SQL INJECTION ATTACK DETECTION

(PHASE–2)

PRESENTED BY

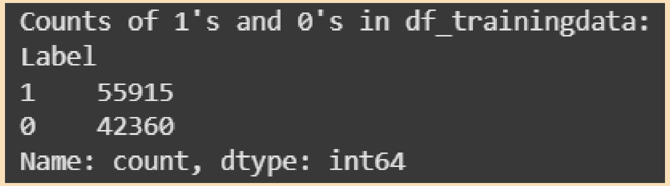
NAME: AVINASH M
REG. NO: 22011101019
CLASS: AI & DS ‘A’ (3RD YEAR)

INTRODUCTION

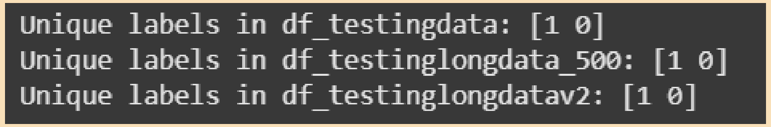
- SQL Injection (SQLi) is an attack that manipulates SQL queries to gain unauthorized access.
- SQL injection attacks exploit vulnerabilities in web applications that allow attackers to insert malicious SQL code into database queries
- SQL injection is a major threat. It can lead to serious data breaches and compromise sensitive information. These attacks can enable attackers to bypass authentication, steal sensitive data, modify or delete database records, and even gain control of the server.
- Objective: To Develop an AI-driven system that can automatically detect and mitigate SQLi attempts in real-time.

DATASET DESCRIPTION

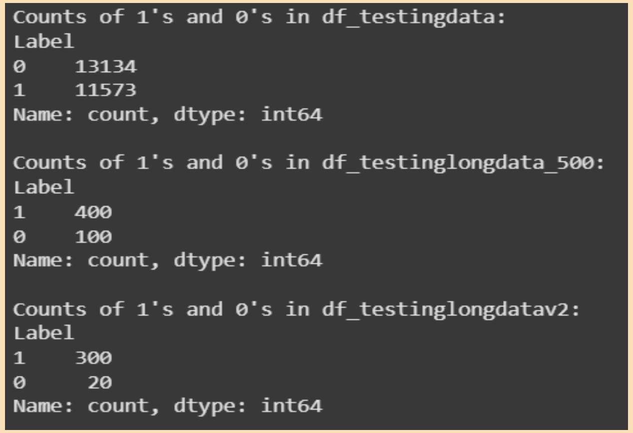
- The datasets (testingdata.csv, testinglongdata_500.csv, testinglongdatav2.csv) contains website traffic data, possibly for security analysis.
- It include features like query length, number of digits, and special characters.
- The target variable, ‘Label’, likely indicates whether a query is malicious(1) or benign (0).



```
Counts of 1's and 0's in df_trainingdata:
Label
1    55915
0    42360
Name: count, dtype: int64
```



```
Unique labels in df_testingdata: [1 0]
Unique labels in df_testinglongdata_500: [1 0]
Unique labels in df_testinglongdatav2: [1 0]
```



```
Counts of 1's and 0's in df_testingdata:
Label
0    13134
1    11573
Name: count, dtype: int64

Counts of 1's and 0's in df_testinglongdata_500:
Label
1    400
0    100
Name: count, dtype: int64

Counts of 1's and 0's in df_testinglongdatav2:
Label
1    300
0     20
Name: count, dtype: int64
```

