**Advanced Network Threat Detection Using Deep Learning Models**

***Project submitted to***

***Shri Ramdeobaba College of Engineering & Management, Nagpur in partial fulfillment of requirement for the award of***

***degree of***

## Bachelor of Engineering

*In*

## COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE

**AND MACHINE LEARNING)**

Ms. Khushi Kharate

Ms. Panchami Vishwakarma

Mr. Aryman Pandey

Ms. Rama Labhe

Guided By:

Dr. Yogesh Thakre

### 

### Shri Ramdeobaba College of Engineering & Management, Nagpur

**440013**

(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University Nagpur)

### December 2024

#### SHRI RAMDEOBABA COLLEGE OF ENGINEERING & MANAGEMENT, NAGPUR

(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj

NagpurUniversity Nagpur)

Department of Computer Science and Engineering

# CERTIFICATE

This is to certify that the project on **“Advanced Network Threat Detection Using Deep Learning Models ”** is a bonafide work of

1. Ms. Khushi Kharate
2. Ms. Panchami Vishwakarma
3. Mr. Aryman Pandey
4. Mr. Rama Labhe

submitted to the Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur in partial fulfillment of the award of a Degree of Bachelor of Engineering, in Computer Science and Engineering (Artificial Intelligence and Machine Learning). It has been carried out at the Department Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur during the academic year 2023-24.

Date: 24-10-2024

Place: Nagpur

Dr. Yogesh Thakre Dr. Preeti Voditel Project guide H.O. D

Department of Computer Science and Engineering (AIML)

### DECLARATION

I, hereby declare that the project titled **“Advanced Network Threat Detection Using Deep Learning Models”** submitted herein,has been carried out in the Department of Computer Science and Engineeringof Shri Ramdeobaba College of Engineering & Management, Nagpur. The work is original and has not been submitted earlier as a whole or part for theaward of any degree

/ diploma at this or any other institution / University

Date: 24-10-2024

Place: Nagpur

#### Ms. Khushi Kharate Ms. Panchami Vishwakarma

**(Roll no.: 06) (Roll no.: 14)**

#### Mr. Aryman Pandey Mr. Rama Labhe

**(Roll no.: ) (Roll no.: 16)**

### Approval Sheet

This thesis entitled **Advanced Network Threat Detection Using Deep Learning Models** by

Ms. Khushi Kharate

Ms. Panchami Vishwakarma

Mr. Aryman Pandey

Ms. Rama Labhe

is approved for the degree of Bachelor of Engineering, in Computer Science & Engineering.

Name & signature of Supervisor(s) Name & signature of External. Examiner(s)

Name & signature of HOD

Date: 17-12-23

Place: Nagpur

### ACKNOWLEDGEMENT

We would like to express our deep and sincere gratitude to our guide **Dr. Yogesh Thakre,** Professor of Computer Science and Engineering Department(AIML), RCOEM,for giving us the opportunity to work on this project and providing valuable guidance throughout the project. It was a great privilege and honor to work under her guidance. We are extremely grateful for the experience we had in this project with her.

We express our sincere gratitude to **Dr. Preeti Voditel,** Head of the Department of Computer Science Department (AIML), RCOEM for his guidance. Talent wins games, but teamwork and intelligence win championships. We would like to take this opportunity to express our deep gratitude to all those who extended their support and guided us to complete this project.

#### Signature and Name of the Students

**Ms. Khushi Kharate Mr. Panchami Vishwakarma**

**Ms. Aryman Pandey Mr. Rama Labhe**

### ABSTRACT

#### In today's rapidly evolving digital landscape, network infrastructures face an ever-increasing number of sophisticated cyber threats. Traditional network security measures, while effective to some extent, often fall short in detecting complex and novel intrusions, leaving networks vulnerable to potential breaches. This project seeks to address these challenges by developing an advanced Network Intrusion Detection System (NIDS) that leverages state-of-the-art machine learning techniques to detect and classify intrusions in real-time.

#### The core objective of this project is to design a NIDS capable of identifying both known and unknown threats through the use of supervised and unsupervised learning models. By utilizing deep learning, anomaly detection algorithms, and pattern recognition, the system can accurately differentiate between legitimate network traffic and malicious activities, such as Denial of Service (DoS) attacks, port scans, malware propagation, and insider threats. The integration of advanced machine learning algorithms allows the NIDS to continuously learn from new data, improving detection accuracy and reducing false positives over time.

#### One of the key features of this NIDS is its ability to operate in real-time, providing instant alerts and automated threat responses to detected intrusions. The system is designed to seamlessly integrate with existing network security infrastructures, enhancing the overall security framework without requiring significant architectural changes. By implementing proactive security measures, the NIDS can mitigate risks before they escalate into full-scale attacks, ensuring minimal downtime and disruption to critical network services.

#### Furthermore, the system's scalability makes it well-suited for both small and large network environments, including cloud-based and distributed networks. By employing a multi-layered detection approach, the system ensures comprehensive coverage across various network protocols and architectures. This project also emphasizes the importance of reducing the time between detection and response, allowing organizations to respond to potential threats in real-time.

