Mechanismen zur automatischen Konfiguration von Netzwerkkomponenten und Services

Andreas Maier
Seminar Innovative Internet-Technologien und Mobilkommunikation, WS 2008/2009
Institut für Informatik, Lehrstuhl Netzarchitekturen und Netzdienste
Technische Universität München
maierand@in.tum.de

ABSTRACT (Kurzfassung)
In diesem Ausarbeitung werden die theoretischen Grundlagen zur automatischen Konfiguration von Netzwerkkomponenten und Services beschrieben: allgemeine mit maschinellem Lernen lösbare Problemfelder und die durch Knowledge Plane zu lösenden Aufgaben.

Keywords (Schlüsselworte)
Knowledge Plane, Maschinelles Lernen, Klassifikation und Regression, Acting & Planning, Interpretieren und Verstehen, Rapid Configuration of Network, Anomaly detection

1. INTRODUCTION
In diesem Vortrag möchte ich die neuen Methoden und Mechanismen zur automatischen Konfiguration von Netzwerkkomponenten und Services betrachten, ein neues vor kurzem entwickeltes Paradigma der Netzwerkverwaltung - Knowledge Plane.

2. Lernen in Computersystemen
Knowledge Plane – ist ein verteiltes und dezentralisiertes Konstrukt für die Sammlung und Verarbeitung von Informationen in einem Netwerk. Die Informationen werden von unterschiedlichsten Schichten gesammelt von Anwendungsschicht bis zum Physical layer. Eigenes Verhalten wird erforscht und aufgetretene Probleme analysiert mit dem Ziel, beispielsweise die Betriebseigenschaften so zu steuern, dass größere Ausfallsicherheit und bessere Arbeitsleistung oder erhöhte Sicherheit erreicht wird. Dazu benötigt die Knowledge Plane das Konzept maschinellen Lernens.

Maschinelles Lernen ist ein Oberbegriff für die „künstliche“ Generierung von Wissen aus Erfahrung: Ein künstliches System lernt aus Beispielen und kann nach Beendigung der Lernphase verallgemeinern. Das heißt, es lernt nicht einfach die Beispiele auswendig, sondern es „erkennt“ Gesetzmäßigkeiten in den Lerndaten. So kann das System auch unbekannte Daten beurteilen.

Anderseits kann auch das Lernen in Computersystemen als Performance Task betrachtet werden. Das Bild 1. veranschaulicht dieses Modell. Das Rechenssystem versucht mit Hilfe von Lernregel die Umgebung ins Wissensmodell umzuwandeln. Die entstandenen Kenntnisse werden für die Verbesserung von Performance verwendet. Optional kann man die vorhandenen Kenntnisse dazu benutzen um den Lernprozess zu beeinflussen oder zu verbessern. Das Performance wiederum beeinflusst die Umgebung. Dabei unterscheidet man zwischen Online und Offline Lernen. Beim Offline Lernen wird das vorhandene Wissen nur einmal ins Performance transformiert dagegen wiederholt sich bei Online Lernen der Vorgang als endlose Schleife.

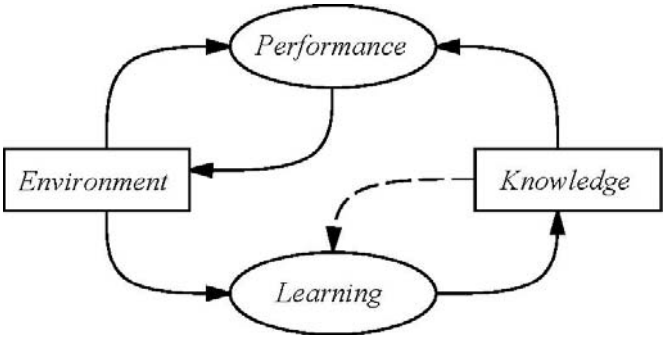

Bild 1. Lernen in Computersystemen als Performance Task.[2]

3. Mit maschinellem Lernen lösbare Problemfelder
In dem maschinellen Lernen werden die Lernprobleme in drei Hauptkategorien unterteilt: Klassifikation und Regression auf Messdaten, Acting & Planning, Interpretieren und Verstehen.