Dataset Link: SQL Injection Dataset

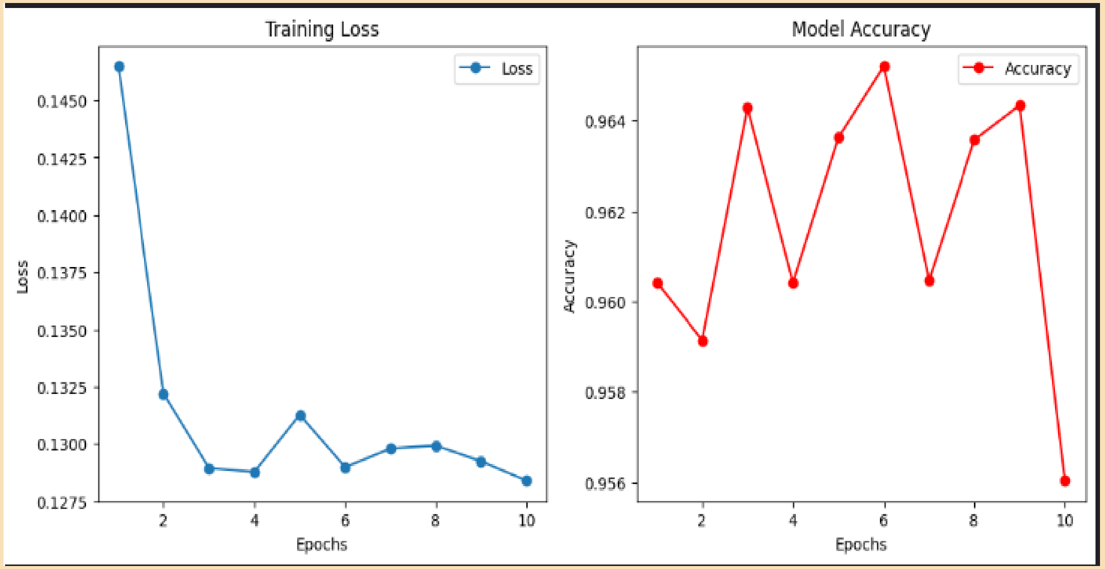
IMPLEMENTED SYSTEM

(Deep Learning Based SQLi Detection)

METHODS	MODEL	ACHIEVED ACCURACY
METHOD 1	GRAPH NEURAL NETWORK	96.10 %
METHOD 2	CONTRASTIVE LEARNING METHOD	95.37 %
METHOD 3	TRANSFORMERS (BERT) + PCA + SVM (HYBRID APPROACH)	63.16 %

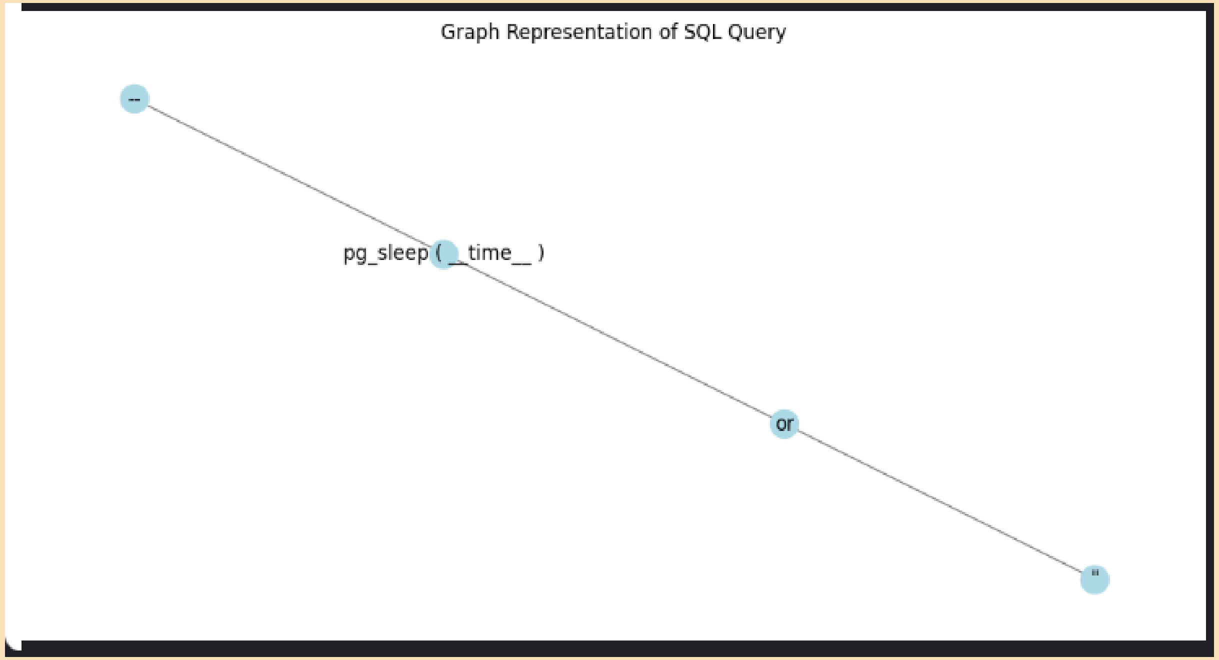
GRAPH NEURAL NETWORK (INFERENCE)