#### In conclusion, this NIDS project aims to significantly enhance network security by providing a robust, adaptive, and scalable solution for detecting and responding to network intrusions. By leveraging cutting-edge machine learning technologies, the system will offer superior protection against evolving cyber threats, contributing to a safer, more resilient network environment.

#### TABLE OF CONTENTS

**List of Figures and Formulas**

**Chapter 1**

[INTRODUCTION 1](#_TOC_250005)

* 1. [Introduction 2](#_TOC_250004)
  2. [Motivation 3](#_TOC_250003)
  3. [Objectives 3](#_TOC_250002)
  4. [Problem Definition 4](#_TOC_250001)
  5. Brief Description of the system 4
     1. Front end 4
     2. Back end 4

Chapter 2

LITERATURE SURVEY 5

Chapter 3

TECHNICAL SPECIFICATION 9

Chapter 4

WORKING 12

Chapter 5

CONCLUSION 16

Chapter 6

[REFERENCES 17](#_TOC_250000)

**Chapter 1**

## INTRODUCTION

#### Introduction

In the field of network security, various methods are used to detect and mitigate cyber threats, typically divided into conventional, software-based, and hardware-based approaches. This project aims to enhance network security by developing an advanced Network Intrusion Detection System (NIDS) that integrates machine learning techniques with traditional security measures. By leveraging both supervised and unsupervised learning models, the NIDS offers real-time threat detection, identifying known and unknown threats such as Denial of Service (DoS) attacks, malware, and insider threats. The approach combines hardware and software solutions to improve detection accuracy, reduce false positives, and enable automated threat response. Challenges such as integrating machine learning within existing network infrastructures and ensuring low-latency detection are considered, aiming to provide a scalable and resilient solution that aligns with modern security needs.

* 1. **Motivation**

In the rapidly evolving digital landscape, network infrastructures face an ever-growing range of sophisticated cyberattacks. Traditional security methods, such as firewalls and antivirus software, typically rely on signature-based detection, making them effective against known threats but inadequate for detecting new, more complex attacks. These conventional techniques struggle to recognize zero-day vulnerabilities, advanced persistent threats, and rapidly evolving attack vectors like Distributed Denial of Service (DDoS) and ransomware. As cybercriminals continue to innovate, organizations are left vulnerable, risking significant operational disruptions and data breaches. The inability of traditional systems to proactively identify and respond to novel threats highlights the urgent need for more adaptive and intelligent security solutions.

This project is driven by the goal of developing a Network Intrusion Detection System (NIDS) that integrates advanced machine learning techniques to enhance real-time detection and response to both known and unknown threats. Machine learning offers the ability to continuously learn from new data and detect anomalies that conventional systems might overlook, significantly improving threat detection accuracy and reducing false positives. As network environments expand, especially with the rise of cloud computing and distributed systems, there is a critical need for scalable, real-time solutions that can adapt to diverse infrastructures. By incorporating machine learning into NIDS, this project aims to provide a proactive, scalable defense mechanism that strengthens overall network security and contributes to greater cyber resilience in an increasingly interconnectedworld.

#### 1.3 Objectives

#### Real-Time Monitoring: Implement continuous monitoring of network traffic to detect and alert on suspicious activities as they occur . Utilize advanced algorithms to ensure timely detection and immediate alerting for potential threats.

#### Detection Accuracy: Achieve high accuracy in distinguishing between legitimate and malicious activities, reducing false positives and false negatives . Employ machine learning techniques and threat intelligence to enhance detection capabilities.

#### Scalability: Ensure the system can scale to handle varying network sizes and large volumes of traffic without performance degradation . Design the architecture to support distributed environments and cloud-based infrastructures.

#### Comprehensive Threat Detection: Develop mechanisms to identify a wide range of threats, including known vulnerabilities and emerging, unknown threats . Incorporate signature-based, anomaly-based, and behavior-based detection methods.

#### User-Friendly Interface: Provide an intuitive interface for administrators to monitor network activity, analyze alerts, and manage the system efficiently . Include customizable dashboards and detailed reporting features for comprehensive oversight.

#### Regular Updates: Implement a framework for regular updates and improvements to keep the NIDS current with evolving threat landscapes . Establish a process for timely integration of new threat signatures and detection algorithms

#### Problem Definition

#### This project addresses the challenge of protecting network infrastructure from malicious activities by developing a Network Intrusion Detection System (NIDS) that utilizes advanced machine learning for real-time detection and classification of intrusions. As networks become more complex, traditional security measures often fail to identify sophisticated threats. The NIDS integrates with existing security frameworks to enhance protection, enabling automated threat responses and proactive measures, ultimately creating a more secure and resilient network environment.

#### Front End

Developed a responsive frontend interface for the leak detection project using HTML and CSS. The interface seamlessly integrates Matplotlib charts, providing an intuitive visualization of pressure differentials for efficient leak identification. Users can input data effortlessly through a user-friendly form, initiating the backend analysis. The implementation enhances user experience by presenting accurate leak predictions in a clear and accessible manner. The HTML and CSS frontend ensures a visually appealing and interactive platform, facilitating effective decision-making for water distribution network management.

