Folie 1:

Guten Tag, ich bin Yorick LALY. Ich bin Masterstudent am KIT. Der Titel meiner Masterarbeit lautet „**Analyse der elektrischen Signale zur Angriffserkennung in industriellen Systemen basierend auf maschinellem Lernen**“. In dieser Masterarbeit beschreibe ich einen neuen Ansatz zur Anomalie Detektion. Ich detektiere Anomalien auf Basis der elektrischen Signale zwischen der SPS und den Sensoren und Aktoren eines industriellen Prozesses.

Folie 2:

Zuerst werde ich das System und das Schaltungsdesign vorstellen. Danach Stelle ich die Analyse der elektrischen Signale vor, die ich abgegriffen habe. Danach beschreibe ich die Lernmodelle die ich für meinen anomaliebasierten Ansatz ausgewählt habe. Zum Schluss stelle ich die Evaluation und Bewertung vor.

Folie 3:

Heutzutage sind industrielle System immer mehr vernetzt darum ist es einfacher für externe Menschen um die Kommunikation zu beobachten und um das System zu verstören. Internen Angriffe sind möglich. Z.B. ein Mitarbeiter kann das System verstören mit einem USB-Stick der ein Angriff Code enthalt.

Beispielsweise, der Stuxnet Angriff in Iran der, nach Analyse des Systems, die Kontrolle des PLC hat und vielen Maschinen zerstören.

Folie 4:

Der klassische Ansatz um Cyber-Angriffe in einem industriellen System zu detektieren basiert auf dem Einsatz eines NIDS. Das NIDS detektiert Anomalien und Cyber-Angriffe anhand der übertragenen Netzwerkpakete am Switch. Cyber-Angriffe, wie Stuxnet oder Irongate, können mit einem NIDS **nicht** erkannt werden. Diese Angriffe verändern den SPS-Code ohne dass wir eine Änderung in den Netzwerkpakten sehen.

Folie 5:

Mit unserem System SAAD, abgreifen und analysieren wir die Signale der Sensoren und Aktoren direkt aus den Prozess um eine Veränderung des Normalverhalten zu detektieren. Somit können auch die Angriffe, wie bei Stuxnet und Irongate erkannt werden.

Folie 6:

Wir testen SAAD an einen Prozess des Fraunhofer IOSBs der Objekte nach ihrer Beschaffenheit sortiert. SAAD soll leicht zu integrieren sein so dass man den Ansatz auch für andere Systeme benutzen kann. Es soll auch nicht das ganze System stören. Die Lernmodellen lernen das Normalverhalten des Prozesses.

Folie 7:

Auf das Bild kann man das ganze System mit SPS Buskoppler und an rechts den Prozess sehen.

Folie 8:

Der Betrieb des Prozesses ist einfach. Ein Topf geht unter das Lichtschranke das das Objekt erkennt and das Laufband an macht. Die Beschaffenheit wird von dem induktiven Sensor erkannt. Wenn der Topf schwarz ist dann die Schranke geht runter sonst es bleib hoch. Die Lampe blinkt, wenn der Prozess an ist und bleibt an, wenn der Prozess in Betrieb ist. Durch drücken des Tasters bewegt sich das Band nach links.

Folie 9:

Um die Signale abzugreifen benutzen wir die Rapsberry Pi 3. Es ist ein kleines Computer mit dem wir die Daten abgreifen und auf ein externe HDD speichern.

Folie 10:

Wir haben das alte Kabel mit ein Y-Splitter ausgetauscht. Die Signale aus dem Prozess und dem Buskoppler werden an die Raspeberry Pi gesendet.

Folie 11:

Die Raspberry lese nur digitalen Signale darum brauchen wir ein AD Converter. Wir haben der MCP3008 ausgewählt. Es hat 8 analogen Channels. Die elektrische Spannung ist zu hoch für die Raspberry. Man benutze Kirchhoffsche Regeln um die Spannung zu reduziert. Der OP isoliert die Montage von der Y-Splitter.

Folie 12:

Das System benutzt 8 verschiedenen Signale zu funktionieren. Das Signal der Schranke zeigt, wenn die Schranke runter geht. Und der induktive Sensor zeigt, wenn ein silberne Topf vor ihr befindet.

Folie 13:

Es gibt ein Signal, wenn das Laufband an ist, eins für den Taster, eines wann das Laufband nach links geht. Wir haben jeweils ein Signal für die Licht, den Schalter und die Lichtschranke.

Folie 14:

In diesem Bild können wir sehen, welche Signale mit welchem ​​Sensor oder Aktuator im Prozess entsprechen.

Folie 15:

Ich habe eine Literaturrecherche gemacht. Hierbei habe ich folgende Methoden ausgewählt….

Ich möchte alle Methoden implementieren und vergleichen, um die beste Methode zu behalten.

