Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-1040 Wien, Weyringergasse 35
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

A-8010 Graz, Inffeldgasse 16a
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

EREIGNIS-KORRELATION
TECHNOLOGIEBEOBACHTUNG
VERSION 1.0 6/2/2009 10:00 PM

Peter Teufl – peter.teufl@a-sit.at

Zusammenfassung: Im A-SIT Projekt ***Ereignis-Korrelation*** wurde eine neuartige Methode entwickelt, um unterschiedliche Daten (Ereignisse) von unterschiedlichen Sensoren zu kombinieren und für weitere Analysen aufzubereiten. Dabei werden die Daten der einzelnen Sensoren und ihr gemeinsames Auftreten in einem assoziativen Netzwerk abgebildet. Durch die Anwendung von Spreading Activation werden Aktivierungsmuster erzeugt, die dann für weitere Analysen verwendet werden. Diese Arbeit beschreibt das Problem der Ereignis-Korrelation, stellt das neuartige Framework vor und evaluiert es anhand von Echtdaten.

Inhaltsverzeichnis

Abbildungs- und Tabellenverzeichnis

8

1 Einleitung

Um der steigenden Anzahl von Angriffen auf IT Systeme entgegen zu wirken, werden Intrusion Detection Systeme (IDS, zu deutsch: Eindringlings-Erkennungs-Systeme) eingesetzt, die etwaige Angriffe oder Anomalien in einer IT Umgebung erkennen sollen. Dabei kann ein IDS Analysen auf unterschiedlichen Ebenen einer IT Umgebung durchführen. Diese Ebenen können von der Analyse des reinen Netzwerkverkehrs bis zur Analyse von Applikationen oder des Benutzerverhaltens auf einzelnen Maschinen reichen. Jede dieser Ebene spielt je nach Art des Angriffs eine unterschiedliche Rolle und kann besser oder schlechter für die Erkennung verwendet werden. Ebenso ist es möglich, dass sich ein Angriff nur durch die Analyse mehrerer unterschiedlichen Ebenen erkennen lässt. Die für das Sammeln der Daten zuständigen Komponenten werden als Sensoren bezeichnet.
Die Analyse der von den Sensoren gelieferten Daten ist dabei eine der wichtigsten Komponenten eines IDS. Folgende Punkte müssen dabei beachtet werden:

- **Analyse:** Es gibt Fälle wo einzelne Sensoren ausreichen, um einen Angriff zu erkennen. Dies ist aber nicht der Normalfall, und man muss davon ausgehen, dass erst die Kombination von unterschiedlichen Sensordaten eine Erkennung zulässt.
- **False Negatives:** Im Idealfall darf die Analyseeinheit des IDS keine Angriffe übersehen. Ist dies trotzdem der Fall, so spricht man von *False Negatives*, also Daten die fälschlicherweise als normal eingestuft wurden.
- **False Positives:** Im Gegensatz zu False Negatives beschreiben *False Positives* Ereignisse, die vom IDS fälschlicherweise als Angriff erkannt werden. Auch solche Ereignisse schränken die Qualität des IDS ein, da sie manuellen Analyseaufwand verursachen.

In dieser Arbeit konzentrieren wir uns auf die Analyse von Sensordaten (oder auch Ereignisse) die von Sensoren auf unterschiedlichen Ebenen erzeugt werden. Das Zusammenfassen und die Analyse dieser Ereignisse, die von unterschiedlichen Sensoren erzeugt werden, bezeichnet man als Ereignis-Korrelation.

Das vorgestellte Framework behandelt dabei folgende Punkte:

- **Unüberwachtes Lernen:** Wir gehen von einem System mit unterschiedlichen Sensoren aus. Jeder dieser Sensoren liefert Daten (Ereignisse), die Schlüsse über das zu analysierende System zulassen. Dabei gehen wir davon aus, dass wir kein Wissen über die Zusammenhänge dieser Ereignisse haben. Das Framework lernt dabei selbstständig die Relationen zwischen einzelnen Ereignissen und fasst das Gesamtverhalten nach Ähnlichkeit zusammen. Im Falle von Netzwerk IDS Daten kann dies dann die automatische Erkennung von unterschiedlichen Clustern sein: Denial of Service (DOS) Angriffe, Probes, normaler Verkehr.
- **Analyse von unterschiedlichen Sensordaten:** Unterschiedliche Sensordaten haben unterschiedliche Wertbereiche und können entweder symbolische oder kontinuierliche Daten liefern. Diese unterschiedlichen Daten erschweren die direkte unüberwachte Analyse. Aus diesem Grund fügt das Framework noch eine Zwischenschicht ein, die die Daten transformiert, sodass diese unüberwachte Analyse wieder möglich wird.
- **Korrelation der Sensordaten:** Die Betrachtung der unterschiedlichen Ereignisse und deren Relationen untereinander (z.B.: gemeinsames Auftreten) werden vom Framework erlernt und ermöglichen Schlüsse über die Zusammenhänge der einzelnen Ereignisse.

- **Anomalieerkennung:** Dabei wird für den normalen Verkehr ein Modell trainiert. Unbekannte Muster – in diesem Fall Angriffe - die während dem Training nicht vorhanden waren, sollen dann als Anomalie erkannt werden. Mit Hilfe der Aktivierungsenergie der im Framework vorgestellten Aktivierungsmuster kann diese Anomalieerkennung realisiert werden, da diese Energie bei Anomalien schwächer ist.

2 Ereignis-Korrelation (Event Correlation)

Das Sammeln von Sensordaten, die aufgrund von Ereignissen (Events) erzeugt wurden und deren Auswertung wird als Ereignis-Korrelation bezeichnet. Wenn wir dabei das Wort Korrelation verwenden, um die Verknüpfung unterschiedlicher Ereignisse zu bezeichnen, dann stellt sich hier gleich die Frage wie das Auftreten unterschiedlicher Ereignisse analysiert und klassifiziert werden kann. Hierbei können auf jeden Fall bestimmte Basisfunktionen angegeben werden, die von einem Ergeignis-Korrelations-System (EKS) erfüllt werden müssen:

- **Filtern von Ereignisse:** Im Falle eines Angriffs werden die Sensoren eines IDS unterschiedliche Daten liefern, die von einem EKS korrekt interpretiert werden müssen. Ein EKS soll dabei die gemeinsame Ursache dieser Sensordaten erkennen und klassifizieren. Der Betreiber des EKS erhält nicht mehr die Datenströme der einzelnen Sensoren, sondern eine Zusammenfassung, die das aktuelle Verhalten auf höherer Ebene beschreibt.
- **Erkennen und Repräsentation von Verhaltensweisen:** Die Kernkomponente eines EKS basiert auf der Analyse der Sensordaten, der Interpretation dieser Analyse und der Repräsentation. Vor allem die Repräsentation spielt eine sehr wichtige Rolle, da es möglich sein muss, dem Benutzer auf möglichst verständliche Weise ein Bild des aktuellen Status zu liefern.