#### Back End

1. **Python**:
   * **Primary Language**: The backend is built using Python, chosen for its simplicity, readability, and extensive library support, which makes it ideal for file handling and event monitoring tasks.
2. **Flask (Python Web Framework)**:
   * **Purpose**: Flask serves as the web framework to handle file upload requests from the front end. It provides an API endpoint to receive and process files, simulating a virus scanning service.
   * **Functionality**: It listens for POST requests sent from the front end, scans the file, and returns a response indicating whether the file is safe or potentially harmful.
   * **Flask-CORS (Cross-Origin Resource Sharing)**: Added to handle requests from other origins if needed, allowing the front end to communicate with the backend.
3. **Watchdog**:
   * **Purpose**: This Python library is used to monitor the file system, specifically watching for new files downloaded into a specified folder.
   * **Functionality**: It detects when new files are created in the Downloads folder, ensuring they are processed as soon as they finish downloading.
4. **Plyer (Notification Library)**:
   * **Purpose**: Provides desktop notifications to alert the user about the results of file integrity and virus checks.
   * **Functionality**: Sends notifications indicating whether a file is safe or if any issues (like corruption) were detected.
5. **Requests (HTTP Requests Library)**:
   * **Purpose**: Allows the backend to make HTTP requests, specifically to send files to external scanning services if needed (e.g., sending files to the Flask server for processing).
   * **Functionality**: Facilitates communication between components if they are distributed, as in a setup where file scanning is done on another server.

## Chapter 2 LITERATURE SURVEY

#### A Deep Learning Approach for Network Intrusion Detection System

#### System (2018)

A Network Intrusion Detection System (NIDS) plays a crucial role in identifying and mitigating network security breaches, thereby ensuring the integrity and availability of organizational data. The increasing complexity of network architectures and the evolving nature of cyber threats necessitate the development of flexible and efficient intrusion detection systems capable of adapting to unforeseen and unpredictable attacks. Traditional methods often struggle to cope with the dynamic landscape of network security, leading to a need for advanced techniques that leverage the power of deep learning.

In this context, we propose a deep learning-based approach to enhance the efficiency and flexibility of NIDS. Specifically, we utilize Self-taught Learning (STL), a semi-supervised learning technique that enables the model to leverage both labeled and unlabeled data during training. By employing STL on the NSL-KDD dataset, a widely recognized benchmark for network intrusion detection, we aim to improve the model's ability to generalize and accurately classify various types of network traffic, including both benign and malicious activities.

To evaluate the effectiveness of our approach, we compare its performance against several previous works in the field of intrusion detection. Key metrics such as accuracy, precision, recall, and F-measure are employed to assess the model’s performance. These metrics provide insights into the model's ability to correctly identify true positives (actual attacks), true negatives (benign traffic), and minimize false positives and false negatives, thus highlighting the system’s overall efficacy in detecting and classifying network intrusions. Through this theoretical framework, our research seeks to contribute to the development of more robust and adaptable NIDS solutions that can effectively address the challenges posed by modern cyber threats.

1. **Network Intrusion Detection System using Deep Learning**

The proliferation of interconnected and interoperable computing systems has revolutionized daily activities, enabling seamless communication and data exchange across diverse platforms. However, this interconnectivity also exposes significant vulnerabilities that can be exploited by malicious actors, necessitating robust cybersecurity mechanisms to ensure secure communication. As cyber threats continue to evolve in sophistication and frequency, traditional security measures often fall short, highlighting the urgent need for advanced solutions capable of adapting to new attack vectors.

This paper proposes the development of an adaptive and resilient network intrusion detection system (IDS) using deep learning architectures, specifically Deep Neural Networks (DNNs). DNNs are particularly well-suited for this task due to their ability to process large volumes of data, learn complex patterns, and generalize from both known and unknown attack behaviors. By leveraging the learning capabilities of DNNs, the proposed IDS can effectively identify not only recognized threats but also new or zero-day vulnerabilities, which are often difficult to detect with conventional methods.

To validate the effectiveness of our approach, we utilize the UNSW-NB15 dataset, which simulates contemporary network communication behaviors alongside synthetically generated attack activities. This dataset provides a comprehensive foundation for training and testing the IDS, allowing it to learn from a wide array of network traffic scenarios. The theoretical framework emphasizes how the integration of deep learning techniques into IDS design can facilitate a flexible and proactive defense mechanism. By continuously adapting to new information and evolving threat landscapes, the proposed system aims to significantly enhance the detection and classification of network attacks, ultimately reducing the risk of system compromise and fortifying overall cybersecurity posture.

