**Automatische Erkennung der implizit angewendeten Regeln und Namenskonventionen im IT-Infrastruktur-Management (Microsoft Active Directory) und deren Anwendung**

**Problembeschreibung:**

Unternehmens-IT-Landschaften bestehen aus vielen individuellen, komplexen, heterogenen Systemen. Die Basis dieser Landschaft bildet mit ca. 90% Marktanteil meist das sog. Active Directory (AD) von Microsoft. Das AD ermöglicht es eine Unternehmens-IT-Infrastruktur hinsichtlich von organisatorischen, geografischen und berechtigungsspezifischen Strukturen zu gliedern. Mit der zentralen und dezentralen Verwaltung von verschiedenen Objekttypen, wie z.B. Benutzer, Gruppen, Computer, Dateifreigaben, usw. und deren Eigenschaften in der hierarchischen Verzeichnisstruktur haben die Administratoren jederzeit die nötige Übersicht bzw. die detaillierten Informationen über das Netzwerk stets verfügbar. Viele andere Applikationen (z.B. Mailsysteme, Adressverzeichnisse, Datenbanken, Betriebsführungsapplikationen…) nutzen mittlerweile das Active Directory und dessen Replikationsmechanismen teilweise auch als zuverlässiges und lokations-unabhängig verfügbares Speichermedium. Eine der wichtigsten Funktionsbestandteile sind jedoch die verschiedenen Authentifizierungs-, Authorisierungsmöglichkeiten sowie die Erteilung von Zugriffsberechtigungen auf verschiedenste Arten von Ressourcen und Applikationen. Von der Integrität des Active-Directory und der Korrektheit der Daten im AD hängt daher ein Großteil des IT-Betriebs unmittelbar ab.

Insbesondere global agierende Konzerne delegieren und/oder outsourcen die Verwaltung und den Betrieb des Active Directory an verschiedene IT Service Provider und haben teilweise unterschiedliche Regeln, bzw. Namenskonventionen zur Pflege der Inhalte von Active Directory Attributen. Es handelt sich dabei um Bildungsvorschriften für aus verschiedenen anderen Attributen zusammengesetzten oder abhängigen Merkmalen, wie z.B.

* Emailadresse(n)
  + Z.B**. <vorname>.<nachname>@maildomain** oder **<vorname>.<nachname>.ext@maildomain** für externe accounts oder  
    **<nachname>.<vorname>@maildomain** (falls **.** <vorname>.<nachname>@maildomain bereits vergeben)
* Name
  + Z.B. **<Vorname> <Nachname>** oder  
    **<Nachname> <Vorname>** oder  
    **<Anmeldename>**
* Anzeigename
  + Z.B. **<Nachname> <Vorname>, <Abteilungsname>** oder   
    **<Vorname> <Nachname> <Abteilungsname>**
* Abteilungsname
  + Z.B. **<Bereichsname> <erste 2 Teile der Abteilungsbezeichnung>** oder  
    **<erste 3 Teile der Abteilungsbezeichnung>**
* Anmeldename
  + Z.B. **<PersonalNummer>** oder  
    **<erster und letzter Buchstabe des Vornamens><erste 5 Zeichen des Nachnamens>**

Die Regeln können dabei z.B. für folgende Bereiche gelten:

* Global
  + Technisch, z.B. Emailadresse, Anmeldename oder userPrincipalName
  + Organisatorisch, z.B. Displayname für die unternehmensweite Adressliste zur Vereinfachung der Suche
* Standortabhängig
  + Landesspezifische Postadresse (physicalDeliveryOfficeName)
* Bereichsabhängig
  + Abteilungsbezeichnung
* Abteilungsabhängig
  + Abteilungsbezeichnung
* Providerabhängig
  + Name im AD, Accounttyp

**Zielsetzung:**

Es liegt im Interesse des Unternehmens, dass trotz der heterogenen Verwaltungsumgebungen das Ergebnis (die Inhalte des AD) homogen erscheint, da manche Attribute sogar extern „sichtbar“ (Stichwort „CI“) sein können. Deswegen müssen der internen IT Möglichkeiten zur Überwachung der Umsetzung der Regeln durch die verschiedenen Dienstleister zur Verfügung gestellt werden.

Die Dr. Hagen & Partner GmbH hat bereits eine Applikation „ADListener“ implementiert, mit der zeitnah alle Änderungen im AD protokolliert und dokumentiert werden können. Der zu realisierende Algorithmus soll als Erweiterungsmodul in diese Software integriert werden. Er wird die zentrale Komponente zum Auffinden und Erkennen unzulässiger Änderungen darstellen. Gleichzeitig soll er auch als Hilfsmittel zum Rollout des ADListeners zur Erfassung und Digitalisierung bereits existierender Regeln dienen.

**Lösungsansätze:**

In einem ersten Schritt sollte deswegen versucht werden, die bestehenden AD Inhalte zu analysieren und alle aktuell angewandten Regeln herauszufinden.

Hier könnte Clustering mittels unüberwachtem Lernen (ML) angewendet werden. Die Anzahl der AD-Objekte kann in Konzernen sehr groß werden (mehrere 100.000 Useraccounts, mehr als 1 Million Gruppen -> Milliarden Kombinationsmöglichkeiten Accounts-Gruppen). Jedes AD-Objekt hat dazu üblicherweise zwischen 50 und 100 belegte Attribute. Dies erschwert es, ohne ML eine Lösung zu finden.

