Analyse

La détection d’anomalie dans le processus se fait via l’apprentissage du comportement normal en utilisant le Machine Learning. Ces dernières années, le Machine Learning a démontré que l’on pouvait détecter des motifs dans des images, pour détecter des objets ou des visages, par exemple, ou bien pour la détection d’anomalie dans des signaux électriques en utilisant l’apprentissage par l’expérience.

J’ai sélectionné plusieurs méthodes qui se basent sur le unsupervised learning car il a l’avantage d’être plus simple à implémenter que le supervised learning. Les méthodes utilisant l’unsupervised learning ont aussi démontré leur capacité à être flexible dans leur apprentissage et peuvent donc être utilisé pour tout type d’attaque sur processus.

Ces méthodes se regroupent en deux types de détection d’anomalie, la détection et la prévision basée sur des séries temporelles. Dans ce rapport, j’utiliserai les définitions de détection et prévision basées sur la taxonomie de [3] qui définit la détection d’anomalie comme une détection se basant sur les données que nous avons et que nous comparons pour détecter une erreur. La prévision basée sur des séries temporelles est définit comme une méthode qui va prédire les valeurs des prochaines séries temporelles et les comparer avec les séries temporelles observées.

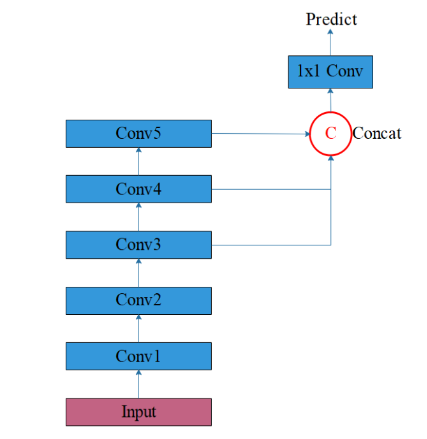
1. CNN

Le Convolutional Neural Network (CNN) est une méthode se basant sur les réseaux de neurones et la convolution pour apprendre des patterns et des motifs dans des signaux ou des images. Lors d’un CNN, la valeur de chaque point du prochain layer est calculé à partir des valeurs voisines de ce point du layer précédent. Ce fonctionnement est représenté par la figure « ».

Figure

Par exemple si on a pour kernel [-1,1,-1], la valeur du point i-ème point sera (i-1)\*(-1)+i+(i+1)\*(-1). Les valeurs du kernel évolueront bien sûr au cours de l’apprentissage dans le but de réduire la fonction de coût.

Le CNN peut aussi bien être utilisé pour de la détection que pour de la prévision. Dans mon cas, j’utiliserai le même réseau de neurone que dans [4] qui utilise leur méthode pour de la prévision. Dans [4], ils utilisent un CNN composé de 5 convolutional layers avec un kernel de taille 3 et un kernel de taille 1. Le réseau de neurone est représenté sur la figure « ».



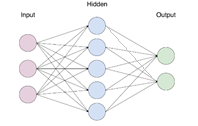
Figure

Les layers 3, 4 et 5 sont concaténés pour éviter que le réseau de neurones ne overfit.

En input du réseau de neurones sont injectés les séries temporelles à partir du temps t jusqu’à t+T-1 de chaque sensors et actuators et en output, nous avons la prévision des séries temporelles à partir du temps t+T jusqu’à t+2-1 pour chaque sensors et actuators. L’output est comparé avec les véritables valeurs des séries temporelles. La fonction de coût est calculée en utilisant le mean square error (MSE).

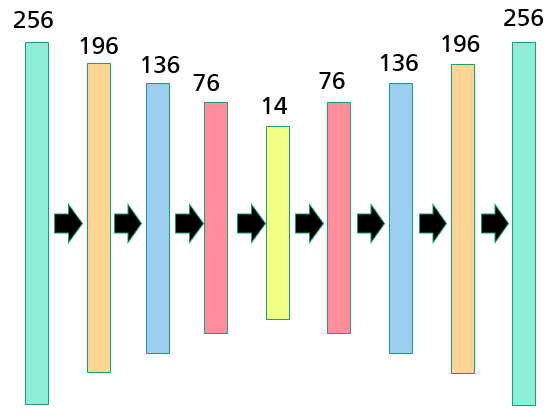
1. Autoencoder

L’autoencoder est un réseau de neurone utilisant la réduction de dimension pour connaître les caractéristiques de son input. L’autoencoder est un réseau de neurone qui est utilisé pour la détection d’anomalie. Chaque layer est relié au suivant en utilisant la fonction Fully Connected Layer, c’est-à-dire que chaque neurone d’un layer est relayé à tous les autres neurones du layer suivant comme représenté sur la figure « ».



Figure

La structure d’autoencoder utilisée est la même que dans [5] avec 9 Fully connected Layers de dimensions 256, 196, 136 ,76 et 14. Le principe de fonctionnement est représenté sur la figure « ».



Figure

En input de l’autoencoder, on envoie pour chaque sensor une fenêtre de 40 valeurs et l’autoencoder a pour objectif de restituer ces fenêtres en évitant l’overfit sinon l’autoencoder serait équivalent à la fonction unité. La fonction de coût est calculée en utilisant le mean square error.

1. LSTM
2. SVM

SVM kategorisiert die Signale in zwei Klassen. Eine Klasse besteht aus allen richtigen Werten der Signale und die andere besteht aus aller falschen Werter etwa den Anomalien.

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.

1. LOF

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.

1. DBSCAN

Density-Based Spatial Clustering of Applications with Noise

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.

[3] Grohmann, J.; Herbst, N.; Chalbani, A.; Arian, Y.; Peretz, N.; Kounev, S. A Taxonomy of Techniques for SLO Failure Prediction in Software Systems. Computers **2020**, 9, 10. https://doi.org/10.3390/computers9010010