Uppgift 3: Förbättra anomalidetektering

### 3.1 Analys av grundmodellen och dess resultat

Grundmodellen implementerar en effektiv pipeline, vilket betyder en strukturerad sekvens av steg som utförs för att förbereda, bearbeta och analysera data, för anomalidetektering med PCA och en SVM-klassificeringsmodell. Först hanteras dataförberedelse genom att rensa bort NaN-värden, i *ANOMALY*-kolumnen.

df = pd.read\_csv('first\_10000\_rows.csv')

if df['ANOMALY'].isna().sum() > 0:

print(f"NaN-värden hittades i 'ANOMALY'-kolumnen: {df['ANOMALY'].isna().sum()} rader.")

# Ta bort rader med NaN-värden i 'ANOMALY'-kolumnen

df.dropna(subset=['ANOMALY'], inplace=True)

print("NaN-värden har tagits bort.")

else:

print("Inga NaN-värden i 'ANOMALY'-kolumnen.")

Endast numeriska variabler väljs, och irrelevanta kolumner som *ID* och *FLOW\_ID* exkluderas. Data standardiseras sedan med *StandardScaler*, vilket är nödvändigt för att PCA och SVM ska fungera korrekt.

include=['int64', 'float64']).columns

numerical\_columns = numerical\_columns.drop(['ID', 'FLOW\_ID', 'ANOMALY'])

X = df[numerical\_columns]

scaler = StandardScaler()

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

PCA används för att reducera datan till tre dimensioner, vilket minskar beräkningskomplexitet och multikollinearitet, men det kan också innebära viss informationsförlust.

pca = PCA(n\_components=3)

pca\_data = pca.fit\_transform(X\_scaled)

Datan delas upp i 70% träning och 30% test för utvärdering. För att hantera obalans i träningsdatan används SMOTE, som syntetiskt ökar mängden data i den underrepresenterade klassen och förbättrar modellens förmåga att identifiera anomalier. En SVM-modell med en RBF-kernel tränas, vilken är effektiv för att upptäcka icke-linjära relationer.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(pca\_data, y, test\_size=0.3, random\_state=42)

smote = SMOTE(random\_state=42)

X\_train\_smote, y\_train\_smote = smote.fit\_resample(X\_train, y\_train)

svm = SVC(kernel='rbf', C=1.0)

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

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

Resultaten analyseras med en *Confusion Matrix* och *Classification Report*. Confusion Matrix ger en visuell överblick av klassificeringsfel, medan metrik som precision, recall och F1-score i *Classification Report* mäter modellens prestanda. Recall är särskilt viktig för anomalidetektering, där falska negativa kan vara kritiska.

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

report = classification\_report(y\_test, y\_pred, target\_names=['Normal', 'Anomaly'], output\_dict=True)

report\_df = pd.DataFrame(report).transpose()

Efter körning av grundmodellen så kunde man konstatera att modellen totala noggrannhet ligger på 88.1%, vilket betyder att 88.1% kvalificerades korrekt. Grundmodellen uppnår F1-score på 78.8% för normal klassen och 91.7% för anomaly klassen, vilket kan antydda att modellen har svårigheter att identifiera korrekt normalklassen.

Table 1Classification Report av Grundmodellen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classification Report (Precision, Recall, F1-Score): | | | | |
|  | precision | recall | f1-score | support |
| Normal | 0.707965 | 0.888889 | 0.788177 | 90.000000 |
| Anomaly | 0.959677 | 0.878229 | 0.917148 | 271.000000 |
| accuracy | 0.880886 | 0.880886 | 0.880886 | 0.880886 |
| macro avg | 0.833821 | 0.883559 | 0.852663 | 361.000000 |
| weighted avg | 0.896924 | 0.880886 | 0.884995 | 361.000000 |

### Förbättringsmöjligheter

* **Alternativa algoritmer**

I studien testades Random Forest och XGBoost, två välkända algoritmer för anomalidetektering. Random Forest, en ensemblemetod baserad på beslutsträd, visade sig vara robust mot överanpassning och outliers. XGBoost, en effektiv implementation av Gradient Boosting, utvärderades också. Resultaten visade att Random Forest uppnådde 100% noggrannhet och F1-score för både Normal och Anomaly-klasserna. XGBoost presterade något lägre med 99,7% noggrannhet och F1-score på 99,4% för Normal samt 99,8% för Anomaly. Random Forest visade sig vara något överlägsen i detta fall.

. A graph of different colored squares

Description automatically generated

Figure 1Confusion Matrix för SVM, Random Forest och XGBoost Modeller

* **Dataförbättring**

För att förbättra modellens prestanda användes SMOTE (Synthetic Minority Over-sampling Technique) för att balansera datasetet. Parametern k\_neighbors i SMOTE styr antalet närliggande datapunkter som används för att generera syntetiska exempel för minoritetsklassen. Standardvärdet för k\_neighbors är 5, men efter att ha testat värdena 3, 7 och 10 visade sig 10 ge den bästa noggrannheten på 92,2%.Vidare förbättrades resultaten genom att öka antalet dimensioner i PCA från 3 till 5 via parametern n\_components. Detta ledde till en betydande prestandahöjning med en noggrannhet på 99,7% samt F1-score på 99,4% för klassen Normal och 99,8% för klassen Anomaly. Dessa justeringar visar på vikten av både datasetbalansering och lämplig dimensionell reduktion för optimal modellprestanda.

A graph of different colored squares

Description automatically generated

Figure 2 Confusion Matrix för SVM, Random Forest och XGBoost efter förbättring av Dataförberedelse