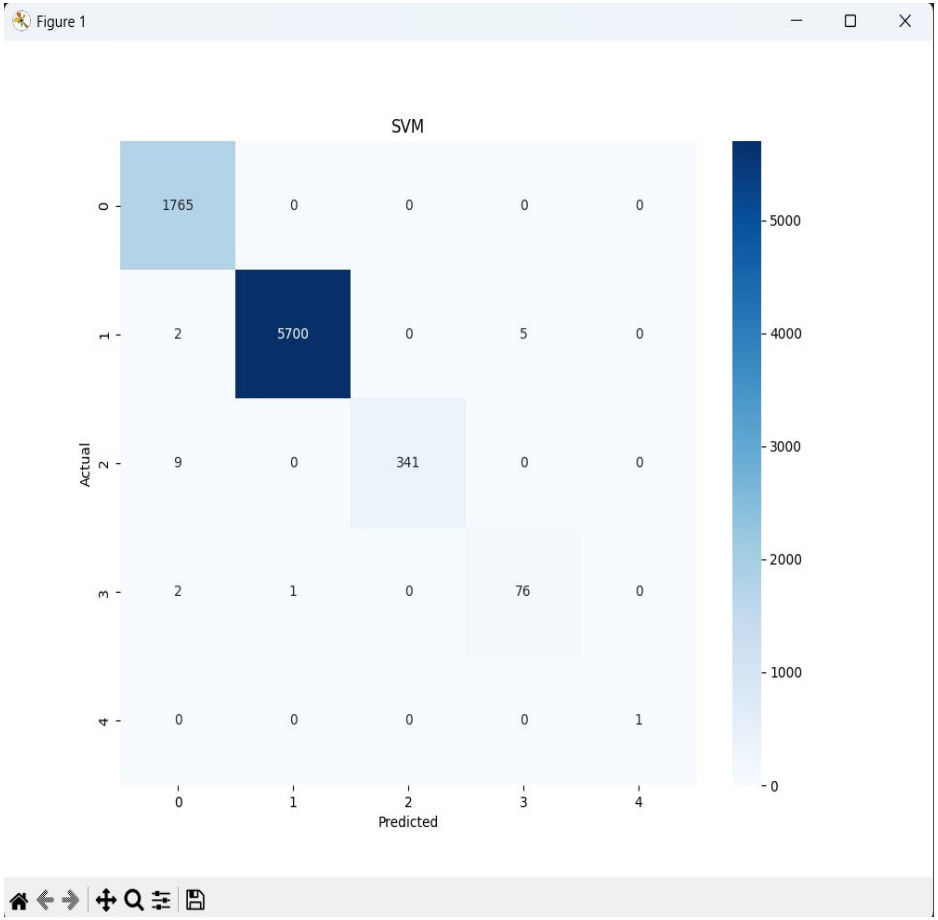


Figure 1
SVM
1765
0
0
0
0
2
5700
0
5
0
9
0
341
0
0
2
1
0
76
0
0
0
0
0
1
Actual
Predicted
0
1
2
3
4
5000
4000
3000
2000
1000
0

