**NETWORK ANOMALY DETECTION**

**A PROJECT REPORT**

*Submitted by*

Batch B Team – 09

ANIRUDH S VARRIER - CB.SC.U4AIE23113

B V DHARSINI SRI – CB.SC.U4AIE23115

CHANJHANA ELANGO – CB.SC.U4AIE23120

KANISKA M – CB.SC.U4AIE23136

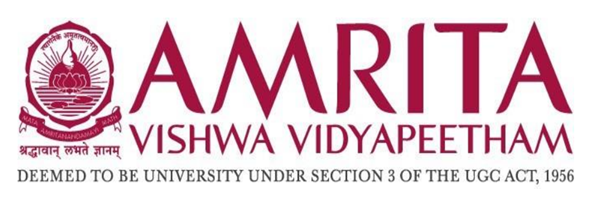
**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE WITH ARTFICIAL INTELLIGENCE**

*As a part of the subject*

**22MAT121- DISCRETE MATHEMATICS**



**Department of Artificial Intelligence**

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112 (INDIA)

**December 2023**

**CONTENTS**

**Acknowledgement……………………………………………………………………….04**

**List of Figures……………………………………………………………………………05**

**Abstract…………………………………………………………………………………..06**

1. **Introduction…………………………………………………………………………07**
   1. What is a Network?
      1. What Data is sent across networks?
      2. How is Data sent across networks?
   2. What is cyber security?
      1. Pervasiveness of cyber-threats
      2. Protection of sensitive information
      3. Preserving trust and confidence
2. **Aim…………………………………………………………………………………...09**
3. **Methodology………………………………………………………………………...10**
4. **Code Implementation and explanation……………………………………………11**
   1. Code implementation - python
      1. Source Code
      2. Code explanation – python
      3. Result of python code
   2. Code implementation - matlab
      1. Source Code
      2. Code explanation – python
      3. Result of matlab code
5. **Scope and Future Possibilities……………………………………………………...16**
6. **Conclusion…………………………………………………………………………...17**
7. **References…………………………………………………………………………...18**

**AKNOWLEDGMENT**

We would like to extend our heartfelt gratitude to **Ms. Ambika P S (**Assistant Prof, Amrita Vishwa Vidhyapeetham) for her invaluable guidance and support throughout the development of our Discrete Mathematics Project: **Network anomaly Detection**. Her insights and mentorship have been instrumental in shaping this project and enhancing our understanding of the subject. We would like to thank **Mr.Sunil** for his tireless support in making the completion of this project possible. Additionally, we would like to extend our heartfelt thanks to **Prof. K.P. Soman (Head of AI, Amrita Vishwa Vidhyapeetham)**. The academic environment created by him has continually inspired us to strive for excellence. We are deeply thankful for the wisdom, guidance, and inspiration.

**LIST OF FIGURES**

Fig 1. Image represents flow chart for project methodology

Fig 2. Result for python code

Fig 3. Result for matlab code

**ABSTRACT**

This project investigates the efficacy of using machine learning techniques, specifically Support Vector Machines (SVM) and the pseudoinverse in the context of network anomaly detection. Leveraging the diverse UNSW-NB15 dataset, we explore the methods SVM and pseudoinverse to enhance the accuracy of identifying anomalous network behaviors. Our study encompasses implementation, performance evaluation, and the comparative analysis of the proposed approach against traditional methods. The results showcase the robustness and adaptability of the SVM and pseudoinverse technique, suggesting its effectiveness in bolstering cybersecurity measures in complex network environments.

**1.0 INTRODUCTION**

**1.1 What is a network?**

A network, in the context of computing, is a group of interconnected computers, servers, or devices that share resources and data. These networks can be small, such as a home network with a few devices, or vast, like the internet, which connects computers worldwide. Networks facilitate communication, collaboration, and resource sharing, making them integral to modern digital infrastructure.

