ECGR 5101 Intro to Machine Learning

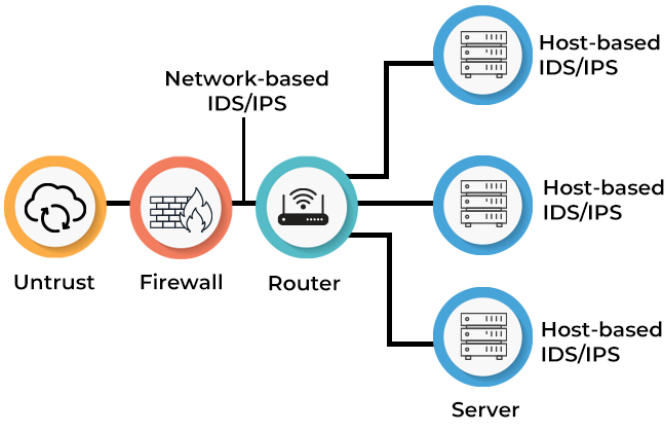
Title: Machine Learning-based Intrusion Detection Based System (IDS)

Name: Behnam Sobhani Nadri **Student ID: 809368949**

GitHub Link: [behnamsn/IntroML]

Introduction

Intrusion Detection Systems (IDS) are essential components in the realm of cybersecurity. They are designed to monitor and analyze network traffic for any signs of malicious activity, unauthorized access, or policy violations. As cyber threats become more sophisticated, traditional rule-based IDS struggle to keep pace. In recent years, the potential of Machine Learning (ML)-based models has been studied in IDSs. Supervised ML algorithms can learn the normal and abnormal data traffic from the characteristics of the network connections, and later detect patterns of intrusions. The goal of this project is to test various ML techniques and find the most optimum ML model to detect the abnormal traffic pattern from normal one. The project go through a classification problem step by step and evaluates the final results of three different supervised algorithms.



Dataset

The dataset that we have used is a dataset from Kaggle website. The data has been obtained from a simulated Local Area Network (LAN) environment. This project use the train_data.csv file as the dataset which consists of connection types such as connections' duration ["**duration**"], amount of the data in bytes at the source and destination ["**src_bytes**","**dst_bytes**"]. The connections can be either **TCP, UDP** or **ICMP**. In the dataset, this feature can be found in the

[“**protocol_type**”]. Since this feature is categorical, we have manually separated it into three other binary features that can identify the type of connection [“**tcp_service**”, “**udp_service**”, “**icmp_service**”].

Methodology

The approach consists of four main stages:

1. **Cross Validation**
   The dataset was divided into training and testing sets, with **75%** allocated for training and **25%** for validation. A random state was also applied to maintain consistency in the data split across different experiments.
2. **Preprocessing and Feature Engineering**
   The heatmap below shows the correlation among the features. It demonstrates that several features have high correlation with respect to each other and as a result, we can remove them from the traning to reduce the redundency and improve the training. Three features [“**num_compromised**”,”**su_attempted**”, “**num_root**” and “**num_access_files**”] have been dropped by using **drop()** method.

3. Standardization

Standardization has been used to improve the training and achieve higher accuracy and lower errors. In this project StandardScaler() has been applied to the dataset.

4. Machine Learning Model Trainings and Evaluations

According to the nature of the problem, several models can be used to classify the connection as normal or abnormal.

1. Logistic Regression (LR)

2. Support Vector Machine (SVM)

- Linear kernel
- Sigmoid kernel
- Polynomial kernel

3. Naive Bayes (Gaussian NB)

In addition to these models, we investigated the impact of Principal Component Analysis (PCA) combined with each model to evaluate the performance. PCA is a dimensionality reduction technique that transforms the data into a set of linearly uncorrelated components, helping to reduce the complexity and computational load. We have done some experiments to choose the most optimum K parameter for the PCA. We compare the results of the models combined with PCA and not combined.

Results and Discussion

1. Logistic Regression (LR)

Logistic Regression (LR) is a linear model that predicts the probability of a binary outcome. It performed well on the IDS dataset:

- Achieved a maximum accuracy of **95.04%.** The confusion matrix for this model (Figure 1) shows a balanced performance across both classes, indicating good discrimination between normal and malicious activities.