Es werden nun die Basiskomponenten eines EKS beschrieben:

Sensor: Ein Sensor liest beliebige Daten aus, wendet Vorverarbeitungsschritte auf diese Daten an und übergibt Ereignisse an höhere Schichten des EKS, wenn bestimmte Bedingungen erfüllt sind. Ein einfaches Beispiel für so eine Bedingung ist ein Temperatur-Sensor der Ereignisse nur an höhere Schichten weiterleitet, wenn die gemessene Temperatur größer als ein bestimmter Schwellwert ist.

Ereignis Modell: Dieses Modell beschreibt die Relationen zwischen unterschiedlichen Ereignissen. Dabei kann es sich um ein einfaches, auf Regeln basierendes Modell oder um ein komplexes mathematisches Modell handeln. Für die Erzeugung der Modelle können unterschiedliche Verfahren verwendet werden – z.B. Algorithmen aus dem maschinellen Lernen.

Korrelations-Komponente: Diese Komponente vergleicht zu überprüfende Ereignisse mit den trainierten Modellen und zieht daraus Schlüsse über das zu analysierende Verhaltensmuster.

Bei der Korrelationskomponente spielen dabei folgende Punkte eine Rolle:

Selektion der Informationen: Es kann sein, dass das gleiche Ereignis von verschiedenen Sensoren gleichzeitig gemeldet wird. In diesem Fall kann es sinnvoll sein, mehrfach vorhandene Ereignisse zu löschen. Allerdings muss darauf geachtet werden, dass in manchen Fällen auch die Information des wiederholten Auftretens eines Ereignisses für die Erkennung von Verhaltensmustern verwendet werden kann.

Filtern: Das EKS muss in der Lage sein, aus den gesamten Ereignissen jene zu extrahieren, die für die weitere Analyse des Verhaltens den größten Informationsgehalt bringen.

Ereignis Repräsentation: Abhängig vom Zustand des EKS kann es Sinn machen, Ereignisse zu unterdrücken oder in einer anderen Art zu repräsentieren.

Zählen von Ereignissen: EKS können diese sehr einfache Methode anwenden, um einen Alarm zu generieren, wenn die Anzahl bestimmter Ereignisse einen bestimmten Schwellwert überschreitet. In dieser Arbeit konzentrieren wir uns aber auf viel höher entwickelte Analyse-Werkzeuge.

Gewichtung von Ereignissen: Das EKS hat die Aufgabe, unterschiedliche Ereignisse je nach Informationsgehalt zu gewichten.

Generalisierung von Ereignissen: Das EKS soll in der Lage sein, spezifische Ereignisse einer allgemeineren Klasse zuzuordnen. Beispiele dafür sind z.B. das Erkennen eines speziellen DOS Angriffs, der zur allgemeinen Klasse der DOS Angriffe zugeordnet wird.
Zeitliche Zusammenhänge: Die zeitliche Abfolge von Ereignissen kann Schlüsse auf das darunter liegende Verhalten zulassen und soll von einem EKS berücksichtigt werden.

Clustern von Ereignissen: Das Clustern oder Zusammenfassen von ähnlichen Verhaltensmustern spielt vor allem dann eine Rolle wenn noch keine Informationen über die möglichen Muster vorliegen. Das System versucht dann selbständig ähnliche Muster zu finden und in Clustern zusammenzufassen (z.B. unterschiedliche Angriffe, normaler Verkehr).
Modelle basierend auf Künstlicher Intelligenz (KI): Solche Modelle versuchen, menschliche Problemlösungs-Strategien abzubilden und für die Analyse von Ereignissen zu verwenden. Es können dabei Expertensysteme oder eine Vielzahl anderer Methoden aus der Künstlichen Intelligenz angewendet werden.

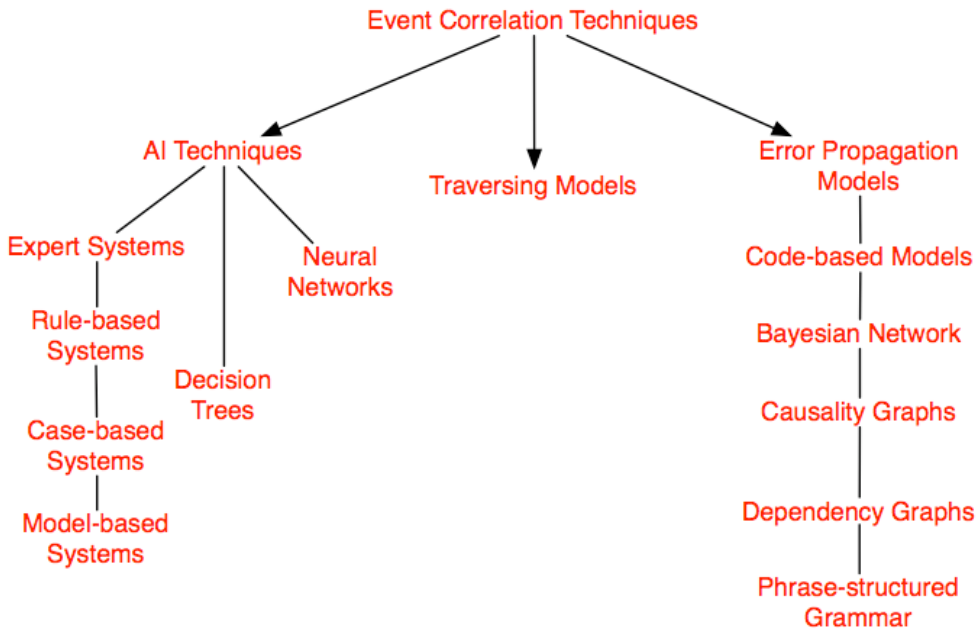

Abbildung 1 - Überblick über Ereignis-Korrelation

3 Methoden

Dieser Abschnitt gibt einen kurzen Überblick über die Algorithmen und Methoden, die vom Framework verwendet werden.

