CREDIT CARD FRAUD DETECTION

Data Mining CI603

Executive Summary

In the age of digital finance, credit card fraud remains a persistent and costly challenge for financial institutions. With billions of transactions processed daily, identifying fraudulent activity quickly and accurately is essential to protecting customers and minimizing financial loss.

This project explores the application of data mining techniques to detect credit card fraud using a real-world dataset of European transactions. The dataset, containing over 280,000 entries, is heavily imbalanced—with fraudulent activity making up less than 0.2% of all cases—making traditional detection methods ineffective.

To address this, four models were implemented and evaluated: a statistical Z-Score Method, an unsupervised Isolation Forest, a probabilistic Naïve Bayes classifier, and a Support Vector Machine. Each model was assessed using precision, recall, F1-score, ROC AUC, and average precision to reflect its ability to handle imbalanced data.

While Naïve Bayes achieved the highest fraud detection rate (recall), it suffered from high false positives. In contrast Support Vector Machine offered the best balance between fraud detection and precision, making them the most suitable model for real-world deployment where both accuracy and efficiency matter.

This report concludes that Support Vector Machines, presented the most reliable approach for integrating fraud detection into a modern financial system of the four models tested; this is supported by strong precision-recall performance and low false positive rates.

George McDonald

22837558

Contents

Introduction

Data Mining plays a critical role in modern digital systems by uncovering patterns, relationships, and anomalies within large dataset. In the financial sector, one of the most significant and persistent threats is credit card fraud—a problem that costs billions globally each year. As digital transactions increase in volume and complexity, fraudsters continue to evolve their methods, making traditional rule-based detection systems in increasingly ineffective.

According to a recent review, credit card fraud detection presents unique challenges due to the high volume and variability of transactional data, often compounded by issues such as imbalanced datasets and the need for real-time processing (Cherif et al., 2023). In response, modern systems increasingly rely on AI and machine learning techniques to improve fraud detection accuracy while reducing operational costs and false alarms.

This project investigates the use of data mining techniques to detect credit card fraud using a real-world dataset. Four models were tested and evaluated: Z-score, Isolation Forest (iForest), Naïve Bayes (NB) and Support Vector Machines (SVMs). Each technique was applied using Python and evaluated using industry standard performance metrics.

This report is structured to inform about the dataset and pre-processing done, then detailing the theoretical background of each model, after which implementation, results, and visualisation are discussed allowing an understanding and comparative assessment of each model, which then leads to a final recommendation for practical deployment in fraud detection systems.

Dataset Overview

The dataset used in this project originates from a publicly available source on Kaggle (ULB), which has a DbCL license. The dataset contains anonymized credit card transaction records made by European cardholders in September 2013. The dataset includes 284,807 transactions, of which only 492 are labelled as fraudulent—roughly 0.17%, making this dataset highly imbalanced. This imbalance presents a real-world challenge: designing models that are sensitive enough to catch rare frauds without overwhelming the system with false positives.

Each transaction is described using 30 features:

- Time: the number of seconds elapsed between this transaction and the first transaction in the dataset.
- Amount: the transaction value.
- Class: the label indicating whether a transaction is fraudulent (1) or legitimate (0).
- V1 – V28: principal components obtained through PCS (Principal Component Analysis), anonymized due to confidentiality reasons.

Despite the anonymized features, the dataset remains highly suitable for pattern recognition and anomaly detection, as the PCA components still preserve structural variance in the data. The datasets purely numerical nature means no categorical encoding is required, simplifying preprocessing.

Data Preprocessing

Before applying any data mining techniques, the dataset underwent several preprocessing steps to ensure it was suitable for model training and evaluation.

Initially the Time and Amount features were standardized using a StandardScaler, which transformed their distributions to have a mean of zero and unit variance. This step was necessary due to the scale of Time and Amount differs from the normalized PCA features and thus required standardization to ensure consistent model performance.

Once scaled, the original Time and Amount columns were dropped and replaced with normalized equivalents. This helped maintain numerical consistency across all features, avoiding potential model bias toward higher-valued columns.

The dataset was then split into training and testing sets using an 80/20 ratio ensuring that both sets preserved the original class distribution through stratified sampling. A critical step given the extreme imbalance in the dataset, as random splitting without stratification could easily produce training or test sets with zero fraud examples.

No oversampling or under-sampling techniques (such as SMOTE or random downsampling) were applied to avoid distorting the raw class proportions or introducing synthetic noise. Instead, class imbalance was addressed during model evaluation using metrics specifically designed for imbalance classification—such as recall, precision, F1-score, AUC, and average precision (AP).

The final pre-processed dataset consisted entirely of numerical and normalized features, making it well-suited for a variety of supervised and unsupervised learning algorithms without further alterations.

Theoretical Background of Techniques

This section explores the theoretical foundations of the four models employed in this project. Each model offers unique mechanisms for identifying fraudulent transactions, and understanding their inner workings is crucial for evaluating their effectiveness.

Z-Score

Z-Score anomaly detection is a statistical method that measures how far a datapoint deviates from the mean in terms of standard deviations. It is calculated as:

$$Z = \frac{(X - \mu)}{\sigma}$$

Where X is the value, μ is the mean, and σ is the standard deviation of the feature. In fraud detection, Z-scores are used to flag anomalous transactions, typically those with a Z-score > 3, under the assumption that most legitimate transactions will fall within a normal distribution (Chandola et al., 2009).

This method is fast and interpretable but has significant limitations in high-dimensional spaces where data often deviates from normality. Additionally, its reliance on linear distance from the means makes it vulnerable to outliers in skewed or non-Gaussian distribution (Aggarwal, 2013).

Isolation Forest

iForest is an unsupervised anomaly detection method that isolates outliers through recursive partitioning. Rather than modelling normal behaviour, it assumes anomalies are "few and different" and are easier to isolate than inliers (Liu et al., 2008).

The algorithm builds random trees where each split isolates a portion of the data. The path length required to isolate a point becomes a proxy for anomaly score; the shorter the path, the more anomalous the point. iForest has a linear time complexity, performs well on high-dimensional datasets, and does not rely on distance or density metrics, which makes it highly efficient and scalable (Hariri et al., 2018).

iForest avoids the pitfalls of assumptions about feature distribution, unlike methods such as Z-score.

Naïve Bayes

NB is a probabilistic classifier based on Bayes' Theorem an assumes conditional independence between features given a class label. The model computes the posterior probability $P(C|X)$, where C is the class (e.g. fraud), and X represents the transactions features.

The model computes:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

Despite its simplicity, NB performs surprisingly well in high-dimensional spaces and imbalanced datasets (Zhang, 2004). The Gaussian NB variant, used in this project assumes each features follows a normal distribution; a reasonable approximation of PCA-transformed data.

However, NB often suffers from overconfidence in predictions when the assumption made is not held (Rish, 2001), in the context of fraud detection, it may prioritise recall at the expense of precision, resulting in a large quantity of false positives.