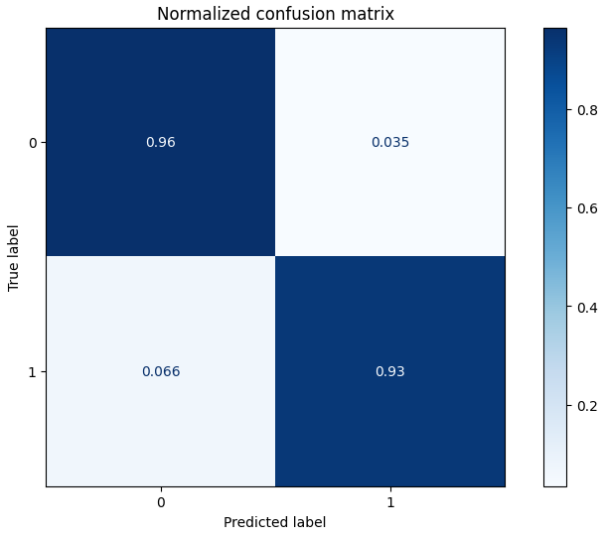


Figure 1

- Using PCA with 18 components **K=18**, the maximum accuracy was 95.04%, indicating a negligible change. This suggests that the original feature space was already optimal for LR, and PCA did not significantly enhance the model's performance.

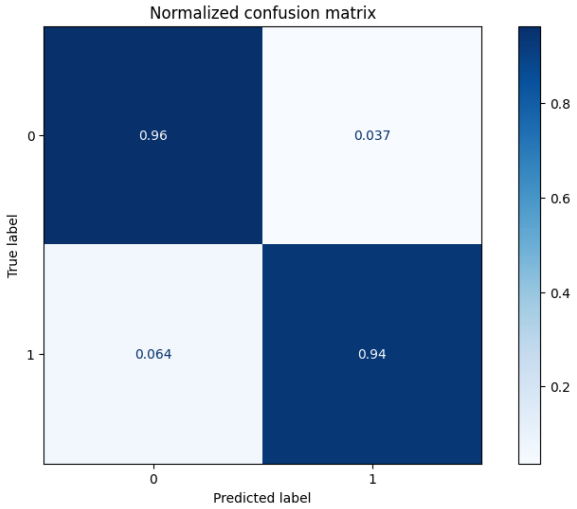
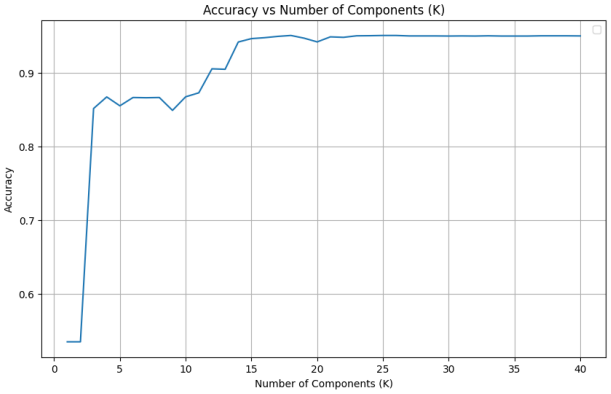
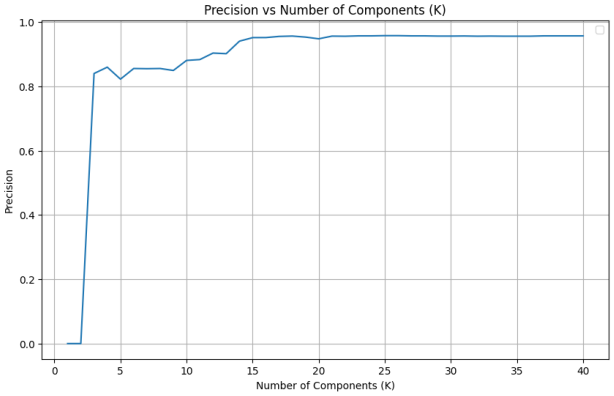


