VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi, Karnataka, INDIA

Mini Project
Report on

Cyber Threat Detection

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering in
Artificial Intelligence and Machine Learning

Submitted By

Aditya Sinha **1GA21AI005**

SP Sri Harshith **1GA21AI054**

Under the Guidance of

VIJAYA DALAWAI
Assistant Professor

Department of Artificial Intelligence and Machine Learning

GLOBAL ACADEMY OF TECHNOLOGY

Rajarajeshwarinagar, Bengaluru - 560 098

2023 – 2024

GLOBAL ACADEMY OF TECHNOLOGY

Department of Artificial Intelligence and Machine Learning

CERTIFICATE

Certified that the project work which is entitled as **CYBER THREAT DETECTION SYSTEM** carried out by **Mr. ADITYA SINHA,** USN **1GA21AI005** and **Mr. SP SRI HARSHITH**, USN **1GA21AI054,** a bonafede student of **Global Academy of Technology** in partial fulfillment for the award of Bachelor of Engineering in Artificial Intelligence and Machine Learning of the Visveswaraya Technological University, Belgaum during the year 2023-24. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of the Project work prescribed for the said Degree.

Prof. Vijaya Dalawai Assistant Professor Dept. of AI&ML	Dr. Roopa BS Professor and Head Dept. of AI&ML	Dr. Rana Pratap Reddy Principal GAT, Bengaluru.

GLOBAL ACADEMY OF TECHNOLOGY
Department of Artificial Intelligence and Machine Learning

DECLARATION

We, **Aditya Sinha**, bearing USN **1GA21AI005**, **S P Sri Harshith**, bearing USN **1GA21AI054**, students of Sixth Semester B.E, Department of Artificial Intelligence and Machine Learning Engineering, Global Academy of Technology, Raja Rajeshwarinagar Bengaluru, declare that the Project Work entitled ”Cyber Threat Detection” has been carried out by us and submitted in partial fulfillment of the course requirements for the award of degree in Bachelor of Engineering in Artificial Intelligence and Machine Learning Engineering from Visvesvaraya Technological University, Belagavi during the academic year 2023 – 2024.

1. Aditya Sinha 1GA21AI005
2. SP Sri Harshith 1GA21AI054

Place: Bengaluru
Date: 31-08-2024

ACKNOWLEDGEMENT

The satisfaction and the euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. The constant guidance of these people and the encouragement provided crowned our efforts with success and glory. Although it is not possible to thank all the members, we take this opportunity to express our gratitude to one and all.

We are grateful to the management and our institute, the **GLOBAL ACADEMY OF TECHNOLOGY,** for its very ideals and inspiration for having provided us with the facilities that made this project a success.

We express our sincere gratitude to **Dr. N. Rana Pratap Reddy**, Principal, Global Academy of Technology for the support and encouragement.

We wish to place on record our thanks to Dr. Roopa BS, Professor and HOD, Department of AI&ML, Global Academy of Technology, for the constant encouragement provided to us.

We are indebted with a deep sense of gratitude for the constant inspiration, encouragement, timely guidance, and valid suggestions given to us by our guide **Mrs. Vijaya Dalawai**, **Assistant Professor**, Department of AI&ML, Global Academy of Technology.

We are thankful to all the staff members of the department for providing relevant information and helping in different capacities in carrying out this project.

Last, but not least, we owe our debts to our parents, friends, and those who directly or indirectly have helped us to make the phase-2 project work a success.

ADITYA SINHA **1GA21AI005**
SRI SP HARSHITH **1GA21AI054**

TABLE OF CONTENTS :

CHAPTER 1

ABSTRACT

In an era where cyber threats are increasingly sophisticated, effective detection mechanisms are crucial for safeguarding digital infrastructures. This project focuses on developing a comprehensive cyber threat detection system using both machine learning (ML) and deep learning (DL) techniques, specifically trained on the NSL-KDD dataset—a widely recognized benchmark for intrusion detection systems.