Support Vector Machine

SVMs are supervised learning models that aim to find the optimal hyperplane that best separates data points from different classes. SVM maximises the margin—the distance between the hyperplane and closest support vectors from each class (Cortes & Vapnik, 1995).

SVMs use kernel functions to handle non-linear relationships. The Radial Basis Function (RBF) kernel, used in this project, maps data into a higher-dimensional space, enabling the SVM to separate classes that are not linearly separable. This makes SVMs particularly effective for fraud detection, where decision boundaries can be highly non-linear and class imbalance is prevalent.

Although computationally expensive on large datasets, SVMs remain robust, accurate, and well-suited to high-dimensional feature spaces.

Model Implementation & Results

Implementation Notes

Each of the four models described in the previous section were implemented using Python, primarily leveraging the scikit-learn library for classification and anomaly detection tasks. The project was developed in VS Code, in a structured Jupyter Notebook environment, and each model followed a consistent pipeline: preprocessing, fitting, prediction and evaluation using performance metrics suitable for imbalanced data.

To ensure a fair comparison, all models were trained on the same stratified training set and evaluated on the same test set, which had been created during preprocessing. The Z-Score model was manually computed using NumPy, applying a threshold of 3 standard deviations. IForest used n_estimators and a contamination rate to help reflect real-world fraud ratios. NB required minimal tuning and was evaluated with inbuilt probability outputs. SVM was implemented with an RBF kernel (Schölkopf & Smola, 2001), it's class_weight parameter was defined as balanced to further help the model counteract skewed class distributions. Due to computational costs, it was strained on a 20,000 row subset of the data and evaluated on the full data test set.

Performance was measured using the following metrics:

- Precision: The proportion of predicted frauds that were fraudulent.
- Recall: The proportion of actual frauds that were correctly identified.
- F1-Score: The harmonic mean of precision and recall, weighing false positives and false negatives.
- AUC: A measure of a model's ability to distinguish between classes.
- AP: A summary of the precision-recall curve, useful for imbalanced datasets.

Results Summary

The following table summarizes the performance of all four models on the test set:

Model	Precision	Recall	F1-Score	AUC	Avg. Precision
Z-Score	0.011050	0.878378	0.021825	NaN	NaN
Isolation Forest	0.325000	0263514	0.291045	0.938790	0.214264
Naïve Bayes	0.060807	0.804054	0.113064	0.955037	0.081252
Support Vector Machine	0.924528	0.331081	0.487562	0.943308	0.696093

Table 1 - Result Summary Table

Z-Score flagged almost every fraud case, but with extremely low precision, making it impractical for real deployment. IForest achieved a moderate balance, outperforming Z-Score, yest lacking the precision required for real-time fraud detection systems. NB demonstrated the highest recall, identifying most fraud cases but generating numerous false positives. SVM provided the best performance overall especially in precision and F1-Score, which are critical when minimizing false alarms in business environments.

Each metric additionally had a bar-chart created for further help visualising the difference (see Appendix B).

The implementation also included visual outputs such as confusion matrixes, ROC curves, and precision-recall plots to support interpretation of numerical scores, discussed in the next section.

Visualization and Analysis

To complement the evaluation of each model, several visualization techniques were used to help interpret performance of each model. These visual outputs highlight patterns in predictions, reveal strengths and weaknesses in classification and provide insights into how each model balances recall and precision.

Confusion Matrix

Confusion Matrixes were generated of each model as heatmaps to visualise the count of true positives, false positives, true negatives and false negatives.

Z-Score (see Figure 3) predictied nealy all actual frauds correctly but misclassified thousands of legitimate transactions as fraudulent, evident by the many false positives. While iForest (Figure 4) had a much more balanced matrix correctly indentifying some frauds, but maintained a lower false positive count. NB (see Figure 5) showed similarly high recall, but also suffered from a large false positive count; similar to Z-Score. SVM (see Figure 6) yielded the most desirable heat map, very little false positives, with a slightly less acurage positive fraud count.

ROC Curve

ROC (Receiver Operating Characteristic) curves for NB, iForest, and SVM were plotted together and separately to compare their ability to distinguish between classes.

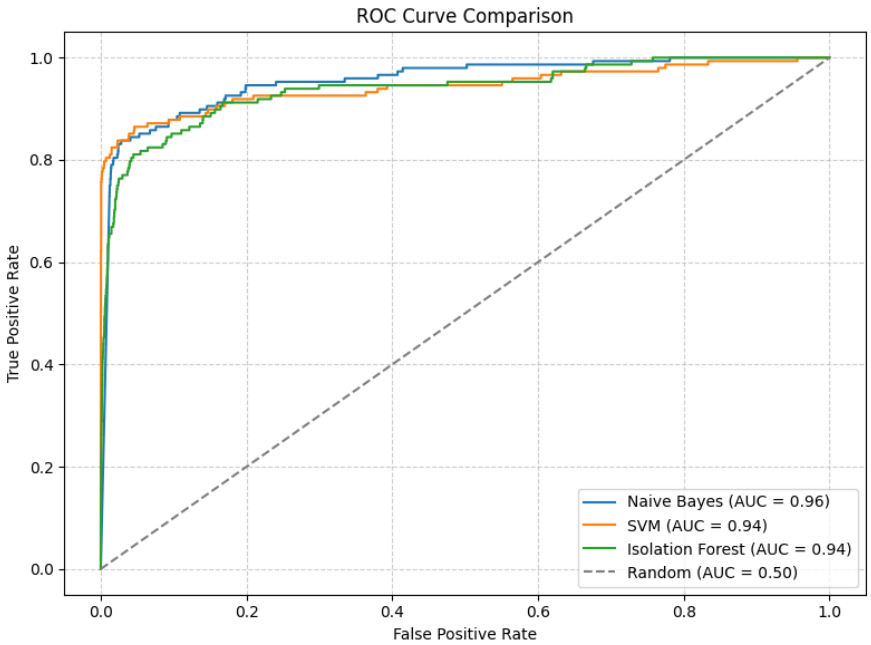

Figure 1 - ROC comparison chart

NB (see Figure 7) had the highest AUC score (0.96), indicating excellent separation between fraudulent and legitimate transactions. Both SVM (see Figure 8) and iForest (see Figure 9) followed behind closely with an AUC score of 0.94, demonstrating strong discriminatory power. While AUC is a valuable metric it can be misleading in highly imbalanced datasets where the model may perform well on negatives but fail to catch positives.

Precision-Recall Curve Comparison

Given the class imbalance in the dataset, precision-recall curves offer a more insightful visualisation of a model's effectiveness in identifying fraud.