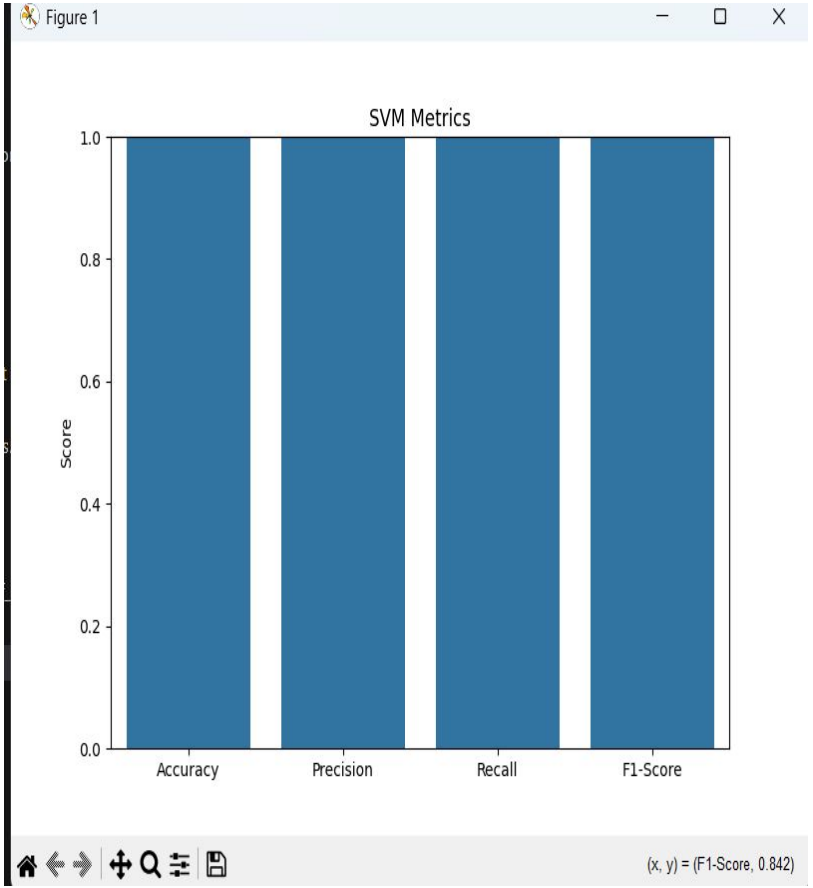
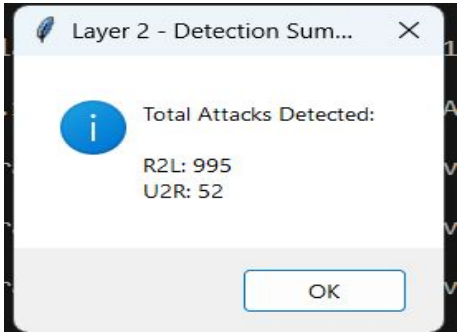


Figure 1
SVM Metrics
Score
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Precision
Recall
F1-Score
(x, y) = (F1-Score, 0.842)