**1.11 What data is sent across networks?**

Networks transmit a wide variety of data types. This can range from simple text messages in a chat application, to complex data structures in a distributed database system. Here are some common types of data sent across networks:

1. **Textual Data**: This includes emails, instant messages, documents, and web pages.
2. **Multimedia Data**: This encompasses images, audio, and video files. Streaming services like Netflix or Spotify rely on network data transmission to deliver content to users.
3. **Application Data**: Many software applications rely on networks to function. This could include cloud-based applications like Google Docs, multiplayer online games, or collaborative coding platforms like GitHub.
4. **System Data**: Networks also transmit data necessary for their own operation, such as IP addresses, routing tables, and network health metrics.

**1.12 How is data sent across networks?**

Data transmission across networks involves several steps and protocols.

1. **Data Segmentation**: The data to be sent is broken down into smaller pieces called packets. This makes the data easier to manage and transmit.
2. **Packetization**: Each packet is then encapsulated with additional information such as the sender’s address, the receiver’s address, and the sequence number of the packet. This process is governed by protocols like the Internet Protocol (IP).
3. **Transmission**: The packets are then sent over the network. Depending on the network’s architecture, this could involve several intermediate nodes (like routers or switches) that help guide the packets to their destination.
4. **Reassembly**: Once all the packets reach their destination, they are reassembled to form the original data.

This process is facilitated by various protocols that make up the TCP/IP model, which is the foundation of modern network communication. These protocols ensure that data is transmitted reliably, securely, and efficiently across networks.

**1.2 What is Cyber-security?**

Cybersecurity refers to the practice of protecting computers, servers, mobile devices, electronic systems, networks, and data from digital attacks, damage, or unauthorized access. It encompasses a broad range of concepts, tools, and methodologies designed to safeguard digital environments.

**1.21 Pervasiveness of Cyber Threats**

In today’s interconnected world, cyber threats have become increasingly pervasive and sophisticated. These threats can come in various forms, such as malware, ransomware, phishing, and denial-of-service attacks. They can disrupt businesses, cause financial losses, and even pose threats to national security. Cybersecurity measures are needed to detect, prevent, and respond to these threats, ensuring the integrity and availability of digital systems and data.

**1.22 Protection of Sensitive Information**

Sensitive information, such as personal data, intellectual property, and proprietary business information, is often stored and transmitted digitally. Unauthorized access to this information can lead to serious consequences, including identity theft, financial fraud, and loss of competitive advantage. Cybersecurity helps protect sensitive information by implementing various controls, such as encryption, access controls, and intrusion detection systems.

**1.23 Preserving Trust and Confidence**

Trust and confidence are fundamental to digital interactions. Users need to trust that their information is secure and that the digital services they use are reliable. Cybersecurity helps preserve trust and confidence by ensuring that digital environments are secure and resilient. This is particularly important for businesses that rely on digital channels to interact with customers, partners, and employees

**2.0 AIM**

The primary objective of a network anomaly analysis project is to identify and investigate unusual patterns or behaviors in network traffic that significantly deviate from what is considered normal. These deviations, often referred to as anomalies, can be indicative of a variety of potential issues. These issues may range from simple system faults or network misconfigurations to more severe and malicious activities such as cyber-attacks or network intrusions.

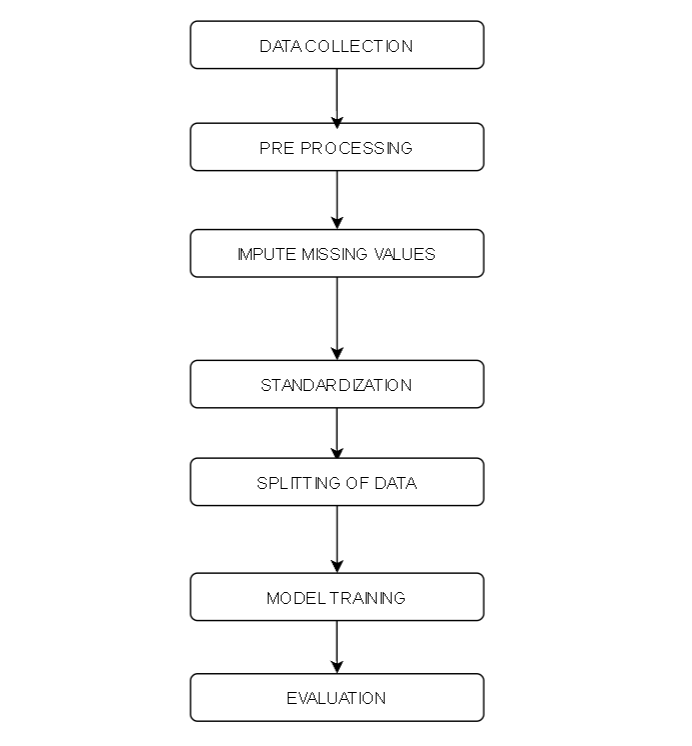
Anomalies can manifest in many forms, such as sudden spikes in network traffic, unexpected system reboots, or unusual login attempts. They can occur at any point within the network, affecting various components such as user devices, routers, switches, and servers. The data affected by these anomalies could be part of the network traffic itself or data stored on networked devices.