Formulation	Performance Task
Classification & Regression	predict y given x predict rest of x given part of x predict $P(x)$ given x
Acting & Planning	iteratively choose action a in state s choose actions $\langle a_1, \ldots, a_n \rangle$ to achieve goal g find setting s to optimize objective $J(s)$
Interpretation & Understanding	parse data stream into tree structure of objects or events

Bild 2. Übersicht der mit maschinellem Lernen lösbaren Problemfeldern.[2]

3.1 Klassifikation und Regression auf Messdaten
Bei Klassifikation wird zu jeder Beobachtung ein Label (Klasse) gegeben. Z.B. „Server antwortet nicht“, „Server ist überlastet“, „Keine Route zum Server“ bei der Untersuchung der Verbindungsabbrüche.

Bei Regression versucht man den unbekannten Wert anhand der vorhandenen Beobachtungen vorherzusagen, z.B. die Zeit, die man braucht um Verbindung zum Server aufzubauen.

Bei manchen Situationen ist es unmöglich alle benötigte Messungen durchzuführen, z.B. wegen allgemeiner Aufwendigkeit oder unvollständiger bzw. verrauschter Muster. In dem Fall sollen die Daten vervollständigt werden. Hier spricht man von Mustervervollständigung(pattern completion, flexible prediction).

Wir betrachten noch die Methoden, die dafür benutzt werden um maschinell zu lernen. Dabei unterscheidet man zwischen betreutem Lernen ("supervised learning"), unbetreutem Lernen ("unsupervised learning") und halb-betreutem Lernen (semi-supervised learning).

Beim betreuten Lernen (supervised learning) gibt ein Mensch gleich ein „Lehrer“ die Werte der Zielfunktion für alle Trainingsbeispiele an. Es gibt Mehrzahl an Paradigmen für betreutes Lernen wie decision tree and rule induction[1] [2], neural network methods[3], nearest neighbor approaches[4], und probabilistic methods[5]. Die Paradigmen unterscheiden sich in Algorithmen der Wissensbeschaffung und Repräsentation des vorhandenen Wissens.

Die zweite Klasse der Wissensbeschaffung ist unbetreutes Lernen (unsupervised learning) - es gibt keine vorkategorisierten Beispiele. Hier gibt es auch mehrere Paradigmen, die man grob in zwei Kategorien unterteilen kann Clustering[6] [7] und density estimation[8]. In Falle vom Clustering ermittelt man Gruppen (Clustern) von Objekten, deren Eigenschaften oder Eigenschaftsausprägungen bestimmte Ähnlichkeiten (bzw. Unähnlichkeiten) aufweisen. Beim density estimation versucht man eine Dichtefunktion zu erstellen, die die vorhandenen Trainingswerte abdecken wird und dann auf neue Werte eingesetzt werden kann.

Die dritte Klasse der Wissensbeschaffung ist halb-betreutes Lernen (semi-supervised learning). Das ist wie der Name schon sagt eine Mischung von zwei schon beschriebenen Klassen. D.h. nur ein Teil der Trainingsbeispiele wird von Menschen klassifiziert. Der Rest sollte vom Rechensystem selbst bearbeitet werden.

3.2 Acting & Planning

Das nächste Aufgabefeld, das wir betrachten ist acting & planning, also das Anwenden von Wissen für Aktions- oder Planauswahl. In einfachster Form wird eine Aktion direkt ausgewählt ohne vorherige Aktionen in Betracht zu ziehen. In diesem Fall kann die Klassifikation direkt mit Aktionsauswahl verbunden werden, d.h. jeder ermittelten Klasse wird eine Aktion zugeordnet. Genauso kann die Aktionsauswahl mit der Regression verknüpft werden um die Werte vorherzusagen oder die Brauchbarkeit von Aktionen zu ermitteln. Aktionen können einzeln ausgewählt werden oder in Form von macro-operators zusammengesetzt um als Pakete ausgeführt zu werden.