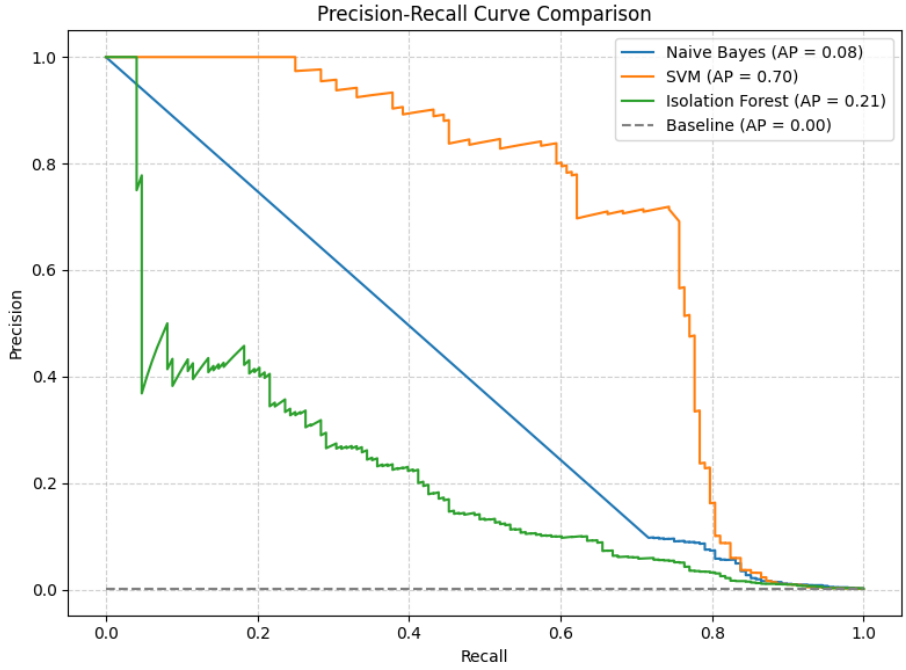

Figure 2 - Precision-Recall comparison chart

In this chart we can see SVM (see Figure 10) clearly outperforms the other models in terms of AP (0.70), indicating a strong balance in fraud detection. IForest (see Figure 11) performed moderately (AP = 0.21), detecting some frauds with acceptable precision. NB (see Figure 12), while highly sensitive achieved only 0.08 AP, again highlighting its tendency to overpredict fraud.

These curves reinforce the conclusion that SVM provides the most reliable results, however while not having the highest true fraud rate, has a lower false positive and false negative rate compared to other models.

Application and Real-World Relevance

Fraud detection systems are a critical part of modern financial infrastructure. In practice, these systems must operate at scale, with speed, and high accuracy to minimise financial losses and customer disruption. The models tested in this project offer distinct strengths that could be applied in various ways depending on business needs and technical constraints.

The Z-Score and iForest models are well-suited for unsupervised anomaly detection in scenarios where labelled data is scarce of unavailable. These could be deployed as early-stage

filters, flagging unusual activity for further review or triggering secondary model layers for verification.

NB, with its high recall, could be useful in environments where catching every potential fraud is the priority, such as post-transactional review. However, high false positive rates could generate unnecessary operational costs if not used with additional methods.

SVM, by contrast, strikes a more practical balance between recall and precision. SVMs are particularly valuable in real-time fraud detection pipelines, where too many false alarms can overwhelm fraud departments or diminish user trust.

Conclusion

This project explores the application of data mining techniques in the challenging environment of credit card fraud, using a highly imbalanced, real-world dataset. Four models were implemented and evaluated: Z-Score anomaly detection, iForest, NB classification and SVMs.

Each model demonstrates unique strengths, which could be beneficial in the right circumstances. Among all, SVM delivered the most balanced and reliable performance; achieving the highest F1-Score and precision, making it the most viable option for real-time fraud detection in financial systems.

The combination of statistical, machine learning and anomaly detection approaches offered a comprehensive overview of the problem. This comparative framework can help guide future implementation strategies in business environments where the cost of fraud must be weighed against the cost of incorrect classification.

Future work could see experiments with deep learning techniques, combining techniques such as SVM and Z-Score, or even generating live data to be fed to the systems to see how they cope with true real-time data.

