# Machine-Learning-Based Electrical Signal Analysis for Intrusion Detection in Industrial Systems

In diesem Schritt, habe ich folgende ML-Methoden implementiert und evaluiert:

- Support-Vector-Maschine (SVM)
- Local Outlier Factor (LOF)
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Zur Implementierung jedes dieser Verfahren wurden verschiedene Initialisierungsparameter erprobt und evaluiert. Außerdem wurden für jede dieser Methoden unabhängig voneinander überprüft, welche Vorverarbeitungsschritte der Eingabedaten zu einem besseren Gesamtergebnis führen. Bei der Vorverarbeitung der Daten wurden unterschiedliche Methoden zur Skalierung der Eingabedaten, sogenannte Scaler, evaluiert. Im Folgenden werden die evaluierten Scaler vorgestellt:
**Kein Scaler:** Die Daten werden nicht vorverarbeitet.
**Standard Scaler:** Entfernt den Mittelwert und skaliert die Daten auf Einheitsvarianz.
**MinMaxScaler:** Skaliert den Datensatz neu, sodass alle Feature-Werte im Bereich [0, 1] liegen.
**MaxAbsScaler:** Skaliert so, dass die Trainingsdaten im Bereich [-1,1] liegen, indem jeder Datenpunkt durch den Maximalwert geteilt wird.
**RobustScaler:** Die Zentrierungs- und Skalierungsstatistik vom RobustScaler basiert auf Perzentilen und wird daher nicht von einigen wenigen sehr großen Randausreißern beeinflusst.
**PowerTransformer:** Wendet eine Leistungstransformation auf jedes Feature an, um die Daten Gauß-ähnlicher zu machen, um die Varianz zu stabilisieren und die Schiefe zu minimieren. Derzeit werden die Yeo-Johnson- und Box-Cox-Transformationen unterstützt und der optimale Skalierungsfaktor wird bei beiden Methoden über die Maximum-Likelihood-Schätzung bestimmt.
**QuantileTransformer:** Wendet eine nichtlineare Transformation an, sodass die Wahrscheinlichkeitsdichtefunktion jedes Merkmals auf eine gleichmäßige oder Gaußsche Verteilung abgebildet wird.
**Normalizer:** Skaliert den Vektor für jede Probe neu, um unabhängig von der Verteilung der Proben eine Einheitsnorm zu erhalten.

Ich habe die Evaluationsergebnisse jedes Scalers verglichen, um den Scaler auszuwählen, der das beste Ergebnis für die Erkennung von Anomalien liefert.

Abschließend möchte ich noch einen Einblick bzgl. der zur evaluierenden Initialisierungsparameter jeder Methode geben:

Für das SVM habe ich den Parameter "nu" geändert, der eine Obergrenze für den Anteil der Trainingsfehler und eine Untergrenze für den Anteil der Unterstützungsvektoren darstellt. Durch Verringern dieses Parameters erhalten wir bessere Ergebnisse für die Erkennung vom Normalverhalten, aber schlechte Ergebnisse für die Erkennung von Anomalien. Durch Erhöhen dieses Parameters können wir bessere skalierungsabhängige Ergebnisse erzielen.

Für LOF muss mit der Anzahl der Nachbarn experimentiert werden, um die Dichteabweichung einer Probe zu berechnen. Durch Ändern dieses Parameters werden je nach Scaler unterschiedliche Verhaltensweisen beobachtet, was bei einigen Scalern zu guten Ergebnissen bei einem niedrigen Wert der Anzahl der Nachbarn führen kann, während bei anderen Scaler eine große Anzahl von Nachbarn erforderlich ist, um eine korrekte Anomalieerkennung zu erzielen.

Für DBSCAN experimentierte ich mit dem Parameter, der den maximalen Abstand zwischen zwei Samples darstellen, damit sie in demselben Sample berücksichtigt werden, und mit dem Parameter, welches die maximale Anzahl von Samples in einer Nachbarschaft darstellt, damit dies als Kernpunkt betrachtet wird. Diese beiden Parameter wurden so gewählt, dass DBSCAN ein einzelnes Cluster

erkennt, wobei der eine das normale Verhalten des Prozesses neu gruppiert und alle Ausreißer die Punkte sind, die eine Anomalie erkennen.

Im letzten Schritt werde ich das Schreiben der Masterarbeit abschließen.