Genauso wie mit Klassifikation und Regression gibt es auch bei Acting und Planning die speziellen Methoden, die dafür benutzt werden um maschinell zu lernen: Learning apprentice oder adaptive interface, programming by demonstration, behavioral cloning, unterstütztes Lernen (reinforcement learning), learning from problem solving and mental search, empirical optimization, z.B surface methodology.

[1] Quinlan, 1993

[2] Clark & Niblett, 1988

[3] Rumelhart, 1986

[4] Aha, 1991

[5] Buntine, 1996

[6] Fisher, 1987

[7] Cheeseman, 1988

[8] Priebe & Marchette, 1993

Learning apprentice[9] oder adaptive interface[10], d.h. das System lernt durch Beobachtung der Aktionen von Benutzern. Das System gibt dem Benutzer Empfehlungen, die er akzeptieren oder auch andere Entscheidungen treffen kann. Dadurch wandeln wir das Problem in das Problem des betreuten Lernens (supervised learning) und können auch die Methoden, die dafür entwickelt wurden, anwenden; sprich decision tree, rule induction,, neural network methods, nearest neighbor approaches, und probabilistic methods.

Das verwandte Paradigma ist programming by demonstration[11], wo dagegen versucht wird ein Set von macro-operatoren zu erstellen.

Ein weiteres verwandtes Modell behavioral cloning[12]. Hier lernt das System durch Beobachtung der Aktionen von Benutzern ohne Benutzer direkt zu befragen.

Außerdem existiert noch unterstütztes Lernen (reinforcement learning): Der Agent bekommt zwar Feedback von der Umwelt zu seiner Kategorisierung, erfährt aber nicht explizit, was die richtige Kategorisierung gewesen wäre, da es eventuell mehrere Schritte notwendig sind um den gewünschten Zustand zu erreichen.

Zum Beispiel in Falle vom dynamischen Netzwerkrouting: das System wird versuchen mehrere Routen aufzubauen, jeder Route schließt mehrere Entscheidungsschritte ein bis sie komplett aufgebaut ist. Jedoch werden die Entscheidungskriterien wie Routenmetrics erst am Ende von Routenaufbau bereit stehen.

Die eng mit unterstütztem Lernen verbundene Methode ist learning from problem solving and mental search[13]. In diesem Fall werden die Änderungen zuerst auf einem Modell erprobt.

Unser Routingbeispiel in diesem Fall würde folgendermaßen aussehen: das Lernsystem würde das gewünschte Verhalten von Netzwerk modellieren, bevor es dieses Verhalten praktisch umsetzen würde.

Und die letzte Methode der Wissensbeschaffung ist empirical optimization. In diesem Fall existiert kein Modell vom Testsystem. Es gibt Anzahl von Parametern, deren Auswirkung auf das Gesamtsystem unbekannt ist. Durch das Verändern von einzelnen Parametern oder Parameterbündeln wird versucht die Auswirkung auf das Gesamtsystem zu ermitteln. Ein Beispiel davon ist surface methodology[14].

3.3 Interpretieren und Verstehen

Das dritte Aufgabenfeld in maschinellem Lernen ist Learning for Interpretation and Understanding. Anstatt wie bei Klassifikation und Regression ein Vorhersagemodell zu erstellen, versucht man hier ein Modell zu erstellen, das eine Erklärung mit Hilfe von tieferen Strukturen möglich macht. Z.B. könnte man ein anomales Transferverhalten in Netwerk dadurch erklären, dass es auf einem der Server eine größere Datei veröffentlicht wurde, auf die die Nachfrage hoch ist.

Es gibt Mehrzahl von Lernaufgaben und damit verbundene Erklärungsmodelle.

[9] Mitchell, 1985

[10] Langley, 1999

[11] Cypher, 1993

[12] Sammut, 1992

[13] Sleeman, 1982

[14] Myers & Montgomery, 1995

Der erste Ansatz ist, dass jede Traininginstanz mit damit verbundenes Erklärungsmodell kommt. Das erlaubt eine effektive Erklärungsmodell zu erstellen, ist aber leider mit hohem Erstellungsaufwand verbunden.

Der zweiter Ansatz ist, dass jede Trainingsinstanz ohne damit verbundenes Erklärungsmodell kommt, besitzt aber Backgroundwissen, woraus ein Erklärungsmodell erstellt werden kann.