The machine learning component of the project employed several algorithms, with XGBoost emerging as the most effective model, achieving the highest accuracy among the tested methods. This model demonstrated superior performance in detecting various types of network intrusions, providing a robust baseline for comparison.

Building on the success of the machine learning models, the project explored advanced deep learning approaches, employing a stacking model and a hybrid model to enhance detection capabilities. The stacking model combined predictions from multiple base models, effectively capturing complex patterns in the data. The hybrid model integrated both machine learning and deep learning techniques, aiming to leverage the strengths of each approach.

The results indicate that the hybrid model, in particular, offered significant improvements in detection accuracy and generalization, outperforming traditional methods. The final system exhibits a high detection rate, low false-positive rate, and strong adaptability to evolving cyber threats, making it a valuable tool for real-time network security.

This project not only highlights the potential of integrating ML and DL techniques for cyber threat detection but also sets a foundation for future research and development in the field of cybersecurity.

CHAPTER 2

INTRODUCTION

The rapid evolution of digital technologies has led to an unprecedented expansion of networked systems, which, while enabling greater connectivity and innovation, has also increased the vulnerability to cyber threats. These threats, including malware, phishing, and network intrusions, pose significant risks to both individuals and organizations, leading to data breaches, financial loss, and compromised security.

Traditional cybersecurity measures, although effective to some extent, struggle to keep pace with the dynamic nature of cyber attacks. This necessitates the development of more sophisticated detection mechanisms that can anticipate, identify, and neutralize threats in real-time. Machine learning (ML) and deep learning (DL) have emerged as powerful tools in this regard, offering the ability to analyze large volumes of data, identify patterns, and predict anomalies with high accuracy.

This project aims to design and implement an advanced cyber threat detection system by leveraging both machine learning and deep learning techniques. The system is trained and evaluated using the NSL-KDD dataset, a benchmark dataset widely used for intrusion detection research. The choice of this dataset ensures that the models are tested against a diverse range of cyber threats, making them more resilient and effective in real-world scenarios.

The project is structured in two phases. In the first phase, several machine learning algorithms are explored, with XGBoost emerging as the most accurate model for detecting network intrusions. The second phase focuses on deep learning approaches, where a stacking model and a hybrid model are developed to further enhance detection capabilities.

CHAPTER 3

MOTIVATION AND CONTRIBUTIONS:

- The motivation behind this research arises from the increasing complexity and frequency of cyber threats, which challenge the security of digital systems globally. Traditional detection methods are becoming inadequate in this rapidly evolving landscape, highlighting the need for intelligent systems capable of real-time threat detection and response. Machine learning (ML) and deep learning (DL) offer powerful tools for enhancing cybersecurity, with their ability to analyze large datasets and predict anomalies. This project is driven by the need to develop a more adaptive and effective cyber threat detection system using these advanced techniques.
- This study significantly contributes to cybersecurity by integrating ML and DL methods, specifically through the application of XGBoost and advanced deep learning models like stacking and hybrid approaches. By evaluating these models on the NSL-KDD dataset, we provide critical insights into their effectiveness against various cyber threats.
- We also introduce a hybrid model that combines the strengths of ML and DL, achieving improved accuracy and better detection of complex threats. Our research focuses on real-world applicability, addressing challenges such as processing speed, resource efficiency, and adaptability to new threats.
- Finally, this project lays the groundwork for future research in intelligent cyber threat detection, demonstrating the potential of ML and DL to advance the field and setting benchmarks for developing more robust and adaptive security systems.