Appendix A

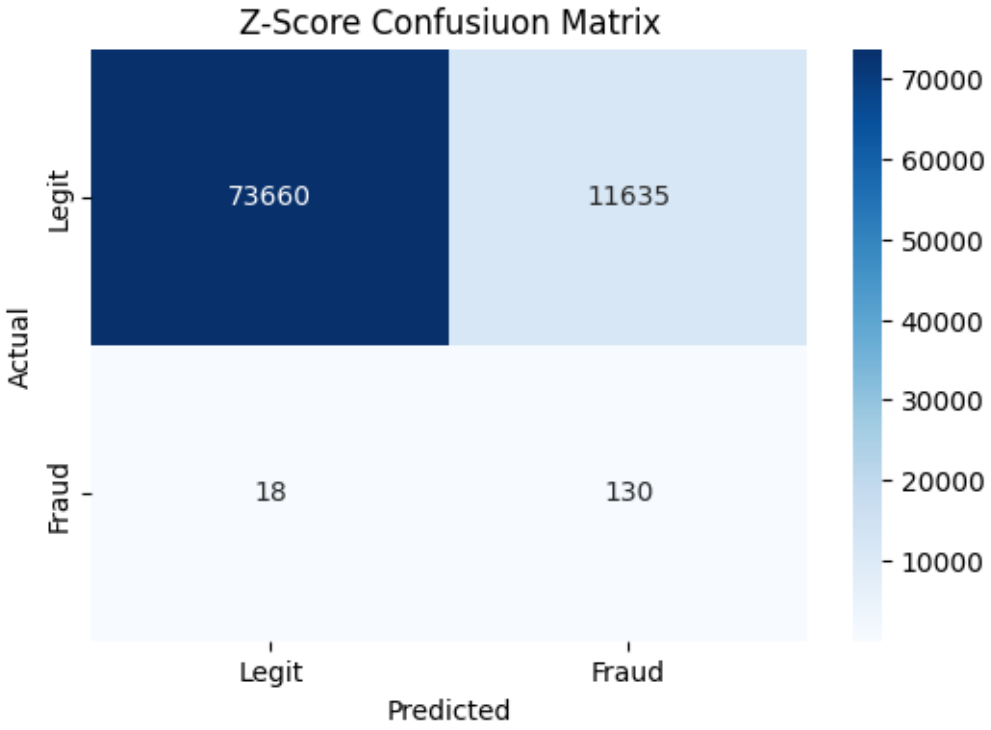

Figure 3 - Z-Score Heatmap

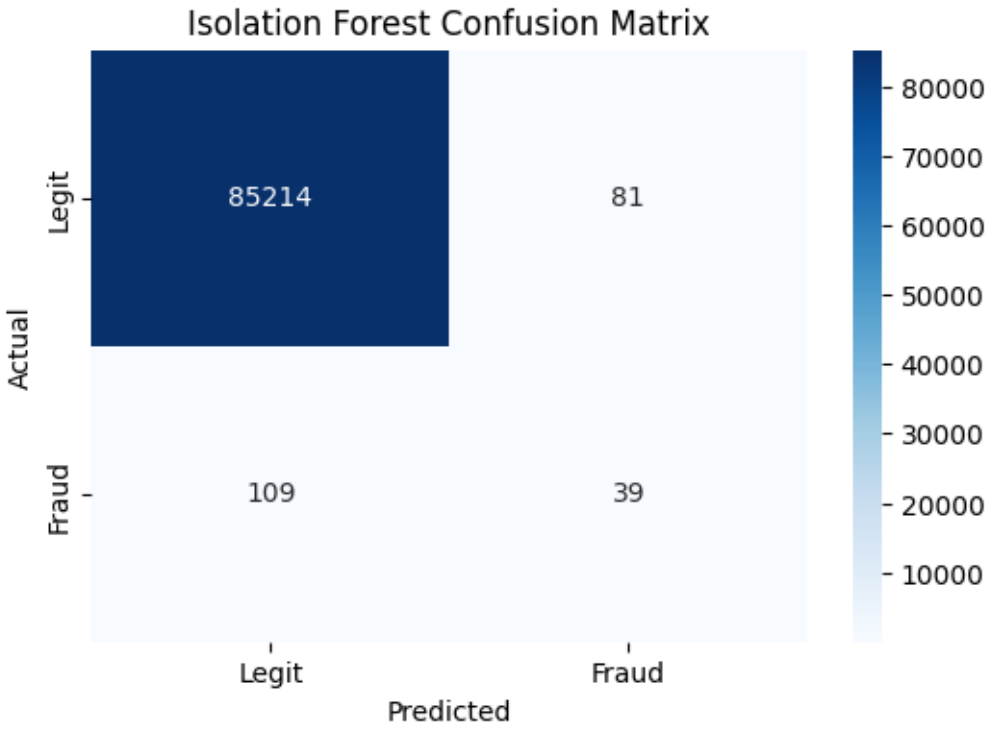

Figure 4 - iForest Heatmap

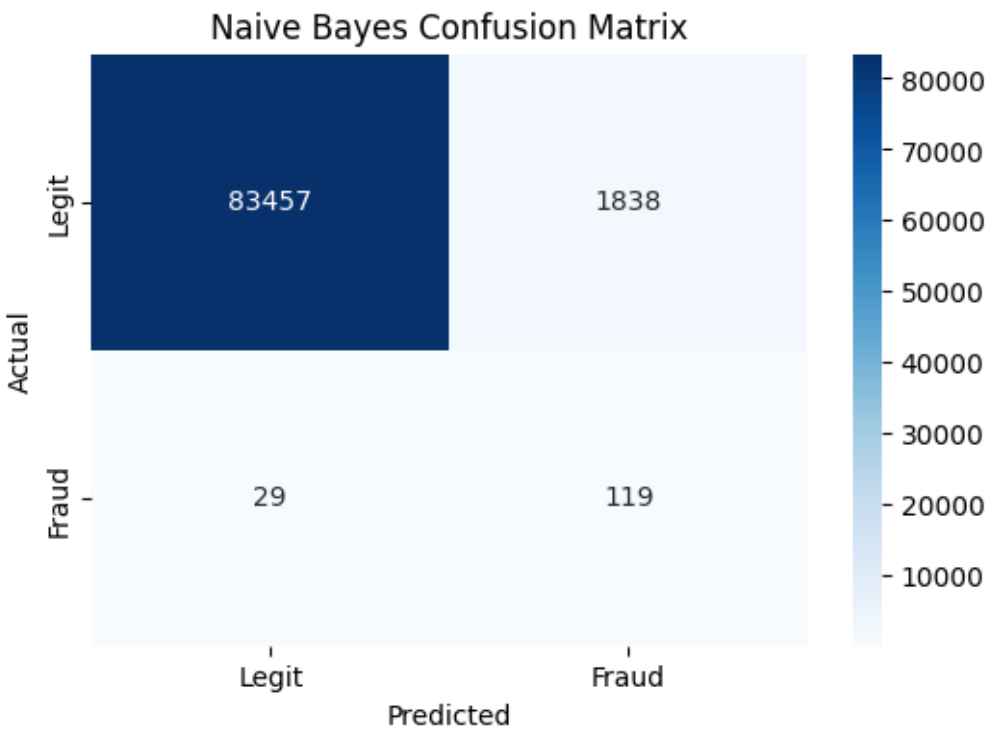

Figure 5 - NB Heatmap

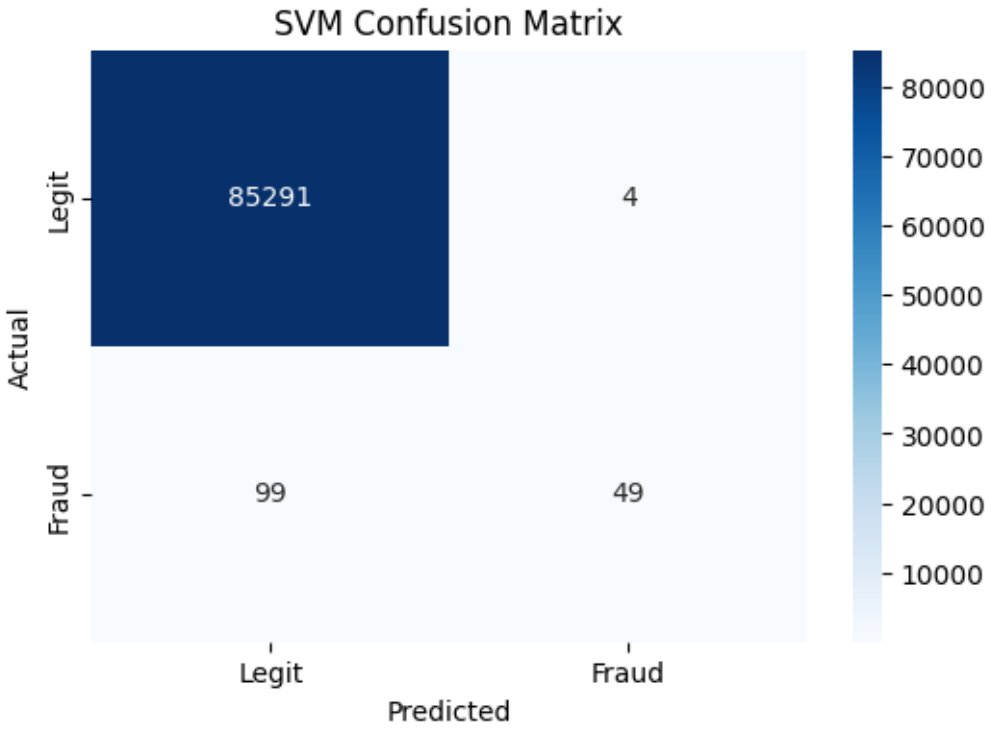

Figure 6 - SVM Heatmap

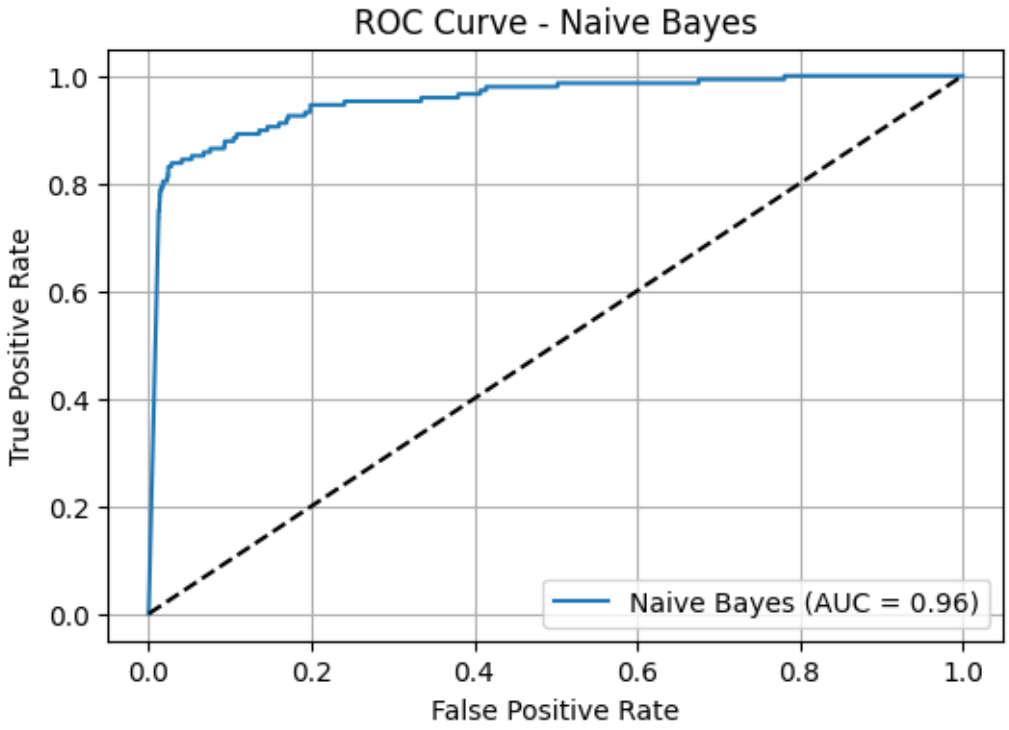

Figure 7 - NB ROC curve

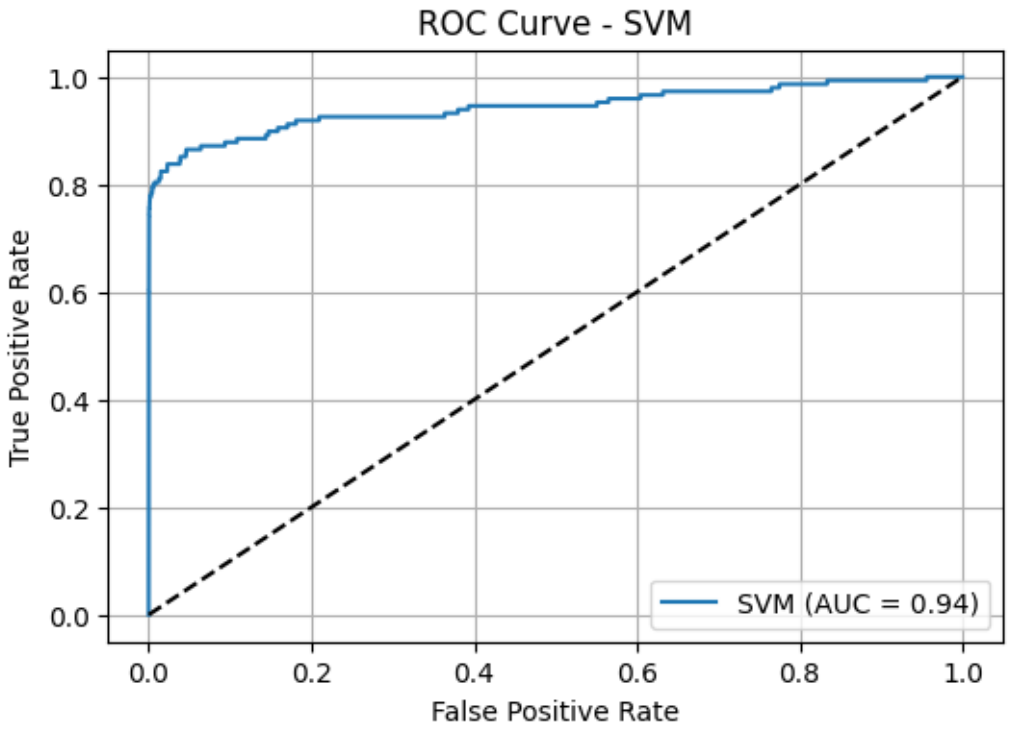

Figure 8 - SVM ROC curve

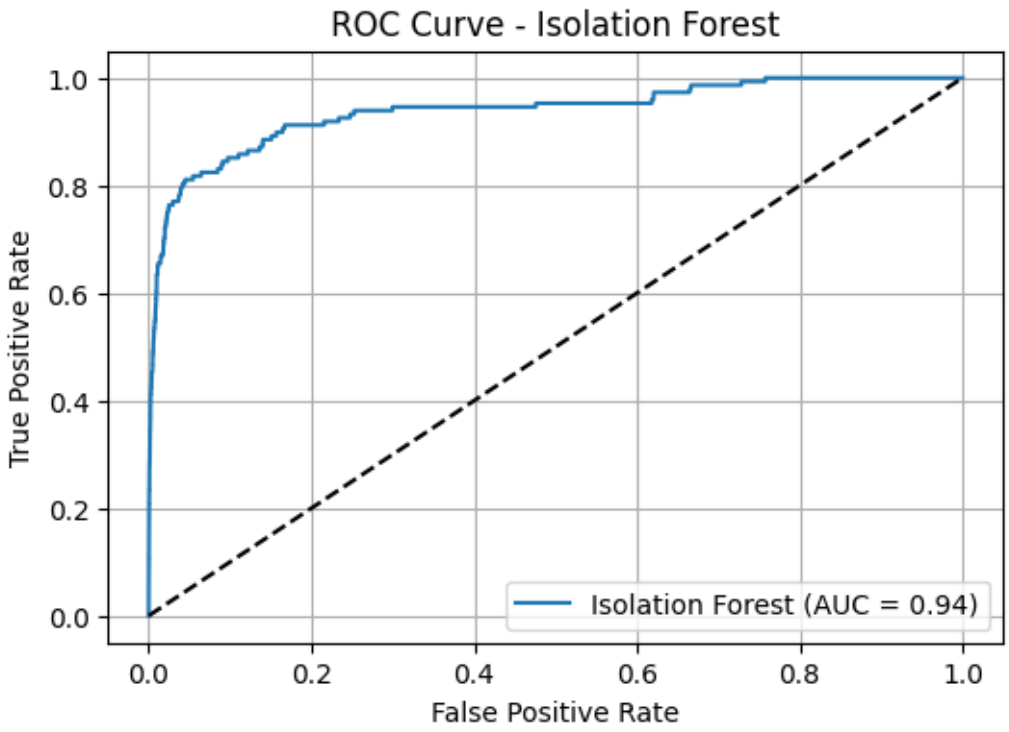

Figure 9 - iForest ROC curve

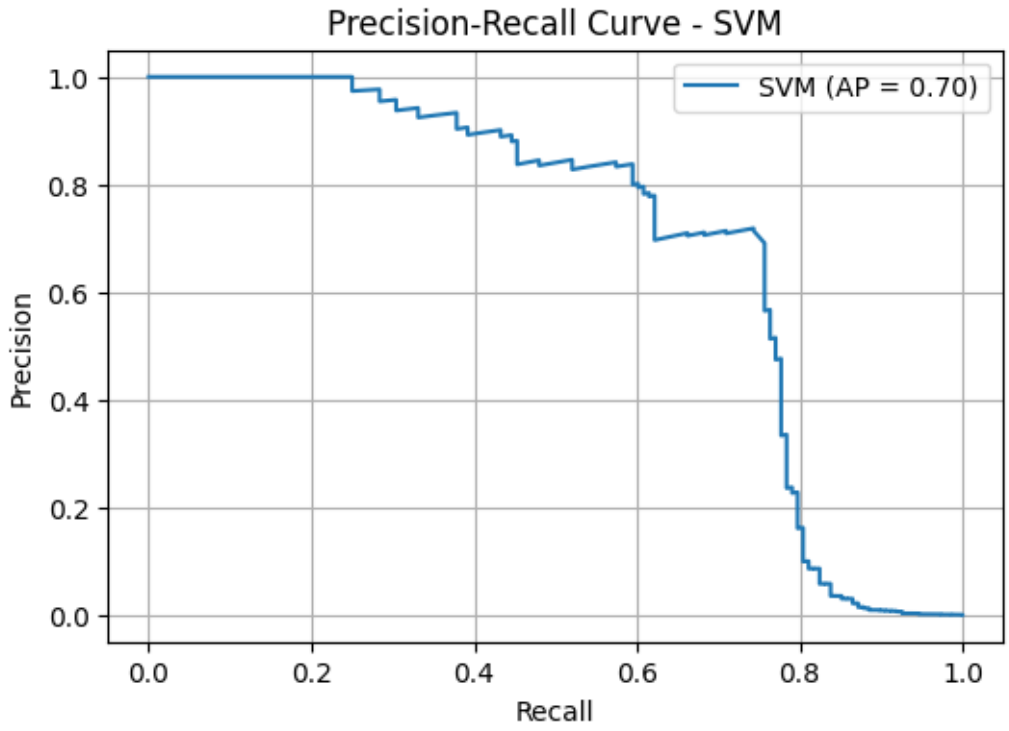