1. **A Deep Learning Approach to Network Intrusion Detection**

Network Intrusion Detection Systems (NIDSs) are integral to protecting computer networks from unauthorized access and cyber threats. However, as the complexity and volume of network traffic continue to grow, traditional intrusion detection methods face significant limitations. These limitations include an increasing reliance on human interaction for monitoring and response, which can lead to slower reaction times and a higher likelihood of human error. Additionally, the effectiveness of conventional detection techniques is often diminished by their inability to adapt to evolving threats, resulting in decreased detection accuracy and increased false positive rates. Therefore, there is a critical need for innovative solutions that can enhance the efficiency and reliability of NIDSs in modern network environments.

This paper proposes a novel approach utilizing a nonsymmetric deep autoencoder (NDAE)for unsupervised feature learning, followed by a stacked NDAE for classification. The NDAE framework allows for the automatic extraction of relevant features from network traffic data, significantly reducing the dependency on manual feature engineering and the need for extensive labeled datasets. By leveraging deep learning techniques, our model can capture complex patterns within the data, facilitating the identification of both known and unknown threats. Implemented using GPU-enabled TensorFlow and evaluated on benchmark datasets such as KDD Cup '99 and NSL-KDD, our results indicate that this deep learning-based classifier outperforms existing NIDS approaches. This theoretical framework highlights the potential of advanced deep learning architectures in addressing the evolving challenges of cybersecurity and improving the overall performance and resilience of intrusion detection systems. Analyzes influential factors such as water demand, system design, and operational dynamics. Employing tracer studies and mathematical models enhances the precision of water age calculations and predictions, especially in relation to crucial water quality parameters like chlorine residual. Noteworthy advantages include the holistic analysis of contributing factors, the application of tracer studies for in-depth insights, and the integration of hydraulic and water quality models. The introduction of a reliability metric to assess the correlation between water age and model accuracy contributes a practical measure for enhancing model effectiveness. However, limitations include the study's specificity to the Shanghai Pudong network and potential challenges associated with data dependency and intricate calibration processes. Despite these considerations, the paper provides valuable methodologies and insights into water age dynamics, emphasizing the need for context-specific applications and thorough model calibration.