CHAPTER 4

CONCISE LITERATURE SURVEY

Paper Title	Author, Year and Publication with citation	Proposed Work	Methodology/Implementation	Advantages	Disadvantages
Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation on Perspective	Kamran Shaukat, Suhuai Luo, Shan Chen (June 18,2021)	It builds on existing works like CodeBlue and AlarmNet, which are based on ECC and AEC security models, respectively. The paper emphasizes the importance of effective data enlistment in various fields	The study uses SVM with specific features and vectors, leveraging TensorFlow for model optimization and parallel processing	The paper proposes a novel approach for detecting data modification intrusion in Wireless Body Area Networks (WBANs) using machine learning techniques	Existing works like CodeBlue and AlarmNet produce limited quality answers and lack comprehensive security solutions
Machine Learning	Hongyu Liu and	References previous works	Utilizing various benchmark	Uses machine learning	Many existing IDS models are

and Deep Learning Methods for Intrusion Detection Systems: A Survey	Bo Lang (17 October 2019)	and datasets such as KDD Cup'99, NSL-KDD, UNSW-NB15, and CSE-CIC-IDS2018.	datasets and machine learning evaluation metrics (e.g., precision, recall, accuracy, F-measure).	techniques to enhance intrusion detection systems (IDS)	evaluated on outdated datasets, leading to lower detection rates for certain attack types.
Cyber Security: Threat Detection Model based on Machine learning Algorithm	Kushal Rashmikant Dalal	The paper introduces a method for detecting anomalies in web log analysis using a combination of clustering algorithms and machine learning models .	The study uses clustering algorithms and a combination of classifiers and models for anomaly detection	It references the first anomaly network and compares its method with other models to highlight performance improvements	Existing methods lack adaptability and proper quality assurance
Machine Learning and Deep Learning Methods	Yang Xin,Lingshuang Kong , Zhi Li, Yuling	It discusses the use of clustering types in intrusion detection and compares the performance of	The study uses a combination of classification, clustering, and sandbox environment	The paper evaluates various machine learning techniques and	Current methods show limited satisfactory results, indicating the

for Cybersecurity	Chen , Yanmiao Li, Hongliang Zhu , Mingcheng Gao	six machine learning methods and six ensemble methods	analysis to evaluate different techniques	their ensemble methods for network intrusion detection	need for further research

CHAPTER 5

PROBLEM STATEMENT AND OBJECTIVES

PROBLEM STATEMENT :

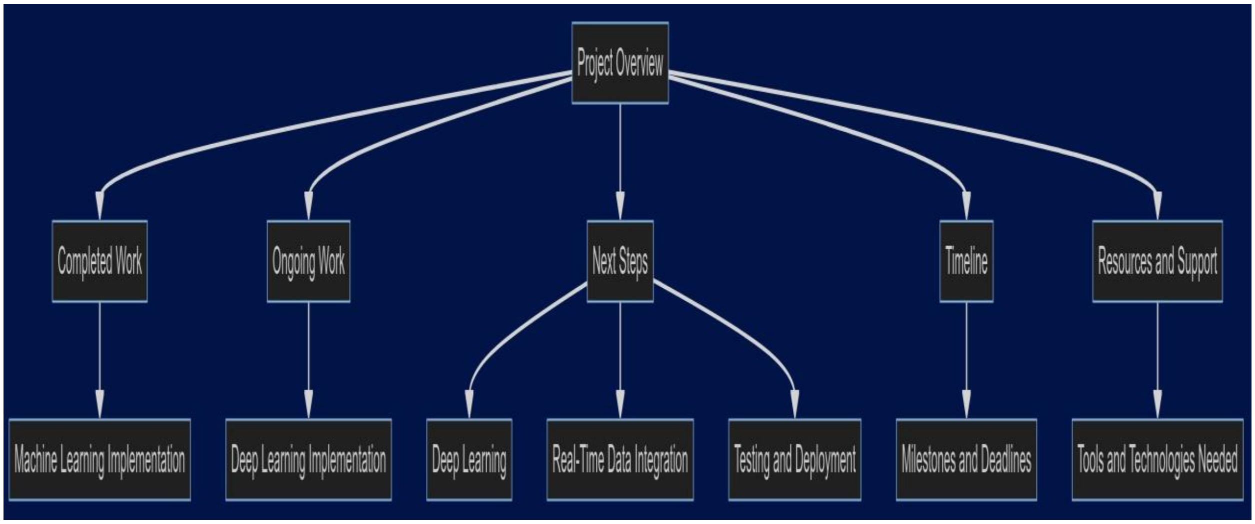
Current intrusion detection systems (IDS) using machine learning (ML) and deep learning (DL) face challenges in accuracy, false positives, and adaptability. This research aims to develop an advanced DL-based IDS framework to enhance threat detection and response by integrating CNNs, RNNs, and autoencoders and few more to validated through benchmark datasets and real-world data.