Figure 2





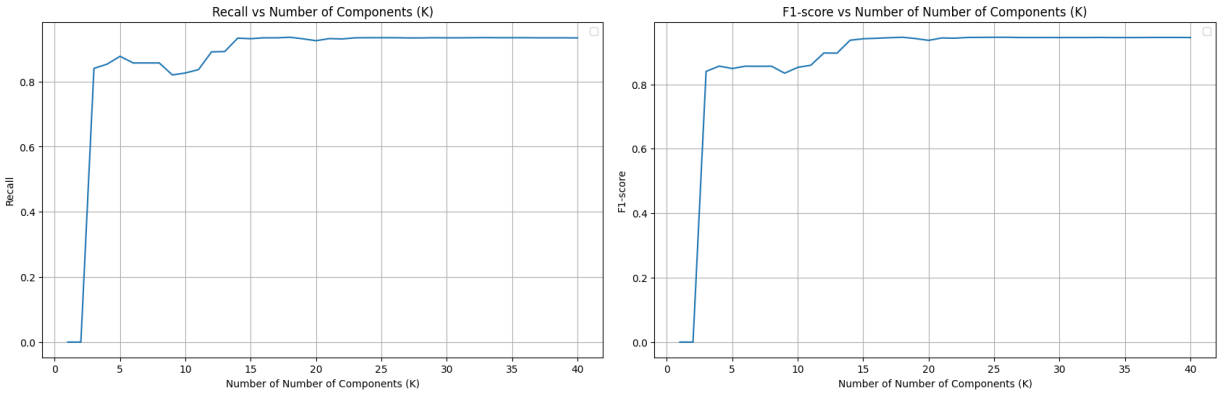


Figure 3

2. Support Vector Machine (SVM)

SVMs are powerful classifiers that find the optimal hyperplane to separate classes. We tested both linear and sigmoid kernels:

- **Linear Kernel:** The maximum accuracy was 95.42%.

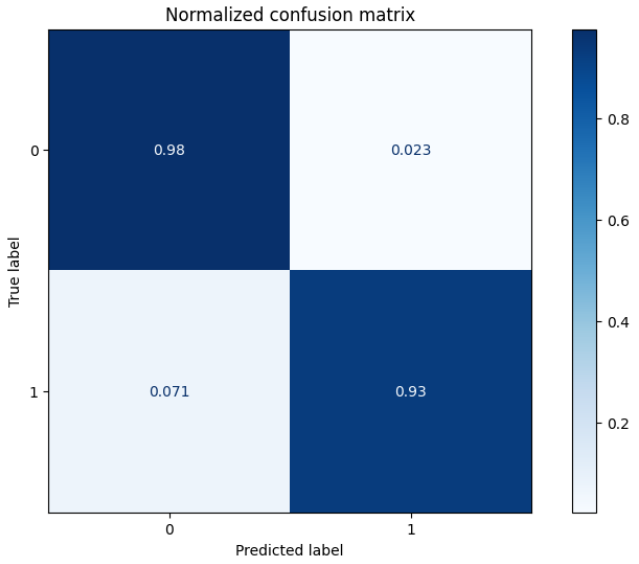


Figure 4

- Using PCA with 32 components (K) the maximum accuracy slightly decreased to 95.20%. Similar to LR, PCA did not contribute significantly to performance improvement in SVM.