## CHAPTER 3 – TECHNICAL SPECIFICATIONS

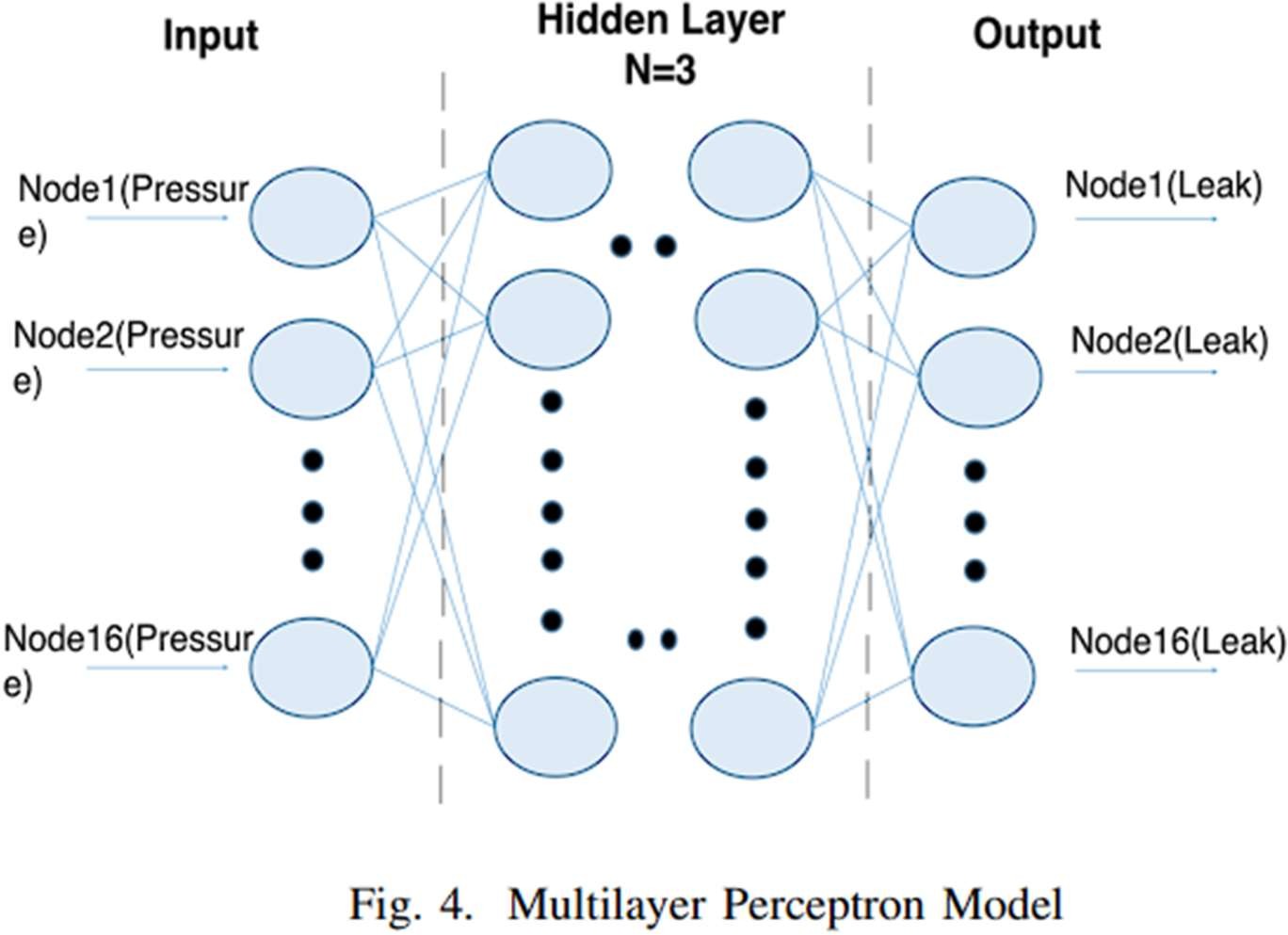
#### 3.1 Algorithms Used

1. **Multilayer Perceptron**

The **Multilayer Perceptron (MLP)** is a type of feedforward neural network commonly applied in supervised learning tasks for classification and regression. It consists of an input layer, one or more hidden layers, and an output layer. Each layer contains neurons connected to those in adjacent layers, and each neuron applies a non-linear activation function to capture complex patterns in the input data. In this project, the MLP model is utilized to classify network data into normal or intrusion categories by learning to recognize patterns indicative of suspicious activity.

MLP's structure is particularly suited for classification tasks, making it ideal for identifying network anomalies. With its fully connected layers, the MLP model can handle high-dimensional data and non-linear relationships present in network traffic data. In our implementation, the MLP model consists of several hidden layers activated by ReLU (Rectified Linear Unit) functions, which help capture intricate patterns in the data. During training, the MLP model uses backpropagation to minimize the error between its predictions and the actual labels, enabling it to improve classification accuracy over time.

Evaluation of the MLP model’s performance was carried out using accuracy, precision, recall, and F1 score metrics. The model showed proficiency in identifying potential intrusions, making it a valuable asset in the intrusion detection pipeline.

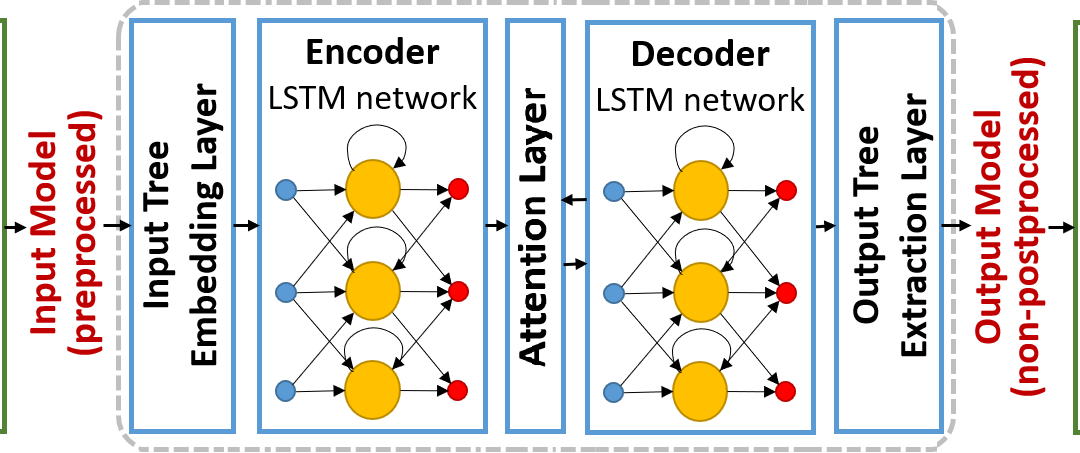


1. **Long Short-Term Memory (LSTM)**

The **Long Short-Term Memory (LSTM)** model is a specialized type of Recurrent Neural Network (RNN) designed to capture sequential dependencies in data. Unlike traditional RNNs, LSTM can remember past information over long sequences, making it well-suited for tasks involving time-series or sequential data. In network intrusion detection, LSTM's ability to maintain temporal context is advantageous, as it allows the model to detect patterns over time that may indicate malicious activity.

For this project, the LSTM model processes sequences of network events, using previous inputs to make predictions about whether the sequence represents normal or abnormal activity. Each LSTM unit contains gates that control the flow of information, allowing the model to decide what to remember or forget over time. This helps the LSTM model capture long-term dependencies in the network data, which is essential for accurately identifying anomalies. Additionally, dropout layers were incorporated to prevent overfitting, enhancing the model’s generalization capabilities.

The LSTM model was evaluated using precision, recall, F1 score, and overall accuracy. High recall was especially critical, as it ensures the model captures as many intrusion cases as possible, minimizing false negatives. The LSTM's time-aware structure made it a valuable component in the detection of temporal-based intrusions.