Model Accuracy: 95.60 %



Training Loss

Model Accuracy



Graph Representation of SQL Query

WHY GRAPH NEURAL NETWORK ?

1) Captures Query Structure Effectively

- GNNs represent SQL queries as graphs, preserving token dependencies and relationships.

2) Handles Variable-Length Queries

- Unlike traditional models, GNNs process queries of different lengths without needing padding.

3) Detects Complex Injection Patterns

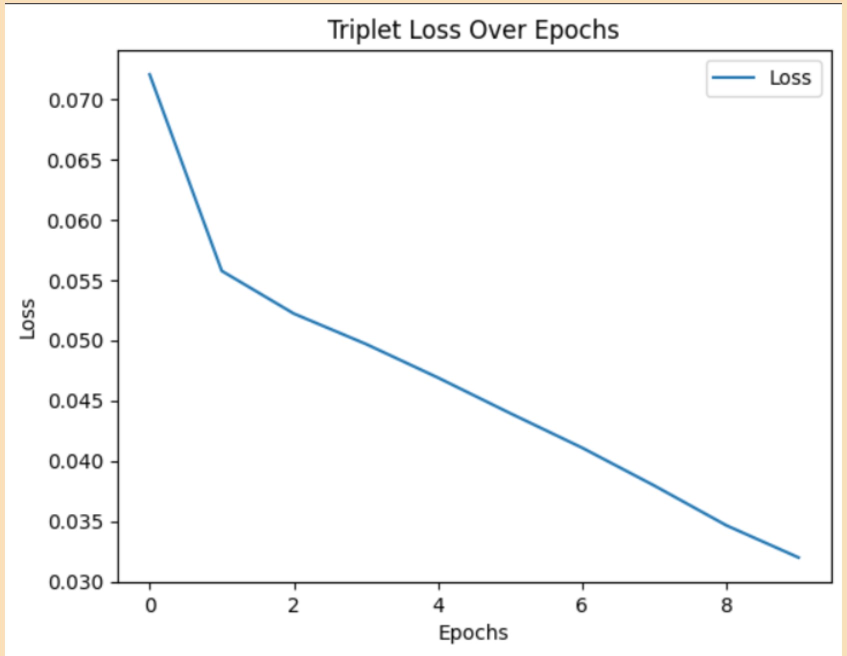
- Graph-based learning helps identify subtle attack patterns by modeling token interactions.

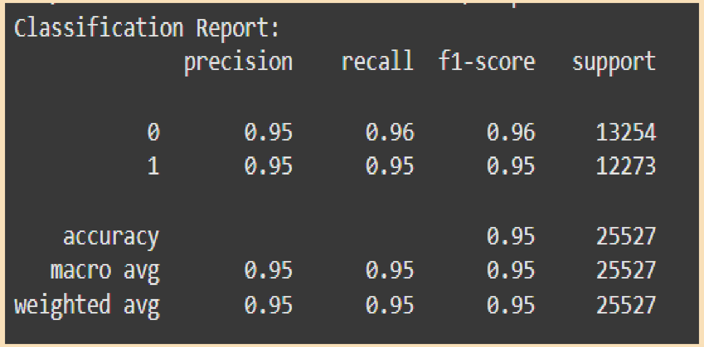
4) Context-Aware Feature Learning

- Word2Vec embeddings and GCN layers extract meaningful semantic and structural features for accurate classification.