Layer 2 - Detection Sum...
Total Attacks Detected:
R2L: 995
U2R: 52
OK

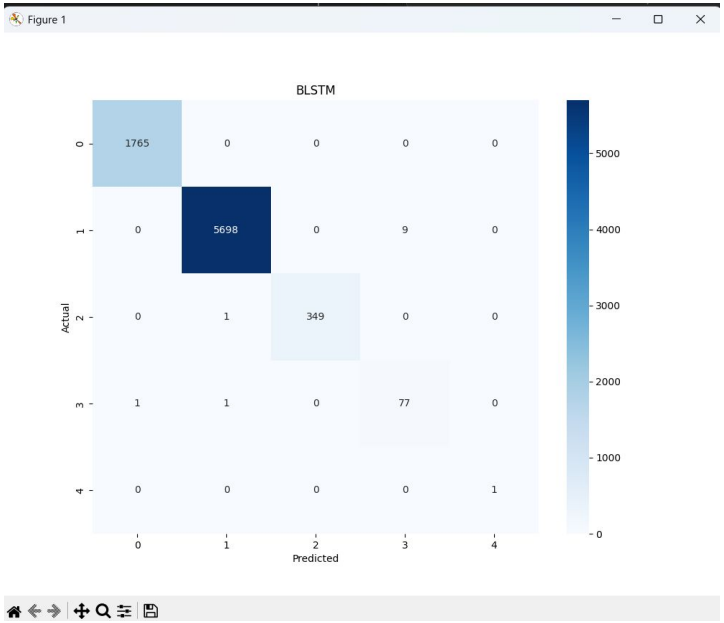


Figure 1
BLSTM
Actual
Predicted
1765
5698
349
77
1

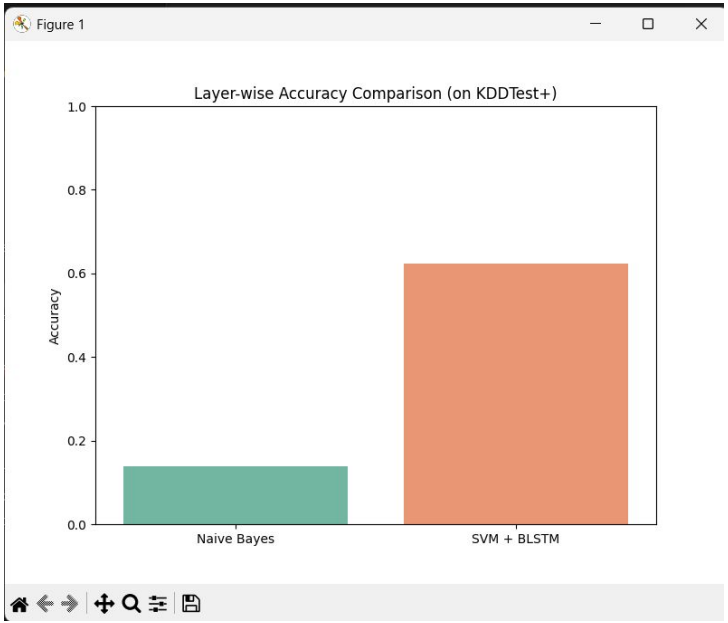


Figure 1
Layer-wise Accuracy Comparison (on KDDTest+)
Accuracy
Naive Bayes
SVM + BLSTM

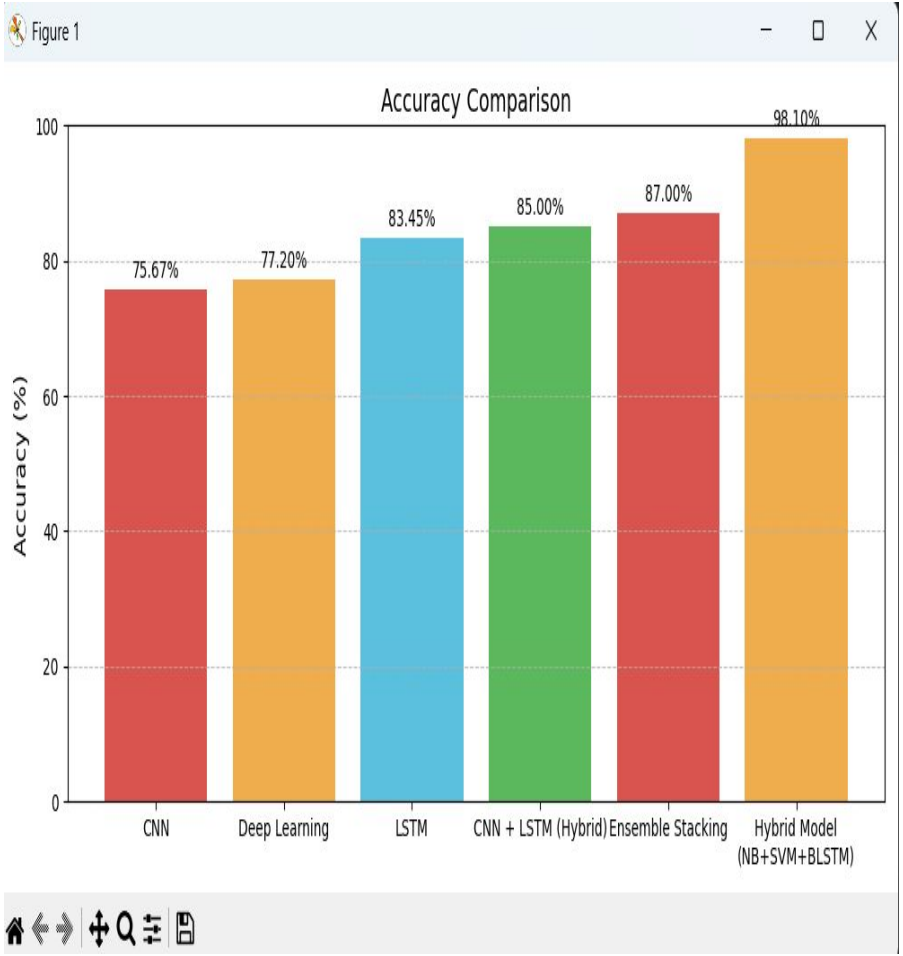


Figure 1
Accuracy Comparison
Accuracy (%)
100
80
60
40
20
0
75.67%
77.20%
83.45%
85.00%
87.00%
98.10%
CNN
Deep Learning
LSTM
CNN + LSTM (Hybrid)
Ensemble Stacking
Hybrid Model (NB+SVM+BLSTM)

