

UNIVERSITÄT LEIPZIG
Fakultät für Mathematik und Informatik
Institut für Informatik

Anwendung von Ontologielernverfahren im medizinischen Bereich

Masterarbeit

Leipzig, 26. April 2011
Betreuer: Dr. rer. nat. Jens Lehmann

vorgelegt von
Christian Kötteritzsch
geb. am: 19.03.1986
Studiengang Informatik

Inhaltsverzeichnis

1	Einleitung	1
2	Stand der Forschung und verwandte Arbeiten	4
3	Medizinische Grundlagen	9
3.1	idiopathische Parkinson	9
3.2	Brustkrebs und Mammographie	11
3.2.1	Brustkrebs	11
3.2.2	Mammographie	14
3.3	Elektrische Impedanz Tomographie	16
3.4	Elektrokardiographie	17
4	Semantik Web Grundlagen	19
4.1	RDF	19
4.2	Web Ontology Language OWL	21
5	Machine Learning Grundlagen	24
5.1	Die Algorithmen	24
5.1.1	Der t-Test	24
5.1.2	Der Algorithmus von Aleph	24
5.1.3	Die Algorithmen von Weka	25
5.1.4	Der Refinement Operator	28
5.2	Die Werkzeuge	32
5.2.1	Aleph	32
5.2.2	Weka	32
5.2.3	DL-Learner	33

6	Vorhersage von mutagenen Eigenschaften	37
6.1	Datensatz	37
6.2	Methoden	40
6.3	Resultate und Schlussfolgerungen	40
7	Suramin-ähnliche Substanzen zur Krebsbekämpfung	52
7.1	Datensatz	53
7.2	Methoden	54
7.3	Resultate und Schlussfolgerungen	55
8	Monitoring von Parkinsonpatienten	61
8.1	Datensatz	61
8.2	Methoden	64
8.3	Resultate und Schlussfolgerungen	64
9	Vorhersage von Herzerkrankungen	71
9.1	Datensatz	71
9.2	Methoden	74
9.3	Resultate und Schlussfolgerungen	74
10	Vorhersage von Brustkrebs anhand von Mammographie Bildern	82
10.1	Datensatz	82
10.2	Methoden	83
10.3	Resultate und Schlussfolgerungen	85
11	Breast Tissue	92
11.1	Datensatz	92
11.2	Methoden	93
11.3	Resultate und Schlussfolgerungen	95

12	Cardiotocography	108
12.1	Datensatz	108
12.2	Methoden	111
12.3	Resultate und Schlussfolgerungen	112
13	Schlussfolgerungen	135
14	Zusammenfassung und Ausblick	138
A	Grafiken für die Breast Tissue Wissensbasis	139
A.1	Grafiken für die Variation des Noise-Parameters	139
A.2	Grafiken für die benötigte Laufzeit bei Variation des Noise Parameters	146
A.3	Grafiken für die fest eingestellte Laufzeit von 10, 100, 300 und 1000 Sekunden	150
B	Grafiken für die Cardiotocography Wissensbasis	154
B.1	Grafiken für die Variation des Noise-Parameters	154
B.2	Grafiken für die benötigte Laufzeit bei Variation des Noise Parameters	167
B.3	Grafiken für die fest eingestellte Laufzeit von 10, 100, 300 und 1000 Sekunden	180

1 Einleitung

Im medizinischen Bereich entstehen in der Forschung heutzutage eine für den Menschen unüberschaubare Menge an Daten. Um diese Daten auswerten zu können, werden Experten benötigt, um die Zusammenhänge der Daten erkennen zu können. Da dieser Vorgang viel Zeit in Anspruch nimmt, ist man seit einiger Zeit dazu übergegangen, solche Probleme mit Hilfe des Computers zu lösen. Dabei nutzte man früher oft Ansätzen der klassischen Statistik, künstlichen neuronalen Netzen oder Bayes Klassifizierern. Die Lösungen, die diese Ansätze errechneten, waren für Mediziner oft schwer verständlich. Ein weiterer Nachteil war, dass man kein Hintergrundwissen, welches schon über die Probleme bekannt war, kodieren konnte. Aus diesem Grund ist man dazu übergegangen, diese medizinischen Probleme mit Hilfe von Induktiver Logik Programmierung (ILP) zu lösen. ILP Systeme produzierten Lösungen, die für Mediziner einfacher verständlich waren, als solche die auf Ansätzen der klassischen Statistik, künstlichen neuronalen Netzen oder Bayes Klassifizierern basieren. Ein weiterer Vorteil von ILP Systemen war die Möglichkeit Hintergrundwissen direkt in der Wissensbasis kodieren zu können. Dieses Wissen konnte genutzt werden, um mit Hilfe der Daten neue und unbekannte Zusammenhänge zu entdecken. Im Kapitel 2 werden einige medizinische Probleme erläutert und beschrieben welche Vorteile die Lösung dieser mit Hilfe von ILP Systemen bieten.[BK94]

Ein weiteres Problem ist, dass es eine Vielzahl von Werkzeugen zum Lösen von medizinischen Problemen gibt. Einige wurden sogar nur für die Lösung spezieller Probleme geschaffen. Dabei unterstützen die Programme oft unterschiedliche Dateiformate, so dass eine Umwandlung der Daten notwendig ist, bevor man einen Vergleich zwischen den verschiedenen Werkzeugen durchführen kann. Bei dieser Umwandlung kann es dazu kommen, dass nicht alle Informationen von einem Datenformat in ein anderes überführt werden können. Somit kann ein Vergleich schwierig sein.

Das Problem, der Unüberschaubarkeit der Daten beschränkt sich nicht allein auf den medizinischen Bereich. Denn auch durch das ständige Wachstum des Internets entsteht eine für den Menschen unüberschaubare Menge an Daten und das gezielte Finden von Informationen zu einem bestimmten Themengebiet ist schwierig. Es können auch keine semantischen Zusammenhänge zwischen den Daten erkannt werden, da diese oft für eine menschliche Darstellung ausgelegt sind und somit für Maschinen schwierig zu interpretieren sind. Um dieses Problem lösen zu können, versucht man mit Hilfe der Semantik Web bestehende Daten mit maschinenlesbaren Informationen zu erweitern. Dadurch wird es möglich automatische Interpretationen und Weiterverarbeitung der Informationen zu gewährleisten. Die Informationen werden in Wissensbasen, den sogenannten Ontologie zur Verfügung gestellt. Dabei sind sie hierarchisch aufgebaut und enthalten logische Beziehungen für die einzelnen Objekte. Es wird somit möglich die Informationen maschinell zu verarbeiten und zu interpretieren, sodass der Benutzer bessere Ergebnisse bei der Suche erzielt. Auch im medizinischen Bereich existieren bereits Wissensbasen, die als Ontologien kodiert sind. Hierbei handelt es sich oft um Wissensbasen, die im

Internet zur Verfügung gestellt wurden.[HKRS08]

In dieser Arbeit werden verschiedene Werkzeuge, die für die Lösung medizinischer Probleme geeignet sind, verglichen. Dabei soll überprüft werden, ob es Vorteile bei der Nutzung von Ontologien gegenüber von ILP Systemen gibt. Zu diesem Zweck wird der DL-Learner, welcher die Induktive Logik Programmierung um Beschreibungslogik erweitert, herangezogen. Dieser wird mit einem ILP System, nämlich Aleph und dem Machine Learning Framework Weka verglichen. Die drei Werkzeuge unterstützen dabei unterschiedliche Dateiformate. Die Informationen bei Aleph sind als Prologdateien gespeichert, bei Weka werden alle Daten als Flat File Database gespeichert und bei dem DL-Learner sind die Informationen als OWL-Ontologien gespeichert. Für den Vergleich wurden zwei Wissensbasen verwendet, die mit Hilfe von Prolog gespeichert sind, nämlich die Wissensbasis zum Erkennen von mutagenen Eigenschaften und die Wissensbasis zum Finden von Suramin-ähnlichen Substanzen zur Krebsbekämpfung. Des Weiteren wurden fünf Wissensbasen, die mit Hilfe einer Flat File Database gespeichert wurden verwendet. Bei diesen Wissensbasen handelt es sich um das Monitoring bei Parkinsonpatienten, die Vorhersage von Herzerkrankungen, die Vorhersage von Brustkrebs anhand von Mammographiebildern, das Erkennen von Läsionen oder Tumoren mit Hilfe der elektrischen Impedanz Tomographie und eine Wissensbasis zur Erkennung des Gesundheitszustandes eines Fötus anhand von Kardiogrammen. Um einen Vergleich der einzelnen Werkzeuge zu gewährleisten, musste in einem ersten Schritt die jeweilige Wissensbasis in eine OWL-Ontologie umgewandelt werden. Im Anschluss daran wurden die einzelnen Programme ausgeführt um neue Informationen aus den Wissensbasen zu lernen. Es wurden dazu alle gelernten Ergebnisse protokolliert, um sie in einem nächsten Schritt auswerten zu können. Dabei wurde überprüft, welche Regeln gelernt wurden und ob sich diese bei den verschiedenen Werkzeugen unterscheiden. Des Weiteren wurde auf die erzielte Genauigkeit der einzelnen Lösungen geachtet.

In diesem Abschnitt wird die Gliederung der Arbeit genauer vorgestellt. Im zweiten Kapitel wird anhand einiger medizinischer Probleme verdeutlicht, welche Vorteile ILP Systeme gegenüber Ansätzen der klassischen Statistik, künstlichen neuronalen Netzen oder Bayes Klassifizierern bieten. Des Weiteren werden einige verwandte Arbeiten vorgestellt. Im Anschluss daran werden im Kapitel drei die medizinischen Grundlagen geklärt. Dabei wird auf Suramin und Suramin-ähnliche Substanzen, die Parkinsonerkrankung, Brustkrebs und Mammographie, die Elektrische Impedanz Tomographie und die Elektrokardiographie eingegangen. Das vierte Kapitel beschäftigt sich mit der Idee des Semantik Web, stellt die Beschreibungssprachen RDF und OWL vor und beschäftigt sich mit Manchester OWL Syntax, die dazu dient komplexe Klassenbeschreibungen leichter lesbar aufzuschreiben. Im fünften Kapitel werden dann die Machine Learning Grundlagen geklärt. Insbesondere wird auf die verwendeten Algorithmen und Werkzeuge, die zum Lernen benötigt wurden, eingegangen. In den folgenden sieben Kapiteln werden die einzelnen medizinischen Lernprobleme vorgestellt. Dabei wird in jedem Kapitel der Datensatz, die Umwandlung in eine OWL-Ontologie, die durchgeführten Tests zum Ler-

nen der Klassenbeschreibungen und die Ergebnisse und Schlussfolgerungen aus den Lernprozessen vorgestellt. Das Kapitel sechs beschreibt das medizinische Lernproblem der Vorhersage von mutagenen Eigenschaften. Danach wird im Kapitel sieben das Lernproblem für Suramin-ähnliche Substanzen zur Krebsbekämpfung vorgestellt. Im Kapitel acht wird die Wissensbasis Monitoring von Parkinsonpatienten vorgestellt. Anschließend wird im neunten Kapitel das Lernproblem zur Vorhersage von Herzerkrankungen genauer vorgestellt und Kapitel zehn beschäftigt sich mit der Vorhersage von Brustkrebs anhand von Mammographiebildern. Kapitel elf beschäftigt sich mit der Erkennung von Läsionen oder Tumoren in der Brust anhand der elektrischen Impedanz Tomographie. Im Kapitel zwölf wird das letzte medizinische Lernproblem vorgestellt, welches sich mit der Erkennung des Gesundheitszustandes eines Fötus anhand von Kardiogrammen beschäftigt. Im Anschluss daran werden in Kapitel dreizehn die Erkenntnisse der Lernvorgänge der letzten sieben Kapitel noch einmal zusammengefasst und auf Besonderheiten eingegangen. Danach wird im Kapitel dreizehn, welches auch das letzte Kapitel ist, die komplette Arbeit noch einmal zusammengefasst und ein Ausblick auf zukünftige Arbeiten gegeben, die noch durchzuführen sind.

2 Stand der Forschung und verwandte Arbeiten

In der Forschung im medizinischen Bereich entsteht heutzutage eine unüberschaubare Menge an Daten, bei der es für Menschen schwierig ist alle sinnvollen Zusammenhänge zu erkennen. Deshalb benötigt man Verfahren, um diese Menge an Daten besser auswerten zu können und um Zusammenhänge ableiten zu können. Aus diesem Grund wird seit vielen Jahren mit Hilfe von Induktiver Logik Programmierung (ILP) versucht diese Probleme zu lösen. Denn bei dieser ist es möglich, Hintergrundwissen direkt in die Wissensbasis zu kodieren. In diesem Kapitel werden verschiedene Experimente dargestellt, in denen ILP Programme mit anderen Verfahren, z.B. aus dem Bereich der klassischen Statistik, künstlichen neuronalen Netzen oder Bayes Klassifizierern verglichen werden. Dabei zeigt sich, dass ILP Programme mindestens genau so gut arbeiten, wie die anderen Verfahren und in einigen Fällen Resultate aufdecken, die zuvor nicht gefunden werden.

In [BK94] werden zwei verschiedene Lernprobleme, die mit Hilfe von Induktiver Logik Programmierung gelöst wurden, vorgestellt und genauer erläutert. Dabei beschäftigen sich die ersten beiden Lernprobleme, nämlich Medikamenten-Design und Vorhersage der Sekundärstruktur von Proteinen, mit der biomolekularen Modellierung. Das Ziel ist es die Beziehungen zwischen den verschiedenen chemischen Formeln, die dreidimensionale Struktur und die Wichtigkeit der Funktion von Molekülen zu verstehen.

Bei dem Lernproblem Medikamenten-Design wurden zwei verschiedene Aufgaben, die Hemmung von Escherichia Coli Dihydrofolate Reductase (DHFR) und die Hemmung des Wachstums der Tumorzellen von Ratten/Mäuse durch Triazines, mit Hilfe der Anwendung GOLEM¹ getestet. Die Wissensbasis wurde mit Hilfe von Prolog abgebildet. Dabei wurden die chemischen Eigenschaften, mit Hilfe einer Heuristik den einzelnen Substanzen zugewiesen. Die Daten die durch die Anwendung von Golem entstanden sind, wurden mit anderen Algorithmen aus den Bereichen symbolische Methoden, neuronale Netzwerke und klassische statistischen Methoden verglichen. Die Algorithmen CART und M5 aus dem Bereich der Klassifikation mit Hilfe von Bäumen, der Algorithmus back-propagation aus dem Bereich neuronale Netze und lineare Regression und nearest-neighbour aus dem Bereich klassische statistische Methoden wurden angewandt. Für das erste Problem wurde eine five-fold-cross-validation und für das zweite Problem eine six-fold-cross-validation durchgeführt. Die Ergebnisse dieser Studie zeigen, dass es keine größeren Unterschiede in den Resultaten der verschiedenen Algorithmen gibt, die über 5% liegen. Jedoch sind die Ergebnisse die von GOLEM produziert wurden viel einfacher für Chemiker zu lesen, als die Ergebnisse der anderen Algorithmen. Dies hängt damit zusammen, dass die Sprache der Logik aus ILP Programmen die ist, die Chemiker benutzen. Zusammenfassend ist zu diesen beiden Problem zu sagen, dass es möglich ist, nützliche und leicht verständliche Regeln zu finden.

¹Quelle: <http://www.doc.ic.ac.uk/shm/Software/golem/>

Das zweite Lernproblem Vorhersage der Sekundärstruktur von Proteinen wurde ebenfalls mit Hilfe von GOLEM durchgeführt. Dabei sind Proteine biologische Makromoleküle, deren Aktivität durch die komplexe dreidimensionale Struktur bestimmt ist. Diese Struktur zu bestimmen ist aber kompliziert, da ungefähr 500 verschiedene dieser Strukturen bekannt sind. Andererseits kann die chemische Struktur, die als Folge von Aminosäuren bekannt ist, einfach mit Hilfe der Gensequenz bestimmt werden. Experimentell konnte nachgewiesen werden, dass es theoretisch möglich ist aus dieser Folge von Aminosäuren, die dreidimensionale Struktur zu bestimmen. Um dies mit Hilfe von GOLEM zu testen, wurden Trainingsdaten mit 12 Proteinen und Testdaten mit 4 Proteinen, die nicht zu den 12 aus den Trainingsdaten gehörten, verwendet. Dazu wurden die Positionen der α -Helices, die Informationen über die Aminosäuresequenz, die Hintergrundinformationen über Eigenschaften der 20 Aminosäuren und eine eingebaute Arithmetik, welche Informationen über die Sequenzbeziehungen der Aminosäuren enthält, angegeben. Die ersten Ergebnisse von GOLEM waren sehr unterschiedlich, so dass ein Bootstrapping Verfahren eingeführt wurde, damit GOLEM die Daten glätten konnte. Als Genauigkeit wurde die Anzahl der Aminosäuren, die richtig vorhergesagt wurden, genommen. Das finale Resultat hatte eine Genauigkeit zwischen 78% und 81%, das im Vergleich zu der Genauigkeit von 76% bei neuronalen Netzen sogar noch etwas besser ausfiel. Zusammenfassend ist zu sagen, dass die Genauigkeit bei der Vorhersage der Sekundärstruktur von Proteinen mindestens genau so gut ist wie die bei neuronalen Netzen.[BK94]

In dem Artikel [QAY⁺02] geht es um die Erkennung von Prostatakrebs mit Hilfe mehrerer Biomarker, da das Prostata-spezifische Antigen keine genaue Unterscheidung zwischen Prostatakrebs und gutartiger Prostata Hyperplasia zulässt. Da es einen Durchbruch in der Massenspektrometrie, Oberflächen optimierten Laser Desorption/Ionisazion(SELDI) gegeben hat, ist es möglich mehrere Proteine auf einem Protein-Chip zu erfassen und darzustellen. Dazu werden 386 Beispiele, 197 mit Prostatakrebs, 92 mit gutartiger Prostata Hyperplasia und 96 gesunde Männer, zufällig in Trainingsdaten mit 326 Männern und Testdaten mit 60 Männern geteilt. Diese Instanzdaten werden mit Hilfe von SELDI analysiert. Aus den so gewonnenen Daten werden die 124 höchsten Werte mit Hilfe eines verstärkten Decision Tree Algorithmus analysiert, so dass eine Klassifizierung erstellt wird, die Prostatakrebs von der Gruppe ohne Krebs unterscheidet. Die so entstandene Klassifizierung wird mit Hilfe der 60 Testdaten, wovon 30 Prostatakrebs haben, 15 Prostata Hyperplasia und 15 gesund sind, überprüft. Daraus wurden zwei verschiedene Klassifizierer entwickelt. Der erste heißt AdaBoost Klassifizierer, welcher die Männer mit Krebs vollständig von den Männern ohne Krebs trennt und somit eine Genauigkeit von 100% erreicht. AdaBoost ist ein Algorithmus welcher einen schwachen Lernalgorithmus nutzt um die Genauigkeit von diesem zu verbessern. Dabei kombiniert er die Summe der wahrscheinlichen Vorhersagen und das Ergebnis des schwachen Lernalgorithmus, um ein besseres Ergebnis zu erzielen. Für weitere Informationen zu AdaBoost kann man den Artikel [FS97] lesen. Der zweite Klassifizierer ist der Boosted Decision Stump Feature Selection classifier. Bei diesem Klassifizierer ist es einfacher die

Ergebnisse zu interpretieren. Dabei erreichte dieser nur eine Genauigkeit von 97% bei den Testdaten. Dies resultiert daraus, dass er nur die 21 höchsten Werte nutzt. Dabei ist ein Entscheidungsstumpf ein Entscheidungsbaum mit nur einem Teilungsvorgang. Auf diesen einen Entscheidungsstumpf wird AdaBoost angewandt, um aus diesem schwachen Klassifizierer einen akkuraten Klassifizierer zu machen. Anschließend wird der so kombinierte Klassifizierer, der aus einem Entscheidungsstumpf, einigen Basisklassifizierern und deren Mitgliedern besteht, zu einem Komitee zusammengefasst, das mit Hilfe eines Mehrheitsvotes eine Entscheidung trifft. Dies wird so lange wiederholt bis eine gewisse Anzahl an Stümpfen entstanden ist.[QAY⁺02]

Der Artikel [MBL08] behandelt das Problem der Auswahl der am Besten geeigneten Embryos, die bei künstlicher Befruchtung der Mutter eingesetzt werden. Dabei spielen morphologische Eigenschaften wie embryonale Fragmentierung, Spaltungsphase und blastomere Einheit eine wichtige Rolle. Der mit Hilfe des Bayes'schen Klassifizierer durchgeführte Analyse, soll Bilder von Embryos nach Wahrscheinlichkeit der Einsetzung bei der Mutter klassifizieren. Ein wesentlicher Vorteil bei der Nutzung eines Bayes'schen Klassifizierers ist, dass die Struktur eine intuitive graphische Repräsentation ist, mit der man den Klassifikations-Prozess graphisch darstellen kann. Es gibt verschiedene Klassifizierer, nämlich den naiven Bayes, den selektiven naiven Bayes, den semi naiven Bayes, tree augmented naive Bayes (TAN) und den k-dependence Bayes (kDB) Algorithmus, die bei dem Versuch eingesetzt wurden. Bei dem tree augmented naive Bayes (TAN) handelt es sich um einen Bayes'schen Netzwerk-Klassifizierer, der Abhängigkeiten zwischen den Variablen zulässt. Die Einschränkung dieses Klassifizierers ist, dass maximal zwei Eltern erlaubt wurden, nämlich die Klassenvariable und eine weitere vorhergesagte Variable. Um so einen Klassifizierer aufzubauen, müssen erst einmal die Abhängigkeiten der einzelnen Variablen bestimmt werden. Danach wird eine Punktzahl anhand der zugrunde liegenden Theorie bestimmt. Für die Validierung der Algorithmen wurden die Informationen von 249 Embryos zu sechs unterschiedlichen Zeitpunkten verwendet. Von denen wurden 35 erfolgreich und 214 nicht erfolgreich implantiert. Mit Hilfe dieser Daten wurde eine 10-fold cross validation durchgeführt. Die Ergebnisse zeigen, dass der TAN Klassifizierer mit einem Wrapper, welcher eine weitere Punktzahl anhand der Güte der Klassifizierung vergibt, die höchste Genauigkeit, nämlich 91%, erzielt. Der semi naive Klassifizierer erzielt die geringste Genauigkeit, mit 78%, da dieser nur aus dem kartesischen Produkt einer Untermenge aller Variablen berechnet wird. Verglichen mit anderen Ansätzen, wie der logistischen Regression, die 82.33% erreicht, oder einer anderen Studie mit völlig verschiedener Methode, die 81.79% erzielt, ist der TAN Klassifizierer mit Wrapper, der 91% erreicht, sehr gut geeignet und selbst der semi naive Bayes'sche Klassifizierer mit 78% immer noch akzeptabel. Zusammenfassend ist zu sagen, dass ein Bayes'scher Klassifizierer für die Auswahl der Embryos mit der höchsten Wahrscheinlichkeit auf eine Schwangerschaft, gut geeignet ist.[MBL08]

Bei [TASM08] geht es um scaffold hopping, was darauf abzielt, chemische Stoffe zu finden die eine andere Struktur aufweisen als die ursprünglichen Substanzen, trotzdem aber die gleiche biologische

Aktivität aufweisen. Dabei startet die Suche mit high-throughput screening (HTS), das sich gegen ein biologisches Ziel, welches meist mit einer bestimmten Krankheit in Verbindung steht, richtet. Bei HTS werden eine große Menge an Substanzen getestet und diese, die sich als wirksam erweisen, werden in einem zweiten Schritt auf Sensitivität, Genauigkeit und andere Eigenschaften getestet. In dem Versuch wurde das ILP System CProgol in der Version 5 benutzt. Zum Vergleich wurden die zwei Algorithmen CATS (chemically advanced template search) und CATS3D verwendet. Beide Algorithmen können scaffold hopping nur mit Hilfe der positiven Beispiele durchführen und benötigen somit keine negativen Beispiele. Der CATS-Algorithmus berechnet Vektoren, die die Häufigkeit von kürzesten Pfaden der Verbindungen zwischen den Pharmakophoren beinhaltet. Die so entstandenen Vektoren werden mit Hilfe der Euklidischen Distanz verglichen. Der CATS3D-Algorithmus verwendet räumliche Distanz anstelle der Pfade, um die Vektoren zu erzeugen. Um einen Vergleich zwischen den einzelnen Vektoren durchzuführen, nutzt er die Manhattan Distanz, welche die Distanz als absolute Distanz der Einzel-Distanzen definiert. Zehn Datensätze der Resultate von HTS und die Strukturdaten aus der PubChem² Datenbank wurden als Testdaten verwendet. Aus jedem der Datensätze wurden 10% der aktiven scaffolds als Testdaten und die verbleibenden 90% wurden als Trainingsdaten verwendet. Bei drei der Datensätze wurden nur aktive scaffolds von dem ILP System gefunden nicht aber von CATS oder CATS3D. In einem der Datensätze fanden CATS und CATS3D aktive scaffolds und das ILP-System nicht und in einem Datensatz fand CATS mehr aktive scaffolds als das ILP System. In allen anderen fünf Datensätzen wurden mehr aktive scaffolds von dem ILP System gefunden. Zusammenfassend ist zu sagen, dass das ILP System signifikant besser als eine zufällige Auswahl von aktiven scaffolds arbeitet und die beiden anderen Algorithmen nicht. Außerdem fand die ILP Methode neue aktive scaffolds, die die anderen beiden nicht gefunden haben. Daraus folgt, dass der ILP basierte Ansatz mindestens genau so gut arbeitet wie die vorherigen Algorithmen und zusätzlich Menschen lesbare Regeln produziert, welche einen Einblick in scaffold hopping liefern. Somit liefert ILP Hilfe bei der Identifizierung neuer aktiver scaffolds.[TASM08]

[Leh10] beschreibt in dem Kapitel 7.2.2 einen Vergleich zwischen dem DL-Learner und dem ILP Werkzeug Aleph. Dabei handelte es sich bei dem Vergleich um das medizinische Lernproblem zum Erkennen von krebserregenden Substanzen. In Industrienationen kommen fast 100000 Chemikalien täglich in großen Mengen vor und 500 bis 1000 neue werden jedes Jahr hinzugefügt. Dabei ist oft unklar ob diese Substanzen krebserregend sein können, da die meisten daran gar nicht getestet werden. Um Substanzen auf ihre Krebserregbarkeit testen zu können wurden viele Ansätze ausprobiert, die sowohl von Menschen als auch von Maschinen durchgeführt wurden. Die Klassifikation solcher Substanzen durch Maschinen ist dabei relativ schwierig durch die hohe Anzahl und Unterschiedlichkeit der chemischen Elemente. Es gibt bereits einige Modelle zur Vorhersage von krebserregenden Substanzen die einige neue Eigenschaften für die Erkennung solcher Chemikalien liefern. Eine davon

²Quelle: <http://pubchem.ncbi.nlm.nih.gov/>

ist die Wissensbasis, die in diesem Vergleich verwendet wurde. Dabei wurden verschiedene chemische Substanzen, wovon einige krebserregend waren, in einer Wissensbasis dargestellt, die als Prolog Datei gespeichert wurde. Es wurden Informationen über Atome und Atomverbindungen gespeichert, sowie einige weitere spezifische Informationen über die Atome. Diese Daten wurden mit Hilfe eines Prolog Parsers und einer Klasse zum Mappen der Prologklauseln in OWL Ausdrücke, in eine OWL-Ontologie umgewandelt. In dem Versuch wurde der Lernalgorithmus OCEL des DL-Learner, welcher ein auf einem Refinement Operator basierender Lernalgorithmus ist, gegen verschiedene Algorithmen von Aleph, wie z.B. Aleph Randomized Rapid Restarts (Aleph RRR), Aleph Deterministic Top Down (Aleph DTD) und Aleph Ensembles verglichen. Dabei wurde auf die Genauigkeit und die Lesbarkeit der Ergebnisse geachtet. Die Ergebnisse dieses Lernvorgangs sahen wie folgt aus. Aleph Ensembles erreichte eine Genauigkeit von 59% bis 64,5%, Aleph DTD erreichte eine Genauigkeit von $57,9\% \pm 9,8\%$ und Aleph RRR erreichte eine Genauigkeit von $57,6\% \pm 6,4\%$. Demgegenüber erreichte der DL-Learner eine Genauigkeit von $67,7\% \pm 11,3\%$. Des Weiteren waren die Ergebnisse, die vom DL-Learner erzielt wurden viel kürzer und besser lesbar, als die, die Aleph produziert hat. Somit ist Abschließend zu sagen, dass der DL-Learner kürzere und besser lesbare Konzepte produziert, die zudem noch eine höhere Genauigkeit als die von Aleph besitzen.[Leh10]

3 Medizinische Grundlagen

In diesem Kapitel werden die nötigen medizinischen Grundbegriffe für das Verständnis der Arbeit geklärt. Dabei werden zuerst medizinisch relevante Fachausdrücke und Zusammenhänge geklärt.

3.1 idiopathische Parkinson

Bei dem idiopathischen Parkinson-Syndrom (IPS) handelt es sich um eine neurodegenerative Erkrankung, welche meist nach dem 50. Lebensjahr auftritt. Es ist das am häufigsten Auftretende Parkinson-Syndrom. Dabei verläuft es mit einem voranschreitenden Verlust der dopaminergen Neuronen in der Substantia nigra pars compacta, mit meist gleichzeitigen Auftreten von Lewy-Körperchen in den degenerierenden Neuronen. Diese Krankheit wird charakterisiert durch das Auftreten von Ruhetremor, Rigidität, posturale Instabilität, Bradykinese (langsame und träge Bewegung) und Akinese (Initiierung von Bewegung gestört), die zu Beginn der Krankheit meist asymmetrisch ausgeprägt ist. Bei einigen Patienten treten zusätzlich noch kognitive Störungen bis hin zur Demenz und Depression auf. Weitere Frühsymptome können Muskel- und Gelenkschmerzen, Missempfindungen, Verspannungen, Veränderung der Körperhaltung, Antriebsarmut, Schwierigkeiten Entscheidungen zu treffen, innere Unruhe, sexuelle Dysfunktion oder Schlafstörungen sein.[SMS⁺08]

Diagnose

Die Diagnose von IPS wird in 4 Schritte eingeteilt. Dabei wird im ersten Schritt die Diagnose hauptsächlich durch Anamnese und klinische Untersuchungen gestellt. Bei Vorhandensein von Akinese, eines der folgenden Symptome muskulärer Rigor, Ruhetremor oder posturale Instabilität und der Asymmetrie der motorischen Symptome, welche sich zu unterschiedlichen Zeitpunkten manifestieren können, liegt ein erhöhter Verdacht auf das idiopathische Parkinson-Syndrom vor. In dem zweiten Schritt sollte der Ausschluss einer symptomatischen (sekundäre) Parkinson-Krankheit untersucht werden. Dazu wird eine Behandlung mit Dopamin- oder Kalziumantagonisten oder Valproat durchgeführt. Des Weiteren muss eine Überprüfung auf zerebrale Raumforderungen oder Hydrozephalus, ein rezidivierendes Schädel-Hirn-Trauma in der Eigenanamnese, eine gesicherte Enzephalitis in der Eigenanamnese und seltene Intoxikationen durchgeführt werden. Wenn eine symptomatische Parkinson-Erkrankung ausgeschlossen wurde, muss in Schritt drei nach Hinweisen für ein atypisches Parkinson-Syndrom untersucht werden. Kriterien für den Hinweis auf eine atypische Parkinson-Erkrankung sind, das Nicht ansprechen auf hohe Dosen L-Dopa (über 1000mg/Tag), frühe schwere autonome Störungen, supranukleäre vertikale Blickparese, zerebelläre Zeichen, Okulogyre Krisen, frühe posturale Instabilität und Stürze, Pyramidenbahn Zeichen, frühe visuelle Halluzinationen ohne Bezug zu Medikamenten, zeitige Demenz vor Auftreten der motorischen Symptome, deutliche

Dysarthrie oder Dysphagie, ausgeprägter Antekollis, Apraxie und spontane Somnolenzphasen oder nach Neuroleptikagebrauch. Sollten keine atypische Parkinson-Krankheit vorliegen, muss im Schritt vier die Diagnose wiederholt kritisch geprüft werden, um die Diagnose auf idiopathische Parkinson - Krankheit zu sichern, vor allem bei Auftreten von Warnsymptomen einer atypischen Parkinson-Krankheit. Die unterstützenden Kriterien für die Diagnose des idiopathischen Parkinson-Syndroms sind der asymmetrische Beginn und eine persistierende Asymmetrie des Krankheitsverlauf, der Ruhetremor, das eindeutig positive Ansprechen auf L-Dopa und ein nicht durch Zusatzsymptome komplizierter klinischer Verlauf über mehr als 10 Jahre.[SMS⁺08]

Verlauf

Bei der idiopathischen Parkinson Erkrankung treten zuerst Symptome, wie die Veränderung der Befindlichkeit geistiger Flexibilität auf. Des Weiteren kann es zum Rückzug der betroffenen Person kommen oder zu einem zunehmend traurigen und starr wirkenden Gesichtsausdruck. Bei diesen Symptomen handelt es sich um sehr frühe Erscheinungen, die oft falsch interpretiert werden.[SR03] Auf diese Symptome folgt dann die typischerweise zu Beginn asymmetrische Ausprägung der Hauptsymptome, wie z.B. Akinese Rigor, Tremor und posturale Instabilität. Des Weiteren kommen im Verlauf der Krankheit motorische Symptome, wie Mikrographie, Hypomimie, Hypophonie, Dysarthrophonie, verminderte Prosodie und Dysphagie hinzu. Zu den nicht motorischen Anzeichen einer IPS gehören vegetative Störungen mit Obstipation, Harndrangbeschwerden, sexuelle Dysfunktion, orthostatische Hypotension, Schweißausbrüche und Seborrhöe, ebenso wie sensorische Symptome mit Parästhesien, Anosmie und Schmerzen. Häufig treten im Verlauf der Krankheit auch Schlafstörungen, Depressionen, Psychomotorische Verlangsamung und kognitive Defizite auf. Diese entwickeln sich im langjährigen Verlauf hin bis zu einer Demenz.[SMS⁺08]

Therapie

Bei der Therapie von IPS wird in medikamentöse und nicht medikamentöse Ansätze unterschieden. Die medikamentöse Behandlung sollte erst bei Beeinträchtigung des alltäglichen Lebens begonnen werden. Hierbei muss in zwei Gruppen eingeteilt werden. Die erste Gruppe sind Patienten die unter 70 Jahre alt sind und bisher nicht behandelt wurden. Hier sollte mit einer initialen Gabe von Dopamin-Rezeptor-Agonisten behandelt werden, welche mit zusätzlicher Gabe von Domperidon besser verträglich ist. Somit können auch höhere Dosen verabreicht werden. Diese Behandlung wird Vorgeschlagen, da jüngere Patienten die an IPS leiden, häufiger zu Dyskinesen und Wirkungsschwankungen neigen. Sollte aber die Antiparkinsonwirkung der Dopamin-Rezeptor-Agonisten nicht mehr ausreichend gewährleistet sein, muss zusätzlich L-Dopa verabreicht werden. Die zweite Gruppe sind Patienten über 70 Jahren. Diese Patienten haben ein geringeres Risiko zu Dyskinesien oder Wirkungsschwankungen,

so dass bei ihnen eine initiale Gabe von L-Dopa vorgeschlagen wird. Die Vorteile von L-Dopa sind, dass es besser verträglich und wirksamer ist als die Dopamin-Rezeptor-Agonisten. Des Weiteren sollten auch Patienten bei denen Zweifel an IPS besteht, also an einer atypischen Parkinsonerkrankung leiden, initial mit L-Dopa behandelt werden, ebenso wie multimorbide Patienten, die bereits zahlreiche Medikamente einnehmen. L-Dopa ist eine direkte Vorstufe von Dopamin und wird im Gehirn in Dopamin umgewandelt. Da sich auf diese Weise auch Dopamin in nicht dopaminergen Nervenenden bildet, führt dies meist zu nicht tolerablen peripheren Wirkungen, die durch die Gabe eines peripheren Hemmers verringert werden sollen. Nach ca. 3 bis 5 Jahren kommt es dann zu Wirkungsschwankungen, welche sich in vorhersagbare, also von der L-Dopa Einnahme abhängigen Schwankungen und nicht vorhersagbare Schwankungen, die nicht von der Einnahme von L-Dopa abhängigen Fluktuationen, einteilen lassen. Einige Thesen gehen davon aus, dass diese Fluktuationen auf die fortschreitende Degeneration der dopaminergen Nervenzellen und eine geringere Speicherkapazität für Dopamin zurückgeführt werden können. Da bei IPS auch nicht motorische Symptome, wie Schmerzen, Schlafstörungen, Psychosen usw. auftreten, müssen diese ebenfalls behandelt werden. Hierzu stehen eine Reihe von medikamentösen und nicht medikamentösen Behandlungen zu Verfügung. Zu den nicht medikamentösen Behandlungen zählen Physiotherapie, Ergotherapie und Logopädie. [Klo03]

3.2 Brustkrebs und Mammographie

In diesem Kapitel werden Ursachen und Behandlungsmöglichkeiten von Brustkrebs aufgezeigt und die Mammographie als Vorsorgeuntersuchung genauer erläutert.

3.2.1 Brustkrebs

Brustkrebs (Mammakarzinom) ist mit 23% aller Erkrankungen die häufigste Tumorart bei Frauen. Die Häufigkeit bei Brustkrebs hat in den letzten Jahren stetig zugenommen, wobei eine Ursache unklar ist. Dies kann jedoch hypothetisch mit dem frühen Beginn der Menarch und dem späten Einsetzen der Menopause in Verbindung gebracht werden. Weitere Ursachen könnten auch wenige und späte Schwangerschaften, veränderte Lebens- und Eßgewohnheiten, Strahlenexposition, eine Genetische Prädisposition oder Hormonsubstitution sein. Trotz zunehmender Häufigkeit bei Brustkrebs ist die Sterberate deutlich zurückgegangen, was sich durch das erhöhte Gesundheitsbewusstsein und die zunehmende Akzeptanz der Mammographie, ausdrückt. Beim Mann tritt Brustkrebs nur in sehr seltenen Fällen auf, wobei die Häufigkeit bei unter 1 pro 100.000 Männern liegt. Die Erkrankungswahrscheinlichkeit ist bei Frauen im Alter von 45 bis 65 am höchsten und beim Mann liegt diese mit 60 bis 70 Jahren deutlich höher.[PSS⁺06]

Das Entstehen von Brustkrebs kann durch eine gesunde Lebensweise und die gezielte Beeinflussung

des Vitamin - und Hormonhaushaltes verringert werden. Dabei zeigte sich, dass durch einen erhöhten Obst- und Gemüsekonsum und das Treiben von Sport, besonders bei jungen Frauen das Brustkrebsrisiko reduziert werden kann. Ebenso hilft die Verringerung des Alkoholkonsums. In Tierversuchen zeigte sich, dass durch die Gabe von Antiöstrogenen die Tumorentwicklung reduziert oder verhindert werden konnte. Dabei werden Medikamente wie Raloxifen, SERM oder Retinoide verwendet. Neben dem Versuch der Prävention von Brustkrebs gibt es heutzutage große Anstrengungen einer Früherkennung. Diese Früherkennung teilt sich auf in die Selbstuntersuchung und die Mammographie, welche im nächsten Kapitelabschnitt näher erläutert wird. Die Selbstuntersuchung sollte bei Frauen mit erhöhtem Brustkrebsrisiko in regelmäßigen Abständen, meist monatlich, durchgeführt werden. Begonnen werden sollte damit ab dem 20. Lebensjahr und kann nur effektiv durchgeführt werden, wenn eine eingehende Erklärung der Untersuchung durch den Gynäkologen oder Hausarzt realisiert wurde. Die zweite Möglichkeit der Früherkennung ist die Mammographie, welche die einzige echte Methode zur Früherkennung von Mammakarzinomen darstellt. Dabei wird in Deutschland zwischen dem 30. und 35. Lebensjahr eine Mammographie für Frauen mit erhöhtem Brustkrebsrisiko empfohlen. Bis zu dem 50. Lebensjahr sollten dann Kontrollmammographien in einem Abstand von 3 Jahren durchgeführt werden. Da das Brustkrebsrisiko zwischen 50 und 60 Jahren am höchsten ist sollte in dieser Zeit eine jährliche Mammographie durchgeführt werden und danach, weil das Brustkrebsrisiko wieder sinkt, nur noch alle 3 Jahre.[PSS+06]

Histologische Klassifikation von Mammakarzinomen

Man kann Brustkrebs in verschiedene Kategorien einteilen. Dazu zählen die nichtinvasiven Karzinome und die invasiven Karzinome. Zu den häufigsten Tumorformen gehören das infiltrierende duktales Karzinom, das Lobulär invasive Karzinom, das medulläre und das Kolloidkarzinom. Zu den seltenen Formen zählen das tubuläre, das adenozytische und das papilläre Karzinom. Eine besondere Form stellt das inflammatorische Mammakarzinom dar. Bei dieser Tumorart kommt es zu einer Verlegung der Lymphgefäße durch die Tumorzellen. Dadurch tritt ein Lymphstau und Hautödem auf. Dies zeigt sich meist durch das klinische Bild der Orangerhaut.[PSS+06]

Diagnose

Eine Diagnose auf Brustkrebs kann nur mit Hilfe einer histologischen Untersuchung belegt werden, denn reine physikalische Untersuchungsmethoden reichen nicht aus, um eine Chemo - oder Strahlentherapie anzuordnen. Bei Verdacht auf einen Tumor sollte vor der Operation durch Röntgen des Thorax, einer Skelettszintigraphie und einer Leberzonographie, bereits stattgefundenen Metastasierung ausgeschlossen oder erfasst werden. Durch eine Exzisionsbiopsie kann gewebeschonend eine

Biopsie vorgenommen werden. Sollte dies nicht möglich sein steht die normale Biopsie zur Verfügung. Bei Brustkrebs gibt es keine speziellen Laboruntersuchungen und auf die Bestimmung von Tumormarkern sollte verzichtet werden.

Zu den Standard Untersuchungsmethoden zählen die beidseitige Mammographie der Brust, eventuell auch mit Galaktographie, eine in zwei Ebenen aufgenommene Röntgenuntersuchung des Thorax, die Skelettszintigraphie, die Sonographie oder ein CT des Abdomen und eine Histopathologische Gewebeuntersuchung nach einer Probebiopsie oder einem Schnellschnitt. Fakultativ kann auch eine Kernspintomographie der Mammae durchgeführt werden, um einen besseren Nachweis von multiplen Lokalisationen des Krebsbefalls nachzuweisen.[PSS⁺06]

Verlauf

Die bei Brustkrebs auftretenden Erstsymptome sind knotige, indolente Verhärtungen der Brust. Diese werden von Patientinnen zu über 60% als Erstsymptom festgestellt. Schmerzen sind ein deutlich selteneres Erstsymptom und wird von weniger als 20% der Patientinnen als Erstsymptom beschrieben. Bildung von Orangenhaut, Einziehungen der Mamille oder Entzündungszeichen sind sehr selten als Erstsymptom und treten nur bei 10 bis 15% der Patientinnen auf. Bei schon sehr fortgeschrittener Krankheit kommt es oft zu Leistungsminderung, Müdigkeit, Gewichtsabnahme und Knochenschmerzen. Der weitere Krankheitsverlauf wird durch verschiedene Faktoren wie Tumorgröße, axillärer Lymphknotenstatus und Alter bestimmt. Adjuvante zytostatische und hormonelle Therapie kann die Zeit bis zum Wiederauftreten verlängern und die Chancen auf Heilung erhöhen. Diese steigt nach einer Beobachtungszeit von 10 Jahren auf 4-12%. Bei 70% der Patientinnen beschränkt sich die Nachsorgeuntersuchung auf Fernmetastasennachweis. Dabei ist bei 50% der Patientinnen das Skelettsystem, bei 15% die Lunge und bei 5% die Leber betroffen. Die Überlebenszeit bei Nachweis einer Metastasenbildung hängt von der Biologie des Tumors, der Ausbreitungsgeschwindigkeit und der Organinfiltration ab. Bei erfolgreicher Chemotherapie liegt die mittlere Remissionzeit bei 7 bis 12 Monaten. Die Ansprechdauer bei Hormontherapie ist deutlich länger. Die Überlebensdauer von mehr als 3 Jahren wird von ca. 40% der Patientinnen erreicht und die Überlebensdauer von über 5 Jahren wird von ca. 20% der Patientinnen erreicht.[PSS⁺06]

Therapie

Bei jedem Tumorverdacht muss eine Operation und eine histologische Beurteilung des Gewebes durchgeführt werden, um zu beurteilen, ob es sich um ein Mammakarzinom handelt. Diese Operation muss so stattfinden, dass nach Gewebeentnahme und Schnellschnittuntersuchung bei Bestätigung des Tumorverdachts sofort weiter operiert werden kann. Da sich Mammakarzinome zu 80-90% in

einem operablen Zustand befinden und nur 5-10% schon Fortgeschritten oder bereits Metastasiert sind, kann in den meisten Fällen sofort eine Operation durchgeführt werden. Bei den fortgeschrittenen Tumoren ist eine präoperative Chemotherapie nötig. Diese wird durchgeführt, um den fortgeschrittenen Tumor wieder in einen operativen Zustand zu überführen. Des Weiteren hat sich bei inflammatorischen Mammakarzinomen eine präoperative Chemotherapie bewährt. Diese wird heutzutage zunehmend auch bei kleineren, operablen Tumoren eingesetzt. Dies dient dem Ziel, den Anteil an Frauen bei denen eine brusterhaltende Operation durchgeführt werden kann, zu erhöhen. Für den Krankheitsverlauf, gilt das Ansprechen auf die präoperative Chemotherapie als eines der wichtigsten Prognosefaktoren für den weiteren Krankheitsverlauf. Bei operablen Tumoren wird meist eine Operation in Verbindung mit einer Strahlentherapie eingesetzt. Die postoperative Behandlung hängt von der Größe des Tumors und dem Auftreten von Metastasen ab. Bei Tumoren mit Metastasen wird meist eine Chemotherapie durchgeführt und bei kleineren Tumoren oder bei Brusterhalt wird eine Strahlentherapie durchgeführt. Bei Patientinnen bei denen keine brusterhaltende Bestrahlung durchgeführt werden kann, besteht die Möglichkeit der Rekonstruktion durch Implantation einer Prothese und Rekonstruktion der Brustwarzen und des Warzenhofes.[PSS⁺06]

3.2.2 Mammographie

Die Mammographie ist eine Röntgenuntersuchung der weiblichen Brust, um Brustkrebs diagnostizieren zu können. Diese Untersuchung ist eine der schwierigsten Untersuchung der Röntgendiagnostik, denn um kleinste Läsionen erkennen zu können, benötigt man eine optimale Aufnahme - und Einstellungstechnik.

Da die Erkennung von kleinsten Läsionen erwünscht ist, muss die Mammographieröhre einen sehr kleinen Brennfleck (<0,4mm) aufweisen und die Belichtungszeit muss kurz sein. Die Darstellung von Drüsengewebe/Bindegewebe und Fettgewebe erfolgt mit Hilfe von energiearmen Photonen (26-38keV) aus Molybdänanode und Molybdänfiltern. Bei strahlendichtem Gewebe wird eine Anoden-Filter-Kombination aus Wolfram/Molybdän oder Rhodium/Rhodium eingesetzt. Der Mammographiefilm besteht aus einer Verstärkerfolie mit Leuchtstoffbeschichtung, sowie einem einseitig beschichteten Spezialfilm. Dabei wird die Schärfe der Abbildung durch die Verstärkerfolie und der Kontrast durch die Spezialfolie gewährleistet. Die Verarbeitung des Spezialfilmes ist von Temperatur, Luftdruck, Entwicklungszeit und Zusammensetzung der Chemikalien abhängig und schlechte Bildqualität ist meist auf Fehler in der Filmverarbeitung zurückzuführen.

Die Standardaufnahmen bei einer Mammographie sind die kraniokaudale Aufnahmeprojektion und die Schrägaufnahme. Bei der kraniokaudalen Aufnahmeprojektion sollte zusätzlich zu dem axillären Drüsenanteil die gesamte Brust dargestellt werden. Außerdem sollte der M. pectoralis gerade noch abgebildet werden, ebenso wie der mediale Anteil der Brustumschlagsfalte. Bei der Schrägaufnah-

me muss die gesamte Brust erfasst werden, wobei in der kranialen Hälfte der Aufnahme der große Brustmuskel schräg und die Brustumschlagsfalte gerade noch dargestellt werden müssen. Bei beiden Aufnahmen wird die Mammille tangential erfasst. Wenn bei diesen Aufnahmen suspekte oder ungeklärte Befunde entstehen, werden weitere Zusatzaufnahmen nötig. Eine dieser Zusatzaufnahmen ist die laterale Aufnahme. Dabei wird der Strahlengang so gewählt, dass der abzuklärende Teil der Brust möglichst nahe am Film liegt. Diese Aufnahme wird bei Mikroverkalkungen zur Darstellung des gutartigen sogenannten Tea-cup-Zeichen, zur räumlichen Zuordnung von Herdbefunden oder zum Ausschluss von Überlagerungsartefakten angeordnet. Die zweite Zusatzaufnahme ist die Vergrößerungsaufnahme. Dabei wird der abzuklärende Bereich mit Hilfe des Mikrofokus um den Faktor 1,4-2,0 vergrößert. Dieser Mikrofokus hat eine Größe von 0,1-0,15 mm und dient der besseren Analyse von Mikroverkalkungen oder Herdbefunden. Außerdem sollte somit ein Ausschluss von Überlagerungsartefakten ermöglicht werden.[BW04]

Das American College of Radiology hat eine Einteilung der Befunde in das Breast Imaging Reporting and Data System (BIRADS) vorgeschlagen. In diesem System wird zwischen Herdbefunden, Verkalkungen, Architekturstörungen und speziellen Befunden unterschieden. Dabei werden die Befunde in 7 Kategorien eingeteilt, die im Folgenden kurz beschrieben werden.

- BIRADS Kategorie 0

Diese Kategorie bezeichnet eine nicht komplette Untersuchung. Es werden weitere Untersuchung benötigt, um eine endgültige Einschätzung zu ermöglichen.

- BIRADS Kategorie 1

Bei der Kategorie 1 handelt es sich um eine negative Mammographie ohne Herde, Verkalkungen oder Architekturstörungen. Es ist der Normalbefund.

- BIRADS Kategorie 2

Hierbei handelt es sich ebenfalls um einen Normalbefund, aber es gibt eindeutig benigne Befunde wie Fibroadenome, multiple sekretorische Verkalkungen, fetthaltige Läsionen oder Ölzysten. Des Weiteren können auch Herde mit gemischter Dichte, intramammäre Lymphknoten, Gefäßkalk oder Narben auftreten.

- BIRADS Kategorie 3

Bei dieser Kategorie liegt die Tumorstromwahrscheinlichkeit bei weniger als 2%. Dabei wird keine Befundsänderung erwartet, es sollte aber trotzdem in kurzen Abständen eine Kontrolle durchgeführt werden, um dies genau abzuklären. Vor der Einordnung in diese Kategorie muss eine vollständige Untersuchung der Patientin durchgeführt werden. In seltenen Fällen z.B. wenn es die Patientin wünscht, kann auch hier eine Biopsie durchgeführt werden.

- BIRADS Kategorie 4

In dieser Kategorie sind Befunde, die nicht den klassischen Tumorkriterien unterliegen, aber eine höhere Wahrscheinlichkeit auf Malignität aufweisen als Kategorie 3. Hier ist eine Biopsie empfehlenswert.

- BIRADS Kategorie 5

Hierbei handelt es sich um Befunde, die zu über 95% maligne sind.

- BIRADS Kategorie 6

Diese Kategorie ist für histologisch gesicherte Malignome reserviert. Dies wird beim Monitoring bei einer Chemotherapie verwendet.[BW04]

3.3 Elektrische Impedanz Tomographie

Bei der Elektrischen Impedanztomographie (EIT) handelt es sich um ein nichtinvasives, strahlungsfreies, bildgebendes Verfahren, bei dem die Impedanzverteilung in dem Querschnitt eines Körpers bestimmt wird. Da die Impedanz (Z) für den Wechselstrom, analog zum Ohmschen Gesetz für Gleichstrom definiert ist, lässt sie sich aus dem Quotienten der gemessenen Spannung (U) und dem eingeleiteten Strom (I) berechnen. Somit ergibt sich für die Impedanz: $Z=U/I$. Es werden eine Reihe von EKG-Elektroden auf der zu untersuchenden Körperoberfläche angebracht, meist 16 oder 32 Elektroden. Dabei werden sie zirkulär in Schichten angebracht und ein fortlaufender Wechselstrom mit konstanter Amplitude zwischen 2 benachbarten Elektroden aufgebaut. Dieser Wechselstrom rotiert kontinuierlich um die restlichen Elektroden. Die jeweils resultierenden Spannungen zwischen den verbleibenden Elektodenpaaren wird gemessen und wenn jede Elektrode als Impulsgeber genutzt wurde ist die Messreihe beendet und man erhält auf diesem Weg 208 verschiedene Messwerte für die Impedanzverteilung. Daraus wird eine Impedanzverteilung für eine Querschnittsebene des Körpers berechnet. Da aber jede Messung einen Referenzwert für die Normalisierung der Darstellung benötigt, wird vor der eigentlichen Messung eine Referenzmessung durchgeführt. Diese normalisierten Impedanzunterschiede werden anschließend mit Hilfe einer 32x32 Matrix in ein zweidimensionales Bild aufgebaut, wobei die Matrix in eine Grau - oder Farbskala umgewandelt wird. Da unterschiedliche Gewebearten eine spezifische Leitfähigkeit aufweisen und Körperflüssigkeiten eine höhere spezifische Leitfähigkeit aufweist als Körpergewebe, wird die EIT für Information über regionale Flüssigkeitsansammlungen oder die Durchblutung von Organen benutzt. Dabei wird sie besonders für die Diagnostik im Bereich der Erkrankung von Lunge und Herz verwendet. Bisher sind keine Nebenwirkung dieser Technik bekannt.

Da diese EIT-Bilder nur eine momentane Veränderung der Impedanz angibt, wurde die EIT zu der funktionellen EIT weiterentwickelt. Diese fasst etwa 240 einfache EIT-Bilder zusammen und be-

schreibt die lokalen Abweichungen der Impedanzverteilung. Diese Abweichung wird als Standardabweichung der Impedanzschwankungen festgelegt und für jede Pixelposition in der Matrix berechnet. Da Gewebe mit geringem Einfluss auf die Impedanz während der Beatmung existieren, wie z.B. Rippen oder Muskeln, haben diese eine geringe Abweichung der relativen Impedanz über die Zeit. Da die Lunge während des Atemvorgangs ständig Luft aufnimmt und abgibt, ändert sich die Impedanz des Lungengewebes ständig. Dabei steigt die Impedanz beim einatmen und sinkt beim ausatmen. Diese Änderungen der Impedanz über die Zeit können mit Hilfe der funktionellen EIT sichtbar gemacht werden und somit ist es möglich Veränderungen im Lungenvolumen zu erfassen.[KDM04]

3.4 Elektrokardiographie

Bei der Elektrokardiographie handelt es sich um ein medizinisches Verfahren zur Aufzeichnung und diagnostischen Auswertung von Elektrokardiogrammen. Dabei wird der elektrische Erregungsvorgang am Herzmuskel in Form von Wellen, Zacken und Strecken gemessen und aufgezeichnet. Normalerweise wird ein EKG durch nicht invasive Ableitung bestimmt. Diese Ableitung wird über die Spannungsdifferenz, des elektrischen Feldes auf der Körperoberfläche gemessen. Diese Spannung entspricht aber nur einem Bruchteil der Spannung, die vom Herzen entwickelt wird und somit kann das Herz als Potentialquelle angesehen werden. Die Ableitung der Spannung durch das EKG entspricht somit einer Ableitung der Erregungsvorgänge am Herzen.

Die Geräte zur Aufzeichnung von EKG's sind rauscharme Differentialverstärker. Diese bestehen aus einem stark gekoppelten Gleichstromverstärker mit invertierendem und nicht invertierendem Eingang und haben einen hohen Verstärkungsgrad. Dabei beträgt die Ausgangsspannung ein Vielfaches der Eingangsspannung, die an den Klemmen anliegt. Somit gelten für EKG Vorverstärker besonders hohe Anforderungen. Zu diesen Anforderungen zählen, dass einstreuende hochfrequente Wechselspannung über Hochfrequenzfilter gedämpft werden müssen, um eine Übersteuerung oder Selbsterregung zu verhindern. Zusätzlich werden extreme Störspannungen über Entladestrecken und antiparallele Dioden abgewehrt und entkoppelte netzfrequente Wechselspannung wird durch eine hohe Gleichtaktunterdrückung eliminiert. Das EKG Signal wird anschließend mit einer 20- bis 30-fachen Vorverstärkung verstärkt und mit Hilfe eines Hochpassfilters von der Gleichspannung getrennt. Dieser Filter besitzt eine Zeitkonstante von 1,5 oder 3,2 Sekunden. Der beim analogen EKG durchgeführte Prozess, dass das EKG Signal über ein Ableitungswähler den jeweiligen Kanalverstärkern (1...>12) zugeführt wird, entfällt bei digitalen EKG Geräten. Hier werden stattdessen die aufgenommenen Signale zu Ableitungen verknüpft und mit Hilfe von Multiplexer und AD-Wandlern an ein Prozessorsystem übergeben. Dieses Signal wird anschließend mit Hilfe von Steuersoftware verarbeitet und die einzelnen Messwerte werden interpoliert. Des Weiteren erfolgt eine Kommunikation mit dem Bediener über eine Tastatur und eine Anzeige und es gibt die Möglichkeit das EKG auszudrucken.

Für die Aufnahme von EKG-Signalen könnten sich die Ableitungspunkte an einer beliebigen Körperstelle befinden, aber es haben sich bestimmte Punkte entwickelt, welche zur Ableitung genutzt werden. Dabei haben sich drei Standardableitungen entwickelt, die bipolare Extremitätenableitung nach Einthoven, die unipolare Extremitätenableitung nach Goldberger, die sich aus der Extremitätenableitung nach Einthoven entwickelt hat und die unipolare Brustwandableitung nach Wilson. Bei der bipolaren Ableitung nach Einthoven wird die Potentialdifferenz zwischen zwei Elektroden gemessen. Dabei befindet sich eine Elektrode am rechten Arm, die zweite am linken Arm, die dritte am linken Fuß und die vierte befindet sich am rechten Fuß. Die Elektrode am rechten Fuß dient als Erde. Somit liegen die drei Ableitungen in der Frontalebene und bilden das sogenannte Einthoven-Dreieck. Bei der unipolaren Extremitätenableitung nach Goldberger werden die Potentialunterschiede jeweils an den Elektroden der Extremitäten und einem elektrischen Nullpunkt, der auf dem Mittelpotential der beiden anderen Extremitäten liegt, gemessen. Dabei werden die gleichen Messpunkte genommen, wie bei der Extremitätenableitung nach Einthoven. Die unipolare Brustwandableitung nach Wilson verwendet sechs Elektroden, welche an sechs vordefinierten Stellen am Thorax platziert werden. Des Weiteren werden 3 Elektroden an den Extremitäten über einen hochohmigen Widerstand zusammengeschlossen. Dieser Zusammenschluss wird als Wilson-Stern oder Central terminal bezeichnet und bildet den elektrischen Nullpunkt. Bei der Brustwandableitung befinden sich die Elektroden sehr nah am Herzen und somit ist die Amplitude größer als bei der Extremitätenableitung. [Kra07]

4 Semantik Web Grundlagen

In diesem Abschnitt werden die Idee und die Grundlagen des Semantik Web genauer erläutert.

Das Internet besteht schon heutzutage aus einer für den Menschen unüberschaubaren Menge an Information, welche unterschiedlich dargestellt werden und in unterschiedlichen Formaten gespeichert sind. In dieser Menge an Daten ist es selbst für Suchmaschinen unmöglich alle relevanten Informationen zu einem Themengebiet herauszufiltern, da die Darstellung meist für Menschen ausgelegt ist und eine Suchmaschine diese nicht interpretieren und weiterverarbeiten kann.

In Abbildung 1³ wird der Layer Cake des Semantik Web dargestellt, der folgende Standards verwendet. Als Grundlage für die Eindeutigkeit werden so genannte Uniform Resource Identifiers (URI's) verwendet, um Objekte darzustellen. Diese werden durch Angabe einer Adresse, z.B. *http://purl.org/dc/elements/1.1/title* dargestellt. Dabei ist es nicht wichtig, ob diese URI's auf eine Internetseite verweisen oder nicht, es wird aber als gute Praxis angesehen, wenn diese tatsächlich Informationen über die entsprechende Ressource enthalten. Weiterhin verwendet das Semantik Web XML als Basis, da es eine standardisierte Meta-Sprache ist, die bereits Maschinenlesbarkeit bereitstellt. Als Datenaustauschformat wird RDF verwendet, da es als Beschreibungssprache für das WWW oder andere elektronische Netze konzipiert wurde. Auf diesen Grundlagen bauen die Ontologiesprachen RDFS und OWL auf, die im nächsten Abschnitt genauer vorgestellt werden. Darauf wird eine vereinigende Logik definiert und mit Hilfe von Beweisen und Vertrauen und einem Crypto Service ein Sicherheitskonzept in das Semantik Web gebracht. Auf den gesamten Grundlagen werden anschließend Benutzerschnittstellen und Anwendung für das Semantik Web definiert.[HKRS08]

4.1 RDF

Resource Description Framework⁴ (RDF) ist eine formale Sprache zur Beschreibung strukturierter Informationen, durch die der Anwender in der Lage ist, Daten auszutauschen ohne dabei deren Bedeutung zu verlieren. Die Darstellung wird meist durch einen gerichteten Graph realisiert, kann aber auch mit Hilfe von XML durch Tripel dargestellt werden. Von einer hierarchischen Anordnung wird meist abgesehen, da diese nicht immer sinnvoll ist, z.B. wenn sich keiner der dargestellten Begriffe unterordnen lässt.[HKRS08] RDF Tripel bestehen aus drei Bestandteilen, wie in Abbildung 2 zu sehen ist.