3.1 Semantische Netzwerke/Assoziative Netzwerke

Semantische Netzwerke [1] bestehen aus Knoten, die Informationen repräsentieren. Diese Knoten werden untereinander mit Kanten verbunden, die die Relationen zwischen den Knoten beschreiben. Dabei werden je nach Relation unterschiedliche Kantentypen eingesetzt. Gute Beispiele für semantische Netzwerke sind WordNet [2], [3] und GermaNet [1].
Assoziative Netzwerke sind allgemeiner als semantische Netzwerke und verwenden generische Kanten ohne Labels. Diese Kanten können auch gewichtet werden und sind dann in der Lage, etwas über die Stärke der Relation auszusagen. In dieser Arbeit werden assoziative Netzwerke eingesetzt, um die Relationen zwischen Ereignissen abzubilden.

3.2 Spreading Activation (SA)

Spreading Activation Algorithmen (SA Algorithmen) werden verwendet, um Informationen aus semantischen/assoziativen Netzwerken zu extrahieren. Dabei werden zuerst ein oder mehrere Knoten im Netzwerk aktiviert. In den folgenden vom Algorithmus durchgeführten Iterationen wird diese Aktivierung je nach Stärke der Relationen auf benachbarte Knoten weitergegeben. Die Anzahl der Iterationen bestimmt die Tiefe der berücksichtigten Nachbarschaftsbeziehungen.
In dieser Arbeit werden SA Algorithmen für folgende Komponenten eingesetzt:

- ***Relationen zwischen Ereignissen:*** Durch die Aktivierung von Knoten, die Ereignisse repräsentieren, und die anschließende Anwendung von SA Algorithmen kann festgestellt werden, welche anderen Ereignisse mit den aktivierten Ereignissen assoziiert sind und wie stark diese Assoziationen sind.
- ***Generierung der Aktivierungsmuster:*** Für die weitere Analyse verwenden wir sogenannte Aktivierungsmuster, die Auskunft über die im assoziativen Netzwerk aktivierten Bereiche geben. Ein Aktivierungsmuster enthält alle Aktivierungswerte, des assoziativen Netzwerks. Die Anzahl der Einträge entspricht somit der Anzahl der Knoten im Netzwerk.

4 Unüberwachtes Lernen

Unüberwachte Lernalgorithmen analysieren Daten und erkennen Ähnlichkeiten innerhalb dieser Daten. Ähnliche Daten können dann in Gruppen/Cluster zusammengefasst werden. Speziell im Bereich Ereignis-Korrelation bei IDS spielt diese Lernmethode eine sehr wichtige Rolle, da man normalerweise nicht über Daten von Angriffen verfügt, die für das Training verwendet werden können. Beispiele für solche Lernalgorithmen sind Self Organizing Maps [6], Neural Gas und dessen Derivate [4], [5] und der EM Algorithmus [7].

5 Überblick über das Framework

Das vorgestellte Framework besteht aus fünf Schichten, die in Abbildung 2 (Seite 12) dargestellt werden.

- ***L1 – Feature Extraction***: In diesem Schritt werden manuell die kontinuierlichen und die symbolischen Features extrahiert. Kontinuierliche Werte können zu größeren Gruppen zusammengefasst werden, wenn es die Wertbereiche erlauben. Das Erstellen von Gruppen verbessert die Performance ist aber keine Notwendigkeit.
- ***L2 – Erstellen der Knoten***: In diesem Schritt werden die Knoten des assoziativen Netzwerks erstellt. Dabei wird zwischen den unterschiedlichen Features unterschieden:
  - **Kontinuierliche Werte:** Bevor diese Werte in das Netzwerk integriert werden können, muss ein unüberwachter Lernalgorithms angewendet werden. Die gefunden Cluster werden dann für die Knoten im Netzwerk verwendet.
  - **Symbolische Werte:** Diese Werte können direkt als Knoten ins Netzwerk integriert werden.
- ***L3 – Erstellen des Netzwerks***: Die Relationen im assoziativen Netzwerk werden aufgrund des gemeinsamen Auftretens unterschiedlichen Ereignisse erstellt. Die Stärke oder das Gewicht der Relationen hängt dabei von der Anzahl des gemeinsamen Auftretens dieser Ereignisse ab.
- ***L4 – Bestimmen der Aktivierungsmuster***: Die Aktivierungsmuster werden nun mit Hilfe des assoziativen Netzwerks und einem SA Algorithmus erstellt. Dazu werden die Knoten der auftretenden Ereignisse im assoziativen Netzwerk aktiviert und anschließend der SA Algorithmus angewendet, um die Aktivierung auf benachbarte Knoten zu verteilen. Anschließend werden die Aktivierungswerte aller Knoten im Netzwerk extrahiert und in einem Vektor gespeichert. Dieser Vektor entspricht dem Aktivierungsmuster und kann für die weitere Analyse verwendet werden.
- ***L5 – Analyse***: Die erstellten Aktivierungsmuster können nun für die weitere Analyse verwendet werden. Dabei können sowohl überwachte als auch unüberwachte Analyse-verfahren angewendet werden. Außerdem ist es mit Hilfe von SA Algorithmen möglich, die Relationen zwischen Ereignissen zu analysieren.

6 Framework im Detail

In diesem Abschnitt werden die einzelnen Schichten des Frameworks detailliert beschrieben.

6.1 L1 – Feature Extraction

Dieser Layer behandelt die Extraktion der Daten, die von unterschiedlichen Sensoren zur Verfügung gestellt werden. Diese Daten können in zwei Kategorien eingeteilt werden:

- **Daten für die ein Distanzmaß definiert werden kann**: In diesem Fall ist es möglich, die Distanz zwischen zwei unterschiedlichen Messwerten zu definieren. Ein Beispiel dafür wäre ein Feature, das die Anzahl der verlorenen Pakete pro Verbindung angibt. Durch die Existenz des Distanzmaßes ist es nun möglich, die Ähnlichkeiten zwischen unterschiedlichen Fehlerraten zu berechnen – eine Fehlerrate von 10% ist näher zu einer Fehlerrate von 12% als zu einer Fehlerrate von 80%.
- **Daten für die kein Distanzmaß definiert werden kann**: In diesem Fall handelt es sich um symbolische Werte, die nicht über eine Distanz in Relation gesetzt werden können. Beispiele für solche Werte sind Protokoll Typen wie TCP, UDP, ICMP. Es ist nicht möglich, Aussagen der Art "TCP ist näher zu UDP als zu ICMP" zu treffen, da kein Distanzmaß zwischen den Protokolltypen definiert ist.

Diese Einteilung muss für alle Features getroffen werden, da beide Kategorien unterschiedlich behandelt werden müssen. Nach dem Zuordnen der Features können für die Werte mit Distanzmaß Gruppen gebildet werden, die Features mit ähnlichen Wertbereichen kombinieren. Beispiele dafür sind Prozentwerte, die Auskunft über unterschiedliche Fehlerraten geben. Die Gründe für diese Gruppenbildung werden in L2 besprochen.