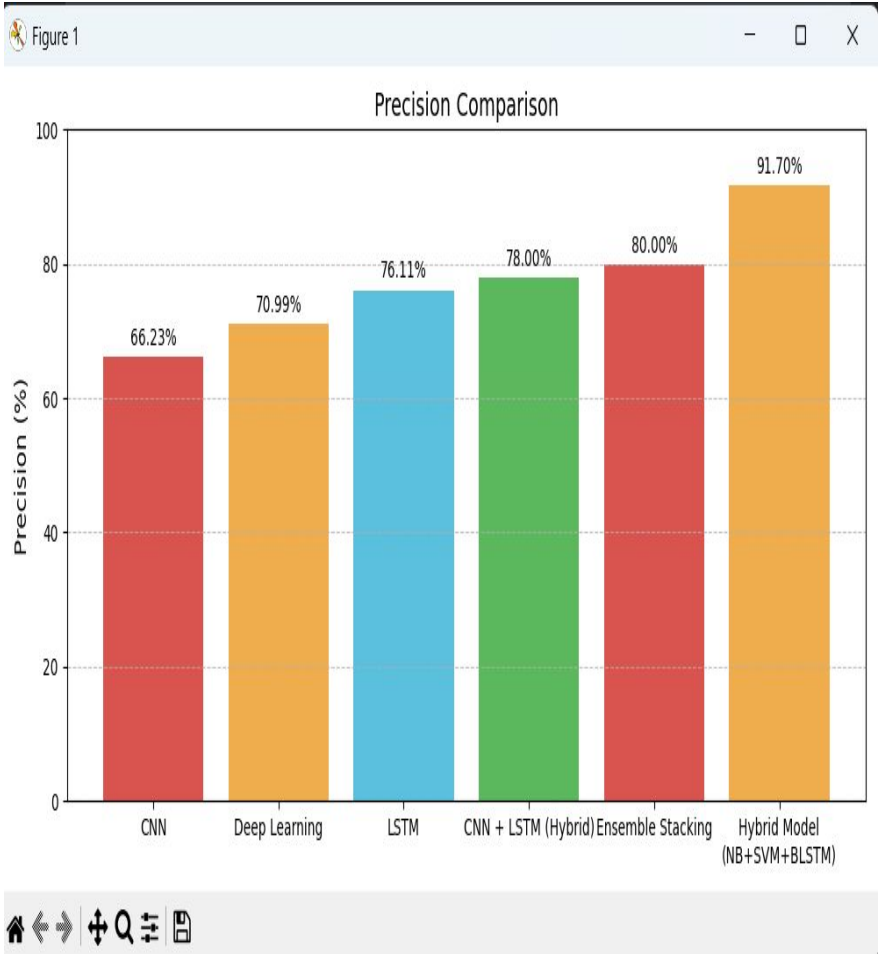


Figure 1
Precision Comparison
Precision (%)
100
80
60
40
20
0
66.23%
70.99%
76.11%
78.00%
80.00%
91.70%
CNN
Deep Learning
LSTM
CNN + LSTM (Hybrid)
Ensemble Stacking
Hybrid Model (NB+SVM+BLSTM)