Und der letzte Ansatz ist, dass jede Trainingsinstanz ohne damit verbundenes Erklärungsmodell kommt, aber auch ohne Backgroundwissen, woraus ein Erklärungsmodell erstellt werden kann. Das Lernsystem erstellt das Erklärungsmodell, indem es die Regelmäßigkeiten in Daten sucht.

4. Die durch Knowledge Plane zu lösenden Aufgaben

Mit Knowledge Plane können mehrere Aufgabenarten gelöst werden. Das sind einige davon: Anomaly Detection and Fault Diagnosis, Responding to intruders and Worms, Rapid configuration of Network

4.1 Network Configuration and Optimization

4.1.1 Das Spektrum von Konfigurationsaufgaben

Tabele 1. Das Spektrum von Konfigurationsaufgaben[2]

Problem	Global parameters	Local parameters	Topology	Components
Parameter Selection	X			
Compatible Parameter Con guration	X	X		
Topological Con guration	X	X	X	
Component Selection and Con guration	X	X	X	X

Es gibt mehrere Aufgaben für die Herstellung und Konfiguration der Netzwerke. Die einfachste davon ist das parameter selection, wo die mehreren Parameter optimiert werden müssen.

Die nächste Aufgabe ist compatible parameter selection. Hier werden die Komponente des Systems nach festen Regeln miteinander verbunden. Die Effektivität des Systems wird dadurch beeinflusst, dass die einzelnen Parameter kompatibel sein müssen um mit einander kommunizieren zu können. Zum Beispiel müssen die IP-Adressen und Subnetmaske bei der Konfiguration eines Netzwerkes bestimmten Topologieregeln folgen um miteinander kommunizieren zu können. Gesamte Systemsleistung kann ziemlich komplex von lokalen Parametern abhängen.

Die dritte Aufgabe beinhält topological configuration. Das System besteht aus einzelnen Komponenten, aber die Verbindungstopologie muss noch ermittelt werden. Zum Beispiel gibt es Mehrzahl an Arbeitsstationen, Gateways, Dateiserver, Drücker, Backupgeräten. Die Aufgabe ist das Netzwerk so zu konfigurieren, damit die Gesamtleistung maximal wäre. Natürlich muss jede einzelne Topologie mit compatible parameter selection optimiert werden.

Die vierte Aufgabe ist component selection and configuration. Am Anfang besteht die Konfiguration aus einem Katalog mit möglichen Komponenten und deren Preisen. Die benötigten Komponente und ihre Anzahl müssen ermittelt werden und danach muss natürlich topological configuration gelöst werden.

4.1.2 Reconfiguration process

Bis jetzt haben wir nur das Problem der Auswahl der richtigen Konfiguration betrachtet. Dennoch existiert das Problem wie die Konfiguration effektiv implementiert werden kann. Zum Beispiel während der Installierung eines Netzwerks werden normalerweise Gateways und Routers dann Datei und Druckserver und danach erst die Arbeitsstation installiert. Die Ursache dafür ist dass man den Konfigurations- und Test- Aufwand minimieren will, der zum Beispiel zur Rekonfiguration nötig wäre. Automatic configuration tools z.B. DHCP können Arbeitsstationen konfigurieren, nachdem der Server installiert worden ist.

4.1.3 Existing AI/ML Work Configuration

4.1.3.1 Parameter Selection

Wie schon früher erwähnt wurde ist Parameter Selection reines Optimierungsproblem, wenn ein Systemmodell bekannt ist. Wenn Systemmodell nicht vorhanden ist dann können die statistischen Methoden angewendet werden (empirische Optimierung).

4.1.3.2 Compatible Parameter Con guration

Das allgemein bekannte Modell der AI ist so genanntes constraint satisfaction problem (CSP). CSP wird als Graph dargestellt. Um das Problem effektiv zu lösen wird eine Gruppe von Algorithmen entwickelt[15]. Ein anderer Einsatz ist CSP als Erfüllbarkeitsproblem der Aussagenlogik darzustellen. Eine andere Möglichkeit wäre das CSP durch die randomisierten Algorithmen zu lösen, z.B. WalkSAT[16].