OBJECTIVES :

- To develop a more accurate and efficient intrusion detection system for WBANs using machine learning .
- To develop an effective anomaly detection system for web log analysis using machine learning .
- To evaluate and identify the most effective machine learning techniques for network intrusion detection .
- Create an advanced IDS that leverages the strengths of both machine learning and deep earning.

CHAPTER 6

SYSTEM ARCHITECTURAL DESIGN



The diagram outlines the structured workflow for the cyber threat detection system:

- **Project Overview**: Central to the architecture, summarizing the entire project workflow.
- **Completed Work**: Focuses on the successful implementation of machine learning models like XGBoost, optimized for threat detection using the NSL-KDD dataset.
- **Ongoing Work**: Involves deep learning implementation, developing advanced models such as stacking and hybrid approaches to improve accuracy.
- **Next Steps**: Plans include real-time data integration for continuous monitoring and enhancing deployment pipelines for real-world application.
- **Timeline**: Outlines key milestones and deadlines to ensure timely progress.
- **Resources and Support**: Highlights the tools, technologies, and support needed for the successful execution and deployment of the system.

CHAPTER 7

IMPLEMENTATION DETAILS

1. **Data Collection and Preparation**:

   The NSL-KDD dataset, comprising approximately 4,900,000 records, is used for training and testing the models. The dataset is categorized into normal traffic and various types of attacks, including DoS, Probe, R2L, and U2R. Each record contains features related to network traffic and connection details, which help in identifying different types of network intrusions. Which are put into classes like :

   - Normal
   - DoS (Denial of Service)
   - Probe (Probing/Scanning)
   - R2L (Remote to Local)
   - U2R (User to Root)

2. **Data Cleaning and Preprocessing**:

   This phase involves handling missing values, outliers, and ensuring data consistency. Preprocessing includes normalization or scaling of features, encoding categorical variables, and splitting the dataset into training and testing sets.

3. **Model Selection**:

   Various machine learning and deep learning models are selected for testing, including:

   - **XGBoost** for efficient, high-performance decision trees.
   - **Stacking Models** for combining the predictions of multiple algorithms to improve accuracy.
   - **Hybrid Models** (e.g., combining ML with DL) to leverage the strengths of both approaches.

4. **Model Training**:

   - **Data Splitting**: Data is split into training and validation sets.
   - **Model Training**: Each model is trained using the processed dataset, focusing on learning the characteristics of both normal traffic and various attacks.
   - **Transfer Learning**: Applied where necessary, particularly in deep learning models, to enhance performance by leveraging pre-trained networks.

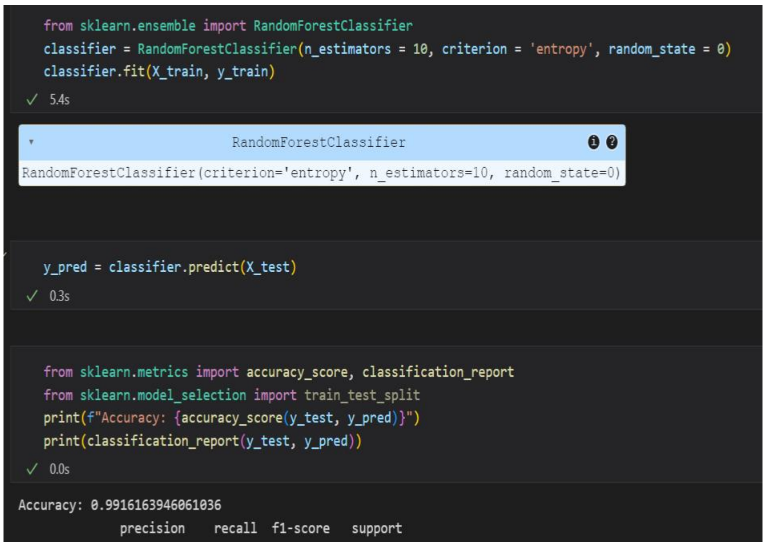
5. **Evaluation**:

   The evaluation involves using a separate test set containing unseen data to assess the models' ability to accurately detect different types of network intrusions. Accuracy, precision, recall, and F1-score are key metrics used to measure the performance of each model across all attack categories.

CHAPTER 8

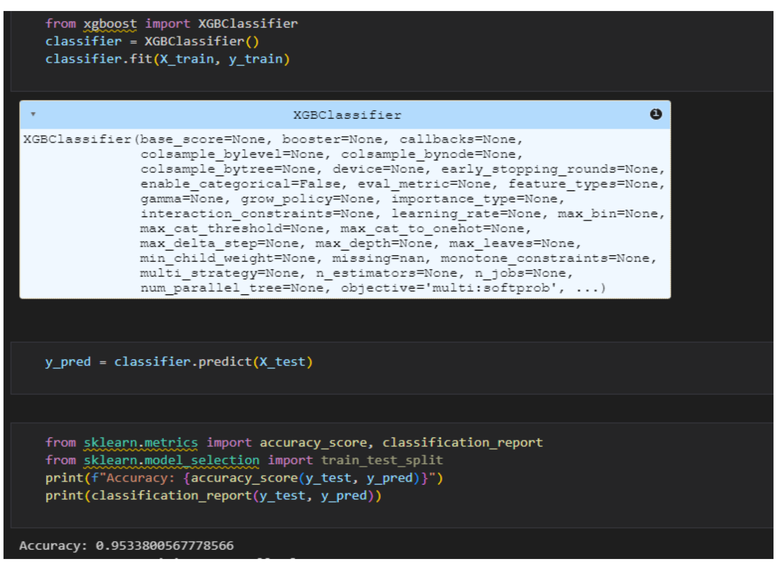
RESULTS AND DISCUSSION

a) Random Forest Classifier :



```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)
✓ 5.4s

RandomForestClassifier
RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)

y_pred = classifier.predict(X_test)
✓ 0.3s

from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print(classification_report(y_test, y_pred))
✓ 0.0s

Accuracy: 0.9916163946061036
              precision    recall  f1-score   support
```

b) XG boost Classifier :



```
from xgboost import XGBClassifier
classifier = XGBClassifier()
classifier.fit(X_train, y_train)

XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, objective='multi:softprob', ...)

y_pred = classifier.predict(X_test)

from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print(classification_report(y_test, y_pred))

Accuracy: 0.9533800567778566
```

c) Decision Tree Classifier :

```
#decision tree model

from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)
✓ 11.2s

DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=0)

y_pred = classifier.predict(X_test)
✓ 0.2s

from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print(classification_report(y_test, y_pred))
✓ 0.0s
Accuracy: 0.997959545777147
              precision    recall  f1-score   support
```

d) SVM Classifier :

```
#svm model

from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)     #linear kernel n state is
classifier.fit(X_train, y_train)

SVC
SVC(kernel='linear', random_state=0)

y_pred = classifier.predict(X_test)

from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print(classification_report(y_test, y_pred))

Accuracy: 0.7893896380411639
              precision    recall  f1-score   support
```

e) KNN Classifier :

```
#KNN model

from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)

KNeighborsClassifier
KNeighborsClassifier()

y_pred = classifier.predict(X_test)

from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print(classification_report(y_test, y_pred))

Accuracy: 0.8486958836053939
              precision    recall  f1-score   support
```

f) Naïve Bayes Classifier :

```
#naive bayes

from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

GaussianNB
GaussianNB()

y_pred = classifier.predict(X_test)

from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print(classification_report(y_test, y_pred))

Accuracy: 0.20231547196593327
              precision    recall  f1-score   support
```