1. Subjekt(Ressource):

³Quelle:<http://www.w3c.it/talks/2008/wsb08/images/layerCake.png>

⁴<http://www.w3.org/RDF/>

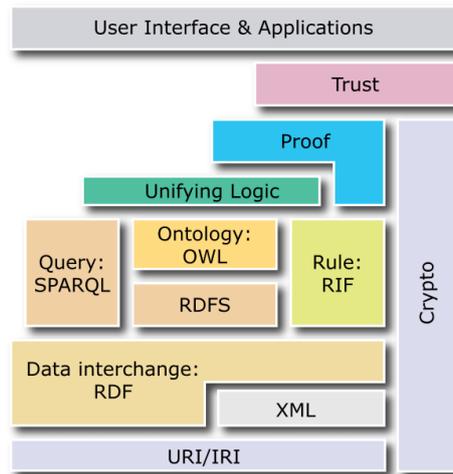


Abbildung 1: Darstellung des Layer Cakes des Semantik Web, das in Schichten aufgebaut ist und einheitliche Standards wie XML verwendet.

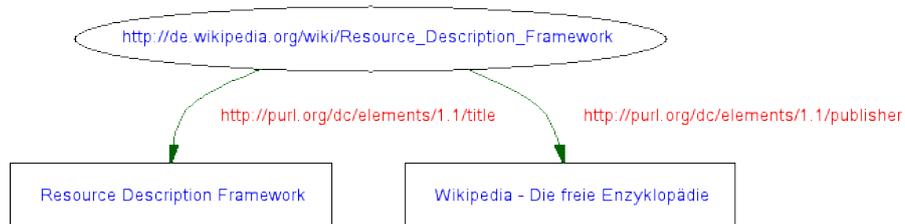


Abbildung 2: Einfacher RDF Graph. Dabei besteht dieser aus zwei Tripeln.

Ressourcen sind alle Dinge, die durch RDF Ausdrücke beschrieben werden können. Sie werden durch Ellipsen in der grafischen Darstellung modelliert.

2. **Prädikat(Eigenschaftselement):**

Das Eigenschaftselement gibt Auskunft über die ihm zugeordnete Ressource und stellt den Bezug zum Objekt her, verbindet also die Ressource mit dem Objekt. In der grafischen Modellierung wird das Eigenschaftselement durch eine benannte Kante dargestellt.

3. **Objekt(Ressource oder Literal):**

Das Objekt beschreibt den Wert eines Prädikates. Dabei kann das Objekt auch ein *Blank Node*, also ein leerer Knoten sein. In der grafischen Darstellung können Objekte entweder als Ellipsen dargestellt werden oder als Rechtecke. Die Darstellung als Ellipse wird bei Ressourcen verwendet und die Darstellung als Rechteck wird bei Literalen verwendet

Eine weitere Sprache zur Wissensrepräsentation ist OWL, welches im nächsten Abschnitt genauer erklärt wird.

4.2 Web Ontology Language OWL

Web Ontology Language⁵ (OWL) ist eine Wissensrepräsentationssprache, welche auf RDF aufbaut und komplexes Wissen effizient darstellen kann. Es basiert auf Beschreibungslogik und ermöglicht somit ein Gleichgewicht zwischen Ausdrucksstärke und effizientem Schlussfolgern. Da Ontologiesprachen, die Sprachkonstrukte mit hoher Komplexität zur Verfügung stellen, oft unentscheidbar sind, gibt es in OWL 3 verschiedene Versionen, wie OWL Lite, OWL DL und OWL Full, die den Anwender eine Wahl zwischen verschiedenen Ausdrucksstärken lasst.[HKRS08]

1. **OWL Lite**

OWL Lite ist eine Teilsprache von OWL DL und OWL Full, mit dem Ziel eine einfach zu implementierende Teilsprache bereitzustellen, die die wichtigsten Sprachelemente enthält. Diese Teilsprache heißt *SHIF*. Dadurch ist OWL Lite entscheidbar, aber weniger Ausdrucksstark.

2. **OWL DL** OWL DL(Description Logic) enthält alle Sprachkonstrukte von OWL Lite, ist aber nur eine Teilsprache von OWL Full. Es ist äquivalent zu einer entscheidbaren Untermenge der Prädikatenlogik erster Stufe und heißt *SHOIN(D)*.

3. **OWL Full**

⁵<http://www.w3.org/2004/OWL/>

OWL Full besteht aus den selben Sprachkonstrukten wie OWL DL, verzichtet aber auf die dort eingeführten Einschränkungen. Dadurch werden prädikatenlogische Ausdrücke eines höheren Grades erreicht, aber die Ontologien werden damit unentscheidbar.

Weiterhin werden Klassen, Eigenschaften und Individuen, die die Instanzen von Klassen darstellen als Grundbausteine von OWL Ontologien deklariert. Dabei gibt es in OWL zwei spezielle Klassen, die jede Ontologie enthält, nämlich *owl:thing* und *owl:nothing*. Die Klasse *owl:thing* ist dabei die allgemeinste Klasse, die alles enthält. Wohingegen *owl:nothing* im Allgemeinen leer ist.[HKRS08]

Die aktuelle Version OWL 2 unterteilt sich in zwei Versionen, nämlich OWL 2 DL und OWL 2 Full. Dabei entspricht OWL 2 DL der SROIQ(D) Logik und OWL 2 Full wurde eingeführt um eine Kompatibilität zu RDFS zu gewährleisten. Zusätzlich zu den zwei Versionen bietet OWL drei Profile an, EL, QL und RL. Dabei ermöglicht jede dieser Profile gewisse Einschränkungen, um effizienteres Reasoning zu gewährleisten. Das Profil OWL 2 EL ist für die Modellierung von ausdrucksstarken Properties zuständig und basiert auf den EL++ Logik. Daraus resultiert polynomielle Reasoningzeit bezüglich der Ontologiegröße für alle Standard Inferenzprobleme. OWL 2 QL ist das Profil, welches für Anwendungen mit sehr vielen Instanzdaten vorgesehen ist. Wie auch in OWL 2 EL sind alle Standard Inferenzprobleme in polinomeller Zeit lösbar. Das dritte Profil heißt OWL 2 RL und ist für skalierbare Anwendungen geschaffen wurden, ohne viel der Ausdrucksstärke zu verlieren. Die Algorithmen für das Reasoning können in regelbasierten Engines implementiert werden und laufen dann ebenfalls in polynomieller Zeit. Zusammenfassen zu den OWL 2 Profilen ist zu sagen, dass OWL 2 EL und OWL 2 QL eine Teilmenge von OWL 2 DL bilden und OWL 2 RL sowohl eine Teilmenge von OWL 2 Full als auch OWL 2 DL ist. Im Folgenden werden nun ein paar Sprachkonstrukte vorgestellt, die Beziehungen von Klassen und Instanzen modellieren.[Leh10]

Sprachkonstrukte in OWL

In OWL gibt es verschiedene Sprachkonstrukte, mit deren Hilfe man die Abhängigkeiten zwischen Klassen und Individuen darstellen kann. Im folgenden werden die wichtigsten Sprachkonstrukte in OWL mit entsprechender DL und Manchester OWL Syntax vorgestellt.

OWL Sprachkonstrukt	DL-Syntax	Manchester Syntax
Thing	\top	Thing
Nothing	\perp	Nothing
subClassOf	$C_1 \sqsubseteq C_2$	C_1 SubClassOf: C_2
equivalentClass	$C_1 \equiv C_2$	C_1 EquivalentTo: C_2
disjointWith	$C_1 \sqcap \neg C_2$	C_1 DisjointWith: C_2
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	C_1 and ... and C_n
unionOf	$C_1 \sqcup \dots \sqcup C_n$	C_1 or ... or C_n
complementOf	$\neg C$	not C
allValuesFrom	$\forall r. C$	r only C
someValuesFrom	$\exists r. C$	r some C
maxCardinality	$\leq n r$	r max n
minCardinality	$\geq n r$	r min n

Tabelle 1: In dieser Tabelle findet man einen Auszug aus den wichtigsten OWL Sprachkonstrukten mit entsprechender DL und Manchester OWL Syntax.[Leh10]

5 Machine Learning Grundlagen

In diesem Kapitel werden die verwendeten Algorithmen und Werkzeuge näher vorgestellt. Dabei werden zuerst die Algorithmen vorgestellt und anschließend die Werkzeuge.

5.1 Die Algorithmen

In diesem Abschnitt werden der t-Test, die Algorithmen von Aleph, Naive Bayes, Naive Bayes Updateable, Random Tree, Random Forrest, PART, Decision Table, OCEL und CELOE genauer vorgestellt.

5.1.1 Der t-Test

Im folgenden Abschnitt wird der t-Test erklärt, welcher für die Errechnung der p-Werte und die Aussage ob Unterschiede statistisch signifikant sind verwendet wurde.

Wenn man Daten von zwei Gruppen von Experimenten gesammelt hat, die durch einen unterschiedlichen Parameter entstanden sind, will man wissen ob der Unterschied der Ergebnisse von diesem Parameter abhängt oder ob dieser Unterschied zufällig entsteht. Dazu wird der p-Wert errechnet, der eine Wahrscheinlichkeit zwischen Null und eins angibt. Wenn der p-Wert Null ist bedeutet das, dass es eine Null prozentige Chance gibt einen Unterschied noch einmal zu beobachten, der genau so groß ist wie der, der bei dem Experiment beobachtet wurde. Ein p-Wert von Eins hingegen bedeutet, dass es eine 100% Chance gibt einen Unterschied noch einmal zu beobachten, der genau so groß ist wie der, der bei dem Experiment beobachtet wurde. Das soll soviel heißen wie, wenn der p-Wert gering ist, ist es unwahrscheinlich, dass die beobachteten Unterschiede durch Zufall entstanden sind. Da es aber keine Aussage darüber gibt, was ein wirklicher Unterschied ist, wird die Unsicherheit als Konfidenzintervall eingeführt. Dabei wurde für die Arbeit ein Konfidenzintervall von 95% angenommen und das bedeutet, man kann sich zu 95% sicher sein, dass die Unterschiede durch den veränderten Parameter entstanden sind und nicht zufällig. Daraus ergibt sich, wenn ein p-Wert kleiner als 0,05 ist, wird der Unterschied in den Daten als statistisch signifikant angesehen, da es eine fünf prozentige Chance gibt einen Unterschied noch einmal zu beobachten, der genau so groß ist wie der, der bei dem Experiment beobachtet wurde.[Mot99]

5.1.2 Der Algorithmus von Aleph

Aleph nutzt bei der Suche nach neuen Konzepten ein sehr simples Verfahren welches in vier Schritten erklärt werden kann.

1. Auswahl der Beispiele

In diesem Schritt wählt Aleph die Beispiele aus die verallgemeinert werden sollen. Sollten keine gefunden werden ist der Algorithmus beendet, ansonsten wird mit Schritt zwei fortgefahren.

2. Erzeugen der spezifischsten Regel

Bei diesem Schritt konstruiert Aleph die Regel die am spezifischsten ist, also die meisten Beispiele abdecken, welche im vorhergehenden Schritt ausgewählt wurden. Dies ist meistens eine eindeutige Regel mit vielen Literalen und wird bottom clause genannt. Manchmal wird dieser Schritt auch saturation genannt.

3. Suchen

Nach der Auswahl der spezifischsten Regel wird versucht eine allgemeinere Regel zu finden. Dies wird erreicht, indem man nur eine Teilmenge der Literale der bottom clause durchsucht. Hierbei werden die Literale mit der höchsten Wertung genutzt. Dabei müssen zwei Dinge angemerkt werden. Erstens, die Beschränkung der Suche auf Teilmengen der bottom clause erzeugt nicht alle Regeln, die allgemeiner als die bottom clause sind, gibt aber eine gute Auswahl. Zweitens, ist die Art der Bewertung einer Regel nicht wichtig. Manchmal wird dieser Schritt reduction Schritt genannt.

4. Redundanz entfernen

Bei diesem Schritt wird die Regel mit der besten Bewertung zur aktuellen Theorie hinzugefügt. Des Weiteren werden alle Beispiele als Redundant gekennzeichnet und entfernt. Dabei kann es passieren, dass die beste Regel, andere Regeln neben den Beispiele redundant machen kann. Dieser Schritt wird cover removal genannt. Nach diesem Schritt wird wieder bei Schritt 1 angefangen.

5.1.3 Die Algorithmen von Weka

Im folgenden Abschnitt werden die Algorithmen von Weka genauer vorgestellt.

Naive Bayes und Naive Bayes Updateable

Bei dem Naive Bayes Klassifizierer handelt es sich um einen simplen Klassifikationsansatz. Der Klassifizierer heißt naive, weil Vereinfachungen angenommen wurden. Die vorhersagenden Attribute sind alle Unabhängig, so dass es keine verborgenen Eigenschaften gibt, die den Vorhersageprozess beeinflussen können. Im folgenden ist C eine zufällig gewählte Variable, die die Klasse einer Instanz

bezeichnet und X einen Vektor mit zufälligen Variablen, die die beobachteten Attribute beschreiben. Des Weiteren repräsentiert c eine bestimmte Klasse und x ein bestimmten Attributwert aus dem Attributvektor. Um die Wahrscheinlichkeit einer Klasse zu errechnen, muss man die Bayes Regel anwenden.

$$p(C = c|X = x) = \frac{p(C=c)p(X=x|C=c)}{p(X=x)}$$

Dabei repräsentiert $X=x$ $X_1 = x_1 \wedge X_2 = x_2 \wedge \dots \wedge X_k = x_k$. Wegen der einfachen Verbindung der Attribute und der Annahme das alle Attribute Unabhängig sind ergibt sich daraus:

$$p(X = x|C = c) = p(\bigwedge X_i = x_i|C = c)$$

$$p(X = x|C = c) = p \prod (X_i = x_i|C = c)$$

Der Naive Bayes Klassifizierer behandelt diskrete und numerische Attribute anders. Denn für diskrete Attribute, wird $p(X = x|C = c)$ als eine einfache reelle Zahl zwischen 0 und 1 behandelt, welches der Wahrscheinlichkeit entspricht, dass das Attribut X den entsprechenden Wert x annimmt, wenn die Klasse c ist. Numerische Attribute hingegen werden als fortlaufende Wahrscheinlichkeitsverteilung über dem Bereich des Attributwerts dargestellt. Eine weiter allgemeine Annahme, die nicht nur für den Bayesansatz gilt, ist die Annahme, dass alle Werte der numerischen Attribute Normalverteilt sind. man kann die Normalverteilung einfach berechnen und für fortlaufende Attribute lautet die Formel dafür:

$$p(X = x|C = c) = g(x; \mu_c, \sigma_c)$$

$$g(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Dies ist die Formel für die Normalverteilung. Aus diesen Berechnungen ergibt sich der Klassifizierer für den Naive Bayes Algorithmus.[JL95]

Der Naive Bayes Updateable Algorithmus unterscheidet sich von dem Naive Bayes Algorithmus dadurch, dass für die Klassifikation ein inkrementeller Naive Bayes Klassifizierer verwendet wird, welcher eine Instanz zu jeder Zeit lernt.[WF05]

Random Tree

Bei einem Random Tree wird der Baum aufgebaut, indem bei jedem Knoten eine bestimmte Anzahl an zufällig ausgewählten Attribute selektiert werden. Diese werden darauf überprüft, ob sie für eine Beschreibung einer Klasse hilfreich sind. Nur die hilfreichen Attribute werden zur Klassenbeschreibung genutzt. Dabei wird keine Beschneidung des Baumes vorgenommen.[WF05]

Random Forrest

Bei dem Algorithmus Random Forrest werden wird eine Sammlung von Baumstruktur Klassifizierern $h(x, \Theta_k)$, $k = 1, \dots$, wobei Θ_k unabhängige identisch verteilte Zufallsvektoren sind und jeder Baum eine Einheitsstimme für die am besten passendste Klasse der Eingabe x abgibt. Das bedeutet, dass für den k ten Baum eine Zufallsvektor Θ_k erzeugt wird, welcher unabhängig von den vorhergehenden Zufallsvektoren $\Theta_1 \dots \Theta_{k-1}$. Nur die Verteilung bleibt die gleiche und der Baum wächst durch die Trainingsdaten und Θ_k . Dies Resultiert dann in dem Klassifizierer $h(x, \Theta_k)$ mit x als Eingabe. Für den Algorithmus wurde bagging in Kombination mit random feature selection gewählt. Dabei wird für jede neue gezogene Trainingsmenge, diese mit der originalen Trainingsmenge ersetzt. Der Baum wächst dann mit Hilfe der Trainingsdaten, auf die random feature selection angewendet wird. Der Bäume die so gewachsen sind werden nicht beschnitten. Anschließend wird für jede Trainingsmenge T_k eine Klassifizierer $h(x, T_k)$ konstruiert und dieser stimmt für eine Klasse die am besten passt mit Hilfe des bagged predictors. Dazu wird für jedes y, x der Trainingsmenge, nur für die Klassifizierer von T_k welche y, x nicht enthalten, die Stimmen zusammengezählt. [Bre01]

PART

Bei PART handelt es sich um einen Algorithmus, der eine Kombination der Algorithmen *C4.5* und *RIPPER* ist, der deren Nachteile vermeidet. Zum einen benötigt dieser Algorithmus keine globale Optimierung, um exakte Regeln zu erzeugen. Aus diesem Grund ist Einfachheit ein Vorteil dieses Algorithmus. Um Regeln zu erzeugen nutzt PART die sperate-and-conquer Strategie. Diese erzeugt eine Regel und anschließend werden alle Instanzen die von dieser Regel abgedeckt werden entfernt. Jetzt wird dieser Vorgang so lange rekursiv durchgeführt, bis keine Instanzen zum Bilden von regeln mehr zur Verfügung stehen. Dabei unterscheidet sich der sperate-and-conquer Ansatz von dem Standardansatz in der Form, dass für das Bilden der Regeln ein beschnittener Entscheidungsbaum genutzt wird, welcher ein gerichteter Baum ist, der zur Darstellung von Entscheidungsregeln dient. Das Blatt, welches die meisten Instanzen abdeckt wird zu einer Regel gemacht. Um dies zu realisieren, wird eine Teil Entscheidungsbaum erzeugt. Dabei handelt es sich um einen Entscheidungsbaum der Äste zu undefinierten Unterbäumen besitzt. Im Folgenden wird der Algorithmus zum Bauen des Baumes kurz beschrieben.[FW98]

Zuerst werden Mengen der Beispiele rekursiv in einen Teilbaum aufgeteilt. Dazu wird im ersten Schritt ein Test ausgewählt, der die Beispiele in Untermengen aufteilt. Die Auswahl des Tests geschieht dabei auf die selbe weise wie bei *C4.5*. Anschließend werden die Untermengen, mit Hilfe ihres mittleren Informationsgehalt erweitert, beginnend bei der Menge mit dem geringsten mittleren Informationsgehalt. Dies geschieht, weil Untermengen mit geringen mittleren Informationsgehalt auch kleine Unterbäume besitzen und somit allgemeinere Regeln produzieren. Dies geschieht

so lange, bis eine Untermenge zu einem Blatt erweitert wird. Wenn dies geschehen ist, beginnt das beschneiden des Baumes. Dazu überprüft der Algorithmus, ob der Knoten durch ein einzelnes Blatt ersetzt werden kann. Wenn eine Ersetzung vorgenommen wurde beginnt der Algorithmus mit dem Backtracking-Vorgang. Dazu erforscht er Geschwisterknoten, des neu ersetzten Knotens. Wenn aber bei dem Backtracking-Vorgang ein Knoten entdeckt wurde, bei dem alle Kindknoten keine Blätter sind, und dies geschieht wenn eine Unterbaumersetzung nicht durchgeführt wurde, werden die entsprechenden Untermengen nicht mehr erforscht und der entsprechende Unterbaum wird als undefiniert markiert. Durch die rekursive Struktur des Algorithmus, führt dies zur Terminierung des Algorithmus.[FW98]

Decision Table

Baut einen einfachen Decision Table Mehrheitsklassifizierer auf. Dabei besteht es aus zwei Komponenten, einem Schema, wo alle Attribute der Tabelle enthalten sind und einem Körper, der alle beschrifteten Instanzen enthält, die mit Hilfe des Schemas definiert wurden. Ein Decision Table Klassifizierer sucht nach Instanzen in der Decision Table, die eine exakten Übereinstimmung mit der unbeschrifteten Instanz besitzt und nutzt dabei nur die Attribute, die im Schema definiert wurden. Sollte keine Instanz gefunden werden, wird die Klasse mit den meisten Instanzen zurückgegeben. Sollte eine oder mehrere Instanzen gefunden werden, wird die Klasse zurückgegeben, die die meisten übereinstimmenden Instanzen besitzt.[Koh95]

5.1.4 Der Refinement Operator

In diesem Abschnitt wird der Refinement Operator kurz vorgestellt.

Definition Refinement Operator: Eine Halbordnung ist eine reflexive und transitive Relation. In einer Halbordnung (S, \leq) , ist ein downward (upward) Refinement Operator p , eine Abbildung von S nach 2^S , so dass für alle $C \in S$ ein $C' \in p(C)$ existiert, so dass gilt $C' \leq C$ ($C \leq C'$). C' wird dabei Spezialisierung (Verallgemeinerung) von C genannt.[LH10]

Diese Idee kann man nutzen, um in einem Konzeptraum zu Suchen. Als Ordnung kann man die Subsumption nutzen. (Die Subsumption ist eine Relation \sqsubseteq der Halbordnung.) Wenn ein Konzept C ein Konzept D zusammenfasst, ($D \sqsubseteq C$), dann werden alle Beispiele die von D abgedeckt werden auch von C abgedeckt. Im folgenden werden wir einen refinement operator für die Beschreibungssprache \mathcal{L} anschauen. Dieser wird \mathcal{L} refinement operator genannt.[LH10]

Definition \mathcal{L} refinement operator Sei \mathcal{L} eine Beschreibungssprache. Ein refinement operator in einem halbgeordneter Raum $(\mathcal{L}, \sqsubseteq)$ wird \mathcal{L} refinement operator genannt.[LH10]

Definition Refinement Kette Eine refinement Kette eines \mathcal{L} refinement operator ρ mit der Länge n von einem Konzept c zu einem Konzept D ist eine endliche Sequenz C_0, C_1, \dots, C_n von Konzepten, so das $C = C_0, C_1 \in \rho(C_0), C_2 \in \rho(C_1), \dots, C_n \in \rho(C_{n-1}), D = C_n$. Die refinement Kette geht durch E wenn es ein i ($1 \leq i \leq n$) so dass $E = C_i$ ist. Wir sagen dann das D von C durch ρ erreicht werden kann, wenn es eine refinement Kette von C nach D gibt. $\rho^*(C)$ bezeichnet die Menge aller Konzepte, welche von C durch ρ erreicht werden können. $\rho^m(C)$ bezeichnet die Menge aller Konzepte, die von C durch ρ mit einer refinement Kette der Länge m erreicht werden können.[LH10]

Definition downward und upward Abdeckung Ein Konzept C ist downward abgedeckt durch D wenn $C \sqsubseteq D$ und es kein Konzept E mit $C \sqsubseteq E \sqsubseteq D$ gibt. Ein Konzept C ist upward abgedeckt durch D wenn $D \sqsubseteq C$ und es kein Konzept E mit $D \sqsubseteq E \sqsubseteq C$ gibt.[LH10]

Im folgenden wir die Annahme der schwachen Gleichheit verwendet. Diese besagt, dass $A1$ und $A2$ ungleich $A2$ und $A1$ ist, aber das $A1$ und $A2$ schwach gleich $A2$ und $A1$ ist. Das bedeutet, dass die Ordnung der Elemente in den Konjunktionen und Disjunktionen nicht wichtig ist.[LH10] Im Anschluss werden die Eigenschaften des Refinement Operators vorgestellt, welche zum Lernen von Konzepten nützlich sein können.

Ein \mathcal{L} refinement operator ρ wird als endlich bezeichnet, wenn $\rho(C)$ für alle Konzepte C endlich ist. Des weiteren wird ein \mathcal{L} refinement operator als Redundant angesehen, wenn es eine refinement Kette von C nach D gibt, die nicht durch das Konzept E geht und eine refinement Kette von C zu einem schwach gleichen Konzept D welche durch E geht. Um die korrektheit eines solchen Operators auszudrücken, muss folgendes gelten. Für alle Konzepte C und D impliziert $D \in \rho(C)$ das $C \neq D$. Die letzte Eigenschaft die ein \mathcal{L} refinement operator besitzen muss, ist dass er ideal ist. Das bedeutet, er muss endlich vollständig (wird im nächsten Abschnitt erklärt) und korrekt sein.[LH10]

Im folgenden werden die Eigenschaften eines \mathcal{L} downward refinement operators ρ erklärt. Dieser wird als vollständig bezeichnet, wenn für alle Konzepte C, D mit $C \sqsubseteq D$, ein Konzept E , mit $E \equiv C$, von D durch ρ erreicht werden kann. Des Weiteren wird er als schwach vollständig angesehen, wenn für alle Konzepte $C \sqsubseteq \top$ ein Konzept E , mit $E \equiv C$, von \top durch ρ erreicht werden kann. Die letzte Eigenschaft, die ein \mathcal{L} downward refinement operators ρ haben muss ist die Minimalität. Dies wird erreicht, wenn alle C , $\rho(C)$ nur downward Abdeckung besitzen und alle seine Elemente durch \sqsubseteq unvergleichbar sind.[LH10]

OWL Class Expression Learner

Im folgenden wird der Lernalgorithmus OWL Class Expression Learner (OCEL) näher erläutert. Der Lernalgorithmus OCEL baut mit Hilfe eines Refinement Operators, einen Suchbaum auf, der aller nicht redundanten Konzepte enthält. Der Suchbaum dient dazu Konzepte in einer geordneten Form

abzuspeichern, um die Suche effizienter zu gestalten. Da der Refinement Operator wie in Kapitel 5.1.4 beschrieben, dazu dient eine Ordnung herzustellen, ist er gut geeignet um den Suchbaum aufzubauen. Dazu verwendet er die Annahme der schwachen Gleichheit. Diese besagt, dass $A1$ und $A2$ ungleich $A2$ und $A1$ ist, aber das $A1$ und $A2$ schwach gleich $A2$ und $A1$ ist. Dies wird gemacht, damit Konzepte die zwar syntaktisch nicht gleich sind, trotzdem als redundant angesehen werden, weil sie das gleiche meinen. Da dieser Refinement Operator unendlich lange Konzept zulässt, muss dieses Problem mit Hilfe der Konzeptlänge n gelöst werden. Dabei wird zu Beginn die maximale Länge auf n festgesetzt, wobei der Algorithmus in der Lage ist die maximale Länge dynamisch während der Laufzeit anzupassen um so immer längere Konzepte zulassen zu können. Des Weiteren wird jedes gefundene Konzept in dem Suchbaum in der Form (C,n,b) gespeichert, wobei C das Konzept ist n die horizontale Expansion des Konzeptes und b ein Boolean ob das Konzept zu Schwach ist oder nicht. Um jetzt festzulegen welches gefundene Konzept zu Schwach ist, wird eine Heuristik zur Bewertung benötigt. Diese Heuristik bedient sich dem Noise Parameter, welcher als $(1 - \text{Noise})$ definiert ist. Die Konzepte die einen geringeren Wert als $(1 - \text{Noise})$ erhalten werden ignoriert. Die Bewertung der einzelnen Konzepte erfolgt dabei durch die Formel

$$\text{accuracy}(C) = \frac{(up+cn)}{|E|}.$$

up = abgedeckten positiven Beispiele

cn = abgedeckten negativen Beispiele

E = alle Beispiele

Wenn jetzt die Genauigkeit des gefundenen Konzeptes größer ist als $(1 - \text{noise})$ wird dieser Knoten erweitert. In dem nächsten Schritt wird für das erweiterte Konzept der Genauigkeitsgewinn berechnet. Dies berechnet sich aus der Genauigkeit des gefundenen Konzeptes minus der Genauigkeit des alten Konzeptes, also $\text{acc_gain}(N) = \text{accuracy}(C) - \text{accuracy}(C')$. Aus diesen Werten wird nun die Punktzahl des neuen Konzeptes berechnet. Dies geschieht durch $\text{score}(N) = \text{accuracy}(C) + \alpha \cdot \text{acc_gain}(N) - \beta \cdot n$ ($\alpha \geq 0, \beta \geq 0$). Für diesen Algorithmus wurden $\alpha = 0.5$ und $\beta = 0.02$ gewählt. Somit ist abschließend zu sagen, dass die Heuristik als Hauptkriterium den Genauigkeitszuwachs nimmt. Dies wird mit α versehen, denn Konzepte die eine Erhöhung der Genauigkeit haben, eine höhere Wahrscheinlichkeit haben, eine bessere Lösung zu sein. Der dritte Parameter n der von β bestimmt wird, soll das Suchkriterium auf kürzere Konzepte und weniger untersuchte Suchgebiete lenken. Die horizontale Expansion des Konzeptes wird anstatt der Länge des Konzeptes verwendet, um zu verhindern, dass der Lernalgorithmus bei Konzepten mit hoher Genauigkeit und Genauigkeitszuwachs hängen bleibt. Die Scorefunktion ist dabei unabhängig von dem Lernalgorithmus implementiert und kann somit leicht ausgetauscht werden, wenn bei einem Problem eine Art Fehler schwerwiegender ist als ein anderer.[Leh10]

Class Expression Learner for Ontology Engineering

In diesem Abschnitt wird der Lernalgorithmus Class Expression Learner for Ontology Engineering (CELOE) genauer vorgestellt. Dieser Algorithmus ist speziell für das Ontology Engineering entwickelt worden. Dabei baut CELOE auf dem OCEL Algorithmus auf und ein Unterschied zwischen beiden Algorithmen ist, dass eine andere Heuristik verwendet wird. Die Heuristik bestraft dabei längere Konzepte stärker als bei OCEL. Dies liegt daran, dass im Gegensatz zu OCEL, wo der β -Parameter auf 0,02 gesetzt wird, der β -Parameter bei CELOE auf 0,1 gesetzt wird. Weitere Unterschiede zu OCEL sind, CELOE ist Speichereffizienter als OCEL und nur geringfügig langsamer, da es die nicht abgedeckten positiven und negativen Beispiele nicht im Suchbaum abspeichert und CELOE kann neben der Vorhersagewahrscheinlichkeit auf F-Measure und andere Bewertungsmethoden optimieren. Um die vorgeschlagenen Konzepte weiter zu vereinfachen wurde der EL Tree Learner (ELTL) Vereinfacher verwendet. Dabei ist der ELTL Algorithmus eine vereinfachte Form des OCEL Algorithmus. Dies kann dadurch erzielt werden, da der ELTL Algorithmus den EL refinement operator benutzt, welcher ideal ist. Das bedeutet, dass der \mathcal{EL} refinement operator unendlich ist und keine Redundanz aufweist. Dadurch können einige Teile von OCEL, wie die Überprüfung auf Redundanz und die Beschränkung der Länge entfernt werden. Damit wird der Algorithmus einfacher. Der Vereinfacher schreibt in einem ersten Schritt die Konzepte um (hier noch das in Klammern einfügen). Wenn dieser Schritt abgeschlossen ist, wird ein Reasoner benutzt um die Konzepte weiter zu vereinfachen. Wenn ein Konzept $D_1 \sqcap \dots \sqcap D_n$ existiert und D_1 ist eine Superklasse von D_i ($2 \leq i \leq n$), wird das entsprechende D_i entfernt. Anschließend wird mit D_2, D_3, \dots fortgefahren. Disjunktionen werden analog behandelt. Dadurch wird das Konzept in der Weise vereinfacht, dass man keine Elemente eine Konjunktion oder Disjunktion entfernen kann ohne die Äquivalenz zu behalten. An einem Beispiel dargestellt, vereinfacht der Vereinfacher das Konzept $\exists hasLeader.T \sqcap Capital$ zu $Capital$, wenn man mit Hilfe der Wissensbasis schließen kann, dass $Capital$ eine Stadt ist und dass jede Stadt einen Leader hat. Zu diesem Vereinfachungsprozess wird ein Reasoner Cache verwendet, um die vereinfachten Konzepte zu speichern. Zusätzlich wird sichergestellt, dass keine redundanten Vorschläge existieren. Dazu werden Konzepte die länger sind als andere und die selben relevanten Individuen gleich klassifizieren, gefiltert. Dadurch werden Ausdrücke verhindert, die irrelevante Subausdrücke enthalten. Im Gegensatz zu dem OCEL Algorithmus, ist bei CELOE das Standardabbruchkriterium die Laufzeit. Dadurch wird ein besser vorhersagbares Verhalten gewährleistet. Es ist aber ebenfalls möglich den Noise Parameter als Abbruchkriterium zu wählen. Da keine Einstellungsmöglichkeit bei CELOE Pflicht sind, kann sofort mit dem Lernvorgang angefangen werden ohne genaue Details über den Algorithmus zu wissen.[Leh10]

5.2 Die Werkzeuge

In diesem Abschnitt werden die Werkzeuge, welche zum Lernen von neuen Konzepten verwendet wurden kurz vorgestellt.

5.2.1 Aleph

Bei Aleph⁶ (A Learning Engine for Proposing Hypotheses) handelt es sich um ein ILP System, welches in Prolog geschrieben ist. Es handelt sich dabei um das populärste ILP System und ist der Nachfolger von P-Progol. Es dient als Prototyp um Ideen zu erforschen. Des Weiteren wurden durch die ständige Erweiterung Funktionalität von anderen ILP Systemen emuliert. [Sri04]

5.2.2 Weka

In diesem Abschnitt wird das Machine Learning Tool Weka⁷ genauer vorgestellt. Bei Weka (Waikato Environment for Knowledge Analysis) handelt es sich um eine Workbench, die Interessenten einen einfachen Zugang zu state-of-the-art Machine Learning Werkzeugen ermöglicht. Weka ist in Java geschrieben und läuft deshalb auf allen Plattformen die Java unterstützen. Dabei bietet es nicht nur Algorithmen, sondern auch ein Framework, welches es Forschern erlaubt auf einfache Weise selbst Algorithmen zu implementieren, ohne Details über Weka verstehen zu müssen. Weka bietet neben einer Vielzahl von Machine Learning Algorithmen auch Algorithmen zum Clustern, Regressionsalgorithmen, Algorithmen zum Assoziationsregeln finden und Algorithmen zur Attributauswahl an. Des Weiteren bietet es Möglichkeiten zur Datenvorverarbeitung, Ergebnisanalyse und -visualisierung an. Es bietet einheitliche Interfaces zur Analyse und Vorverarbeitung der Daten an. Dabei gibt es verschiedene Möglichkeiten Weka anzuwenden. Eine Möglichkeit ist die Anwendung eines Lernalgorithmus auf die Daten und die anschließende Auswertung der Ergebnisse um mehr über die Daten zu erfahren. Eine andere Möglichkeit, ist die Nutzung der gelernten Modelle, um Vorhersagen über neue Instanzen machen zu können. Eine dritte Möglichkeit ist, mehrere Lernalgorithmen zu Vergleichen um den Algorithmus auszuwählen, der die besten Ergebnisse für die Daten liefert. Um dies durchführen zu können bietet Weka verschiedene grafische Interfaces.[WF05]

Die am häufigsten Benutzte grafische Oberfläche ist der *Explorer*, welche die Möglichkeit bietet, auf einfache Weise Daten zu Laden, diese Vorzuverarbeiten und Lernalgorithmen auf diesen Daten anwenden zu können. Des Weiteren bietet es die Möglichkeit die Daten und die Ergebnisse zu visualisieren. Da beim *Explorer* alle Daten in den Hauptspeicher geladen werden, können nur kleine oder

⁶Quelle: <http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/aleph.html> Stand: 26. April 2011

⁷Quelle: <http://www.cs.waikato.ac.nz/ml/weka/> Stand:26. April 2011

mittelgroße Datensätze genutzt werden. Um aber auch sehr große Datensätze verarbeiten zu können bietet Weka eine grafische Benutzeroberfläche an, die es ermöglicht Daten inkrementell laden zu können. Diese Oberfläche heißt *Knowledge Flow*. Die letzte grafische Benutzeroberfläche ist der *Experimenter*. Hier lassen sich verschiedene Algorithmen zur Klassifikation oder Regression auf die Datensätze anwenden um diese miteinander zu Vergleichen. Neben den verschiedenen grafischen Oberflächen gibt es auch ein command-line Interface, welches die gleichen Möglichkeiten bietet, wie die grafischen Interfaces.[WF05]

Es ist relativ einfach Lernalgorithmen, die Weka zur Verfügung stellt, auf die eigenen Daten anwenden zu können, da alle Lernalgorithmen die Eigabedaten als einfache relationale Tabelle im ARFF Format benötigen. Im ARFF Format werden die Daten wie in Listing 5.2.2 dargestellt. Zuerst wird mit *@relation name Relation* der Name der Relation festgelegt, in diesem Fall ist das die Relation *weather*. Am Anschluss daran werden die Attribute mit *@attribut attributeName* und ihren möglichen Attributsausprägungen im geschweiften Klammern angegeben. Nur numerische werten werden durch das Schlüsselwort *numeric* ohne geschweifte Klammern angegeben. Nach der Angabe der der Attribute wird mit *@data* angegeben, dass jetzt nur noch die Instanzdaten kommen. Jede dieser Instanzen wird in einer extra Zeile angegeben und ihre Attribute werden hintereinander durch Komma getrennt aufgeschrieben.[WF05]

5.2.3 DL-Learner

DL-Learner⁸ steht für Description Logic Learner, dessen Ziel es ist, ein DL/OWL basiertes Machine Learning Tool zur Verfügung zu stellen, das überwachte Lernaufgaben (engl.. supervised learning) lösen soll und Wissensingenieuren bei der Erstellung von Wissensbasen hilft. Dabei unterstützt er auch Wissensbasen in anderen Formaten.

Als Eingabe erhält der DL-Learner ein Lernproblem, positive und negative Beispiele und einen Lernalgorithmus. Als Wissensbasis nutzt er eine Ontologie und einen Reasoner zur Klassifizierung dieser. Er überprüft die Wissensbasis und gibt mit Hilfe der positiven und negativen Beispiele neue Konzepte an, die er für geeignet hält. Die Generierung der Konzepte wird mit Hilfe eines Refinement Operators oder der genetischen Programmierung realisiert. Im Anschluss wird geprüft, ob das vorgeschlagene Konzept alle angegebenen Beispiele abdeckt. Anhand dieser Abdeckung wird das Qualitätsmaß des Konzeptes bestimmt und geprüft ob es hinreichend geeignet ist. Dieser Prozess wird so lange wiederholt, bis ein Konzept gefunden wurde, welches die Anforderungen an das Qualitätsmaß erfüllt. Die Abbildung 3 stellt dies noch einmal grafisch dar.[Leh09]

⁸<http://dl-learner.org>

Listing 1: Hier ist der Aufbau einer ARFF Datei dargestellt. Zuerst wird der Relationsname mit *@relation* definiert und im Anschluss daran werden die Attribute mit ihren möglichen Ausprägungen mit *@attribute* festgelegt. Mit *@data* wird signalisiert, dass nur noch Instanzdaten folgen. Jede Instanz wird auf einer neuen Zeile dargestellt und die Attributsausprägungen werden hintereinander, durch Komma getrennt angegeben.

```
% ARFF file for the weather data with some numeric features
%
@relation weather
@attribute outlook { sunny , overcast , rainy }
@attribute temperature numeric
@attribute humidity numeric
@attribute windy { true , false }
@attribute play? { yes , no }
@data

sunny , 85 , 85 , false , no
sunny , 80 , 90 , true , no
overcast , 83 , 86 , false , yes
...
```

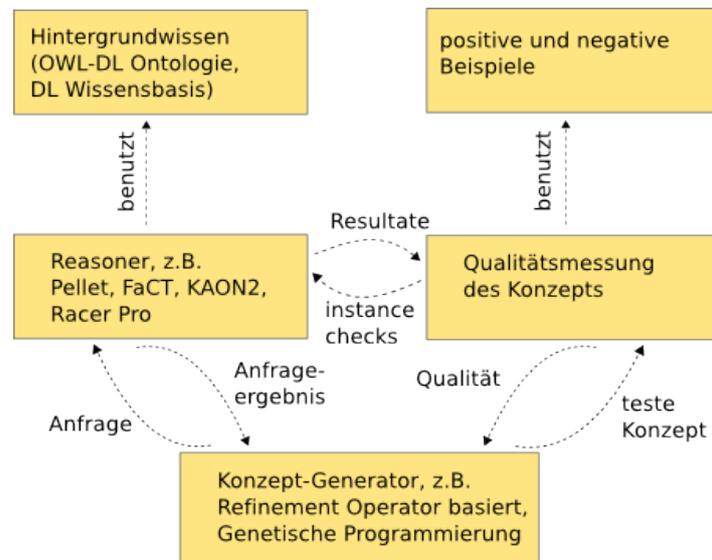


Abbildung 3: „Generate and Test“ Ansatz, der beim DL-Lerner zum lernen von Konzepten verwendet wird.[Leh09]

Die Architektur des DL-Learners

Der DL-Learner ist ein Machine-Learning Framework. Dazu stellt er verschiedene Machine Learning Algorithmen um Lernprobleme zu lösen bereit, unterstützt Wissensbasen in unterschiedlichen Formaten, stellt eine OWL-Bibliothek und ein Reasoner Interface bereit. Um auf diese Funktionalität zugreifen zu können stellt der DL-Learner eine Reihe von Interfaces bereit. Ebenfalls existieren einige Tools, welche Algorithmen des DL-Learners nutzen, wie z.B. ein Plugin für Protégé⁹ und eine Reihe von nützlichen Scripten. Der DL-Learner basiert auf einem Komponentenmodell, damit er flexibel und leicht erweiterbar ist. Dieses besteht auf vier verschiedenen Komponenten, der Wissensbasis-Komponente, der Lernproblem-Komponente, der Reasoning-Service-Komponente und der Lernalgorithmus-Komponente. Für jede dieser Komponenten existieren mehrere Implementierungen und diese haben verschiedene Konfigurationsoptionen. Mit Hilfe dieser Konfigurationsoptionen können die Parameter der einzelnen Komponenten verändert werden. Um verschiedene Komponenten zu erstellen, kombinieren und zu konfigurieren muss ein Komponentenmanager verwendet werden. Das Komponentenmodell ist in Abbildung 4 dargestellt.[Leh09]

Im folgenden werden die vier Komponententypen genauer vorgestellt.

1. **Wissensbasis-Komponente** Die Wissensbasis-Komponente kümmert sich um die Integration von Hintergrundwissen. Dabei werden fast alle OWL Formate durch die OWL API¹⁰ unterstützt, wie z.B. RDF/XML oder Manchester OWL Syntax. Des weiteren kann über die Wissensbasis Komponente auch mehrere Wissensbasen auf einmal integriert werden. Auch das extrahieren von Fragmenten einer Wissensbasis durch SPARQL¹¹ Endpunkte unterstützt der DL-Learner.
2. **Reasoning-Service-Komponente** Mit Hilfe dieser Komponente kann man bestehende oder selbst entwickelte Reasoner mit dem DL-Learner verlinken. Er besitzt bereits zwei Interfaces, das DIG-Reasoner¹² und OWL API Reasoner Interface, womit man alle Standard OWL Reasoner mit Hilfe von HTTP oder XML basierten Mechanismen verbinden kann. Weiterhin bietet der DL-Learner einen eigenen Reasoner, welcher Pellet für den Bootstrapping Vorgang und das Laden des Modells in den Speicher zuständig ist. Anschließend wird die Überprüfung der Instanzdaten sehr effizient mit Hilfe von (Lokaler closed world Annahme) durchgeführt.
3. **Lernproblem-Komponente** Durch diese Komponente wird der Problemtyp, welcher von einem Algorithmus gelöst werden soll spezifiziert. Es existieren zur Zeit drei Problem-

⁹<http://protege.stanford.edu>

¹⁰<http://owlapi.sourceforge.org>

¹¹<http://www.w3.org/TR/rdf-sparql-query/>

¹²<http://dl.kr.org/dig/>

Komponenten. Erstens das Lernen von positiven und negativen Beispielen, zweitens das Lernen nur von positiven Beispielen und drittens das Lernen von Klassenaxiomen. Dabei ist dieses noch einmal unterteilt in das Lernen von Klassendefinitionen und das Lernen von Superklassen. Die Lernprobleme bieten effiziente Abdeckungschecks, die für Lernalgorithmen genutzt werden können.

4. **Lernalgorithmus-Komponente** Diese Komponente bietet Methoden an, um ein oder mehrere Lernproblemtypen lösen zu können. Neben einigen simplen Algorithmen, welche Brute Force - oder Zufallsratetechniken benutzen, besitzt der DL-Lerner auch einige ausgefeilteren Algorithmen, welche auf genetischer Programmierung mit einem neuartigen genetischen Operator, oder auf einem Refinement Operator für Beschreibungslogik ALC basieren. Außerdem bietet der DL-Lerner einen erweiterten Operator an, der viele Eigenschaften von OWL, wie Datentyp Unterstützung, beinhaltet. Es gibt auch einen Algorithmus der speziell auf die Ontologieerstellung zugeschnitten ist. Dieser legt wert auf kurze und gut lesbare Konzepte.[Leh09]

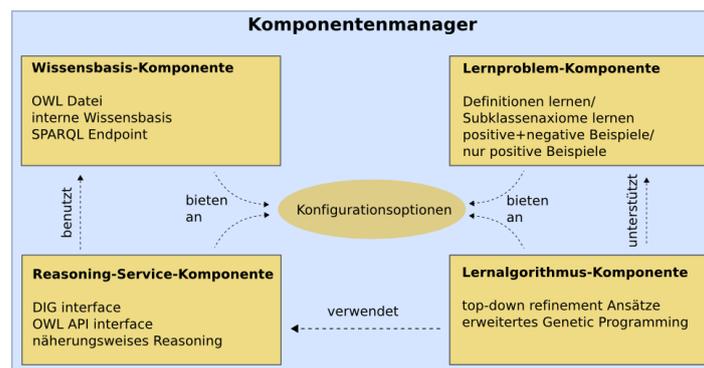


Abbildung 4: Darstellung der Architektur des DL-Learners. Dabei sind die vier Komponententypen, der Wissensbasis-Komponente, der Lernproblem-Komponente, der Reasoning-Service-Komponente und der Lernalgorithmus-Komponente dargestellt, welche alle ihre eigenen Konfigurationsoptionen besitzen. Um verschiedene Komponenten zu kombinieren, muss ein Komponenten-Manager verwendet werden.[Leh09]

6 Vorhersage von mutagenen Eigenschaften

Das Problem Vorhersage von mutagenen Eigenschaften beschäftigt sich mit der Erkennung von Substanzen, welche Schäden an der DNA verursachen können. Dabei beschränkt man sich nicht nur auf die Intensiv mutagenen Präparate, sondern auch auf solche, die ihre Wirkung erst nach längerer Einnahme entfalten. Für die Vorhersage wird das sehr Allgemeine QSAR Modell verwendet, um Eigenschaften bei aromatischen und heteroaromatischen Nitropräparaten zu finden, welche verändernde Wirkung haben. Vor allem aromatische Nitrosubstanzen sind von großem Interesse. Da diese sowohl mutagen, als auch krebserregend sind und auch heutzutage noch in einigen Medikamenten zu finden sind, ist es wichtig zu untersuchen welche Eigenschaften dazu führen, dass chemische Verbindungen auf Nitrobasis, geringe mutagenen Eigenschaften besitzen. Aber auch Nitroaromatische Substanzen sind von Interesse, da diese krebserregend sind und in Abgasen von Autos und Dieselfahrzeugen vorkommt. Krebserregend bedeutet nicht das diese Stoffe gleichzeitig mutagen sind, aber der Zusammenhang zwischen beiden ist so groß, dass man mit Hilfe von Test auf mutagene Eigenschaften, eine preiswerte Methode besitzt um gefährliche Substanzen zu identifizieren. Deshalb wurde von testenden Ansätzen zu vorhersagenden Ansätzen übergegangen. Dabei wurde mit Hilfe von Prolog eine Wissensbasis mit Substanzen erstellt, um Eigenschaften zu identifizieren die mutagene Auswirkungen haben können. Es wird nun überprüft ob, mit Hilfe der Ontologiesprache OWL, weitere Eigenschaften gelernt werden können, die mit Hilfe von Prolog nicht gelernt wurden. Im nächsten Abschnitt wird genauer auf die Struktur der Daten eingegangen.[DLdCD⁺91]

6.1 Datensatz

Der Datensatz stammt von der ILP Machine Learning Group Home Page ¹³ und besteht aus 230 Substanzen mit 138 als mutagen wirkenden Substanzen, die als aktiv gekennzeichnet wurden und den verbleibenden 92 Stoffen, welche nicht mutagen sind und als inaktiv gekennzeichnet wurden. Diese 230 Präparate wurden noch einmal in zwei Gruppen unterteilt. Die erste Gruppe besteht aus 188 Substanzen, welche durch lineare Regression zugeschnitten werden konnten und die andere Gruppe besteht aus den verbleibenden 42 Datensätze, welche nicht durch lineare Regression zugeschnitten werden konnten. Diese Daten waren in Form einer PROGOL Wissensbasis dargestellt. Dabei werden die Substanzen mit Hilfe ihrer Atom und Atomverbindungen dargestellt, welche wie folgt dargestellt sind. Die Atome werden mit *atm(compound,atom,element,atomtype,charge)* dargestellt. Das bedeutet, dass das Atom *atom* in der chemischen Verbindung *compound* vom Element *element* des Atomtypen *atomtype* ist und eine Ladung *charge* besitzt. An einem Beispiele dargestellt, bedeutet *atm(d1,d1_20,c,22,-0.117)*, dass das Atom *d1_20* in der Verbindung *d1* ein aroma-

¹³Quelle:<http://www.doc.ic.ac.uk/shm/mutagenesis.html> Stand:26. April 2011

tisches Karbon(22) mit einer Ladung von -0.117 ist. Die Atomverbindungen werden mit Hilfe von $bond(compound,atom1,atom2,bondtype)$ dargestellt und bedeutet, dass in der Struktur $compound$ eine Verbindung vom Typ $bondtype$ zwischen den Atomen $atom1$ und $atom2$ besteht. An einem Beispiel dargestellt, bedeutet z.B. $bond(d1,d1_1,d1_2,7)$, dass eine aromatische Verbindung(7) zwischen den Atomen $d1_1$ und $d1_2$ in der Struktur $d1$ vorliegt. Des weiteren haben alle chemischen Verbindungen noch 5 weitere Eigenschaften, welche im folgenden vorgestellt werden.

- **log P** Bei dieser Eigenschaft handelt es sich um den logarithmischen Wert des Octanol/Wasser Koeffizienten dieser Substanz.
- ϵ_{LUMO}
Diese Eigenschaft beschreibt die Energie des niedrigsten unbesetzten molekularen Orbitals in der chemischen Verbindung.
- **act** Hierbei handelt es sich um die Angabe wie hoch die mutagenen Eigenschaften sind. Dabei sind Werte > 0 sehr Mutagen und werte darunter nur gering Mutagen
- **I₁**
Dabei stellt I₁ eine Indikatorvariable dar, welche auf 1 gesetzt ist, wenn die Verbindung drei oder mehr verschmolzene Ringe besitzt.
- **I_a**
I_a ist eine Indikatorvariable die auf 1 gesetzt wird, wenn es fünf Acenthrylene Moleküle gibt, welche aus unbekanntem Gründen weniger Aktiv sind als erwartet.

In dem folgenden Beispiel(Listing 6.1) wird Auszugsweise die Codierung der Eigenschaften einer bestimmten Substanz in Prolog dargestellt.

Im folgenden Abschnitt wird die Umwandlung der Prolog Datei in eine OWL Ontologie beschrieben.

Umwandlung in eine OWL Ontologie

Da der DL-Learner nicht mit Prolog Dateien lernen kann, mussten die vorliegenden Dateien zuerst in eine OWL Ontologie umgewandelt werden. Dazu wurde der Prolog Parser von Sebastian Bader, welches in Java geschrieben wurde, verwendet und eine Klasse geschrieben, die die Prologklauseln in OWL Ausdrücke umwandelt. In der Abbildung 5 ist der Ablauf zum Mappen der Prologklauseln, mit den wichtigsten Methoden dargestellt und im folgenden werden die wichtigsten von ihnen beschrieben. Die *main()* Methode steuert den Ablauf der Umwandlung. Diese beginnt mit dem Einlesen der

Listing 2: Prolog Datei der Mutagenesis Wissensbasis. Hier sind die Atom und Atombindungen dargestellt. Die Blöcke 3 bis 7 sind für die einzelnen Eigenschaften die oben aufgezählt sind.

```
atm(d100, d100_1, c, 22, -0.128).  
atm(d100, d100_3, c, 22, -0.128).  
  
bond(d100, d100_1, d100_2, 7).  
bond(d100, d100_3, d100_4, 7).  
  
lumo(d21, -1.665).  
lumo(d22, -1.602).  
  
logp(d37, 4.18).  
logp(d44, 4.53).  
  
act(d14, -1.10).  
act(d15, 2.78).  
  
inda(d22, 0.0).  
inda(d23, 1.0).  
  
ind1(d13, 1.0).  
ind1(d14, 0.0).
```

Prolog Dateien. Anschließend überprüft ein Prologparser ob die Prologdateien korrekte Ausdrücke beinhalten. Dann wird die OWL Ontologie vorbereitet und es werden für alle *Data - und ObjectProperties* die *Domain und Range* festgelegt. Diese benötigt der DL-Learner bei dem Lernvorgang um seine Suche einzuschränken. Danach wird das Mapping der Prologklauseln durchgeführt. Diese Aufgabe übernimmt die Methode *mapClause()*. Dabei benutzt es die Methoden *getAtomClass()*, *getRoleAssertion()*, *getBooleanDatatypePropertyAssertion()*, *getDoubleDatatypePropertyAssertion()*, *getConceptAssertion()*, *addStructureSubclass()*, *getAtomicConcept()* und *getFullElementName()*, um die Ausdrücke umzuwandeln. Im Anschluss an das Umwandeln wird die entsprechende OWL Ontologie geschrieben und die zum Lernen benötigten *conf Dateien* erzeugt. Dabei wird zuerst der Kopf durch *generateConfFile()* erzeugt, welcher die Informationen für Lernalgorithmus, Reasoner, Laufzeit und den Noisewert enthält. Danach werden mit *generatePositiveExamples()* die positiven und negativen Beispiele zum Lernen erzeugt und anschließend mit *appendExamples()* in die *conf Datei* geschrieben. Bei der Umwandlung entstand eine Ontologie mit 86 Klassen, 5 Object Properties, 6 Data Properties und 14145 Instanzdaten.

6.2 Methoden

In diesem Abschnitt werden die einzelnen Tests, die durchgeführt wurden, genauer beschrieben. Zuerst wurde die Ontologie mit Hilfe des OCEL Algorithmus überprüft. Dazu wurde der Noise Parameter schrittweise von 30%, um jeweils 1% reduziert und eine 10-fold-cross validation durchgeführt, bis keine Ergebnisse mehr entstanden. Anschließend wurde dieser Vorgang mit dem CELOE Algorithmus wiederholt. Ausgehend von diesen Ergebnissen wurde mit Hilfe von Aleph ein weiterer Durchlauf gemacht. es wurde wieder der Noise Parameter von 30% um jeweils 1% reduziert und eine 10-fold-cross-validation durchgeführt. Anhand der damit gewonnenen Daten wurde ein Vergleich der verschiedenen Algorithmen durchgeführt. In einem weiteren Test wurden mit OCEL und CELOE jeweils mit fester Laufzeit von 10, 100, 300 und 1000 Sekunden anhand einer 10-fold-cross-validation verglichen welcher Lernalgorithmus bessere Ergebnisse erzielt. Des Weiteren wurden die Laufzeitunterschiede von OCEL und CELOE genauer betrachtet. Um ein möglichst gleiches Verhalten der beiden Algorithmen zu erzielen wurde der β Parameter, welcher die Bestrafung für längere Konzepte durchführt, bei OCEL und CELOE auf 0.02 gesetzt. Alle Tests wurden auf einem 3,2 GHz Quad Core mit 8GB Ram durchgeführt. Diese Ergebnisse und die sich daraus ergebenden Schlussfolgerungen werden im nächsten Abschnitt genauer beleuchtet.

6.3 Resultate und Schlussfolgerungen

In Abbildung 6 werden die Vorhersagewahrscheinlichkeiten der einzelnen Algorithmen in Bezug auf den eingestellten Noise Parameter dargestellt. Dabei wird der Noisewert von 30% in 1% Schritten

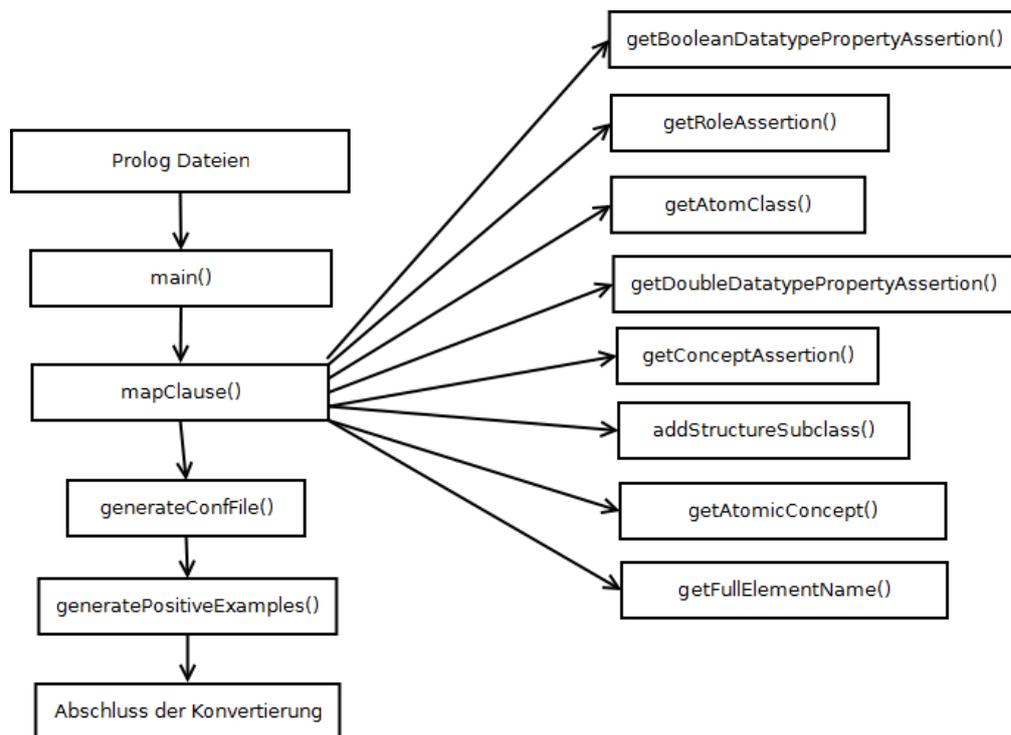


Abbildung 5: Hier ist der Ablauf zur Konvertierung der Prologdateien der Mutagenesis Wissensbasis in eine OWL Ontologie mit ihren Methoden dargestellt.

solange verringert, bis bei den einzelnen Algorithmen keine Lernergebnisse mehr erzeugt werden. Dazu ist anzumerken, dass sich bei OCEL und CELOE die Genauigkeiten der Testdaten in gewissen Bereichen wieder verringern. Dies kann daran liegen, dass der DL-Lerner Konzepte lernt, welche gut zu den Trainingsdaten passen, aber für die Testdaten nicht so geeignet oder zu speziell sind. Für Aleph und den DL-Lerner entstanden folgende Ergebnisse, die in Tabelle 2 dargestellt sind.

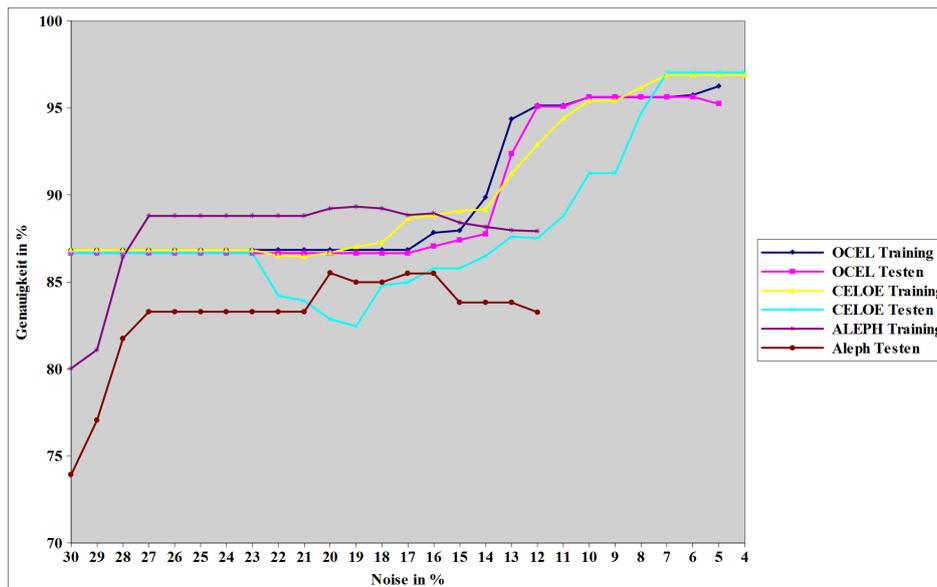


Abbildung 6: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für den ersten Teil der Mutagenesis Wissensbasis. Der Abfall der Genauigkeit bei OCEL - und CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind.