Das standarte CSP hat eine fixe Graphstruktur, die aber durch zusätzliche Graphen oder Beschränkungen erweitert werden kann. Auf diesem Gebiet können Methoden constraint logic programming (CLP)[17] und dafür entwickelte Programmiersprachen angewendet werden.

4.1.3.3 Topological Con guration

Es gibt zwei Hauptansätze für die topological configuration: refinement und repair.

Refinement Methoden starten mit einem einzigen „box“, das das ganze zu konfigurierende System darstellt. Dieses „box“ hat formale Spezifikation von gewünschtem Verhalten. Die Refinement-Regeln analysieren diese Spezifikation und ersetzen dieses „box“ mit zwei oder mehreren anderen „boxen“ mit entsprechenden Verbindungen. Zum Beispiel das kleine Office-Netzwerk sollte zuerst als ein „box“ dargestellt werden, das eine bestimmte Anzahl an Arbeitsstationen, Dateiserver, Drücker mit DSL- Leitung verbindet. Die Refinement- Regel soll ersetzen dieses „box“ mit dem lokalen Netzwerk und Router/NAT box. Die andere Refinement-Regel soll dieses Netzwerk als ein drahtloses Access-point und eine Anzahl an Netzwerkarten definieren. (Alternativ als Ethernet-switch eine Anzahl von Ethernetkarten und Verbindungskabeln). Es existieren die

[15] Kumar, 1992

[16] Selman, 1993

[17] Jaffar & Maher, 1994

Forschungen, die maschinelles Lernen auf dem Gebiet anwenden[18].

Der auf dem repair basierende Ansatz geht von einer Startkonfiguration aus, die die gewünschte Spezifikation nicht erfüllt und dann wird es versucht diese Konfiguration zu reparieren, bis sie gewünschte Bedingungen erfüllt. Zum Beispiel soll die Startkonfiguration alle Drücker, Computer und andere Geräte mit einem einzigen Switch verbinden, was viel zu teuer und gross sein könnte. Die repair-Regel ersetzt diesen einzigen Switch mit mehreren kleineren und billigeren Switchs. Es existieren mehrere auf repair basierende Algorithmen[19] [20] [21].

4.1.3.4 Component Selection and Con guration

Die bereits beschriebenen auf dem refinement und repair basierenden Methoden können weiter erweitert werden um component selection und configuration zu behandeln.

4.1.3.5 Changing Operating Conditions

Bis jetzt haben wir das Problem der Konfigurationsoptimierung unter den konstanten operativen Bedingungen betrachtet. Dennoch kann es passieren, das die optimale Konfiguration an die wechselnden Bedingungen anpassen sollte, wie es bei großen Netzwerkumgebungen üblich ist. Bis jetzt existieren leider keine Erforschungen auf dem Gebiet.

4.2 Anomaly Detection

Bei der Anomaly Detection wird ermittelt, ob etwas Untypisches oder Unerwünschtes im Netzwerkverhalten vorhanden ist.

Dabei gibt es zwei generelle Einsätze für Anomaly Detection: One-Class Learning und Density Estimation.

One-Class Learning – Classifier erstellt eine kompakte Beschreibung, die den gewünschten prozentuellen Anteil (z.B. 95%) von „normal“ Netwerktraffic deckt, der Rest wird als anomal betrachtet.

Dichteschätzung (Density Estimation) – das beobachtete System wird als die Sammlung von Werten modelliert. Der Zustand mit geringer Wahrscheinlichkeit wird als anomal betrachtet.

Bei der Anomaly Detection sollen das „level of analysis“ und Kontrollvariablen ausgewählt werden, außerdem sollen die Daten von Sensoren interpretiert und aufsummiert werden. Da die Anomalie auf einem Ebene nicht erkennbar sein könnte, müssen eventuell die Daten von mehreren Ebenen noch aufsummiert werden. Z.B. Worm kann auf einem Computer nicht erkannt werden, dagegen wenn wir Daten von mehreren Computer betrachten, kann untypisches Traffic erkannt werden.

