Enhancing Network Security Awareness: Visual Cyber Threat Analysis

Lokeshwar Reddy Nandanapalli, *2299460*, Ben Gideon Dokiburra, *2283917,* and Abhigna Sowgandhika Vadlamudi, *2268166*

[1]

***Abstract*—Cybersecurity professionals and researchers often deal with complex datasets related to network intrusions and security incidents. The goal of this project is to develop effective visualization techniques for the KDD Cup 1999 Data. These datasets contain information about network intrusions and normal activities, and visualization of this data helps obtain valuable insights about network intrusions. We use Principal Component Analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), and visualizing clusters of attack types using k-means, bar graphs, and pie charts, etc to show the number of attacks of each type visually. This data is very useful for obtaining valuable insights about intrusions since the attacks are visually represented for analysis. A standalone application is developed using QT Framework to interact with the visualization.**

Index Terms—Cybersecurity, Network Intrusions, Visualization, KDD Cup 1999 Data, Principal Component Analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), K-means, Bar Graphs, Pie Charts, Visualization Techniques, Attack Types, Cluster, Qt Framework

I. INTRODUCTION

The KDD Cup 1999 dataset serves as the foundation for the Third International Knowledge Discovery and Data Mining Tools Competition, held in conjunction with KDD-99, the Fifth International Conference on Knowledge Discovery and Data Mining. The primary objective of the competition was to develop a robust network intrusion detector, a predictive model proficient in distinguishing between "bad" connections, denoting intrusions or attacks, and "good" normal connections within a simulated military network environment. This extensive dataset includes a standardized set of auditable data, encompassing a diverse array of intrusions that mimic real-world scenarios. The dataset is available in various subsets, such as the complete dataset, a 10% subset, and unlabeled test data, each facilitating distinct analyses. Researchers and participants can explore the KDD cup names file for a comprehensive list of features, the training attack types file detailing intrusion types. This resource remains valuable for visualizing the attack types and understanding attacks in a network environment.

Preprocessing the KDD Cup 1999 data involves a crucial step to address the inherent class imbalance and manage the vast dataset effectively. In this context, a custom Python script named "sample.py" was employed for strategic data sampling. The script utilized random sampling techniques to generate a representative subset of the original dataset while maintaining the distribution of the target variable. By extracting the last column values and their respective counts, the script calculated the proportional sample sizes for each unique value. Subsequently, it randomly selected rows for each value, ensuring a balanced representation in the final sampled dataset. This preprocessing step not only aids in handling the scale of the original data but also contributes to the development of more efficient and accurate models by mitigating the challenges associated with class imbalance. The sampled dataset, stored in the "sample.csv" file, serves as an optimized foundation for subsequent analysis and visualization techniques aimed at enhancing the understanding of network intrusions and security incidents.

Visualization plays a pivotal role in comprehending complex datasets, especially in the realm of network security where distinguishing between normal activities and intrusions is paramount. In the context of the KDD Cup 1999 data, various visualization techniques have been employed to unravel patterns and insights within the dataset. Principal Component Analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) offer powerful tools for reducing dimensionality, allowing for the representation of intricate relationships among features. K-means clustering facilitates the identification of natural groupings within the data. Furthermore, bar graphs and pie charts provide intuitive visualizations of the distribution of attack types, offering a clear overview of the prevalence of different intrusion categories. To enhance the accessibility of these visualizations, a standalone application has been developed using the QT Framework. The QT-based application provides an interactive platform for users to explore and analyze the intricacies of the KDD Cup 1999 data, fostering a deeper understanding of network intrusions and supporting the development of robust cybersecurity solutions.

II. RELATED WORKS

[1] "UNSW-NB15 Computer Security Dataset: Analysis through Visualization" by Z. Zoghi and G. Serpen (2021):
This work explores the UNSW-NB15 Computer Security

[1]

Dataset and provides a detailed analysis through visualization techniques. The authors likely employed various visualization methods to gain insights into the dataset, similar to the approach taken in the current project. The paper could offer additional perspectives on visualizing cybersecurity datasets.