In der Tabelle 2 sind die höchsten Genauigkeiten der 10-fold-cross-validation für die Lernalgorithmen mit den entsprechenden Laufzeiten und ihren Konzeptlängen dargestellt. Dazu ist anzumerken, dass bei Aleph die Laufzeit und Konzeptlänge nicht mitprotokolliert wurde. Diese Ergebnisse dienen als Grundlage zur Berechnung des p-Werts in Tabelle 3. Es ist zu erkennen, dass CELOE die höchste durchschnittliche Testgenauigkeit hat, dicht gefolgt von OCEL. Dabei benötigen beide Algorithmen ungefähr die gleiche Laufzeit. Nur die Konzeptlänge, die bei OCEL 10,5 und bei CELOE 3 betrug ist unterschiedlich. Des Weiteren ist zu sehen, dass Aleph die Ergebnisse mit der geringsten Genauigkeit produziert.

Um herauszufinden ob diese Unterschiede aus Tabelle 2 statistisch signifikant sind oder nicht, wurde der t-Test durchgeführt. Die Ergebnisse und ob eine statistische Signifikanz vorlag wird in Tabelle 3 dargestellt. Es ist zu sehen, dass die beiden Algorithmen OCEL und CELOE statistisch signifikant besser sind als Aleph. Nur im direkten Vergleich zwischen beiden Algorithmen liegt keine statistische

Algorithmus	Testgenauigkeit	Laufzeit	Konzeptlänge
OCEL	95,63%±3,90%	2,233s±0,972s	10,5±2,415
CELOE	97,03%±2,854%	2,377s±4,003s	3±0
Aleph	85,51%±9,08%	–	–

Tabelle 2: In dieser Tabelle werden die höchsten Genauigkeiten und die dazu entsprechenden Laufzeiten und Konzeptlängen der 10-fold-cross-validation der Algorithmen OCEL, CELOE und Aleph, für den ersten Teil der Mutagenesis Wissensbasis dargestellt

Signifikanz vor.

Algorithmen	p-Wert	statistische Signifikanz
OCEL vs. Aleph	0.0046	ja
CELOE vs. Aleph	0.0012	ja
CELOE vs. OCEL	0.3717	nein

Tabelle 3: In dieser Tabelle wird die statistische Signifikanz der Algorithmen dargestellt. Dabei ist deutlich zu sehen, dass sowohl OCEL als auch CELOE statistisch signifikant besser Arbeiten als Aleph. Nur in dem Vergleich zwischen OCEL und CELOE liegt keine statistische Signifikanz vor.

In diesem Abschnitt werden die besten Ergebnisse der Algorithmen in Tabelle dargestellt. Es ist zu sehen, dass alle drei Algorithmen Ergebnisse mit einer Genauigkeit von 100% vorschlagen. Das Ergebnis von OCEL bedeutet, dass Substanzen mit einem act-Wert größer gleich -0.205, mehr als 4 Strukturen, die nicht Methyl sind besitzt und einen logp-Wert größer gleich 1.725 oder einen lumo-Wert kleiner gleich -1.765, mutagene Eigenschaften besitzen. Bei CELOE bedeutet das vorgeschlagene Konzept, dass Substanzen mit einem act-Wert größer gleich -0.205 mutagene Eigenschaften haben. Somit erzeugt der CELOE Algorithmus eine viel kürzere Lösung, die nur den act-Wert berücksichtigt. Die Theorie die Aleph ausgegeben hat besagt, dass ein Atom mit einem logp-Wert von 4,23 (dargestellt durch Regel 1) und ein Atom, welches ein aromatisches Karbon mit einer Ladung von -0,119 (dargestellt durch Regel 2) und ein Atom welches einen lumo-Wert von -1,749 (dargestellt durch Regel 3) und ein Atom mit einer Ringgröße von 5 die einen logp-Wert von 2,74 (dargestellt durch Regel 5) und eine Ball3 Ringstruktur besitzen, zu den Substanzen gehören, die mutagene Eigenschaften haben. Um herauszufinden warum CELOE ein kürzeres Konzept vorschlägt als OCEL wurden für beide Algorithmen der Suchbaum ausgegeben. Die Auswertung dieses Suchbaumes für OCEL ergab, dass $act \geq -0.20500000000000002$ nicht alleine als Konzept auftaucht. Es erscheint immer nur in Kombination eines lumo Wertes und erreicht dabei eine maximale DL-Learner interne Genauigkeit von 94,048%. OCEL findet somit durch die leicht veränderte Heuristik im Gegensatz zu CELOE das Konzept $act \geq -0.20500000000000002$ nicht. Des Weiteren werden die Konzepte auch

anders bewertet. CELOE hingegen bewertet das Konzept $act \geq -0.20500000000000002$ mit einer DL-Learner internen Genauigkeit von 95,83% und selbst wenn Längere Konzepte gefunden werden kann keine Verbesserung der Genauigkeit mehr erzielt werden. Aus diesem Grund gibt CELOE das beste gefundene Konzept mit $act \geq -0.20500000000000002$ an.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	$(Compound \ AND \ act \ \geq \ -0.20500000000000002 \ AND \ \geq \ 4 \ hasStructure.(NOT \ Methyl) \ AND \ (logp \ \geq \ 1.725 \ OR \ lumo \ \leq \ -1.7650000000000001))$	100%	–	19
CELOE	$(act \ \geq \ -0.20500000000000002)$	100%	–	–
Aleph	$[Rule \ 1] [Pos \ cover = 53 \ Neg \ cover = 5] \ active(A) :- \ logp(A,B), \ gteq(B,4.23).$ $[Rule \ 2] [Pos \ cover = 6 \ Neg \ cover = 0] \ active(A) :- \ atm(A,B,c,22,C), \ C = -0.119.$ $[Rule \ 3] [Pos \ cover = 61 \ Neg \ cover = 6] \ active(A) :- \ lumo(A,B), \ lteq(B,-1.749).$ $[Rule \ 5] [Pos \ cover = 42 \ Neg \ cover = 1] \ active(A) :- \ logp(A,B), \ gteq(B,2.74), \ ring_size_5(A,C).$ $[Rule \ 6] [Pos \ cover = 26 \ Neg \ cover = 0] \ active(A) :- \ ball3(A,B).$	100%	–	–

Tabelle 4: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL, CELOE und Aleph für den ersten Teil der Wissensbasis Mutagenesis dargestellt. Dabei ist deutlich zu sehen, dass alle drei Algorithmen Ergebnisse mit einer Genauigkeit von 100% vorschlagen.

In der Abbildung 7 ist die Laufzeit von CELOE und OCEL in Abhängigkeit des eingestellten Noise Parameters dargestellt. Dabei ist zu erkennen, dass die Laufzeit bei beiden Algorithmen bis 6% Noise bei unter 10 Sekunden liegt. Jedoch nimmt die Laufzeit von OCEL bei 5% Noise sehr stark zu und liegt bei 158 Sekunden. Die Laufzeit von CELOE hingegen bleibt ungefähr bei 4 - 5 Sekunden. Dieser Unterschied ist damit zu erklären, dass CELOE für die Suche eines Konzeptes welches 5% Noise besitzt nicht viel Zeit benötigt. Das liegt an der Einfachheit des Konzeptes, in diesem Fall $act \geq -0.20500000000000002$, denn dieses wird direkt gefunden und somit muss der Algorithmus nicht lange suchen, um das geeignete Konzept zu finden. OCEL hingegen benötigt mehr Zeit um ein geeignetes Konzept zu finden, welches mit einem Noise Wert von 5% bewertet wird, denn es muss mehr

Möglichkeiten in dem Suchbaum durchgehen. Um zu überprüfen ob die Unterschiede in der Laufzeit des Algorithmus statistisch Signifikant sind wird für 5% Noise ein t-Test durchgeführt. Dabei wurde ein p-Wert von 0,0915 errechnet und das bedeutet, dass der unterschied in der Laufzeit nicht statistisch Signifikant ist.

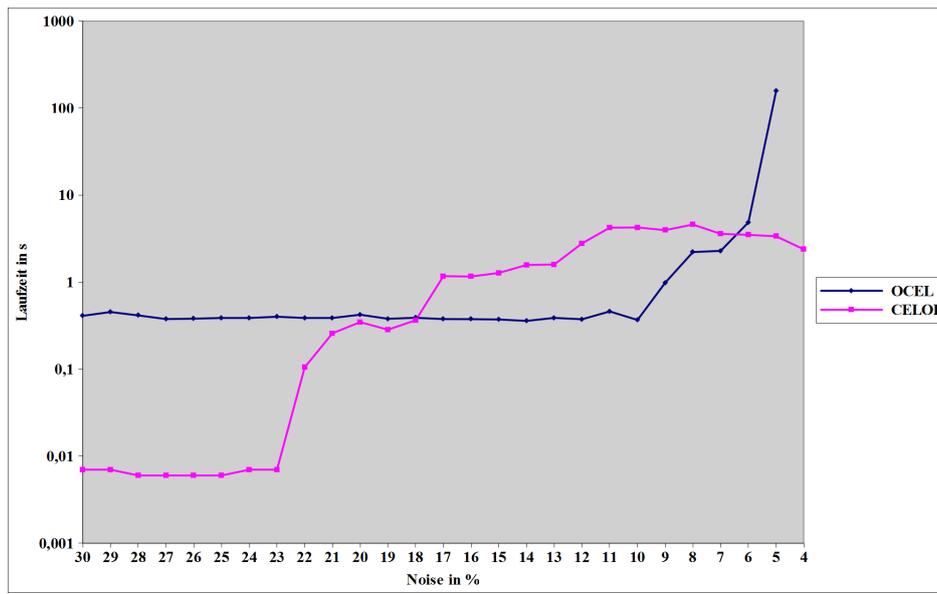


Abbildung 7: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für den ersten Teil der Mutagenesis Wissensbasis. Dabei ist deutlich zu sehen, dass die Laufzeiten bei geringem Noise Parameter für OCEL stark zunehmen.

In der Abbildung 8 wird die Genauigkeit von OCEL und CELOE in Bezug auf die Laufzeit dargestellt. Dabei wurden als Laufzeiten 10, 100, 300 und 1000 Sekunden gewählt. Das Fehlen der Ergebnisse der OCEL Test- und Trainingsdaten bei 1000 Sekunden liegt daran, dass der OCEL Algorithmus viel Arbeitsspeicher benötigt um die Konzepte zu speichern und vor Ende der gesamten Laufzeit eine *OutOfMemoryException* eingetreten ist. Dazu ist zu sagen, dass OCEL im Suchbaum auch die nicht abgedeckten positiven und negativen Beispiele speichert. Diese werden wegen der Subsumtion auch von den Kindsknoten nicht abgedeckt. Da es deswegen zu weniger Reasoner aufrufen kommt, da die nicht abgedeckten Beispiele in den Knoten gespeichert sind, ist der Speicherbedarf von OCEL höher als der von CELOE. Aus diesem Grund kam es bei OCEL zu der *OutOfMemoryException*. Die geringere Testgenauigkeit im Gegensatz zu dem obigen Test, wo der Noise Parameter fest eingestellt wurde, lässt sich damit erklären, dass CELOE viele Konzepte mit geringer, DL-Learner intern vergebener Genauigkeit findet. Diese werden nicht verworfen, da bei dem Laufzeit Tests keine Mindestgenauigkeit vergeben wurde. Somit durchsucht der Algorithmus viele Konzepte mit geringer Genauigkeit und braucht länger um zu den besser bewerteten Ergebnissen zu kommen. Somit

steigt die Testgenauigkeit viel langsamer als in dem vorherigen Test wo ein fester Noise Wert vergeben wurde. Des Weiteren ist bei der Auswertung des Suchbaumes aufgefallen, dass CELOE im Gegensatz zu OCEL oft als besten Knoten den Knoten Thing angibt und somit bei den einfacheren Konzepten, die eine schlechtere Bewertung bekommen haben bleibt. Das CELOE bei diesem Konzepten bleibt, liegt daran, dass keine Ergebnisse mit einer besseren intern berechneten Genauigkeit gefunden wurden und nach der Bestrafung von längeren Konzepten Thing als kürzestes und somit am besten bewertetes Konzept übrig bleibt. Aus diesem Grund steigt die Genauigkeit bei CELOE viel langsamer als die von OCEL.

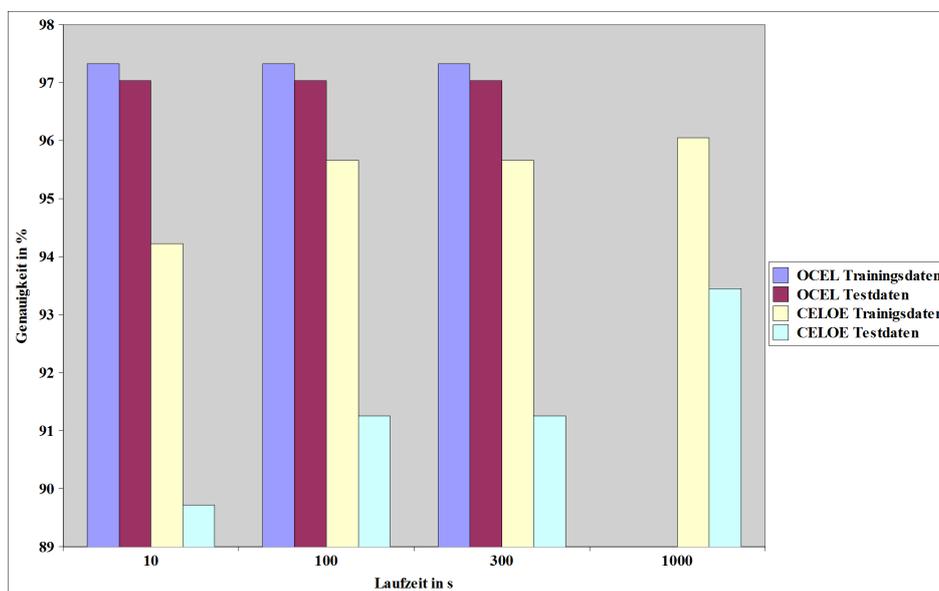


Abbildung 8: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für den ersten Teil der Mutagenesis Wissensbasis. Das Fehlen der Ergebnisse für 1000s bei OCEL liegt daran, dass nicht genügend Arbeitsspeicher zur Verfügung stand und der Algorithmus mit einer *OutOfMemoryException* abgebrochen ist.

In Tabelle 5 werden die Testgenauigkeiten der 10-fold-cross-validation dargestellt. Des Weiteren werden noch die p-Werte des t-Tests dargestellt und ob eine statistische Signifikanz zwischen den Algorithmen vorlag. Dabei ist zu erkennen, dass bei der Veränderung der Laufzeiten eine Veränderung der Testgenauigkeit bei CELOE erreicht wurde. Die Unterschiede sind aber in allen drei Fällen nicht statistisch signifikant.

In der Abbildung 9 werden die einzelnen Genauigkeiten der Algorithmen OCEL, CELOE und Aleph in Bezug auf den eingestellten Noise Parameter dargestellt. Dabei wurde der Noise Parameter wieder von 30% jeweils um 1% verringert, bis keine Lernergebnisse mehr erzielt wurden. Hierbei ist deutlich zu sehen, dass OCEL und CELOE eine höhere Genauigkeit erzielen als Aleph.

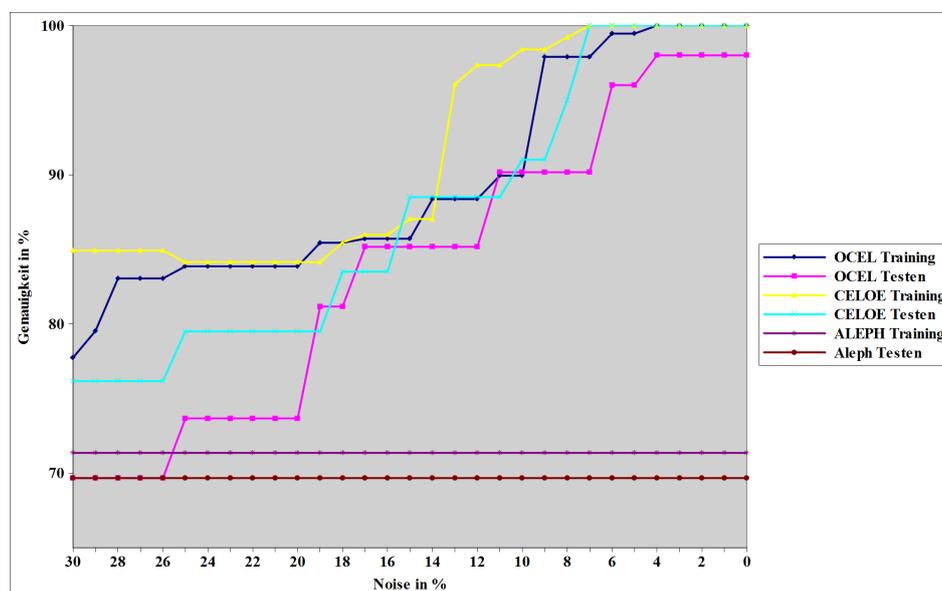


Abbildung 9: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Nnoise Parameter für den zweiten Teil der Mutagenesis Wissensbasis. Dabei ist zu erkennen, dass CELOE sowohl bei Trainings- als auch Testdaten 100% erreicht und OCEL nur bei den Trainingsdaten. Hierbei erreicht OCEL nur 98%. Aleph erreicht bei diesem Teil der Wissensbasis nur ca. 72%.

Laufzeit	Testgenauigkeit Algorithmus 1	Testgenauigkeit Algorithmus 2	p-Wert	statistische Signifikanz
10 Sekunden	97,031%±2,854%	89,72%±14,245%	0.0534	nein
100 Sekunden	97,031%±2,854%	91,25%±8,983	0.0683	nein
300 Sekunden	97,031%±2,854%	91,25%±8,983	0.0683	nein

Tabelle 5: Bei dieser Tabelle handelt es sich um die Darstellung der Testgenauigkeit der Algorithmen OCEL und CELOE bei 10, 100 und 300 Sekunden Laufzeit und die Ergebnisse des t-Tests und ob eine statistische Signifikanz in den Unterschieden der Genauigkeiten vorliegt.

In der Tabelle 6 sind die höchsten Testgenauigkeiten der Algorithmen OCEL, CELOE und Aleph, mit den entsprechenden Laufzeiten und Konzeptlängen dargestellt. Dabei ist deutlich zu sehen, dass sowohl OCEL als auch CELOE viel höhere Genauigkeiten liefern als Aleph. Der Unterschied zwischen OCEL und CELOE lässt sich damit erklären, dass OCEL bei der 10-fold-cross-validation bei dem ersten Fold das Konzept *Compound AND act* ≥ 0.4 vorschlägt, welches zwar eine Trainingsgenauigkeit von 100% besitzt, aber nicht alle positiven Testdaten abdeckt. In diesem Fall wurden zwei positive Instanzen nicht abgedeckt. Dieses wird trotzdem als Lösung ausgegeben weil der Noise-Wert der Trainingsdaten von 0% erreicht wurde. Die Ergebnisse der restlichen Folds stimmen wieder überein. Aus diesem Grund erreicht OCEL eine geringere Testgenauigkeit als CELOE.

Algorithmus	Testgenauigkeit	Laufzeit	Konzeptlänge
OCEL	98%±6,325%	0.387s±0.051s	5±0
CELOE	100%±0%	0.014s±0.011s	3±0
Aleph	69.667%±7.149%	–	–

Tabelle 6: In dieser Tabelle werden die höchsten Genauigkeiten, Laufzeiten und Konzeptlängen der 10-fold-cross-validation der Algorithmen OCEL, CELOE und Aleph für den zweiten Teil der Mutagenesis Wissensbasis dargestellt

Im nächsten Schritt wurde der t-Test durchgeführt um zu überprüfen ob die Ergebnisse statistisch Signifikant sind. Die Tabelle 7 zeigt dabei die Ergebnisse des t-Tests. Bei dem Vergleich von OCEL und CELOE mit Aleph kamen sehr kleine p-Werte heraus, so dass bei diesen Vergleich eine extreme statistische Signifikanz zwischen den Algorithmen vorlag. Jedoch bei dem Vergleich zwischen OCEL und CELOE liegt keine statistische Signifikanz vor, da der p-Wert größer 0.05 ist. Abschließend ist dazu zu sagen, das der DL-Learner statistisch relevante Unterschiede zu Aleph liefert, aber keine statistisch wichtigen Unterschiede zwischen OCEL und CELOE liefert.

In der Tabelle 8 werden die Konzepte mit der höchsten Bewertung vorgestellt. Dabei ist zu sehen, dass OCEL und CELOE das gleiche Konzept vorschlagen. Der Vorschlag von OCEL und CELOE bedeutet, dass eine Substanz einen act-Wert größer gleich -0.205 besitzen muss um mutagene Ei-

Algorithmen	p-Wert	statistische Signifikanz
OCEL vs. Aleph	0.0001	ja
CELOE vs. Aleph	0.0001	ja
CELOE vs. OCEL	0.3306	nein

Tabelle 7: In dieser Tabelle wird die statistische Signifikanz der Algorithmen dargestellt. Dabei ist deutlich zu sehen, dass sowohl OCEL als auch CELOE statistisch signifikant besser arbeiten als Aleph. Nur in dem Vergleich zwischen OCEL und CELOE liegt keine statistische Signifikanz vor.

genschaften zu besitzen. Der OCEL Algorithmus erreicht aber eine geringere durchschnittliche Testwahrscheinlichkeit, als CELOE, weil er bei dem ersten Fold das Konzept *Compound AND act >= 0.4* vorschlägt, welches zwar eine Trainingsgenauigkeit von 100% besitzt, aber nicht alle positiven Testdaten abdeckt. In diesem Fall wurden zwei positive Instanzen nicht abgedeckt. Dieses wird trotzdem als Lösung ausgegeben, weil der Noise-Wert der Trainingsdaten von 0% erreicht wurde. Die Ergebnisse der restlichen Folds stimmen wieder überein. Dadurch ist abschließend zu sagen, dass OCEL und CELOE bei dem zweiten Teil der Mutagenesis Ontologie die selben Ergebnisse erzielt. Aleph konnte bei dem zweiten Teil der Mutagenesis Wissensbasis keine Regeln finden, die Substanzen beschreiben, die mutagene Eigenschaften besitzen. Lediglich die Substanz *f5* konnte als Lösung angegeben werden, da diese genügend Testdaten abgedeckt hat. Eine Regel ließ sich daraus aber nicht bilden.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Compound AND act >= -0.20500000000000002)</i>	100%	–	5
CELOE	<i>(act >= -0.20500000000000002)</i>	100%	–	3
Aleph	<i>active(f5).</i>	0,75	–	–

Tabelle 8: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL, CELOE und Aleph für den zweiten Teil der Wissensbasis Mutagenesis dargestellt. Dabei ist deutlich zu sehen, dass OCEL und CELOE das gleiche Konzept vorschlagen.

In der Abbildung 10 wird die durchschnittliche Laufzeit der Algorithmen OCEL und CELOE in Bezug auf den eingestellten Noise Parameter dargestellt. Dabei ist zu erkennen, dass die Laufzeit von CELOE geringer ist als die Laufzeit von OCEL, da jedoch die Laufzeit im allgemeinen sehr gering ist, bei OCEL $\leq 0.5s$ und bei CELOE $\leq 0.01s$, kann man sagen, dass dieses Lernproblem nicht sehr schwierig für den DL-Lerner zu lösen war. Die höheren Laufzeiten für OCEL lassen sich durch den erhöhten Speicherverbrauch beim Aufbau des Suchbaumes, da er die nicht abgedeckten positiven und negativen Beispiele mit in den jeweiligen Knoten speichert und den damit erhöhten Suchzeiten für die Lösung erklären.

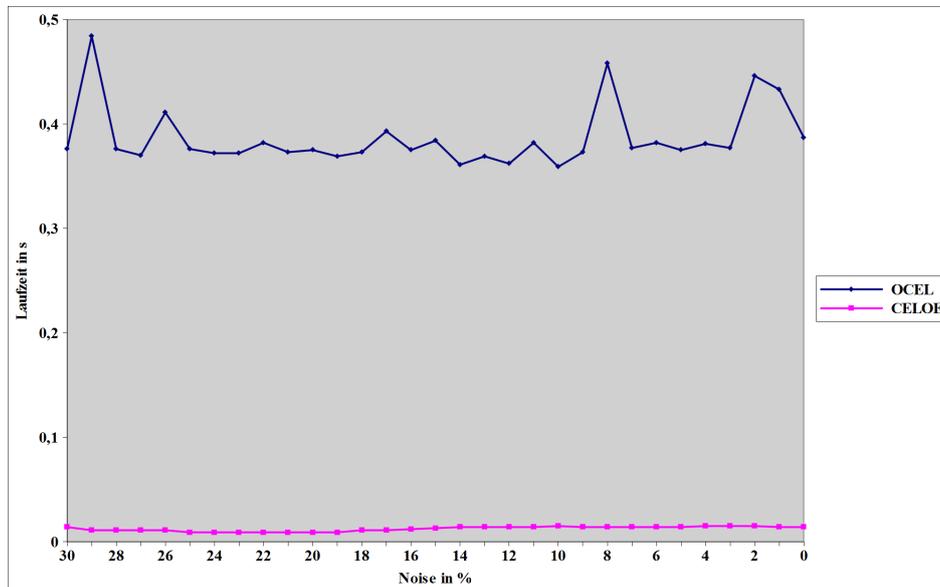


Abbildung 10: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für den zweiten Teil der Mutagenesis Wissensbasis. Dabei ist deutlich zu sehen, dass die Laufzeiten für OCEL höher sind als die für CELOE.

Die Abbildung 11 stellt die Genauigkeit der Algorithmen OCEL und CELOE in Bezug auf den eingestellten Laufzeit Parameter dar. Es wurden dabei die Laufzeiten 10, 100, 300 und 1000 Sekunden gewählt. Bei OCEL konnte kein Test mit 1000 Sekunden durchgeführt werden, da der Algorithmus viel Arbeitsspeicher benötigt und beim Ausführen eine *OutOfMemoryException* erschien. Bei allen Laufzeiten und bei beiden Algorithmen wurden jeweils eine Trainingsgenauigkeit von 100% und einer Testgenauigkeit von 98%. Das die Testgenauigkeit bei CELOE nur bei 98% liegt, im Gegensatz zu der Testgenauigkeit von 100% bei der Nutzung des Noise-Parameters, könnte daran liegen, dass der Algorithmus beim weiterlernen noch Konzepte findet, die eine Trainingsgenauigkeit von 100% besitzen, aber nicht alle Testinstanzen abdecken können. In diesem Fall ist dass das Konzept $act \geq 0.4$, welches eine Trainingsgenauigkeit von 100% hat, aber zwei der positiven Beispiele der Testdaten nicht abdeckt. Das gleiche Problem ist auch bei OCEL zu erkennen. Da es bei der Laufzeituntersuchung keine Unterschiede zwischen den beiden Algorithmen OCEL und CELOE gab, wurde kein t-Test durchgeführt.

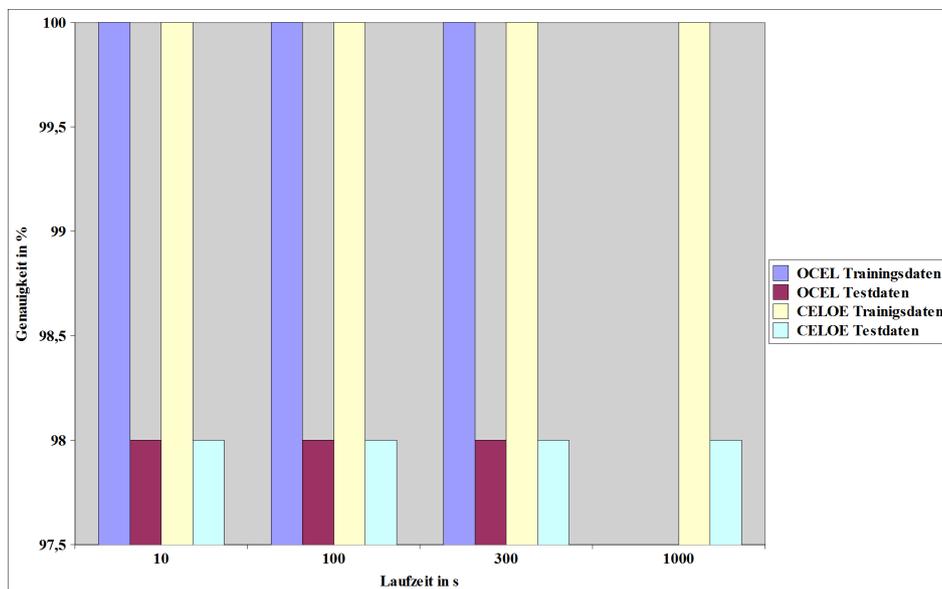


Abbildung 11: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für den zweiten Teil der Mutagenesis Wissensbasis. Das Fehlen der Ergebnisse für 1000s bei OCEL liegt daran, dass nicht genügend Arbeitsspeicher zur Verfügung stand und der Algorithmus mit einer *OutOfMemoryException* abgebrochen ist.

7 Suramin-ähnliche Substanzen zur Krebsbekämpfung

Bei diesem Lernproblem geht es um die Erforschung von Eigenschaften von Substanzen die eine ähnliche Struktur wie Suramin besitzen und bestimmte Krebsarten bekämpfen können. Dabei sollen solche Substanzen die Bildung von Blutgefäßen für die Tumorzellen verhindern. In der Abbildung 12 ist die Strukturformel von Suramin dargestellt. Suramin ist ein polysulfoniertes Naphtylurea, welches 1925 von der Bayer AG synthetisiert wurde. Es handelt sich um ein für die Zelle sehr toxisches Mittel, das bei geringer Konzentration Parasiten abtöten kann. Kurze Zeit nach der Synthetisierung wurde es als Therapie gegen parasitäre Erkrankungen, insbesondere Trypanosomiasis und Onchozerkose, bei Menschen und Rindern eingesetzt. Dabei beruht der Wirkmechanismus wahrscheinlich auf der Hemmung des Kohlenhydratstoffwechsels des Erregers. In den 1970er und 1980er Jahren wurde zusätzlich seine Wirkung bei malignen Lymphomen, adrenokortikalen Karzinomen, Prostatakarzinomen und anderen malignen Erkrankungen entdeckt. Einige Studien belegen sogar eine Wirksamkeit gegen den HI- Virus. Da Suramin eine Wirksamkeit gegen einige Krankheiten bot, aber sehr toxisch für den Menschen ist, werden in mit Hilfe dieser Wissensbasis Substanzen gesucht, die eine Wirksamkeit gegen die gleichen Krankheit wie Suramin haben, aber nicht so toxisch sind.[RKJH04]

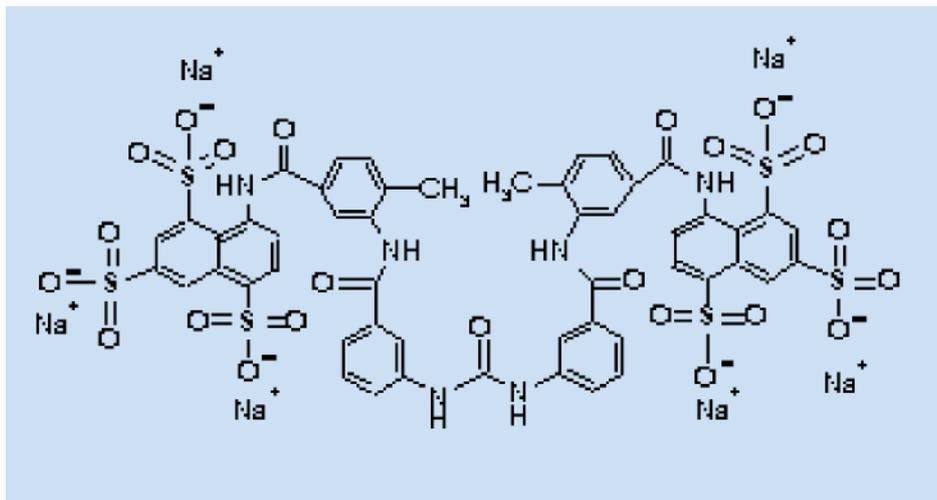


Abbildung 12: Die hier dargestellte Abbildung zeigt die Strukturformel von Suramin. Die entsprechende Summenformel lautet $C_{51}H_{34}N_6O_{23}S_6Na_6$. [RKJH04]

7.1 Datensatz

Dieser Datensatz stammt von der ILP Machine Learning Group Home Page¹⁴ und beinhaltet 17 Instanzdaten. Diese wurden in 7 positive und 10 negative Beispiele eingeteilt. Dabei ist die Wissensbasis als Prolog Datei gespeichert. Die Substanzen werden mit Hilfe von Atom und Atombindungen dargestellt. Die verschiedenen Substanzen werden mit Hilfe von *drug(cpd7)* dargestellt und die Atome werden mit *atomid(cpd0_3)* beschrieben. Des Weiteren werden die Eigenschaften der Atome mit *atm(compound,atom,element,atomtype,charge)* repräsentiert. Das bedeutet, dass das Atom *atom* in der chemischen Verbindung *compound* vom Element *element* des Atomtypen *atomtype* ist und eine Ladung *charge* besitzt. An einem Beispiele dargestellt, bedeutet *atm(d1,d1_20,c,22,-0.117)*, dass das Atom *d1_20* in der Verbindung *d1* ein aromatisches Karbon(22) mit einer Ladung von *-0.117* besitzt. Die Atomverbindungen werden mit Hilfe von *bond(compound,atom1,atom2,bondtype)* dargestellt und bedeuten, dass in der Struktur *compound* eine Verbindung vom Typ *bondtype* zwischen den Atomen *atom1* und *atom2* besteht. An einem Beispiel dargestellt, bedeutet z.B. *bond(d1,d1_1,d1_2,7)*, dass eine aromatische Verbindung(7) zwischen den Atomen *d1_1* und *d1_2* in der Struktur *d1* vorliegt. Im folgenden Abschnitt wird die Umwandlung der Prolog Datei in eine OWL-Ontologie beschrieben.

Umwandlung in eine OWL Ontologie

Die Wissensbasis für Suramin-ähnliche Substanzen lag in Prolog vor. Da der DL-Learner nicht mit Prolog Dateien umgehen kann, musste die Wissensbasis erst einmal in eine OWL Ontologie konvertiert werden. Es wurde der Prolog Parser von Sebastian Bader verwendet, welcher in Java geschrieben wurde. Dazu musste eine Javaklasse geschrieben werden, welche die Prologklauseln in OWL Ausdrücke umwandelt. In der Abbildung 13 wird der Ablauf der Konvertierung in eine OWL-Ontologie mit den einzelnen Methoden dargestellt. Im Folgenden wird der Ablauf der Umwandlung und die dazu nötigen Methoden näher beschrieben. Für die Steuerung des Ablaufes der Umwandlung ist die Methode *main()* zuständig. In einem ersten Schritt werden für die jeweiligen chemischen Elemente die Namen mit Hilfe der Methode *createChemElementsMapping()* übergeben. Im Anschluss daran wird die komplette Prologdatei eingelesen und mit Hilfe des Prolog Parsers geparkt. Dabei wird überprüft ob in der Prologdatei nur korrekte Prologausdrücke stehen. Danach wird eine neue Wissensbasis angelegt und die einzelnen chemischen Elemente als Subklassen von *Atom* deklariert. Des Weiteren werden für alle Object und Data Properties die Domain und Range festgelegt. In einem nächsten Schritt werden die Prolog Klauseln mit Hilfe der Methode *mapClause()* in die entsprechenden OWL Ausdrücke umgewandelt. Dazu werden die Methoden *getRoleAssertion()* verwendet, die für die Erzeugung der Object Properties, *getConceptAssertion()*, welche für die einzelnen Instanzdaten, die

¹⁴Quelle:<http://www.doc.ic.ac.uk/~shm/suramin.html> Stand:26. April 2011

zugehörige Klasse festlegt und *getDoubleDatatypePropertyAssertion()*, die die Data Properties, welche ein Double Wert besitzen festlegt. Des Weiteren wird die Methode *getAtomClass()* benutzt, um die Atome den richtigen Atomklassen zuzuordnen. Nach Abschluss der Umwandlung wird die Konfigurationsdatei für den DL-Learner geschrieben. Dies geschieht in zwei Schritten. Im ersten Schritt wird der Kopf der Konfigurationsdatei, mit Informationen über Lernalgorithmus, Lernproblem und verwendeten Reasoner geschrieben. Im zweiten Schritt werden mit Hilfe der Methode *appendExamples()*, die positiven und negativen Beispiele an die Konfigurationsdatei angehängt. Damit ist die Umwandlung der Prolog Datei in eine OWL Ontologie abgeschlossen und es ist eine Ontologie mit 46 Klassen, 3 Object Properties, 1 Data Property und 2979 Instanzen entstanden.

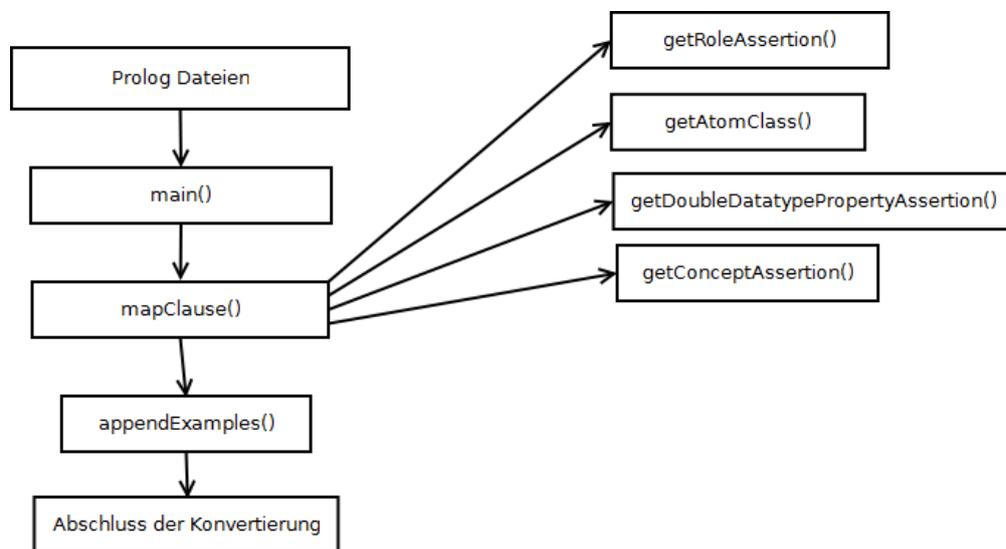


Abbildung 13: Hier ist der Ablauf zur Konvertierung der Prologdateien der Suramin Wissensbasis in eine OWL Ontologie mit ihren Methoden dargestellt.

7.2 Methoden

In diesem Abschnitt werden die Tests, welche durchgeführt wurden, näher erläutert. In einem ersten Test wurde die Ontologie mit Hilfe des Lernalgorithmus OCEL überprüft. Dazu wurde der Noise Parameter des Lernalgorithmus von 30%, in 1% Schritten gesenkt und eine 10-fold-cross-validation durchgeführt, bis keine Ergebnisse mehr produziert wurden. Anschließend wurde die gleiche Pro-

zedur mit dem Lernalgorithmus CELOE wiederholt. Des Weiteren wurde mit Hilfe von Aleph eine 10-fold-cross-validation durchgeführt und auch hier der Noise Parameter von 30%, in 1% Schritten gesenkt. Die daraus resultierenden Ergebnisse wurden miteinander verglichen. Des Weiteren wurde die Laufzeit von OCEL und CELOE aus den 10-fold-cross-validations für den Noise Parameter genutzt um einen weiteren Vergleich zwischen den beiden Algorithmen durchzuführen. In einem zweiten Test wurde für die Lernalgorithmen OCEL und CELOE eine feste Laufzeit von 10, 100, 300 und 1000 eingestellt und eine 10-fold-cross validation durchgeführt. Da es für Aleph keine Möglichkeit gibt den Laufzeit Parameter einzustellen, konnte der Test für Aleph nicht durchgeführt werden. Da der Lernalgorithmus CELOE einen höheren β Parameter aufweist, als OCEL, um längere Konzepte höher zu bestrafen, wurde der β Parameter von CELOE auf 0,02 gesetzt, wie es auch bei OCEL eingestellt ist. Des Weiteren wurden alle Tests auf einem Quad Core mit 3,2Ghz und 8GB Ram durchgeführt. Diese Maßnahmen dienen der besseren Vergleichbarkeit der Algorithmen, insbesondere dem Vergleich der Laufzeiten. Im folgenden Abschnitt werden die Ergebnisse genauer vorgestellt.

7.3 Resultate und Schlussfolgerungen

In Abbildung 14 werden die Ergebnisse des Lernvorgangs für Suramin-ähnliche Substanzen dargestellt. Dabei ist zu sehen, dass der Lernalgorithmus OCEL die höchste Genauigkeit für die Testdaten ausweist. Der Abfall der Testgenauigkeit ab 24% lässt sich damit erklären, dass OCEL mit Hilfe der Trainingsdaten Konzepte lernt, die für die Trainingsdaten gut geeignet sind, aber nicht zu den Testdaten passen oder zu speziell sind. Des Weiteren ist zu sehen, dass Aleph die geringste Testgenauigkeit erreicht. Außerdem ist festzustellen, dass CELOE nur bis 24% Noise Ergebnisse findet. Dies lässt sich damit erklären, dass CELOE bei 23% Noise keine Konzepte mehr findet, die von dem Algorithmus mit dieser Genauigkeit bewertet werden. Aus diesem Grund gibt es bei dem Algorithmus CELOE ab 23% keine Ergebnisse mehr.

In der Tabelle 9 sind die höchsten Testgenauigkeiten der einzelnen Algorithmen OCEL, CELOE und Aleph mit den entsprechenden Laufzeiten und Konzeptlängen dargestellt. Dazu ist anzumerken, dass bei Aleph die Laufzeit und Konzeptlänge nicht mitprotokolliert wurde. Dabei ist zu sehen, dass OCEL die höchste Genauigkeit erreicht, dicht gefolgt von CELOE. Dabei benötigen die beiden Algorithmen ungefähr die gleiche Laufzeit. Nur die durchschnittliche Konzeptlänge ist mit 7 bei OCEL und 10 bei CELOE unterschiedlich. Die geringste Testgenauigkeit erreicht Aleph.

Um herauszufinden ob diese Unterschiede der einzelnen Lernalgorithmen aus der Tabelle 9 statistisch Signifikant sind, wurde der t-Test durchgeführt. Die Ergebnisse des t-Tests wurden in Tabelle 10 dargestellt. Dabei ist deutlich zu sehen, dass trotz der Unterschiede in den Genauigkeiten der Algorithmen, keine statistische Signifikanz vorliegt.

In der Tabelle 11 werden die Ergebnisse, die die höchste Genauigkeit besitzen, mit ihren Laufzeiten,

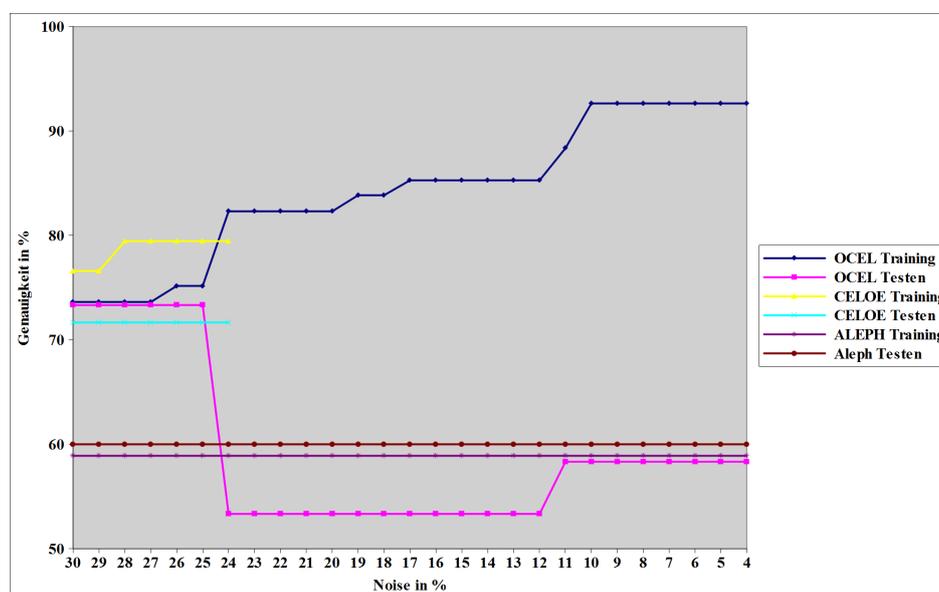


Abbildung 14: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Suramin Wissensbasis. Der Abfall der Genauigkeit bei OCEL - und CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind.

den Konzeptlängen und der erreichten Genauigkeit dargestellt. Dazu ist anzumerken, dass bei Aleph die Laufzeit und die Konzeptlänge nicht mitprotokolliert wurde. Im folgenden werden die einzelnen Konzepte erklärt. Das Konzept, welches OCEL vorgeschlagen hat bedeutet, dass eine Substanz, welche mindestens ein Hydrogen-8 Atom und mehr als 5 Oxygen-45 Atome besitzt, eine Substanz ist, welche ähnliche Eigenschaften besitzt wie Suramin. Das Konzept welches CELOE vorgeschlagen hat bedeutet, dass eine Substanz, welche mehr als 4 Nitrogen-32 Atome und mehr als 4 Oxygen-45 Atome besitzt und eine Ladung die größer gleich -0.6295 ist, Suramin-ähnliche Eigenschaften besitzt. Im Gegensatz dazu findet Aleph keine Regeln, um ein Konzept für Suramin-ähnliche Substanzen vorschlagen zu können. Aus diesem Grund schlägt Aleph die Substanz *cpd8* vor, die Eigenschaften besitzt, welche von einigen Testdaten abgedeckt werden. Weitere Vorschläge liefert Aleph auch bei Verringerung des Noise-Parameters nicht. Um herauszufinden, warum OCEL und CELOE unterschiedliche Ergebnisse liefern wurde bei beiden Algorithmen der Suchbaum ausgegeben. Die Auswertung des Suchbaumes hat ergeben, dass bei CELOE das Teilergebnis *hasAtom.Hydrogen-8* gar nicht im Suchbaum vor kam und somit auch nicht betrachtet wurde. Dieses erschien aber bei allen Ergebnissen der 5-fold-cross-validation bei OCEL als Teilstück vor. Des Weiteren traten bei OCEL zwar die Teilkonzepte *hasAtom.Nitrogen-32* und *hasAtom.Oxygen-45* auf, jedoch nicht zusammen. Das Teilergebnis *hasAtom.(Oxygen-45 AND charge >= -0.6295)* war auch nicht in dem Suchbaum

Algorithmus	Testgenauigkeit	Laufzeit	Konzeptlänge
OCEL	73,333%±30,845%	0,695s±0,502s	7±2,739
CELOE	71,667%±18,257%	0,773s±0,842s	10±1.732
Aleph	58,888%±2,154%	–	–

Tabelle 9: In dieser Tabelle werden die höchsten Genauigkeiten und die dazu entsprechenden Laufzeiten und Konzeptlängen der 10-fold-cross-validation der Algorithmen OCEL, CELOE und Aleph, für die Wissensbasis Suramin dargestellt. Dabei ist zu sehen, dass OCEL die höchste Genauigkeit erreicht und Aleph die geringste.

Algorithmen	p-Wert	statistische Signifikanz
OCEL vs. Aleph	0.3267	nein
CELOE vs. Aleph	0.1587	nein
CELOE vs. OCEL	0.9198	nein

Tabelle 10: In dieser Tabelle wird die statistische Signifikanz der Algorithmen OCEL, CELOE und Aleph, für die Wissensbasis Suramin dargestellt. Dabei ist zu sehen, dass die Genauigkeitsunterschiede für die Algorithmen nicht statistisch Signifikant sind.

von OCEL vorhanden, nur ein ähnliches Konzept, nämlich (*EXISTS hasAtom.Oxygen-45 AND ALL hasAtom.(charge >= -0.6295 AND (NOT Carbon-10))*). Dieses wurde aber als zu Schwach klassifiziert und nicht weiter verfolgt. In 3 von 5 folds der 5-fold-cross-validation kamen beide Teilergebnisse zusammen vor und in den verbleibenden 2 folds traten die Teilkonzepte *hasAtom.Nitrogen-32* und *hasAtom.Oxygen-45* in jeweils einem Konzept auf. Da diese Teilkonzepte von OCEL als zu Schwach klassifiziert wurden, konnten diese nicht weiter ausgebaut werden und kamen somit auch nicht in den Ergebnissen von OCEL vor. Aus diesen Unterschieden in den Suchbäumen, können sich die unterschiedlichen Konzepte und Genauigkeiten der beiden Lernalgorithmen ergeben.

In der Abbildung 15 wird die Laufzeit der Algorithmen OCEL und CELOE in Bezug auf den eingestellten Noise Parameter dargestellt. Dabei ist zu sehen, dass OCEL und CELOE am Anfang ungefähr die gleiche Zeit benötigen um Konzepte zu finden die den Anforderungen von (1 - Noise) genügen. Diese Laufzeit beträgt bei beiden Algorithmen ungefähr 0.7 Sekunden. Weiterhin ist zu sehen, dass CELOE ab 23% keine Ergebnisse mehr produziert die diesen Ansprüchen genügen und aus diesem Grund wird die Linie für CELOE auch nur bis 24% angezeigt. Die höchste Laufzeit die bei CELOE gemessen wurde war bei 24% und betrug 0,773s±0,842s. Der Anstieg der Laufzeit bei OCEL ab 11% lässt sich damit erklären, dass die Konzepte die eine Genauigkeit größer gleich (1 - Noise) eine längere Zeit brauchen um gefunden zu werden. Dabei traten einige Schwankungen beim finden des Konzeptes für Fold 2 auf. Es wurde zwar immer das gleiche Konzept nämlich (*Compound AND*

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Compound AND EXISTS hasAtom.Hydrogen-8 AND >= 5 hasAtom.Oxygen-45)</i>	100%	0,68s	10
CELOE	<i>(>= 4 hasAtom.Nitrogen-32 AND >= 4 hasAtom.(Oxygen-45 AND charge >= -0.6295))</i>	100%	0,526s	13
Aleph	active(cpd8)	66,667%	–	–

Tabelle 11: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL, CELOE und Aleph für die Wissensbasis Suramin dargestellt. Dabei ist deutlich zu sehen, dass alle drei Algorithmen Ergebnisse mit einer Genauigkeit von 100% vorschlagen.

>= 4 hasAtom.Carbon-27 AND >= 4 hasAtom.Oxygen-45 AND >= 5 hasAtom.(Hydrogen-3 AND charge <= 0.178)) gelernt, aber die Laufzeit um dieses zu finden schwankte dabei zwischen 1800s und 2100s. Diese Schwankungen lassen sich dadurch erklären, dass der DL-Learner im Suchbaum Konzepte hat, die bei geringer eingestelltem Noise-Parameter als zu Schwach bewertet werden und somit erst einmal andere Konzepte gefunden werden müssen, die mit einer ausreichenden internen Genauigkeit bewertet werden. Da dies unter Umständen länger dauern kann erhöht sich die Laufzeit dadurch. Es kann aber auch dazu kommen, dass wenn ein Konzept im Suchbaum als zu Schwach bewertet wurde, der Algorithmus ein anderes findet, welches beim Weitersuchen schneller eine Lösung findet, die dem eingestellten Noise-Parameter genügt und somit die Laufzeit verkürzt wird. Aus diesen Gründen kommt es zu Schwankungen in der Laufzeit. Die höchste gemessene Laufzeit bei OCEL wurde bei einem Noise-Wert von 9% gemessen und beträgt $558,983s \pm 923,606s$.

Die in der Abbildung 16 dargestellt Grafik zeigt die Genauigkeit der Trainings- und Testdaten der beiden Lernalgorithmen OCEL und CELOE in Abhängigkeit der eingestellten Laufzeit. Deutlich zu erkennen ist, dass die Testgenauigkeit von 58% bei allen eingestellten Laufzeiten gleich bleibt. Dies lässt sich damit erklären, dass die gefundenen Konzepte bei allen eingestellten Laufzeiten keine bessere Genauigkeit bei den Testdaten erzielt haben. Des Weiteren lässt sich der Unterschied zur Abbildung 14, wo die Genauigkeit der beiden Algorithmen höher war, damit erklären, dass der Algorithmus beim Lernen mit fester Laufzeit nur Konzepte findet welche 58% Genauigkeit haben. Da die Ergebnisse mit 70% Testgenauigkeit auch innerhalb kürzerer Zeit gelernt wurden kann es sein, dass durch zu die zusätzliche Laufzeit nur noch Konzepte errechnet wurden, die eine geringere Testgenauigkeit besitzen. Lediglich eine Verbesserung der Trainingsgenauigkeit zwischen 10 Sekunden und 100 Sekunden ist festzustellen. Dabei steigt sowohl die Genauigkeit bei OCEL als auch bei CELOE. Die Trainingsgenauigkeit bei OCEL steigt von 86% auf 94% und die Trainingsgenauigkeit von CELOE steigt von

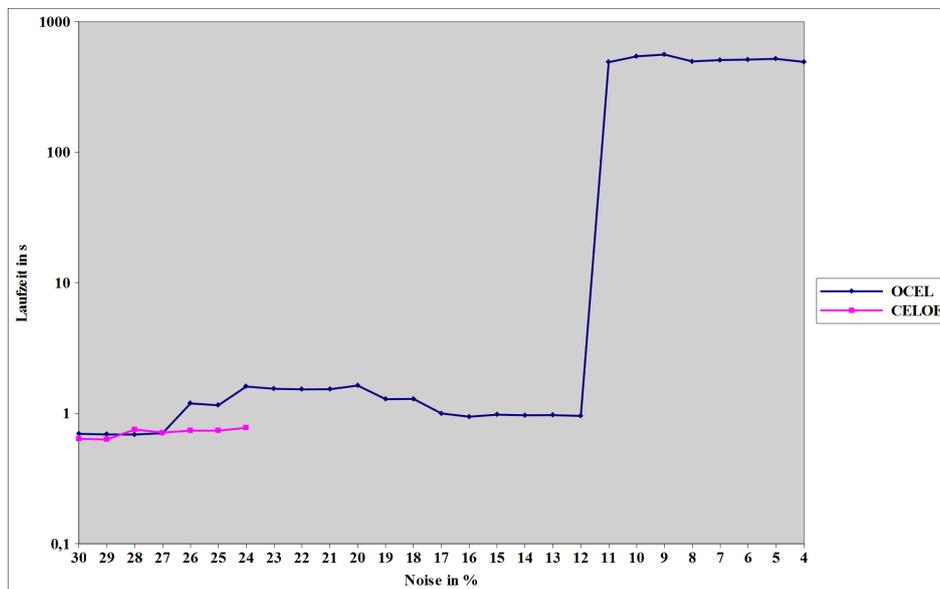


Abbildung 15: Die hier dargestellte Grafik zeigt die Laufzeit der Algorithmen OCEL und CELOE in Bezug auf den eingestellten Noise Parameter für die Suramin Wissensbasis. Zu erkennen ist, dass die Laufzeit beider Algorithmen am Anfang gleich ist. Da CELOE ab 23% keine Ergebnisse mehr produzierte geht die Anzeige nur bis 24%. Der Anstieg der Laufzeit bei OCEL ab 11% lässt sich damit erklären, dass der Lernalgorithmus mehr Zeit benötigt um Konzepte zu finden die eine Genauigkeit der Testdaten von mindestens $(1 - \text{Noise})$ haben, also in dem Fall von mindestens 89%.

79% auf 88%. Dies lässt sich damit erklären, dass die beiden Algorithmen durch die zusätzliche Zeit die sie zur Verfügung haben, weitere Konzepte finden die besser auf die Trainingsdaten passen. Da kein weiterer Anstieg der Trainingsgenauigkeit durch weitere Laufzeiterhöhung mehr erzielt wurde, kann daran liegen, dass die beiden Lernalgorithmen keine Konzepte mehr finden die besser auf die Trainingsdaten passen um somit eine Genauigkeitserhöhung zu erzielen. Da bei allen Laufzeiten die Testdaten die gleiche Genauigkeit aufweisen wurde kein t-Test durchgeführt.

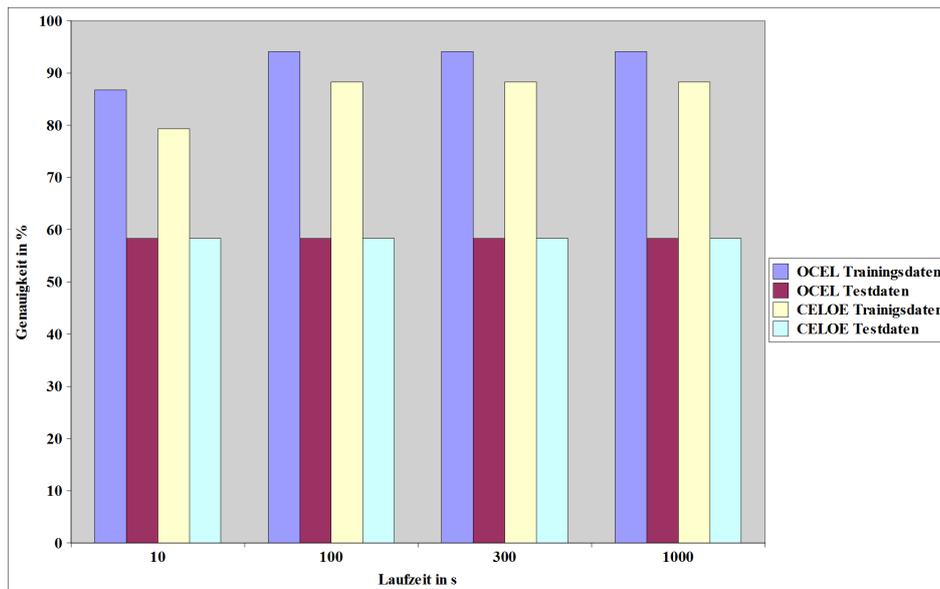


Abbildung 16: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Suramin Wissensbasis. Dabei ist zu erkennen, dass die Trainingsgenauigkeit bei OCEL und CELOE von 10 Sekunden auf 100 Sekunden Laufzeit, steigt, aber in allen anderen Fällen gleich bleibt. Die Testgenauigkeit bei OCEL und CELOE bleibt bei allen Laufzeiten gleich.

8 Monitoring von Parkinsonpatienten

Die in Kapitel 3.1 eingeführten Diagnosekriterien, werden meist erst später bemerkt und sind außerdem nicht eindeutig. Aus diesem Grund soll mit Hilfe dieser Wissensbasis, die Sprachaufnahmen von Personen analysiert werden, um die die an Parkinson leiden zu Unterscheiden. Es geht also um die Klassifikation der Patienten, ob diese an Parkinson leiden oder nicht, anhand einer Reihe von Sprachaufnahmen.[LMR⁺07]

8.1 Datensatz

Der Datensatz stammt aus dem UCI Machine Learning Repository¹⁵ und besteht aus 195 Instanzdaten, welche von 31 Personen stammen, von denen 23 an der Parkinsonkrankheit leiden. Von diesen 197 Aufnahmen waren 48 von Parkinsonpatienten und die verbleibenden 147 Aufnahmen waren von gesunden Personen. Die zur genauen Klassifikation verwendeten Attribute war 23, wobei das Attribut *status* zur Klassifikation diente, ob die Aufnahme von einer gesunden oder unter Parkinson leidenden Person stammte. Die restlichen 22 Attribute, welche hauptsächlich Informationen über die Frequenz und Amplitude der Stimme enthalten, werden für die genaue Klassifikation verwendet. Das Wissen wurde mit Hilfe einer Flat file Database dargestellt, die alle Werte hintereinander durch ein Komma getrennt darstellt. Die einzelnen Werte werden im folgenden genauer beschrieben.

- MDVP:Fo(Hz)

Das Attribut MDVP:Fo(Hz) beschreibt die durchschnittliche Grundfrequenz der Stimme in Hz.

- MDVP:Fhi(Hz)

Hierbei handelt es sich um die maximale Grundfrequenz der Stimme in Hz.

- MDVP:Flo(Hz)

Die Eigenschaft MDVP:Flo(Hz) stellt die minimale Grundfrequenz der Stimme in Hz dar.

- MDVP:Jitter(%), MDVP:Jitter(Abs), MDVP:RAP, MDVP:PPQ und Jitter:DDP

Bei diesen Attributen handelt es sich um verschiedene Maße, die die Veränderung in der Grundfrequenz der Stimme darstellen.

- MDVP:Shimmer, MDVP:Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ und Shimmer:DDA

Diese Eigenschaften beschreiben verschiedene Maße, die die Veränderung in der Amplitude der Stimme beschreiben.

¹⁵Quelle:<http://archive.ics.uci.edu/ml/datasets/Parkinsons> Stand:26. April 2011

- NHR und HNR

Diese zwei Attribute beschreiben das Verhältnis zwischen Lautstärke und anderen tonalen Komponenten in der Stimme.

- status

Die Eigenschaft status beschreibt den Gesundheitsstatus des Patienten. Dabei bedeutet 1 das der Patient Parkinson hat und 0 das der Patient gesund ist.

- RPDE und D2

Diese Eigenschaften beschreiben nicht lineare dynamische Komplexitätsmaße.

- DFA

Dieses Attribut beschreibt den fraktalen Skalierungsexponenten des Signals.

- spread1, spread2 und PPE

Bei diesen 3 Attributen handelt es sich um nicht lineare Maße, die die Grundfrequenzveränderung der Stimme beschreiben.