By successfully identifying these anomalies, organizations can take a proactive stance towards potential threats. This allows them to enhance their network security measures, ensuring the integrity and availability of their systems. It also plays a crucial role in protecting sensitive information, which could include personal data, intellectual property, or proprietary business information.

Moreover, a network anomaly analysis project is not just about threat detection and prevention. It can also provide valuable insights for improving network performance and efficiency. By understanding the normal behavior of the network and its components, organizations can optimize their network configurations, manage their network resources more effectively, and improve the overall performance of their network.

**3.0 METHODOLOGY**

The methodology involves several steps: data preprocessing, feature selection, model training, and evaluation. The data preprocessing step involves cleaning the data and handling missing values. Feature selection is done to reduce the dimensionality of the dataset and improve the model’s performance. The model is then trained on the preprocessed data and evaluated using various metrics.



**Fig 1: Image represents flow chart for project methodology**

**4.0 CODE IMPLEMENTATION AND EXPLANATION**

**4.1 SOURCE CODE**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

from sklearn.svm import SVC

from sklearn.impute import SimpleImputer

from sklearn.metrics import classification\_report, confusion\_matrix

from sklearn.model\_selection import cross\_val\_score

# Load training data

train\_data = pd.read\_csv('/UNSW\_NB15\_training-set (1).csv')

# Assuming 'label' column contains the labels (0 for normal, 1 for anomaly)

train\_data = train\_data.dropna(subset=['label'])  # Drop rows with NaN in the target variable

X\_train = train\_data.drop('label', axis=1)

y\_train = train\_data['label']

# Load testing data

test\_data = pd.read\_csv('/UNSW\_NB15\_testing-set(1).csv')

# Assuming 'label' column contains the labels (0 for normal, 1 for anomaly)

test\_data = test\_data.dropna(subset=['label'])  # Drop rows with NaN in the target variable

X\_test = test\_data.drop('label', axis=1)

y\_test = test\_data['label']

y\_test.head()

# Impute missing values (NaNs) using the mean strategy

imputer = SimpleImputer(strategy='mean')

X\_train = imputer.fit\_transform(X\_train)

X\_test = imputer.transform(X\_test)

# Standardize features by removing the mean and scaling to unit variance

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Create an SVM model

svm\_model = SVC(kernel='linear', C=1.0)

# Train the model

svm\_model.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = svm\_model.predict(X\_test)

# Evaluate the model

conf\_matrix= confusion\_matrix(y\_test, y\_pred)

print("Confusion Matrix:\n", conf\_matrix)

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues',

            xticklabels=['Normal', 'Anomaly'], yticklabels=['Normal', 'Anomaly'])

plt.title('Confusion Matrix')

plt.xlabel('Predicted Label')

plt.ylabel('True Label')

plt.show()

# Cross-validation scores

cv\_scores = cross\_val\_score(svm\_model, X\_train, y\_train, cv=5)

print(f"Cross-Validation Scores: {cv\_scores}")

print(f"Mean CV Score: {np.mean(cv\_scores)}")

**4.2 CODE EXPLANATION - PYTHON**

The training data is loaded from a CSV file into a pandas DataFrame. It’s assumed that the ‘label’ column contains the labels (0 for normal, 1 for anomaly). Any rows with NaN in the target variable are dropped. The features (X\_train) and labels (y\_train) for the training set are then separated.

The same steps are repeated for the testing data to obtain the features (X\_test) and labels (y\_test) for the test set.