g) Stacking DL Model :

```
history = model.fit(stacked_features_train, y_train, validation_data=(stacked_features_test, y_test), epochs=10, v
[37]                                                                                                        Python
...
Epoch 1/10
c:\Users\Pyush\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\backend\tensorflow\nn.py:707: UserW
  output, from_logits = _get_logits(
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 25s 5ms/step - accuracy: 0.9987 - loss: 0.0544 - val_accuracy: 0.7936 - val_loss: 1.707
Epoch 2/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 22s 5ms/step - accuracy: 0.9999 - loss: 0.0067 - val_accuracy: 0.7908 - val_loss: 2.094
Epoch 3/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 22s 6ms/step - accuracy: 0.9998 - loss: 0.0056 - val_accuracy: 0.7911 - val_loss: 2.079
Epoch 4/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 22s 6ms/step - accuracy: 0.9999 - loss: 0.0046 - val_accuracy: 0.8024 - val_loss: 1.959
Epoch 5/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 22s 6ms/step - accuracy: 0.9999 - loss: 0.0047 - val_accuracy: 0.7896 - val_loss: 2.273
Epoch 6/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 41s 6ms/step - accuracy: 0.9999 - loss: 0.0046 - val_accuracy: 0.7934 - val_loss: 2.176
Epoch 7/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 22s 6ms/step - accuracy: 0.9999 - loss: 0.0048 - val_accuracy: 0.7897 - val_loss: 2.497
Epoch 8/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 22s 6ms/step - accuracy: 0.9999 - loss: 0.0046 - val_accuracy: 0.7897 - val_loss: 2.194
Epoch 9/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 22s 6ms/step - accuracy: 0.9999 - loss: 0.0044 - val_accuracy: 0.7855 - val_loss: 2.397
Epoch 10/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 41s 6ms/step - accuracy: 0.9999 - loss: 0.0047 - val_accuracy: 0.7662 - val_loss: 2.628
```

h) Hybrid DL Model :

```
hy_model.fit(hybrid_train, y_train, validation_data=(hybrid_test, y_test), epochs=10, verbose=1)
[44]                                                                                                        Python
...
Epoch 1/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 25s 5ms/step - accuracy: 0.9723 - loss: 0.2117 - val_accuracy: 0.8018 - val_loss: 1.377
Epoch 2/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 21s 5ms/step - accuracy: 0.9983 - loss: 0.0391 - val_accuracy: 0.7941 - val_loss: 1.525
Epoch 3/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 21s 5ms/step - accuracy: 0.9990 - loss: 0.0232 - val_accuracy: 0.7881 - val_loss: 1.339
Epoch 4/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 22s 6ms/step - accuracy: 0.9989 - loss: 0.0219 - val_accuracy: 0.7985 - val_loss: 1.835
Epoch 5/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 27s 7ms/step - accuracy: 0.9991 - loss: 0.0216 - val_accuracy: 0.7876 - val_loss: 1.829
Epoch 6/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 28s 7ms/step - accuracy: 0.9990 - loss: 0.0195 - val_accuracy: 0.7834 - val_loss: 2.033
Epoch 7/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 35s 6ms/step - accuracy: 0.9992 - loss: 0.0192 - val_accuracy: 0.7954 - val_loss: 1.839
Epoch 8/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 41s 6ms/step - accuracy: 0.9989 - loss: 0.0236 - val_accuracy: 0.7823 - val_loss: 2.236
Epoch 9/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 30s 7ms/step - accuracy: 0.9993 - loss: 0.0187 - val_accuracy: 0.7886 - val_loss: 1.877
Epoch 10/10
3937/3937 ━━━━━━━━━━━━━━━━━━━━ 28s 7ms/step - accuracy: 0.9994 - loss: 0.0171 - val_accuracy: 0.7892 - val_loss: 2.345
```

CHAPTER 9

CONCLUSION

This project successfully demonstrated the potential of machine learning and deep learning techniques in enhancing cyber threat detection. By leveraging the NSL-KDD dataset, we developed and evaluated various models, including XGBoost and hybrid deep learning approaches, to effectively identify and classify network intrusions. The results highlight the superior accuracy and adaptability of these models in detecting both known and novel threats, addressing the limitations of traditional security measures.