6.2 L2 – Node Generation

Dieser Layer ist für die Erstellung der Knoten im assoziativen Netzwerk verantwortlich. Dazu muss nun eine Unterscheidung zwischen den in L1 festgelegten Kategorien getroffen werden: Symbolische Daten (z.B. UDP, TCP, ICMP) können direkt als Knoten in das assoziative Netzwerk integriert werden. Bei den Daten für die es möglich ist ein Distanzmaß zu definieren, muss allerdings noch ein weiterer Verarbeitungsschritt eingefügt werden. Eine direkte Repräsentation dieser Werte als Knoten ist nicht möglich, da sie typischerweise nicht abzählbar sind. Selbst wenn dies der Fall ist, macht es keinen Sinn, jeden möglichen Wert als Knoten zu repräsentieren, da dann die Distanz-Information verloren geht. Diese Information sagt etwas über die Ähnlichkeit zwischen einzelnen Werten aus und kann daher verwendet werden um Häufungen (Cluster) von ähnlichen Werten zu finden. Diese Cluster repräsentieren dann die Knoten im assoziativen Netzwerk.
Um diese Cluster zu finden, sind folgende Schritte notwendig:

1. Jede Feature Gruppe, bestehend aus einem oder mehreren Features, wird mit einem unüberwachten Lernalgorithmus analysiert. Dieser Lernalgorithmus findet Cluster innerhalb der Daten. In dieser Arbeit wird hierfür ein Algorithmus verwendet, der auf Growing Neural Gas basiert.
2. Die gefundenen Cluster stellen die Knoten dar, die im assoziativen Netzwerk hinzugefügt werden.
3. Werte können nun wie folgt auf die im Netzwerk vorhandenen Knoten abgebildet werden: Für einen unbekannten Wert wird der am nächsten liegende Cluster bestimmt. Da die Cluster direkt als Knoten repräsentiert werden, stellt der gefundene Cluster auch direkt den im Netzwerk gefundenen Knoten dar.

Die in L1 besprochene Gruppenbildung macht aus Sicht der Komplexität Sinn, da der unüberwachte Lernalgorithmus nur einmal pro Gruppe angewendet werden muss.

6.3 L3 – Network Generation

Es werden die zu analysierenden Daten nun dazu verwendet, die Relationen zwischen den Ereignissen im assoziativen Netzwerk herzustellen. Dazu müssen die Daten in einem Format vorliegen, das das gemeinsame Auftreten von Ereignissen (Sensordaten) beschreibt. Die genaue Definition des gemeinsamen Auftretens hängt nun von der Art der zu analysierenden Daten ab: z.B. ein gewisser Zeitraum oder eine andere Abhängigkeit, die nicht auf Zeit basiert (z.B.: im Bereich Intrusion Detection: Ereignisse mit gleicher Quelladresse).
Sensordaten die im Rahmen dieser definierten Gemeinsamkeit aufgezeichnet wurden, stellen einen Datenvektor dar, der für das Training des assoziativen Netzwerks verwendet wird. Dabei wird für jeden Datenvektor in den Trainingsdaten wie folgt vorgegangen:

1. Die Features (Sensordaten, Ereignisse) pro Datenvektor werden anhand der in L1 getroffenen Einteilung und Gruppierung analysiert. Features, für die kein Distanzmaß festgelegt werden kann, werden direkt auf die Knoten im Netzwerk abgebildet. Für die anderen Features werden die in L2 trainierten unüberwachten Modelle hergenommen und die am nächsten liegenden Cluster bestimmt. Diese Cluster repräsentieren die Knoten im Netzwerk.
2. Es werden nun Links zwischen allen Ereignissen im Datenvektor hergestellt. Ist ein Link zwischen zwei Ereignissen noch nicht vorhanden, so wird dieser mit dem Gewicht 1 initialisiert. Wenn der Link schon vorhanden ist, dann wird dieses Gewicht um 1 erhöht.

Nach der Analyse aller Datenvektoren hat man nun ein assoziatives Netzwerk, das die Assoziationen zwischen Ereignissen über die Kanten zwischen den Knoten herstellt. Das Gewicht der Kanten entspricht der Anzahl des gemeinsamen Auftretens. Die Knoten selbst repräsentieren die Werte der unterschiedlichen Features.
Um in den weiteren Layers den SA Algorithmus anwenden zu können, müssen diese Gewichte normiert werden, so dass Maximalwerte von 1.0 vorkommen. Es gibt unterschiedliche Strategien, um diese Normierung durchzuführen:

- **Globale Maximum Norm**: In diesem Fall wird das maximale Gewicht im Netzwerk genommen und alle anderen Gewichte werden damit normiert.
- **Lokale Maximum Norm**: In diesem Fall wird das maximale Gewicht aller Kanten, die von einem Knoten ausgehen, genommen und alle Kanten dieses Knotens mit diesem Gewicht normiert. Dies wird für alle Knoten durchgeführt.
- **Lokale Summen Norm**: In diesem Fall wird die Summe der von einem Knoten ausgehenden Gewichte genommen und jedes Gewicht mit dieser Summe normiert.

Bei der Anwendung einer lokalen Norm geht im Gegensatz zu einer globalen Normierung die Symmetrie der Kanten verloren.
Für die in dieser Arbeit vorgestellten Ergebnisse wurde die lokale Summennorm verwendet. Die Gründe dafür sind:

- Eine lokale Normierung ermöglicht es, Assoziationen zu verstärken, die zwar global gesehen schwach sind, aber lokal eine wichtige Bedeutung haben.
- Die Summennorm bewirkt, dass für jeden Knoten eine Aktivierungsenergie von 1.0 zur Verfügung steht. D.h. die Aktivierungsenergie wird auf alle ausgehenden Kanten verteilt. Je mehr Kanten vorhanden sind, umso weniger Energie steht für jede einzelne

Kante zur Verfügung. Dies entspricht bereits der Funktion eines einfachen Fanout Faktors.

Nach der Normierung steht nun ein assoziatives Netzwerk zur Verfügung, das für die weiteren Analysen verwendet wird.

6.4 L4 – Generation of Activation Patterns

Die für das Training des assoziativen Netzwerks verwendeten Datenvektoren werden nun verwendet, um die sogenannten Aktivierungsmuster zu generieren. Dazu wird für jeden Datenvektor wie folgt vorgegangen:

1. Es werden die Features aus dem Datenvektor extrahiert und auf die Knoten im Netzwerk abgebildet. Dabei wird gleich wie in L3 vorgegangen.
2. Alle Knoten erhalten den Aktivierungswert 1.0.
3. Durch Anwenden des SA Algorithmus werden die Aktivierungswerte auf die Nachbarn verteilt. Die Stärke der Aktivierung der Nachbarn hängt dabei von dem Gewicht der Kanten ab. In dieser Arbeit wird nur eine Iteration verwendet, da sonst die Aktivierung zu weit im Netzwerk verbreitet wird.