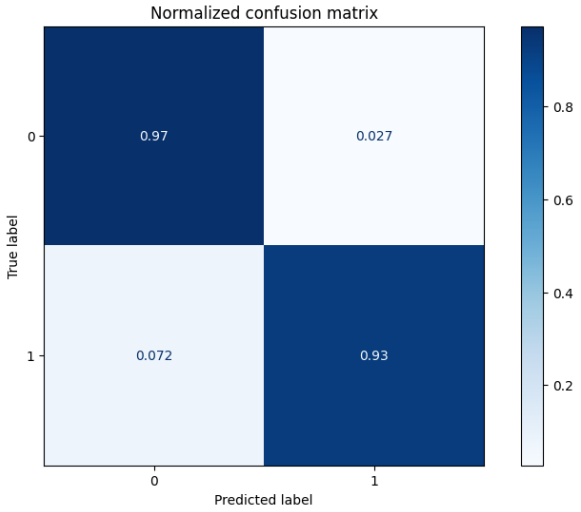


Figure 5

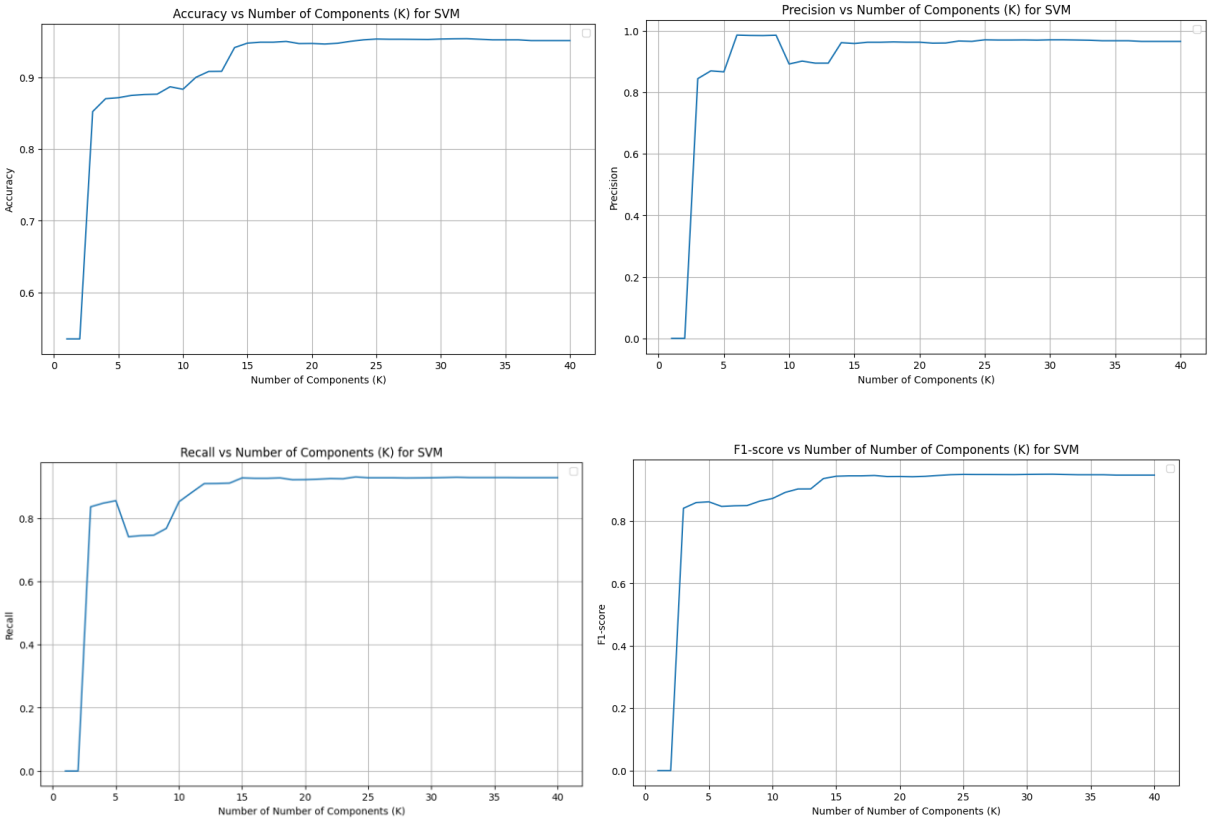


Figure 6

- **Sigmoid Kernel:** The sigmoid kernel, with a maximum accuracy of 90.34%, underperformed compared to the linear kernel. This result suggests that the sigmoid kernel, which is less commonly used for binary classification, is not useful for this dataset.