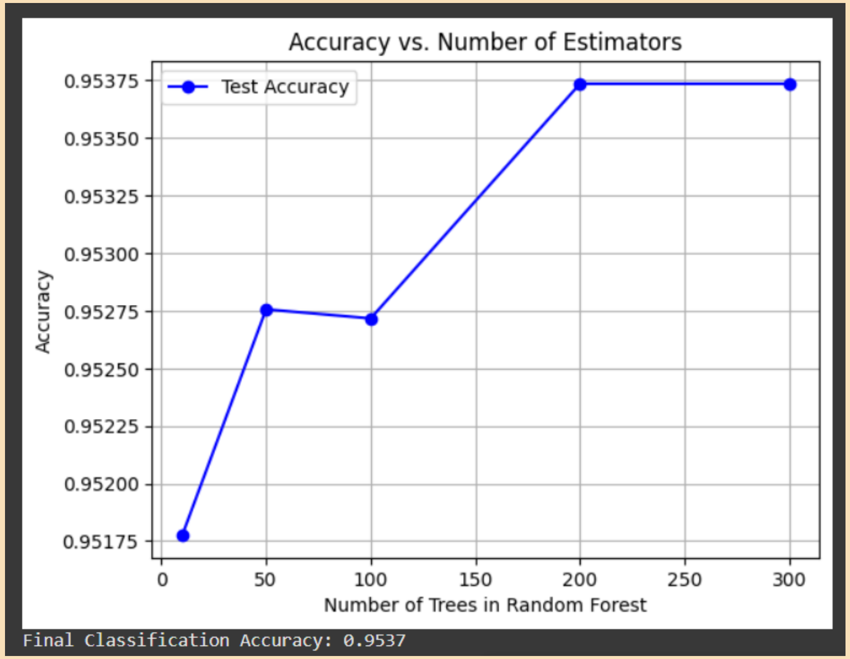
CONTRASTIVE LEARNING (INFERENCE)

Model Accuracy: 95.37 %





```
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.96      0.96     13254
           1       0.95      0.95      0.95     12273

    accuracy                           0.95     25527
   macro avg       0.95      0.95      0.95     25527
weighted avg       0.95      0.95      0.95     25527
```



```
Final Classification Accuracy: 0.9537
```

WHY CONTRASTIVE LEARNING ?

1) Learns Query Similarities & Differences

- The **Siamese Network with Triplet Loss** trains on both normal and injected queries.
- Helps the model **distinguish between legitimate and malicious patterns**.

2) Detects Zero-Day SQL Injection Attacks

- Unlike traditional classifiers, contrastive learning does **not rely on predefined attack patterns**.
- Can generalize to **new types of SQL injections** effectively.

3) Generates Robust Query Representations

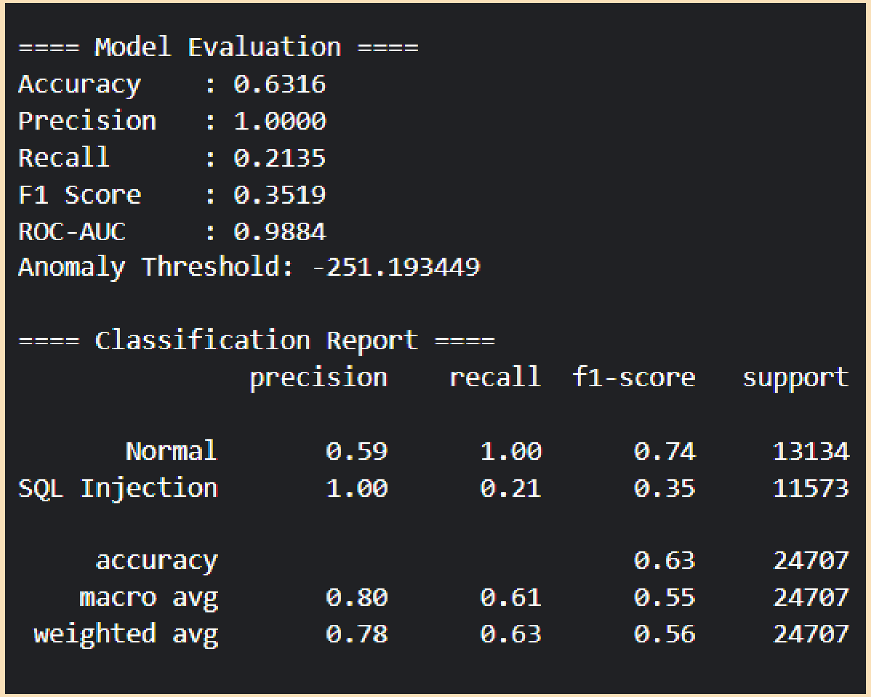
- Word2Vec and LSTM-based embeddings **capture contextual relationships** in SQL queries.
- The **embedding space separates normal and injected queries**, improving classification.