Next, any missing values (NaNs) in the features are imputed using the mean strategy. This is done using the SimpleImputer from sklearn, which replaces missing values with the mean value of each column.

The features are then standardized using the StandardScaler from sklearn. This removes the mean and scales features to unit variance, which is a common requirement for many machine learning estimators.

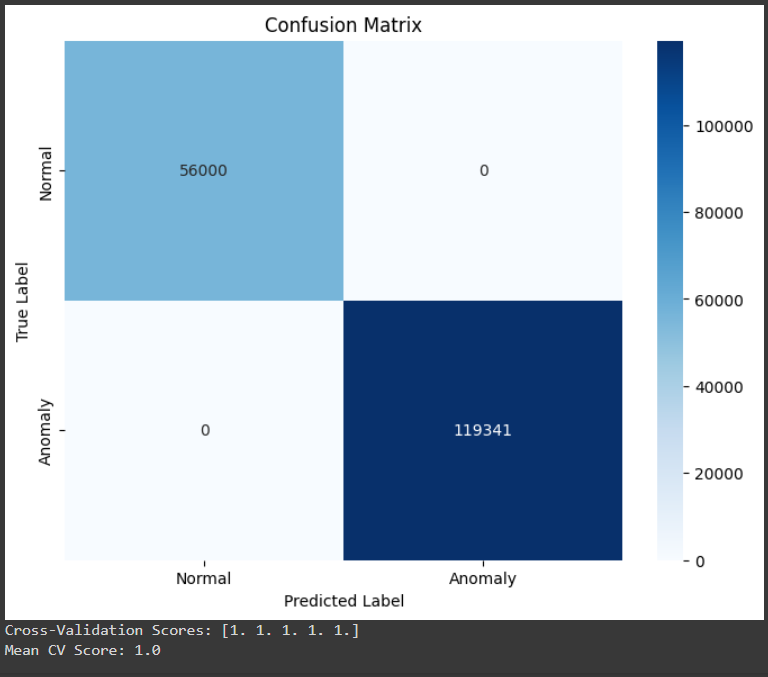
An SVM model is created with a linear kernel and a regularization parameter, C, set to 1.0. The model is then trained on the training data.

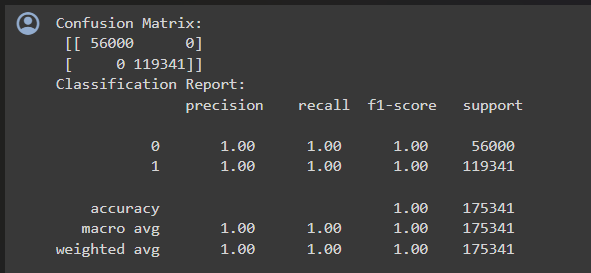
The trained model is used to make predictions on the test set. The predictions are compared with the true labels to evaluate the model’s performance. The confusion matrix and classification report are printed to provide detailed insights into the model’s performance.

A heatmap of the confusion matrix is also plotted using seaborn. This provides a visual representation of the model’s performance, showing the number of true positives, true negatives, false positives, and false negatives.

Cross-validation is performed on the training set to assess the model’s performance. The cross\_val\_score function from sklearn is used to calculate the cross-validation scores. The mean of these scores is printed to provide an overall measure of the model’s performance.

**4.2 RESULT OF PYTHON CODE**

****

****

**Fig 2. Results for python code**

**4.2 CODE IMPLEMENTATION – MATLAB**

**4.2.1 SOURCE CODE**

clear all

close all

clc

tic

fprintf('Classification using pseudoInverse\n');

fprintf('Data read : starting\n');

A1 = csvread("standardiseddata (1).csv"); % Training data

B1 = csvread("Bvalues (1).csv"); % Training - classes

A2 = csvread("standardiseddata (2).csv"); % Testing data

B2 = csvread("Bvalues (2).csv"); % testing data - used for checking accuracy

toc

fprintf('Data read : completed\n');

Btrain = [B1, ones(size(B1))-B1];

Btest = [B2, ones(size(B2))-B2]; % n x 2

rA1=rank(A1);

% rank is < 44,

[~,jb1]=rref(A1);

% col 27 is dependent on other cols

Ar1 = A1(:,[1:26,28:44]);