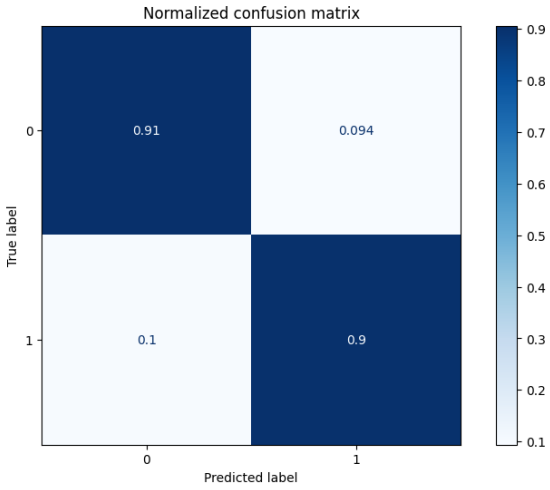


Figure 7

- Using PCA with 32 components (K) the maximum accuracy dramatically reduce to 86.44%. Similar to the previous case, PCA wasn't useful to enhance the performance of SVM.

- **Polynomial Kernel:** This setup achieved the highest accuracy among all models, with a maximum of **98.06%.**

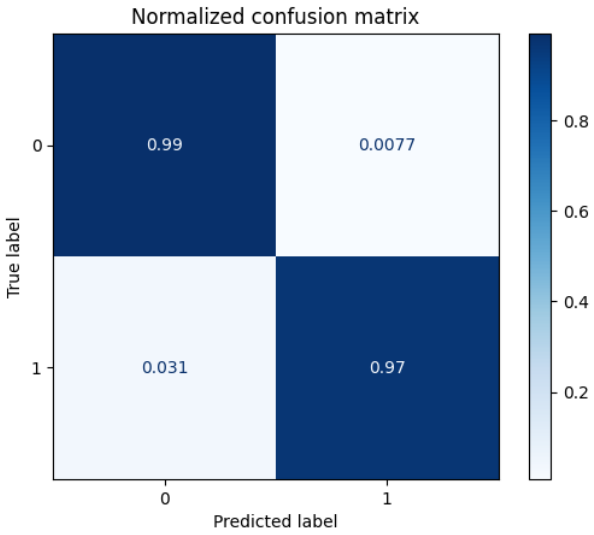


Figure 8

- Using PCA with 32 components **K=32** the maximum accuracy peaks at **98.80%.** This is the highest accuracy that was achieved among all test of the three models

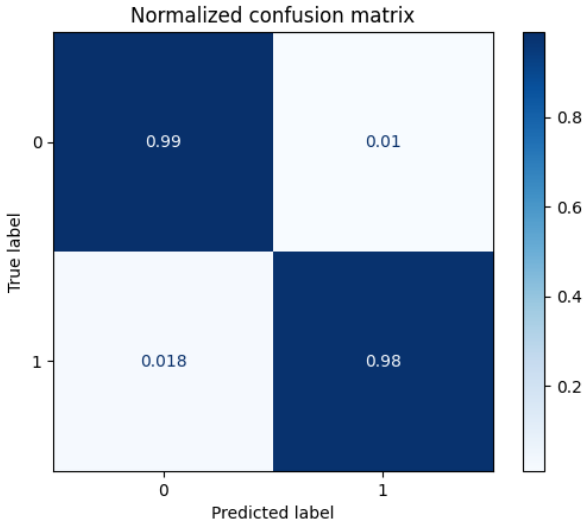


Figure 9

3. Naïve Bayes Classifier (Gaussian)

Naive Bayes classifiers assume independence among predictors, making them simple yet efficient:

- Achieved a maximum accuracy of **92.1%**. The confusion matrix (Figure 10) indicates that the model is effective but less accurate than LR and SVM.