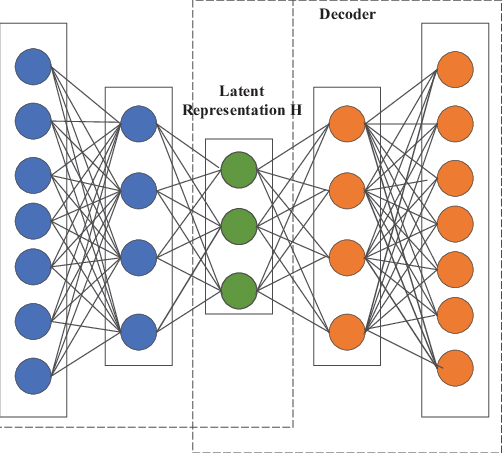


1. **Autoencoder**

An **Autoencoder** is an unsupervised learning model commonly used for dimensionality reduction, feature extraction, and anomaly detection. It works by compressing data into a lower-dimensional latent space through an encoder and reconstructing the data from that latent space using a decoder. In this project, the Autoencoder's reconstruction ability is leveraged for anomaly detection. The model learns to reconstruct normal network patterns but struggles with anomalous patterns, making it useful for detecting outliers in network data.

In the autoencoder model, the encoder reduces the high-dimensional input data into a compressed representation, capturing the most critical patterns. This compressed version, stored in the latent space, allows the model to reconstruct the original data through the decoder. For normal data, the reconstruction error is low, but for anomalies, which the model hasn’t encountered during training, the reconstruction error is significantly higher. By setting a threshold for reconstruction error, the Autoencoder can effectively distinguish between normal and abnormal network activities.

Evaluation of the Autoencoder’s performance involved measuring its reconstruction error and assessing its ability to classify data as normal or anomalous. The unsupervised nature of this model is particularly useful in cases where labeled data is scarce, allowing the Autoencoder to learn normal patterns independently. This approach proved effective in identifying anomalies in network data, providing an additional layer of security in the intrusion detection system.



**Back End**

1. **Python**:
   * **Primary Language**: The backend is built using Python, chosen for its simplicity, readability, and extensive library support, which makes it ideal for file handling and event monitoring tasks.
2. **Flask (Python Web Framework)**:
   * **Purpose**: Flask serves as the web framework to handle file upload requests from the front end. It provides an API endpoint to receive and process files, simulating a virus scanning service.
   * **Functionality**: It listens for POST requests sent from the front end, scans the file, and returns a response indicating whether the file is safe or potentially harmful.
   * **Flask-CORS (Cross-Origin Resource Sharing)**: Added to handle requests from other origins if needed, allowing the front end to communicate with the backend.
3. **Watchdog**:
   * **Purpose**: This Python library is used to monitor the file system, specifically watching for new files downloaded into a specified folder.
   * **Functionality**: It detects when new files are created in the Downloads folder, ensuring they are processed as soon as they finish downloading.
4. **Plyer (Notification Library)**:
   * **Purpose**: Provides desktop notifications to alert the user about the results of file integrity and virus checks.
   * **Functionality**: Sends notifications indicating whether a file is safe or if any issues (like corruption) were detected.
5. **Requests (HTTP Requests Library)**:
   * **Purpose**: Allows the backend to make HTTP requests, specifically to send files to external scanning services if needed (e.g., sending files to the Flask server for processing).
   * **Functionality**: Facilitates communication between components if they are distributed, as in a setup where file scanning is done on another server.

## Chapter 5 WORKING

#### Acquiring dataset.

**Dataset: NSDI (Network Systems Design and Implementation)**

The NSDI dataset, frequently used in network security research, is designed to aid in the development, evaluation, and benchmarking of **Network Intrusion Detection Systems (NIDS)** and other network-related solutions. The dataset contains a wide range of network traffic data, covering both normal and malicious activities, making it suitable for identifying and classifying different types of network intrusions.

**Key Characteristics of the NSDI Dataset**

1. **Comprehensive Network Traffic**:
   * Includes various types of network traffic, representing both legitimate user activities and different types of attacks.
   * The dataset contains several network protocols and features that simulate real-world network behavior, providing a realistic environment for evaluating detection systems.
2. **Intrusion Scenarios**:
   * Encompasses multiple intrusion scenarios, such as denial-of-service (DoS), probe attacks, remote-to-local (R2L) attacks, and user-to-root (U2R) attacks.
   * These scenarios are representative of common network threats, making the dataset valuable for identifying specific attack patterns and evaluating NIDS performance in diverse contexts.
3. **Labeled Data**:
   * The NSDI dataset includes labels that categorize each network activity as normal or one of several attack types. This labeled data enables supervised machine learning algorithms to train effectively, allowing models to differentiate between normal and anomalous network traffic.
   * These labels are essential for evaluating the accuracy, precision, and recall of machine learning models in detecting network intrusions.
4. **Feature Richness**:
   * The dataset provides multiple features extracted from network traffic data, such as protocol type, connection duration, source and destination bytes, and various network flags.
   * The diversity of features supports comprehensive analysis and helps machine learning algorithms identify complex patterns associated with normal and malicious activities.