Wir benutzen Unsupervised Lernen, weil es einfach zu implementieren ist und es für allen Angriffe verwenden werden kann.

Wir haben die folgenden Methoden ausgewählt. Wir wollen alle zusammen trainieren und vergleichen um die besten Methoden zu behalten. Wir benutzen Unsupervised Lernen, weil es einfach zu implementieren ist und es für allen Angriffe verwenden kann.

Folie 16:

Die erste Lernmodelle ist 1D-CNN mit 6 Layers. In Input gibt es 8 Arrays, eins für jede Sensor, von 5 bis 66 Punkten. Das Modell lernt die nächste 20 Punkten. Die verbunden Layers sind um Overfitting zu vermeiden.

Folie 17:

Die Autoencoder hat 9 Hidden Layers. Es lernt die Daten Punkten in Input. Allen Layers sind Fully Connected Layers mit Linear Fonction.

Folie 18:

SVM erstellt eine Hyperebene, in der theoretisch die Daten, die dem Normaleverhalten entsprechen, und außerhalb der Hyperebene die Anomalien sind.

Bei der Eingabe der SVM-Methode haben wir die Referenzdaten, die das normale Verhalten des Prozesses mit den Testdaten oder den Angriffsdaten enthalten. In der Ausgabe gibt die Methode ein Array mit 1s und -1s aus, wobei 1 bedeutet, dass es keine Anomalie gibt und -1 die Anomalieerkennung ist.

Folie 19:

Die LOF-Methode (Local Outlier Factor) ist eine Klassifizierungsmethode, die auf der lokalen Dichte eines Objekts basiert und die Anzahl der Nachbarn dieses Punkts in einem Radius um diesen vordefinierten Punkt darstellt. Je höher die Dichte eines Objekts ist, desto mehr Nachbarn gibt es. Dies bedeutet also, dass dieses Objekt zu einer bestimmten Klasse gehört. Im Fall der Erkennungsanomalie befinden sich idealerweise alle Objekte, die das normale Verhalten des Systems darstellen, in einer einzigen Klasse mit hoher Dichte, und alle anderen Punkte mit niedrigerer Dichte sind die Ausreißer, die gesuchten Anomalien.

Folie 20:

Die dichtebasierte räumliche Clusterbildung von Anwendungen mit Rauschen (DBSCAN) ähnelt der LOF-Methode, mit der Ausnahme, dass der Benutzer mit DBSCAN nicht nur die Anzahl der Punkte um eine Probe auswählen kann, die als Cluster betrachtet werden sollen, sondern auch den Abstand zwischen den Proben.

Folie 21:

Jeden Lernmodellen lernen das Normalverhalten des Prozesses. Es gibt 4 Normalverhalten, eins für den silbernen Topf und eins für den schwarzen Topf, eins für den Taster und eins für den Notaus Taster. Jeden Neuronen Netzwerk Lernmodellen lernen über 10 Epochen.

Folie 22:

Wenn der Taster gedrückt wird, bewegt sich das Laufband nach links. Es erscheinen also drei Signale, eines zeigt an, dass das Laufband gedrückt ist, ein zweites zeigt an, dass sich das Laufband bewegt, und ein drittes zeigt an, dass sich das Laufband nach links bewegt.

Folie 23:

Différents scalers ont été utilisés dans le but d’optimiser les résultats de chaque méthode. No Scalers bedeutet, dass die Daten nicht vorbereitet werden und daher direkt den Methoden zugewiesen werden. MinMaxScaler: Skaliert den Datensatz neu, sodass alle Feature-Werte im Bereich [0, 1] liegen. StandardScaler: Entfernt den Mittelwert und skaliert die Daten auf Einheitsvarianz. BooleanScaler: Alle Werte unter 1 werden auf 0 gesetzt, Werte über 1 werden auf 1 gesetzt.

Folie 24:

MaxAbsScaler: Skaliert so, dass die Trainingsdaten im Bereich [-1,1] liegen, indem jeder Datenpunkt durch den Maximalwert geteilt wird. 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 werde die Yeo-Johnson-Transformationen unterstützt und der optimale Skalierungsfaktor wird bei der Methode über die Maximum-Likelihood-Schätzung bestimmt. Normalizer: Skaliert den Vektor für jede Probe neu, um unabhängig von der Verteilung der Proben eine Einheitsnorm zu erhalten.

Folie 25:

RobustScaler: Die Zentrierungs- und Skalierungsstatistik vom RobustScaler basiert auf Perzentilen und wird daher nicht von einigen wenigen sehr großen Randausreißern beeinflusst. QuantileTransformer: Wendet eine nichtlineare Transformation an, sodass die Wahrscheinlichkeitsdichtefunktion jedes Merkmals auf eine gleichmäßige oder Gaußsche Verteilung abgebildet wird.