Nach Anwendung des SA Algorithmus auf die im Netzwerk aktivierten Knoten werden die Aktivierungswerte aller Knoten im Netzwerk ausgelesen. Diese Werte werden dann in einen Vektor eingetragen. Dieser Vektor und wird als Aktivierungsmuster bezeichnet.

6.5 L5 – Analysis

In L4 wurden die Aktivierungsmuster für alle Datenvektoren erstellt. Diese Aktivierungsmuster können nun für weitere Analyse Schritte verwendet werden. Dazu gehören:

- **Unüberwachtes Finden von Clustern**: Es kann ein beliebiger unüberwachter Lernalgorithmus eingesetzt werden, der Cluster innerhalb der Aktivierungsmuster findet. Solche Cluster repräsentieren unterschiedliche Vorkommnisse. Im Intrusion Detection Bereich können dies z.B. unterschiedliche Typen von Angriffen sein (DDOS, Syn Flood).
- **Überwachtes Lernen**: Wenn man bereits Informationen über die Zugehörigkeit (z.B. bekannte Angriffe) der zu analysierenden Datenvektoren hat, können überwachte Lernalgorithmen eingesetzt werden, um ein Modell für die Daten zu lernen. Dieses Modell kann dann benutzt werden, um unbekannte Datenvektoren zu klassifizieren.
- **Suche nach assoziierten Ereignissen:** Das assoziative Netzwerk und der SA Algorithmus können dazu verwendet werden, um verwandte Ereignisse zu finden. Dazu werden die Knoten eines oder mehrerer Ereignisse im Netzwerk aktiviert. Danach wir der SA Algorithmus verwendet, um die Aktivierung auf Nachbarknoten zu verteilen. Die Stärke der Aktivierung dieser Nachbarknoten lässt den Schluss über die Stärke der Assoziation zu anderen Ereignissen zu.
- **Vergleich von Aktivierungsmustern:** Die Stärke der Aktivierung der einzelnen Knoten wird im Aktivierungsmuster gespeichert. Der Vergleich zwischen zwei Aktivierungsmustern gibt nun Auskunft darüber, welche Bereiche im Netzwerk (Typen von Ereignissen) aktiv sind und wo die Unterschiede liegen.
- **Anomalieerkennung:** Als Aktivierungsenergie bezeichnen wir die Summe der Aktivierungswerte der Knoten des assoziativen Netzwerks nach Anwendung des SA Algorithmus. Die Aktivierungsenergie gibt Auskunft darüber, wie gut vorgegebene Muster in das trainierte Netzwerk passen. Muster, die ähnlich zu den im Training vorhanden Muster sind, ergeben eine höhere Aktivierungsenergie, da die Links zwischen den einzelnen Features stärker sind. Bei unbekannten Mustern ist es

typischerweise der Fall, dass Werte der einzelnen Features in anderen Kombinationen auftreten und somit Knoten im Netzwerk aktivieren, deren Verknüpfung untereinander schwächer ist. Dies spiegelt sich in der Aktivierungsenergie wider und kann für die Anomalieerkennung genutzt werden.

Feature Extraction

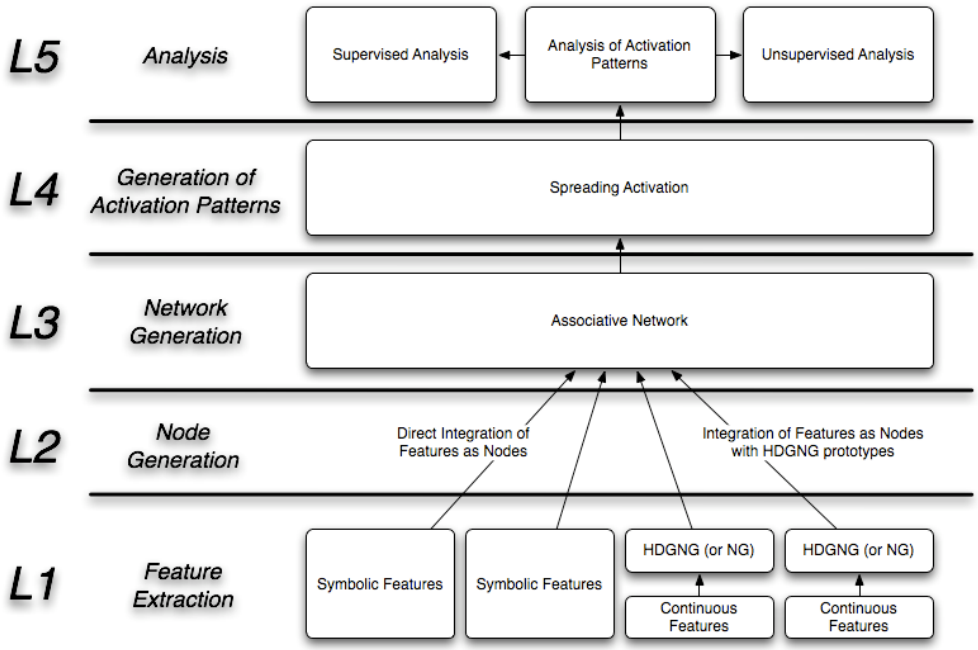

Abbildung 2 - Überblick über das vorgestellte Framework

7 Ergebnisse

Um die Möglichkeiten des Frameworks zu zeigen, wird der *Automobiles* Datensatz aus dem UCI Machine Learning Repository [9] genommen. In diesem Datensatz werden Autos von unterschiedlichen Herstellern aus dem Jahr 1985 beschrieben. Für diese Beschreibung werden 25 symbolische und Distanz-basierte Features genommen. Tabelle 1 beschreibt die Bedeutung der Features und deren möglichen Werte.