[2] "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" by T. Hastie, R. Tibshirani, and J. Friedman (2009):
While not specific to cybersecurity, this influential book covers various aspects of statistical learning and data mining. It may provide foundational knowledge on statistical techniques that can be applied to the analysis of cybersecurity datasets.

[3] "Python Data Science Handbook" by J. VanderPlas (2016):
This handbook is a comprehensive resource on data science using Python. It covers topics such as data manipulation, visualization, and machine learning. Cybersecurity professionals working with Python for data analysis may find this book valuable.

[4] "Scikit-learn: Machine Learning in Python" by F. Pedregosa et al. (2011):
Scikit-learn is a popular machine learning library in Python. The documentation and user guide provide insights into various machine learning algorithms, including clustering and dimensionality reduction. The authors of the current project likely used Scikit-learn for implementing some of their visualization techniques.

[5] "Representation Learning: A Review and New Perspectives" by Y. Bengio, A. Courville, and P. Vincent (2013):
This review paper explores representation learning, a crucial aspect in machine learning and data mining. Understanding how data is represented can be beneficial in cybersecurity analysis. The paper may offer insights into advanced techniques beyond traditional visualization. Link to the paper

III. METHODOLOGY AND IMPLEMENTATION

Visualizing complex datasets is fundamental in extracting meaningful insights, especially in the context of cybersecurity where the identification of network intrusions is paramount. The KDD Cup 1999 dataset, utilized for network intrusion detection, presents a rich but intricate collection of information. To unravel patterns and relationships within this vast dataset, an array of visualization techniques has been employed. From dimensionality reduction methods like PCA and t-SNE to clustering techniques and traditional visualizations like bar graphs and pie charts, each approach offers a unique perspective on the dataset. These visualizations not only enhance the interpretability of the data but also empower cybersecurity professionals and researchers to make informed decisions in fortifying network defenses. The subsequent paragraphs delve into the specifics of each visualization technique employed in this project and their unique contributions to understanding and addressing the challenges posed by network intrusions.

A. Principal Component Analysis (PCA):
The application of Principal Component Analysis (PCA) in visualizing the KDD Cup 1999 dataset is instrumental for capturing the underlying structures and patterns within the high-dimensional feature space. PCA reduces the dimensionality of the dataset while preserving its variance, allowing for the transformation of complex data into a lower-dimensional representation. By plotting the principal components in a three-dimensional space, distinct clusters or patterns in the data become discernible. This technique provides a holistic overview of the dataset's intrinsic structure, enabling cybersecurity professionals to identify potential correlations and anomalies.

```
# Input: Data matrix X with n samples and m features
# Output: Principal components and transformed data

# Step 1: Standardize the data
X_standardized = StandardScaler().fit_transform(X)

# Step 2: Compute the covariance matrix
covariance_matrix = np.cov(X_standardized, rowvar=False)

# Step 3: Compute the eigenvalues and eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(covariance_matrix)

# Step 4: Sort eigenvalues and corresponding eigenvectors
sorted_indices = np.argsort(eigenvalues)[::-1]
eigenvalues = eigenvalues[sorted_indices]
eigenvectors = eigenvectors[:, sorted_indices]

# Step 5: Choose the top k eigenvectors to form the transformation matrix
k = desired_number_of_dimensions
transformation_matrix = eigenvectors[:, :k]

# Step 6: Project the data onto the new subspace
X_pca = X_standardized.dot(transformation_matrix)
```

B. t-Distributed Stochastic Neighbor Embedding (t-SNE):
t-SNE is employed to visualize the KDD Cup 1999 dataset in a two-dimensional space, emphasizing the local relationships between data points. This technique is particularly useful for revealing intricate structures and clusters within the dataset that might not be apparent in higher-dimensional representations. By mapping instances with similar characteristics closer together in the visual space, t-SNE facilitates the identification of distinct groups, shedding light on the underlying nature of network intrusions and normal activities.

```
# Input: Data matrix X with n samples and m features
```

```
# Output: t-SNE embeddings

# Step 1: Compute pairwise similarities in the high-dimensional space
P = compute_pairwise_similarity(X)

# Step 2: Initialize low-dimensional representations randomly
Y = randomly_initialize_low_dimensional_space(n, desired_dimensions)

# Step 3: Set parameters
learning_rate = 200
iterations = 1000

# Step 4: Perform gradient descent
for iteration in range(iterations):
    Q = compute_pairwise_similarity(Y)
    gradient = compute_gradient(P, Q, Y)
    Y = update_embedding(Y, gradient, learning_rate)

# Output: Low-dimensional representations Y
```