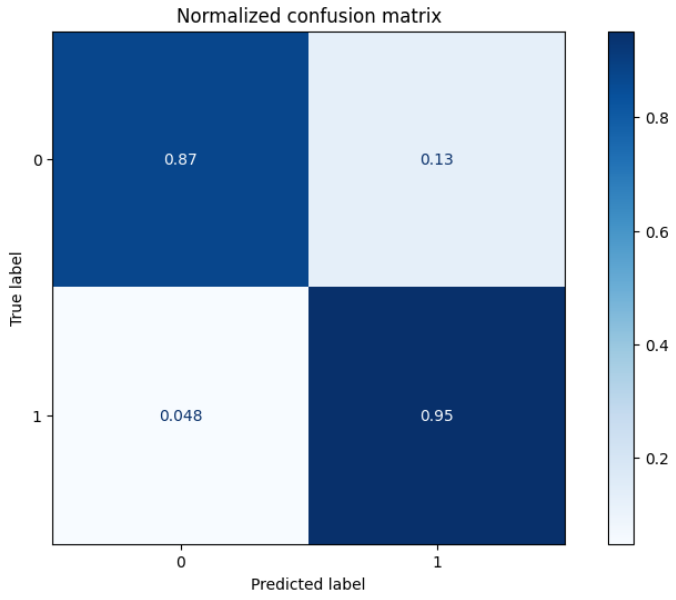


Figure 10

- Using PCA **(K=21)**: The accuracy remained constant at **92.1%**. The application of PCA did not improve the model's performance, possibly because Gaussian NB relies on the assumption of feature independence, which PCA does not necessarily align with.