Umwandlung in eine OWL Ontologie

Da diese Daten ebenfalls nicht als OWL-Ontologie vorlagen, wurden die Daten mit Hilfe eines eigens geschriebenen Parsers in eine Ontologie umgewandelt. In der Abbildung 17 sieht man den Ablauf zur Konvertierung der Parkinson Flat File Database in eine OWL Ontologie mit ihren einzelnen Methoden. Es folgt eine Beschreibung des Ablaufes der Umwandlungen mit den dafür verantwortlichen Methoden. In der *main()* Methode wird der gesamte Ablauf der Konvertierung gesteuert. Dabei wird zuerst eine neue Ontologie erstellt und mit der Methode *generateDomainAndRangeForObjectProperties()* werden alle Domains und Ranges für die Object- und Data Properties festgelegt. Anschließend wird die Datei zeilenweise eingelesen und die einzelnen Attribute werden an die *mapClauses()* Methode übergeben. In dieser werden alle Klassen, Data- und Object Properties mit den Methoden *getConceptAssertion()*, *getDoubleDatatypePropertyAssertion()* angelegt. Im Anschluss daran werden die so erzeugten Daten in die Ontologie geschrieben und die Konfigurationsdatei mit Hilfe der Methoden *generateConfFile()* und *generateExamples()* erzeugt. Die erste Methode schreibt nur den Kopf der Konfigurationsdatei, welcher Informationen über Lernalgorithmus, Lernproblem und verwendeten Reasoner enthält und die zweite Methode fügt die positiven und negativen Beispiele ein. Nach diesem Schritt ist die Umwandlung abgeschlossen. Dabei ist eine Ontologie mit 1 Klasse, 22 Data Properties und 195 Instanzdaten entstanden.

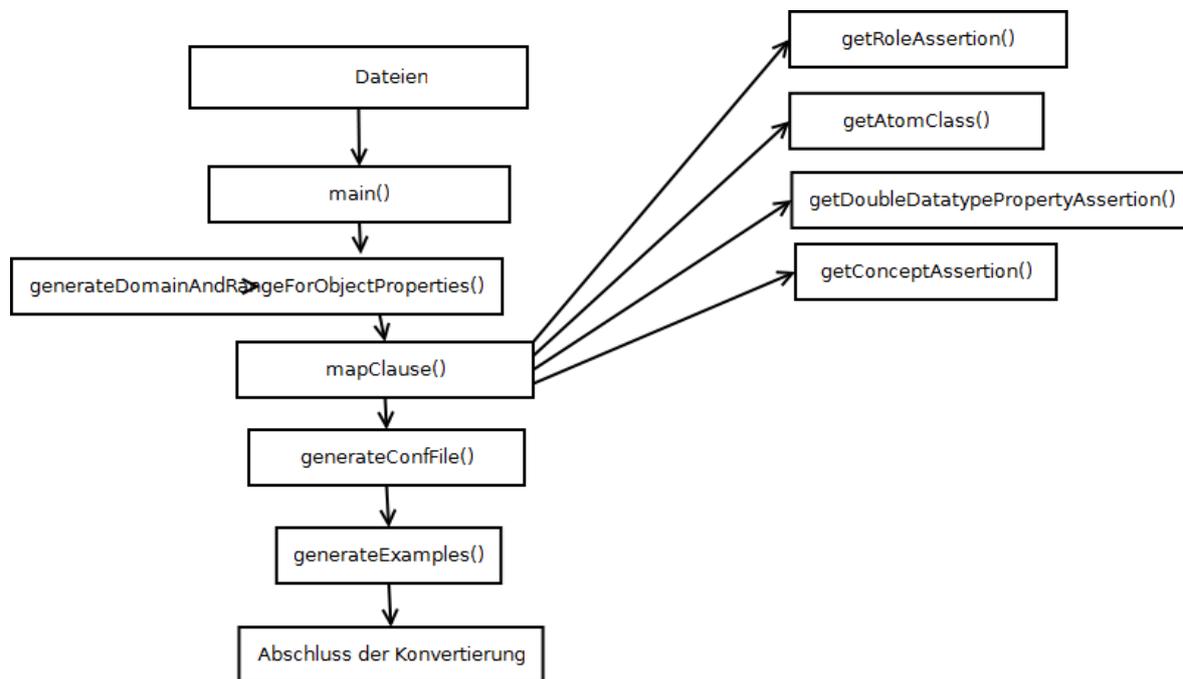


Abbildung 17: Hier ist der Ablauf zur Konvertierung der Prologdateien der Parkinsons Wissensbasis in eine OWL Ontologie mit ihren Methoden dargestellt.

8.2 Methoden

Dieser Abschnitt befasst sich mit den durchgeführten Experimenten. Dabei wurde die Ontologie in einem ersten Schritt mit Hilfe des Lernalgorithmus OCEL überprüft. Es wurde der Noise Parameter des Algorithmus von 30%, in 1% Schritten reduziert und eine 10-fold-cross-validation durchgeführt, bis keine Ergebnisse mehr erzeugt wurden. Anschließend wurde der Vorgang mit dem Lernalgorithmus CELOE noch einmal ausgeführt. Die dadurch entstandenen Ergebnisse wurden mit den Algorithmen naive Bayes, naive Bayes Updateable, random Tree, random Forrest, PART und Decision Table des Machine Learning Werkzeugs Weka verglichen. Hierzu wurde ebenfalls eine 10-fold-cross-validation durchgeführt. Da es aber keine Einstellmöglichkeit für den Noise-Parameter für die Algorithmen von Weka gab, wurde der Test nur einmal durchgeführt. In einem nächsten Schritt wurde der Laufzeitparameter von OCEL und CELOE auf 10, 100, 300 und 1000 Sekunden fest gesetzt und mit Hilfe einer 10-fold-cross-validation überprüft, um einen weiteren Vergleich der beiden Lernalgorithmen durchzuführen. Da Weka keine Einstellung einer festen Laufzeit bot, konnte der Vergleich nur mit den Lernalgorithmen des DL-Learners durchgeführt werden. Im letzten Schritt wurde die Laufzeit der beiden Algorithmen OCEL und CELOE verglichen. Um ein möglichst gleiches Verhalten der beiden Algorithmen OCEL und CELOE für den Vergleich zu erzielen wurde der β Parameter von CELOE, wie bei OCEL, auf 0.02 gesetzt. Dieser Parameter bestraft längere Konzepte. Um ebenfalls eine bessere Vergleichbarkeit zu erzielen wurden alle Tests auf einem Quad Core mit 3.2 GHz und 8Gb Ram durchgeführt. Im nächsten Abschnitt werden alle Ergebnisse genauer vorgestellt.

8.3 Resultate und Schlussfolgerungen

Die in Abbildung 18 dargestellte Grafik zeigt die Genauigkeit der Lernalgorithmen im Verhältnis zu dem eingestellten Noise Parameter. Da es bei den Algorithmen von Weka keine Einstellung für den Noise Parameter gab wurde nur eine 10-fold-cross-validation durchgeführt. Dieses Ergebnis wurde mit Hilfe einer Geraden in der Grafik dargestellt. Das Absinken der Testgenauigkeit bei OCEL und CELOE lässt sich damit erklären, dass die gelernten Konzepte zwar die Anforderung an die Trainingsgenauigkeit von $(1 - \text{Noise})$ erfüllt hatten, also eine Trainingsgenauigkeit von mindestens $(1 - \text{Noise})$ besaßen, aber für die Testdaten zu spezifisch waren. Das bei OCEL die Lernergebnisse nur bis zu einem Noise Wert von 8% gehen liegt daran, dass beim Lernen mit einem Noise Wert von 7% eine *OutOfMemoryException* auftrat und somit keine Ergebnisse mehr erzeugt wurden. Deutlich zu sehen ist, dass die Testgenauigkeit sowohl bei OCEL als auch bei CELOE am höchsten ist.

In der Tabelle 12 werden die Ergebnisse der 10-fold-cross-validation dargestellt. Dabei handelt es sich um die höchsten Testgenauigkeiten und die dazu benötigten Laufzeiten und Konzeptlängen. Dabei ist deutlich zu erkennen, dass die beiden Algorithmen des DL-Learners die höchsten Testgenauigkeiten

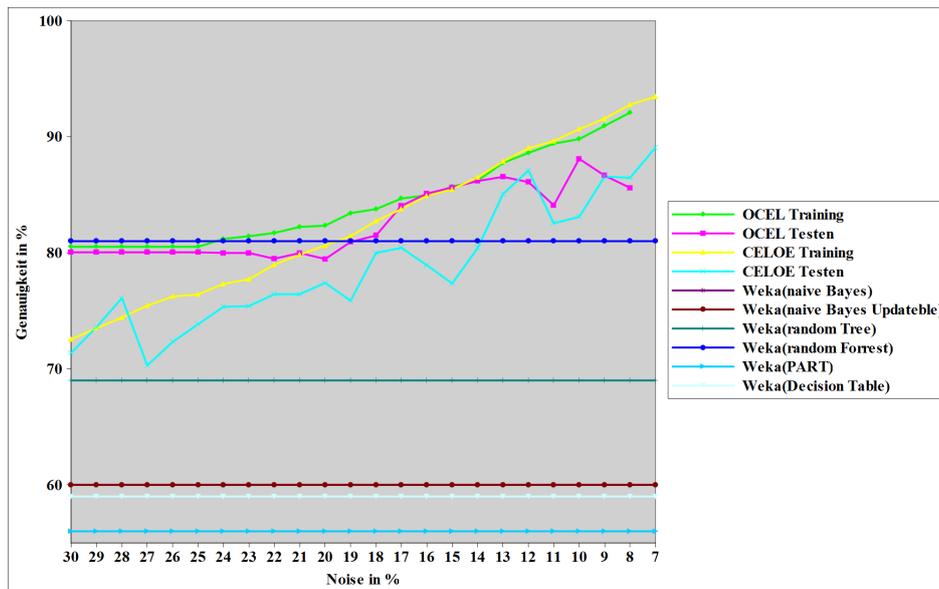


Abbildung 18: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Parkinson Wissensbasis. Der Abfall der Genauigkeit bei OCEL - und CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind. Ebenfalls zu erkennen ist, dass alle Ergebnisse bei Weka durch eine Gerade dargestellt wurde. Dies liegt daran, dass bei Weka kein Noise Parameter zur Verfügung stand. Da Naive Bayes und Naive Bayes Updateble die gleiche Genauigkeit erzielt haben liegen die beiden Geraden übereinander so dass die Gerade von Naive Bayes nicht sichtbar ist.

erreichen und die Algorithmen von Weka, bis auf den Algorithmus Random Forrest, eine viel geringere Testgenauigkeit erzielen. Des Weiteren ist zu erkennen, dass die durchschnittlichen Laufzeiten von Weka viel geringer sind als die der beiden Algorithmen OCEL und CELOE. Dies lässt sich dadurch erklären, dass die Algorithmen von Weka, im Gegensatz zu den Algorithmen des DL-Learners, keinen Reasoner benötigen um ein Konzept zu errechnen. Des Weiteren sind die Algorithmen auf das Eingabeformat, nämlich die ARFF Datei, so dass die Berechnung der Konzepte gleich durchgeführt werden kann. Außerdem sind die Algorithmen auf numerische Werte spezialisiert, so dass diese viel schneller zu Ergebnissen kommen.

Algorithmus	Testgenauigkeit	Laufzeit	Konzeptlänge
OCEL	88,096%±6,139%	2,393s±4,334s	15±6,325
CELOE	89,067%±8,196%	86,528s±95,553s	20,2±2,7
Naive Bayes	60,17%±12,89%	0,0008s±0,0008s	–
Naive Bayes Updateable	60,17%±12,89%	0,0008s±0,0008s	–
Random Tree	69,15%±15,16%	0,0013s±0,0005s	–
Random Forrest	81,11%±12,75%	0,0114s±0,0005s	–
PART	56,28%±20,94%	0.0093s±0,0021s	–
Decision Table	58,95%±21,60%	0,0268s±0,0053s	–

Tabelle 12: In dieser Tabelle werden die höchsten Genauigkeiten und die dazu entsprechenden Laufzeiten und Konzeptlängen der 10-fold-cross-validation der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für die Wissensbasis Parkinsons dargestellt. Dabei ist zu sehen, dass die beiden Algorithmen des DL-Learner die besten Testgenauigkeiten erreichen und die Algorithmen von Weka, bis auf Random Forrest eine viel geringere Testgenauigkeit besitzen. Des Weiteren ist zu erkennen, dass die Algorithmen von Weka eine deutlich geringere Laufzeit benötigen als die Algorithmen des DL-Learners.

Um herauszufinden ob die in der Tabelle 12 dargestellten Ergebnisse statistisch signifikant sind, wurde für alle Algorithmen der t-Test durchgeführt. In der Tabelle 13 sind die Ergebnisse dieses Tests dargestellt, mit p-Werten und Aussage ob dieser Wert statistische Signifikanz besitzt. Dabei ist zu erkennen, dass die beiden Algorithmen OCEL und CELOE in fünf von sechs Fällen statistisch Signifikant bessere Ergebnisse liefern als die Algorithmen von Weka, nur bei dem Algorithmus Random Forrest ist keine statistische Signifikanz nachweisbar. Bei einem Vergleich zwischen OCEL und CELOE konnte ebenfalls keine statistische Signifikanz nachgewiesen werden.

In der folgenden Tabelle 14 werden die Ergebnisse mit den höchsten Genauigkeiten der Algorithmen OCEL und CELOE dargestellt. Bei Weka wurden keine Konzepte angezeigt, die als beste Lösung des Lernproblems dienten. Im folgenden werden diese vorgeschlagenen Konzepte erläutert. Das vorgeschlagene Konzept von OCEL bedeutet, dass ein Patient mit einem Spread1 Wert kleiner gleich

Algorithmen	p-Wert	statistische Signifikanz
OCEL vs. Naive Bayes	0,0001	ja
OCEL vs. Naive Bayes Updateable	0,0001	ja
OCEL vs. Random Tree	0,0018	ja
OCEL vs. Random Forrest	0,1359	nein
OCEL vs. PART	0,0002	ja
OCEL vs. Decision Table	0,0007	ja
CELOE vs. Naive Bayes	0,0001	ja
CELOE vs. Naive Bayes Updateable	0,0001	ja
CELOE vs. Random Tree	0,0018	ja
CELOE vs. Random Forrest	0,1142	nein
CELOE vs. PART	0,0002	ja
CELOE vs. Decision Table	0,0006	ja
CELOE vs. OCEL	0,7677	nein

Tabelle 13: In dieser Tabelle wird die statistische Signifikanz der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für die Wissensbasis Parkinsons dargestellt. Dabei ist zu erkennen, dass sowohl OCEL als auch CELOE in fünf von sechs Fällen statistisch signifikant bessere Ergebnisse liefert als die Algorithmen von Weka, nur Random Forrest liefert keine statistisch signifikant schlechteren Ergebnisse. Auch bei einem Vergleich zwischen OCEL und CELOE konnte keine statistische Signifikanz in den Unterschieden der Genauigkeit festgestellt werden.

-5,8680535 und einen MDVPFo Wert kleiner gleich 143,131 oder einem MDVPFo wert größer gleich 167,409 und einem Spread2 Wert kleiner gleich 0,2103685 an Parkinson erkrankt ist. Dem hingegen bedeutet das von CELOE vorgeschlagene Konzept, dass ein Patient mit einem MDVPAPQ Wert kleiner gleich 0,019525 und einem MDVPFhi Wert größer gleich 206,452 und einem D2 Wert der kleiner gleich 2,2291955 oder einem Spread2 Wert kleiner gleich 0,1780455 besitzt, an Parkinson erkrankt ist. Da es Unterschiede in der Genauigkeit zwischen OCEL und CELOE gab, wurde für beide Algorithmen der Suchbaum für den Noise-Parameter ausgegeben, welcher die höchste Testgenauigkeit erzielt hat. Dabei ist zu erkennen, dass die Algorithmen OCEL und CELOE das Konzept *Patient AND hasMDVPAPQ <= 0.03612500000000004* mit einer unterschiedlichen internen Genauigkeit versehen. Auch bei der Weiterverfolgung dieses Konzeptes, bei dem sich nur der Wert für *hasMDVPAPQ* verändert hat, wird mit unterschiedlichen Genauigkeiten versehen. Da die Algorithmen also schon zu Beginn die Konzepte unterschiedlich bewerten, wird der gesamte Suchbaum unterschiedlich aufgebaut. Da die unterschiedliche Bewertung der gefundenen Konzepte dazu führt das andere Knoten im Suchbaum erweitert werden um weiterzusuchen, kommt es zu unterschiedlichen Ergebnissen in der

Tabelle 14

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	$(Patient \text{ AND } hasMDVPAPQ \leq 0.019525 \text{ AND } hasMDVPFhi \geq 206.452 \text{ AND } (hasD2 \leq 2.2291955 \text{ OR } hasSpread2 \leq 0.1780455))$	100%	0,772s	17
CELOE	$(hasSpread1 \leq -5.8680535 \text{ AND } (hasMDVPFo \leq 143.131 \text{ OR } (hasMDVPFo \geq 167.409 \text{ AND } hasSpread2 \leq 0.2103685)))$	100%	6,997s	15

Tabelle 14: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL, CELOE und Weka für die Wissensbasis Parkinsons dargestellt.

In der Abbildung 19 wird die Laufzeit der Algorithmen OCEL und CELOE in Abhängigkeit des eingestellten Noise Parameters dargestellt. Dabei ist deutlich zu sehen, dass beide Algorithmen am Anfang ungefähr gleich lange brauchen um Konzepte mit dem geforderten Noise Wert zu lernen. Ab ca. 25% Noise steigt die Laufzeit für CELOE stärker an als bei OCEL. Dies lässt sich dadurch erklären, dass der Algorithmus mehr Konzepte durchsuchen muss, um ein geeignetes Konzept mit entsprechendem Noise-Wert zu finden. Der Abfall der Laufzeit bei 10% Noise bei OCEL lässt sich dadurch erklären, dass der Algorithmus im Suchbaum ein Konzept welches vorher ausreichend gut bewertet wurde, jetzt als zu Schwach definiert wurde. Aus diesem Grund muss er einen anderen Knoten im Suchbaum erweitern, so dass dadurch besser bewertete Konzepte in einer kürzeren Zeit gefunden werden können. Damit lässt sich der Abfall der Laufzeit erklären. Bei 9% Noise sind die Laufzeiten beider Algorithmen ungefähr gleich bei ca. 57 Sekunden und bei 7% Noise steigt die Laufzeit bei OCEL sehr stark an. Hier beträgt sie für OCEL 192 Sekunden und bei CELOE nur 81 Sekunden. Auf Grund einer *OutOfMemoryException* kann OCEL keine Ergebnisse mehr bei 7% Noise produzieren. Die höchste gemessene Laufzeit bei OCEL beträgt $192,445s \pm 329,859s$ und bei CELOE $86,528s \pm 95,553s$. Somit braucht der Lernalgorithmus OCEL ungefähr doppelt so viel Zeit wie der Lernalgorithmus CELOE.

Die in der Abbildung 20 dargestellten Grafik zeigt die Trainings- und Testgenauigkeit der beiden Algorithmen OCEL und CELOE bei fest eingestellten Laufzeiten. Dabei ist deutlich zu sehen, dass sowohl die Trainings- als auch die Testgenauigkeit bei OCEL höher ist als die bei CELOE. Das die Testgenauigkeit bei OCEL höher ist, als bei CELOE liegt daran, dass beim Durchsuchen des Suchbaumes der Algorithmus OCEL besser bewertet Konzepte in der vorgegebenen Laufzeit gefunden hat als CELOE. Dies wiederum kann daran liegen, dass OCEL Knoten mit Konzepten gefunden hat, die

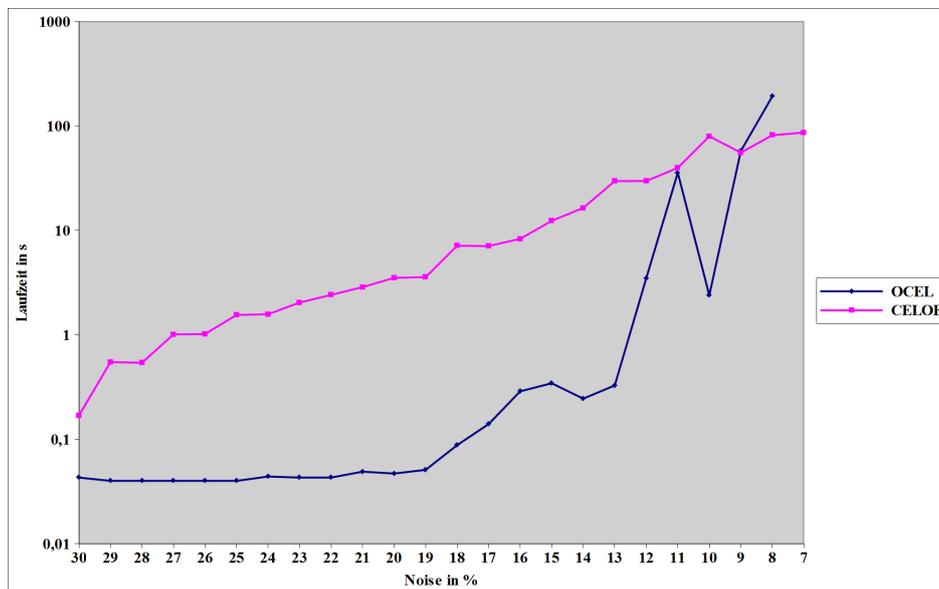


Abbildung 19: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Parkinson Wissensbasis. Dabei ist zu erkennen das der CELOE Algorithmus eine längere Laufzeit besitzt, aber für einen geringen Noise Parameter eine nicht so starke Laufzeiterhöhung hat, wie der OCEL Algorithmus.

entweder besser bewertet waren, oder durch diese schneller Konzepte gefunden werden konnten, die höher bewertet wurden. Das OCEL Ergebnisse mit einer höheren Genauigkeit als CELOE errechnete und dies bei der Verringerung des Noise-Parameters, welcher in Tabelle 18 dargestellt wird, genau umgekehrt ist, lässt sich dadurch erklären, dass bei der Festlegung der Laufzeit keine Konzepte als zu Schwach verworfen werden. Dies führt dazu das die Algorithmen mehr Konzepte durchsuchen müssen. Bei OCEL wurden durch diese uneingeschränkte Suche bessere Konzepte gefunden. CELOE hingegen benötigt mehr Zeit um Konzepte mit einer höheren Genauigkeit zu finden. Aus diesem Grund erreicht OCEL eine höhere Genauigkeit als CELOE.

Die Tabelle 15 zeigt die Testgenauigkeiten der Algorithmen OCEL und CELOE für die eingestellten Laufzeiten von 10, 100, 300 und 1000 Sekunden. Des Weiteren sind die p-Werte, als Ergebnisse des t-Tests und eine Aussage ob eine statistische Signifikanz vorlag, dargestellt. Dabei ist zu erkennen, dass nur bei 10 Sekunden Laufzeit eine statistische Signifikanz für die Unterschiede der Genauigkeit der Testdaten vorlag. In den anderen drei Fällen ist der Unterschied in der Genauigkeit nicht statistisch signifikant.

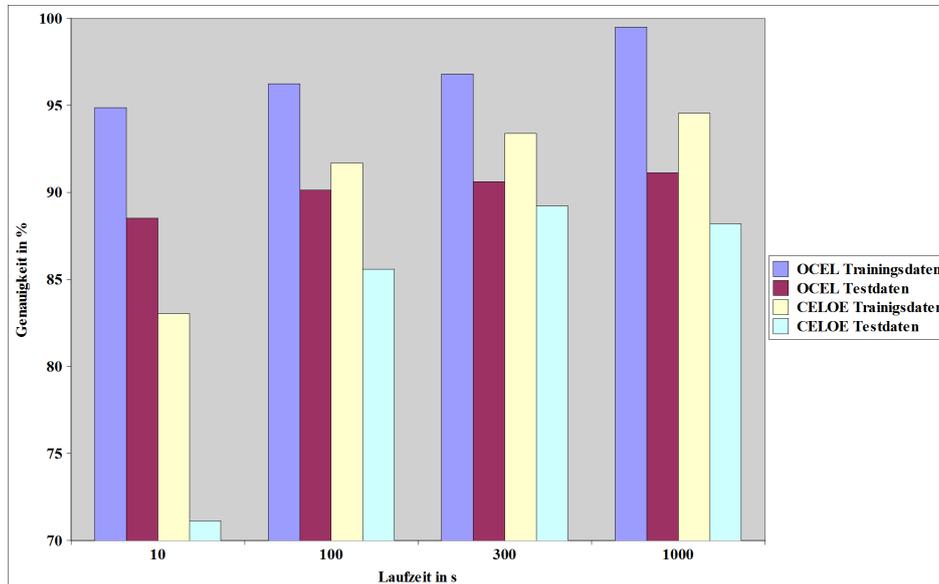


Abbildung 20: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Parkinson Wissensbasis.

Laufzeit	Testgenauigkeit Algorithmus 1	Testgenauigkeit Algorithmus 2	p-Wert	statistische Signifikanz
10 Sekunden	88,515%±11,023%	71,137%±10,987%	0,0024	ja
100 Sekunden	90,123%±8,959%	85,594%±9,575	0,2891	nein
300 Sekunden	90,623%±7,021%	89,205%±6,581%	0,6468	nein
1000 Sekunden	91,126%±8,656%	88,205%±5,415%	0,3776	nein

Tabelle 15: Bei dieser Tabelle handelt es sich um die Darstellung der Testgenauigkeit der Algorithmen OCEL und CELOE bei 10, 100, 300 und 1000 Sekunden Laufzeit und die Ergebnisse des t-Tests und ob eine statistische Signifikanz in den Unterschieden der Genauigkeiten vorliegt. Dabei ist zu erkennen, dass nur bei 10 Sekunden Laufzeit eine statistische Signifikanz in den Unterschieden der Genauigkeit der Trainingsdaten vorlag. In den anderen drei Fällen sind die Ergebnisse nicht statistisch signifikant.

9 Vorhersage von Herzerkrankungen

Bei diesem Datensatz handelt es sich um eine Wissensbasis, die beim Finden von Herzerkrankungen helfen soll. Dabei sollen Eigenschaften gefunden werden, die auf eine Erkrankung des Herzens schließen lassen.

9.1 Datensatz

Der Datensatz stammt aus dem UCI Machine Learning Repository¹⁶ und besitzt 270 Instanzen mit 14 Attributen. Dabei sind 13 Attribute für die Klassifikation und 1 Attribut dient zur Einteilung ob die Instanz eine Herzerkrankung hat oder nicht.

- age

Bei diesem Attribut handelt es sich um das Alter des Patienten.

- sex

Diese Eigenschaft beschreibt das Geschlecht des Patienten.

- chest pain type

Hierbei handelt es sich um den Brustschmerztyp, der durch die Werte 1 bis 4 dargestellt wird.

- resting blood pressure

Dieses Attribut gibt an wie hoch der Ruheblutdruck ist.

- serum cholestoral in mg/dl

Mit diesem Wert wird angegeben, wie hoch der Colesterinspiegel im Blut ist.

- fasting blood sugar > 120 mg/dl

Bei diesem Attribut handelt es sich um den Blutzuckerspiegel im nüchternen Zustand. Dabei steht der Wert 0 für Blutzuckerspiegel unter 120mg/dl und 1 für über 120mg/dl.

- resting electrocardiographic results

Diese Eigenschaft gibt die elektrokardiographischen Werte im Ruhezustand an. Dabei liegen die Werte zwischen 0 und 2.

- maximum heart rate achieved

Hierbei handelt es sich um die maximal gemessene Herzfrequenz der Patienten.

¹⁶Quelle:[http://archive.ics.uci.edu/ml/datasets/Statlog+\(Heart\)](http://archive.ics.uci.edu/ml/datasets/Statlog+(Heart)) Stand:26. April 2011

- exercise induced angina

Bei diesem Attribut wird angegeben, ob eine durch Sportübungen ausgelöste Angina vorlag. Der Wert 0 beschreibt, dass keine Angina vorlag und der Wert 1 beschreibt, dass eine Angina vorlag.

- oldpeak

Mit diesem Wert wird der Rückgang des ST Teils des Kardiogramms, welcher durch sportliche Übungen ausgelöst wurde, relativ zum ST Rückgang bei Ruhe dargestellt.

- the slope of the peak exercise ST segment

Durch dieses Attribut wird das Gefälle des ST Segments, bei maximaler Belastung angegeben.

- number of major vessels

Hierbei handelt es sich um die Anzahl der Hauptgefäße, die zwischen 0 und 3 liegen können.

- thal

Mit diesem Attribut wird angegeben, ob das Herz Normal ist, was durch den Wert 3 dargestellt wird, oder ob ein unveränderlicher Defekt am Herzen vorlag, was durch den Wert 6 dargestellt wird, oder ob ein umkehrbarer Defekt am Herzen vorlag, was durch den Wert 7 dargestellt wird.

- class

Das Attribut class beschreibt das Vorhandensein von einer Herzkrankheit. Dabei bedeutet 2 das eine Herzerkrankung vorliegt und 1 das keine Herzerkrankung vorliegt.

Umwandlung in eine OWL Ontologie

Die Wissensbasis zum Erkennen von Herzkrankheiten lag nicht in einer OWL Ontologie vor. Aus diesem Grund musste die Wissensbasis mit Hilfe eines eigens geschriebenen Parsers umgewandelt werden. Die Abbildung 21 zeigt den Ablauf zur Umwandlung mit ihren Methoden. Im folgenden werden die wichtigsten dieser Methoden genauer beschrieben. Die Methode *main()* steuert den gesamten Ablauf der Umwandlung. Dazu werden in einem ersten Schritt die numerischen Thal Werte, mit Hilfe der Methode *mapThalValues()*, in die entsprechenden Strings umgewandelt, die dann als Name für die entsprechenden Subklassen dienen. Des Weiteren wird mit Hilfe der Methode *generateDomainAndRangeForObjectProperties()* die Domain und Range der einzelnen Object und Data Properties festgelegt. Anschließend wird die Ausgangswissensbasis zeilenweise eingelesen und die entsprechenden Attribute mit Hilfe der Methode *mapClauses()* in OWL Object und Data Properties

und die OWL Klassen umgewandelt. Die Data Properties werden mit Hilfe der Methoden *getDoubleDatatypePropertyAssertion()* und *getBooleanDatatypePropertyAssertion()* erzeugt. Die Klassen werden mit der Methode *getConceptAssertion()* erstellt. Nach dieser Umwandlung wird die Ontologie mit Hilfe der Methode *exportKBToOWL()* des OWLAPIReasoners geschrieben. In einem letzten Schritt wird noch die Konfigurationsdatei geschrieben, welche Informationen über Lernalgorithmus, Lernproblem, Reasoner und positive und negative Beispiele enthält. Dies geschieht in zwei Schritten. Im ersten Schritt wird mit Hilfe der Methode *generateConfFile()*, die Informationen über Lernalgorithmus, Lernproblem und Reasoner geschrieben und im zweiten Schritt werden die positiven und negativen Beispiele mit Hilfe der Methode *generateExamples()* hinzugefügt. Durch die Umwandlung in eine OWL Ontologie entstand eine Ontologie mit 7 Klassen, 4 Object Properties, 9 Data Properties und 280 Instanzdaten.

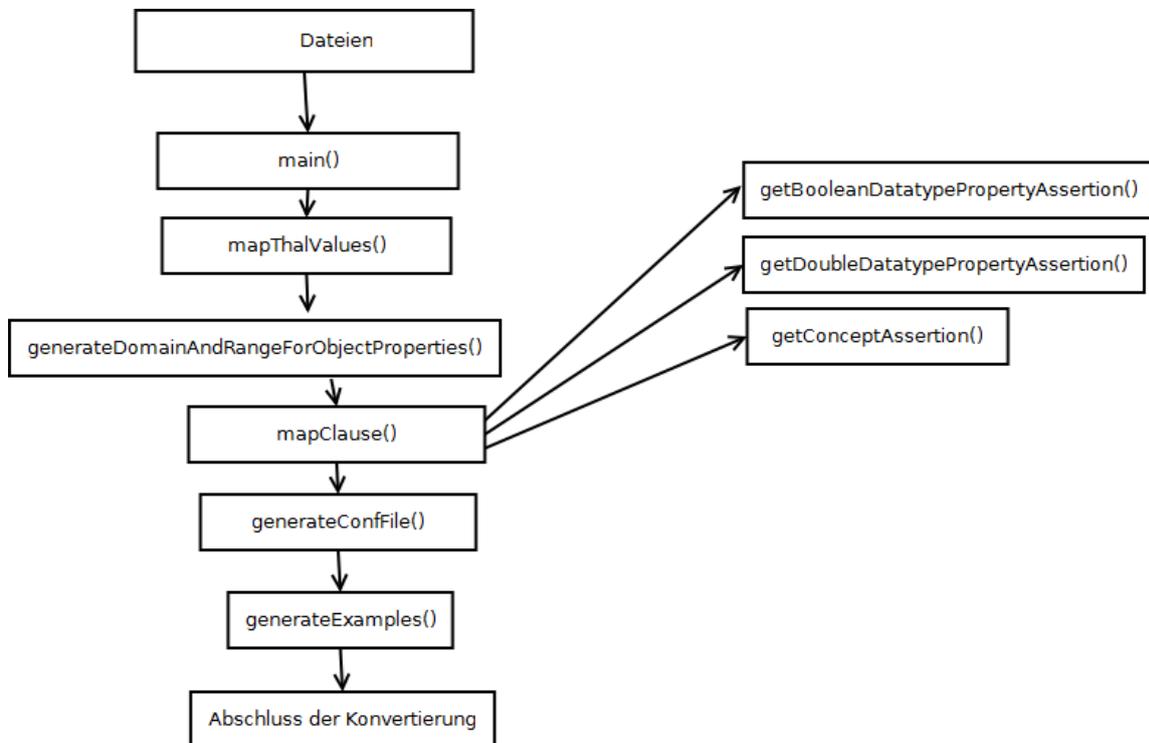


Abbildung 21: Hier ist der Ablauf zur Konvertierung der Prologdateien der Heart Wissensbasis in eine OWL Ontologie mit ihren Methoden dargestellt.

9.2 Methoden

Dieser Abschnitt zeigt welche Tests mit Hilfe der Werkzeuge DL-Learner und Weka durchgeführt wurden. In einem ersten Test wurde der Algorithmus OCEL überprüft. Dazu wurde der Noise Parameter schrittweise von 30%, um jeweils 1% reduziert und eine 10-fold-cross-validation durchgeführt, bis keine Ergebnisse mehr erzeugt wurden. Im Anschluss daran wurde der Vorgang mit dem Lernalgorithmus CELOE wiederholt. Des Weiteren wurde für die beiden Algorithmen die Laufzeit festgehalten, um in einem weiteren Vergleich der Algorithmen, im Hinblick auf die Laufzeit zu haben. Danach wurde für die Algorithmen naive Bayes, naive Bayes Updateable, random Tree, random Forrest, PART und Decision Table des Machine Learning Werkzeugs Weka ebenfalls eine 10-fold-cross-validation durchgeführt. Da es aber keine Einstellmöglichkeit für den Noise Parameter gab wurde dieser Test nur jeweils ein mal durchgeführt. Im Anschluss daran wurden die entstandenen Ergebnisse miteinander verglichen. Bei einem weiteren Test wurde für die Algorithmen OCEL und CELOE eine feste Laufzeit von 10, 100, 300 und 1000 Sekunden gewählt und jeweils eine 10-fold-cross-validation durchgeführt und die Ergebnisse im Anschluss verglichen. Auch hier gab es keine Einstellmöglichkeit für eine feste Laufzeit bei den Algorithmen von Weka und somit konnte der Test nur für die Algorithmen des DL-Learners durchgeführt werden. Damit ein etwa gleiches Verhalten der beiden Algorithmen erzielt werden kann, wurde der β Parameter, welcher die Bestrafung für längere Konzepte durchführt, bei CELOE auf 0,02 gesetzt. Dies entspricht dem Wert von OCEL. Um eine Vergleichbarkeit im Hinblick auf die Laufzeit zu erzielen wurden alle Test auf einem 3,2 GHz Quad Core mit 8GB Ram durchgeführt. Die Ergebnisse und die sich daraus ergebenden Schlussfolgerungen werden im nächsten Abschnitt genauer betrachtet.

9.3 Resultate und Schlussfolgerungen

Die in der Abbildung 22 dargestellte Grafik, zeigt die Genauigkeit der Algorithmen, in Abhängigkeit des eingestellten Noise Parameters. Dabei ist deutlich zu sehen, dass alle Genauigkeiten der Algorithmen des Tools Weka mit einer Geraden dargestellt wurden. Das liegt daran, dass es bei Weka keine Möglichkeit zur Einstellung eines Noise Parameters gab. Somit wurde nur ein Durchlauf gemacht und die Resultate der einzelnen Algorithmen als Gerade dargestellt. Eine weitere Auffälligkeit ist, dass es sechs Algorithmen in der Legende gibt, aber nur 5 Geraden dargestellt sind. Das liegt daran, dass die Algorithmen naive Bayes und naive Bayes Updateable die gleiche Genauigkeit erzielt haben und somit die Gerade von naive Bayes von der Gerade naive Bayes Updateable verdeckt wurde. Des Weiteren ist zu sehen, dass die Testgenauigkeit bei OCEL und CELOE bei 29% Noise fällt und bis 26% steigt und fällt. Um herauszufinden warum die Testgenauigkeit abnimmt wurde für beide Algorithmen für 30% und 29% Noise der Suchbaum ausgegeben. Bei der Auswertung des Suchbaumes für OCEL ist aufgefallen, dass der Algorithmus bei 30% Noise das Konzept (*Patient AND hasAngina value true*)

mit einem DL-Learner internen Genauigkeitsbewertung von 70,37% findet, welches er noch einen Schritt weiterverfolgt und das Konzept (*Patient AND hasAngina value true AND hasStSegment >= 1.5*) mit einer Bewertung von 71,193% findet. Bei der Reduzierung des Noise Parameters auf 29% ist dieses Konzept zu Schwach und wird nicht weiter verfolgt. Bei der Suche eines Konzeptes welches nicht zu Schwach ist, werden viele Knoten durchsucht, wodurch die Konzepte immer Komplexer werden. Dabei ist es Möglich das der Algorithmus ein Konzept findet, welches eine Bewertung für die Trainingsdaten bekommt, die den Anforderung (1-Noise) genügt, aber nicht so gut auf die Testdaten passt und dadurch eine schlechtere Testbewertung bekommt. Aus diesem Grund ist es möglich, dass trotz einer Steigerung der Trainingsgenauigkeit die sinkt. Bei der Auswertung des Suchbaumes von CELOE ergab sich ein ähnliches Bild, so dass dies nicht noch einmal näher erläutert wird. Ab 25% steigt die Genauigkeit der Testdaten wieder an, wobei bei CELOE bei 24% keine Ergebnisse mehr erzeugt wurden. Dabei lässt sich der Anstieg der Testgenauigkeit damit erklären, dass die gefundenen Konzepte ab 25% Noise wieder besser auf die Testdaten passen und somit eine höhere Genauigkeitsbewertung bekommen.

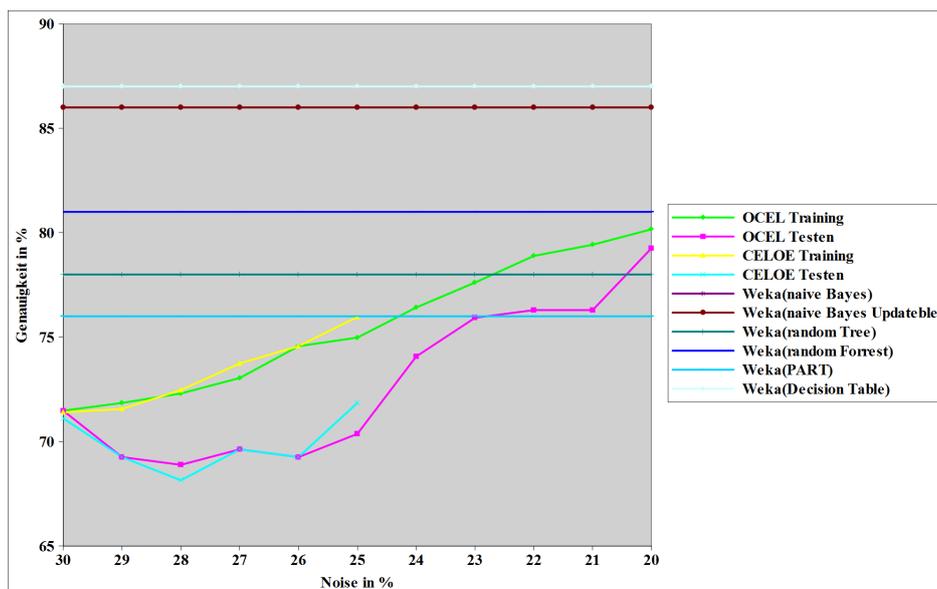


Abbildung 22: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Heart Wissensbasis. Sowohl bei OCEL als auch bei CELOE fällt die Genauigkeit der Testdaten bei 29% und 28% ab. Anschließend steigt diese wieder. Dabei erreicht CELOE die geringste Genauigkeit bei den Testdaten und OCEL liegt an 4. Stelle. Die Ergebnisse der Algorithmen von Weka wurden alle als Gerade dargestellt, da es keine Möglichkeit gab einen Noise Parameter einzustellen wurde nur ein Durchlauf gemacht und die Ergebnisse als Gerade dargestellt.

In der Tabelle 16 werden die höchsten Genauigkeiten der Testdaten bei der 10-fold-cross-validation

und ihre Laufzeiten mit entsprechenden Konzeptlängen angegeben. Dabei ist zu sehen, dass Decision Table, Naive Bayes, Naive Bayes Updateable und Random Forrest Ergebnisse mit einer höheren Genauigkeit als OCEL und CELOE liefern. Danach kommt OCEL und es schließen sich die Algorithmen Random Tree und PART an. Die Ergebnisse mit den schlechtesten Genauigkeiten liefert CELOE. Des Weiteren ist zu erkennen, dass die Algorithmen von Weka eine deutlich geringere Laufzeit aufweisen als die des DL-Learners. Dies lässt sich dadurch erklären, dass die Algorithmen von Weka, im Gegensatz zu den Algorithmen des DL-Learners, keinen Reasoner benötigen um ein Konzept zu errechnen. Des Weiteren sind die Algorithmen auf das Eingabeformat, nämlich die ARFF Datei, so dass die Berechnung der Konzepte gleich durchgeführt werden kann. Außerdem sind die Algorithmen auf numerische Werte spezialisiert, so dass diese viel schneller zu Ergebnissen kommen.

Algorithmus	Testgenauigkeit	Laufzeit	Konzeptlänge
OCEL	79,259%±6,807%	233,143s±164,342s	14,8±1,932
CELOE	71,852%±10,649%	201,811s±121,084s	12,6±3,406
Naive Bayes	85,64%±5,86%	0,0006s±0,0009s	–
Naive Bayes Updateable	85,64%±5,86%	0,0008s±0,0010s	–
Random Tree	77,95%±10,34%	0,0014s±0,0008s	–
Random Forrest	80,52%±5,12%	0,0112s±0,0009s	–
PART	75,79%±8,01%	0,0177s±0,0272s	–
Decision Table	86,60%±5,63%	0,0166s±0,0013s	–

Tabelle 16: In dieser Tabelle werden die höchsten Genauigkeiten und die dazu entsprechenden Laufzeiten und Konzeptlängen der 10-fold-cross-validation der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für die Wissensbasis Heart dargestellt.

Um zu überprüfen, ob die Genauigkeitsunterschiede statistisch signifikant sind wird der t-Test durchgeführt. Die p-Werte des t-Tests und eine Aussage ob diese Werte statistisch Signifikant sind, wird in Tabelle 17 dargestellt. Zu den Ergebnissen von OCEL im Vergleich zu den anderen Algorithmen ist zu sagen, dass OCEL in drei Fällen statistisch Signifikant schlechter ist als die anderen Algorithmen, nämlich bei Naive Bayes, Naive Bayes Updateable und Decision Table. In einem Fall, nämlich bei Random Forrest, ist OCEL nicht statistisch signifikant schlechter. Bei den letzten beiden Fällen, nämlich Random Tree und PART ist OCEL nicht statistisch Signifikant besser. Bei der Auswertung der t-Tests von CELOE mit den Algorithmen von Weka wurden folgende Ergebnisse festgestellt. CELOE hat in vier Fällen eine statistisch signifikant schlechtere Genauigkeit als die Algorithmen von Weka. Die vier Algorithmen die eine statistisch Signifikant bessere Genauigkeit aufweisen heißen Naive Bayes, Naive Bayes Updateable, Random Forrest und Decision Table. Nur in zwei Fällen, nämlich bei Random Tree und PART war CELOE nicht statistisch signifikant schlechter. Der Vergleich zwischen den beiden Algorithmen OCEL und CELOE ergab, dass der Unterschied in der Genauigkeit

nicht statistisch Signifikant ist.

Algorithmen	p-Wert	statistische Signifikanz
OCELOE vs. Naive Bayes	0,0375	ja
OCELOE vs. Naive Bayes Updateable	0,0375	ja
OCELOE vs. Random Tree	0,7420	nein
OCELOE vs. Random Forrest	0,6453	nein
OCELOE vs. PART	0,3105	nein
OCELOE vs. Decision Table	0,0171	ja
CELOE vs. Naive Bayes	0,0021	ja
CELOE vs. Naive Bayes Updateable	0,0021	ja
CELOE vs. Random Tree	0,2103	nein
CELOE vs. Random Forrest	0,0323	ja
CELOE vs. PART	0,3624	nein
CELOE vs. Decision Table	0,0011	ja
CELOE vs. OCELOE	0,0803	nein

Tabelle 17: In dieser Tabelle wird die statistische Signifikanz der Algorithmen OCELOE, CELOE und den Algorithmen von Weka, für die Wissensbasis Heart dargestellt. Dabei ist zu sehen, dass OCELOE in drei Fällen statistisch signifikant schlechtere Ergebnisse, in einem Fall nicht statistisch signifikant schlechtere Ergebnisse und in zwei Fällen nicht statistisch signifikant bessere Ergebnisse liefert, als die anderen Algorithmen. Bei CELOE zeigt es das es in vier Fällen statistisch signifikant schlechtere Ergebnisse und in 2 Fällen nicht statistisch signifikant schlechtere Ergebnisse erzeugt wurden. Bei einem Vergleich zwischen OCELOE und CELOE kam heraus, dass CELOE nicht statistisch signifikant schlechter war als OCELOE.

In der Tabelle 18 sind die von den Algorithmen vorgeschlagenen Konzepte mit der höchsten Genauigkeit dargestellt. Des Weiteren wurden auch die Laufzeiten der Algorithmen und die Konzeptlänge angegeben. Dabei ist zu sehen, dass sowohl CELOE als auch OCELOE Konzepte mit der gleichen Testgenauigkeit vorgeschlagen haben. Nur das Ergebnis von OCELOE ist Länger als das von CELOE, wobei zwei von den drei Eigenschaften die von CELOE angegebenen wurden, mit in dem Konzept von OCELOE enthalten sind. Im folgenden werden die Konzepte kurz erklärt. Das Konzept, welches von OCELOE vorgeschlagen wurde bedeutet, dass ein Patient, der ein StSegment das größer gleich 1,5 und mehr Hauptgefäße als 1 oder eine Angina und einen Blutdruck der größer gleich 119 ist, eine Herzerkrankung hat. Demgegenüber bedeutet das von CELOE vorgeschlagene Konzept, dass eine Person die ein StSegment, das größer gleich 1,5 und Angina oder ein Alter von mehr als 45 Jahren besitzt, eine Herzerkrankung hat. Um herauszufinden warum es zwischen OCELOE und CELOE zu un-

terschieden in den vorgeschlagenen Konzepten und den Genauigkeiten dieser kommt wurde für beide Algorithmen der Suchbaum ausgegeben. Dabei ergab die Untersuchung der Suchbäume, dass die Algorithmen beim Aufbau des Suchbaumes, gefundene Konzepte intern anders bewerten. Aus diesem Grund werden unterschiedliche Pfade im Suchbaum weiter durchsucht. Deshalb kommt es dazu, dass die Algorithmen unterschiedliche Konzepte vorschlagen. Jedoch die Elemente *hasStSegment* ≥ 1.5 und *hasAngina value true* kommen in beiden Vorschlägen vor, so dass man sagen kann, dass diese Teilkonzepte besonders wichtig sind, um Personen herauszufiltern, die an einer Herzerkrankung leiden.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasStSegment ≥ 1.5 AND (hasMajorVessels ≥ 0.5 OR (hasAngina value true AND hasBloodPressure ≥ 119.0)))</i>	88,889%	1,296s	16
CELOE	<i>(hasStSegment ≥ 1.5 AND (hasAngina value true OR hasAge ≥ 45.5))</i>	88,889%	3,455s	10

Tabelle 18: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE dargestellt. Bei Weka wurden keine gelernten Konzepte angezeigt, so dass hier nur die Ergebnisse des DL-Learners angezeigt werden.

In der Abbildung 23 wird die Laufzeit der Algorithmen OCEL und CELOE in Abhängigkeit des eingestellten Noise Parameters dargestellt. Dabei ist zu sehen, dass bei CELOE die Laufzeit schneller ansteigt als bei OCEL und die höchste Laufzeit bei 26% Noise besitzt. Diese liegt bei ungefähr 300 Sekunden. Der Anstieg lässt sich durch das weiter oben beschriebene Problem erklären, dass die Algorithmen länger brauchen um Konzepte zu finden, die der Anforderung (1-Noise) genügen. Das der Anstieg der Laufzeit bei CELOE schneller geht als bei OCEL liegt daran, dass die Heuristik, welche die Konzepte bewertet, sehr viele Konzepte nur mit einer sehr geringen Genauigkeit bewertet als OCEL und somit länger bei der Suche eines geeigneten Konzeptes braucht. Danach sinkt die Laufzeit auf ca. 200 Sekunden. OCEL hingegen hat bis 25% Noise eine geringe Laufzeit, die unterhalb von 0,7 Sekunden liegt. Anschließend steigt die Laufzeit auf 320 Sekunden an und fällt bei 20% Noise wieder auf 230 Sekunden. Der Abfall der Laufzeit bei CELOE liegt daran, dass der Algorithmus bei Fold 3 und 4 zwar jeweils die gleichen Konzepte lernt, aber bei 26% deutlich länger braucht als bei 25%. Des Weiteren lernt CELOE bei Fold 7 bei 26% Noise ein deutlich komplexeres Konzept als bei 25% Noise. Dieser Vorgang dauert somit auch viel länger. Das CELOE bei 26% Noise ein komplexeres Konzept lernt als bei 25% liegt daran, dass der Algorithmus bei 25% Noise ein Konzept als zu schwach bewertet, welches bei 26% noch ausreichend war. Dies führt dazu dass ein anderes Konzept gefunden werden muss, welches den Anforderungen genügt. Dabei hat der Algorithmus ein Konzept,

welches nur 25% Noise besitzt, schneller gefunden, da er im Suchbaum einen anderen Weg gegangen ist. Durch diese beiden Unterschiede bei den Folds 3,4 und 7 sind die unterschiedlichen Laufzeiten entstanden. Bei der Auswertung der Ergebnisse von OCEL kam heraus, dass der Algorithmus bei Fold 2, 4, 9 etwas andere Konzepte bei 20% Noise findet, als bei 21%. Diese Konzepte werden viel schneller gefunden, als die bei 21%, so dass die Laufzeitunterschiede von diesen 3 Fold kommen, da die anderen Folds eine ungefähr gleiche Laufzeit besaßen. Das OCEL bei dem Fold 2, 4, 9 andere Konzepte lernt, liegt wieder daran, dass die ein gefundenen Konzept im Suchbaum, welches bei 21% Noise noch ausreichend war, für 20% Noise zu Schwach ist. Dies führt dazu, dass der Algorithmus einen anderen Pfad im Suchbaum durchsucht. Dadurch ist es möglich dass der Algorithmus bei der Wahl eines anderen Pfades, schneller zu dem Ergebnis gelangt, welches eine ausreichend hohe Bewertung besitzt. Durch das schnellere Finden eines geeigneten Konzeptes kommt zur Verringerung der Laufzeit. Aus diesem Grund kam es zur Verringerung der durchschnittlichen Laufzeit.

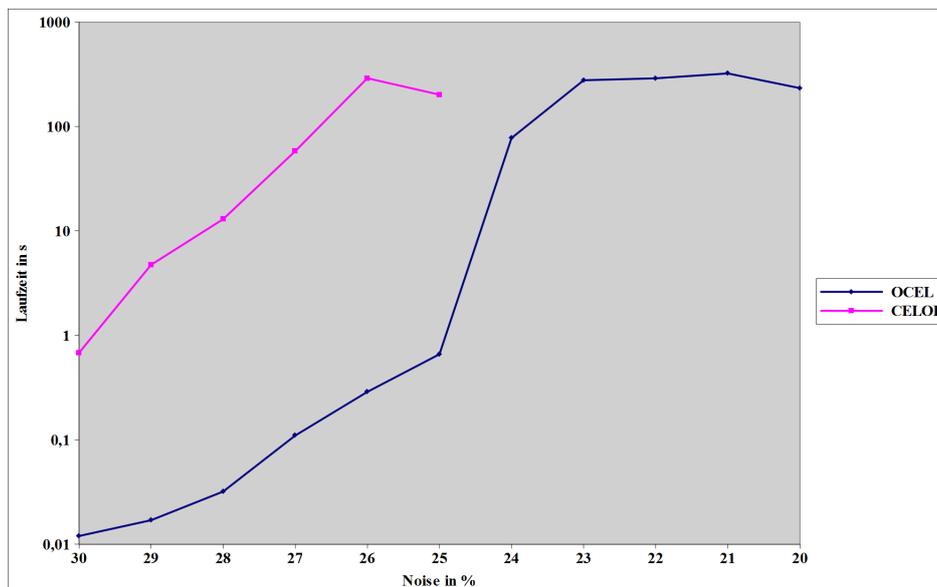


Abbildung 23: Die Grafik zeigt die Laufzeiten der Algorithmen OCEL und CELOE anhand des eingestellten Noise Parameter für die Heart Wissensbasis. Deutlich zu sehen ist, dass beide Algorithmen bei der längsten Laufzeit ca. 300 Sekunden benötigen und die Laufzeit anschließend abfällt. Jedoch findet CELOE bei 24% Noise keine Ergebnisse mehr.

Die in der Abbildung 24 dargestellte Grafik, zeigt die Genauigkeit der Algorithmen OCEL und CELOE bei fest eingestellten Laufzeiten von 10, 100, 300 und 1000 Sekunden. Da es bei Weka keine Möglichkeit gab, eine feste Laufzeit für die Algorithmen einzustellen wurden diese Tests für die Algorithmen von Weka nicht durchgeführt. Es ist dabei zu sehen, dass CELOE, bei allen viel eingestellten Laufzeiten, Ergebnisse mit einer höheren Genauigkeit liefert.

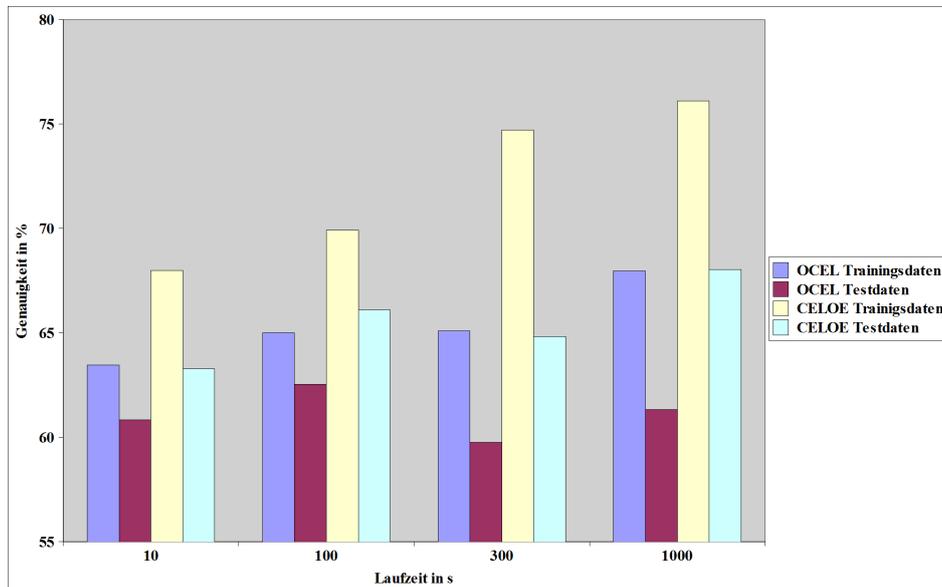


Abbildung 24: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Heart Wissensbasis. Deutlich zu sehen ist, dass der CELOE Algorithmus bei allen eingestellten Laufzeiten, sowohl bei Trainings- als auch Testgenauigkeit höhere Werte hat als OCEL.

In der Tabelle 19 sind die Testgenauigkeiten der Algorithmen OCEL und CELOE für die eingestellten Laufzeiten, die p-Werte und eine Aussage ob der errechnete p-Wert statistisch signifikant ist, dargestellt. Dabei ist zu erkennen, dass es bei allen viel Laufzeiten CELOE keine statistisch signifikant besseren Ergebnisse liefert als OCEL. Diese Ergebnisse sind aber schlechter als die, die bei fest eingestellten Noise-Parameter errechnet wurden (Tabelle 22). Dies liegt daran, dass wenn nur der Laufzeit-Parameter eingestellt ist, werden keine Konzepte wegen einer zu geringen internen Bewertung verworfen. Aus diesem Grund muss der Algorithmus viel mehr Konzepte weiterverfolgen, so dass er in der Laufzeit, die zur Verfügung steht weniger Konzepte finden kann, die eine höhere interne Bewertung besitzen. Deshalb sind die Genauigkeiten der Ergebnisse bei der fest eingestellten Laufzeit geringer als bei fest eingestelltem Noise-Parameter. Des Weiteren ist zu erkennen, dass bei fest eingestellter Laufzeit, CELOE Ergebnisse mit höherer Genauigkeit als OCEL erzielt und bei eingestelltem Noise-Parameter (siehe Abbildung 22) es sich genau anders herum verhält. Dies liegt daran, dass die Algorithmen bei fest eingestellter Laufzeit, keine Konzepte als zu Schwach deklarieren und verwerfen. Dadurch werden mehr Konzepte durchsucht. Das führt dazu dass bei CELOE schneller Konzepte mit höherer Genauigkeit gefunden werden, als bei OCEL. Aus diesem Grund kommt es dazu, dass CELOE eine höhere Testgenauigkeit erreicht als OCEL.

Laufzeit	Testgenauigkeit OCEL	Testgenauigkeit CELOE	p-Wert	statistische Signifi- kanz
10 Sekunden	60,831%±2,318%	65,002%±1,222%	0,3804	nein
100 Sekunden	62,528%±2,364%	66,098%±7,954%	0,1905	nein
300 Sekunden	59,763%±5,671%	64,829%±8,554%	0,1359	nein
1000 Sekunden	61,317%±4,213%	68,031%±10,586%	0,0788	nein

Tabelle 19: Bei dieser Tabelle handelt es sich um die Darstellung der Testgenauigkeit der Algorithmen OCEL und CELOE bei 10, 100, 300 und 1000 Sekunden Laufzeit, für die Wissensbasis Heart und die Ergebnisse des t-Tests, ob eine statistische Signifikanz in den Unterschieden der Genauigkeiten vorliegt. Dabei ist zu erkennen, dass es bei allen vier Laufzeit Einstellungen keine statistische Signifikanz in den Unterschieden der Genauigkeit gibt.

10 Vorhersage von Brustkrebs anhand von Mammographie Bildern

Bei diesem Datensatz handelt es sich um die Klassifizierung von Mammographie Bildern. Dabei soll mit Hilfe typischen Merkmalen der Bilder und der vorher beschriebenen BI-RADS Klassifikation, Eigenschaften gefunden werden, um Brüste mit Brustkrebs von denen ohne Brustkrebs zu unterscheiden.

10.1 Datensatz

Dieser Datensatz stammt aus dem UCI Machine Learning Repository¹⁷ und hat 961 Instanzdaten. Diese Instanzdaten wurden eingeteilt in 516 Personen ohne Brustkrebs und 445 Personen mit Brustkrebs. Dabei wurden 6 Attribute angegeben, wobei 5 zur eigentlichen Klassifikation benutzt wurden und das 6. Attribut für die Kennzeichnung ob die Person Brustkrebs hat verwendet. Die Attribute waren BI-RADS assessment, Age, Shape, Margin, Density und Severity. Im folgenden werden sie genauer beschrieben.

- BI-RADS assessment

Das BI-RADS assessment beschreibt die Einteilung der Mammographie Ergebnisse, wie sie in Abschnitt 3.2.2 beschrieben wurden.

- Age

Hiermit wird das Alter des Patienten angegeben, zu dem Zeitpunkt als die Mammographie durchgeführt wurde.

- Shape

Bei dem Attribut Shape wird die die Form der Masse die bei der Mammographie gefunden wurde charakterisiert. Dabei stehen 4 Möglichkeiten, nämlich 1 für rund, 2 für oval, 3 für lobulär und 4 für formwidrig zur Verfügung.

- Margin

Bei Margin handelt es sich um die Angabe des Randsaum der in der Mammographie gefundenen Masse. Für dieses Attribut stehen 5 Möglichkeiten zur Verfügung. es handelt sich um 1 für begrenzt, 2 für microlobulated, 3 für undeutlich, 4 für ungenau bestimmt und 5 für spiculated.

- Density

¹⁷Quelle:<http://archive.ics.uci.edu/ml/datasets/Mammographic+Mass> Stand:26. April 2011

Das Attribut Density gibt Aussage über die Dichte der in der Mammographie gefundenen Masse. Dabei stehen 4 Varianten zur Verfügung, nämlich 1 für hohe Dichte, 2 für iso, 3 für geringe Dichte und 4 für enthaltenes Fett.

- Severity

Severity enthält die Klassifikation ob diese Person Brustkrebs hat oder nicht. Dabei steht 0 für kein Brustkrebs und 1 für Brustkrebs.