C. K-Means Clustering:

The application of K-Means clustering involves grouping data points based on similarity, providing insights into distinct patterns or clusters of network activity. This visualization technique is crucial for identifying commonalities among instances of intrusions or normal behavior. By leveraging K-Means, the dataset is partitioned into clusters, and each point is assigned to the cluster with the nearest centroid. Visualizing these clusters offers a tangible representation of the different types of network intrusions present in the data, aiding in the understanding and classification of malicious activities.

```
# Input: Data matrix X with n samples and m features, number of clusters k
# Output: Cluster assignments for each data point

# Step 1: Initialize cluster centroids randomly
centroids = initialize_centroids(X, k)

# Step 2: Assign each data point to the nearest centroid
assignments = assign_to_nearest_centroid(X, centroids)

# Step 3: Update centroids based on the assigned points
centroids = update_centroids(X, assignments, k)

# Step 4: Repeat Steps 2 and 3 until convergence or a fixed number of iterations

# Output: Final cluster assignments
```

D. Bar Graphs:

Bar graphs are employed to visually represent the distribution of various attack types within the KDD Cup 1999 dataset. Each unique attack type is plotted along the x-axis, while the corresponding y-axis illustrates the frequency or count of occurrences. This visualization provides a clear and concise overview of the prevalence of different intrusion categories, allowing for quick identification of major threats. Bar graphs are particularly effective in conveying the relative frequencies of attacks, aiding cybersecurity professionals in prioritizing their focus on the most prominent security concerns.

E. Pie Charts:

Utilizing pie charts offers an alternative perspective on the distribution of attack types within the dataset. Each slice of the pie corresponds to a specific attack category, with the size of each slice proportional to the percentage of instances it represents. Pie charts provide an intuitive and visually appealing representation of the dataset's composition, enabling stakeholders to grasp the relative importance of each attack type at a glance. This visualization technique adds a layer of accessibility to the analysis, supporting a broader audience in understanding the distribution of network intrusions.

F. QT Framework Application:

To enhance the accessibility and interactivity of the visualizations, a standalone application is developed using the QT Framework. This application provides users with an intuitive interface to explore and analyze the KDD Cup 1999 dataset. Through the application, users can seamlessly select different visualization methods, input datasets, and adjust parameters. The integration of the QT Framework ensures a user-friendly experience, allowing cybersecurity professionals and researchers to interact with the data visualizations effortlessly.

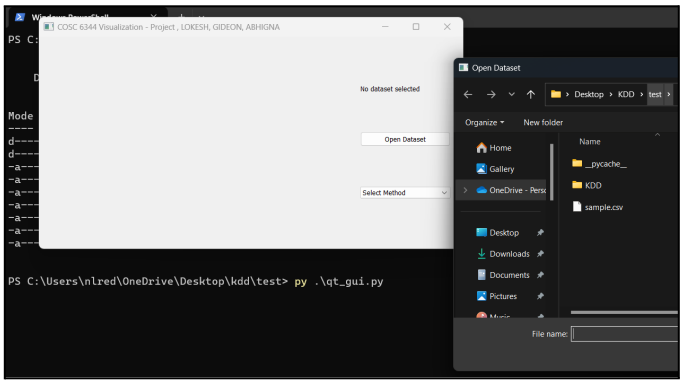

Fig. 6. QT interface

Preprocessing the KDD Cup 1999 dataset is a critical phase in preparing the data for subsequent analysis. Given the extensive nature of the dataset, strategic measures are implemented to handle its complexity and ensure optimal model performance. One key aspect of preprocessing involves addressing class imbalance, a common challenge in

cybersecurity datasets where instances of normal network activity significantly outnumber instances of intrusions. To mitigate this issue, a custom Python script named "sample.py" is employed. This script utilizes random sampling techniques to generate a representative subset of the original dataset while maintaining the distribution of the target variable. By strategically selecting a proportional number of instances for each unique value in the last column, the script creates a balanced and manageable dataset for further exploration and model training.