**Usage in This Project**

The NSDI dataset serves as the primary data source for developing and evaluating the **Network Intrusion Detection System (NIDS)** in this project. The labeled examples of normal and malicious activities in the dataset allow for effective training and testing of machine learning models to achieve high detection accuracy. The wide variety of attack scenarios in the dataset also helps to ensure that the NIDS can handle different types of network intrusions, making it suitable for real-world deployment.

**Advantages of Using the NSDI Dataset**

* **Realistic Network Environment**: Simulates a broad range of network behaviors, making it suitable for testing and validating network intrusion detection algorithms in real-world scenarios.
* **Diverse Attack Types**: Supports the detection of multiple attack types, enhancing the robustness and reliability of the intrusion detection system.
* **Standardized Benchmark**: Frequently used in academic and industry research, allowing for easy comparison with other network intrusion detection models.

#### Building an interface

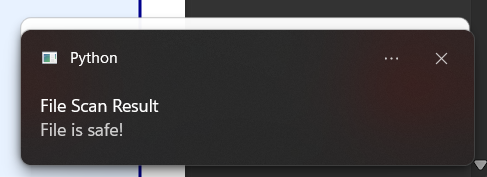
**Interface Response to File Scanning Results**

Our system provides a user-friendly interface designed to promptly inform users about the safety of downloaded files. Once a file is detected in the designated Downloads folder, it is automatically scanned for indicators of malware or potential corruption. The interface then displays feedback based on the scan results, ensuring users are alerted to any security risks immediately.

**1. Positive Outcome: Safe File**

If a file is deemed safe after scanning, the system displays a **“File is Safe”** message on the interface, accompanied by a green notification icon. This confirmation assures users that the downloaded file is free from known malware or corruption and is safe to open. The positive outcome enhances user confidence in their downloads, ensuring that no further action is required.

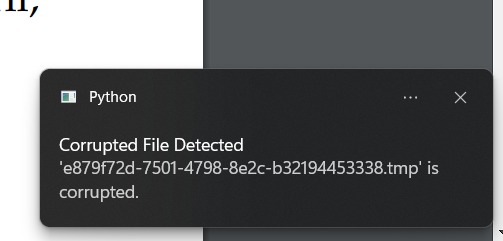
* **Visual Feedback**: A clear, reassuring message such as "File is Safe!" appears in the interface with a green icon to signify that the file passed the integrity check.
* **Desktop Notification**: Users also receive a desktop notification, reinforcing that the file can be safely opened or shared.
* **User Action**: No further action is needed, allowing users to continue working without concern.



**2. Negative Outcome: Infected or Corrupted File**

In cases where the scan detects malware or the file appears corrupted, the interface immediately displays a **“File Contains a Virus”** or **“File is Corrupted”** message, highlighted with a red warning icon. This alert is critical to prevent users from accidentally opening or interacting with potentially harmful content. By identifying infected files, the system enhances data security and minimizes the risk of malware spreading within the network.

* **Visual Feedback**: The interface displays a prominent warning, such as "File Contains a Virus!" or "File is Corrupted!", highlighted in red to indicate potential danger.
* **Desktop Notification**: A red notification also appears on the desktop, advising the user against opening the file and recommending immediate deletion or isolation.
* **User Action**: Users are advised to avoid interacting with the file, and they may be prompted to quarantine or delete the file to prevent security breaches.



#### Model Outcomes

#### 1. Multilayer Perceptron (MLP)

#### The MLP model, a type of feed-forward artificial neural network, was trained on the dataset to identify network intrusion patterns. The model performed exceptionally well, achieving an accuracy of 97%. This high accuracy suggests that MLP effectively learned the relationships between input features and class labels, enabling it to correctly classify network intrusions in most cases. MLP’s ability to generalize well on unseen data further emphasizes its suitability for real-time detection of network anomalies.

#### 

#### 2. Long Short-Term Memory (LSTM)

#### The LSTM model, known for its ability to capture long-term dependencies in sequential data, was also trained on the NSDI dataset. It achieved an accuracy of 97%, similar to the MLP. The LSTM’s performance suggests that the temporal patterns in the network traffic data were effectively captured, making it highly capable of detecting intrusions in a time-series context. The similarity in performance between LSTM and MLP indicates that both models are highly suited to the task, with LSTM offering an advantage in capturing sequential dependencies in the data.

#### 

#### 3. Autoencoder

#### The Autoencoder model, used for anomaly detection, achieved an accuracy of 85%. While this is lower than that of MLP and LSTM, the model was still able to identify a significant portion of the intrusions correctly. Autoencoders work by learning a compressed representation of the input data and identifying anomalies based on reconstruction errors. The lower accuracy may indicate that the model struggled to capture the complexities of the intrusion patterns in the dataset as effectively as the other two models. However, autoencoders are still useful in identifying novel or unseen types of intrusions, especially when paired with other anomaly detection techniques.