Umwandlung in eine OWL Ontologie

Da auch dieser Datensatz nur in einer Flat File Database vorhanden war, wurde ein eigener Parser geschrieben, welcher die Daten in eine OWL Ontologie konvertiert. Dazu wurde eine Java-Klasse geschrieben. In der Abbildung 25 ist der Ablauf der Umwandlung mit ihren Methoden dargestellt. Im folgenden wird der Ablauf der Umwandlung beschrieben und dabei die wichtigsten Methoden genannt. Der Ablauf des Programms wird mit Hilfe der *main()* Methode gesteuert. Dabei wird zuerst eine neue leere OWL Ontologie angelegt. Anschließend werden mit den Methoden *createShapeMapping()*, *createMarginMapping()* und *createDensityMapping()* die Mappings für die im vorherigen Abschnitt aufgezählten Attribute durchgeführt, damit keine Zahlen als Klassen vorkommen. Im nächsten Schritt werden die Subklassen für die Superklassen Measure, Shape, Margin und Density mit der Methode *setMeasures()* vorbereitet und anschließend als Subklassen deklariert. Danach wird mit der Methode *generateDomainAndRangeForObjectProperties()* die Domain und Range der Object- und Data Properties festgelegt. Im folgenden wird die Datei zeilenweise eingelesen und die Attribute an die Methode *mapClauses()* übergeben. Hier werden nun die einzelnen Attribute für die Ontologie vorbereitet. Dies geschieht mit Hilfe der Methoden *getConceptAssertion()*, welche die Instanzen in die Klassen einordnet, *getDoubleDatatypePropertyAssertion()*, welche die Double Data Properties der Instanzen anlegt und *getRoleAssertion()*, welche die Object Properties der Klassen anlegt. Anschließend wird die Ontologie und die Konfigurationsdatei geschrieben, welche für den Lernvorgang benötigt wird. Dies geschieht mit Hilfe der Methoden *generateConfFile()*, welche den Kopf der Datei erzeugt und *generateExamples()*, welche die positiven und negativen Beispiele in die Konfigurationsdatei einfügt. Nach diesen Schritten ist die Ontologie fertig und dabei sind 19 Klassen 3 Object Properties, 2 Data Properties und 975 Instanzdaten entstanden.

10.2 Methoden

In diesem Abschnitt werden die Experimente zum Vergleich der einzelnen Lernalgorithmen genauer beschrieben. Das erste Experiment wurde mit OCEL durchgeführt. Dazu wurde der Noise Parameter

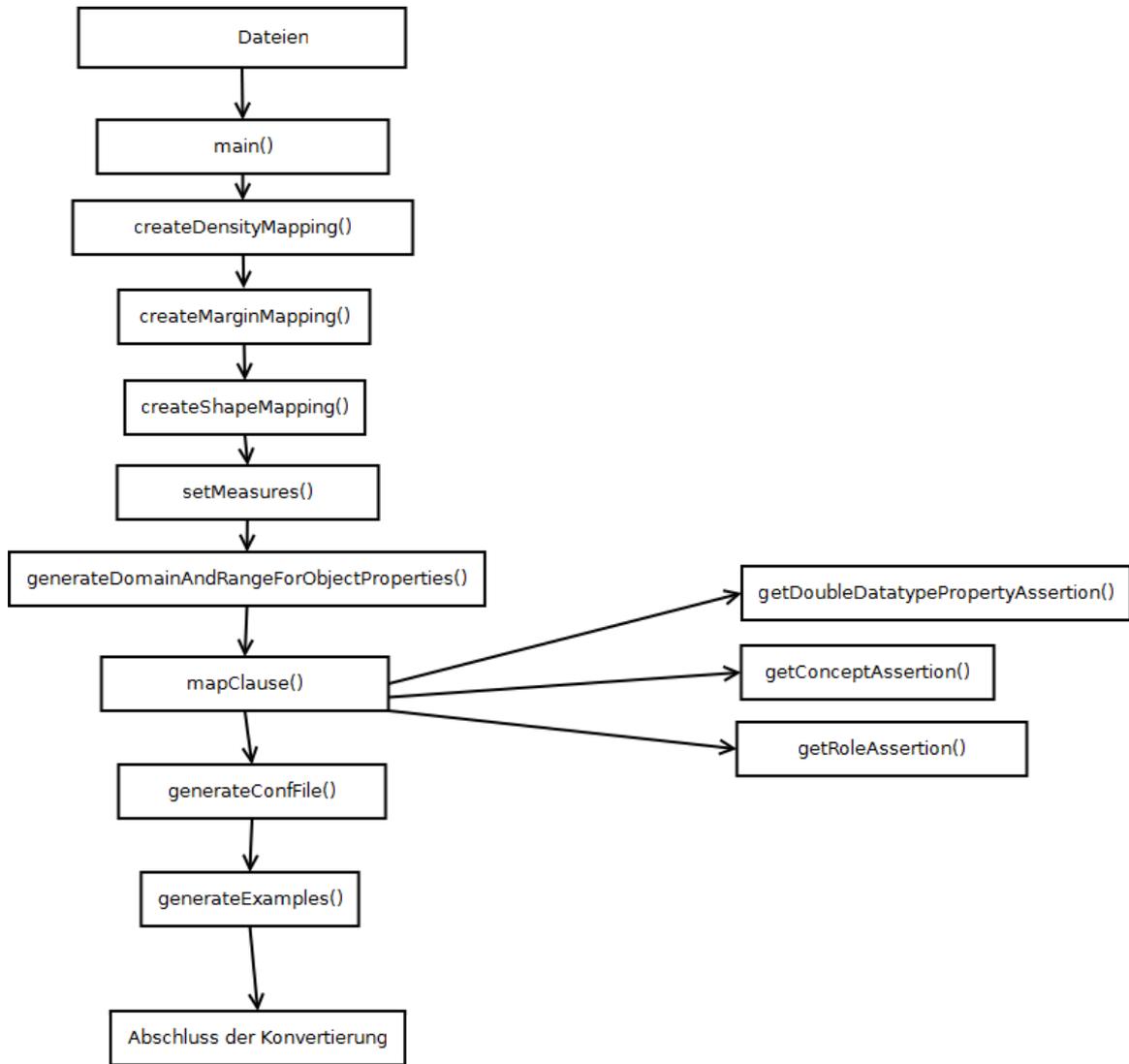


Abbildung 25: Hier ist der Ablauf zur Konvertierung der Prologdateien der Mammographic Wissensbasis in eine OWL Ontologie mit ihren Methoden dargestellt.

des Lernalgorithmus von 30%, in 1% Schritten reduziert, bis keine Ergebnisse mehr produziert wurden. Anschließend wurde das gleiche Experiment mit dem CELOE Algorithmus durchgeführt. Dazu wurde jeweils eine 10-fold-cross-validation durchgeführt und anschließend der Noise Parameter reduziert. Danach wurde eine einzelne 10-fold-cross-validation für die Algorithmen naive Bayes, naive Bayes Updateable, random Tree, random Forrest, PART und Decision Table des Machine Learning Werkzeugs Weka durchgeführt. Da es bei diesen Algorithmen keinen Noise Parameter gab, konnte nur eine 10-fold-cross-validation durchgeführt werden. Des Weiteren wurde für die Algorithmen OCEL und CELOE eine feste Laufzeit von 10, 100, 300 und 1000 Sekunden gewählt und jeweils eine 10-fold-cross-validation durchgeführt. Auch bei den Algorithmen in Weka stand keine Option für die Wahl einer festen Laufzeit zur Verfügung, so dass auch hier keine Tests bezüglich der Laufzeit durchgeführt werden konnten. Da es auch hier Unterschiede des β Parameters für die Algorithmen OCEL und CELOE vorlagen, wurde der β Parameter von CELOE auf 0.02 gesetzt, wie er auch bei OCEL ist. Des Weiteren wurden alle Tests auf einem Quad Core mit 3.2 GHz und 8GB Ram durchgeführt um eine Vergleichbarkeit zu gewährleisten. Im folgenden Abschnitt werden die Ergebnisse genauer vorgestellt.

10.3 Resultate und Schlussfolgerungen

Die Abbildung 26 zeigt die Genauigkeit der Lernergebnisse in Abhängigkeit des eingestellten Noise Parameters. Da die Algorithmen naive Bayes und naive Bayes Updateable die gleichen Genauigkeiten besitzen, sind die Geraden übereinander dargestellt. Aus diesem Grund kann man die Ergebnisse der unterschiedlichen Lernalgorithmen schlecht sehen. Des Weiteren ist die Genauigkeit von OCEL Training und OCEL Testen nur sehr gering unterschiedlich, so dass auch hier eine Unterscheidung der beiden Algorithmen schwierig ist. Außerdem besitzt der Lernalgorithmus CELOE eine sehr geringe Genauigkeit im Gegensatz zu den anderen Algorithmen, mit einem sehr deutlichen Genauigkeitseinbruch bei den Testdaten bei 25%. Da auch bis 25% Noise die Trainingsgenauigkeit immer mal abgefallen ist, wurde ein Suchbaum ausgegeben, um festzustellen warum dies geschieht. Dabei ist aufgefallen, dass wenn der Noise-Parameter reduziert wird, Konzepte, welche vorher eine ausreichend hohe Bewertung besitzen als zu Schwach deklariert werden, da die die Anforderungen an den Noise-Parameter nicht mehr erfüllen. Das führt dazu, dass der Algorithmus andere Konzepte finden muss, die dem eingestellten Noise-Parameter entsprechen. Dabei kann es dazu kommen, dass Konzepte gefunden werden, die eine ausreichend hohe interne Genauigkeitsbewertung erhalten haben, aber trotzdem niedriger bewertet worden als die Konzepte davor. Dies kann daran liegen, dass ein andere Pfad im Suchbaum durchsucht wurde. Bei den neuen Konzepten mit einer geringeren internen Bewertung kann es dazu kommen, dass auch das Konzept nicht ausreichend Testdaten abdecken kann, so dass es zu dem Abfall der Testgenauigkeit kommt, wie in Abbildung 26 bei 25% Noise zu sehen ist. Die Ergebnisse der Lernalgorithmen von Weka wurde als Gerade angegeben, weil es bei

diesen Algorithmen keine Möglichkeit gab, einen Noise Parameter zu setzen. Das Ergebnisse von OCEL ist auch als Gerade dargestellt, da der Algorithmus direkt ein Konzept mit 82% Testgenauigkeit gefunden hat. Trotz Verringerung des Noise-Parameters fand der Algorithmus keine besseren Ergebnisse.

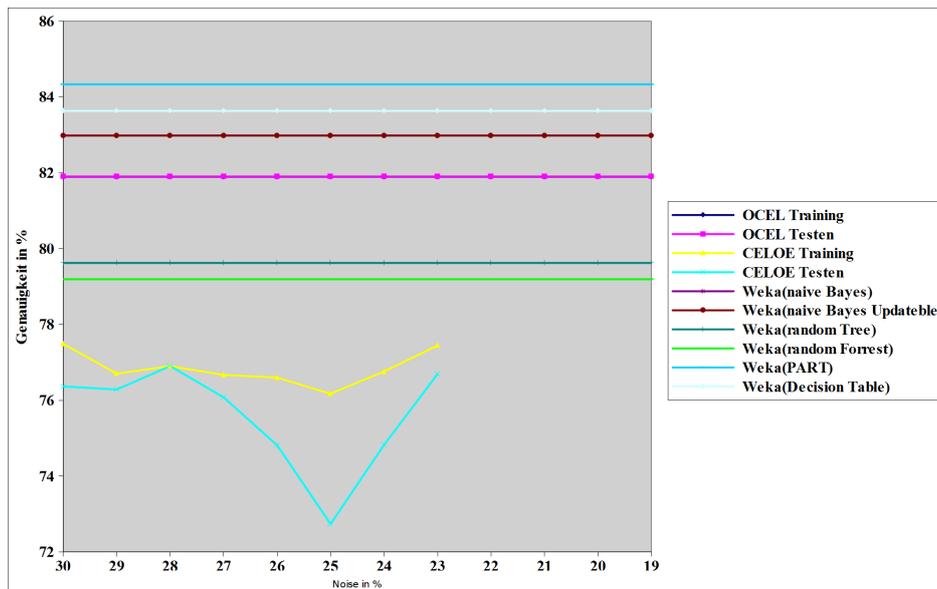


Abbildung 26: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Mammographic Masses Wissensbasis.

In der Tabelle 20 sind für die einzelnen Algorithmen, die höchsten Genauigkeiten mit den entsprechenden Laufzeiten und Konzeptlängen angegeben. Dabei ist zu sehen, dass PART die höchste Genauigkeit erreicht hat, dicht gefolgt von Decision Table und Naive Bayes und Naive Bayes Updateable. Die viert höchste Genauigkeit erzielte OCEL, dicht gefolgt von Random Forrest und Random Tree. Die schlechteste Genauigkeit erreicht CELOE. Des Weiteren ist zu erkennen, dass alle Algorithmen von Weka eine viel geringere Laufzeit haben als die Algorithmen des DL-Learners. Dies lässt sich dadurch erklären, dass die Algorithmen von Weka, im Gegensatz zu den Algorithmen des DL-Learners, keinen Reasoner benötigen um ein Konzept zu errechnen. Des Weiteren sind die Algorithmen auf das Eingabeformat, nämlich die ARFF Datei, so dass die Berechnung der Konzepte gleich durchgeführt werden kann. Außerdem sind die Algorithmen auf numerische Werte spezialisiert, so dass diese viel schneller zu Ergebnissen kommen. Ein weiterer Punkt ist, dass bei den Algorithmen von Weka, bei der 10-fold-cross-validation keine Konzepte ausgegeben wurden, so dass auch keine Konzeptlänge angegeben werden konnte.

Die in der Tabelle 21 dargestellten p-Werte sind die Ergebnisse des t-Tests. Zu diesen p-Werten wurde noch eine kurze Anmerkung angefügt, ob eine statistische Signifikanz in den Unterschieden der Ge-

Algorithmus	Testgenauigkeit	Laufzeit	Konzeptlänge
OCEL	81,899%±4,114%	0,104s±0,158s	5±0
CELOE	76,69%±5,679%	7512s±12374,955s	11±3,399
Naive Bayes	82,98%±1,64%	0,0010s±0,0010s	–
Naive Bayes Updateable	82,98%±1,64%	0,0010s±0,0008s	–
Random Tree	79,62%±1,73%	0,0053s±0,0010s	–
Random Forrest	79,19%±3,44%	0,0495±0,0019s	–
PART	84,33%±2,25%	0,0094s±0,0024s	–
Decision Table	83,63%±1,62%	0,0155s±0,0016s	–

Tabelle 20: In dieser Tabelle werden die höchsten Genauigkeiten und die dazu entsprechenden Laufzeiten und Konzeptlängen der 10-fold-cross-validation der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für die Wissensbasis Mammographic dargestellt. Bei Weka wurden keine Konzepte ausgegeben, so dass die Konzeptlänge nicht angegeben werden kann.

naugigkeit vorlag. Dabei ist deutlich zu sehen, dass obwohl OCEL nur die viert höchste Genauigkeit besitzt, die Unterschiede in den Testgenauigkeiten bei allen sechs Fällen nicht statistisch signifikant sind. Bei CELOE hingegen sieht es anders aus. In fünf der sechs Fälle liefert CELOE statistisch signifikant schlechtere Ergebnisse als die Algorithmen von Weka. In einem Fall ist CELOE nicht statistisch signifikant Schlechter als Weka. Bei einem direkten Vergleich der beiden Lernalgorithmen OCEL und CELOE, des DL-Learners ergab sich, dass OCEL statistisch signifikant bessere Ergebnisse errechnet hat als CELOE.

In der Tabelle 22 werden die Konzepte aus den Lernprozessen mit der höchsten Genauigkeit angegeben. Des weiteren werden noch Laufzeit und Konzeptlänge angegeben. Deutlich zu sehen ist, dass OCEL und CELOE unterschiedliche Konzepte lernen, um einen Patienten mit Brustkrebs zu beschreiben. Die Bedeutung des von OCEL vorgeschlagenen Konzeptes lautet, dass ein Patient der in eine BIRADS Kategorie von 4,5 oder höher als in mindestens die BIRADS Kategorie 5 eingeordnet wird an Brustkrebs leidet. Diese Klassifikation ist logisch, da wie in Kapitel 3.2.2 dargestellt, Personen die in die BIRADS Kategorie 5 eingeordnet werden, zu über 95% maligne Brustveränderungen besitzen. Da die BIRADS Klassifikation ein wichtiges Einteilungsmerkmal ist, kann das Ergebnis als richtig angesehen werden. Das von CELOE vorgeschlagene Konzept bedeutet, dass ein Patient, der alle im Mammographiebild dargestellten Formen, formwidrig oder lobular sind und das Alter größer gleich 43,5 Jahre ist oder alle Formen formwidrig sind, in die Gruppe der Patienten mit Brustkrebs eingeteilt werden kann. Zu dieser Aussage ist zu sagen, dass das Alter mit 43,5 Jahren fast wie in Kapitel 3.2.1 mit dem höchsten Brustkrebsrisiko übereinstimmt, welches bei 45 bis 65 Jahren liegt. Zu der Aussage, dass formwidrige und lobulare Formen, wurde in der Literatur nichts ausgesagt, es

Algorithmen	p-Wert	statistische Signifikanz
OCEL vs. Naive Bayes	0,4502	nein
OCEL vs. Naive Bayes Updateable	0,4502	nein
OCEL vs. Random Tree	0,1237	nein
OCEL vs. Random Forrest	0,1276	nein
OCEL vs. PART	0,1185	nein
OCEL vs. Decision Table	0,2316	nein
CELOE vs. Naive Bayes	0,0001	ja
CELOE vs. Naive Bayes Updateable	0,0001	ja
CELOE vs. Random Tree	0,0010	ja
CELOE vs. Random Forrest	0,1271	nein
CELOE vs. PART	0,0001	ja
CELOE vs. Decision Table	0,0001	ja
CELOE vs. OCEL	0,0031	ja

Tabelle 21: In dieser Tabelle wird die statistische Signifikanz der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für die Wissensbasis Mammographic dargestellt. Es ist zu sehen, dass der Algorithmus OCEL zwar nur die viert beste Genauigkeit erreicht, aber die Unterschiede sind in allen sechs Fällen nicht statistisch signifikant. Anders sieht es bei CELOE aus. Da ergab der t-Test das CELOE in fünf Fällen statistisch signifikant schlechtere Ergebnisse liefert als die anderen Algorithmen und in einem Fall ist CELOE nicht statistisch signifikant schlechter. Der Vergleich zwischen OCEL und CELOE ergab, dass OCEL statistisch signifikant bessere Ergebnisse errechnet als CELOE.

kann aber davon ausgegangen werden, dass dies ebenfalls stimmt, so dass das ganze Konzept, welches von CELOE vorgeschlagen wurde, sinnvoll erscheint. Das CELOE für das Lernen des Konzeptes sehr viel Zeit beansprucht, kann daran liegen, dass im Suchbaum viele Konzepte gefunden werden, die keine ausreichend hohe Bewertung von CELOE bekommen haben, so dass der Algorithmus sehr viele Konzepte durchsucht hat, bis es ein Konzept gefunden hat, welches den Anforderungen an den Noise-Parameter genügt. Dies ist der Grund für die lange Laufzeit des Algorithmus. OCEL hingegen hat direkt (*Patient AND hasBiRads* ≥ 4.5) gefunden. Da es von OCEL eine ausreichend hohe Bewertung erhalten hat musste der Suchbaum nicht lange durchlaufen werden und die Laufzeit war entsprechend gering.

In der Abbildung 27 ist zu sehen, wie sich die Laufzeit der Algorithmen OCEL und CELOE, durch die Herabsetzung des Noise Paramteres jeweils um 1%, verändert. Dabei ist deutlich zu sehen, dass die Laufzeit von OCEL sehr gering bleibt, bis 19% Noise und die Laufzeit von CELOE schon ab 27% Noise sehr stark ansteigt. Dazu wurde für den größten Laufzeitunterschied ein t-Test durchge-

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasBiRads >= 4.5)</i>	88,542%	0,075s	5
CELOE	<i>(ALL hasShape.(irregular OR lobular) AND (hasAge >= 43.5 OR ALL hasShape.irregular))</i>	88,421%	10270,527s	13

Tabelle 22: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL, CELOE und Weka für die Wissensbasis Mammographic dargestellt.

führt. In diesem Fall ist das bei 23% Noise. Dabei hatte OCEL eine Laufzeit von $0,107s \pm 0,158s$ und CELOE eine Laufzeit von $7512,618s \pm 12374,955s$. Aus diesen Werten ergab sich beim t-Test ein p-Wert von 0,0709. Daraus ergibt sich, dass die Laufzeitunterschiede zwar sehr hoch sind, diese jedoch noch nicht statistisch Signifikant sind, da erst ab p-Werten kleiner 0,05 eine statistische Signifikanz vorliegt. Um herauszufinden warum es einen sehr starken Anstieg in der Laufzeit gibt, wurden die Suchbäume der beiden Algorithmen ausgegeben und verglichen. Dabei ist festzustellen, dass CELOE viel mehr Zeit benötigt, um Konzepte zu finden, die dem Noise-Parameter genügen. Dies liegt daran, dass bei Verringerung des Noise-Parameters, Konzepte, die vorher eine ausreichend hohe interne Bewertung bekommen haben, jetzt als zu Schwach verworfen werden. Deshalb muss der Algorithmus länger suchen um andere Konzepte zu finden die eine ausreichend hohe Genauigkeit besitzen. OCEL hingegen, findet direkt Konzepte welche eine durchschnittliche Genauigkeit von 81,899% besitzen. Aus diesem Grund bleibt die Laufzeit bei OCEL immer bei ca. 0,1 Sekunde. Bei 18 % Noise findet er keine besseren Ergebnisse mehr so das ein Anstieg der Laufzeit hier nicht feststellbar war.

Die in der Abbildung 28 dargestellten Grafik, zeigt die Genauigkeitsunterschiede für Trainings- und Testdaten, der beiden Algorithmen OCEL und CELOE, bei fest eingestellten Laufzeiten von 10, 100, 300 und 1000 Sekunden. Die fehlenden Säulen für OCEL Training und Test, bei 1000 Sekunden Laufzeit, liegt daran, dass der Algorithmus eine *OutOfMemoryException* geworfen hat, und es somit nicht möglich war den Test zu beenden. Dies liegt daran, dass OCEL die nicht abgedeckten positiven und negativen Beispiele in dem Knoten speichert. Aus diesem Grund kommt es zu einer höheren Speicherauslastung, die letztendlich zu der *OutOfMemoryException* geführt hat. In der Grafik ist deutlich zu sehen, dass CELOE Ergebnisse mit einer höheren Genauigkeit erzielt hat wie OCEL. Um eine Erklärung für die Unterschiede in den Genauigkeiten zu erhalten wurde für beide Algorithmen bei 10 Sekunden Laufzeit der Suchbaum ausgegeben. Die Auswertung dieses Suchbaumes hat ergeben, dass die Algorithmen unterschiedliche Konzepte lernen, da die Heuristik die gelernten Konzepte unterschiedlich bewertet. Aufgrund der Unterschiedlichen Bewertung der Konzepte, werden im Suchbaum unterschiedliche Pfade gewählt, die weiterverfolgt werden. Somit kommt es zum Lernen von unterschiedlichen Konzepten der Algorithmen OCEL und CELOE, die eine unterschiedliche Genauigkeit

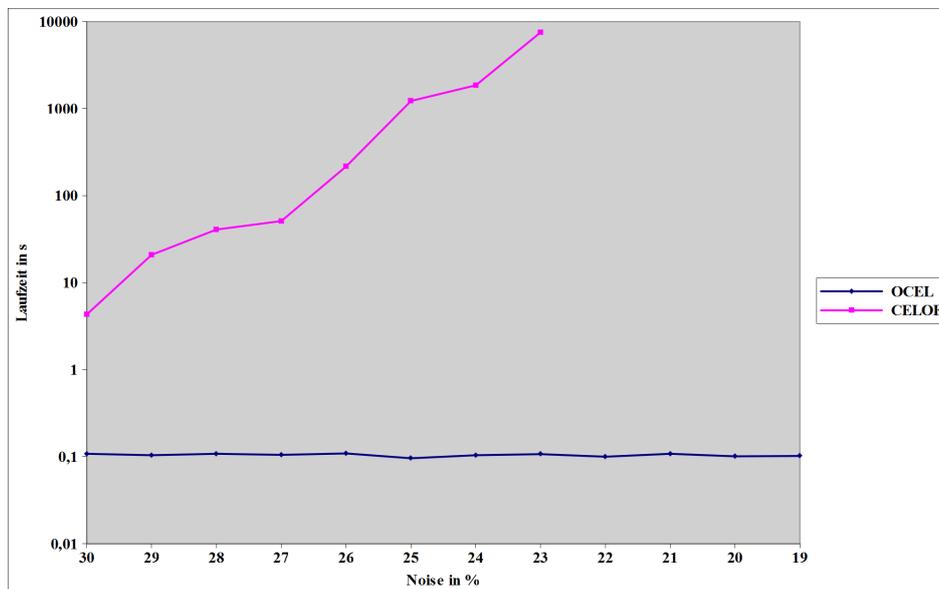


Abbildung 27: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Mammographic Masses Wissensbasis.

besitzen.

Die in der Tabelle 23 dargestellten Ergebnisse, zeigen die Testgenauigkeiten der Algorithmen OCEL und CELOE bei 10, 100 und 300 Sekunden Laufzeit, sowie den p-Wert des t-Tests und eine Aussage ob die Unterschiede bei den Testgenauigkeiten statistisch Signifikant sind. Dabei kann man sehen, dass die Unterschiede der Testgenauigkeit bei den beiden Algorithmen nur sehr gering ist. ebenfalls zu erkennen ist, dass die Auswertung des t-Tests ergeben hat das die Genauigkeitsunterschiede keine statistische Signifikanz aufweisen.

Laufzeit	Testgenauigkeit OCEL	Testgenauigkeit CELOE	p-Wert	statistische Signifikanz
10 Sekunden	65,682%±0,328%	66,387%±2,777%	0,4853	nein
100 Sekunden	65,782%±0,299%	66,49%±3,117%	0,4838	nein
300 Sekunden	65,384%±0,57%	66,49%±3,117%	0,1359	nein

Tabelle 23: Bei dieser Tabelle handelt es sich um die Darstellung der Testgenauigkeit der Algorithmen OCEL und CELOE bei 10, 100 und 300 Sekunden Laufzeit, für die Wissensbasis Mammographic und die Ergebnisse des t-Tests, ob eine statistische Signifikanz in den Unterschieden der Genauigkeiten vorliegt. Dabei ist zu erkennen, dass es bei allen drei Laufzeit Einstellungen keine statistische Signifikanz in den Unterschieden der Genauigkeit gibt.

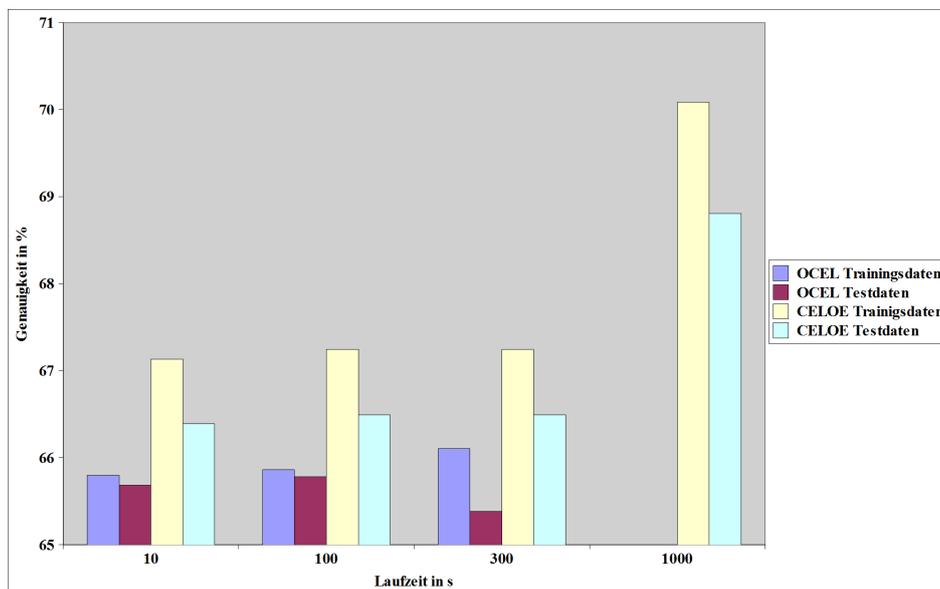


Abbildung 28: Diese Grafik zeigt die Lernergebnisse der Mammographic Wissensbasis in Abhängigkeit des eingestellten Laufzeit Parameter. Dabei ist deutlich zu sehen, dass OCEL bei 1000 Sekunden keine Ergebnisse mehr liefert, denn durch eine *OutOfMemoryException* konnte der Lernvorgang nicht beendet werden. Des Weiteren ist zu sehen, dass CELOE die Ergebnisse mit einer höheren Trainings- und Testgenauigkeit erreicht.

11 Breast Tissue

Dieses Problem beschäftigt sich mit dem weiblichen Brustgewebe beim Menschen und dem Erkennen von Läsionen oder Tumoren mit Hilfe von elektrischer Impedanz Tomographie(EIT). Dazu wurde ein eigener Prototyp erstellt, welcher mit Hilfe von EIT Elektroden, die auf der Brust angebracht werden, Impedanzdaten mit verschiedenen Frequenzen misst.[Jos96] Im folgenden wird der Datensatz, welcher zum Vergleich der verschiedenen Algorithmen verwendet wurde, genauer vorgestellt.

11.1 Datensatz

Der Datensatz, welcher aus dem UCI Machine Learning Repository¹⁸ genommen wurde, besteht aus 106 Instanzdaten mit jeweils 10 verschiedenen Attributen. Diese 106 Instanzen wurden in die 6 Klassen Carcinoma(Car), Fibro-adenoma(Fad), Mastopathy(Mas), Glandular(Gla), Connective(Con) und Adipose(Adi) unterteilt. Dabei bestand die erste Klasse aus 21, die zweite aus 15, die dritte aus 18, die vierte aus 16, die fünfte aus 14 und die sechste aus 22 Instanzen. Das Lernproblem ist also ein 6 Klassen Problem, da aber eine genaue Unterscheidung zwischen den Klassen Fibro-adenoma, Mastopathy und Glandula nicht wichtig ist und ebenfalls nicht genau durchgeführt werden kann, kann dieses Lernproblem ebenfalls als 4 Klassen Problem betrachtet werden. Jede der Instanzen hatte 10 Attribute zur genauen Klassifikation. Da das Attribut *class* nur der Einteilung in die verschiedenen Klassen diene, werden für den Lernvorgang nur die restlichen 9 Attribute verwendet. Im folgenden werden die einzelnen Attribute genauer vorgestellt.

- **I0**

Bei dieser Eigenschaft wird der Impedanzwert, in Ohm, für die Frequenz von 0KHz angegeben.

- **PA500**

Die Eigenschaft PA500 beschreibt den Phasenwinkel bei 500KHz.

- **HFS**

Bei HFS handelt es sich um die Neigung des Phasenwinkels bei hohen Frequenzen.

- **DA**

Bei dieser Eigenschaft handelt es sich um die Impedanzdistanz zwischen den spektralen Enden.

- **AREA**

Die Eigenschaft AREA beschreibt die Gesamtfläche unter dem Spektrum.

¹⁸<http://archive.ics.uci.edu/ml/datasets/Breast+Tissue> Stand:26. April 2011

- **A/DA**

Hierbei handelt es sich um die Normalisierte Fläche, die mit Hilfe der Impedanzdistanz zwischen den spektralen Enden normalisiert wurde.

- **MAX IP**

Hierbei wird das maximal mögliche Spektrum angegeben.

- **DR**

Mit der Eigenschaft DR wird der Abstand zwischen IO und dem Realteil der maximalen Frequenz angegeben.

- **P**

Diese Eigenschaft beschreibt die Länge der spektralen Kurve

- **Class**

Hierbei handelt es sich um die Einteilung der Instanzdaten in die entsprechenden 6 Klassen Carcinoma(Car), Fibro-adenoma(Fad), Mastopathy(Mas), Glandular(Gla), Connective(Con) und Adipose(Adi).

Umwandlung in eine OWL Ontologie

Da der Datensatz in einer xls Datei vorlag mussten die nötigen Informationen erst in eine Textdatei überführt werden. Anschließend wurde mit Hilfe einer eigens geschriebenen Javaklasse die Daten in eine OWL Ontologie überführt. Der Ablauf der Umwandlung ist dabei in der Abbildung 29 dargestellt. Im ersten Schritt wird eine neue OWL Ontologie angelegt und mit *generateDomainAndRangeForObjectProperties()* werden für die einzelnen Objectproperty und Dataproperties die Domain und Range festgelegt. Anschließend wird die Datei in der *main()* Methode eingelesen und die Attribute der einzelnen Instanzdaten an die Methode *mapClause()* übergeben. Diese erstellt die einzelnen Instanzdaten in OWL und weist ihnen ihre Attribute zu. Danach wird nur noch die OWL-Datei geschrieben und die einzelnen Konfigurationsdateien für das Lernen der einzelnen Klassen, mit den Methoden *generateConfFile()* und *generateExamples()*, geschrieben. Ein Klassendiagramm mit den einzelnen Methoden ist in Abbildung 29 zu sehen. Die fertige Ontologie enthält 1 Klasse 9 Data Properties und 106 Instanzdaten.

11.2 Methoden

Für die jeweiligen Klassen, zu denen es Eigenschaften zu lernen gab, wurde eine eigene Konfigurationsdatei geschrieben, welche Informationen über den zu verwendenden Lernalgorithmus und die positiven

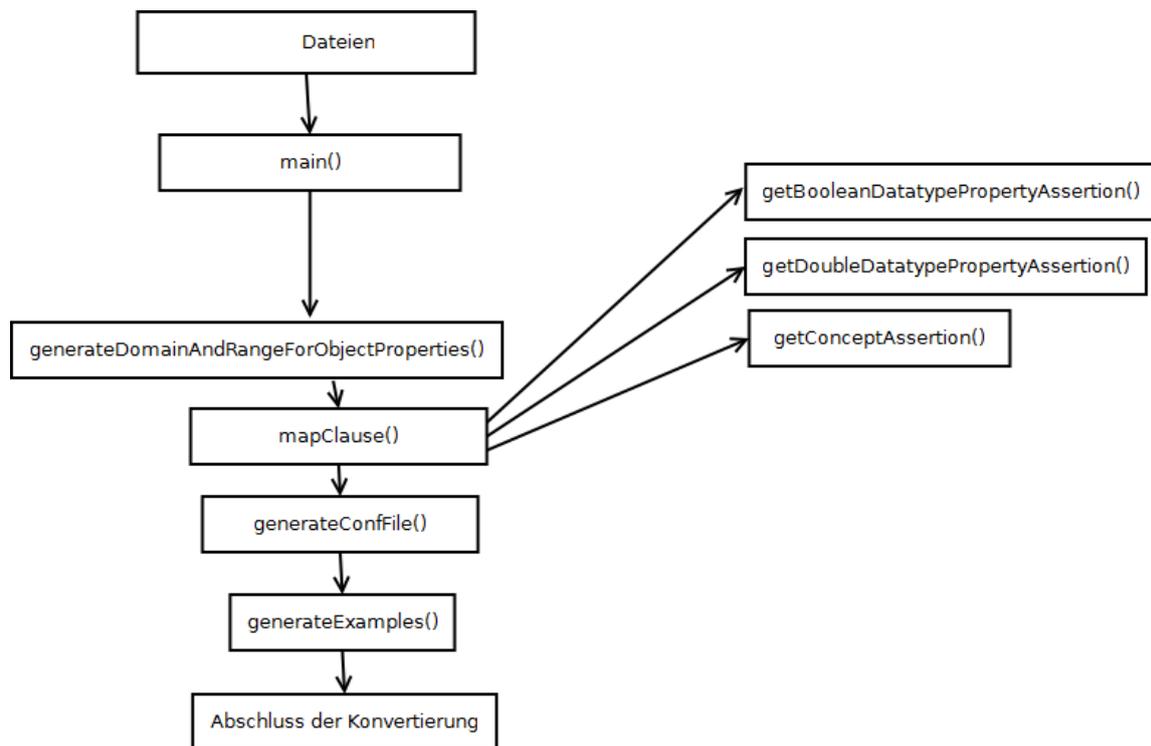


Abbildung 29: Hier ist der Ablauf zur Konvertierung der Prologdateien der Breast Tissue Wissensbasis in eine OWL Ontologie mit ihren Methoden dargestellt.

und negativen Beispiele angegeben. Da es Interessant war zu überprüfen welche Unterschiede es zwischen dem 6 Klassen und dem 4 Klassen Lernproblem es gab, wurden beide Probleme betrachtet. Diese wurden im DL-Learner mit dem Algorithmen OCEL und CELOE untersucht. Dabei wurde der Noise Parameter von 30% in 1% Schritten reduziert und jeweils eine 10-fold-cross-validation durchgeführt. Diese Ergebnisse wurden mit den Algorithmen naive Bayes, naive Bayes Updateable, Random Tree, Random Forrest, PART und Decision Table des Machine Learning Tools Weka verglichen. Da es bei den Algorithmen in Weka keine Möglichkeit gab, einen Noise Parameter zu verändern, wurde nur eine einzelne 10-fold-cross-validation durchgeführt. Des Weiteren wurden für die Algorithmen OCEL und CELOE eine feste Laufzeit von 10, 100, 300 und 1000 Sekunden eingestellt und die Ergebnisse mit Hilfe einer 10-fold-cross-validation verglichen. Da auch hier keine Einstellmöglichkeiten der Laufzeiten der Algorithmen in Weka möglich waren, konnte dieser Test für Weka nicht durchgeführt werden. In einem letzten Test wurde die Laufzeit der beiden Algorithmen OCEL und CELOE verglichen. Auch bei diesem Test wurde der β Parameter von CELOE auf 0.02 gesetzt um einen besseren Vergleich mit OCEL zu erzielen. Dabei wurden alle Tests auf einem Quad Core mit 3.2 GHz und 8GB Ram durchgeführt. Im folgenden Abschnitt werden die Ergebnisse genauer vorgestellt.

11.3 Resultate und Schlussfolgerungen

In der Abbildung 30 sind die Ergebnisse der Lernprozesse für die 6 Klassen der Brest Tissue Wissensbasis dargestellt. Dabei handelt es sich um die Durchschnittsgenauigkeiten aus allen Lernprozessen. Die Einzelergebnisse sind im Anhang die Abbildungen 41 bis 47. Dabei ist zu sehen, dass die Testgenauigkeit bei den Algorithmen des DL-Learners an einigen Stellen abfallen. Dies liegt daran, dass der Algorithmus bei einer oder mehreren Klassen Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten nicht geeignet oder zu spezifisch sind. Des Weiteren ist zu sehen, dass alle Algorithmen des Machine Learning Tool Weka mit Hilfe einer Gerade dargestellt wurden. Das liegt daran, dass es bei den Algorithmen keine Möglichkeit zur Einstellung eines Noise Parameters gab. Aus diesem Grund wurde jeweils nur ein Test durchgeführt und die Ergebnisse als Gerade dargestellt. Deutlich zu sehen ist, dass der Lernalgorithmus CELOE die höchste durchschnittliche Testgenauigkeit erzielt hat. Die zweit höchste Genauigkeit wurde von dem Lernalgorithmus Decision Table des Machine Learning Tool Weka. Der Algorithmus OCEL erreichte die dritt höchste Testgenauigkeit. An vierter Stelle kam der Algorithmus PART und die fünft Beste Genauigkeit erreichte Random Forrest, gefolgt von Random Tree. Die schlechtesten Durchschnittsgenauigkeiten erreichten die beiden Algorithmen naive Bayes und naive Bayes Updateable. Dabei ist anzumerken, dass bei den Algorithmen des DL-Learners bei geringer Laufzeit, nicht für alle Lernprobleme eine Lösung errechnet werden konnte. Bei diesen Fällen wurde nur die durchschnittliche Genauigkeit der verbliebenen Probleme die eine Lösung erzeugten berücksichtigt. Aus diesem Grund erreichen OCEL und CELOE so hohe

Genauigkeiten, da teilweise nur noch von einem Lernproblem die Genauigkeit verwendet wurde. Des Weiteren ist anzumerken, dass die Gerade für die Algorithmen Naive Bayes und Naive Bayes Updateable direkt übereinander liegen, da sie die gleiche Genauigkeit erreicht haben. Aus diesem Grund kann man diese zwei Geraden nicht sehen, sondern nur eine.

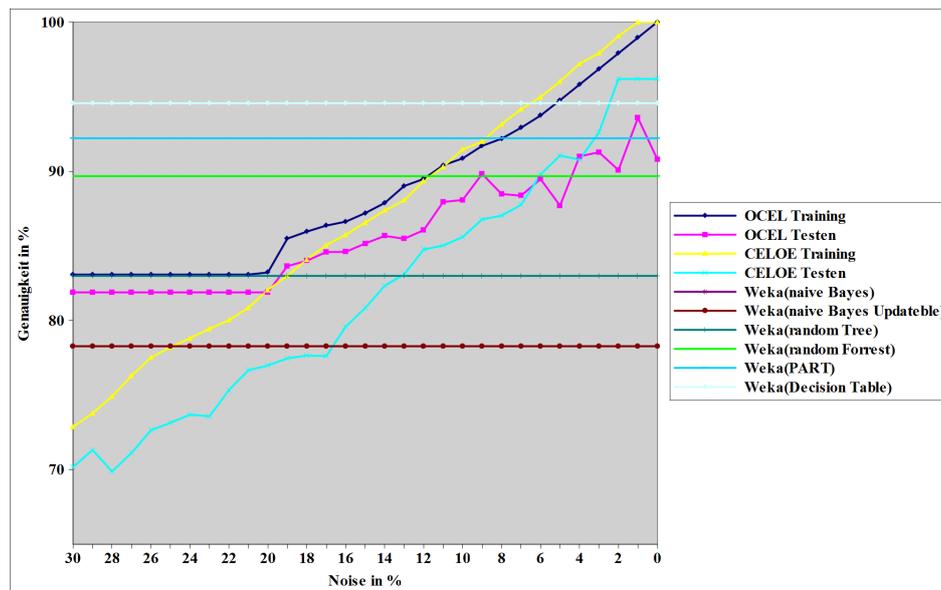


Abbildung 30: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für alle Klassen der Wissensbasis Breast Tissue. Der Abfall der Genauigkeit bei CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind.

In der Tabelle 24 sind die höchsten Genauigkeiten der einzelnen Algorithmen angegeben. Dabei ist deutlich zu sehen, dass OCEL die höchste durchschnittliche Genauigkeit erreicht, dicht gefolgt von CELOE. Die dritt höchste Genauigkeit erzielt der Algorithmus Decision Table und direkt danach kommt der Algorithmus PART. Die fünft höchste Genauigkeit wurde von dem Algorithmus Random Forrest erreicht und daran schließt sich der Algorithmus Random Forrest an. Die schlechteste Genauigkeit erreichen die Algorithmen Naive Bayes und Naive Bayes Updateable. Dabei ist Anzumerken, dass die in der Tabelle 24 dargestellten Werte, Durchschnittsgenauigkeiten sind, welche aus den Genauigkeiten der Einzelergebnissen errechnet wurden.

Bei der Tabelle 25, handelt es sich um die aus der Tabelle 24 errechneten p-Werte des t-Tests und eine Anmerkung, ob diese Werte statistisch signifikant sind. Dabei ist deutlich zu erkennen, dass OCEL und CELOE nur bei den Algorithmen Naive Bayes und Naive Bayes Updateable statistisch signifikant bessere Ergebnisse liefern. OCEL erreicht in drei weiteren Fällen nicht statistisch signifikant bessere Ergebnisse und in einem Fall nicht statistisch signifikant schlechtere Ergebnisse. Die weiteren Ergeb-

Algorithmus	Testgenauigkeit
OCEL	93,576%±8,231%
CELOE	96,182%±6,539%
Naive Bayes	78,27%±15,96%
Naive Bayes Updateable	78,27%±15,96%
Random Tree	82,983%±25,901%
Random Forrest	89,668%±18,56%
PART	92,208%±17,41%
Decision Table	94,556%±13,985%

Tabelle 24: In dieser Tabelle werden die höchsten durchschnittlichen Genauigkeiten der 10-fold-cross-validation der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für die Wissensbasis Breast Tissue dargestellt, die aus den Einzelergebnissen der Lernprobleme entstanden sind.

nisse des t-Tests von CELOE zeigen, dass CELOE im Vergleich mit den anderen vier Algorithmen nicht statistisch signifikant bessere Ergebnisse liefert. Der direkten Vergleich der beiden Algorithmen OCEL und CELOE, ergab das CELOE nicht statistisch signifikant besser arbeitet als OCEL. Anzumerken ist, dass in der Tabelle 24, die p-Werte der durchschnittlichen Genauigkeitswerte darstellen, welche aus den sechs Klassen errechnet wurden.

Die Abbildung 31 wird die durchschnittliche Laufzeit der Algorithmen OCEL und CELOE dargestellt. Dabei handelt es sich um den mittelwert aus den Lernergebnissen der 6 Klassen Carcinoma, Fibro-adenoma, Mastopathy, Glandular, Connective und Adipose. Dabei ist zu erkennen, dass die Laufzeit bei CELOE und OCEL an einigen Stellen abfällt. Das liegt daran, dass bei Abfall der Laufzeit für mindestens eine Klasse keine Konzepte mehr gelernt wurden. Das hat zur Folge, dass die Laufzeiten, die vorher etwas höher ausgefallen sind wegfallen und somit keinen Einfluss mehr auf die Durchschnittswertung haben. Aus diesem Grund kommt es bei OCEL und CELOE zu den Schwankungen in den durchschnittlichen Laufzeiten. In der Grafik ist weiterhin zu sehen, dass OCEL am Anfang eine geringere Laufzeit als CELOE besitzt und am Ende eine höhere durchschnittliche Laufzeit aufweist. Dabei betrug die höchste Laufzeit bei OCEL $254,44s \pm 179,907s$ und bei CELOE betrug die höchste Laufzeit $49,93s \pm 32,453s$. Um herauszufinden ob diese Unterschiede statistisch Signifikant sind wurde ein t-Test durchgeführt. Der errechnete p-Wert beträgt 0,0024. Das bedeutet, dass dieser Unterschied in den Laufzeiten nicht nur sehr groß ist, sondern das dieser Unterschied auch statistisch signifikant ist. Dieser Unterschied in den Genauigkeiten lässt sich dadurch erklären, dass die Algorithmen eine unterschiedlich lange Zeit benötigen um ein Konzept zu finden, welches dem Noise-Parameter genügt. Des Weiteren braucht der Algorithmus OCEL auch keine Reasoneranfragen, wegen den nicht abgedeckten positiven und negativen Beispielen stellen, da diese mit im Suchbaum

Algorithmen	p-Wert	statistische Signifikanz
OCEL vs. Naive Bayes	0,0148	ja
OCEL vs. Naive Bayes Updateable	0,0148	ja
OCEL vs. Random Tree	0,2336	nein
OCEL vs. Random Forrest	0,5503	nein
OCEL vs. PART	0,8248	nein
OCEL vs. Decision Table	0,8507	nein
CELOE vs. Naive Bayes	0,0041	ja
CELOE vs. Naive Bayes Updateable	0,0041	ja
CELOE vs. Random Tree	0,1356	nein
CELOE vs. Random Forrest	0,3090	nein
CELOE vs. PART	0,5078	nein
CELOE vs. Decision Table	0,7429	nein
CELOE vs. OCEL	0,4433	nein

Tabelle 25: In dieser Tabelle wird die statistische Signifikanz der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für die Wissensbasis Breast Tissue dargestellt. Dabei ist deutlich zu sehen, dass nur der Unterschied von OCEL bzw. CELOE und Naive Bayes bzw. Naive Bayes Updateable statistisch signifikant ist. Die anderen Unterschiede sind nicht statistisch signifikant.

gespeichert werden. Dies spart etwas Zeit und somit ist der Algorithmus OCEL am Anfang etwas schneller als CELOE. Es ist anzumerken, dass die Abbildung 31 eine Durchschnittslaufzeit der Algorithmen darstellt, die aus den Abbildungen 48 bis 53 errechnet wurde. Diese Abbildungen sind die Laufzeiten, die für das Lernen der 6 Klassen erreicht wurden und befinden sich im Anhang der Arbeit.

In der Abbildung 32 werden die durchschnittlichen Genauigkeiten der Algorithmen OCEL und CELOE, für 10, 100, 300 und 1000 Sekunden Laufzeit dargestellt. Dabei ist zu sehen, dass bei 10 und 100 Sekunden Laufzeit der Algorithmus OCEL eine höhere Durchschnittsgenauigkeit erreicht hat als CELOE. Bei 300 Sekunden Laufzeit erreicht CELOE die höhere Genauigkeit und bei 1000 Sekunden erzielt wieder OCEL die höhere Genauigkeit. Das die beiden Algorithmen OCEL und CELOE eine geringere Durchschnittsgenauigkeit als bei dem Test mit der Verringerung des Noise-Parameters (siehe Abbildung 30) haben, liegt daran, dass bei dem Test mit der fest eingestellten Laufzeit, keine Konzepte wegen zu geringer interner Genauigkeit verworfen werden. Deshalb müssen mehr Konzepte durchsucht werden und es kommt dazu, dass die Algorithmen eine nicht so hohe Genauigkeit erreichen. Aus diesem Grund erzielen die Algorithmen bei fest eingestellter Laufzeit nicht so hohe Genauigkeitswerte, wie bei eingestelltem Noise-Parameter. In der Abbildung 32 ist deutlich zu sehen, dass Testgenauigkeiten, bei allen vier eingestellten Laufzeiten von 10, 100, 300 und 1000 Sekunden,

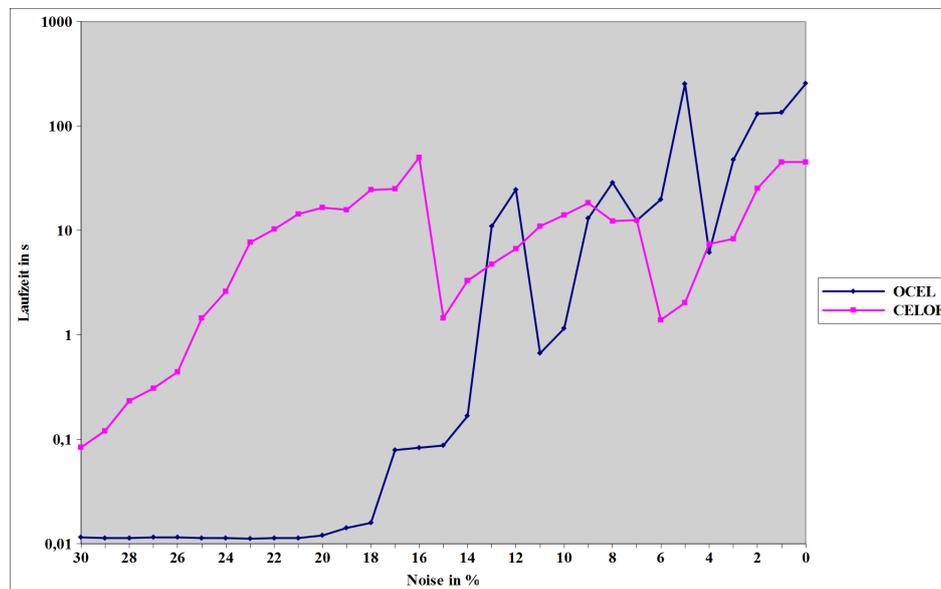


Abbildung 31: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für alle Klassen der Wissensbasis Breast Tissue

nur sehr gering unterschiedlich sind. Dies ist darauf zurück zu führen, dass diese Abbildung eine Zusammenfassung der Abbildungen 55 bis 60 darstellt. Die Abbildungen befinden sich im Anhang der Arbeit. Bei diesen Abbildungen kam es zu deutlichen Schwankungen in den erreichten Genauigkeiten für die einzelnen Klassen, so dass der daraus gebildete Durchschnittswert ca. gleich bleibt. Diese Schwankung in den Genauigkeiten der einzelnen Klassen lässt sich darauf zurückführen, dass es bei diesem Lernproblem einige Klassen gab, bei der nur eine geringere Genauigkeit erzielt wurde. Dabei konnte der DL-Learner keine Konzepte mit einer höheren Genauigkeit finden, da es keine Konzepte gab, die mehr positive Beispiele abgedeckt und negative Beispiele nicht abgedeckt hat.

In der Tabelle 26 sind die Genauigkeiten der Algorithmen OCEL und CELOE dargestellt. Des Weiteren sind die aus diesen Genauigkeiten errechneten p-werte des t-Tests und eine Aussage ob dieser Wert statistisch signifikant ist, angegeben. Es ist zu sehen, dass die Unterschiede der durchschnittlichen Genauigkeiten nur minimal sind. Des Weiteren ist zu sehen, dass die Unterschiede der Genauigkeiten für die Algorithmen OCEL und CELOE für die vier gemessenen Laufzeiten 10, 100, 300 und 1000 Sekunden, nicht statistisch signifikant sind. Es ist anzumerken, dass die in der Tabelle 26 angegebenen Genauigkeiten Durchschnittswerte sind, die aus den Ergebnissen der Einzeltests errechnet wurden.

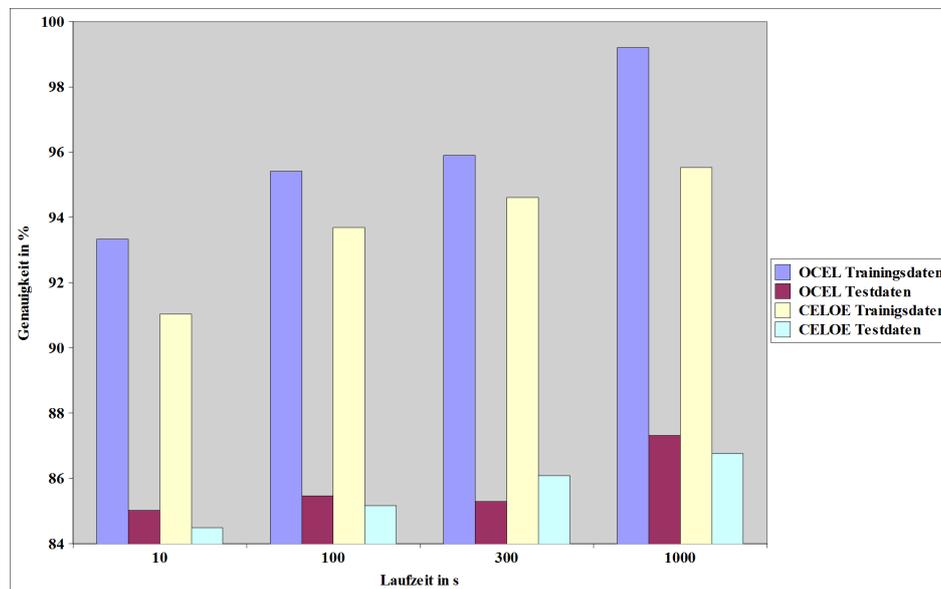


Abbildung 32: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für alle Klassen der Wissensbasis Breast Tissue.

Resultate der einzelnen Lernprozesse

Um zu überprüfen ob die beiden Algorithmen des DL-Learners ähnliche Konzepte für die einzelnen Klassen vorschlagen oder in wie weit sich diese unterscheiden wird in diesem Abschnitt geklärt. Dabei wird zuerst auf die Unterschiede in der Klasse Carcinoma eingegangen.

Carcinoma

In Tabelle 27 werden die gelernten Konzepte der Algorithmen OCEL und CELOE dargestellt. Dabei bedeutet das Konzept von OCEL, dass ein Patient mit einem Imedanzwert bei 0Khz der kleiner gleich 501 ist und ein maximal mögliches Spektrum, das größer gleich 50,95 ist, in die Klasse Carcinoma eingeteilt wird. Das Konzept, welches von CELOE vorgeschlagen wurde, ist fast das gleiche nur das das Attribut *hasMaximumOfTheSpectrum* größer gleich 34,04 ist. Diese minimalen Unterschiede zwischen den beiden Algorithmen lässt sich darauf zurückführen, dass OCEL und CELOE bei der Berechnung des Attributwertes unterschiedliche Trainingsdaten genutzt hat, die leicht unterschiedliche Attributwerte besitzen, so dass die Unterschiede davon kommen.

Laufzeit	Testgenauigkeit OCEL	Testgenauigkeit CELOE	p-Wert	statistische Signifi- kanz
10 Sekunden	85,018%±9,196%	84,492%±10,925%	0,9086	nein
100 Sekunden	85,466%±10,435%	85,162%±10,527%	0,9490	nein
300 Sekunden	85,299%±10,64%	86,086%±9,848%	0,8656	nein
1000 Sekunden	87,318%±10,326%	86,760%±9,951%	0,9034	nein

Tabelle 26: Bei dieser Tabelle handelt es sich um die Darstellung der Testgenauigkeit der Algorithmen OCEL und CELOE bei 10, 100, 300 und 1000 Sekunden Laufzeit, für die Wissensbasis Breast Tissue und die Ergebnisse des t-Tests, ob eine statistische Signifikanz in den Unterschieden der Genauigkeiten vorliegt. Dabei ist zu erkennen, dass für alle vier eingestellten Laufzeiten keine statistische Signifikanz vorlag.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasImpedivityAtZeroFrequency <= 501.0 AND hasMaximumOfTheSpectrum >= 50.95)</i>	100%	333,943s	9
CELOE	<i>(hasImpedivityAtZeroFrequency <= 501.0 AND hasMaximumOfTheSpectrum >= 34.04)</i>	100%	0,031s	7

Tabelle 27: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der Klasse Carcinoma für die Wissensbasis Breast Tissue dargestellt. Dabei ist deutlich zu sehen, dass die beiden Algorithmen fast die gleichen Konzepte lernen, nur der Attributwert bei *hasMaximumOfTheSpectrum* ist leicht unterschiedlich.

Fibro-Adenoma

In Der Tabelle 28 werden die gelernten Konzepte mit der höchsten Genauigkeit, der Algorithmen OCEL und CELOE dargestellt. Das Konzept von OCEL bedeutet, dass ein Patient mit einem Abstand zwischen dem Impedanzwert bei 0Khz Frequenz und dem Realteil der maximal Frequenz von 13,585 oder größer und eine Impedanzwert bei 0Khz von 260,085 und einen Phasenwinkel bei 500KHz, der größer gleich 0,185 ist oder einen Phasenwinkel bei 500Khz der kleiner gleich 0,08499 ist, in die Klasse Fibro-Adenoma eingeteilt wird. Der Vorschlag von CELOE bedeutet, dass ein Patient der einen Abstand zwischen dem Impedanzwert bei 0Khz Frequenz und dem Realteil der maximal Frequenz von 64,115 und kleiner und eine Länge der spektralen Kurve, die kleiner gleich 283,79 ist oder einen Phasenwinkel bei 500Khz, der größer gleich 0,084999 ist, besitzt, wird in die Klasse Fibro-Adenoma eingeteilt. Die beiden Algorithmen benötigen für die Beschreibung der Klasse die Attribute *has-*

DistanceBetweenI0AndRealPartOfTheMaximumFrequencyPoint und *hasPhaseAngleAt500KHz*, nur bei *hasDistanceBetweenI0AndRealPartOfTheMaximumFrequencyPoint* kamen unterschiedliche Attributsausprägungen heraus, was auf Unterschiedliche Trainingsdaten zugeführt werden kann. Des Weiteren haben OCEL und CELOE ein Attribut, welches unterschiedlich ist für die Beschreibung der Klasse Fibro-Adenoma heraus. Dies lässt sich darauf zurückführen, dass die Algorithmen bei der Bewertung der Konzepte unterschiedliche interne Genauigkeiten vergeben haben. Dadurch werden unterschiedliche Pfade weiterverfolgt, so dass unterschiedliche Teilkonzepte heraus kamen.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasDistanceBetweenI0AndRealPartOfTheMaximumFrequencyPoint >= 13.585 AND hasImpedivityAtZeroFrequency <= 260.085 AND (hasPhaseAngleAt500KHz >= 0.185 OR hasPhaseAngleAt500KHz <= 0.0849999999999999))</i>	100%	0,114s	17
CELOE	<i>(hasDistanceBetweenI0AndRealPartOfTheMaximumFrequencyPoint <= 64.155 AND (hasLengthOfTheSpectralCurve <= 283.79 OR hasPhaseAngleAt500KHz >= 0.0849999999999999))</i>	100%	0,773s	11

Tabelle 28: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der Klasse Fibro-Adenoma für die Wissensbasis Breast Tissue dargestellt. Es ist zu erkennen, dass die beiden Algorithmen unterschiedliche Konzepte lernen und das Konzept von CELOE deutlich länger ist als das von OCEL.