Eine weitere Aufgabe ist die falsche und wiederholte Alarme auszufiltern, dafür muss eventuell Methoden des überwachten Lernens angewendet werden. Z.B. ein Teil der Anomalien kann unwichtig oder ungefährlich sein.

Fehlereingrenzung erfordert globales Wissen innerhalb der Knowledge Plane. Z.B. ein Netzwerkroute ist überlastet: die Netzwerküberlastung kann lokal z.B. an jedem Router festgestellt werden, dagegen kann die Ermittlung von der Höhe und den Grenzen von der Überlastung für die Gesamtroute erst global erfolgen.

Bei der nächsten Aktivität dem Diagnosis werden die Quellen des Problems vermutet. Also werden die Quellen von dem unerwünschten Verhalten gesucht. Dabei können sowohl die bekannten Probleme erkannt werden, die früher von Operator aufgelistet wurden, als auch die neuen Probleme charakterisiert werden, bei denen nur einige Teile bekannt sind. Normalerweise folgt „Diagnosis“ der Fehlereingrenzung, aber man kann auch die Fehler in System vermuten ohne genau Fehlerquelle zu kennen. Genau wie bei „Anomaly detection“ können auch hier die Methoden des überwachten Lernens angewendet werden.

Sowohl Fehlereingrenzung als auch Diagnosis erfordern aktive Messungen. Dabei muss das Balance zwischen Messkosten und Informationswert erhalten werden.

Fault Isolation und Diagnosis erfordern auch das Systemmodell, das diagnostiziert wird. Um so ein Modell zu erstellen können die Methoden aus dem maschinellen Lernen wie Learning for Interpretation and Understanding angewendet werden.

Nachdem das Problem diagnostiziert wurde, können die Methoden des überwachten Lernens angewendet werden, um den Netzwerkoperator zu unterstützen oder Repairstrategie zu teilen.

4.3 Abwehr von Angriffen und Worms

Die Aufgaben zum Schutz eines Netzwerks vor Angriffen und Worms kann man so aufteilen wie sie normalerweise von Netzwerkmanager durchgeführt werden: Prävention (Prevention Tasks), Erkennung (Detection Tasks), Erwiderung und Wiederherstellung (Response and Recovery Tasks).

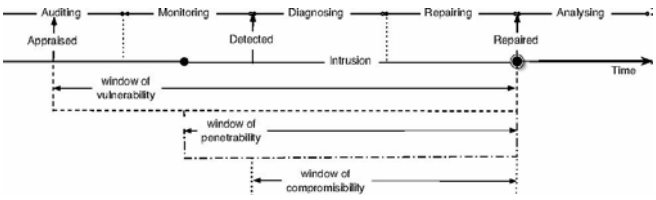

Bild 3. Abwehr von Angriffen und Worms[2]

Das Ziel des Netzwerksmanagers, dass die drei zeitlichen Abschnitte (window of vulnerability, penetrability, compromisibility) zu einem zeitlichen Punkt konvergieren. Die Aufgaben für die Abwehr von Angriffen unterscheiden sich unwesentlich von dem Wiederherstellen nach einem kritischen Fehler.

4.3.1 Prävention (Prevention Tasks)

Netzwerkadministrator strebt the window of vulnerability zu minimieren (die Zeitspanne zwischen dem Zeitpunkt, als eine Schwachstelle bekannt geworden ist und dem Anwenden von einem Patch oder einer neuen Konfiguration). Die Grundstrategie für dieses Ziel ist den Gefährdungsgrad zu minimieren (z. Beispiel Ausschalten von unnötigen Diensten) und ständige Beobachtung von neu aufgedeckten Schwachstellen um sie möglichst früh in eigenem System festzustellen.

Dafür werden Scan tools wie Nessus, Satan oder Oval benutzt. Da es immer neue Sicherheitsrisiken und Softwarefehler entdeckt werden, sollte das benutzte Tool immer aktuell gehalten werden. Wenn die Schwachstelle entdeckt wurde und kein Patch dafür

[18] Mitchell, 1985

[19] Zweben, 1994

[20] Zhang and Dietterich, 1995

[21] Boyan and Moore, 2000

existieret, musste Entscheidung getroffen werden, ob der betroffene Service ausgeschaltet werden kann, dabei wird das Risiko und die Bedienungsqualität beachtet.