4) Enhances Classification Accuracy

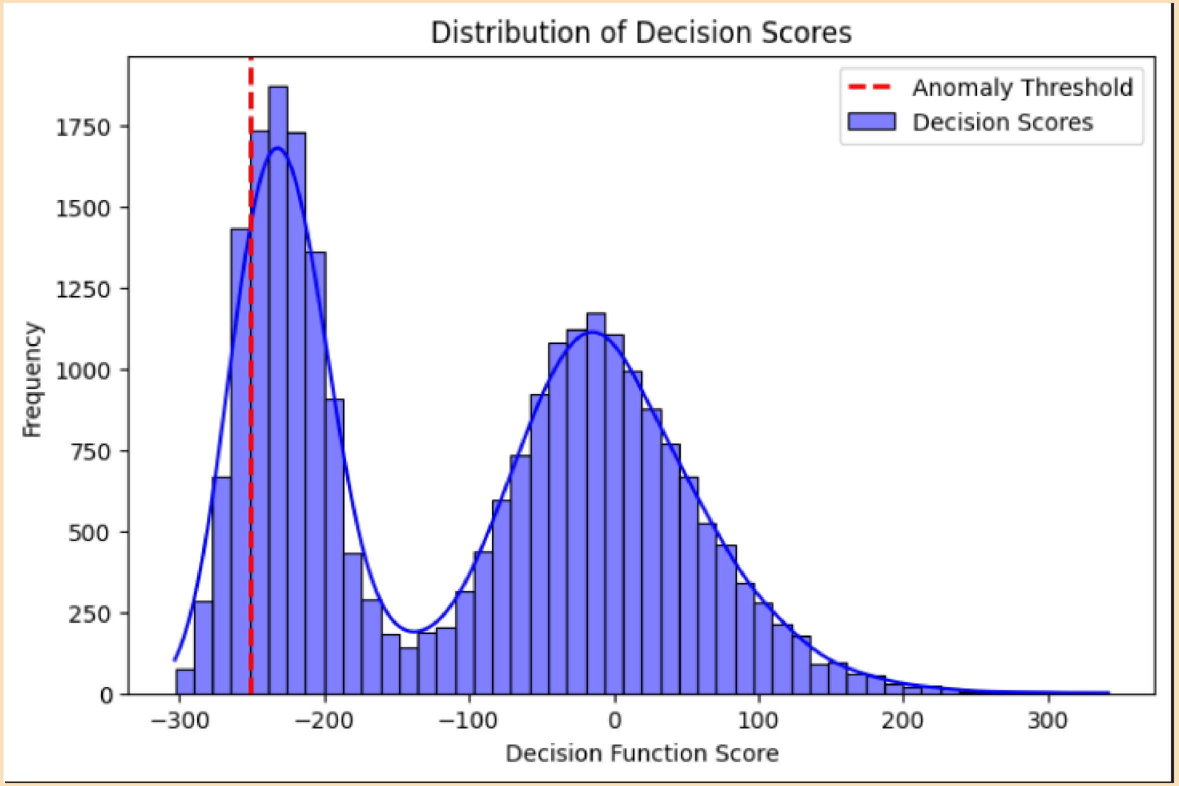
- Embeddings from the Siamese model **boost the performance** of classifiers like **Random Forest**.
- Helps achieve **higher accuracy** compared to standard feature-based models.