Our hybrid model, which combines the strengths of machine learning and deep learning, proved particularly effective in identifying complex attack patterns, underscoring the importance of integrating multiple methodologies for robust cybersecurity solutions. The insights gained from this project not only contribute to the ongoing development of more intelligent and adaptive security systems but also set a benchmark for future research in the field.

Moving forward, the integration of real-time data and further optimization of these models could enhance their applicability in dynamic, real-world environments, ensuring that cybersecurity measures remain resilient against evolving threats.

CHAPTER 10

FUTURE WORKS

a) **Model Optimization**:

Further refining the hybrid models by experimenting with different combinations of machine learning and deep learning techniques could lead to even higher detection accuracy. Additionally, exploring new architectures, such as transformer-based models, could provide insights into more efficient threat detection.

b) **Scalability and Deployment**:

Scaling the models for deployment in large-scale network environments is another critical area for future work. This includes testing the models in cloud-based infrastructures and ensuring they can maintain performance under varying network conditions and loads.

c) **Adaptation to Emerging Threats**:

As cyber threats continue to evolve, it will be essential to keep the models updated. Incorporating continuous learning mechanisms that allow the models to adapt to new types of attacks without manual intervention could enhance the system's long-term effectiveness.

d) **Realtime Data Integration:**

Future efforts could focus on integrating real-time data streams to enable the models to detect and respond to cyber threats as they occur. This would involve optimizing the existing models for low-latency performance, ensuring they can handle the demands of real-world applications.

e) **Cross Domain Application**

Finally, exploring the application of these models in other domains, such as IoT security or mobile network protection, could extend their utility and impact, providing robust solutions across various cybersecurity challenges.

REFERENCES

[1] K. Shaukat, S. Luo, S. Chen and D. Liu, "Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective," 2020 International Conference on Cyber Warfare and Security (ICCWS), Islamabad, Pakistan, 2020, pp. 1-6, doi: 10.1109/ICCWS48432.2020.9292388. keywords: {Machine learning;Computer crime;Unsolicited e-mail;Malware;Cyberspace;Support vector machines;Decision trees;Cyber Threat;Cybercrime;Performance Evaluation;Machine Learning Application;Intrusion Detection System;Malware Detection;Spam Classification},

[2] Liu, H.; Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Appl. Sci.* 2019, *9*, 4396. https://doi.org/10.3390/app9204396

[3] K. R. Dalal and M. Rele, "Cyber Security: Threat Detection Model based on Machine learning Algorithm," 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2018, pp. 239-243, doi: 10.1109/CESYS.2018.8724096. keywords: {Machine learning algorithms;Machine learning;Data models;Training;Computer security;Malware;Machine Learning;Sandbox},

[4] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," in IEEE Access, vol. 6, pp. 35365-35381, 2018, doi: 10.1109/ACCESS.2018.2836950.keywords: {Machine learning;Intrusion detection;Feature extraction;Machine learning algorithms;Computer security;Cybersecurity;intrusion detection;deep learning;machine learning},

[5] Network Intrusion Detection System using Deep Learning,Procedia Computer Science,https://doi.org/10.1016/j.procs.2021.05.025.(https://www.sciencedirect.com/science/article/pii/S1877050921011078). The widespread use of interconnectivity and interoperability of computing systems have become an indispensable necessity to enhance our daily activities. Simultaneously, it opens a path to exploitable vulnerabilities that go well beyond human control capability. The vulnerabilities deem cyber-security mechanisms essential to assume communication exchange. Secure communication requires security measures to combat the threats and needs advancements to security measures that counter evolving security threats.