Letztendlich überwacht der Netzwerkadministrator das System, um die Verhaltensmuster vor und nach dem Eindringen vergleichen zu können.

4.3.2 Erkennung (Detection Tasks)

Die Aufgabe des Netzwerksmanagements ist window of penetrability (die Zeitspanne zwischen dem Zeitpunkt, in dem das Computersystem aufgebrochen und dem Moment, wenn das Computersystem vollständig repariert worden ist) möglichst klein zu halten. Die korrekte Diagnostik erlaubt dem Netzwerkmanager entsprechend zu reagieren. Dabei soll das Gleichgewicht zwischen der Qualität und Schnelligkeit erhalten werden.

Je früher das Eindringen erkannt wird, desto mehr Chance gibt es unautorisiertes Benutzen oder Missbrauch des Systems zu verhindern. Deswegen versucht der Netzwerkadministrator das System zu überwachen, indem er Protokolle, Alarmsignale beobachtet. Jeder, der einmal ein Netzwerk administriert hat, weiß, dass man mit Informationen und Fehlalarmen überschwemmt wird. Aus diesem Grund werden die Netzwerkgeräte so konfiguriert, dass die Anzahl von Falschalarmen akzeptabel gehalten wird, was folglich das Risiko von Nichterkennen von Eindringung erhöht.

4.3.3 Erwiderung und Wiederherstellung (Response and Recovery Tasks)

Als die Diagnostik das Eindringen entdeckt hat, muss der Netzwerkmanager entsprechend handeln. Die Aufgabe des Netzwerksmanagements ist window of compromisibility (die Zeitspanne zwischen dem Zeitpunkt, in dem das Eindringen entdeckt worden ist und dem Moment, wenn das Computersystem entsprechend reagiert hat) zu reduzieren, indem er die automatic intrusion response systems einsetzt. Leider sind diese Systeme heutzutage nicht in der Lage sogar bei manuell gesteuerten Schutzmaßnahmen zu unterstützen. Aus diesem Grund erstellen die Netzwerkmanagers entsprechende Prozeduren für jede einzelne Angriffsart.

Eine Antwort auf einen Angriff kann aus dem Schließen von Benutzerprozessen über Blockieren des Benutzerskontos, Blockieren der IP- Adresse bis zum Ausschalten des Netzwerks bestehen. Da bei Wiederherstellungsmaßnahmen (damage recovery bzw. repairing) die Funktionalität des Systems erhalten werden muss, sind solche Maßnahmen schwer zu automatisieren. Die Reaktion sollte den Einfluss auf das Gesamtsystem minimieren (zum Beispiel nicht alle Netzwertsports schließen wenn das Blockieren von einem einzigen IP ausreichend ist).

Nachdem der Angriff abgewehrt worden ist, sollte Angriffsursachen und –folgen mit vorhandenen Protokollen analysiert und dokumentiert werden.

5. Fazit

In diesem Vortrag haben wir die Methoden und Aufgaben des maschinellen Lernen, sowie auch theoretische und bereits existierende Algorithmen betrachtet. Danach haben wir einige, aber natürlich nicht alle, durch die Knowledge Plane zu lösenden Aufgaben diskutiert wie Rapid Configuration of Network, Anomaly detection, Responding to intruders and Worms.

Obwohl auf dem Gebiet der Automatischen Konfiguration viel erforscht wird und mehrere Algorithmen z.B. aus maschinellem Lernen angewendet werden, befindet sich das Problem noch in der Frühphase der Entwicklung. Für die Erstellung eines Autonomes Netzwerk- oder Computersystems werden noch weitere Forschungen sowohl konzeptueller als auch experimenteller Art benötigt.

6. Literatur

[1] Pat Langley, John E. Laird, Seth Rogers - Cognitive Architectures: Research Issues and Challenges

[2] Tom Dietterich, Pat Langley - Machine Learning for Cognitive Networks: Technology Assessment and Research Challenges