Figure 10 - SVM precision-recall curve

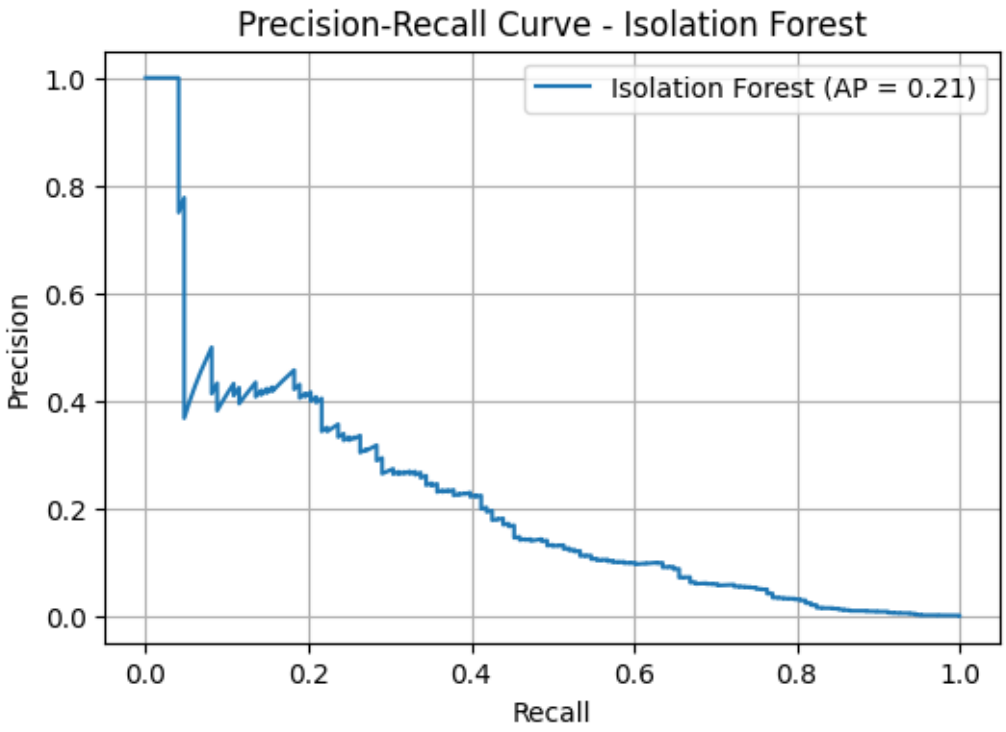

Figure 11 - iForest precision-recall curve

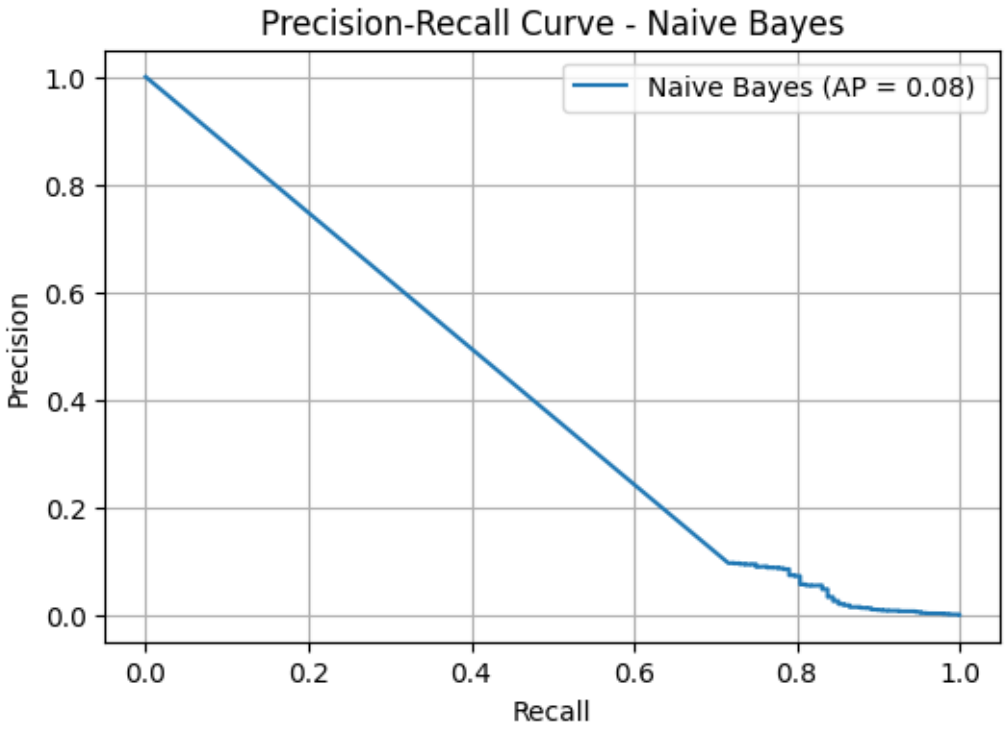

Figure 12 - NB precision-recall curve

Appendix B

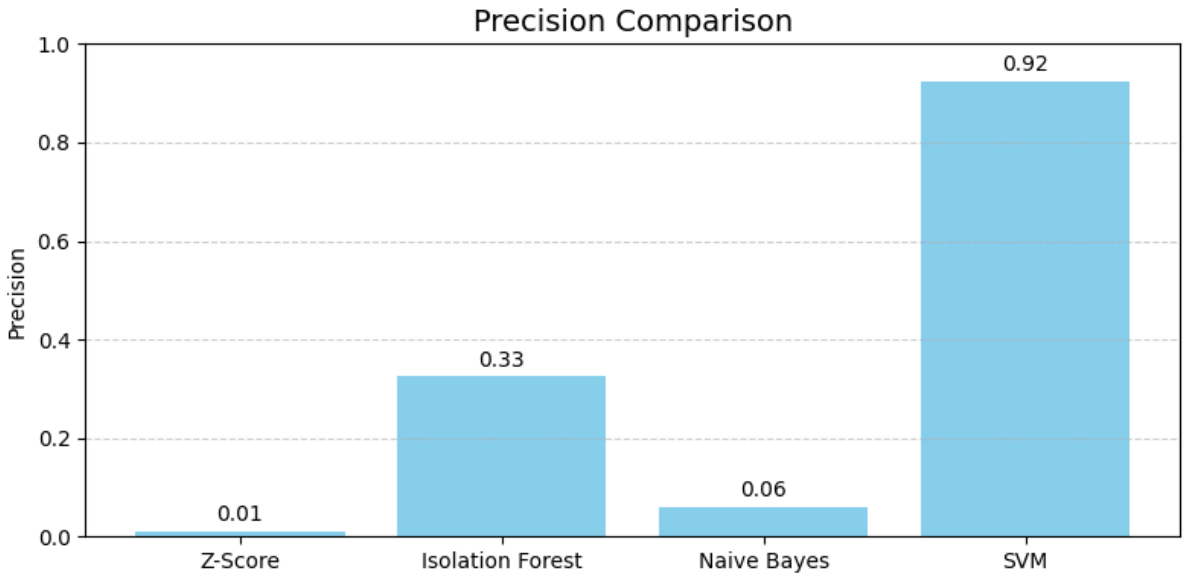

Figure 13 - Precision Comparison Chart

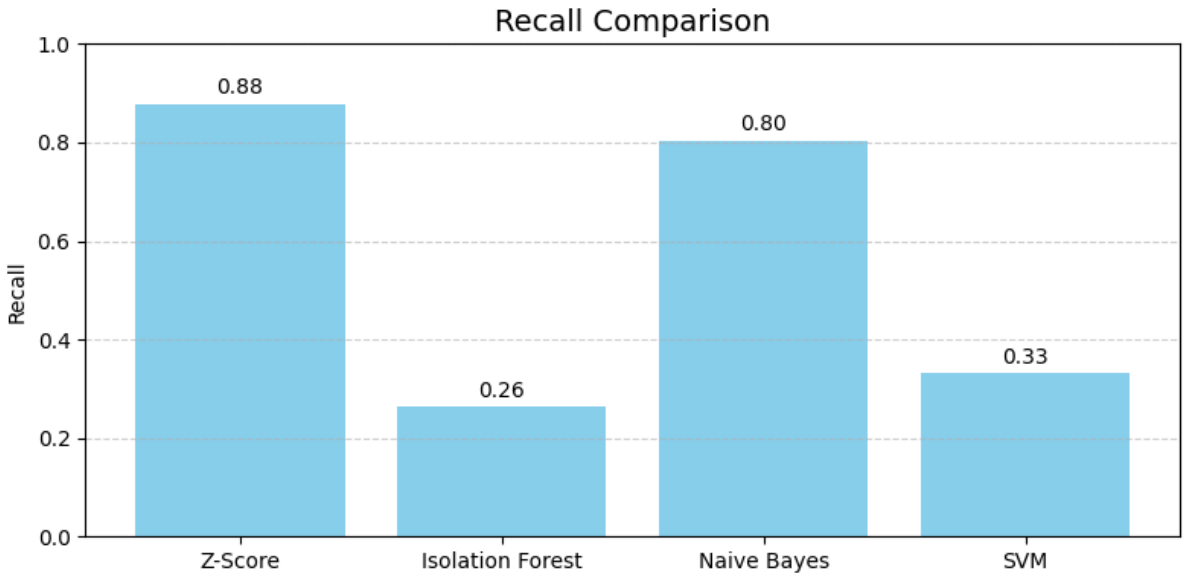

Figure 14 - Recall Comparison Chart

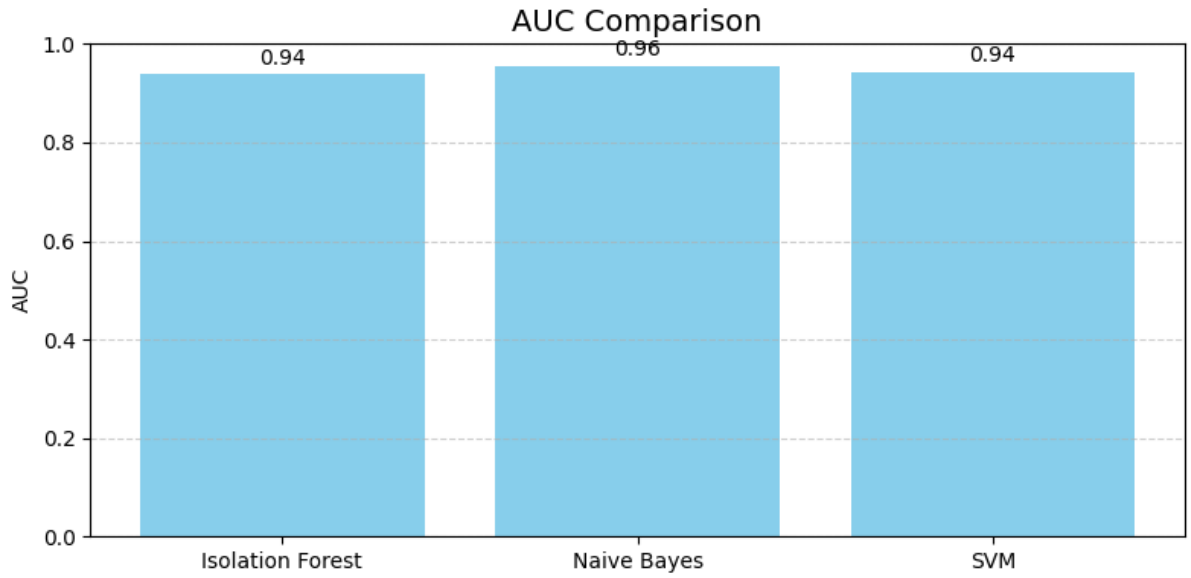

Figure 15 - AUC Comparison Chart

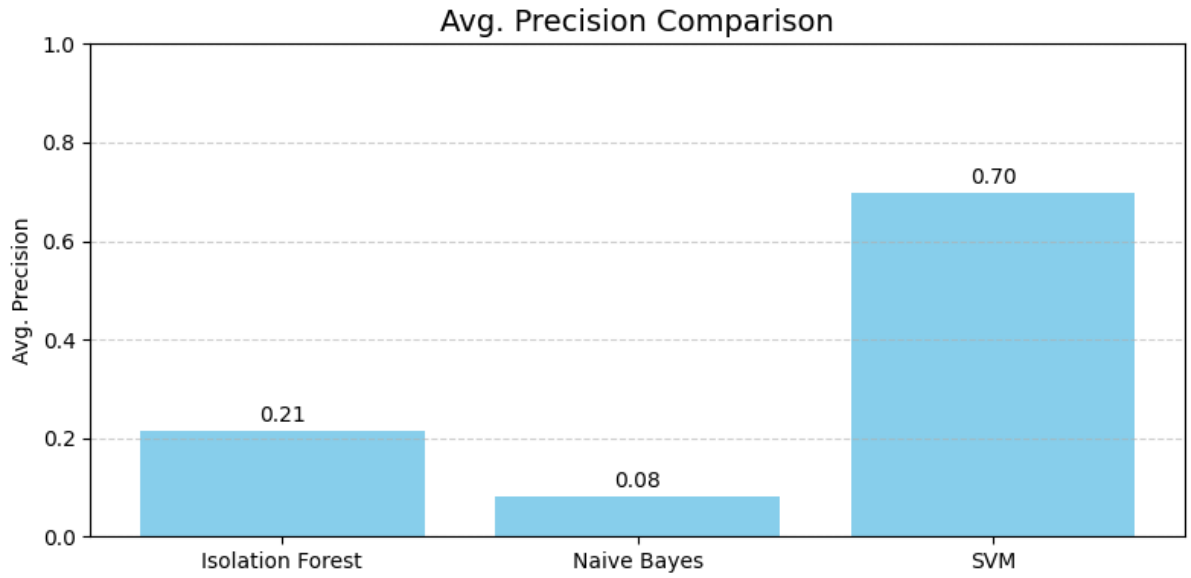