Nach der Ist-Analyse erfolgt dann eine „manuelle“ Selektion der gültigen Bildungsvorschriften für die aktuelle Infrastruktur.

Abhängigkeiten zwischen Attributen, wovon diese Vorschriften abhängig sein können, können sehr vielfältig sein. Hier sind zu betrachten:

o   direkte Abhängigkeiten: Inhalte sind eine Kombination aus verschieden anderen Attributen  
o   Indirekte Abhängigkeiten: z.B. Wenn Attribut AB == „XY“ dann Attribut CD := „KLM“, sonst Attribut CD := „NOP“  
o   Mischform aus indirekten und direkten Abhängigkeiten.

Für diesen Schritt wäre die Anwendung der Algorithmen sowohl von unüberwachtem Lernen (ML), also Clustering, als erster Schritt als auch von überwachtem Lernen (ML), also Classification, sinnvoll.

Schließlich sollte der Algorithmus in der Lage sein, die Regelkonformität jedes einzelnen AD Objektes (z.B. zwischen 1 und 10) zu bewerten.

Zu diesem Schritt könnte ein Programm mit statistischen Berechnungen eine gute Lösung sein.

**Aufgaben in der Bachelorarbeit und einzusetzende Technologien:**  
Mittels des KI-Tools Tensorflow und Scikit-Learn von Python sollen zunächst die für die Aufgabenstellung geeignete ML-Ansätze gefunden und hinsichtlich eines Lösungsansatzes für das konkrete Problem evaluiert und bewertet werden. Dazu werden entsprechende simulierte IT-Infrastrukturumgebungen als Lerndaten verfügbar gemacht. Ergebnis soll dann die Entscheidung für einen, max. zwei konkrete passende Lösungsansätze sein.

Anschließend soll in einem 2.ten Schritt in einem oder in beiden gefundenen Modellen der Algorithmus sukzessive optimiert und verfeinert werden, um schließlich dann an möglichst Realitätsnahen IT-Infrastrukturumgebungen den Einsatz und das Potential einer solchen Vorgehensweise zu evaluieren und zu bewerten.

# Einleitung:

# Fragestellung:

In dieser Wissenschaftlicher Arbeit handelt es sich um

**Ziel**: Es wird gestrebt, dass man die Anomalien regelt.

Data Analysis:

**Technologien:**

Jupyter Notebook

Jupyter enhält viele wichtige Bibliotheken von Python. Dies erleichtert die Arbeit, ohne die nötwendigkeit sie zu installieren.

# Daten Vorbereitung:

Das erste Ziel ist die Emails zu analysieren. Als erster Schritt

ML- Algorithms sind keine Magie, kein Blackbox, man muss verstehen wie der Algorithmus funktioniert

Mit welcher Datenmenge funkioniert er besser.liefert gute Ergebnisse aus ausliefern sollten.

Während supervised learning auf Features und Labels passiert /Input&Output, basiert unsupervised learning lediglich auf Features, deswegen sind Features-Auswahl sehr erheblich. Dies erfordert einen guten Blick über die Daten zu haben, damit man sich für die wesentliche / relevante Features entscheiden kann.

Noisy Daten. Zielen ohne ein wichtiges Features sollen betrachtet werden und dafür eine Entscheidung genommen werden .. löschen ? mit mean Value ausfüllen, ignorieren, mit rechnen weil das auch signifikant sein könnte.

Dataanalysis / datamining / Pandas und Matplotlab sind auch sehr wichtig ... weil ... Pandas ..., Matlab ...,

PCA algorithm? Hilfreich oder nicht ? warum? Ergebnisse?

Nach betrachtung der Daten

Features oder Predictors, input, Attributes, each column ist a Feature

Vector # Matrix

response oder target, label, output

Observations : samples, instances, records,

Building a Model:

In order to build a model, the Features must be numeric .

Must have the same Features in the same order.

# Angewendete Bibliotheken:

Python Bibliotheken:

Pandas, re, sklearn, Matplotlib,

Pandas ermöglicht uns die Daten effektiv zu studieren, ohne viel Zeit die Daten arbeiten zu müssen und gibt uns. Diese Bibliothek Von CSV zu Dataframe zwei dimentionale Datastruktur mit Labels.

erleichtert Data analyse, dadurch dass man die Daten in Zeilen und Spalten teilt.

Re:

Re ist die beste Bibliothek für Strings bearbeiten patterns suchen und ersetzen.

Sklearn:

Mit Sklearn können Machine-Learning-Algorithmen leicht angewendet werden.

# Angewendete Algorithmen und deren Resultate:

# Probleme/ die Anforderungen:

Es gibt mehr Texte als numerical Data.

Wir haben emails .. strings .. sollten die Daten als numerical gemacht werden oder nicht/ warum.

Frameworks ; Scikit-Learn Tensorflow, Keras, welches wird verwendet, warum?!

Quellen:

<https://docs.python.org/3/library/re.html>

<https://pandas.pydata.org/>

<https://scikit-learn.org/stable/index.html>

<https://docs.python.org/3/library/re.html>