ID	Feature	Werte
1	symboling	-3, -2, -1, 0, 1, 2, 3
2	normalized-losses	continous from 65 to 256.
3	make	alfa-romero, audi, bmw, chevrolet, dodge, honda, isuzu, jaguar, mazda, mercedes-benz, mercury, mitsubishi, nissan, peugot, plymouth, porsche, renault, saab, subaru, toyota, volkswagen, volvo
4	fuel-type	diesel, gas
5	aspiration	std, turbo
6	num-of-doors	four, two
7	body-style	hardtop, wagon, sedan, hatchback, convertible
8	drive-wheels	4wd, fwd, rwd
9	engine-location	front, rear
10	wheel-base	continuous from 86.6 120.9.
11	length	continuous from 141.1 to 208.1.
12	width	continuous from 60.3 to 72.3.
13	height	continuous from 47.8 to 59.8.
14	curb-weigth	continuous from 1488 to 4066.
15	engine-type	dohc, dohcv, l, ohc, ohcf, ohcv, rotor.
16	num-of-cylinders	eight, five, four, six, three, twelve, two.
17	engine-size	continuous from 61 to 326.
18	fuel-system	1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi.
19	bore	continuous from 2.54 to 3.94.
20	stroke	continuous from 2.07 to 4.17.
21	compression-ratio	continuous from 7 to 23.
22	horsepower	continuous from 48 to 288.
23	peak-rpm	continuous from 4150 to 6600.
24	city-mpg	continuous from 13 to 49.
25	highway-mpg	continuous from 16 to 54.

Tabelle 1 - Beschreibung der Features des Automobiles Datensatzes

Warum wurde dieser Datensatz gewählt und wie ist die Verknüpfung zum Thema Ereignis Korrelation?

- ***Sensoren:*** Die einzelnen Features können als unterschiedliche Sensoren betrachtet werden, wobei von jedem Sensor unterschiedliche Aspekte abgedeckt werden. Dabei gibt es sowohl Sensoren, die symbolischen Werte (vergleichbar mit Log-Einträgen von IDS Systemen) und distanz-basierte Werte abdecken.
- ***Ereignisse:*** Die Werte, die von den einzelnen Sensoren geliefert werden, stellen die Ereignisse dar, die wir aus dem Bereich Intrusion Detection kennen. Die Verknüpfung

dieser Ereignisse liefert Informationen über die Art des Fahrzeugs (bei IDS: z.B.: Angriff) und ermöglicht Schlussfolgerungen über die Relationen der Ereignisse untereinander (z.B. Motorstärke und Verbrauch, bei IDS: Protokoll-Typ und Anzahl der Verbindungen pro Minute).

- ***Verständlichkeit:*** Die Analyse-Ergebnisse des Algorithmus können leicht überprüft werden, da man mit vielen der Features des Datensatzes gut vertraut ist: z.B.: Länge/Breite des Fahrzeugs, Stärke des Motors, Verbrauch Stadt/Überland usw.
- ***Anzahl der Daten:*** Im Datensatz werden 205 Fahrzeuge beschrieben. Diese überschaubare Menge ermöglicht es, die Möglichkeiten des Frameworks besser zu zeigen.

Für die Evaluierung werden nun vier Analysen durchgeführt:

- ***Unüberwachtes Clustern:*** Dabei werden ähnliche Fahrzeuge in Clustern/Gruppen unüberwacht zusammengefasst.
- ***Relationen zwischen den Features:*** Dabei werden die Relationen der einzelnen Features betrachtet und Fragen der Art "Wie hängen die Stärke des Motors und der Verbrauch zusammen?" beantwortet.
- ***Suche nach verwandten Mustern:*** Dabei wird ein Suchmuster angegeben und nach verwandten Mustern gesucht. Dabei kann es sich bei dem Suchmuster um ein Fahrzeug aus dem Datensatz handeln, bei dem die Werte aller 25 Features festgelegt sind. Weiters ist es auch möglich ausgewählte Features zu bestimmen und nach verwandten Mustern zu suchen: z.B.: Festlegung von Verbrauch und Länge.

Für die Analyse der Daten wurden zwar alle Features verwendet, um aber die Relevanz der Cluster zu zeigen, beschränkt sich die Tabelle dabei aber auf ein paar wenige leicht verständliche Features: Marke, Länge, PS, MPG1 (Miles Per Gallon Stadt), MPG2 (Miles Per Gallon Überland), Zylinder, Antrieb (Vorderrad, Hinterrad, Vierrad), Treibstoff (Diesel/Benzin) und max. RPM.

7.1 Unüberwachtes Finden von Clustern

Bei dieser Analyse werden die Operationen der Layer 1-4 auf den Datensatz angewendet. Die gewonnen Aktivierungsmuster werden anschließend mit einem unüberwachten Lernalgorithmus analysiert. Dabei werden ähnliche Aktivierungsmuster in Gruppen (Cluster) zusammengefasst. Die unüberwachte Analyse spielt hier eine wichtige Rolle, da wir davon ausgehen, dass wir keine weiteren Informationen über die zu analysierenden Daten haben. Bei dem vorliegenden Datensatz gibt es aber den Vorteil, dass wir über die einzelnen Features und deren Relationen Bescheid wissen und somit eine bestimmte Clusterbildung erwarten können. In diesem Fall darf man erwarten, dass Autos anhand ihrer Stärke und Größe gruppiert werden. Dies lässt sich damit erklären, dass viele der vorhandenen Features diese Eigenschaften beschreiben und untereinander korreliert sind. Ein Beispiel dafür wäre ein Auto mit einem hohen Treibstoffverbrauch. In diesem Fall wird mit ziemlicher Sicherheit auch ein starker Motor verwendet werden. Dies reflektiert sich z.B. auch in den Features Leistung und Zylinder. Die Ergebnisse der unüberwachten Clusterbildung sind in Tabelle 2 zu sehen.

ID (Anzahl)	***Marke***	***Länge***	***PS***	***MPG1***	***MPG2***	***Zyl.***	***Antrieb***	***B/D***	***RPM***
C1 (71)	Toyota	188	92	19	24	4	RWD	Gas	5364
C2 (52)	Toyota	167	69	39	37	4	FWD	Gas	5364
C3 (9)	Toyota	176	69	30	32	4	FWD	Gas	4874
C4 (4)	Mazda	167	92	16	24	4	RWD	Gas	5916
C5 (24)	Toyota	172	92	23	32	4	FWD	Gas	5364
C6 (8)	Toyota	167	69	30	32	4	FWD	Gas	5364

C7 (13)	Toyota	176	92	23	32	4	FWD	Gas	4874
C8 (8)	Mercedes	188	180	16	24	4	RWD	Gas	4874
C9 (10)	Subaru	172	92	23	32	4	FWD	Gas	4874
C10 (6)	Toyota	176	114	23	28	4	FWD	Gas	4874

Tabelle 2 - Ergebnisse des unüberwachten Clusterings

Für die Tabelle wird für jeden Cluster der Mittelwert aller Aktivierungsmuster pro Cluster berechnet und ausgegeben. Da Toyota einen großen Bereich von unterschiedlichen Fahrzeugen abdeckt, ist diese Marke bei den Mittelwerten oft aktiviert. In der Tabelle kann man aber anhand des Verbrauchs und der PS sehen, dass die Fahrzeuge hauptsächlich nach diesen Kriterien gruppiert werden. Um einen guten Überblick über die 10 Cluster zu bekommen, wird auf die Liste der Gesamtergebnisse (***cluster.csv***) verwiesen.