Figure 16 - AP Comparison Chart

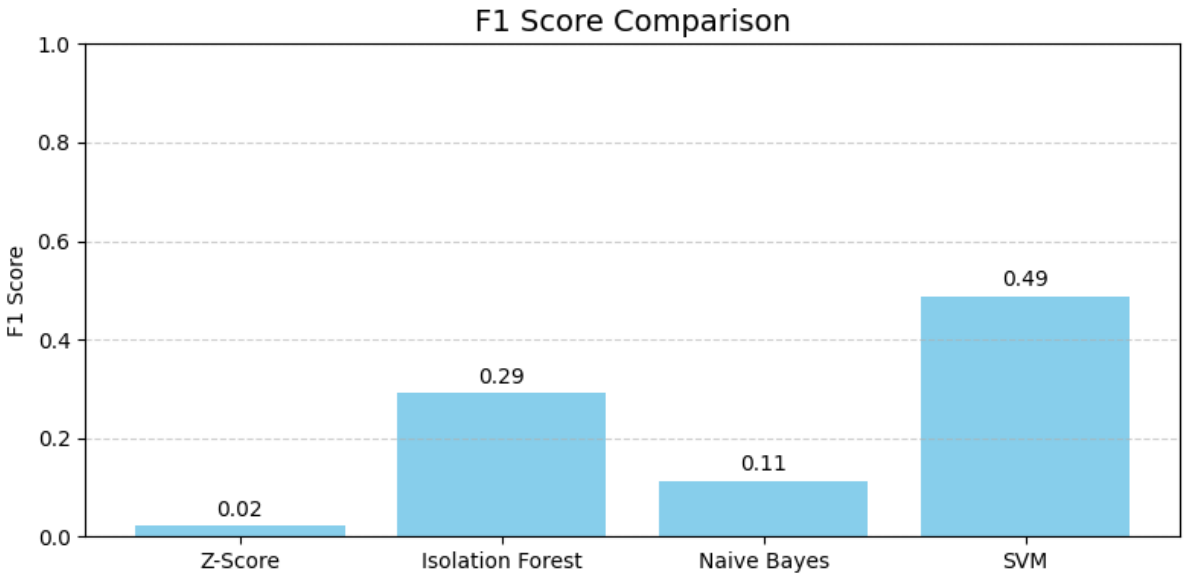

Figure 17 - F1 Comparison Chart

Bibliography

Aggarwal, C. C. (2013). *Outlier Analysis*. Springer. https://doi.org/10.1007/978-1-4614-6396-2

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey. *ACM Computing Surveys*, *41*(3), 1-58. https://doi.org/10.1145/1541880.1541882

Cherif, A., Badhib, A., Ammar, H., Alshehri, S., Kalkatawi, M., & Imine, A. (2023). Credit card fraud detection in the era of disruptive technologies: A systematic review. *Journal of King Saud University - Computer and Information Sciences*, *35*(1), 145-174. https://doi.org/https://doi.org/10.1016/j.jksuci.2022.11.008

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273-297. https://doi.org/10.1007/BF00994018

Hariri, S., Kind, M. C., & Brunner, R. J. (2018). *Extended Isolation Forest*. https://arxiv.org/abs/1811.02141

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. Proceedings of the 2008 Eighth IEEE International Conference on Data Mining,

Rish, I. (2001). An Empirical Study of the Naive Bayes Classifier. IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence,

Schölkopf, B., & Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press. https://mitpress.mit.edu/books/learning-kernels

ULB, M. L. G.-. *Credit Card Fraud Detection* (Version 3). https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Zhang, H. (2004). The Optimality of Naive Bayes. Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference,

Table of Figures and Tables