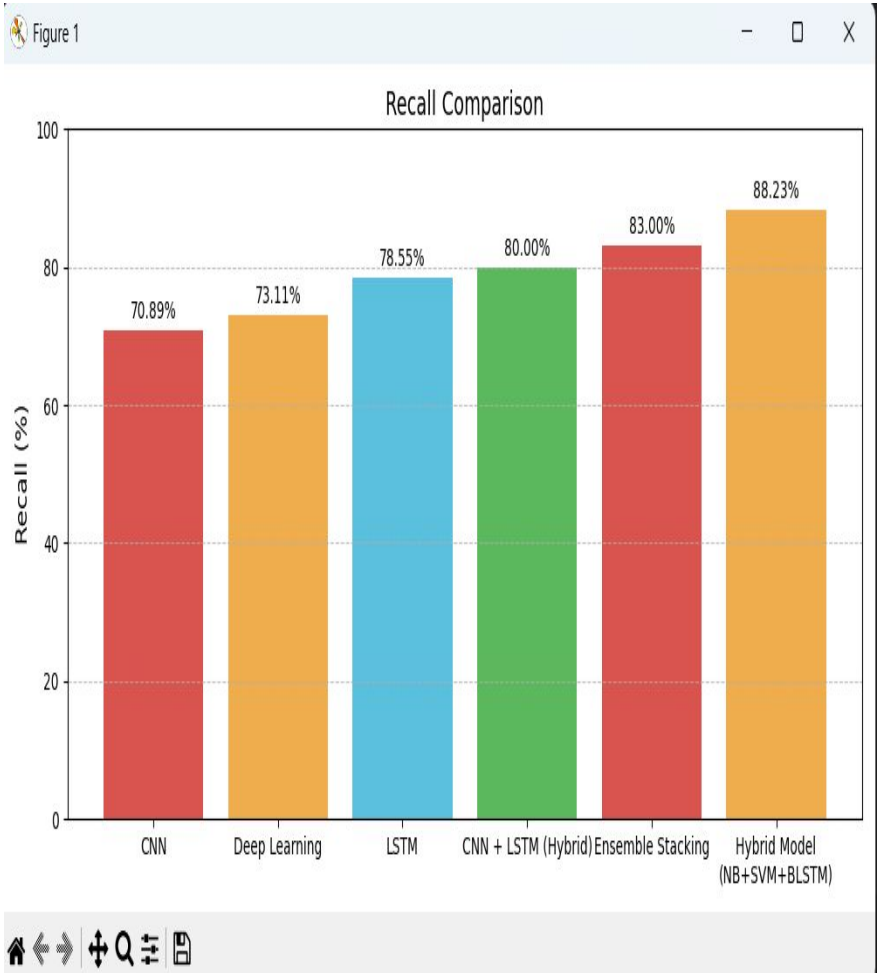


Figure 1
Recall Comparison
Recall (%)
100
80
60
40
20
0
70.89%
73.11%
78.55%
80.00%
83.00%
88.23%
CNN
Deep Learning
LSTM
CNN + LSTM (Hybrid)
Ensemble Stacking
Hybrid Model (NB+SVM+BLSTM)