Mastopathy

In der Tabelle 29 werden die von den Algorithmen OCEL und CELOE gelernten Konzepte vorgestellt, mit entsprechender Genauigkeit, Laufzeit und Konzeptlänge. Das Konzept von OCEL bedeutet, dass ein Patient mit einem Impedanzwert bei 0Khz, der kleiner gleich 213,70499 und einem Maximum des Spektrums von 34,04 oder größer, in die Klasse Mastopathy eingeteilt wird. Wohingegen das Konzept von CELOE bedeutet, dass eine Person mit einer durch die Impedanzdistanz zwischen den spektralen Enden normalisierten Wert von 33,28 oder größer und einem Impedanzwert bei 0Khz Frequenz von 501 oder kleiner und einem Phasenwinkel bei 500KHz von 0,135 oder kleiner oder ein maximales Spektrum das größer gleich 34,04 ist und einen Phasenwinkel bei 500Khz von 0,185 oder

kleiner hat, in die Klasse Mastopathy eingeteilt wird. Es ist zu sehen, dass OCEL und CELOE die gemeinsamen Teilkonzepte *hasImpedivityAtZeroFrequency* und *hasMaximumOfTheSpectrum* besitzen, nur bei *hasImpedivityAtZeroFrequency* kommt es zu einer unterschiedlichen Attributsausprägung. Dies lässt sich dadurch erklären, dass die Trainingsdaten unterschiedlich waren und eine unterschiedliche Attributausprägung für *hasImpedivityAtZeroFrequency* hatten. Dadurch schlägt der Algorithmus unterschiedliche Attributsausprägungen vor. Das der Algorithmus CELOE weitere Attribute benötigt um die Klasse Mastopathy zu beschreiben, liegt daran, dass in dem Suchbaum die Bewertung für das kürzere Konzept noch nicht ausreichend war, so dass noch Attribute benötigt wurden, um die Bewertung zu erhöhen. Auf Grund dessen, kam es zu einem deutlich längeren Konzept bei CELOE.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasImpedivityAtZeroFrequency <= 213.7049999999998 AND hasMaximumOfTheSpectrum >= 34.04)</i>	90,909%	0,031s	9
CELOE	<i>(hasImpedanceDistanceBetweenSpectral >= 33.28 AND hasImpedivityAtZeroFrequency <= 501.0 AND (hasPhaseAngleAt500KHz <= 0.135 OR (hasMaximumOfTheSpectrum >= 34.04 AND hasPhaseAngleAt500KHz <= 0.185)))</i>	100%	–	19

Tabelle 29: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der Klasse Mastopathy für die Wissensbasis Breast Tissue dargestellt. Dabei ist zu sehen, dass OCEL das kürzere Konzept lernt und die Attribute die bei OCEL vorkommen auch bei dem Vorschlag von CELOE verwendet werden. Nur die Attributausprägung für *hasImpedivityAtZeroFrequency* ist unterschiedlich.

Glandular

Die in der Tabelle 30 dargestellten Konzepte, mit entsprechenden Genauigkeiten, Laufzeiten und Konzeptlängen, stellen die Vorschläge der Algorithmen OCEL und CELOE dar, um die Klasse Glandular zu beschreiben. Dabei bedeutet das von OCEL vorgeschlagene Konzept, dass ein Patient mit einem Impedanzwert der kleiner gleich 260,085 ist und ein maximales Spektrum von 18,345 und größer und einen Phasenwinkel bei 500Khz von 0,155 oder weniger hat, in die Klasse Glandular eingeteilt wird. Da der Teil *hasPhaseAngleAt500KHz <= 0.155 OR hasPhaseAngleAt500KHz <= 0.155*

gleich ist bräuchte man *hasPhaseAngleAt500KHz* ≤ 0.155 nur einmal und das würde die Konzeptlänge auf 13 reduzieren. Das Konzept von CELOE hingegen bedeutet, dass ein Patient mit einem Impedanzwert der kleiner gleich 260,085, in die Klasse Glandular eingeordnet wird. Der Algorithmus CELOE brauchte somit nur den Teil *hasImpedivityAtZeroFrequency* ≤ 260.085 um die Klasse Glandular zu beschreiben, während OCEL noch zusätzlich den Teil *hasMaximumOfTheSpectrum* ≥ 18.345 AND (*hasPhaseAngleAt500KHz* ≤ 0.155 OR *hasPhaseAngleAt500KHz* ≤ 0.155) benötigt um die Klasse zu beschreiben. Das CELOE im Gegensatz zu OCEL ein viel kürzeres Konzept vorschlägt, lässt sich dadurch erklären, dass die Algorithmen im Suchbaum die Konzepte unterschiedlich bewerten, so dass bei CELOE die kürzere Beschreibung schon eine ausreichende Genauigkeit besaß. Im Gegensatz dazu muss das Konzept viel Länger sein, um mit einer ausreichend hohen Genauigkeitsbewertung versehen zu werden, da OCEL das kürzere Konzept nicht so hoch bewertet hat. Auf Grund diesen Unterschieden in der Bewertung, kommt es zu den unterschiedlich vorgeschlagenen Konzepten.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasImpedivityAtZeroFrequency</i> ≤ 260.085 AND <i>hasMaximumOfTheSpectrum</i> ≥ 18.345 AND (<i>hasPhaseAngleAt500KHz</i> ≤ 0.155 OR <i>hasPhaseAngleAt500KHz</i> ≤ 0.155))	100%	0,123s	17
CELOE	<i>hasImpedivityAtZeroFrequency</i> ≤ 260.085	100%	0,01s	3

Tabelle 30: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der Klasse Glandular für die Wissensbasis Breast Tissue dargestellt. Hier ist deutlich zu erkennen, dass CELOE das kürzer Konzept lernt und dass das Konzept von CELOE auch in OCEL enthalten ist, nur das OCEL noch weitere Attribute benötigt um die Klasse zu beschreiben.

Connective

In der Tabelle 31 werden die Konzepte mit entsprechender Genauigkeit, Laufzeit und Konzeptlänge dargestellt, welche von den Algorithmen OCEL und CELOE zum Lernen der Klasse Connective verwendet werden. das Konzept von OCEL bedeutet, dass ein Patient mit einer Impedanzdistanz zwischen den spektralen Enden normalisierten Wert von 9,129 oder kleiner und einem maximalen Spektrum von 23,3849 oder größer und einem Phasenwinkel bei 500Khz von 0,0849 oder kleiner in die Klasse Connective eingeteilt werden kann. Es fällt wieder auf, dass das Teilstück *hasPhase-*

AngleAt500KHz ≤ 0.0849999999999999 doppelt vorkommt. Dabei würde es genügen wenn diese Eigenschaft einmal vorkommt und sich die Konzeptlänge somit auf 13 verkürzen könnte. Wohingegen das Konzept von CELOE bedeutet, dass ein Patient mit einer Impedanzdistanz zwischen den spektralen Enden normalisierten Wert von 23,13 oder kleiner und einer Impedanzdistanz zwischen den spektralen Enden von 147,4549 oder größer, in die Klasse Connective eingeteilt werden kann. Bei diesem beiden Vorschlägen fällt auf, dass nur das Attribut *hasAreaNormalizedByDA* in beiden Konzepten vorkommt. Die restliche Beschreibung ist unterschiedliche. Dies kann daran liegen, dass zum Lernen der Klasse unterschiedliche Trainingsdaten verwendet wurden, bei denen die Attribute unterschiedlich ausgeprägt sind. Dadurch kam es zu unterschiedlichen Pfaden, die im Suchbaum weiterverfolgt wurden, so dass die vorgeschlagenen Konzepte sich unterscheiden. Auf Grund dieser Unterschiede kam es zu den unterschiedlichen Vorschlägen bei den Algorithmen OCEL und CELOE.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasAreaNormalizedByDA ≤ 9.12999999999999 AND hasMaximumOfTheSpectrum ≥ 23.38499999999998 AND (hasPhaseAngleAt500KHz ≤ 0.0849999999999999 OR hasPhaseAngleAt500KHz ≤ 0.0849999999999999))</i>	100%	0,074s	17
CELOE	<i>(hasAreaNormalizedByDA ≤ 23.130000000000003 AND hasImpedanceDistanceBetweenSpectralEnds ≥ 147.45499999999998)</i>	100%	0,406s	7

Tabelle 31: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der Klasse Connective für die Wissensbasis Breast Tissue dargestellt.

Adipose

Die in der Tabelle 32 dargestellten Konzepte mit entsprechenden Genauigkeiten, Laufzeiten und Konzeptlängen, stelle die Konzepte von OCEL und CELOE dar, die zur Beschreibung der Klasse Adipose dienen. Dabei bedeutet das Konzept von OCEL, dass ein Patient mit einem maximalen Spektrum, dass größer gleich 81,015 ist und einem Phasenwinkel bei 500Khz, der kleiner gleich 0,0849 ist, in die Klasse Adipose eingeordnet werden kann. Dieses Konzept von CELOE, dass ein Patient mit einem maximalen Spektrum von 50,95 oder größer und einem Phasenwinkel bei 500 Khz von 0,135

oder kleiner in die Klasse Adipose eingeordnet werden können. Die beiden Algorithmen OCEL und CELOE schlagen hierbei die gleichen Eigenschaften zur Beschreibung der Klasse Adipose vor, nur die Wertebereiche der Beschreibungen unterscheiden sich bei den beiden Algorithmen. Diese Unterschiede in den Attributsausprägungen lassen sich dadurch erklären, dass zum Lernen der Klassenbeschreibung unterschiedliche Trainingsdaten zur Verfügung standen. Diese Trainingsdaten hatten leicht andere Attributsausprägungen, so dass die Unterschiede beim Lernen von diesen unterschiedlichen Trainingsdaten kommt.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasMaximumOfTheSpectrum >= 81.015 AND hasPhaseAngleAt500KHz <= 0.0849999999999999)</i>	100%	0,049s	9
CELOE	<i>(hasMaximumOfTheSpectrum >= 50.95 AND hasPhaseAngleAt500KHz <= 0.135)</i>	–	–	7

Tabelle 32: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der Klasse Adipose für die Wissensbasis Breast Tissue dargestellt. Hier ist zu sehen, dass sowohl OCEL als auch CELOE fast das gleiche Konzept lernt, nur mit unterschiedlichen Attributsausprägungen.

Zusammenfassung der Klassen Fibro-Adenoma, Mastopathy und Glandular

In der Tabelle 33 werden die Konzepte, welche OCEL und CELOE für die Zusammenfassung der Klassen Fibro-Adenoma, Mastopathy und Glandular gelernt hat, mit entsprechender Genauigkeit, Laufzeit und Konzeptlänge. Dabei bedeutet das Konzept von OCEL, dass ein Patient mit einem Impedanzwert bei 0Khz von 353,83 oder kleiner zu diesen drei Klassen gehört. Das Konzept von CELOE bedeutet, dass ein Patient mit einem Impedanzwert der kleiner gleich 400,5 ist zu den drei Klassen Fibro-Adenoma, Mastopathy und Glandular gehört. Auch bei diesem Problem lernen OCEL und CELOE die gleichen Attribute, nur die Wertebereiche unterscheiden sich wieder bei beiden Algorithmen. Diese unterschiedlichen Attributsausprägungen lassen sich dadurch erklären, dass unterschiedliche Trainingsdaten verwendet wurden, die in den verschiedenen Folds verwendet wurden. Diese können leicht unterschiedliche Attributausrprägungen besitzen, so dass sich der Unterschied dadurch erklären lässt.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasImpedivityAtZeroFrequency <= 353.83000000000004)</i>	100%	0,009s	5
CELOE	<i>(hasImpedivityAtZeroFrequency <= 400.5)</i>	100%	0,012s	3

Tabelle 33: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE für die Zusammenfassung der Klassen Fibro-Adenoma, Mastopathy und Glandular für die Wissensbasis Breast Tissue dargestellt. Dabei ist zu erkennen, dass sowohl OCEL als auch CELOE das gleiche Konzept lernt, nur mit leicht unterschiedlichen Attributsausprägungen.

12 Cardiotocography

Bei dieser Wissensbasis handelt es sich um Kardiogramme von Föten, die zur Untersuchung dienen, ob der Fötus gesund ist.

12.1 Datensatz

Dieser Datensatz stammt aus dem UCI Machine Learning Repository ¹⁹ und besteht aus 2126 Kardiogrammen von Föten mit 23 Attributen. Von diesen 23 Attributen wurden 21 zur Klassifikation verwendet. Die letzten beiden Eigenschaften dienen der Einteilung in die einzelnen Klassen. Da dieses Lernproblem sowohl in ein 10 Klassen Lernproblem als auch in ein 3 Klassen Lernproblem eingeteilt werden kann, sind zwei Attribute zur Einteilung nötig. Im folgenden werden die Attribute genauer vorgestellt.

- textbfLB

Bei diesem Attribut handelt es sich um die Herzschläge des Fötus pro Minute.

- textbfAC

Diese Eigenschaft beschreibt die Anzahl der Erhöhung der Herzschläge pro Sekunde.

- textbfFM

Hiermit wird die Anzahl der Bewegungen des Fötus pro Sekunde angegeben.

- textbfUC

Dieses Attribut beschreibt die Anzahl der Kontraktionen der Gebärmutter pro Sekunde.

- textbfDL

Bei dieser Eigenschaft handelt es sich um die Anzahl der leichten Verringerungen der Herzschläge pro Sekunde.

- textbfDS

Hierbei handelt es sich um das Attribut, welches die Anzahl der schwerwiegenden Verringerungen der Herzschläge pro Sekunde angibt.

- textbfDP

Dieses Attribut gibt die Anzahl der anhaltenden Verringerungen der Herzschläge pro Sekunde an.

¹⁹<http://archive.ics.uci.edu/ml/datasets/Cardiotocography> Stand:26. April 2011

- textbfASTV

Hiermit wird die Zeit in der abnormale Kurzzeitveränderungen der Herzschläge auftreten, in Prozent angegeben.

- textbfMSTV

Diese Eigenschaft beschreibt den Mittelwert der Kurzzeitveränderung der Herzschläge an.

- textbfALTV

Durch dieses Attribut wird die Zeit, in der abnormale Langveränderungen der Herzschlagfrequenz auftreten, in Prozent angegeben.

- textbfMLTV

Hierbei handelt es sich um das Attribut, welches den Mittelwert der Langzeitveränderung der Herzschlagfrequenz angibt.

- textbfWidth

Das Attribut gibt die Breite der Herzrate eines Fötus an

- textbfMin

Mit Hilfe dieses Attributs wird das Minimum der Herzrate des Fötus angegeben.

- textbfMax

Hiermit wird das Maximum der Herzrate des Fötus angegeben.

- textbfNmax

Diese Eigenschaft gibt die Anzahl wie oft die maximale Herzfrequenz erreicht wird an.

- textbfNzeros

Hiermit wird die Anzahl angegeben, wie oft die Nullherzfrequenz erreicht wird.

- textbfMode

Diese Eigenschaft gibt den häufigsten Wert an, der im Kardiogramm erreicht wurde.

- textbfMean

Hiermit wird der im Kardiogramm erreichte Durchschnittswert angegeben.

- textbfMedian

Dieses Attribut beschreibt den Mittelwert, der im Kardiogramm erreicht wurde.

- textbfVariance

Diese Eigenschaft gibt die Varianz der Werte die im Kardiogramm erreicht wurden an.

- textbfTendency

Hierbei handelt es sich um die Tendenz des Kardiogramms, ob es gleich bleiben, fallend oder steigend ist.

- textbfCLASS

Einteilung der Patienten in die zehn Klassen

- textbfNSP

Hiermit wird die Einteilung der Patienten in die Klassen Normal, Suspect und Pathologic vorgenommen.

Umwandlung in eine OWL Ontologie

Da die Wissensbasis nicht in einer OWL Ontologie vorlag, sondern in einer Excel Tabelle, mit der der DL-Learner nicht zurecht kommt, musste die Wissensbasis in eine OWL Ontologie umgewandelt werden. Im folgenden wird erläutert, wie dies gemacht wurde. Zuerst wurden alle Daten aus dieser Excel Tabelle kopiert und in einer Datei gespeichert. Anschließend wurde ein Parser geschrieben, der die Daten in OWL Ausdrücke umwandelt. Dieser Parser wurde in Java geschrieben. In der Abbildung 33 ist der Ablauf der Umwandlung mit den wichtigsten Methoden dargestellt. Diese werden im Folgenden genauer beschrieben. Für den gesamten Ablauf der Umwandlung ist die methode *main()* verantwortlich. In dieser wird in einem ersten Schritt eine neue Ontologie angelegt und mit Hilfe der Methode *generateDomainAndRangeForObjectProperties()* die Domain und Range der Object und Data Properties festgelegt. Im Anschluss daran wird die Wissensbasis zeilenweise eingelesen und mit der Methode *mapClauses()* in die entsprechenden OWL Ausdrücke umgewandelt. Dabei verwendet die Methode *mapClauses()* weitere Methoden um die Umwandlung durchzuführen, nämlich hauptsächlich die Methode *getDoubleDatatypePropertyAssertion()*, die die einzelnen Werte in eine Double Datatype Property umwandelt. Im Anschluss daran, werden die einzelnen Instanzdaten für die jeweiligen Klassen entweder als positive oder als negative Beispiele gespeichert. Wenn die Umwandlung abgeschlossen ist, werden die OWL Ausdrücke in die vorher angelegte Ontologie gespeichert und die Konfigurationsdateien für den DL-Learner geschrieben. Das Schreiben der Konfigurationsdateien geschieht in 2 Schritten. Im ersten Schritt wird der Kopf der Konfigurationsdatei geschrieben, der Informationen über verwendeten Lernalgorithmus, Reasoner und Lernproblem enthält. Im zweiten Schritt werden die positiven und negativen Beispielen für den Lernvorgang angehängt. Dabei entstehen 13 Konfigurationdateien, jeweils eine für die entsprechende Klasse die gelernt werden soll. Wenn

dieser Vorgang abgeschlossen ist, ist die Umwandlung fertig und es entstand eine OWL Ontologie mit einer Klasse, 21 Data Properties und 2126 Instanzdaten.

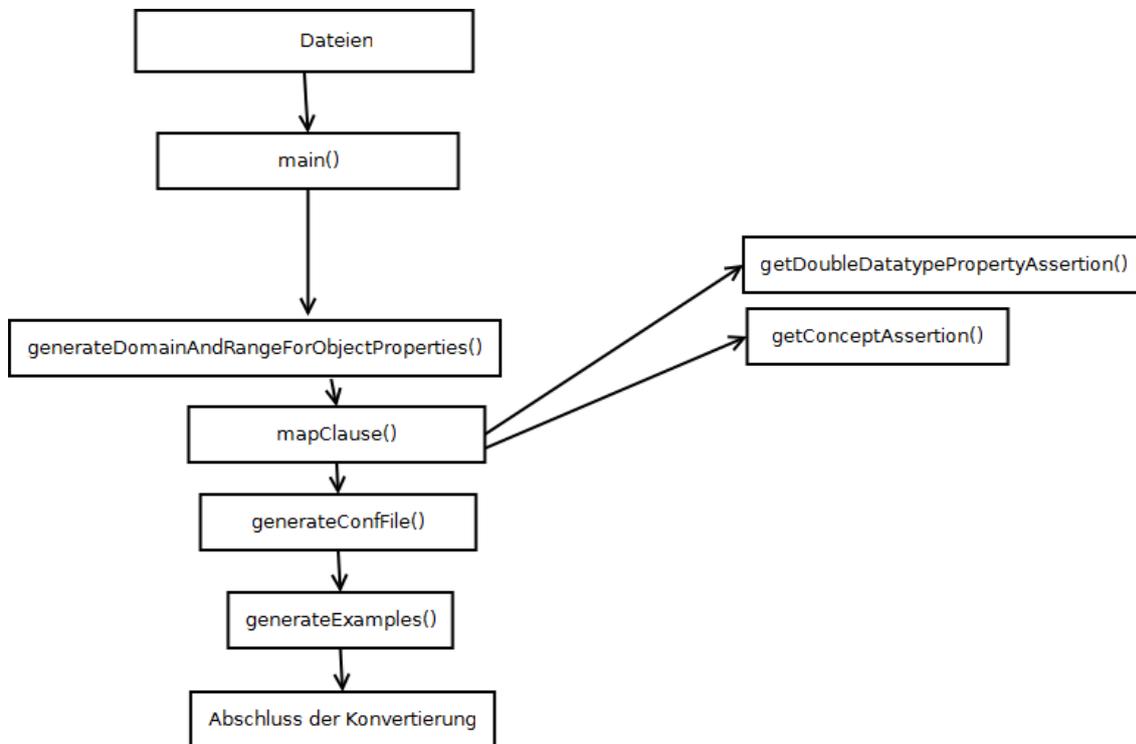


Abbildung 33: Hier ist der Ablauf zur Konvertierung der Prologdateien der Cardiotocography Wissensbasis in eine OWL Ontologie mit ihren Methoden dargestellt.

12.2 Methoden

Da es in dieser Wissensbasis verschiedene Klassen zu lernen gab wurde für jede Klasse ein eigener Lernprozess durchgeführt. Zuerst wurde mit Hilfe von OCEL für die einzelnen Klassen der Noise Parameter von 30%, um jeweils 1% gesenkt und eine 10-fold-cross-validation durchgeführt, bis keine Ergebnisse mehr erzeugt wurden. Anschließend wurde dieser Vorgang für CELOE wiederholt. Im Anschluss daran wurde mit Hilfe des Machine Learning Tools Weka, für die Algorithmen naive Bayes, naive Bayes Updateable, Random Tree, Random Forrest, PART und Decision Table ebenfalls eine 10-fold-cross-validation durchgeführt. Da es für die Algorithmen in Weka keine Möglichkeit zur Einstellung eines Noise Parameters gab, konnte der Test nur einmal durchgeführt werden. Des Weiteren

wurden die Laufzeiten der Algorithmen OCEL und CELOE für die einzelnen Lernvorgänge gespeichert um diese auch Vergleichen zu können. Ein weiterer Test wurde für die Algorithmen OCEL und CELOE durchgeführt. Dazu wurde eine feste Laufzeit von 10, 100, 300, 1000 Sekunden eingestellt und wieder eine 10-fold-cross-validation durchgeführt. Bei Weka gab es für die einzelnen Algorithmen keine Möglichkeit die Laufzeit einzustellen, so dass dieser Test nicht mit den Algorithmen von Weka durchgeführt werden konnte. Da CELOE und OCEL eine unterschiedliche Einstellung für den β Parameter besitzen und dieser für die Bestrafung längerer Konzepte zuständig ist, wurde dieser Einheitlich für beide Algorithmen auf 0,02 gesetzt. Des Weiteren wurden alle Tests auf einem 3,2GHz Quad Core mit 8Gb Ram durchgeführt, um eine Vergleichbarkeit der Laufzeiten zu gewährleisten. Im folgenden Abschnitt werden die Ergebnisse der Lernvorgänge genauer beschrieben.

12.3 Resultate und Schlussfolgerungen

In der Abbildung 34 sind die Ergebnisse des Lernproblems zum Lernen der zehn Klassen der Wissensbasis Cardiotocography dargestellt. Die einzelnen Ergebnisse sind in Tabelle 62 bis 71 im Anhang abgebildet. Dabei wurden die Ergebnisse der Algorithmen von Weka als Gerade dargestellt, da es bei Weka keine Einstellmöglichkeiten für die Genauigkeit gab, die der Algorithmus mindestens erreichen soll. Deutlich zu sehen ist, dass der Algorithmus OCEL die höchste Genauigkeit erzielt, dicht gefolgt von dem Algorithmus CELOE und random Forrest. Die viert höchste Genauigkeit wurde von dem Algorithmus PART erreicht, der von dem Algorithmus Random Tree und Decision Table gefolgt wird. Die schlechteste Genauigkeit erzielen dabei die Algorithmen Naive Bayes und Naive Bayes Updateable. Dabei ist anzumerken, dass bei den Algorithmen des DL-Learners bei geringer Laufzeit, nicht für alle Lernprobleme eine Lösung errechnet werden konnte. Bei diesen Fällen wurde nur die durchschnittliche Genauigkeit der verbliebenen Probleme die eine Lösung erzeugten berücksichtigt. Aus diesem Grund erreichen OCEL und CELOE so hohe Genauigkeiten, da teilweise nur noch von einem Lernproblem die Genauigkeit verwendet wurde. Des Weiteren ist anzumerken, dass die Gerade für die Algorithmen Naive Bayes und Naive Bayes Updateable direkt übereinander liegen, da sie die gleiche Genauigkeit erreicht haben. Aus diesem Grund kann man diese zwei Geraden nicht sehen, sondern nur eine.

In der Tabelle 34 werden die höchsten Genauigkeiten der Algorithmen OCEL, CELOE und von Weka, für der zehn Klassen Lernproblem der Wissensbasis Cardiotocography dargestellt. Es ist deutlich zu sehen, dass OCEL der Algorithmus mit der höchsten erreichten durchschnittlichen Genauigkeit ist. Der Algorithmus mit der zweit höchsten Genauigkeit ist CELOE und dann kommt der Algorithmus Random Forrest. An diese Algorithmen schließt sich der Algorithmus PART an, dicht gefolgt von Random Tree und Decision Table. Die Algorithmen mit der schlechtesten durchschnittlichen Genauigkeit sind Naive Bayes und Naive Bayes Updateable. Es ist anzumerken, dass die Ergebnisse

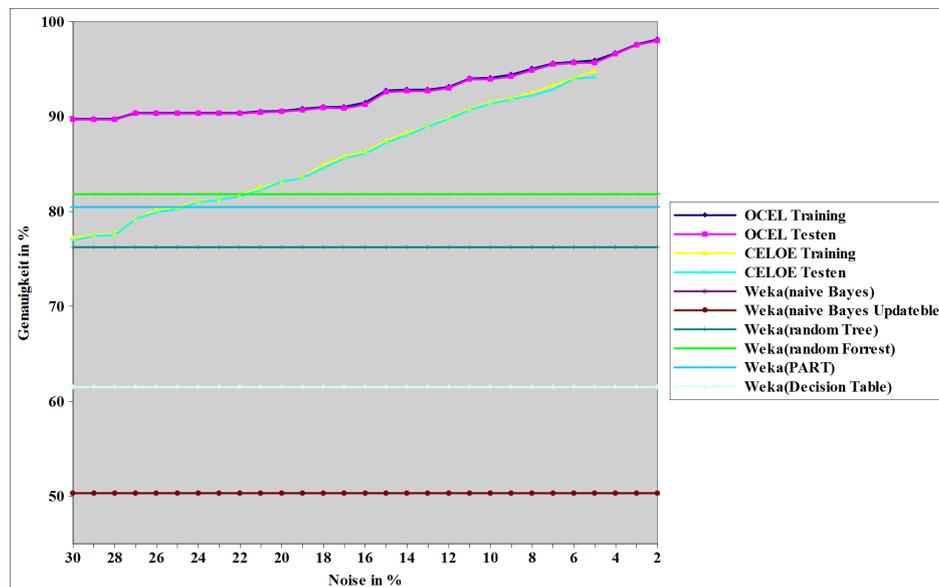


Abbildung 34: Die in der Grafik dargestellten Ergebnisse zeigen die Ergebnisse des Lernprozesses für das zehn Klassen Lernproblem der Wissensbasis Cardiotocography, anhand des eingestellten Noise Parameters. Dabei ist zu sehen, dass OCEL und CELOE die höchsten Genauigkeiten erzielen und die Algorithmen von Weka alle schlechtere Ergebnisse erreichen.

der Tabelle aus den drei Einzellernproblemen entstanden ist, in dem der Durchschnitt der Werte der Einzelprobleme ermittelt wurde.

Die in der Tabelle 35 dargestellten Ergebnisse, zeigen die p-Werte des t-Tests und eine Aussage ob diese Werte eine statistische Signifikanz aufweisen, für die Algorithmen OCEL, CELOE und alle Algorithmen von Weka. Dabei ist deutlich zu erkennen, dass sowohl OCEL als auch CELOE in allen sechs Fällen statistisch signifikant bessere Ergebnisse liefert als die Algorithmen von Weka. Auch bei dem direkten Vergleich zwischen OCEL und CELOE wurde festgestellt, dass OCEL statistisch signifikant bessere Ergebnisse liefert als CELOE.

Die in der Abbildung 35 dargestellte Graphik, zeigt die benötigte Laufzeit der Algorithmen OCEL und CELOE für den jeweils eingestellten Noise-Parameter. Dabei ist anzumerken, dass diese Grafik eine Durchschnittswert aus den zehn Klassen darstellt. Die einzelnen Ergebnisse befinden sich in den Abbildungen 88 bis 97 im Anhang. Weiterhin ist anzumerken, dass der Anstieg der Laufzeit daraus resultiert, dass bei diesen Noise-Werten der Algorithmus für ein Problem nur noch schwer eine Lösung findet. dadurch braucht er länger und die Laufzeit steigt dadurch. Der Abfall der Laufzeit lässt sich dadurch erklären, dass der Algorithmus jetzt keine Lösungen mehr für das vorher erwähnte schwere Problem findet. Dadurch beeinflussen die hohen Laufzeiten das Ergebnis nicht weiter und es kommt

Algorithmus	Testgenauigkeit
OCEL	97,977%±0,481%
CELOE	94,144%±1,542%
Naive Bayes	50,321%±4,786%
Naive Bayes Updateable	50,321%±4,786%
Random Tree	76,233%±9,074%
Random Forrest	81,815%±10,328%
PART	80,455%±7,993%
Decision Table	61,495%±6,649%

Tabelle 34: In dieser Tabelle werden die höchsten durchschnittlichen Genauigkeiten der 10-fold-cross-validation der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für die Wissensbasis Cardiotocography dargestellt, die aus den Einzelergebnissen der zehn Lernprobleme entstanden sind.

zu einem Abfall der Laufzeit. In der Grafik ist deutlich zu sehen, dass OCEL eine geringere Laufzeit besitzt als CELOE. Dies ist zum Einen darauf zurückzuführen, dass OCEL die nicht abgedeckten positiven und negativen Beispiele in den Knoten des Suchbaumes speichert, so dass weniger Reasonerfragen nötig sind. Dadurch ist der Algorithmus leicht schneller als der von CELOE. Zum Anderen benötigt der Algorithmus OCEL weniger Zeit um Konzepte zu finden die den Anforderungen an den Noise-Parameter genügen. Dies liegt daran, dass OCEL die gefundenen Konzepte im Suchbaum anders bewertet und diese eine höhere interne Bewertung von dem Algorithmus bekommen. Aus diesen Gründen benötigt OCEL weniger Zeit als CELOE. Die höchste gemessene Laufzeit bei OCEL beträgt $58,272s \pm 34,774s$ und bei CELOE beträgt diese $764,683s \pm 312,448s$. Um herauszufinden ob diese Unterschiede statistisch signifikant sind wurde ein t-Test durchgeführt. Der resultierende p-Wert ist 0,0001 und das bedeutet, dass die Unterschiede der Laufzeit nicht nur sehr groß ist, sondern das diese Unterschiede auch statistisch signifikant sind.

In der Abbildung 36 werden die durchschnittlichen Genauigkeiten, die für die drei Klassen Normal, Suspect und Patologic, bei fest eingestellter Laufzeit von 10, 100, 300 und 1000 Sekunden Laufzeit, erreicht wurden. Dabei fällt auf, dass bei OCEL für 1000 Sekunden Laufzeit keine Genauigkeiten angegeben wurden. Dies liegt daran, dass OCEL für den Aufbau des Suchbaumes zu viel Arbeitsspeicher verwendet hat, da er zusätzlich zu den Konzepten auch die nicht abgedeckten positiven und negativen speichert, so dass vor Beendigung der Algorithmus eine *OutOfMemoryException* geworfen hat. Des Weiteren fällt auf, dass sich die Genauigkeit der beiden Algorithmen mit Erhöhung der Laufzeit immer weiter erhöht und das CELOE eine geringere Genauigkeit aufweist als OCEL. Dabei lässt sich die geringere Genauigkeit bei CELOE dadurch erklären, dass es länger gedauert hat, bis CELOE Konzepte gefunden hat, die eine höhere interne Bewertung bekommen haben, so dass es insgesamt

Algorithmen	p-Wert	statistische Signifikanz
OCEL vs. Naive Bayes	0,0001	ja
OCEL vs. Naive Bayes Updateable	0,0001	ja
OCEL vs. Random Tree	0,0001	ja
OCEL vs. Random Forrest	0,0001	ja
OCEL vs. PART	0,0001	ja
OCEL vs. Decision Table	0,0001	ja
CELOE vs. Naive Bayes	0,0001	ja
CELOE vs. Naive Bayes Updateable	0,0001	ja
CELOE vs. Random Tree	0,0001	ja
CELOE vs. Random Forrest	0,0015	ja
CELOE vs. PART	0,0001	ja
CELOE vs. Decision Table	0,0001	ja
CELOE vs. OCEL	0,0001	ja

Tabelle 35: In dieser Tabelle wird die statistische Signifikanz der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für das drei Klassen Lernproblem der Wissensbasis Cardiotocography dargestellt. Dabei ist deutlich zu sehen, dass OCEL und CELOE in allen sechs Fällen statistisch signifikant besser arbeitet als die Algorithmen von Weka. auch bei einem direkten Vergleich zwischen OCEL und CELOE wurde festgestellt, dass OCEL statistisch signifikant besser arbeitet.

länger gedauert hat bessere Konzepte zu finden. Dies führte dazu das CELOE nach Ablauf der Zeit eine geringer Genauigkeit aufweist als OCEL. Die Ergebnisse sind Durchschnittswerte aus den drei Klassen und die einzelnen Ergebnisse bei fest eingestellter Laufzeit sind in den Tabellen 98 bis 100 im Anhang zu sehen.

In der Tabelle cardioLaufzeit1Sig sind die Genauigkeiten, sowie p-wert und eine Aussage, ob statistische Signifikanz vorlag, für die Algorithmen OCEL und CELOE bei fest eingestellter Laufzeit von 10, 100 und 300 Sekunden dargestellt. Dabei ist zu erkennen, dass es zwischen den Algorithmen OCEL und CELOE größere Unterschiede in den Genauigkeiten gibt. Des Weiteren ist zu erkennen, dass die Unterschiede nicht nur groß sind, sondern auch statistisch Signifikant bei allen drei eingestellten Laufzeiten von 10, 100 und 300 Sekunden. Es ist anzumerken, dass diese Tabelle aus den Durchschnittswerten der Ergebnisse erstellt wurden, die sich im Anhang der Arbeit befinden.

In der Abbildung 37 sind die Ergebnisse des Lernproblems zum Lernen der drei Klassen Normal, Suspect und Pathologic dargestellt. Die einzelnen Ergebnisse sind in Tabelle 72 bis 74 im Anhang abgebildet. Dabei wurden die Ergebnisse der Algorithmen von Weka als Gerade dargestellt, da es bei Weka keine Einstellmöglichkeiten für die Genauigkeit gab, die der Algorithmus mindestens erreichen

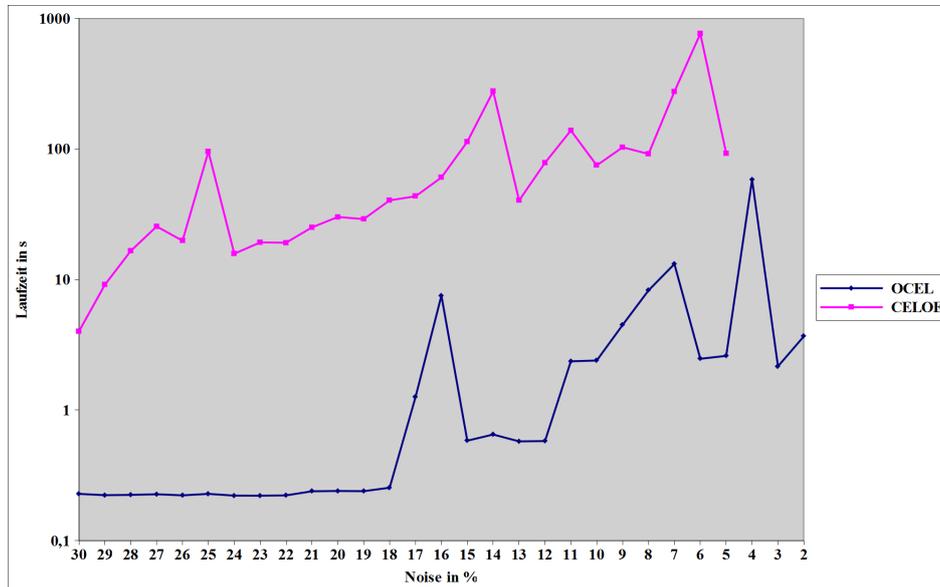


Abbildung 35: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für das zehn Klassen Lernproblem der Wissensbasis Cardiotoecography

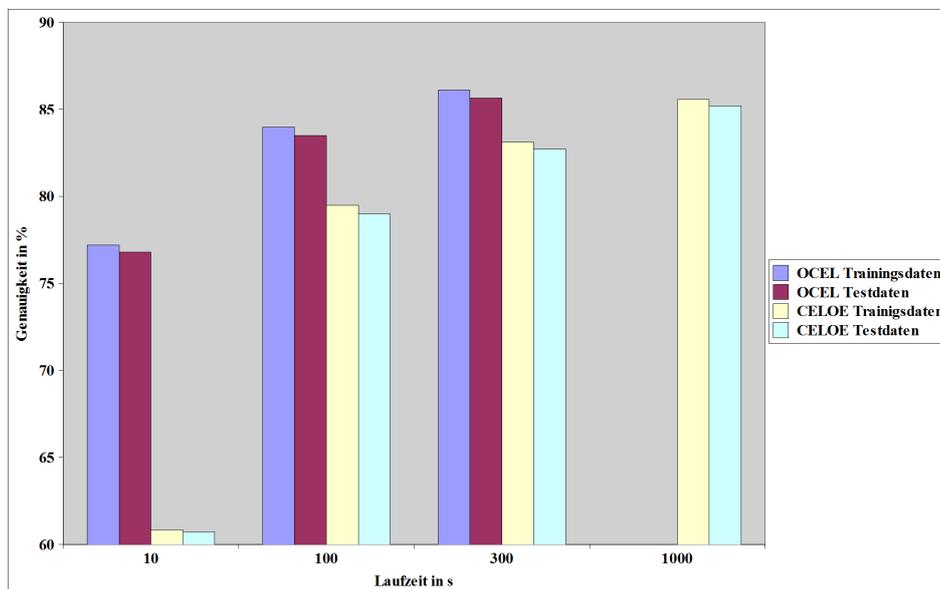


Abbildung 36: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für das zehn Klassen Lernproblem der Wissensbasis Cardiotoecography.

Laufzeit	Testgenauigkeit OCEL	Testgenauigkeit CELOE	p-Wert	statistische Signifi- kanz
10 Sekunden	76,783%±4,076%	60,741%±5,311%	0,0001	ja
100 Sekunden	83,496%±3,359%	78,995%±5,396%	0,0380	ja
300 Sekunden	85,635s%±2,675%	82,704%±3,441%	0,0475	ja

Tabelle 36: Bei dieser Tabelle handelt es sich um die Darstellung der Testgenauigkeit der Algorithmen OCEL und CELOE bei 10, 100 und 300 Sekunden Laufzeit, für das zehn Klassen Lernproblem der Wissensbasis Cardiotocography und die Ergebnisse des t-Tests, ob eine statistische Signifikanz in den Unterschieden der Genauigkeiten vorliegt.

soll. Deutlich zu sehen ist, dass der Algorithmus OCEL die höchste Genauigkeit erzielt, dicht gefolgt von dem Algorithmus CELOE und random Forrest. Die viert höchste Genauigkeit wurde von dem Algorithmus PART erreicht, der von dem Algorithmus Random Tree und Decision Table gefolgt wird. Die schlechteste Genauigkeit erzielen dabei die Algorithmen Naive Bayes und Naive Bayes Updateable. Dabei ist anzumerken, dass bei den Algorithmen des DL-Learners bei geringer Laufzeit, nicht für alle Lernprobleme eine Lösung errechnet werden konnte. Bei diesen Fällen wurde nur die durchschnittliche Genauigkeit der verbliebenen Probleme die eine Lösung erzeugten berücksichtigt. Aus diesem Grund erreichen OCEL und CELOE so hohe Genauigkeiten, da teilweise nur noch von einem Lernproblem die Genauigkeit verwendet wurde. Des Weiteren ist anzumerken, dass die Gerade für die Algorithmen Naive Bayes und Naive Bayes Updateable direkt übereinander liegen, da sie die gleiche Genauigkeit erreicht haben. Aus diesem Grund kann man diese zwei Geraden nicht sehen, sondern nur eine.

In der Tabelle 37 werden die höchsten Genauigkeiten der Algorithmen OCEL, CELOE und von Weka, für der zehn Klassen Lernproblem der Wissensbasis Cardiotocography dargestellt. Es ist deutlich zu sehen, dass OCEL der Algorithmus mit der höchsten erreichten durchschnittlichen Genauigkeit ist. Der Algorithmus mit der zweit höchsten Genauigkeit ist CELOE und dann kommt der Algorithmus Random Forrest. An diese Algorithmen schließt sich der Algorithmus PART an, dicht gefolgt von Random Tree und Decision Table. Die Algorithmen mit der schlechtesten durchschnittlichen Genauigkeit sind Naive Bayes und Naive Bayes Updateable. Es ist anzumerken, dass die Ergebnisse der Tabelle aus den drei Einzellernproblemen entstanden ist, in dem der Durchschnitt der Werte der Einzelprobleme ermittelt wurde.

Die in der Tabelle 38 dargestellten Ergebnisse, zeigen die p-Werte des t-Tests und eine Aussage ob diese Werte eine statistische Signifikanz aufweisen, für die Algorithmen OCEL, CELOE und alle Algorithmen von Weka. Dabei ist deutlich zu erkennen, dass sowohl OCEL als auch CELOE in allen sechs Fällen statistisch signifikant bessere Ergebnisse liefert als die Algorithmen von Weka. Nur

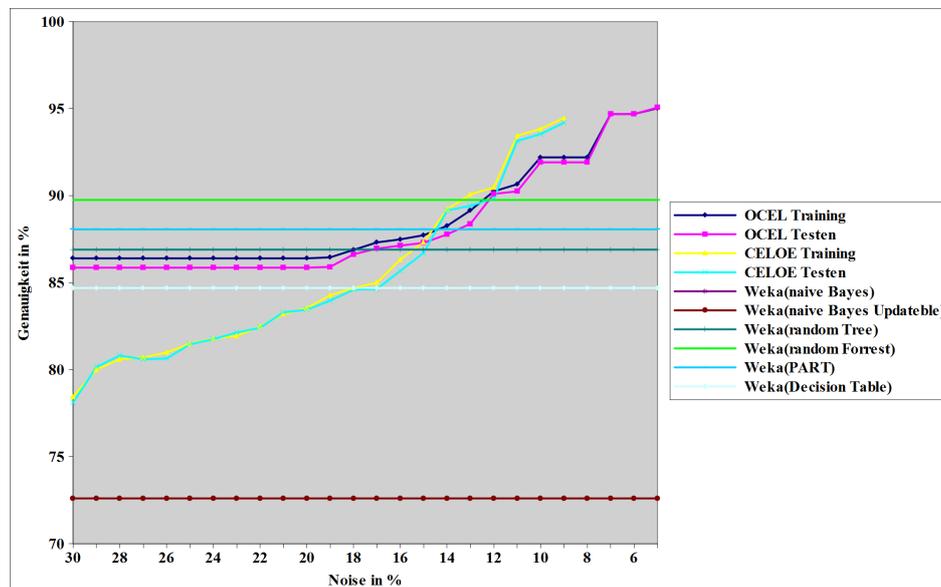


Abbildung 37: Die in der Grafik dargestellten Ergebnisse zeigen die Ergebnisse des Lernprozesses für die drei Klassen Normal, Suspect und Pathologic der Wissensbasis Cardiotocography. anhand des eingestellten Noise Parameters. Dabei ist zu sehen, dass OCEL und CELOE die höchsten Genauigkeiten erzielen und die Algorithmen von Weka alle schlechtere Ergebnisse erreichen.

bei einem direkten Vergleich der beiden Algorithmen OCEL und CELOE konnte keine statistische Signifikanz nachgewiesen werden.

Die in Abbildung 38 dargestellten Kurven, zeigen den Verlauf der Laufzeit bei Verringerung des Noise-Parameters als Zusammenfassungsverlauf für die Klassen Normal, Suspect und Patologic. Dabei ist zu sehen, dass CELOE eine geringere Laufzeit besitzt als OCEL. Dies liegt daran, dass zum einen OCEL die nicht abgedeckten positiven und negativen Beispiele im Suchbaum speichert, so dass weniger Reasoneranfragen nötig sind. Dies führt zur Verkürzung der Laufzeit. Zum Anderen benötigt OCEL viel weniger Zeit um Konzepte zu finden, die für den eingestellten Noise-Parameter eine ausreichend hohe Bewertung bekommen. Dies liegt daran, dass OCEL die gefundenen Konzepte im Suchbaum höher bewertet als CELOE und der Algorithmus so schneller zu einer Lösung kommt. Die höchste Laufzeit bei OCEL beträgt $19,332s \pm 4,958s$ und die höchste Laufzeit bei CELOE beträgt $981,55s \pm 186,077s$. Um herauszufinden, ob die Laufzeitunterschiede für die höchsten Laufzeiten statistisch signifikant sind wurde ein t-Test durchgeführt. Der p-Wert des t-Tests beträgt weniger als 0,0001. Damit steht fest, dass diese Unterschiede nicht nur sehr groß sind, sondern dass sie auch statistisch signifikant sind.

In der Abbildung 39 werden die durchschnittlichen Genauigkeiten, die für die drei Klassen Normal,

Algorithmus	Testgenauigkeit
OCEL	95,06%±1,34%
CELOE	94,183%±1,026%
Naive Bayes	72,603%±4,686%
Naive Bayes Updateable	72,603%±4,686%
Random Tree	86,896%±4,273%
Random Forrest	89,75%±4,586%
PART	88,06%±4,016%
Decision Table	84,693%±5,006%

Tabelle 37: In dieser Tabelle werden die höchsten durchschnittlichen Genauigkeiten der 10-fold-cross-validation der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für die Wissensbasis Cardiotocography dargestellt, die aus den Einzelergebnissen der drei Lernprobleme entstanden sind.

Suspect und Patologic, bei fest eingestellter Laufzeit von 10, 100, 300 und 1000 Sekunden Laufzeit, erreicht wurden. Dabei fällt auf, dass bei OCEL für 1000 Sekunden Laufzeit keine Genauigkeiten angegeben wurden. Dies liegt daran, dass OCEL für den Aufbau des Suchbaumes zu viel Arbeitsspeicher verwendet hat, da er zusätzlich zu den Konzepten auch die nicht abgedeckten positiven und negativen speichert, so dass vor Beendigung der Algorithmus eine *OutOfMemoryException* geworfen hat. Des Weiteren fällt auf, dass sich die Genauigkeit der beiden Algorithmen mit Erhöhung der Laufzeit immer weiter erhöht und das CELOE eine geringere Genauigkeit aufweist als OCEL. Dabei lässt sich die geringer Genauigkeit bei CELOE dadurch erklären, dass es länger gedauert hat, bis CELOE Konzepte gefunden hat, die eine höhere interne Bewertung bekommen haben, so dass es insgesamt länger gedauert hat bessere Konzepte zu finden. Dies führte dazu das CELOE nach Ablauf der Zeit eine geringer Genauigkeit aufweist als OCEL. Die Ergebnisse sind Durchschnittswerte aus den drei Klassen und die einzelnen Ergebnisse bei fest eingestellter Laufzeit sind in den Tabellen 98 bis 100 im Anhang zu sehen.

In der Tabelle *cardioLaufzeit2Sig* sind die Genauigkeiten, sowie p-wert und eine Aussage, ob statistische Signifikanz vorlag, für die Algorithmen OCEL und CELOE bei fest eingestellter Laufzeit von 10, 100 und 300 Sekunden dargestellt. Dabei ist zu erkennen, dass es zwischen den Algorithmen OCEL und CELOE größere Unterschiede in den Genauigkeiten gibt. Des Weiteren ist zu erkennen, dass die Unterschiede nicht nur groß sind, sondern auch statistisch Signifikant bei allen drei eingestellten Laufzeiten von 10, 100 und 300 Sekunden. Es ist anzumerken, dass diese Tabelle aus den Durchschnittswerten der Ergebnisse erstellt wurden, die sich im Anhang der Arbeit befinden.

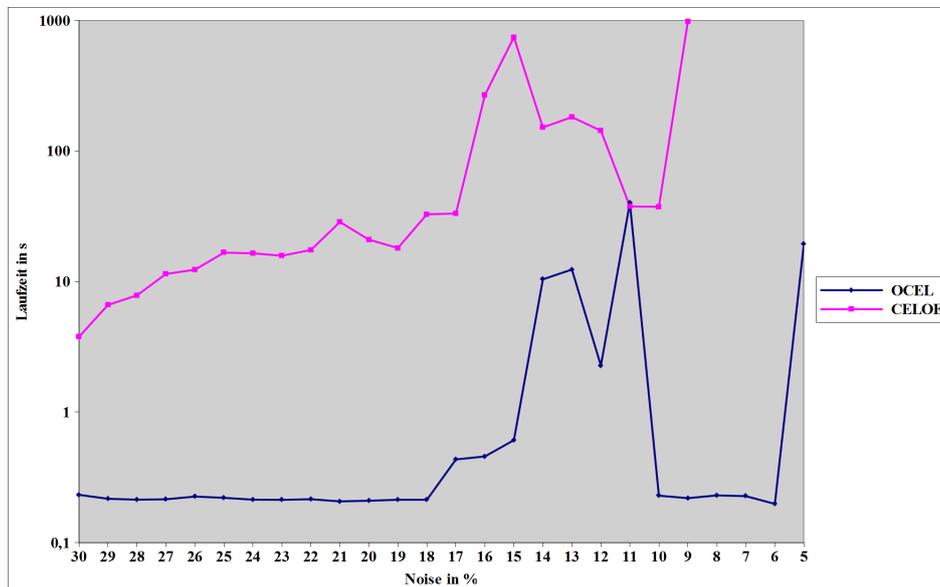


Abbildung 38: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für das drei Klassen Lernproblem der Wissensbasis Cardiotoctography

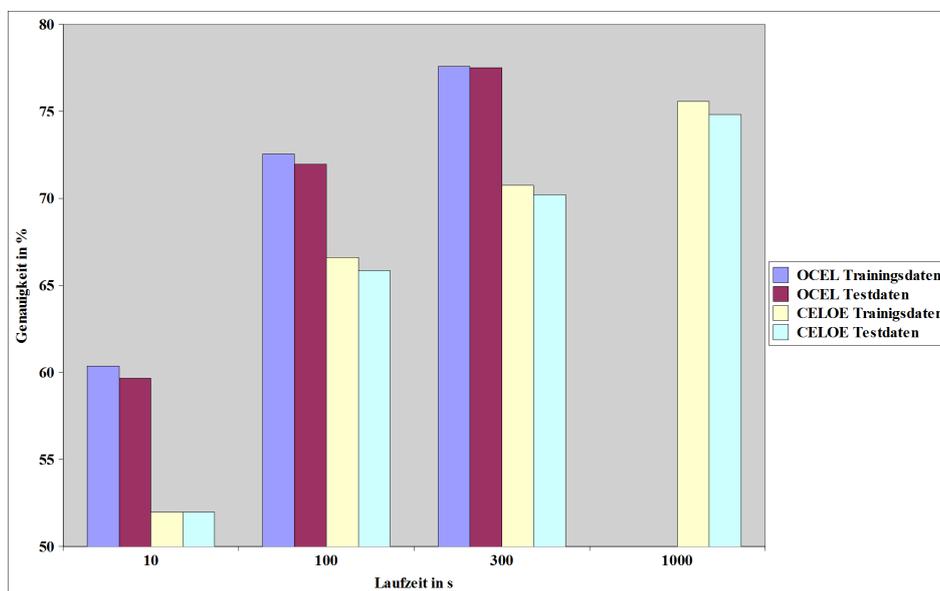


Abbildung 39: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für das drei Klassen Lernproblem der Wissensbasis Cardiotoctography.

Algorithmen	p-Wert	statistische Signifikanz
OCEL vs. Naive Bayes	0,0001	ja
OCEL vs. Naive Bayes Updateable	0,0001	ja
OCEL vs. Random Tree	0,0001	ja
OCEL vs. Random Forrest	0,0025	ja
OCEL vs. PART	0,0001	ja
OCEL vs. Decision Table	0,0001	ja
CELOE vs. Naive Bayes	0,0001	ja
CELOE vs. Naive Bayes Updateable	0,0001	ja
CELOE vs. Random Tree	0,0001	ja
CELOE vs. Random Forrest	0,0080	ja
CELOE vs. PART	0,0002	ja
CELOE vs. Decision Table	0,0001	ja
CELOE vs. OCEL	0,1177	nein

Tabelle 38: In dieser Tabelle wird die statistische Signifikanz der Algorithmen OCEL, CELOE und den Algorithmen von Weka, für das drei Klassen Lernproblem der Wissensbasis Cardiotocography dargestellt. Dabei ist deutlich zu sehen, dass OCEL und CELOE in allen sechs Fällen statistisch signifikant besser arbeitet als die Algorithmen von Weka. Nur in einem direkten Vergleich zwischen OCEL und CELOE konnte keine statistische Signifikanz nachgewiesen werden.

Resultate des 10 Klassen Lernproblems

In diesem Abschnitt werden die gelernten Konzepte der beiden Algorithmen OCEL und CELOE miteinander verglichen. Dabei werden die Konzepte erst vorgestellt und dann begründet, warum diese unterschiedlich sind.

Klasse 1

In der Tabelle 40 sind die von den Algorithmen OCEL und CELOE vorgeschlagenen Konzepte mit entsprechenden Genauigkeiten, Laufzeiten und Konzeptlängen, für die erste Klasse der Wissensbasis Cardiotocography dargestellt. Dabei bedeute das von OCEL gelernte Konzept, dass ein Patient, mit einem am häufigsten erreichten Wert der kleiner gleich 150,5 und mehr als 23,5 anhaltende Verringerungen der Herzfrequenz und mehr wie 3,5 Kontraktionen der Gebärmutter pro Sekunde oder mehr als 15,5 Kontraktionen der Gebärmutter pro Sekunde, in die Klasse ein eingeteilt wird. Das von CELOE gelernte Konzept hingegen bedeutet, dass ein Patient, der weniger als 2,5 Erhöhungen der Herzfrequenz pro Sekunde und mehr wie 0,5 anhaltende Verringerungen der Herzfrequenz pro

Laufzeit	Testgenauigkeit OCEL	Testgenauigkeit CELOE	p-Wert	statistische Signifi- kanz
10 Sekunden	59,675%±3,175%	51,958%±2,121%	0,0001	ja
100 Sekunden	71,969%±4,281%	65,862%±3,87%	0,0036	ja
300 Sekunden	77,494%±3,747%	70,209%±3,382%	0,0002	ja

Tabelle 39: Bei dieser Tabelle handelt es sich um die Darstellung der Testgenauigkeit der Algorithmen OCEL und CELOE bei 10, 100 und 300 Sekunden Laufzeit, für das drei Klassen Lernproblem der Wissensbasis Cardiotocography und die Ergebnisse des t-Tests, ob eine statistische Signifikanz in den Unterschieden der Genauigkeiten vorliegt.

Sekunde haben, in die Klasse eins eingeteilt werden kann. Dabei ist zu erkennen, dass OCEL und CELOE unterschiedliche Konzepte zur Beschreibung der Klasse lernen. Dies lässt sich darauf zurückführen, dass für das Lernen der Klasse unterschiedliche Trainingsdaten zur Verfügung standen. Diese haben unterschiedliche Attributausprägungen, so dass die Konzepte im Suchbaum anders bewertet wurden. Somit wurden andere Pfade im Suchbaum weiterverfolgt, so dass es dadurch zu den unterschiedlichen Konzepten gekommen ist.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasHistogramMode <= 150.5 AND hasProlonguedDecelerationsPerSecond >= 23.5 AND (hasUterineContractionsPerSecond >= 3.5 OR hasUterineContractionsPerSecond >= 15.5))</i>	85,446%	11,726s	17
CELOE	<i>(hasAccelerationsPerSecond <= 2.5 AND hasProlonguedDecelerationsPerSecond >= 0.5)</i>	79,717%	1,112s	7

Tabelle 40: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der ersten Klasse für die Wissensbasis Cardiotocography dargestellt. Dabei ist deutlich zu erkennen, dass OCEL und CELEO unterschiedliche Beschreibungen für die Klasse eins errechnet haben.

Klasse 2

Die in der Tabelle 41 dargestellten Konzepte, sind die von den Algorithmen OCEL und CELOE vorgeschlagenen Konzepte, die als Beschreibung der Klasse zwei dienen. Dabei bedeutet das von OCEL

gelernte Konzept, dass ein Patient der mehr wie 2,5 Erhöhungen der Herzfrequenz pro Sekunde und einen Mittelwert der Kurzzeitveränderung der Herzschläge pro Sekunde von weniger als 1,5 besitzt, in die Klasse zwei eingeteilt werden kann. CELOE hingegen beschreibt die Klasse zwei wie folgt. Ein Patient der mehr wie 0,5 Erhöhungen der Herzfrequenz und einen Mittelwert der Kurzzeitveränderung der Herzschläge pro Sekunde von weniger als 1,5 und mehr wie 2,5 Erhöhungen der Herzfrequenz oder eine Breite der Herzfrequenz, die größer gleich 46,5 ist, wird zur Klasse zwei zugeordnet. Dabei ist zu erkennen, dass OCEL und CELOE unterschiedliche Konzepte vorschlagen und dass das von OCEL vorgeschlagene Konzept in dem Konzept von CELOE enthalten ist, nur mit unterschiedlichen Attributwerten bei dem Attribut *hasAccelerationsPerSecond*. Dieser Unterschied lässt sich damit erklären, dass beim Lernen der Beschreibungen unterschiedliche Trainingsdaten verwendet wurden, die unterschiedliche Attributausprägungen besitzen. Dadurch kam es zu einer unterschiedlichen internen Bewertung der Konzepte, so dass beide Algorithmen unterschiedliche Pfade im Suchbaum weiterverfolgt haben. Dies führte dazu dass unterschiedliche Konzepte gelernt wurden.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasAccelerationsPerSecond >= 2.5 AND hasMeanValueOfShortTermVariability <= 1.5)</i>	93,868%	0,504s	9
CELOE	<i>(hasAccelerationsPerSecond >= 0.5 AND hasMeanValueOfShortTermVariability <= 1.5 AND (hasAccelerationsPerSecond >= 2.5 OR hasWidthOfFHRHistogram >= 46.5))</i>	91,549%	247,008s	15

Tabelle 41: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der zweiten Klasse für die Wissensbasis Cardiotocography dargestellt. Dabei ist zu erkennen, dass OCEL und CELOE unterschiedliche Konzepte vorschlagen und dass das Konzept von OCEL als Teilkonzept bei CELOE wieder auftaucht nur mit einer anderen Attributausprägung bei *hasAccelerationsPerSecond*

Klasse 3

In der Tabelle 42 werden die Klassenbeschreibungen mit entsprechenden Genauigkeiten, Laufzeiten und Konzeptlängen, für die Algorithmen OCEL und CELOE angegeben. Dabei bedeutet das von OCEL gelernte Konzept, dass ein Patient, der mehr als 0,5% der Zeit eine abnormale Langzeitveränderung der Herzfrequenz besitzt, in die Klasse 3 eingeteilt werden kann. Das Konzept, welches CELOE gelernt hat, bedeutet, dass ein Patient, der eine maximale Herzfrequenz die kleiner gleich 169,5

und mehr als 0,75 schwerwiegenden Verringerungen der Herzfrequenz und der in 11,35% der Zeit abnormale Kurzzeitveränderungen in der Herzfrequenz, oder eine kleiner breite des Kardiogramms von 4 hat, wird in die Klasse drei eingeordnet. Bei diesen gelernten Konzepten ist deutlich zu sehen, dass diese sich sehr stark voneinander unterscheiden. Dies lässt sich dadurch erklären, dass unterschiedliche Trainingsdaten zum Lernen mit unterschiedlichen Attributausprägungen zur Verfügung standen. Aus diesem Grund werden unterschiedliche Pfade im Suchbaum durchschritten, so dass unterschiedliche Konzepte unterschiedlich bewertet werden. Deshalb werden bei Klasse drei unterschiedliche Konzepte gelernt.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasPercentageOfTime-WithAbnormalLongTermVariability >= 0.5)</i>	97,642%	0,215s	5
CELOE	<i>(hasMaximumOfFHRHistogram <= 169.5 AND hasSevereDecelerations-PerSecond >= 0.75 AND (hasPercentageOfTimeWithAbnormalShortTerm-Variability >= 11.350000000000001 OR hasWidthOfFHRHistogram <= 4.0))</i>	92,925%	59,146s	15

Tabelle 42: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der dritten Klasse für die Wissensbasis Cardiotocography dargestellt. Dabei ist deutlich zu erkennen, dass sich die gelernten Konzepte, welche von den Algorithmen OCEL und CELOE vorgeschlagen wurden, stark unterscheiden.

Klasse 4

In der Tabelle 43 werden die von den Algorithmen OCEL und CELOE vorgeschlagenen Konzepte für die Beschreibung der Klasse vier, mit entsprechenden Genauigkeiten, Laufzeiten und Konzeptlängen angegeben. Das von OCEL gelernte Konzept bedeutet dabei, dass ein Patient, der mehr als 17,5 Bewegungen des Fötus und weniger als 21,6% der Zeit abnormale Kurzzeitveränderungen der Herzfrequenz und mehr als 6,5 Erhöhungen der Herzfrequenz, oder eine Breite des Kardiogramms von weniger als 4 besitzt, wird in die Klasse vier eingeteilt. Das von CELOE gelernte Konzept hingegen bedeutet, dass ein Patient mit mehr als 4,5 Erhöhungen der Herzfrequenz und eine durchschnittliche Kurzzeitveränderung der Herzfrequenz, die kleiner gleich 1,5 ist, oder mehr als 1,25 schwerwiegende Verringerungen in der Herzfrequenz besitzt in die Klasse vier eingeteilt wird. Auch hier ist zu

erkennen, dass OCEL und CELOE wieder stark unterschiedliche Konzepte lernen. Dies lässt sich auch darauf zurückführen, dass beim Lernen unterschiedliche Trainingsdaten mit unterschiedlichen Attributausprägungen zur Verfügung standen. Dabei kam es dann dazu, dass die Algorithmen unterschiedliche Konzepte im Suchbaum aufgebaut haben, die eine unterschiedliche interne Bewertung besitzen. Aus diesem Grund werden unterschiedliche Pfade im Suchbaum durchlaufen, so dass am ende verschiedene Konzepte vorgeschlagen wurden.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasFetalMovementsPerSecond >= 17.5 AND hasPercentageOfTimeWithAbnormalShortTermVariability <= 21.6 AND (hasAccelerationsPerSecond >= 6.5 OR hasWidthOfFHRHistogram <= 4.0))</i>	98,122%	13,522s	17
CELOE	<i>(hasAccelerationsPerSecond >= 4.5 AND hasMeanValueOfShortTermVariability <= 1.5 AND (hasMaximumOfFHRHistogram >= 177.5 OR hasSevereDecelerationsPerSecond >= 1.25))</i>	95,283%	62,753s	15

Tabelle 43: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der vierten Klasse für die Wissensbasis Cardiotocography dargestellt. Dabei ist deutlich zu sehen, dass OCEL und CELOE stark unterschiedliche Konzepte vorgeschlagen haben.