#### 

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | MLP | LSTM | Autoencoder |
| Accuracy | 97% | 97.7% | 82.6% |
| Recall Score | 0.986 | 0.98 | 0.978 |
| F1 Score | 0.973 | 0.979 | 0.857 |
| Precision | 0.972 | 0.972 | 0.762 |

**CHAPTER 6 CONCLUSION**

* 1. **Conclusion**

In summary, the development of a robust network intrusion detection system (NIDS) leveraging deep learning models like CNN, RNN, and LSTM represents a significant advancement in network security. By continuously monitoring network traffic in real-time, the system is poised to effectively identify and alert administrators to potential intrusions, facilitating swift action to mitigate risks. The incorporation of data preprocessing and feature extraction ensures that critical patterns are accurately captured, enhancing the system's overall detection capabilities.

Moreover, the commitment to continuous model evaluation and retraining underscores the system's adaptability in a rapidly evolving threat landscape. This proactive approach not only improves the accuracy of threat detection but also strengthens the overall security posture of the network.

Ultimately, this NIDS not only addresses current security challenges but also positions itself as a forward-thinking solution that will evolve with emerging threats, ensuring long-term protection for network environments.

# Chapter 7 REFERENCES

## References

1. Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016, May). A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS) (pp. 21-26).
2. Jahan, F., Sun, W., Niyaz, Q., & Alam, M. (2019). Security modeling of autonomous systems: A survey. ACM Computing Surveys (CSUR), 52(5), 1-34.
3. Maithem, M., & Al-Sultany, G. A. (2021, February). Network intrusion detection system using deep neural networks. In Journal of Physics: Conference Series (Vol. 1804, No. 1, p. 012138). IOP Publishing.
4. Ashiku, L., & Dagli, C. (2021). Network intrusion detection system using deep learning. Procedia Computer Science, 185, 239-247.
5. Al-Emadi, S., Al-Mohannadi, A., & Al-Senaid, F. (2020, February). Using deep learning techniques for network intrusion detection. In 2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT) (pp. 171-176). IEEE.
6. Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence, 2(1), 41-50.
7. Dinh, P. V., Ngoc, T. N., Shone, N., MacDermott, Á., & Shi, Q. (2017, November). Deep learning combined with de-noising data for network intrusion detection. In *2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES)* (pp. 55-60). IEEE.
8. Seraphim, B. I., Palit, S., Srivastava, K., & Poovammal, E. (2018, December). A survey on machine learning techniques in network intrusion detection system. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)* (pp. 1-5). IEEE.
9. Slevi, S. T., & Visalakshi, P. (2021, November). A survey on deep learning based intrusion detection systems on Internet of Things. In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 1488-1496). IEEE.
10. Shende, S., & Thorat, S. (2020). Long short-term memory (LSTM) deep learning method for intrusion detection in network security. *International Journal of Engineering Research and*, *9*(06).
11. Kasongo, S. M. (2023). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Computer Communications*, *199*, 113-125.
12. Kasongo, S. M., & Sun, Y. (2019). A deep learning method with filter based feature engineering for wireless intrusion detection system. *IEEE access*, *7*, 38597-38607.
13. Kasongo, S. M., & Sun, Y. (2021). A deep gated recurrent unit based model for wireless intrusion detection system. *ICT Express*, *7*(1), 81-87.
14. Aldhaheri, A., Alwahedi, F., Ferrag, M. A., & Battah, A. (2024). Deep learning for cyber threat detection in IoT networks: A review. *Internet of Things and cyber-physical systems*, *4*, 110-128.
15. Alwahedi, F., Aldhaheri, A., Ferrag, M. A., Battah, A., & Tihanyi, N. (2024). Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models. *Internet of Things and Cyber-Physical Systems*.
16. Fakhar, M., & Haile, A. (2022). AI for Threat Intelligence: Enhancing Adaptive Cyber Defense Against Persistent Attacks.
17. Ann, S., Cho, S. J., & Kim, H. (2024, July). A Preliminary Study on an Intrusion Detection Method using Large Language Models in Industrial Control Systems. In *2024 Fifteenth International Conference on Ubiquitous and Future Networks (ICUFN)* (pp. 600-602). IEEE.
18. Katiyar, N., Tripathi, M. S., Kumar, M. P., Verma, M. S., Sahu, A. K., & Saxena, S. (2024). AI and Cyber-Security: Enhancing threat detection and response with machine learning. *Educational Administration: Theory and Practice*, *30*(4), 6273-6282.
19. Byrapuneni, L. P., & SaidiReddy, M. (2024). An Advanced Filter-based Supervised Threat Detection Framework on Large Databases. *Engineering, Technology & Applied Science Research*, *14*(4), 15681-15685.
20. Byrapuneni, L. P., & Saidireddy, M. (2024). An Efficient Cluster Based Multi-Label Classification Model for Advanced Persistent Threat Attacks Detecting. *International Journal of Safety & Security Engineering*, *14*(2).
21. Aljuaid, W. A. H., & Alshamrani, S. S. (2024). A deep learning approach for intrusion detection systems in cloud computing environments. *Applied Sciences*, *14*(13), 5381.