7.2 Relationen

Bei dieser Analyse stellt man sich die Frage wie die Relationen zwischen unterschiedlichen Features sind. Ein Beispiel dafür ist die Frage "Mit welchen Werten für Motor Leistung, Verbrauch ist die Marke Porsche assoziiert". Es werden nun einige Beispiele in Tabelle 3 gezeigt:

	Marke	***Länge***	***PS***	***MPG1***	***MPG2***	***Zyl.***	***Antrieb***	***B/D***	***RPM***
Marke: ***Porsche test-b.csv***	Porsche	167	205	16	24	6	RWD	Benz.	5961
Marke: ***VW test-c.csv***	VW	172	92	27	32	4	FWD	Benz.	5364
MPG2: ***40 test-d.csv***	Chevrolet	157	70	37	42	4	FWD	Benz.	5364
PS: ***200 test-g.csv***	Porsche	167	205	16	24	6	RWD	Benz.	5961

Tabelle 3 - Relationen

Die in der Tabelle dargestellten Ergebnisse entsprechen den Erwartungen. Die Marke Porsche ist mit einer großen PS Zahl, einem Hinterradantrieb und einem hohen Verbrauch assoziiert. VW entspricht hier erwartungsgemäß genau dem Gegenteil. Das dritte Beispiel aktiviert einen Knoten der einem niedrigen Verbrauch entspricht. Hier sieht man, dass die Marke Chevrolet und ein schwacher Motor damit assoziiert sind. Auch das vierte Beispiel bringt die erwarteten Ergebnisse. Eine hohe PS Anzahl ist mit Porsche und einem hohen Verbrauch assoziiert. Diese Ergebnisse können leicht durch eine manuelle Überprüfung des Datensatzes verifiziert werden.

7.3 Suche

Hier soll ein Aktivierungsmuster vorgegeben werden und nach verwandten Mustern gesucht werden. Es kann sich hierbei um bestehende Aktivierungsmuster handeln, die im Zuge des Analyseprozesses der L1-L4 erzeugt wurden (z.B. Verwendung eines Autos aus dem Datensatz) oder um ein neues Aktivierungsmuster handeln, das für die Suche erstellt wird. Im zweiten Fall ist es auch möglich, nur wenige Features festzulegen, daraus ein Aktivierungsmuster zu erstellen und anschließend nach verwandten Mustern zu suchen. In Tabelle 4 werden nun Beispiele für beide Arten der Suche gezeigt:

	Marke	***Länge***	***PS***	***MPG1***	***MPG2***	***Zyl.***	***Antrieb***	***B/D***	***RPM***
M1	***VW***	***172***	***58***	***37***	***48***	***4***	***FWD***	***Diesel***	***4874***

test-e.csv									
4	Toyota	172	58	37	49	4	FWD	Diesel	4319
6	VW	167	92	27	32	4	FWD	Benz.	5364
11	Mazda	177	70	37	42	4	FWD	Diesel	4874
204	Jaguar	188	260	16	18	12	RWD	Benz.	4874
205	Mercedes	200	180	16	18	8	RWD	Benz.	4319
M2 test-f.csv	***Jaguar***	***200***	***180***	***16***	***18***	***6***	***RWD***	***Benz.***	***4874***
3	BMW	200	180	16	18	6	RWD	Benz.	5364
11	Peugeot	200	92	19	24	4	RWD	Benz.	4874
15	Audi	188	152	16	18	5	FWD	Benz.	5364
204	Honda	147	70	30	37	4	FWD	Benz.	5961
205	Chevrolet	147	58	46	48	3	FWD	Benz.	4874
M3 test-b.csv	***Porsche***	-	-	-	-	-	-	-	-
1	Porsche	167	205	16	24	6	RWD	Benz.	5961
6	Nissan	172	205	16	24	6	RWD	Benz.	5364
8	Mazda	172	205	16	24	6	RWD	Benz.	5364
203	VW	172	70	37	42	4	FWD	Diesel	4319
205	Toyota	176	70	30	32	4	FWD	Diesel	4874
M4 test-c.csv	***VW***	-	-	-	-	-	-	-	-
1	VW	172	70	37	42	4	FWD	Diesel	4319
13	Mazda	176	70	37	42	4	FWD	Diesel	4874
14	Toyota	167	58	37	42	4	FWD	Diesel	4874
203	Porsche	167	205	16	24	6	RWD	Benz.	5961
205	Jaguar	200	180	16	18	6	RWD	Benz.	4874
M5 test-a.csv	-	***170***	***100***	-	***40***	-	-	-	-
1	VW	172	70	37	42	4	FWD	Diesel	4319
2	Subaru	172	92	26	32	4	FWD	Benz.	4874
12	Nissan	172	92	26	32	4	FWD	Benz.	5364
203	Jaguar	188	260	16	18	12	RWD	Benz.	4874
205	Mercedes	200	180	16	18	8	RWD	Benz.	4319

Tabelle 4 - Suchergebnisse

Die Ergebnisse aller Beispiele entsprechen auch den Erwartungen. In allen Fällen werden typischerweise als beste Ergebnisse zuerst Autos der gleichen Marke und dann Fahrzeuge mit ähnlichen Merkmalen ausgewählt. Die Autos mit der geringsten Ähnlichkeit werden auch angegeben (203-205). Für die Beispiele M1 und M2 werden dabei zwei komplette Autos als Suchmuster verwendet. D.h. alle der Features werden im Netzwerk assoziiert und anschließend mit den anderen Aktivierungsmustern verglichen. Bei den Beispielen M3-M5 werden nur einzelne Features selektiert und aktiviert. Im Falle von M3 wird nur Porsche aktiviert. Nach Anwendung des SA Algorithmus werden aber durch das assoziative Netzwerk nun auch die mit Porsche assoziierten Features aktiviert (siehe dazu das Porsche Beispiel bei der Analyse Relationen). Aus diesem Grund können auch Autos anderer Hersteller gefunden werden, die Werte haben, die für einen Porsche typisch sind. Bei M5 werden 3 Features vorgegeben: Die Länge, die PS und MPG1. Auch hier können wieder Autos gefunden werden, die am ehesten diesen Vorgaben entsprechen.