Klasse 5

Die in der Tabelle 44 dargestellten Konzepte, mit entsprechender Genauigkeit, Laufzeit und Konzeptlänge, sind die von den Algorithmen OCEL und CELOE vorgeschlagenen Klassenbeschreibungen für die fünfte Klasse der Wissensbasis Cardiotocography. Dabei bedeutet das von OCEL gelernte Konzept, dass ein Patient, der mehr als 0,5% der Zeit abnormale Langzeitveränderungen in der Herzfrequenz aufweist, zu der Klasse fünf zugeordnet werden kann. Das von CELOE vorgeschlagene Konzept hingegen bedeutet, dass ein Patient, welcher anhaltende Verringerungen der Herzfrequenz zwischen 0,5 und 55,5 und einen i Kardiogramm am häufigsten auftretenden Wert der größer gleich 142,5 und eine durchschnittliche Kurzzeitveränderung der Herzfrequenz die kleiner gleich 1,5 oder eine Varianz im Kardiogramm größer gleich 0,5 und weniger als 0,25 schwerwiegende Verringerungen

der Herzfrequenz besitzt in die Klasse fünf eingeteilt werden kann. Dabei ist hier zu sehen, dass OCEL und CELOE stark voneinander unterschiedliche Konzepte lernen und OCEL das kürzere von beiden vorschlägt. Diese starken Unterschiede lassen sich auch hier auf die unterschiedlichen Trainingsdaten zurückführen, welche unterschiedliche Attributausprägungen besitzen. Diese führen wiederum dazu das die Algorithmen unterschiedliche Pfade im Suchbaum durchschreiten, da die interne Bewertung der Konzepte unterschiedlich ist. Aus diesem Grund kommt es zu sich stark unterschiedlichen Konzepten, die von den Algorithmen OCEL und CELOE vorgeschlagen wurden

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasPercentageOfTime-WithAbnormalLongTermVariability >= 0.5)</i>	96,698%	0,205s	5
CELOE	<i>(hasProlonguedDecelerationsPerSecond >= 0.5 AND hasProlonguedDecelerationsPerSecond <= 55.5 AND ((hasHistogramMode >= 142.5 AND hasMeanValueOfShortTermVariability <= 1.5) OR (hasHistogramVariance >= 0.5 AND hasSevereDecelerationsPerSecond <= 0.25)))</i>	89,72%	784,233s	23

Tabelle 44: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der fünften Klasse für die Wissensbasis Cardiotocography dargestellt. Es ist deutlich zu sehen, dass sich die gelernten Konzepte von OCEL und CELOE sehr stark voneinander unterscheiden und das OCEL das kürzere von beiden lernt.

Klasse 6

In der Tabelle 45 werden die von den Algorithmen OCEL und CELOE vorgeschlagenen Konzepte mit entsprechender Genauigkeit, Laufzeit und Konzeptlänge, für die Klasse sechs der Wissensbasis Cardiotocography dargestellt. Das Konzept, welches OCEL vorschlägt bedeutet dabei, dass ein Patient mit einer durchschnittlichen Kurzzeitveränderung der Herzfrequenz die größer gleich 1,5 ist und eine Erhöhung der Herzfrequenz die größer gleich 2,5 oder 18,5, in die Klasse sechs eingeteilt wird. Das von CELOE gelernte Konzept hingegen bedeutet, dass ein Patient der mehr als 0,5 Erhöhungen der Herzfrequenz und eine durchschnittliche Kurzzeitveränderung der Herzfrequenz, die größer gleich 1,5 ist, in die Klasse sechs eingeteilt wird. Hier ist deutlich zu erkennen, das sich die Konzepte nur durch die Attributausprägung von *hasAccelerationsPerSecond* unterscheiden. Dies lässt sich mit

den verschiedenen Trainingsdaten, die zum Lernen zur Verfügung standen erklären. Dabei werden die Trainingsdaten unterschiedliche Attributausprägungen für *hasAccelerationsPerSecond* gehabt haben, so dass die Unterschiede in den vorgeschlagenen Konzepten von diesen verschiedenen Werten kommt.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasMeanValueOfShortTermVariability >= 1.5 AND (hasAccelerationsPerSecond >= 2.5 OR hasAccelerationsPerSecond >= 18.5))</i>	97,17%	3,678s	13
CELOE	<i>(hasAccelerationsPerSecond >= 0.5 AND hasMeanValueOfShortTermVariability >= 1.5)</i>	97,17%	18,419s	7

Tabelle 45: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der sechsten Klasse für die Wissensbasis Cardiotocography dargestellt. Hier ist zu erkennen, dass sich die Konzepte nur durch die Attributausprägung von *hasAccelerationsPerSecond* unterscheiden.

Klasse 7

Die in der Tabelle 46 dargestellten Klassenbeschreibungen, samt Genauigkeit, Laufzeit und Konzeptlänge, sind die Konzepte, welche von den Algorithmen OCEL und CELOE für die Klasse 7 der Wissensbasis Cardiotocography gelernt wurden. Dabei bedeutet das von OCEL gelernte Konzept, dass ein Patient, der weniger als 2,5 Erhöhungen der Herzfrequenz und eine durchschnittliche Kurzzeitveränderung der Herzfrequenz von 1,5 und mehr und eine durchschnittliche Kurzzeitveränderung der Herzfrequenz, die kleiner gleich 1,5, oder weniger wie 1,75 schwerwiegende Verringerungen der Herzfrequenz und weniger wie 0,5 Erhöhungen der Herzfrequenz und einen durchschnittlichen Kardiogrammwert der größer gleich 103,5 oder weniger wie 25,5 geringfügige Verringerungen der Herzfrequenz und eine breite des Kardiogramms, welches größer gleich 102,5 ist, besitzt, wird in die Klasse 7 eingeteilt. Das von CELOE gelernte Konzept hingegen bedeutet, dass ein Patient, der weniger als 2,5 Erhöhungen der Herzfrequenz und eine durchschnittliche Kurzzeitveränderung der Herzfrequenz von 1,5 und mehr und mehr wie 0,5 Erhöhungen der Herzfrequenz oder eine Kardiogrammtendenz die größer gleich -0,5 ist, besitzt, wird in die Klasse sieben eingeteilt. Auch bei dieser Klasse wurden sehr starke Unterschied in den vorgeschlagenen Klassenbeschreibungen festgestellt. Diese lassen sich dadurch erklären, dass unterschiedliche Trainingsdaten, mit unterschiedlichen Attributausprägungen zum Lernen zur Verfügung standen. Daraus folgt, dass der Suchbaum anders aufgebaut wird und die

Konzepte eine unterschiedliche interne Bewertung bekommen. Das bedeutet, dass die Algorithmen OCEL und CELOE unterschiedliche Pfade im Suchbaum durchlaufen, so dass am Ende unterschiedliche Konzepte gelernt werden.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasAccelerationsPerSecond <= 2.5 AND hasMeanValueOfShortTermVariability >= 1.5 AND (hasMeanValueOfLongTermVariability <= 1.5 OR hasSevereDecelerationsPerSecond <= 1.75) AND ((hasAccelerationsPerSecond <= 0.5 AND hasHistogramMean >= 103.5) OR (hasLightDecelerationsPerSecond <= 25.5 AND hasWidthOfFHRHistogram >= 102.5)))</i>	98,113%	927,629s	33
CELOE	<i>(hasAccelerationsPerSecond <= 2.5 AND hasMeanValueOfShortTermVariability >= 1.5 AND (hasAccelerationsPerSecond >= 0.5 OR hasHistogramTendency >= -0.5))</i>	96,226%	22,424s	15

Tabelle 46: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der siebten Klasse für die Wissensbasis Cardiotocography dargestellt. Auch hier ist deutlich zu erkennen, dass sich die von OCEL und CELOE gelernten Konzepte deutlich voneinander unterscheiden.

Klasse 8

In der Tabelle 47 werden die gelernten Konzept, mit entsprechenden Genauigkeiten, Laufzeiten und Konzeptlängen, für die Klasse acht der Wissensbasis Cardiotocography dargestellt. Das Konzept, welches OCEL vorgeschlagen hat bedeutet, dass ein Patient, der eine durchschnittliche Langzeitveränderung der Herzfrequenz die größer gleich 1,5 ist besitzt, in die Klasse acht eingeteilt wird. Das Konzept von CELOE unterscheidet sich nur darin, dass die Langzeitveränderung der Herzfrequenz bei CELOE größer gleich 0,5 sein muss, um in die Klasse acht eingeteilt werden zu können. Hier ist zu erkennen, dass sich die Konzepte nur durch die Attributausprägung von *hasMeanValueOfLongTermVariability* unterscheiden, die aufgrund von unterschieden in den Attributausprägungen der Trainingsdaten

zurückzuführen ist.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasMeanValueOfLongTermVariability >= 1.5))</i>	99,061%	0,276s	5
CELOE	<i>(hasMeanValueOfLongTermVariability >= 0.5)</i>	97,183%	0,198s	3

Tabelle 47: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der achten Klasse für die Wissensbasis Cardiotocography dargestellt. Deutlich zu erkennen ist, dass beide Algorithmen fast die gleichen Konzepte lernen, nur die Attributausprägung von *hasMeanValueOfLongTermVariability* ist unterschiedlich.

Klasse 9

Die in der Tabelle 48 dargestellten Klassenbeschreibungen mit Genauigkeit, Laufzeit und Konzeptlänge, sind die Konzepte, welche von OCEL und CELOE vorgeschlagen wurden, um die Klasse neun zu beschreiben. Dabei bedeutet das von OCEL vorgeschlagene Konzept, dass ein Patient, der mehr wie 71,5 andauernde Verringerungen der Herzfrequenz pro Sekunde in die Klasse neun eingeteilt wird. Das Konzept von CELOE hingegen bedeutet, dass ein Patient, der mehr wie 46,5 leichte Verringerungen der Herzfrequenz und eine kleinere Breite des Kardiogramms von 74,5 und mehr wie 63,5 andauernde Verringerungen der Herzfrequenz pro Sekunde oder mehr wie 66,5 andauernde Verringerungen der Herzfrequenz pro Sekunde besitzt, in die Klasse neun eingeteilt wird. Es ist deutlich zu sehen, dass OCEL das kürzere Konzept vorschlägt, welches sich als Teilkonzept bei dem Vorschlag von CELOE wiederfindet nur mit einer anderen Attributausprägung von *hasProlonguedDecelerationsPerSecond*. CELOE hingegen benötigt noch weitere Attribute um die Klasse zu beschreiben. Dies lässt sich darauf zurückführen das der Algorithmus CELOE eine viel geringere interne Bewertung für das Konzept *(Patient AND hasProlonguedDecelerationsPerSecond >= 63.5)*, so dass weitere Attribute hinzugefügt werden müssen um eine ausreichend hohe Bewertung zu erhalten. Auf Grund dieser Unterschiede in der Bewertung der Konzepte kommt es zu dem deutlich längeren Konzept bei CELOE.

Klasse 10

In der Tabelle 49 werden die vorgeschlagenen Klassenbeschreibungen, samt Genauigkeit, Laufzeit und Konzeptlänge, die von den Algorithmen OCEL und CELOE vorgeschlagen wurden. Dabei bedeutet das von OCEL vorgeschlagene Konzept, dass ein Patient, der Varianz im Kardiogramm, die

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasProlonguedDecelerationsPerSecond >= 71.5)</i>	99,057%	0,212s	5
CELOE	<i>(hasLightDecelerationsPerSecond >= 46.5 AND hasWidthOfFHRHistogram <= 74.5 AND (hasProlonguedDecelerationsPerSecond >= 63.5 OR (hasLightDecelerationsPerSecond >= 66.5 AND hasUterineContractionsPerSecond <= 3.5)))</i>	96,714%	82,4s	19

Tabelle 48: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der neunten Klasse für die Wissensbasis Cardiotocography dargestellt. Es ist zu erkennen, dass das Konzept von OCEL viel kürzer ist als das von CELOE und auch als Teilkonzept in dem Vorschlag von CELOE vorkommt, nur mit einer anderen Attributausprägung für *hasProlonguedDecelerationsPerSecond*.

kleiner gleich 0,5 ist und der mehr wie 7,5 anhaltende Verringerungen der Herzfrequenz und der eine durchschnittliche Langzeitveränderung der Herzfrequenz, die größer gleich 3,5 ist oder der mehr wie 0,5 Bewegungen des Fötus pro Sekunde und der weniger als 71,5 anhaltende Verringerungen der Herzfrequenz hat, in die Klasse zehn eingeteilt wird. Das von CELOE vorgeschlagenen Konzept hingegen bedeutet, dass ein Patient, der mehr als 0,5 anhaltende Verringerungen der Herzfrequenz und weniger als 0,75 schwerwiegende Verringerungen der Herzfrequenz in einer Sekunde und mehr wie 59,5 leichte Verringerungen der Herzfrequenz und weniger als 71,5 anhaltende Verringerungen der Herzfrequenz oder eine anhaltende Verringerung der Herzfrequenz zwischen 55,5 und 71,5 hat, in die Klasse zehn eingeteilt wird. Dabei ist deutlich zu erkennen, dass auch hier die Algorithmen OCEL und CELOE unterschiedliche Klassenbeschreibungen lernt. Dies liegt daran, dass durch die unterschiedlichen Testdaten unterschiedliche Attributausprägungen vorlagen. Dadurch wurde der Suchbaum der beiden Algorithmen anders aufgebaut und die darin enthaltenen Konzepte wurden anders bewertet. Das führte dazu das unterschiedliche Pfade im Suchbaum beschriftet wurden, so dass es im Endergebnis dazu führte, das OCEL und CELOE unterschiedliche Konzepte lernen.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasHistogramVariance <= 0.5 AND hasProlonguedDecelerationsPerSecond >= 7.5 AND (hasMeanValueOfLongTermVariability >= 3.5 OR (hasFetalMovementsPerSecond >= 0.5 AND hasProlonguedDecelerationsPerSecond <= 71.5)))</i>	93,427%	68,665s	21
CELOE	<i>(hasProlonguedDecelerationsPerSecond >= 0.5 AND hasSevereDecelerationsPerSecond <= 0.75 AND ((hasLightDecelerationsPerSecond >= 59.5 AND hasProlonguedDecelerationsPerSecond <= 71.5) OR (hasProlonguedDecelerationsPerSecond >= 55.5 AND hasProlonguedDecelerationsPerSecond <= 71.5)))</i>	95,755%	1853,471s	23

Tabelle 49: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der zehnten Klasse für die Wissensbasis Cardiotocography dargestellt. Dabei ist deutlich zu erkennen, dass sich die gelernten Konzepte von OCEL und CELOE deutlich unterscheiden.

Resultate des 3 Klassen Lernproblems

Klasse Normal

In der Tabelle 50 werden, die von den Algorithmen OCEL und CELOE vorgeschlagenen Konzepte inklusive Genauigkeit, Laufzeit und Konzeptlänge vorgestellt, die die Klasse Normal beschreiben sollen. Dabei bedeutet das von OCEL vorgeschlagene Konzept, dass ein Patient, der einen Durchschnittswert im Kardiogramm vom 103,5 oder kleiner und weniger als 39,5 anhaltende Verringerungen der Herzfrequenz pro Sekunde und weniger als 59,5 geringe Verringerungen der Herzfrequenz pro Sekunde oder mehr als 1,5 Kontraktionen der Gebärmutter besitzt, in die Klasse Normal eingeteilt werden kann. Das von CELOE vorgeschlagene Konzept hingegen bedeutet, dass ein Patient, der einen Durchschnittswert im Kardiogramm von 103,5 oder mehr und eine durchschnittliche Langzeitveränderung der Herzfrequenz von 1,5 oder weniger und weniger als 59,5 geringe Verringerungen der Herzfrequenz oder weniger als 7,5 anhaltende Verringerungen der Herzfrequenz besitzt, in die Klas-

se Normal eingeteilt wird. Bei den beiden Vorschlägen fällt auf, dass einige Teilkonzepte in beiden Vorschlägen vorkommen und einige nur in einem von beiden. Dies lässt sich durch die unterschiedlichen Trainingsdaten beim lernen erklären. Sie haben unterschiedliche Attributausprägungen, aber einige der Attributausprägungen sind in beiden Fällen gleich, so dass nur einige Teile der Vorschläge unterschiedlich sind. Die Attribute die gleich vorgeschlagen sind, sind die Hauptattribute, die am wichtigsten für die Beschreibung der Klasse Normal sind. Die anderen Eigenschaften sind auf Grund von unterschiedlicher Bewertung in Suchbaum entstanden.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>((Patient AND hasHistogramMean >= 103.5 AND hasProlonguedDecelerationsPerSecond <= 39.5 AND (hasLightDecelerationsPerSecond <= 59.5 OR hasUterineContractionsPerSecond >= 1.5))</i>	92,019%	37,926s	17
CELOE	<i>(hasHistogramMean >= 103.5 AND hasMeanValueOfLongTermVariability <= 1.5 AND (hasLightDecelerationsPerSecond <= 59.5 OR hasProlonguedDecelerationsPerSecond <= 7.5))</i>	93,897%	1338,225s	15

Tabelle 50: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der Klasse Normal für die Wissensbasis Cardiotocography dargestellt. Dabei ist deutlich zu sehen, dass es Attribute gibt die in beiden Konzepten vorkommen und einige die nur in einem der beiden Vorschläge angezeigt werden.

Klasse Suspect

Die in der Tabelle 51 dargestellten Konzepte mit Genauigkeit, Laufzeit und Konzeptlänge, sind die von OCEL und CELOE vorgeschlagenen Beschreibungen, um die Klasse Suspect zu definieren. Dabei bedeute das von OCEL vorgeschlagene Konzept, dass ein Patient mit anhaltenden Verringerungen der Herzrate pro Sekunde die zwischen 7,5 und 71,5 liegen und einer durchschnittlichen Variabilität der Langzeitveränderung der Herzfrequenz von 0,5 oder größer, oder mehr als 0,5 Bewegungen des Fötus pro Sekunde und weniger als 3,5 Kontraktionen der Gebärmutter besitzt, in die Klasse Suspect eingeteilt wird. Das von CELOE vorgeschlagene Konzept hingegen bedeutet, dass ein Patient, der mehr als 0,5 anhaltende Verringerungen der Herzfrequenz pro Sekunde und einen am häufigsten im

Kardiogramm auftretenden Wert von 150,5 oder größer oder die Anzahl der geringe Verringerungen der Herzfrequenz pro Sekunde von 59,5 oder mehr und weniger als 71,5 anhaltende Verringerungen der Herzfrequenz besitzt, in die Klasse Suspect eingeteilt wird. Dabei fällt auf, dass beide Algorithmen unterschiedliche Konzepte lernen. Dies kann daran liegen, dass unterschiedliche Trainingsdaten, mit unterschiedlichen Attributausprägungen zum Lernen genutzt wurden. Dabei kam es dann zu einem unterschiedlichen Aufbau der Suchbäume und zu unterschiedlichen internen Bewertung der Konzepte. Das führte wiederum dazu, dass unterschiedliche Pfade im Suchbaum durchschritten wurden, so dass am Ende unterschiedliche Konzepte gelernt wurden.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasProlonguedDecelerationsPerSecond >= 7.5 AND hasProlonguedDecelerationsPerSecond <= 71.5 AND (hasMeanValueOfLongTermVariability >= 0.5 OR (hasFetalMovementsPerSecond >= 0.5 AND hasUterineContractionsPerSecond <= 3.5)))</i>	90,566%	93,281s	21
CELOE	<i>(hasProlonguedDecelerationsPerSecond >= 0.5 AND (hasHistogramMode >= 150.5 OR (hasLightDecelerationsPerSecond >= 59.5 AND hasProlonguedDecelerationsPerSecond <= 71.5)))</i>	90,141%	671,155s	15

Tabelle 51: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der Klasse Suspect für die Wissensbasis Cardiotocography dargestellt. Hier ist deutlich zu sehen, dass die Algorithmen OCEL und CELOE unterschiedliche Konzepte lernen.

Klasse Patologic

In der Tabelle 52 sind die Klassenbeschreibungen, inklusive Genauigkeit, Laufzeit und Konzeptlänge, für die beiden Algorithmen OCEL und CELOE dargestellt, die gelernt wurden, um die Klasse Patologic zu beschreiben. Dabei bedeutet das von OCEL vorgeschlagene Konzept, dass ein Patient, der eine durchschnittliche Langzeitveränderung der Herzfrquenz von 0,5 oder mehr und eine Tendenz des Kardiogramms von -0,5 oder kleiner oder eine durchschnittliche Langzeitveränderung der Herzfrequenz von 1,5 oder größer besitzt, in die Klasse Patologic eingeteilt wird. Das von CELOE

vorgeschlagene Konzept hingegen bedeutet, dass ein Patient, der einen im Kardiogramm am häufigsten auftretenden Wert der kleiner gleich 150,5 und mehr als 32,5 geringer Verringerungen der Herzfrequenz pro Sekunde und eine Varianz im Kardiogramm, die größer gleich 48,5 oder mehr als 59,5 geringe Verringerungen der Herzfrequenz pro Sekunde und eine durchschnittliche Langzeitveränderung der Herzfrequenz von 0,5 oder größer oder weniger als 0,5 Erhöhungen der Herzfrequenz und mehr als 115,5 Herzschläge pro Minute besitzt in die Klasse Patologic eingeteilt wird. Es fällt auf, dass die beiden Algorithmen unterschiedliche Konzepte lernen. Dies lässt sich auf die unterschiedliche Bewertung der Konzepte im Suchbaum zurückführen. Dabei wurden von beiden Algorithmen unterschiedliche Pfade weiterverfolgt, so dass es letztendlich zu den unterschiedlichen vorgeschlagenen Konzepten kam.

Algorithmus	Konzept	Genauigkeit	Laufzeit	Konzeptlänge
OCEL	<i>(Patient AND hasMeanValueOfLongTermVariability >= 0.5 AND (hasHistogramTendency <= -0.5 OR hasMeanValueOfLongTermVariability >= 1.5))</i>	96,714%	25,559s	13
CELOE	<i>(hasHistogramMode <= 150.5 AND hasLightDecelerationsPerSecond >= 32.5 AND (hasHistogramVariance >= 48.5 OR hasLightDecelerationsPerSecond >= 59.5) AND (hasMeanValueOfLongTermVariability >= 0.5 OR (hasAccelerationsPerSecond <= 0.5 AND hasBeatsPerMinute >= 115.5)))</i>	88,263%	2544,434s	27

Tabelle 52: In dieser Tabelle werden die Konzepte mit der höchsten Bewertung der Algorithmen OCEL und CELOE der Klasse Patologic für die Wissensbasis Cardiotocography dargestellt. Hier ist deutlich zu erkennen, dass die beiden Algorithmen OCEL und CELOE unterschiedliche Konzepte vorschlagen, die zur Beschreibung der Klasse Patologic dienen.

13 Schlussfolgerungen

In diesem Kapitel werden die wichtigen Erkenntnisse dieser Arbeit noch einmal zusammengefasst. Dabei wird kurz auf die Besonderheiten der einzelnen Wissensbasen eingegangen.

In der Abbildung 40 sind die Genauigkeiten aller Lernalgorithmen, für alle Klassen jeder Wissensbasis die getestet wurde, dargestellt. Dabei wurden diese Ergebnisse aus den Einzelergebnissen errechnet indem der Durchschnittswert aus jeder Klasse gebildet wurde. Es ist zu sehen, dass CELOE die höchste Genauigkeit erreicht, dicht gefolgt von OCEL. Daran schließen sich die Algorithmen Random Forrest, PART, Random Tree, Decision Table und Aleph an. Die Algorithmen, die die schlechtesten Ergebnisse Produzieren sind die Algorithmen Naive Bayes und Naive Bayes Updateable. Des weiteren ist zu erkennen, dass es bei OCEL und CELOE zu einem Abfall der Testgenauigkeit kommt. Dies liegt daran, dass die Algorithmen Konzepte lernen, die auf die Trainingsdaten gut passen, aber für die Testdaten nicht gut geeignet sind oder zu spezifisch sind. Im Folgenden werden die Ergebnisse der einzelnen Wissensbasen kurz Zusammengefasst.

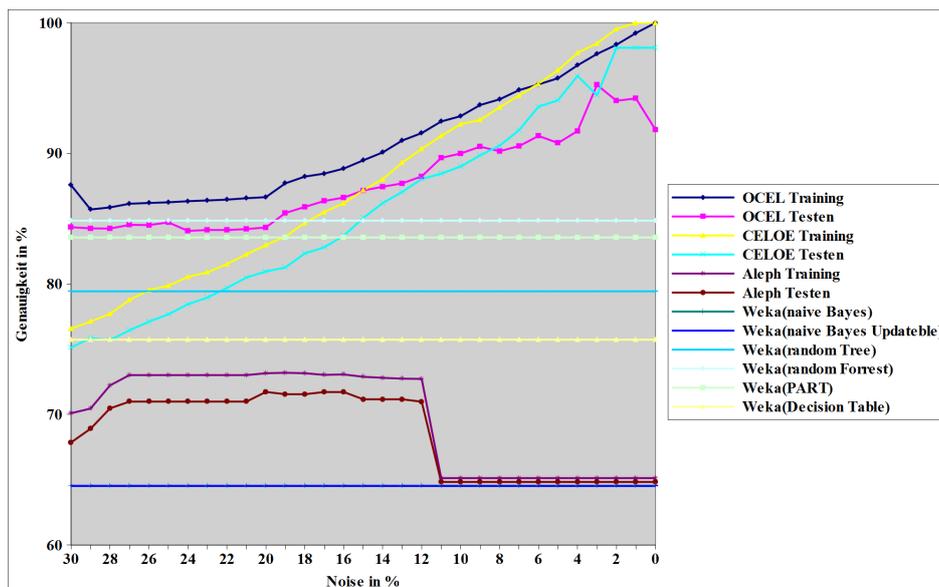


Abbildung 40: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für alle Klassen. Es ist zu sehen, dass CELOE die höchste Genauigkeit erzielt. Daran schießen sich die Algorithmen OCEL, Random Forrest, PART, Random Tree, Decision Table und Aleph an. Die schlechtesten Ergebnisse erreichten die Algorithmen Naive Bayes und Naive Bayes Updateable.

Bei der Wissensbasis Mutagenesis zeigte sich, dass der Algorithmus CELOE leicht bessere Ergebnisse lieferte als OCEL. Dabei zeigte sich aber dass diese Unterschiede nicht statistisch signifikant sind.

Des Weiteren wurde Aleph für den Vergleich eingesetzt und bei diesem zeigte sich, dass OCEL und CELOE statistisch signifikant bessere Ergebnisse liefern.

Für die Wissensbasis Suramin wurden ebenfalls die Algorithmen OCEL, CELOE und Aleph zum Testen verwendet. Dabei zeigte sich, dass OCEL die höchste Testgenauigkeit erzielt hat dicht gefolgt von CELOE. Auch erreichte Aleph nur die Ergebnisse mit der schlechtesten Genauigkeit. Es zeigte sich, dass die Unterschiede zwischen OCEL und CELOE nicht statistisch signifikant waren, aber beide Algorithmen statistisch signifikant besser arbeiten als Aleph.

Die Parkinson Wissensbasis wurde mit Hilfe der Algorithmen OCEL, CELOE und den Algorithmen von Weka getestet. Dabei erreicht CELOE die höchste Genauigkeit dicht gefolgt von OCEL. Daran schlossen sich die Algorithmen von Weka an. Es zeigte sich das sowohl OCEL als auch CELOE bei allen Algorithmen bis auf Random Forrest statistisch signifikant bessere Ergebnisse liefern. Bei einem direkten Vergleich zwischen OCEL und CELOE konnte auch keine statistische Signifikanz festgestellt werden.

Die für die Wissensbasis Heart durchgeführten Tests, zeigen, dass hier der Algorithmus OCEL nur die viert beste Genauigkeit erreicht hat und CELOE die schlechteste. Dabei war OCEL statistisch signifikant schlechter als Decision Table, Naive Bayes und Naive Bayes Updateable. CELOE hingegen war auch schlechter als Random Forrest. Nur in dem direkten Vergleich zwischen den beiden Algorithmen OCEL und CELOE konnte keine statistische Signifikanz festgestellt werden.

Bei der Wissensbasis Mammographic erreichte der Algorithmus OCEL die fünft beste Genauigkeit und CELOE erzielte nur die Ergebnisse mit der schlechtesten Genauigkeit. Dabei waren aber die Ergebnisse von OCEL in keinem Fall statistisch signifikant schlechter als die Ergebnisse der Algorithmen, die bessere Ergebnisse erzielten. Bei CELOE hingegen waren alle Algorithmen statistisch signifikant besser bis auf Random Forrest. Auch bei dieser Wissensbasis konnte keine statistische Signifikanz zwischen den Unterschieden der beiden Algorithmen OCEL und CELOE festgestellt werden.

Für die Wissensbasis Breast Tissue erreichte CELOE wieder die höchste Genauigkeit dicht gefolgt von Decision Table und an dritter Stelle kam OCEL. Die schlechtesten Ergebnisse wurden von Naive Bayes und Naive Bayes Updateable erzielt. Dabei wurde bei dem t-Test festgestellt, das sowohl OCEL als auch CELOE statistisch signifikant bessere Ergebnisse liefert als Naive Bayes und Naive Bayes Updateable. Bei einem direkten Vergleich zwischen OCEL und CELOE konnte keine statistische Signifikanz nachgewiesen werden.

Die letzte Wissensbasis Cardiocography wurde in ein drei und zehn Klassen Lernproblem eingeteilt. Dabei wurde festgestellt, dass bei beiden Lernproblemen OCEL die höchste Genauigkeit erreichte, dicht gefolgt von CELOE. Die schlechteste Genauigkeit erreicht Naive Bayes und Naive Bayes Updateable. Auch bei den t-Tests wurde festgestellt, dass bei beiden Lernproblemen OCEL und CELOE

statistisch signifikant bessere Ergebnisse produzieren als die Algorithmen von Weka. Nur bei einem direkten Vergleich zwischen OCEL und CELOE gab es bei den beiden Lernproblemen Unterschiede. Denn bei dem zehn Klassen Lernproblem arbeitet OCEL statistisch signifikant besser als CELOE und bei dem drei Klassen Lernproblem konnte keine statistische Signifikanz zwischen den beiden Algorithmen festgestellt werden.

Zusammenfassend ist hier zu sagen, dass beiden Algorithmen OCEL und CELOE gute Ergebnisse liefern und in vielen Fällen sogar statistisch signifikant besser arbeiten als die anderen Werkzeuge. Dabei ist aufgefallen, dass der Algorithmus CELOE bei nicht hierarchisch aufgebauten Ontologien schlechtere Ergebnisse liefert als OCEL, aber bei hierarchisch aufgebauten Ontologien bessere Ergebnisse liefert als OCEL. Daraus folgt, dass der Algorithmus nicht so gut mit Ontologien umgehen kann und nicht so hohe Genauigkeiten wie OCEL erzielt, wenn die viele Data Properties in diesen Ontologien enthalten sind und deswegen keine Hierarchie aufgebaut wird.

14 Zusammenfassung und Ausblick

Zusammenfassend ist zu sagen, dass in dieser Arbeit ein erster Schritt, für die Analyse ob Ontologielernverfahren im medizinischen Bereich von Nutzen sein kann, getan wurde. Dazu wurden eine Reihe von Wissensbasen in OWL-Ontologien umgewandelt und mit dem DL-Learner getestet. Des Weiteren wurden diese Ergebnisse mit dem ILP Werkzeug Aleph und dem Tool Weka, aus dem Bereich des Machine Learning, verglichen. Dazu wurde verschiedene Tests durchgeführt. Zum einen wurde, wenn dies möglich war der Noise-Parameter von einem vorher ausgewählten Wert in 1% Schritten reduziert und eine 10-fold-cross-validation durchgeführt. Des Weiteren wurde auch die Laufzeit auf einen festen Wert gesetzt und wieder eine 10-fold-cross-validation durchgeführt. Dabei kam heraus, dass sich Ontologielernverfahren, die hier mit Hilfe des DL-Learners durchgeführt wurden, gute Ergebnisse liefern. Je nach Wissensbasis lieferten die Algorithmen OCEL und CELOE die besten Ergebnisse, aber bei einigen konnten die Ergebnisse auch unter denen der anderen Werkzeuge liegen. Des Weiteren ist zu sagen, dass bei den Ontologien die nicht hierarchisch aufgebaut waren, erreichte der Lernalgorithmus CELOE deutlich geringere Genauigkeiten als OCEL.

Da es im Bereich des Machine Learning viele weitere Algorithmen und Ansätze zum Lösen von Lernproblemen existieren sollten noch weitere Tests mit anderen Algorithmen durchgeführt werden. Es könnten weitere Algorithmen wie den ETLT-Learner des DL-Learners zum Testen verwendet werden. Auch könnte man weitere Tests hinzufügen, wie das Lernen mit einer vorgegebenen Anzahl an positiven und negativen Beispielen, so dass nur ein Teil zum Lernen verwendet wird. Da einige der Wissensbasen schon etwas älter sind, in dem Fall die Muatgenesis und Suramin Wissensbasis, sollte hier versucht werden aktuellere Daten zu bekommen und die Lernergebnisse eventuell noch verbessern zu können. Des Weiteren könnte mit Hilfe des Web of Data versucht werden zusätzliches Wissen in die Ontologie zu integrieren um auch hier eventuell eine Verbesserung der Lernergebnisse zu erzielen. In einem weiteren Schritt sollten Wissensbasen gesucht werden, die größer sind, so dass auch das Verhalten der Werkzeuge bei größeren Wissensbasen zu überprüfen. Dies ist wichtig, da im medizinischen Bereich sehr viele Daten aufkommen und somit Werkzeuge zur Verfügung stehen müssen, die mit einer großen Datenmenge umgehen können.

Abschließend ist zu sagen, dass es sich bei dieser Arbeit um einen ersten Schritt handelt, um zu überprüfen, ob sich Ontologielernverfahren für den medizinischen Bereich eignen. Die Ergebnisse dieser Arbeit zeigen zwar, dass sich Ontologielernverfahren eignen können, es müssten aber weitere Tests durchgeführt werden, um diese Frage besser klären zu können.

A Grafiken für die Breast Tissue Wissensbasis

A.1 Grafiken für die Variation des Noise-Parameters

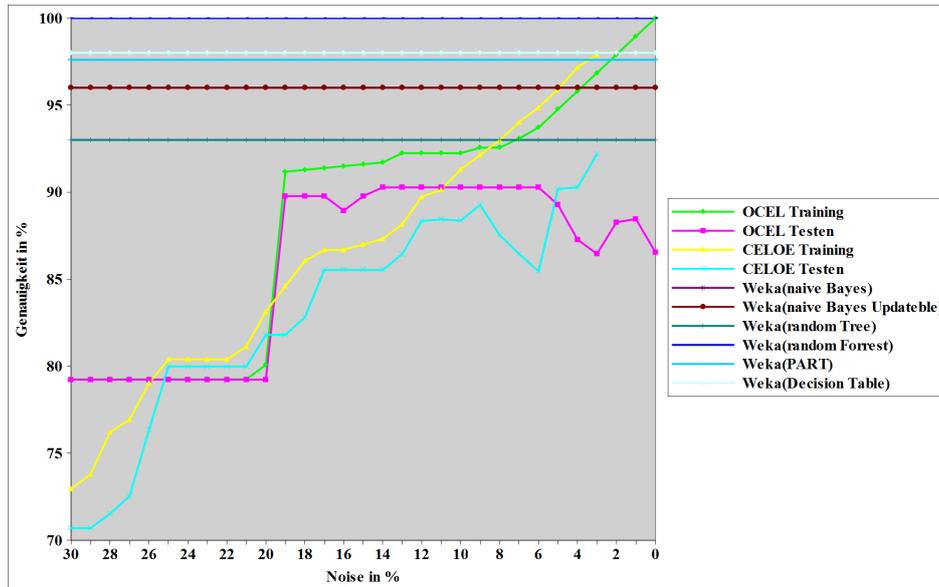


Abbildung 41: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Klasse Carcinoma der Wissensbasis Breast Tissue. Der Abfall der Genauigkeit bei CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind.

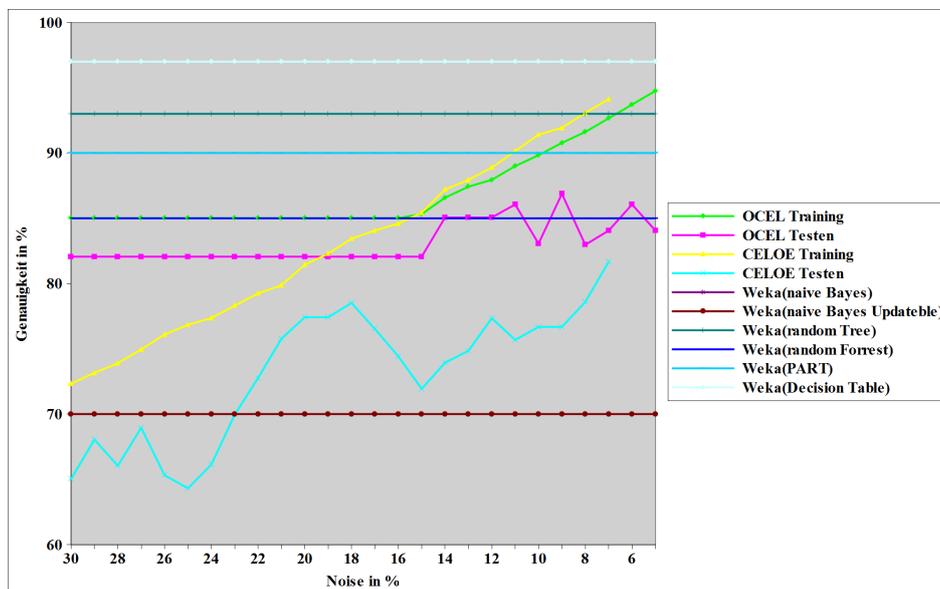


Abbildung 42: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Klasse Fibro-adenoma der Wissensbasis Breast Tissue. Der Abfall der Genauigkeit bei OCEL - und CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind.

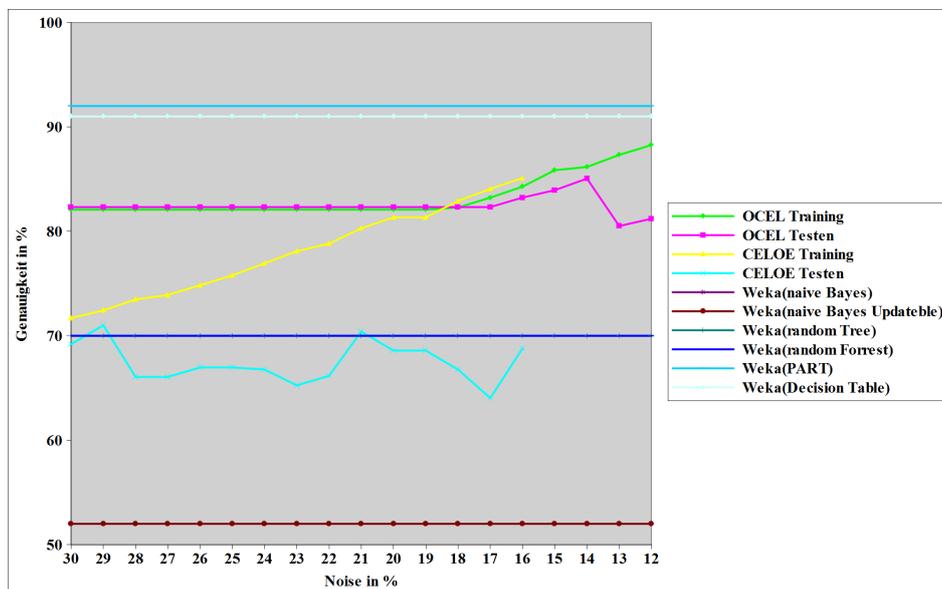


Abbildung 43: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Klasse Mastopathy der Wissensbasis Breast Tissue. Der Abfall der Genauigkeit bei OCEL - und CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind.

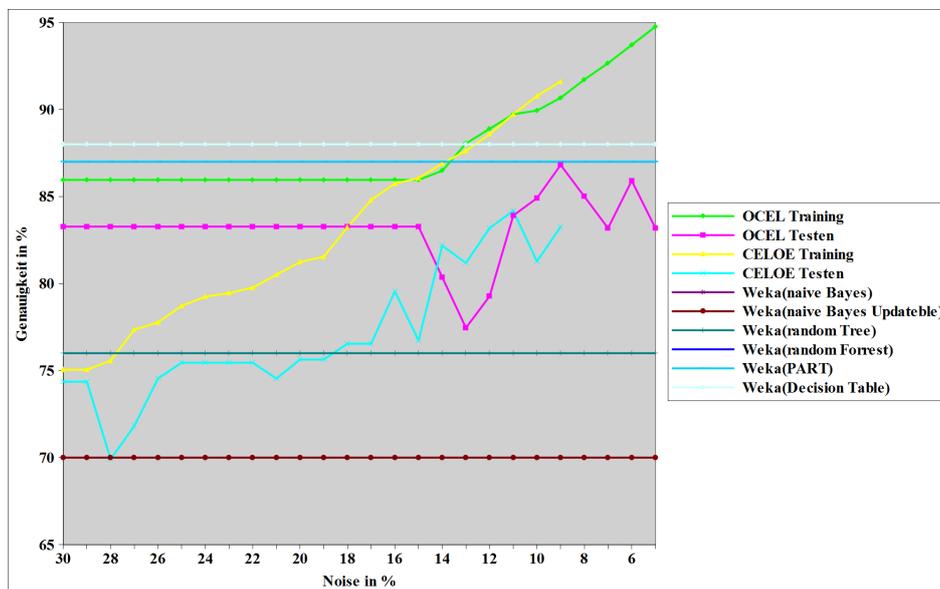


Abbildung 44: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Klasse Glandular der Wissensbasis Breast Tissue. Der Abfall der Genauigkeit bei CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind.

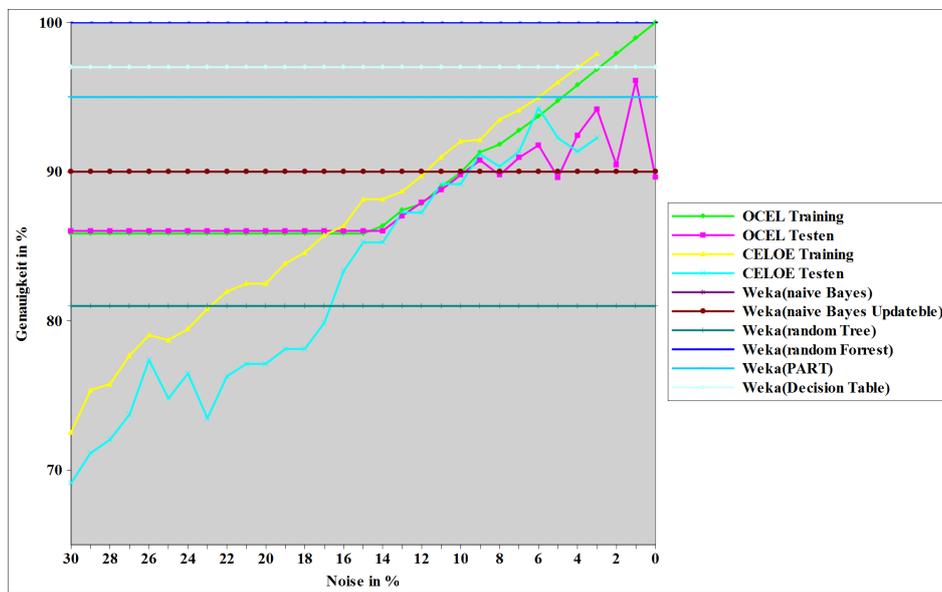


Abbildung 45: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Klasse Connective der Wissensbasis Breast Tissue. Der Abfall der Genauigkeit bei OCEL - und CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind.

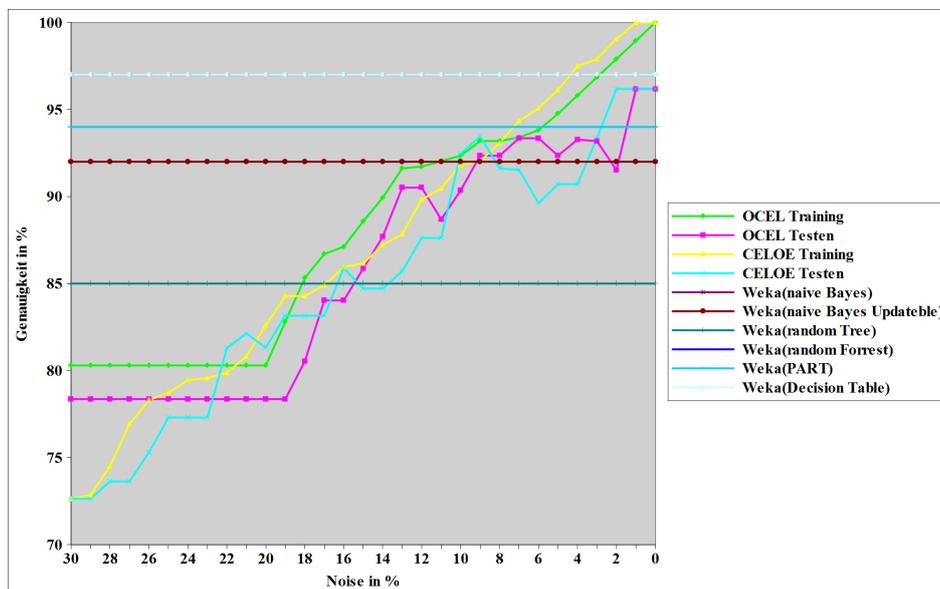


Abbildung 46: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Klasse Adipose der Wissensbasis Breast Tissue. Der Abfall der Genauigkeit bei CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind.

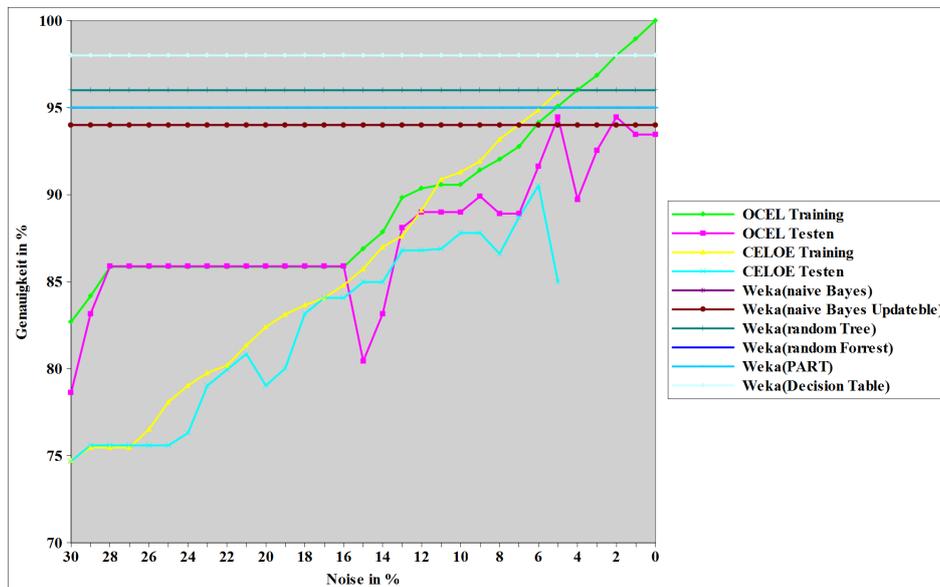


Abbildung 47: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Klassen Fibro-adenoma, Mastopathy und Glandula der Wissensbasis Breast Tissue. Da die Unterscheidung zwischen diesen Klassen schwierig und nicht zwingend nötig ist, wurden diese für den Lernprozess als eine Klasse zusammengefasst. Der Abfall der Genauigkeit bei OCEL - und CELOE Testen lässt sich damit erklären, dass der DL-Lerner Konzepte lernt, die auf die Trainingsdaten passen, aber für die Testdaten weniger geeignet sind.

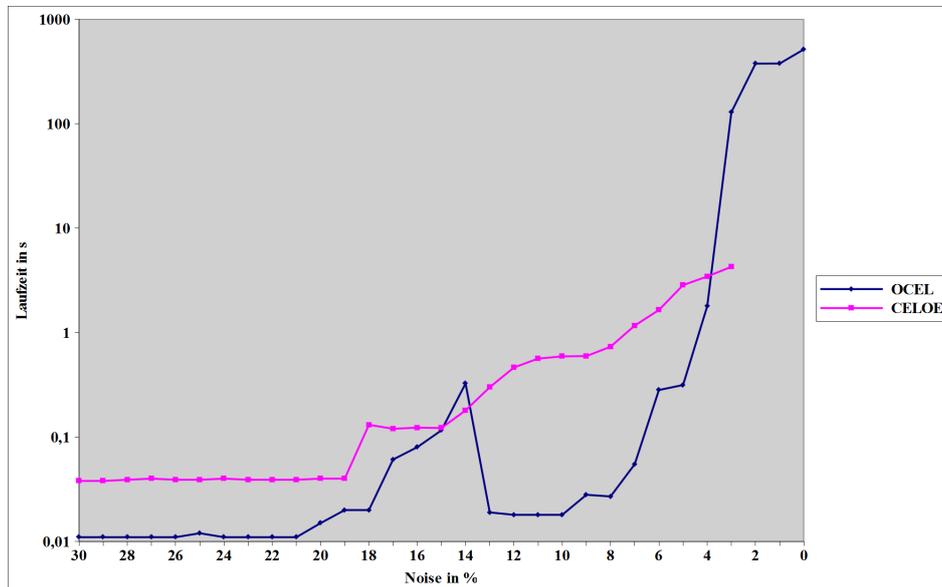
A.2 Grafiken für die benötigte Laufzeit bei Variation des Noise Parameters

Abbildung 48: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse Carcinoma der Wissensbasis Breast Tissue. Die y-Achse ist logarithmisch dargestellt.

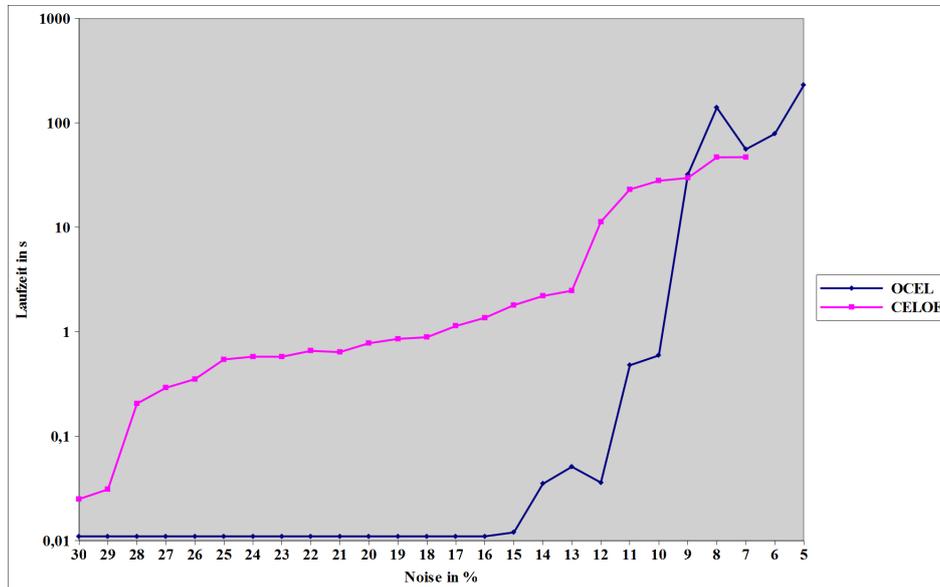


Abbildung 49: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse Fibro-adenoma der Wissensbasis Breast Tissue. Die y-Achse ist logarithmisch dargestellt.

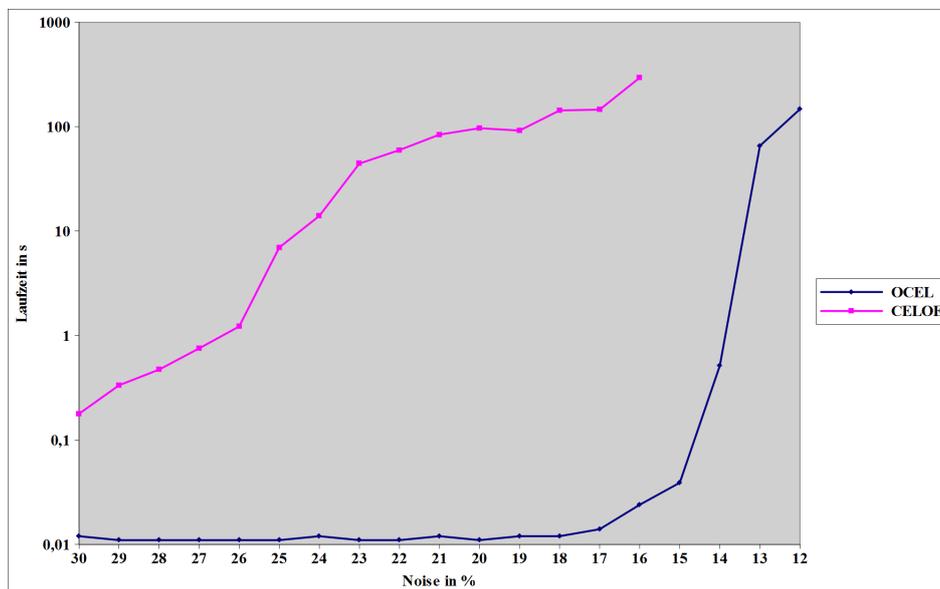


Abbildung 50: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse Mastopathy der Wissensbasis Breast Tissue. Die y-Achse ist logarithmisch dargestellt.

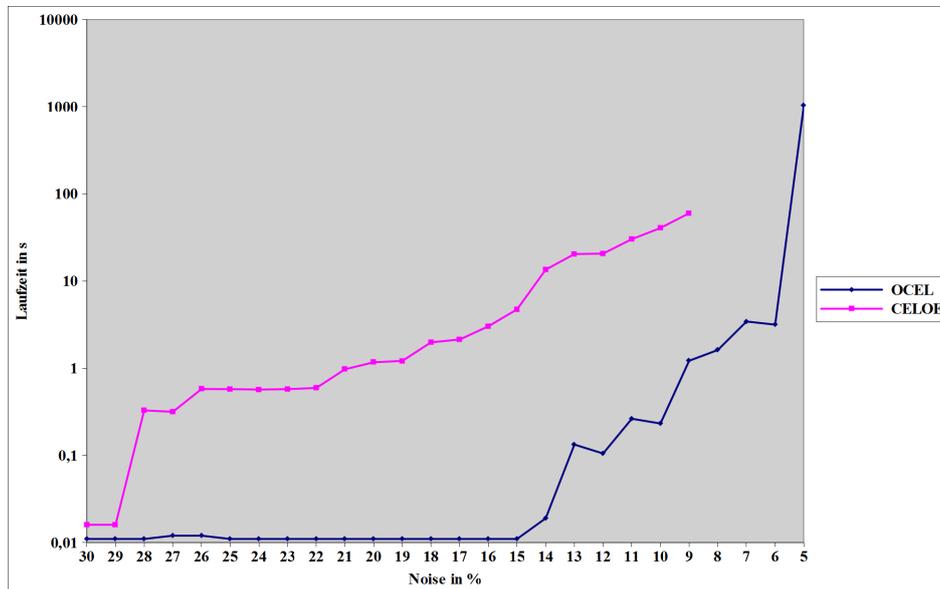


Abbildung 51: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse Glandular der Wissensbasis Breast Tissue. Die y-Achse ist logarithmisch dargestellt.

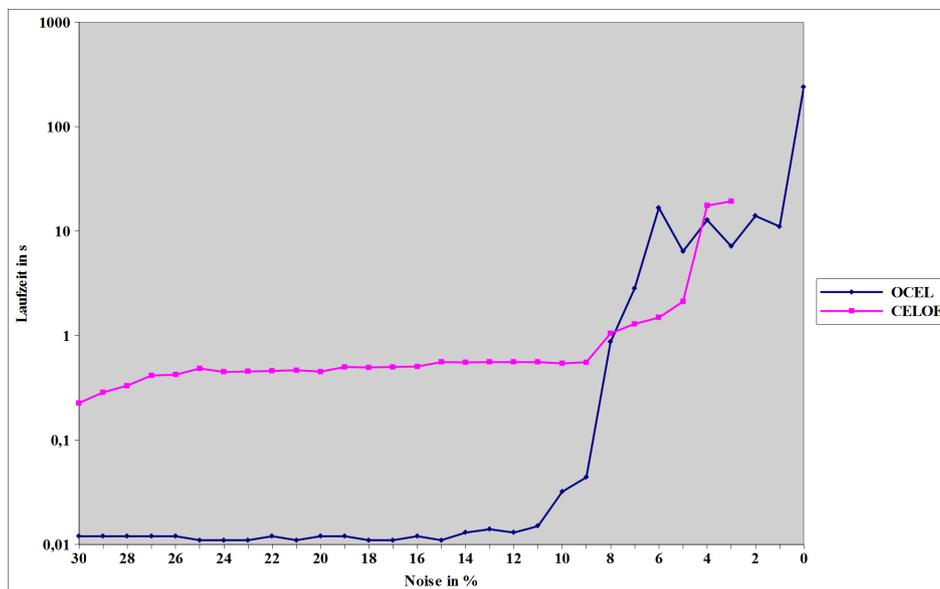


Abbildung 52: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse Connective der Wissensbasis Breast Tissue. Die y-Achse ist logarithmisch dargestellt.

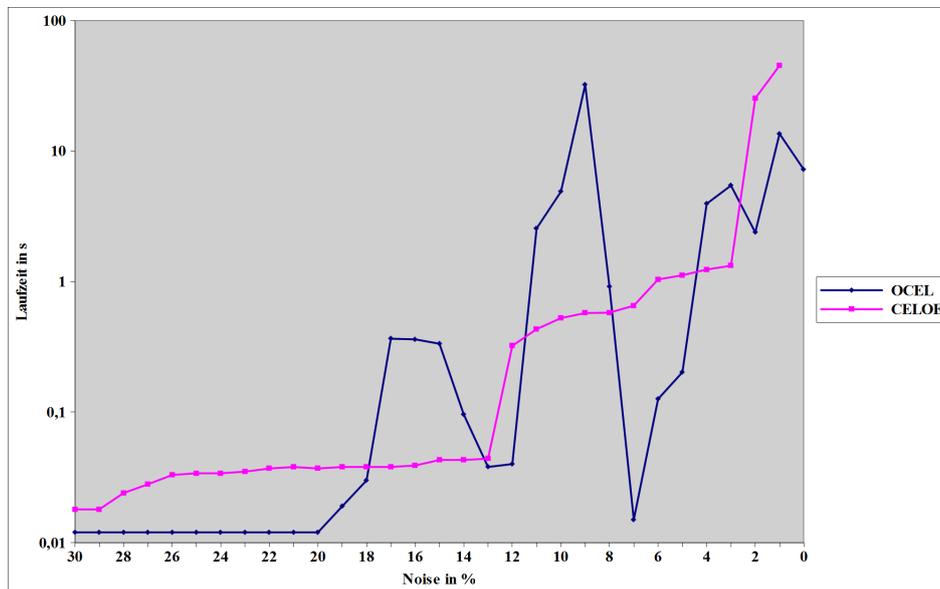


Abbildung 53: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse Adipose der Wissensbasis Breast Tissue. Die y-Achse ist logarithmisch dargestellt.

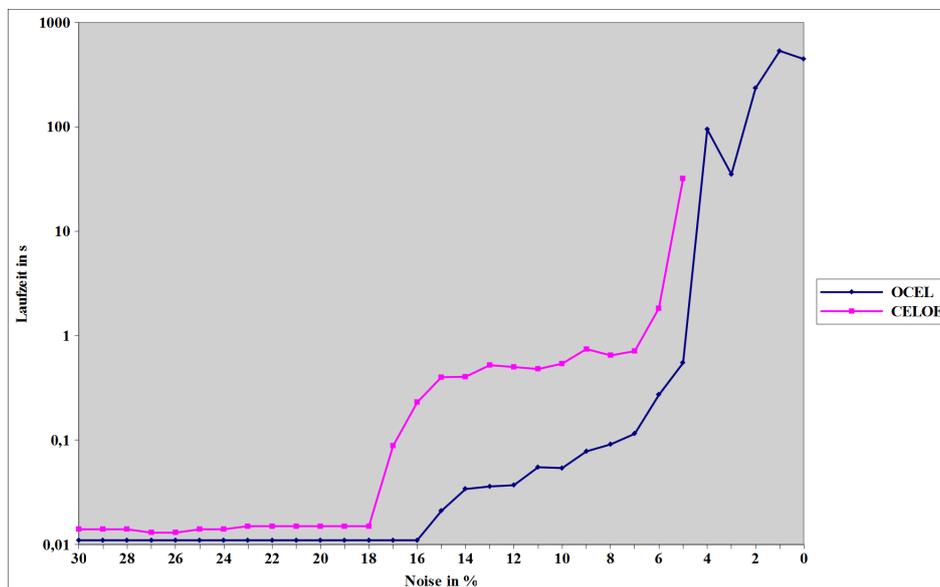


Abbildung 54: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klassen Fibro-adenoma, Mastopathy und Glandula der Wissensbasis Breast Tissue. Die y-Achse ist logarithmisch dargestellt.