Folie 26:

Es gibt verschiedene Arten von Angriffen. Einige sollen das ursprüngliche Verhalten des Prozesses ändern wie PLCinject und Profinet-Replay Angriff. Andere Angriffe ändern nichts daran. Unseren Lernmodellen sollen eine Anomalie detektieren wann es ein PLCinject oder Profinet-Replay Angriff gibt.

Folie 27:

Um die Lernmodellen zu evaluieren, gibt es vier Angriffen, drei Angriffen die das Verhalten verändert und ein Angriff die das Verhalten nicht verändert. Leet

Folie 28:

Der letzte Angriff, der das Verhalten des Systems ändert, ist an sich kein wirklicher Angriff, da er nicht durch Injizieren eines Computerwurms oder durch Ändern eines Teils des Systems ausgeführt wurde. Es wurde ein Fake Angriff genannt. Ich habe einen Lauf mit einem silbernen Topf und dann einen mit dem schwarzen Topf aufgenommen. Ich habe dann die Schranke-Signale zwischen den beiden Läufen ausgetauschtwurde.

Folie 29:

Dieser Angriff ist der Sicherheitsangriff. Damit soll verhindert werden, dass die Stromversorgung des Systems unterbrochen wird, wenn der Not-Aus-Taster gedrückt wird. Daher stoppt der Prozess nicht mehr, wenn die Not-Aus-Taste gedrückt wird, und läuft weiter, wenn ein Topf in den Prozess eingefügt wurde.

Folie 30:

CNN wurde mit verschiedenen Fenstergrößen (5, 10, 20, 33, 66) und für verschiedene Skalierer, No Scaler, MinMaxScaler, BooleanScaler und StandardScaler, getestet. Verschiedene Schwellenwerte wurden verwendet, um den Fortschritt der Anomalieerkennung zu beobachten.

Folie 31:

Wir beobachten in allen Fällen, dass die Genauigkeit der Anomalieerkennung mit zunehmendem Schwellenwert abnimmt und umgekehrt die Ergebnisse für die Erkennung normaler Verhaltensweisen zunehmen. Dies ist normal, da wir bei einem höheren Schwellenwert die Unterschiede zwischen den Output Werten des Netzwerks und den realen Werten mehr akzeptieren.

Folie 32:

Ebenso zur Erkennung von Safety Angriff, die mit abnehmendem Schwellenwert zunehmen.

Folie 33:

Wie CNN wird der Autoencoder mit verschiedenen Schwellenwerten getestet. Der Autoencoder wird auch mit verschiedenen Klassifikatoren, Punkt Classifier (wie CNN) und Fenster Classifier getestet. Ähnliche Ergebnisse werden bei CNN beobachtet. Die beiden Classifiers haben ähnliche Ergebnisse. Folie 34:

Ähnliche Ergebnisse werden bei CNN beobachtet.

Folie 35:

SVM wird mit verschiedenen Scalers und verschiedenen Werten für Nu getestet. « Nu » darstellt eine Obergrenze für den Anteil der Trainingsfehler und eine Untergrenze für den Anteil der Unterstützungsvektoren.

Folie 36:

Bei den meisten Scalers nimmt die Genauigkeit der Erkennung des normalen Verhaltens mit zunehmendem "Nu" ab.

Folie 37:

Folie 38:

Die Genauigkeit für Safety angriffe erhöht sich mit „Nu“.

Folie 39:

LOF wird mit verschiedenen Scalers und verschiedenen Nachbarn Werten getestet.

Folie 40:

Folie 41:

Folie 42:

Folie 43:

Für DBSCAN experimentiere 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 mindeste 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.

Wir beobachten das beste Ergebnis mit MinMaxScalers.

Folie 44:

Wir beobachten das beste Ergebnis mit MinMaxScalers für eps = 0,6003 und min\_samp = 15000.

Folie 45:

Diese Tabelle zeig das beste Ergebnis für jede Methode zusammen. Wir stellen fest, dass DBSCAN uns den besten Wert für die Erkennung von Anomalien bietet.

Folie 46:

In dieser Masterarbeit konnte ich die Signale eines Prozesses erfassen, mit dem Objekte nach ihrer Beschaffenheit sortiert wurden. Mit diesen Signalen habe ich sie in zwei Datensätze unterteilt, einen Trainingsdatensatz und einen Testdatensatz. Die Datensätze wurden mit Scalern erstellt, um die Abhängigkeit von Lernmodellen von Scalern zu beobachten. 5 Methoden wurden verwendet: CNN, Autoencoder, SVM, LOF, DBSCAN. DBSCAN liefert uns die besten Ergebnisse.

Diese Ergebnisse könnten verbessert werden, indem die Anomalie auf jedem einzelnen Sensor erkannt wird oder indem bessere Methoden zur Schwellenwertauswahl angewendet werden. Zur Vorbereitung der Daten können andere Methoden als Scalern verwendet werden, z. B. PCA oder maschinelles Lernen.