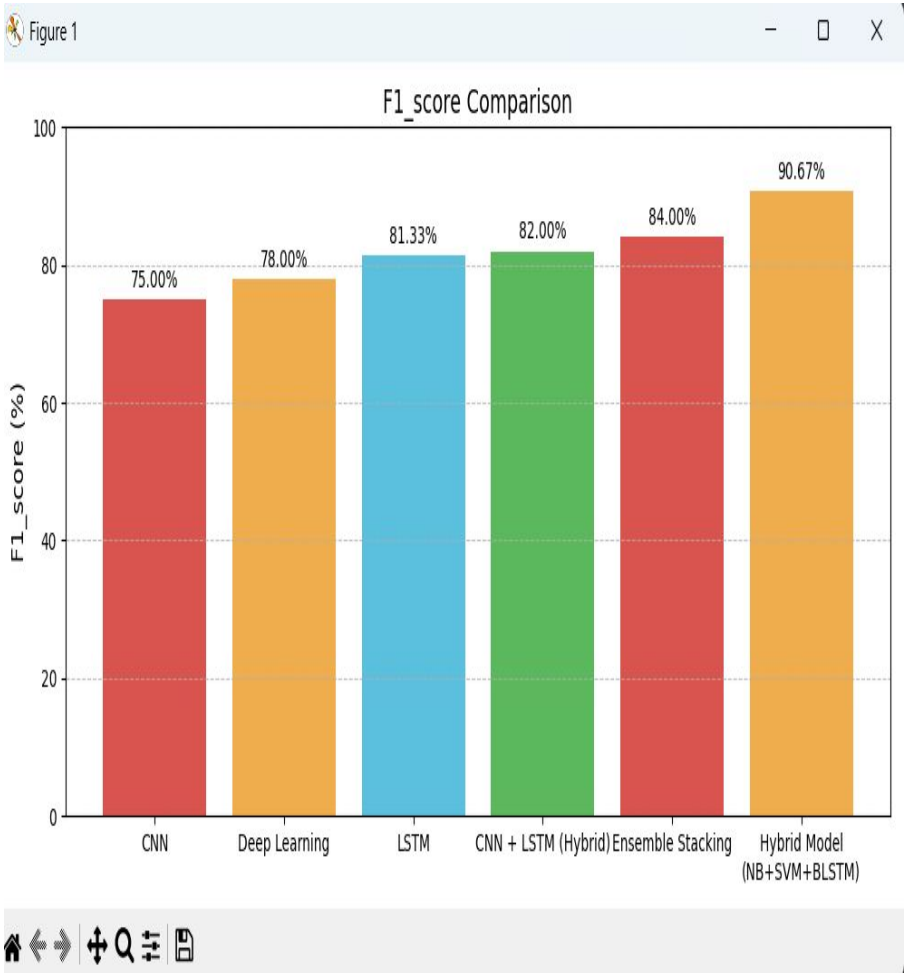


Figure 1
F1_score Comparison
F1_score (%)
100
80
60
40
20
0
75.00%
78.00%
81.33%
82.00%
84.00%
90.67%
CNN
Deep Learning
LSTM
CNN + LSTM (Hybrid)
Ensemble Stacking
Hybrid Model (NB+SVM+BLSTM)

Conclusion:

- **Enhanced Detection Accuracy**: The proposed Double Layered Hybrid Approach (DLHA), combining Naive Bayes, SVM, and BLSTM, significantly improves detection accuracy for both frequent (DoS, Probe) and rare (R2L, U2R) attack types compared to traditional methods.

- **Effective Use of PCA and BLSTM**: The use of Principal Component Analysis (PCA) for dimensionality reduction and BLSTM for temporal pattern recognition enables the system to handle complex, sequential network traffic more efficiently.

- **Robust and Scalable Solution**: DLHA provides a robust, scalable, and adaptable framework for real-time intrusion detection, making it suitable for modern dynamic network environments facing evolving cyber threats.

FUTURE SCOPE:

- **Real-time detection** – Upgrade to monitor live network traffic.
- **System integration** – Connect with firewalls and SIEM tools.
- **Adaptive learning** – Enable the model to learn from new attacks.
- **IoT deployment** – Optimize for use in smart and edge devices.
- **Multi-source analysis** – Use logs and external data for better accuracy.

REFERENCES

[1] M. Mohammadi, A. Navaras, & M. H. Amini (2021). Double Layered Hybrid Approach for Network Intrusion Detection System.*IEEE Xplore.* DOI: 10.1109/ICICT52614.2021.9562534.

[2] S. Shi, D. Han, & M. Cui (2023). A Multimodal Hybrid Parallel Network Intrusion Detection Model**.** *Connection Science, Taylor & Francis.* 35(1), 2227780. DOI: 10.1080/09540091.2023.2227780

[3] Muhammad Sajid, Kaleem Razzaq Malik, Ahmad Almogren, Tauqeer Safdar Malik, Ali Haider Khan, Jawad Tanveer, & Ateeq Ur Rehman (2024). Enhancing Intrusion Detection: A Hybrid Machine and Deep Learning Approach. *Journal Cloud Computing, Springer.* Volume 13, Article number: 123. DOI: 10.1186/s13677-024-00685-x.

[4] R. Jalili, S. Imani, & M. R. Aminzadeh (2021). A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs. *Proceedings of the 10th ACM International Conference on Security of Information and Networks, ACM Digital Library.* DOI: 10.1145/3465481.3469190.

[5] Network Intrusion Detection and Prevention System Using Hybrid Machine Learning with Supervised Ensemble Stacking Model. (2024). *Security and Communication Networks, Hindawi.* DOI: 10.1155/2024/5775671.

Thank you