Additionally, preprocessing encompasses the transformation of symbolic data into a format suitable for analysis and model training. Techniques such as label encoding are applied to convert categorical variables into numerical representations, facilitating the compatibility of the data with various machine-learning algorithms. Standardization techniques, such as scaling, are also employed to bring numerical features to a common scale, preventing certain features from dominating the learning process due to their larger magnitudes. These preprocessing steps collectively contribute to a more refined and standardized dataset, setting the stage for the application of advanced machine learning techniques.

Furthermore, the preprocessing pipeline involves the correction of discrepancies within the dataset. The script "sample.py" not only addresses class imbalance but also ensures that the sampled dataset accurately reflects the underlying characteristics of the original data. Typos or errors within the dataset are identified and rectified, promoting data integrity and mitigating potential biases that could arise during analysis. Overall, the preprocessing phase is pivotal in laying a solid foundation for subsequent knowledge discovery and data mining endeavors, ensuring that the KDD Cup 1999 dataset is optimized for effective analysis and model development in the field of network security.

G. Team Membres Contribution:

Lokeshwar Reddy Nandanapalli (2299460)

1. Created a QT application with a file opening button to select data files and a dropdown menu to select the type of visualization.
2. Created PCA plot using matplotlib and integrated it with qt application.
3. Created a Bar graph using matplotlib to visualize attack types and integrated it with the QT application.

Ben Gideon Dokiburra (2283917)

1. Created a pie chart using Matplotlib to visualize attack types and integrated it with the qt application.
2. Created visualization of attack-type clusters using K-means in sci-kit-learn and integrated it with the qt application.

Abhigna Sowgandhika Vadlamudi (2268166)

1. Preprocessed the data and created sample.csv by writing a program to sample a required number of rows by ensuring that the attack types remain in equal proportion in the sampled data.
2. Created t-SNE visualization using sci-kit-learn and matplotlib for the data and integrated it with the qt application.