AtA = Ar1'\*Ar1;

psinA = inv(AtA)\*Ar1';

weights = psinA\*Btrain; % n x 2

fprintf('weights estimated...\n')

Ar2 = A2(:,[1:26,28:44]);

pred = Ar2\*weights; % n x 2

%pred(max())

% pred(pred<0.5)=0;

% pred(pred>0.5)=1;

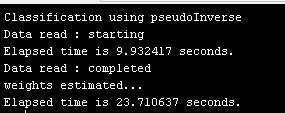
pred1=(pred(:,1) < pred(:,2));

toc

**4.2.2 CODE EXPLANATION – MATLAB**

The provided code is implementing a binary classification model using the pseudo-inverse method in MATLAB. It starts by reading training and testing data from CSV files. The class vectors are then converted into binary matrices. The rank of the training feature matrix A1 is calculated, and it’s found that one of the columns (column 27) is dependent on other columns. Therefore, this column is removed from the feature matrices (Ar1 and Ar2). The pseudo-inverse of the modified training feature matrix Ar1 is calculated. The weights for the binary classification model are estimated by multiplying the pseudo-inverse of Ar1 with Btrain. The weights are then used to make predictions on the testing data. The prediction is a matrix where each row has two elements representing the predicted probabilities for each class. Finally, a binary prediction vector pred1 is created by comparing the predicted probabilities for the two classes. The tic and toc commands are used to measure the execution time of the code. The fprintf commands are used to print status messages during the execution.

**4.2.3 RESULT FOR MATLAB CODE**



**Fig 3. Result for matlab code**

**5. SCOPE AND FUTURE POSSIBILITIES**

**Model Tuning**: Techniques such as hyperparameter optimization and kernel experimentation can refine model performance.

**Data Preprocessing**: Addressing data imbalance and outlier detection can improve the model's robustness.

**Model Evaluation**: Beyond k-fold cross-validation, diversifying evaluation metrics can offer a more holistic view of model efficacy.

**6. CONCLUSION**

In conclusion, employing machine learning for network anomaly detection proves instrumental in fortifying cybersecurity measures. By leveraging advanced algorithms to discern irregular patterns, organizations can proactively identify and respond to potential threats, ensuring the integrity and resilience of network infrastructures. This approach enhances overall security, providing a crucial defence against evolving cyber threats.

**7. REFERENCES**

**[7.1]** [Richa Singh](https://ieeexplore.ieee.org/author/37089503340); [Nidhi Srivastava](https://ieeexplore.ieee.org/author/37085621094); [Ashwani Kumar](https://ieeexplore.ieee.org/author/37405497200) - [Machine Learning Techniques for Anomaly Detection in Network Traffic | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/9702647)

**[7.2]** [Sohaila Eltanbouly](https://ieeexplore.ieee.org/author/37088397992); [May Bashendy](https://ieeexplore.ieee.org/author/37088397591); [Noora AlNaimi](https://ieeexplore.ieee.org/author/37088397747); [Zina Chkirbene](https://ieeexplore.ieee.org/author/37072673600); [Aiman Erbad](https://ieeexplore.ieee.org/author/37828634400) - [Machine Learning Techniques for Network Anomaly Detection: A Survey | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/9089465)

**[7.3]** [Song Wang](https://ieeexplore.ieee.org/author/37086252777); [Juan Fernando Balarezo](https://ieeexplore.ieee.org/author/37088758052); [Sithamparanathan Kandeepan](https://ieeexplore.ieee.org/author/37294706400); [Akram Al-Hourani](https://ieeexplore.ieee.org/author/37085343022) - [Machine Learning in Network Anomaly Detection: A Survey | IEEE Journals & Magazine | IEEE Xplore](https://ieeexplore.ieee.org/document/9610045)

**[7.4]** [Shan Suthaharan](https://link.springer.com/chapter/10.1007/978-1-4899-7641-3_9#auth-Shan-Suthaharan) - <https://link.springer.com/chapter/10.1007/978-1-4899-7641-3_9>

**[7.5]** Ilias Tougui, Abdelilah Jilbab, Jamal El Mhamdi - https://synapse.koreamed.org/articles/1147752