TRANSFORMERS + PCA +SVM (HYBRID APPROACH) (INFERENCE)

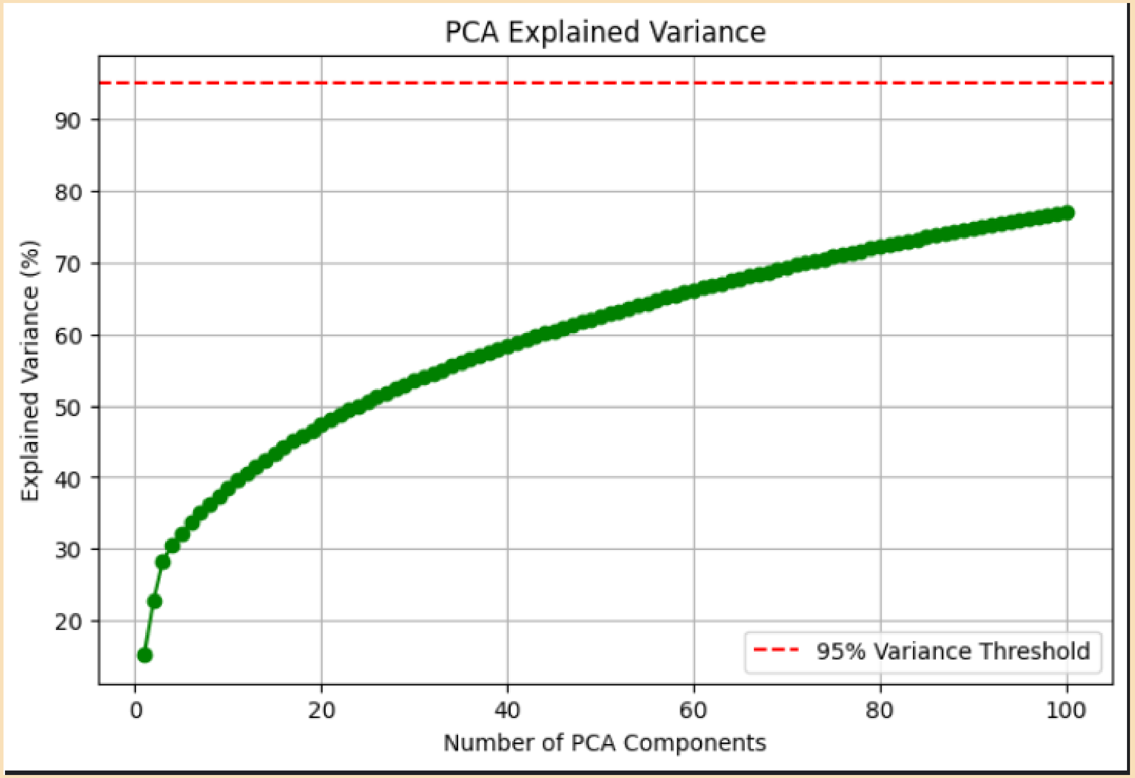
Model Accuracy: 63.16 %

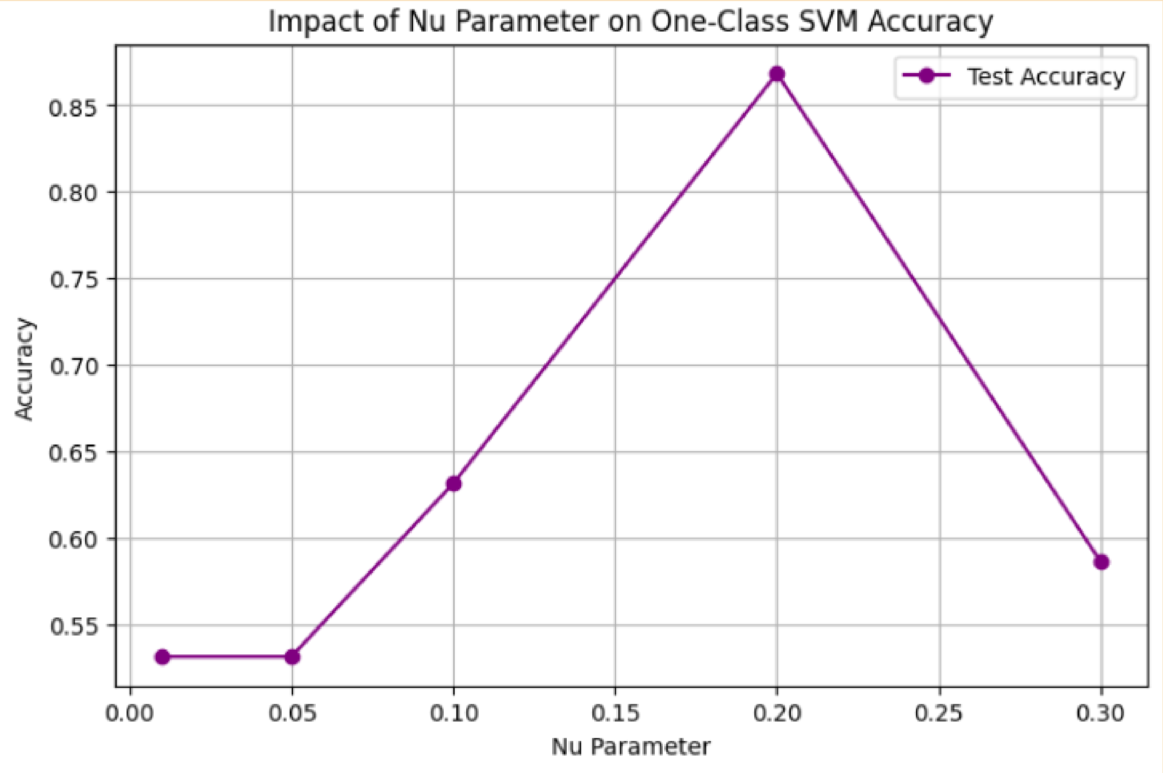


```
==== Model Evaluation ====
Accuracy    : 0.6316
Precision   : 1.0000
Recall      : 0.2135
F1 Score    : 0.3519
ROC-AUC     : 0.9884
Anomaly Threshold: -251.193449

==== Classification Report ====
               precision    recall  f1-score   support

       Normal       0.59      1.00      0.74     13134
SQL Injection       1.00      0.21      0.35     11573

     accuracy                           0.63     24707
    macro avg       0.80      0.61      0.55     24707
 weighted avg       0.78      0.63      0.56     24707
```



TRANSFORMERS + PCA +SVM (HYBRID APPROACH) (INFERENCE)





WHY HYBRID APPROACH ?
(TRANSFORMER (BERT) + PCA +SVM)

1) Extracts Powerful Query Representations

- **Sentence Transformer (SBERT)** generates **context-rich embeddings**, preserving SQL query meaning.

2) Reduces Computational Complexity

- **PCA** reduces high-dimensional embeddings to **100 key features**, improving efficiency without losing important information.

3) Enhances Anomaly Detection

- **One-Class SVM** learns patterns from normal queries and effectively flags **zero-day SQL injection attacks**.

4) Optimizes Detection Performance

- The **decision score distribution & hyperparameter tuning (nu values)** ensure **high accuracy and robustness** in detecting malicious queries.

CONCLUSION

1) GNN Achieves the Best Performance (96.10% Accuracy)

- Graph Neural Networks (GNNs) effectively capture **structural relationships** in SQL queries, making them the most accurate.

2) Contrastive Learning is Highly Effective (95.37% Accuracy)

- The **Siamese Network with Triplet Loss** learns **query similarities & differences**, improving generalization to new attack patterns.

3) Hybrid Approach (Transformer + PCA + SVM) Underperforms (63.16% Accuracy)

- While computationally efficient, **PCA reduces feature quality**, and **SVM struggles with complex SQL injection patterns**.

4) Graph-Based Learning is the Most Robust

- GNNs outperform other models by **leveraging SQL token dependencies**, making them more resistant to evasion techniques.

5) Future Improvements

- **Combining GNN with Contrastive Learning** may further **enhance detection accuracy & generalization**.

REFERENCES

- https://link.springer.com/article/10.1007/s11227-025-07109-w
- https://journal.uinjkt.ac.id/index.php/inprime/article/view/41025
- https://www.mdpi.com/2076-3417/15/2/571