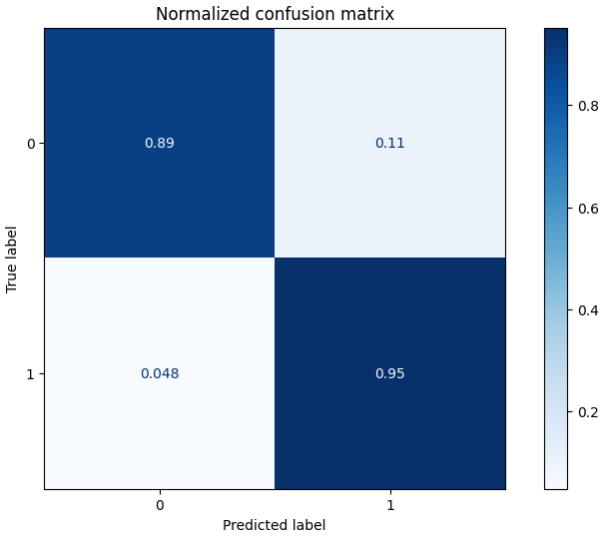


Figure 11

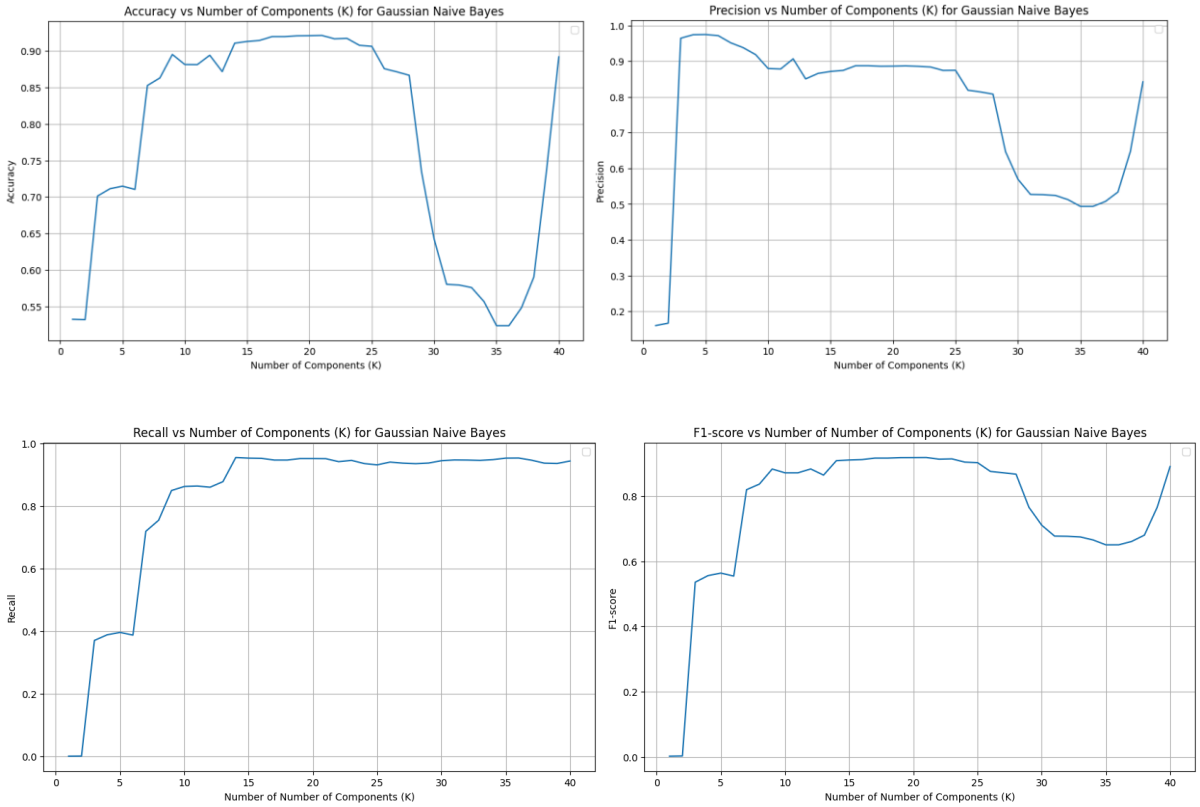


Figure 12

Analysis

The comparison of the three models reveals several insights:

1. **Performance**: The SVM with a polynomial kernel slightly outperformed the other models, demonstrating the highest accuracy of **98.06%** followed closely by Logistic Regression. The Gaussian NB model with **92.12%** was the lowest accurate model among all three models.

2. **Impact of PCA**: The results show that PCA increased the SVM's accuracy up to **3%** for polynomial kernel and **5%** with linear kernel. However, the LR and Gaussian NB's accuracy remains constant even after applying PCA to the pipeline. Overall, PCA has been useful for SVM.

3. **Model Selection**: The highest accurate model among the three is SVM with polynomial kernel. PCA has also increased the performance of SVM significantly. The linear kernel was more effective than the sigmoid kernel, underscoring the importance of kernel selection in SVM. This also shows that the dataset's feature space supports linear separability.

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression (LR)	95.04%	95.67%	93.58%	94.61%
SVM with polynomial kernel	**98.06%**	**99.05%**	**96.75%**	**97.89%**
SVM with linear kernel	95.42%	97.17%	92.86%	94.97%
SVM with sigmoid	90.34%	89.30%	90.03%	89.66%
Gaussian Naïve Bayes	92.12%	88.70%	95.18%	91.83%

Model with PCA	Accuracy	Precision	Recall	F1-score
Logistic Regression (LR)	95.04%	95.67%	93.58%	94.61%
SVM with polynomial kernel	**98.80%**	**98.80%**	**98.62%**	**98.71%**
SVM with linear kernel	95.20%	96.70%	92.78%	94.70%
SVM with sigmoid	86.44%	85.06%	85.70%	85.38%
Gaussian Naïve Bayes	92.12%	88.70%	95.18%	91.83%

Conclusion

The project successfully demonstrated the potential of ML models in an Intrusion Detection System. The SVM with a polynomial kernel was the most accurate ML algorithm followed closely by Logistic Regression. On the other hand, the Gaussian NB model did not achieve a satisfying level of accuracy. PCA, a common technique for reducing dimensionality, enhanced the performance of SVM significantly, while other algorithms could not improve their accuracy by applying PCA to the solution. This can demonstrate that the dataset was already well-represented by its original features in Naïve Bayes and Logistic Regression.

Future work could explore other feature engineering techniques, ensemble learning methods, or even deep learning approaches to further improve the system's accuracy and robustness. Additionally, real-world application would require the IDS to handle data in real-time, they require certain considerations for computational efficiency and response time.

References

Network Intrusion Detection System (IDS) - Kaggle

IDS vs. IPS: Key Difference and Similarities (spiceworks.com)