IV. RESULTS

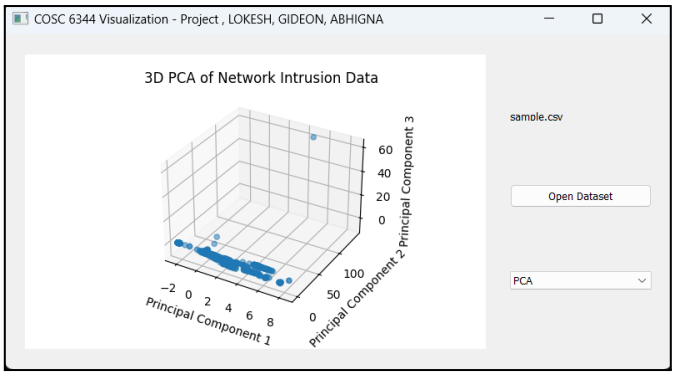

Fig. 1. PCA plot of sample.csv

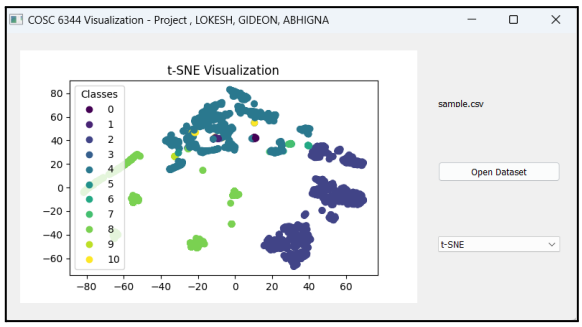

Fig. 2. t-SNE plot of sample.csv

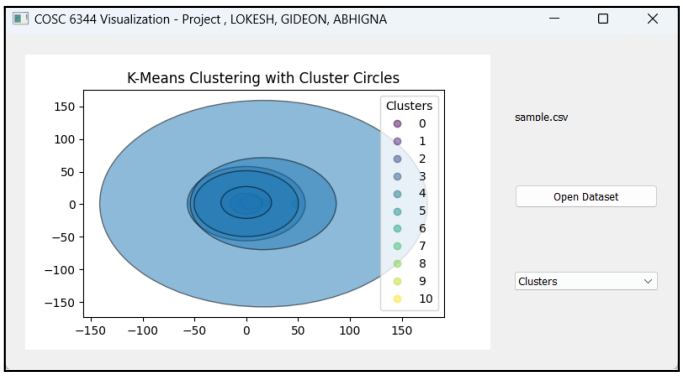

Fig. 3. Cluster plot of sample.csv

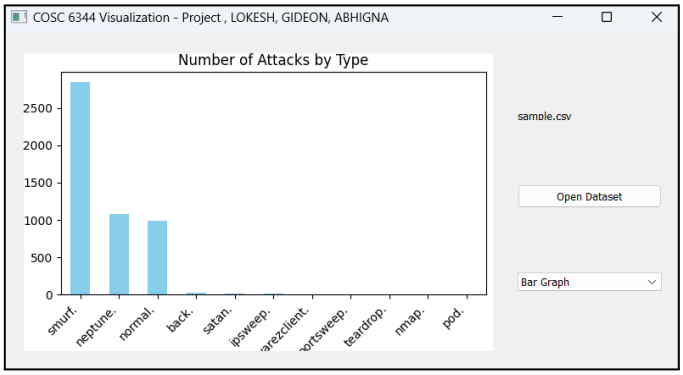

Fig. 4. Bar graph of sample.csv

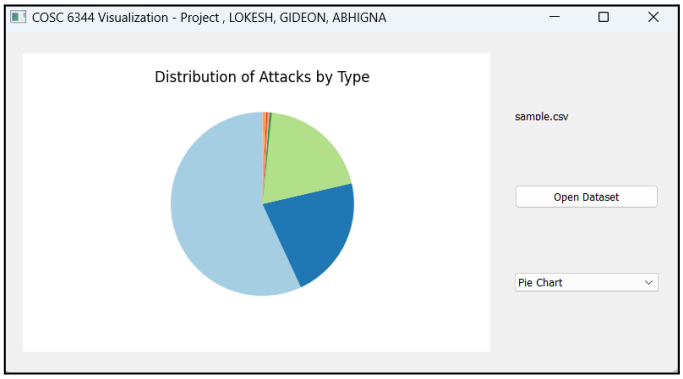

Fig. 5. Pie chart of sample.csv

V. CONCLUSION AND FUTURE WORK

The application of diverse visualization techniques on the KDD Cup 1999 dataset has yielded valuable insights into the realm of network intrusion detection. Through the lens of Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE), the inherent complexities within the dataset have been distilled into visually comprehensible representations, enabling the identification of distinct patterns and clusters. K-Means clustering further enhances our understanding by grouping similar instances, shedding light on the diverse nature of network activities. Bar graphs and pie charts contribute a concise visual summary of the distribution of various attack types, offering cybersecurity professionals an accessible means of prioritizing threats.

Moreover, the integration of the QT Framework into a standalone application has significantly enriched the user experience, providing a seamless interface for exploring and interacting with the visualizations. This project not only exemplifies the power of advanced visualization techniques in cybersecurity analysis but also underscores the importance of accessibility and user-friendly interfaces in facilitating informed decision-making.

As we move forward, the insights gained from these visualizations will serve as a solid foundation for the development and implementation of robust network intrusion detection systems. By combining the strengths of data visualization with advanced machine learning techniques, future endeavors in this domain are poised to make significant strides in bolstering network security and mitigating the evolving landscape of cyber threats. The amalgamation of cutting-edge visualizations and sophisticated frameworks positions this project at the forefront of knowledge discovery and data mining in the field of cybersecurity.

Adding functionality for more types of visualizations and improving the dataset by doing better preprocessing to discover hidden features can be done in future.