7.4 Anomalieerkennung

Hier wird gezeigt, dass das Framework auch für die Erkennung von Anomalien verwendet werden kann. Im Falle von IDS macht es Sinn ein Modell für den normalen Verkehr zu

erstellen. Neue Muster, die während dem Training nicht vorhanden waren, und von den trainierten abweichen, werden als Anomalien bezeichnet und können z.B. Angriffe darstellen. Im Falle des hier analysierten Datensatzes wird die Anomalieerkennung anhand der zuvor gefundenen Cluster evaluiert. Dazu wird nur mit den Daten von C1 trainiert. Es handelt sich dabei um Autos unterschiedlicher Hersteller, die einen mittleren Verbrauch haben. Die anderen Cluster enthalten dann entweder stärkere Autos oder Fahrzeuge mit wenig PS und großer Reichweite. In Tabelle 5 wird die durchschnittliche Aktivierungsenergie pro Cluster (Am) und die Standardabweichung (As) angegeben. Man kann hier sehen, dass die durchschnittlichen Energiewerte bei C2-C10 immer unter dem der Trainingsdaten liegen. Um dies besser darzustellen, wird in Abbildung 3 die Energie für jede Instanz der Trainingsdaten (rote Linie) und der Anomaliedaten (schwarze Linie) gezeigt. Bis auf einen kleinen Teil der Anomalien könnten diese im Prinzip durch den Vergleich mit einem festgelegten Schwellwert erkannt werden.

Data	***Am***	***As***
Train (C1)	668	45
C2	459	40
C3	487	43
C4	473	8
C5	537	43
C6	422	20
C7	570	27
C8	524	60
C9	513	39
C10	636	10

Tabelle 5 - Aktivierungsenergie der Trainingsdaten und Anomaliedaten

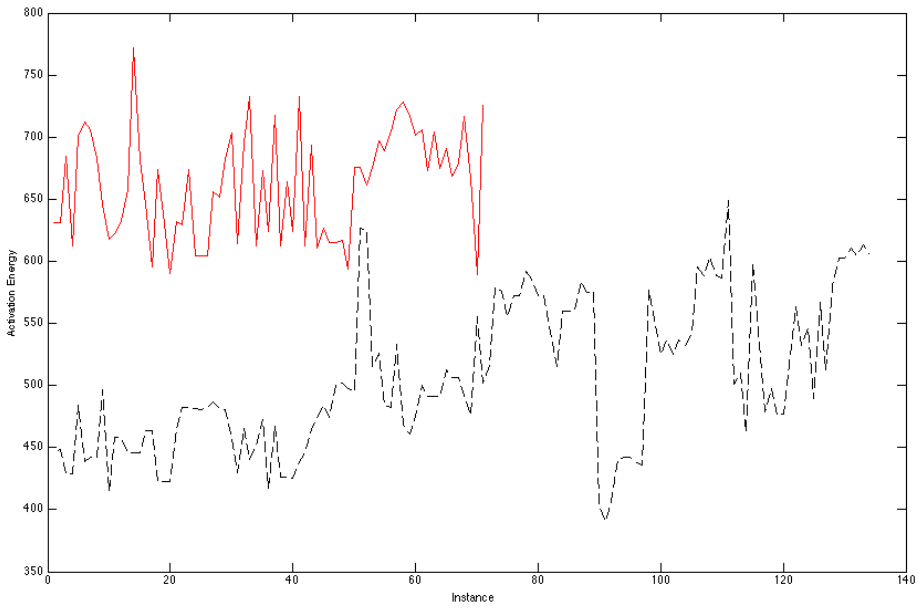

Abbildung 3 - Aktivierungsenergie der Trainingsdaten (rot - oben) und der Anomalien (schwarz - unten)

8 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Framework vorgestellt, das Ereignis-Korrelation basierend auf maschinellem Lernen ermöglicht. Besonderes Augemerk wird dabei auf die unüberwachte Analyse von Ereignissen gelegt. Anhand eines leicht interpretierbaren Datensatzes werden die Möglichkeiten des Frameworks aufgezeigt.
Bei der Evaluierung des Frameworks wurden noch Punkte erkannt, die in Zukunft genauer analysiert werden.

- **Verbesserungen beim assoziativen Netzwerk und bei den eingesetzten SA Algorithmen:** Um die Ergebnisse des SA Algorithmus zu verbessern, müssen noch weitere SA Strategien integriert werden: Beispiele dafür sind die Integration eines FanOut Faktors, der den Einfluss von Knoten mit einer großen Anzahl von Verbindungen zu anderen Knoten verringert. Weiters werden noch Verbesserungen bei der Art der Aktivierungsfunktion benötigt.
- **Anwendung auf unterschiedliche Datensätze:** Um mehr über die Eigenschaften des Frameworks zu erfahren, müssen noch weitere Datensätze aus unterschiedlichen Bereichen analysiert werden.

9 Referenzen

[1] **Book**
Quillian, M. R. (1968). Semantic memory. In M. L. Minsky (Ed.), Semantic information processing. Cambridge, MA: MIT Press

[2] **Book**
Fellbaum, C. (1998). Wordnet: An Electronic Lexical Database, Bradford Books

[3] **Internet reference**
Miller, A. et al., WordNet – a lexical database for the English language, retrieved December 4, 2008 from http://wordnet.princeton.edu/.

[4] **Journal paper**
Martinetz, T.M. et al. (1991). A neural-gas network learns topologies, in T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, Artificial Neural Networks, pages 397-402. North-Holland, Amsterdam

[5] **Book**
Fritzke, B. (1994). A growing neural gas network learns topologies, in Neural Information Processing Systems, pages 625-632

[6] **Book**
Kohonen T. (1995). Self Organizing Maps, Springer

[7] **Book**
McLachlan G. (2008). The EM Algorithm and Extensions, 2nd Edition, Wiley

[8] **Chapter in proceedings**
Hamp, B. (1997). GermaNet – a Lexical-Semantic Net for German, in Proceedings of ACL workshop Automatic Information Extaction and Building of Lexical Semantic Resources for NLP Applications, Madrid

[9] **Website**
Asuncion, A., Newman, D. (2007). UCI machine learning repository, retrieved February 6, 2009 from http://archive.ics.uci.edu/ml/.

10 Anhänge

Alle Ergebnisse sind in den folgenden beigelegten Dateien notiert. Bei den relevanten Stellen im Dokument wird auf diese Dokumente referenziert.

test-a.csv
test-b.csv
test-c.csv
test-d.csv
test-e.csv
test-f.csv
test-g.csv
cluster.csv

Historie

Version	Datum	Kommentar
1.0	06.02.2009	Finale Version
Ersteller		
Peter Teufl		

507