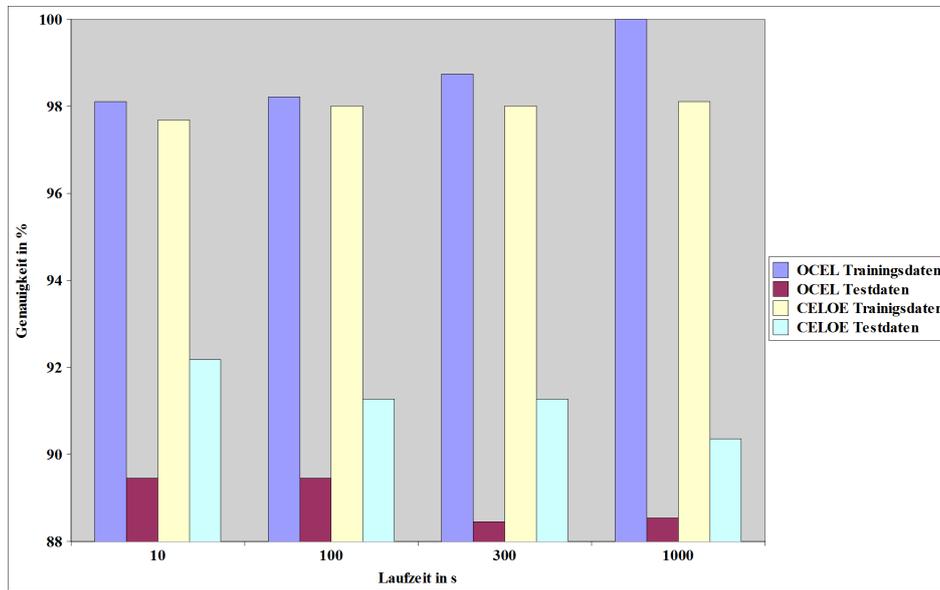
A.3 Grafiken für die fest eingestellte Laufzeit von 10, 100, 300 und 1000 Sekunden

Abbildung 55: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse Carcinoma der Wissensbasis Breast Tissue.

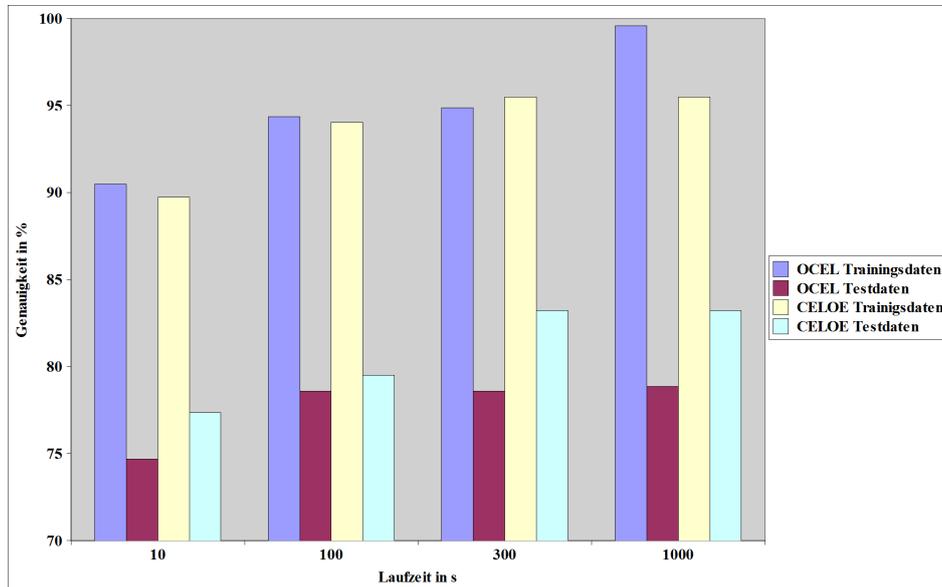


Abbildung 56: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse Fibro-adenoma der Wissensbasis Breast Tissue.

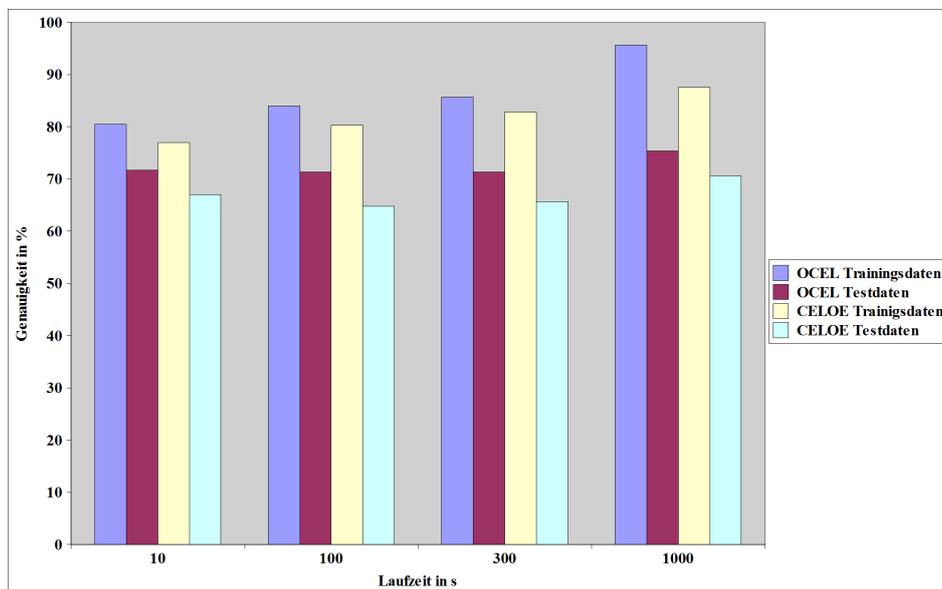


Abbildung 57: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse Mastopathy der Wissensbasis Breast Tissue.

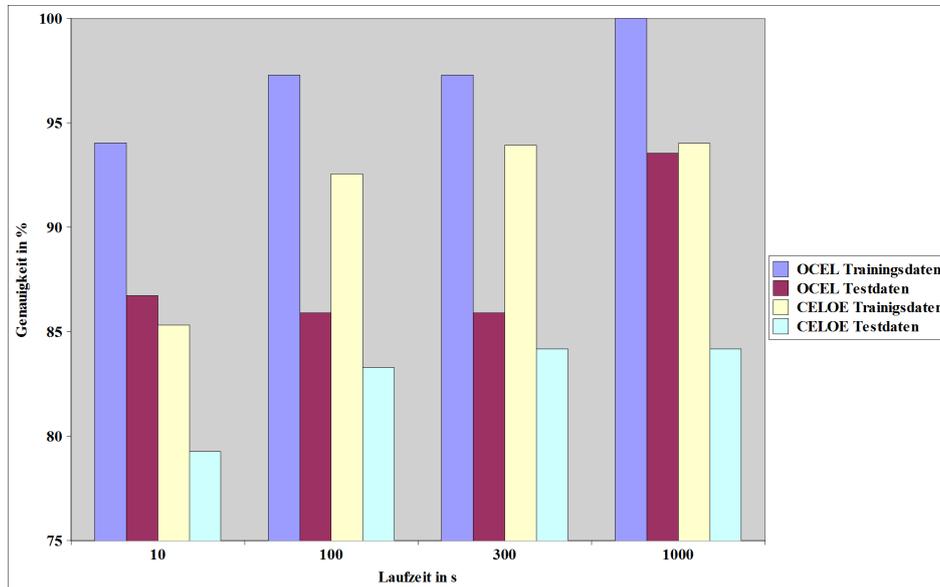


Abbildung 58: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse Glandular der Wissensbasis Breast Tissue.

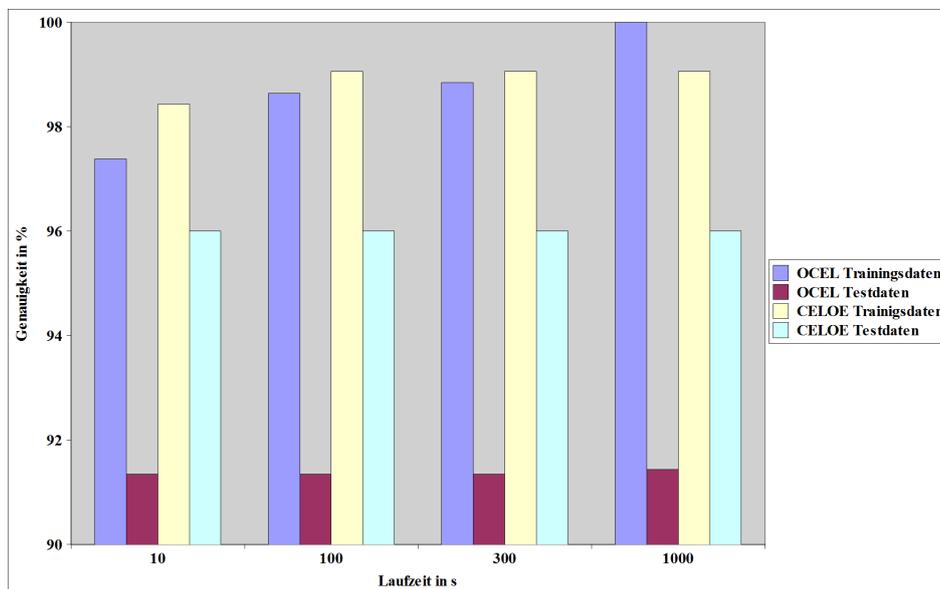


Abbildung 59: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse Connective der Wissensbasis Breast Tissue.

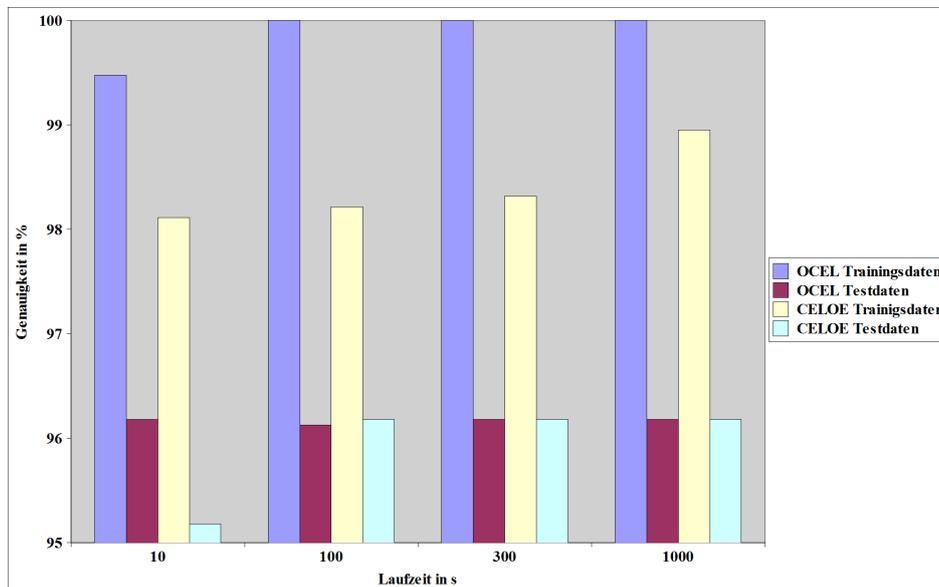


Abbildung 60: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse Adipose der Wissensbasis Breast Tissue.

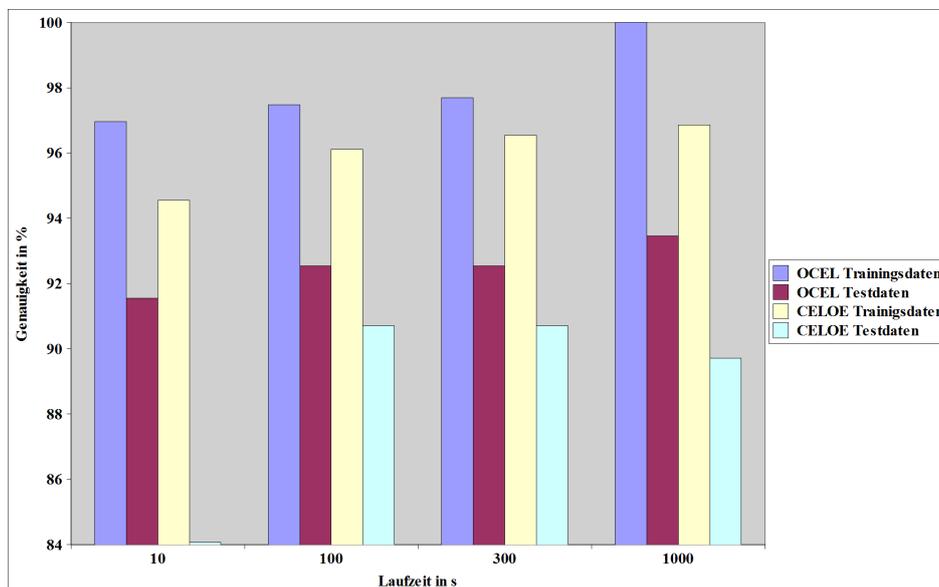


Abbildung 61: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klassen Fibro-adenoma, Mastopathy und Glandula der Wissensbasis Breast Tissue. Da die Unterscheidung zwischen diesen Klassen schwierig und nicht zwingend nötig ist, wurden diese für den Lernprozess als eine Klasse zusammengefasst.

B Grafiken für die Cardiotocography Wissensbasis

B.1 Grafiken für die Variation des Noise-Parameters

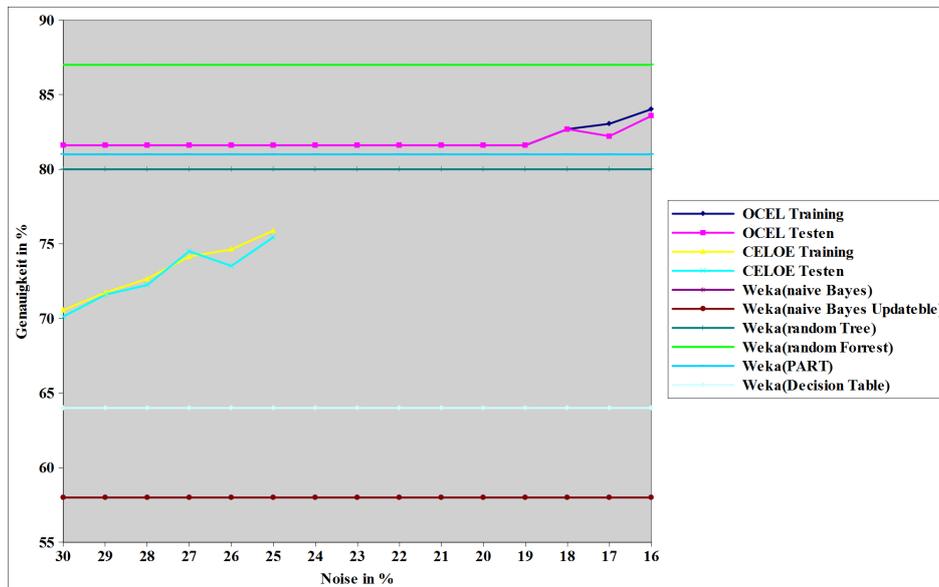


Abbildung 62: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die erste Klasse der Wissensbasis Cardiotocography. Es ist zu sehen, dass CELOE bereits bei 24% Noise keine Ergebnisse mehr produziert. Des Weiteren ist zu sehen, dass Random Forrest die höchste Genauigkeit erzielt, gefolgt von OCEL. Im Anschluss daran kommt der Algorithmus PART. Die viert beste Genauigkeit erreichte Random Tree gefolgt von CELOE und Decision Table. Die schlechtesten Ergebnisse produzieren die Algorithmen Naive Bayes und Naive Bayes Updateable.

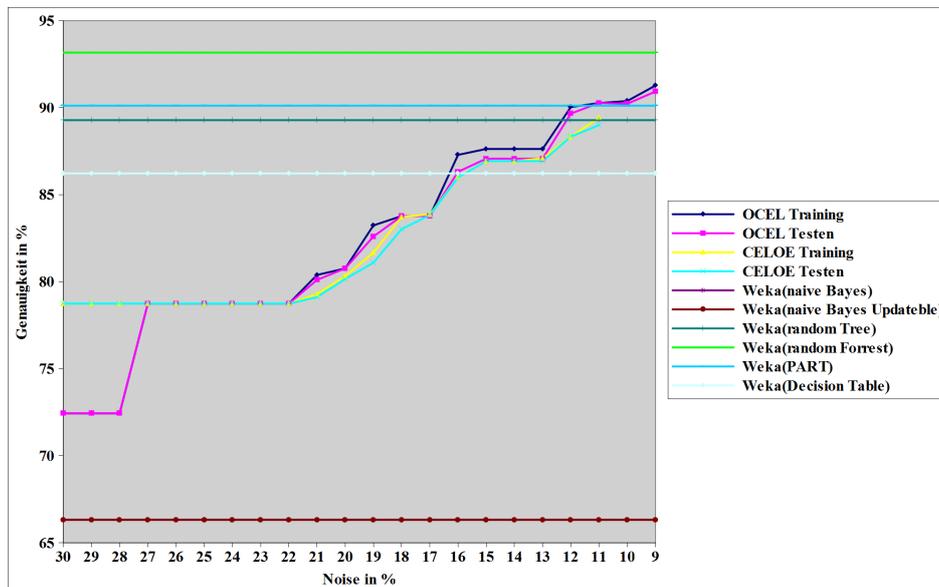


Abbildung 63: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die zweite Klasse der Wissensbasis Cardiotoctography. Es ist zu sehen, dass Random Forrest die höchste Genauigkeit erzielt, gefolgt von OCEL, PART und Random Tree. Daran schließt sich CELO an, was von Decision Table gefolgt wird. Die schlechteste Genauigkeit erreichte Naive Bayes und Naive Bayes Updateable. Des Weiteren ist zu sehen, dass CELOE bei 10% Noise keine Ergebnisse mehr produziert und OCEL erst bei 8%.

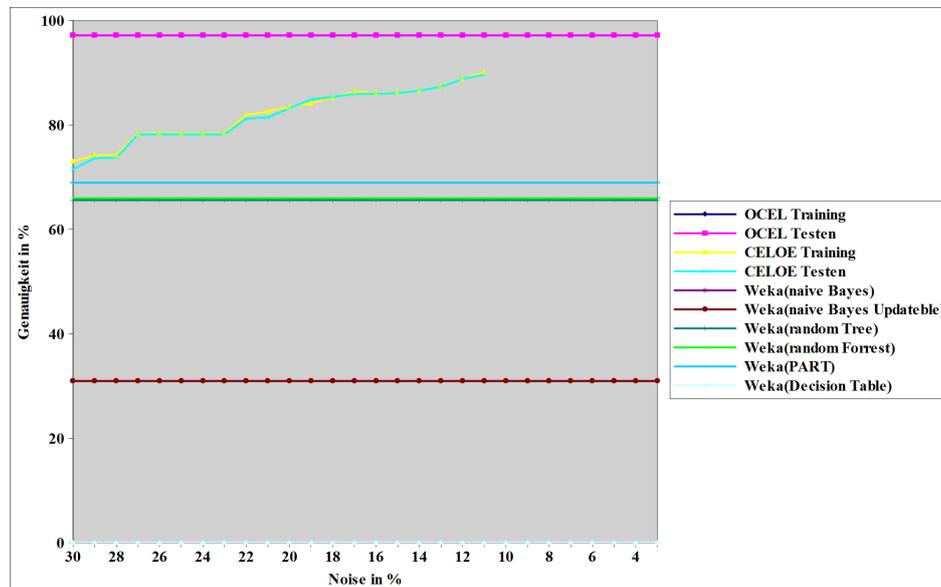


Abbildung 64: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die dritte Klasse der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL die höchste Genauigkeit erzielt, gefolgt von CELOE. Daran schließen sich an die Algorithmen PART, Random Forrest, Random Tree, Naive Bayes und naive Bayes Updateable. Die schlechtesten Ergebnisse liefert Decision Table mit 0% Genauigkeit. Des Weiteren ist zu sehen, dass CELOE bei 10% Noise keine Ergebnisse mehr produziert und OCEL erst bei 2%.

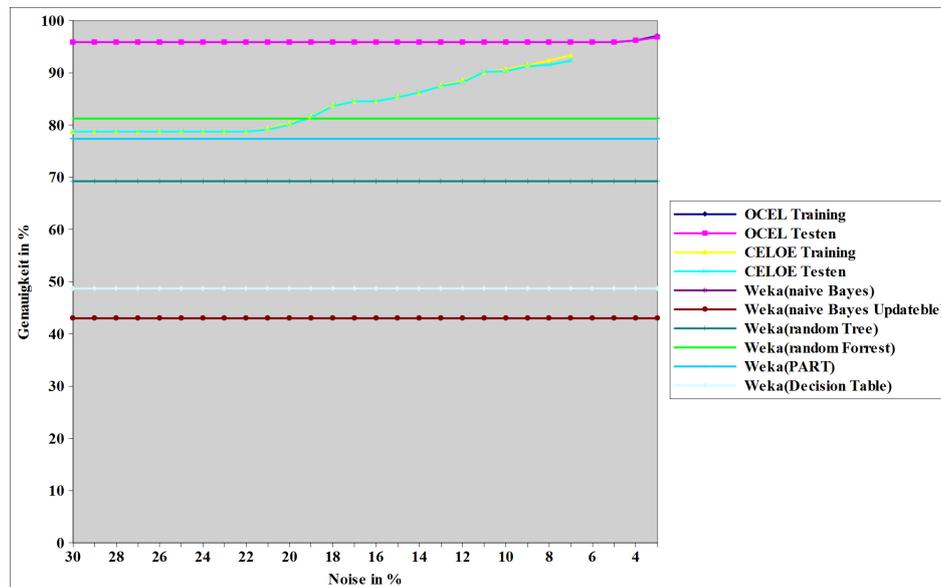


Abbildung 65: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die vierte Klasse der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL die höchste Genauigkeit erzielt, gefolgt von CELOE. Daran schließen sich an die Algorithmen Random Forrest, PART, Random Tree und Decision Table. Naive Bayes und Naive Bayes Updateable erreichen die schlechtesten Genauigkeiten. Des Weiteren ist zu sehen, dass CELOE bei 6% Noise keine Ergebnisse mehr produziert und OCEL erst bei 2%.

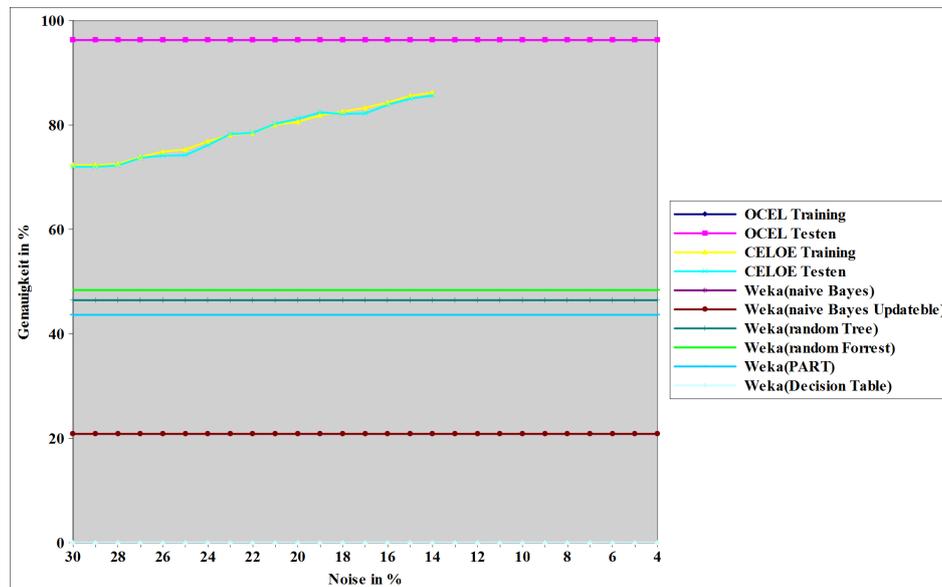


Abbildung 66: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die fünfte Klasse der Wissensbasis Cardiotoctography. Es ist zu sehen, dass OCEL die höchste Genauigkeit erzielt, gefolgt von CELOE. Daran schließen sich an die Algorithmen Random Forrest, Random Tree, PART, Naive Bayes und Naive Bayes Updateable. Decision Table erreicht mit 0% die schlechteste Genauigkeit. Des Weiteren ist zu sehen, dass CELOE bei 13% Noise keine Ergebnisse mehr produziert und OCEL erst bei 3%.

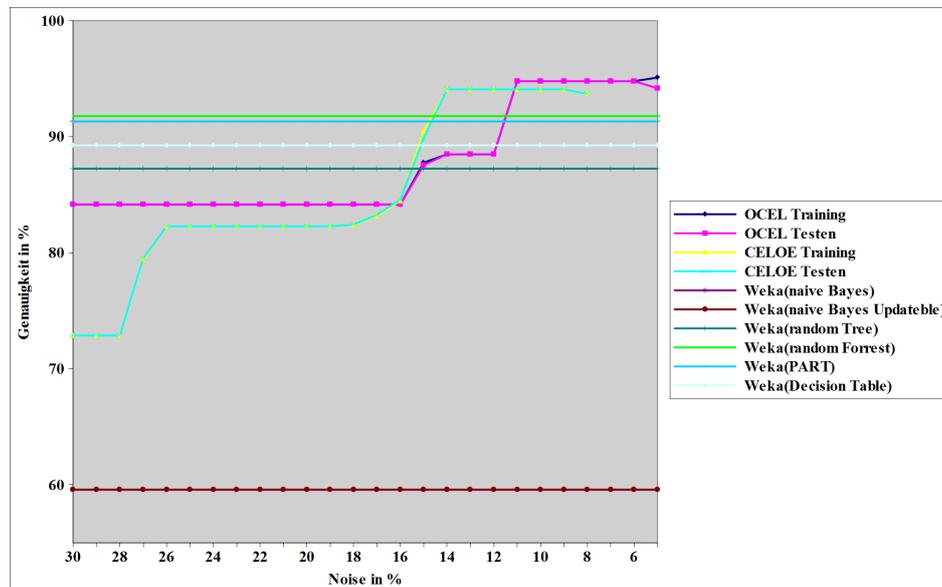


Abbildung 67: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die sechste Klasse der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL die höchste Genauigkeit erzielt, gefolgt von CELOE. Daran schießen sich an die Algorithmen Random Forrest, PART, Decision Table und Random Tree. Die schlechtesten Ergebnisse erreichten die Algorithmen Naive Bayes und Naive Bayes Updateable. Des Weiteren ist zu sehen, dass CELOE bei 7% Noise keine Ergebnisse mehr produziert und OCEL erst bei 4%.

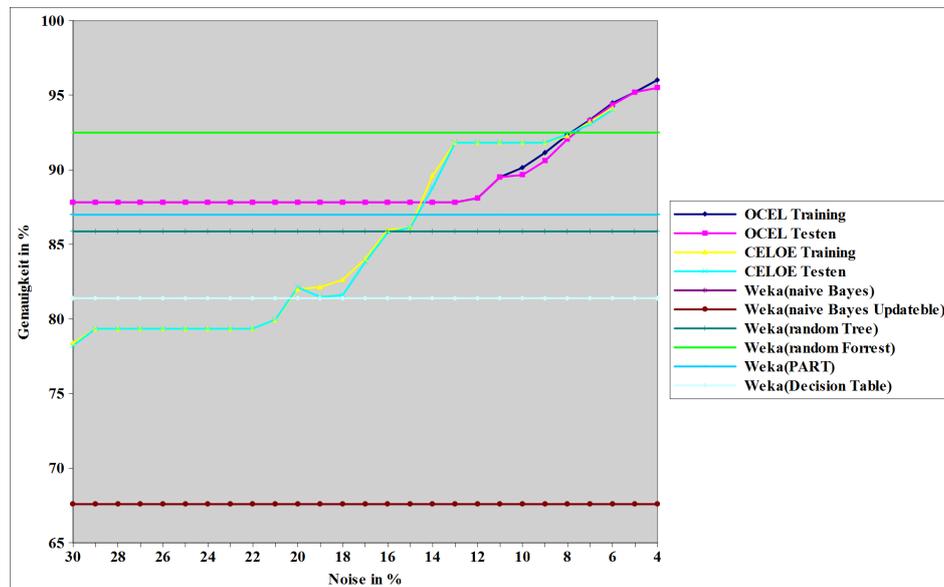


Abbildung 68: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die siebte Klasse der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL die höchste Genauigkeit erzielt, gefolgt von CELOE. Daran schließen sich an die Algorithmen Random Forrest, PART, random Tree und Decision Table. Die schlechtesten Ergebnisse erreichten die Algorithmen Naive Bayes und Naive Bayes Updateable. Des Weiteren ist zu sehen, dass CELOE bei 5% Noise keine Ergebnisse mehr produziert und OCEL erst bei 3%.

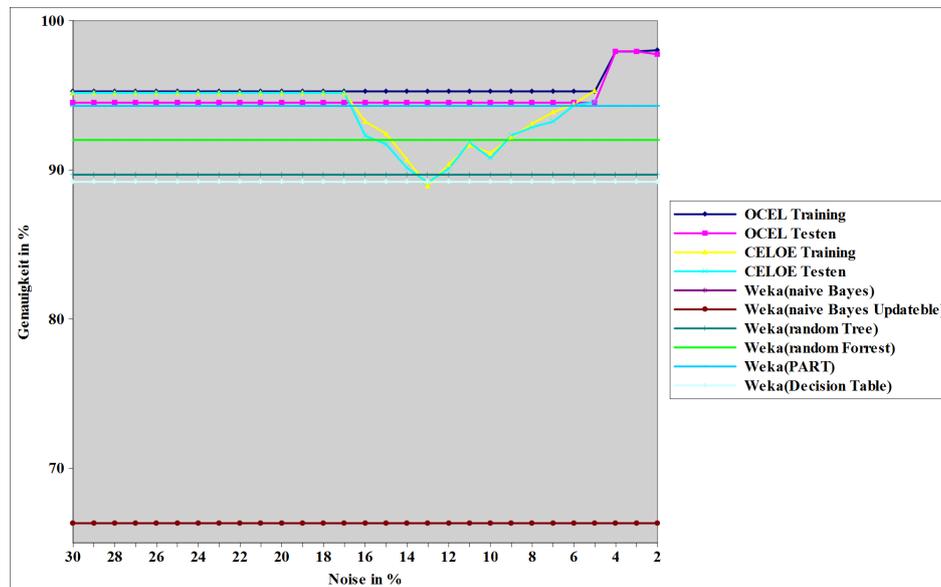


Abbildung 69: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die achte Klasse der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL die höchste Genauigkeit erzielt, gefolgt von CELOE. Daran schießen sich an die Algorithmen PART, Random Forrest, Random Tree und Decision Table. Die schlechtesten Ergebnisse erreichten die Algorithmen Naive Bayes und Naive Bayes Updateable. Des Weiteren ist zu sehen, dass CELOE bei 4% Noise keine Ergebnisse mehr produziert und OCEL erst bei 1%.

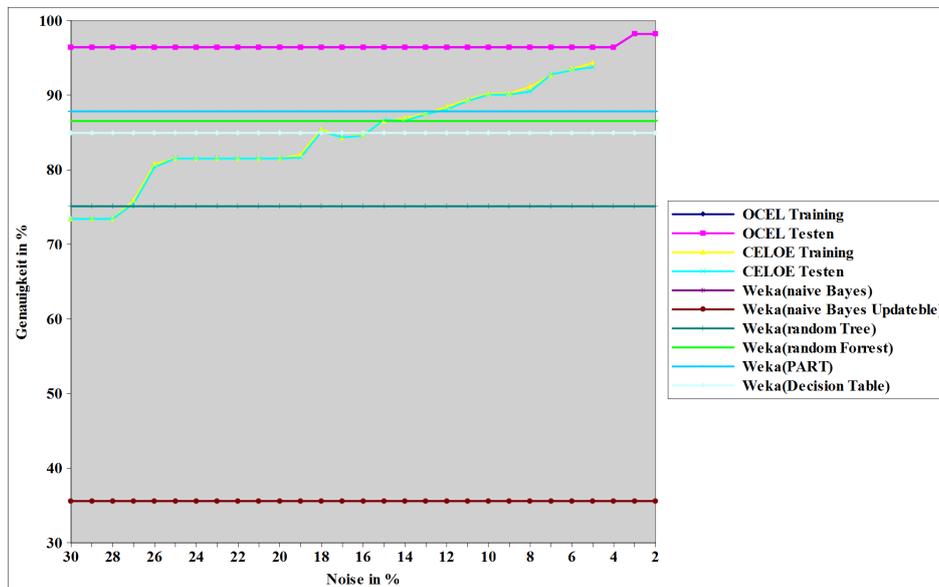


Abbildung 70: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die neunte Klasse der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL die höchste Genauigkeit erzielt, gefolgt von CELOE. Daran schießen sich an die Algorithmen PART, Random Forrest, Decision Table und Random Tree. Die schlechtesten Ergebnisse erreichten die Algorithmen Naive Bayes und Naive Bayes Updateable. Des Weiteren ist zu sehen, dass CELOE bei 4% Noise keine Ergebnisse mehr produziert und OCEL erst bei 1%.

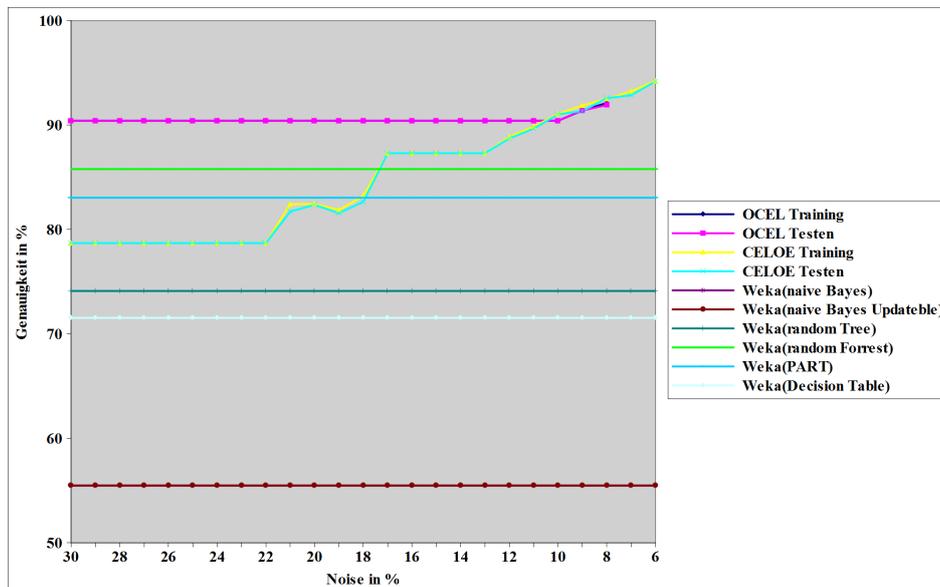


Abbildung 71: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die zehnten Klasse der Wissensbasis Cardiotoctography. Es ist zu sehen, dass CELOE die höchste Genauigkeit erzielt, gefolgt von OCEL. Daran schießen sich an die Algorithmen Random Forrest, PART, Random Tree und Decision Table. Die schlechtesten Ergebnisse erreichten die Algorithmen Naive Bayes und Naive Bayes Updateable. Des Weiteren ist zu sehen, dass OCEL bei 7% Noise keine Ergebnisse mehr produziert und CELOE erst bei 5%.

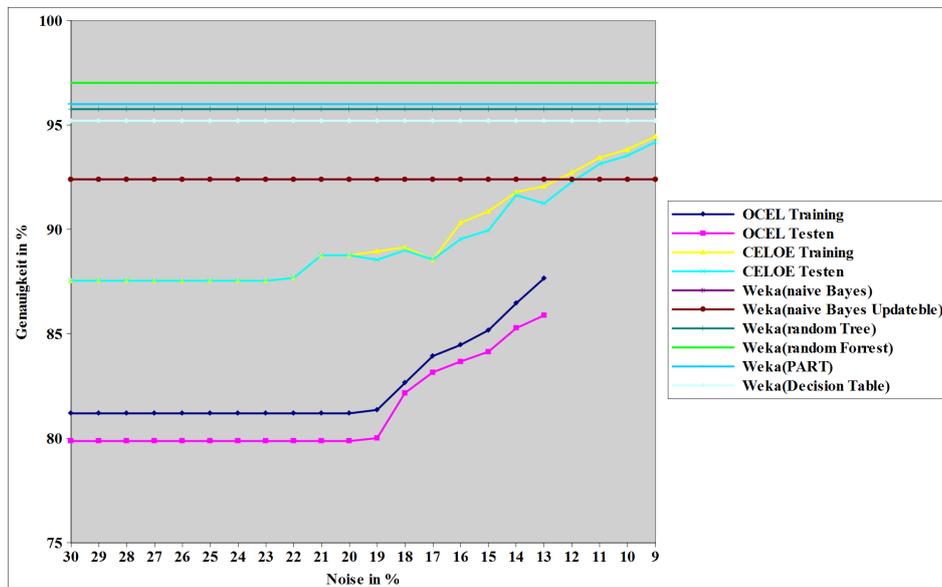


Abbildung 72: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Klasse Normal der Wissensbasis Cardiotocography. Es ist zu sehen, dass Random Forrest die höchste Genauigkeit erreicht. Daran schließen sich die Algorithmen PART, Random Forrest, Decision Table, CELOE, Naive Bayes und Naive Bayes Updateable an. Das schlechteste Ergebnis erreichte OCEL. Des Weiteren ist zu sehen, dass OCEL bei 12% Noise keine Ergebnisse mehr produziert und CELOE erst bei 8%.

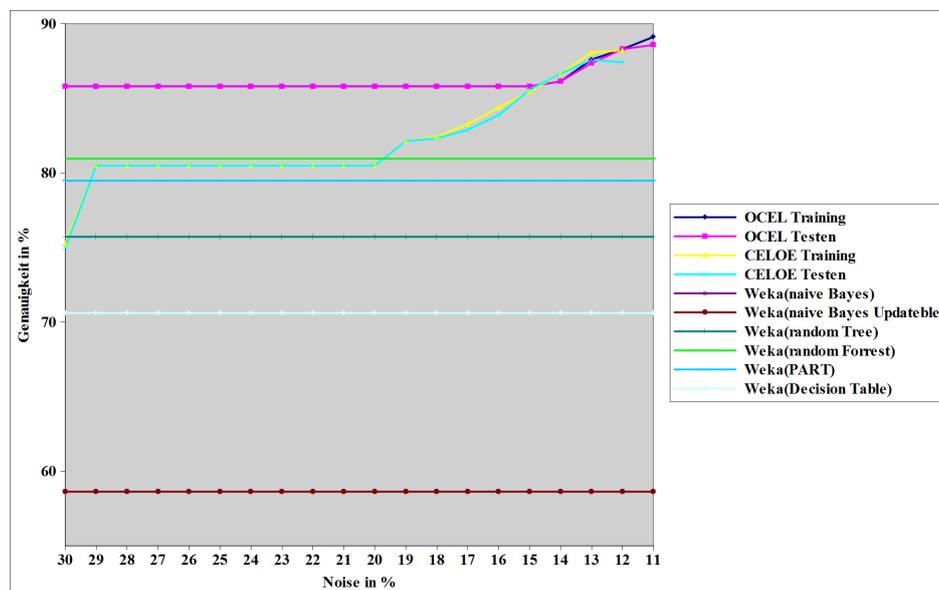


Abbildung 73: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Klasse Suspect der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL die höchste Genauigkeit erzielt, gefolgt von CELOE. Daran schließen sich an die Algorithmen Random Forrest, PART, Random Tree und Decision Table. Die schlechtesten Ergebnisse erreichten die Algorithmen Naive Bayes und Naive Bayes Updateable. Des Weiteren ist zu sehen, dass CELOE bei 11% Noise keine Ergebnisse mehr produziert und OCEL erst bei 10%.

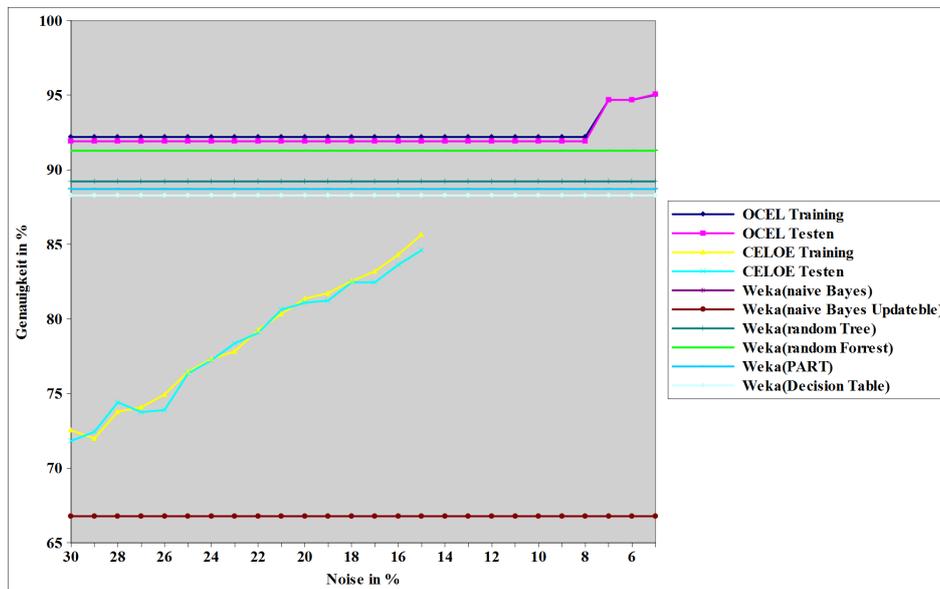


Abbildung 74: Die hier dargestellte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Noise Parameter für die Klasse Pathological der Wissensbasis Cardiotoecography. Es ist zu sehen, dass OCEL die höchste Genauigkeit erzielt. Daran schießen sich die Algorithmen Random Forrest, Random Tree, PART, Decision Table und CELOE an. Die schlechtesten Ergebnisse erreichten die Algorithmen Naive Bayes und Naive Bayes Updateable. Des Weiteren ist zu sehen, dass CELOE bei 14% Noise keine Ergebnisse mehr produziert und OCEL erst bei 4%.

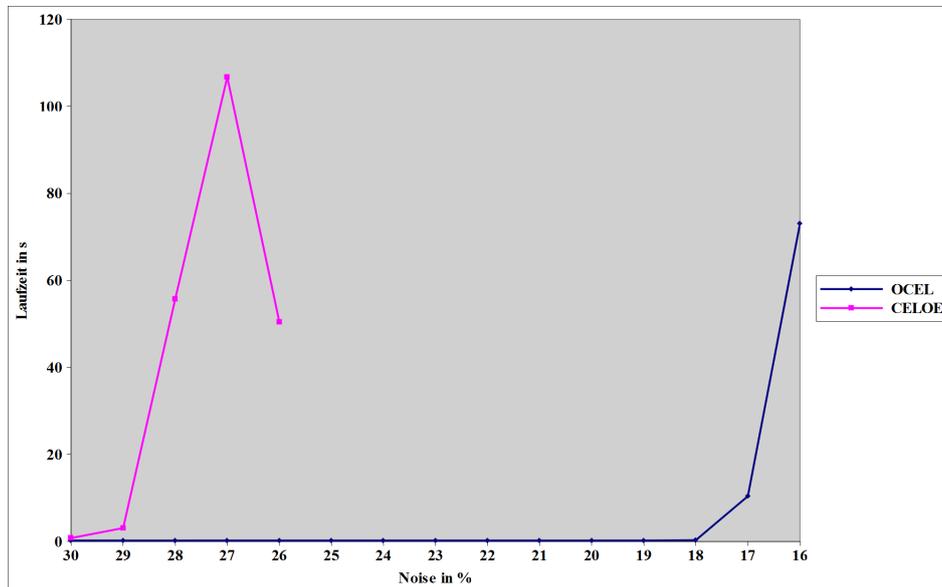
B.2 Grafiken für die benötigte Laufzeit bei Variation des Noise Parameters

Abbildung 75: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse eins der Wissensbasis Cardiotocography. Die y-Achse ist logarithmisch dargestellt. Dabei ist zu sehen, dass bei CELOE die Laufzeit bis 27% Noise stark ansteigt und bei 26% wieder abfällt. OCEL hingegen hat bis 18% Noise eine sehr geringe Laufzeit, die im Anschluss ebenfalls stark ansteigt.

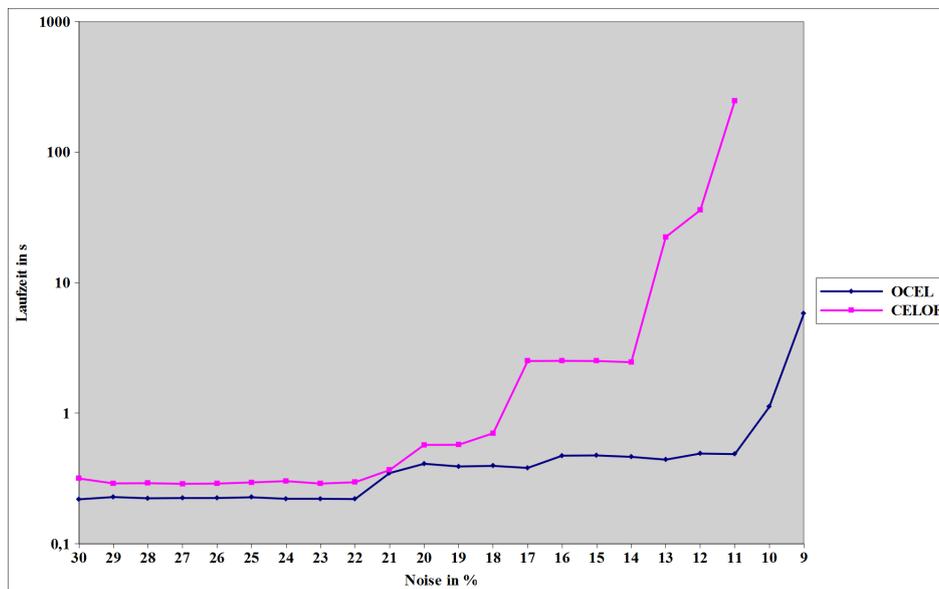


Abbildung 76: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse zwei der Wissensbasis Cardiotocography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass CELOE durchgängig eine höhere Laufzeit benötigt als OCEL. Des Weiteren ist zu sehen, dass beide Algorithmen mit einer geringen Laufzeit anfangen und die Laufzeit bei starker Verringerung des Noise-Parameters zunehmen. Bei CELOE geschieht dies nur schneller als bei OCEL.

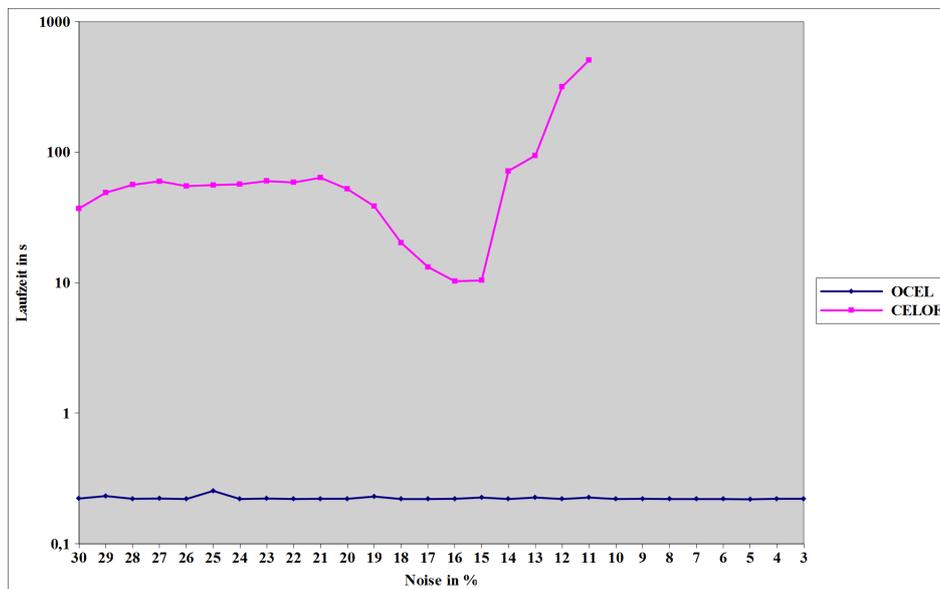


Abbildung 77: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse drei der Wissensbasis Cardiotocography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass CELOE durchgängig eine höhere Laufzeit benötigt als OCEL. Des Weiteren ist zu sehen, dass CELOE schon zu Beginn eine sehr hohe Laufzeit besitzt. Diese sinkt von 21% bis 15% Noise wieder und steigt anschließend sehr stark an. Die Laufzeit von OCEL bleibt bis 3% Noise unter einer Sekunde.

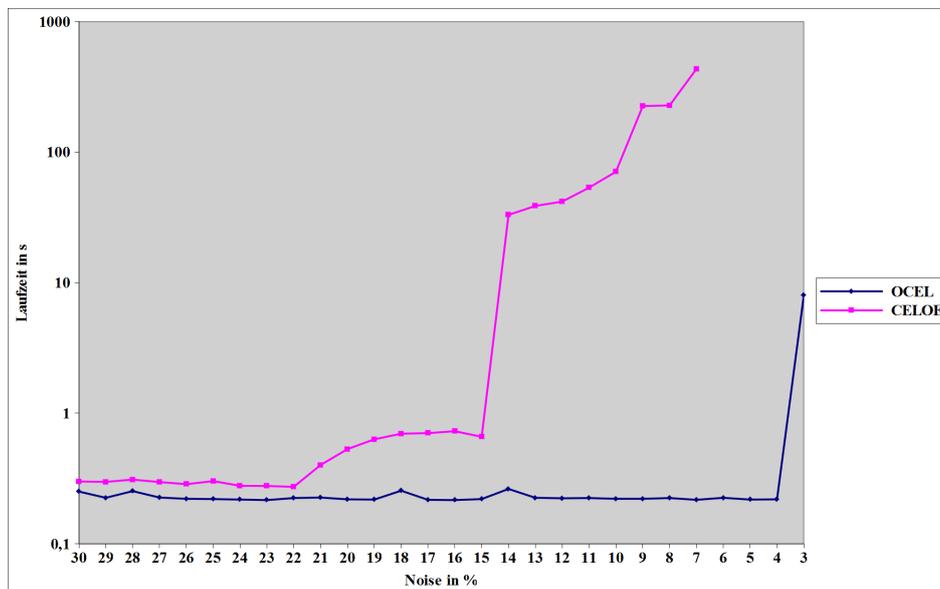


Abbildung 78: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse vier der Wissensbasis Cardiotocography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass CELOE durchgängig eine höhere Laufzeit benötigt als OCEL. Beide Algorithmen beginnen mit einer geringen Laufzeit und. Wenn der Noise-Parameter stark verringert wird steigen die Laufzeiten stark an. Nur bei CELOE geschieht dies schneller und im höheren Maße als bei OCEL.

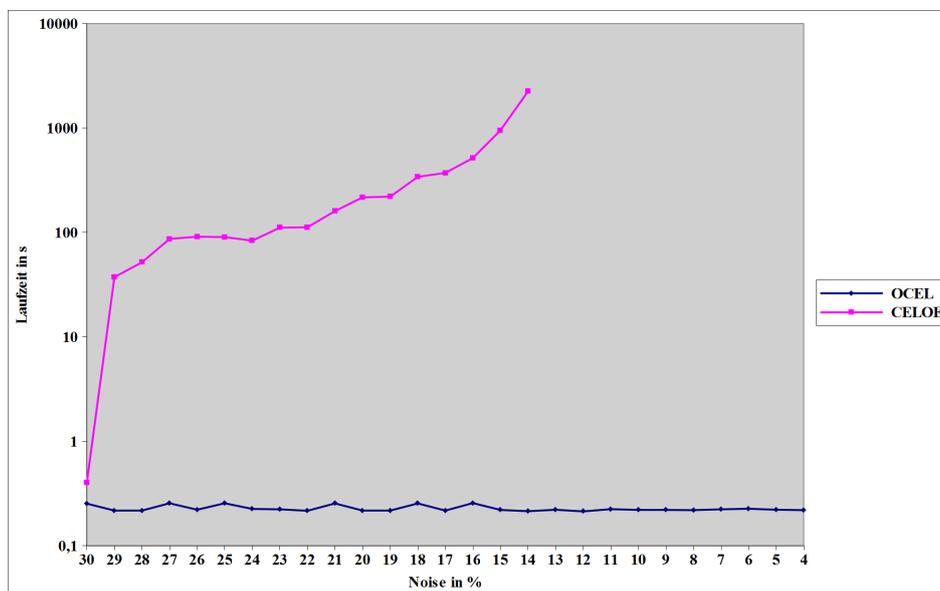


Abbildung 79: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse fünf der Wissensbasis Cardiotocography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass CELOE durchgängig eine höhere Laufzeit benötigt als OCEL. Dabei steigt die Laufzeit von CELOE schon bei geringer Verringerung des Laufzeit-Parameters sehr stark an. Die Laufzeit von OCEL bleibt immer unter einer Sekunde.

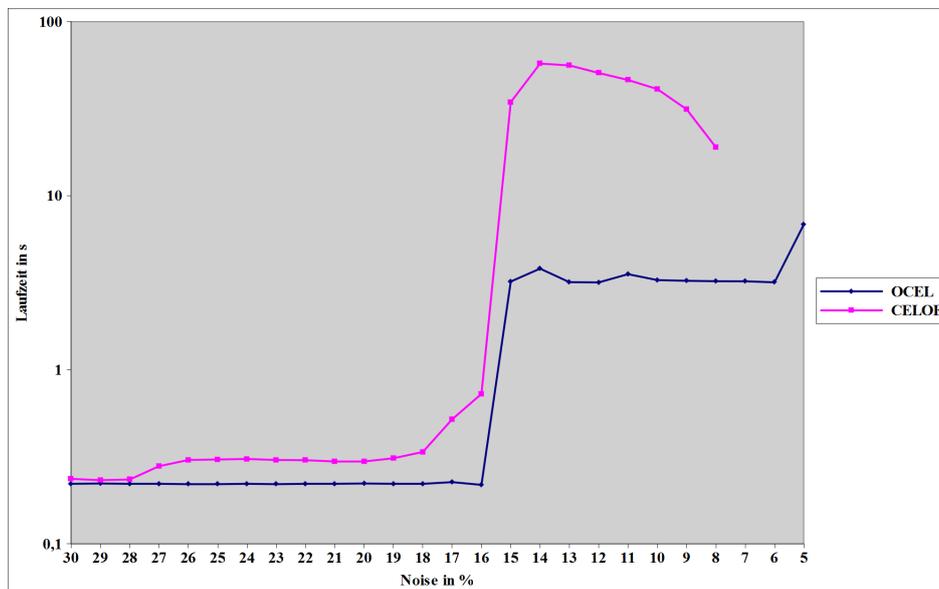


Abbildung 80: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse sechs der Wissensbasis Cardiotoctography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass CELOE durchgängig eine höhere Laufzeit benötigt als OCEL. Beide Algorithmen beginnen mit einer geringen Laufzeit. Wenn der Noise-Parameter stark verringert wird steigen die Laufzeiten stark an. Nur bei CELOE geschieht dies schneller und im höheren Maße als bei OCEL.

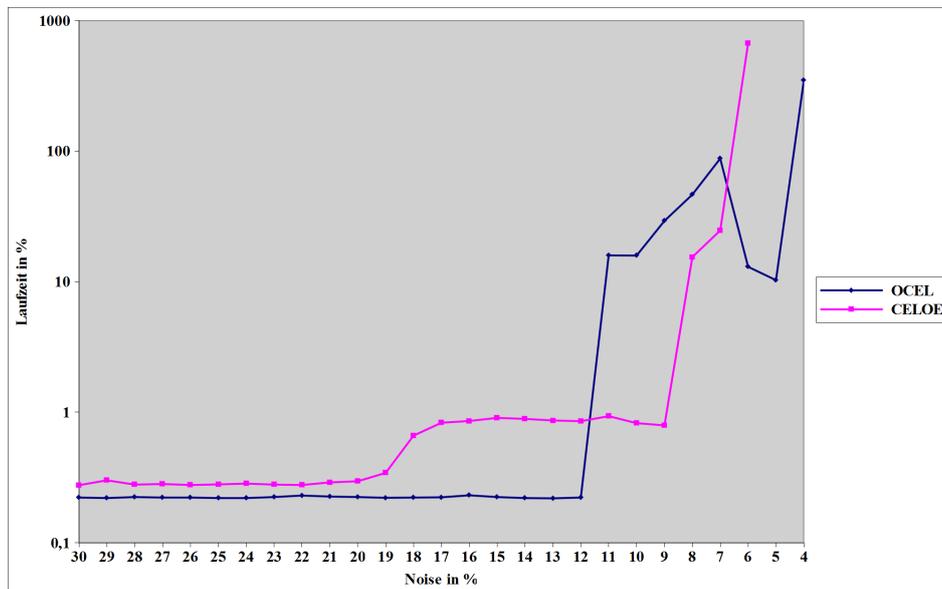


Abbildung 81: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse sieben der Wissensbasis Cardiotocography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass CELOE teilweise höhere und teilweise niedrigere Laufzeiten als OCEL besitzt. Beide Algorithmen beginnen mit einer geringen Laufzeit und bei 12% Noise steigt die Laufzeit bei OCEL stärker an als die von CELOE. Bei 7% Noise kehrt sich dies wieder um. Beide Algorithmen haben am Ende eine sehr hohe Laufzeit.

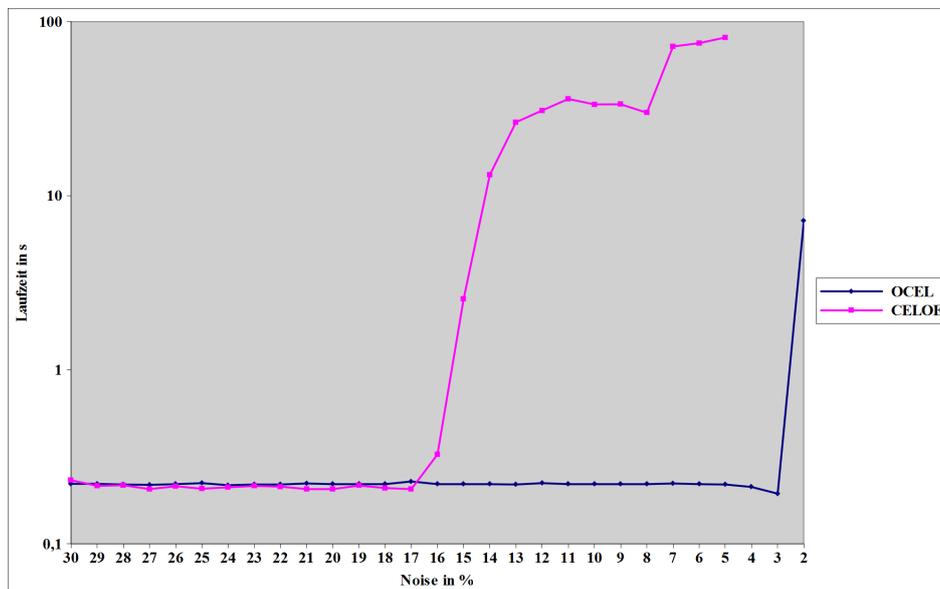


Abbildung 82: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse acht der Wissensbasis Cardiotocography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass beide Algorithmen beginnen mit einer geringen Laufzeit beginnen, wobei CELOE teilweise eine geringere Laufzeit aufweist. Wenn der Noise-Parameter stark verringert wird steigen die Laufzeiten stark an. Nur bei CELOE geschieht dies schneller und im höheren Maße als bei OCEL.

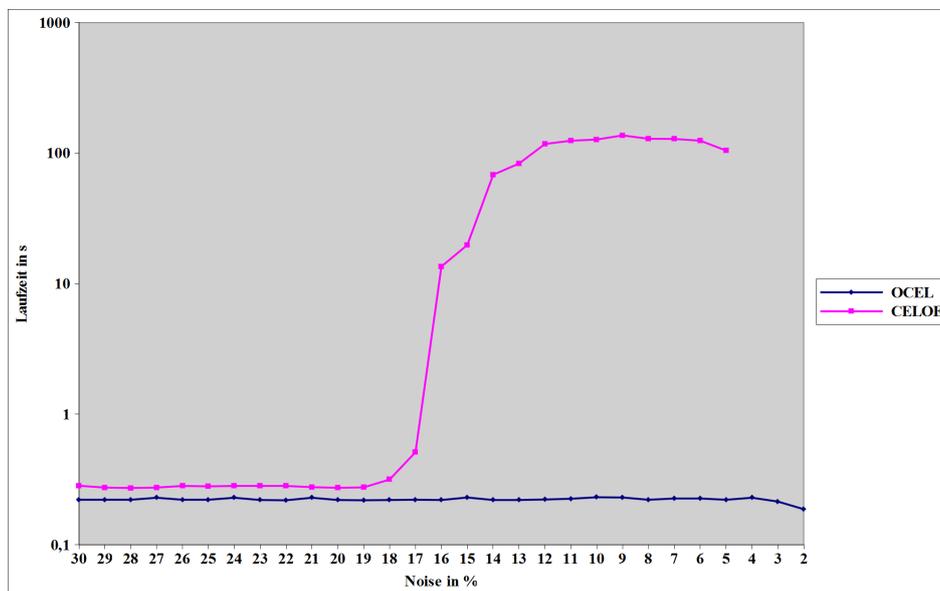


Abbildung 83: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse sechs der Wissensbasis Cardiocography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass CELOE durchgängig eine höhere Laufzeit benötigt als OCEL. Beide Algorithmen beginnen mit einer geringen Laufzeit. Wenn der Noise-Parameter stark verringert wird steigen die Laufzeiten von CELOE stark an. Nur die Laufzeit von OCEL bleibt konstant.

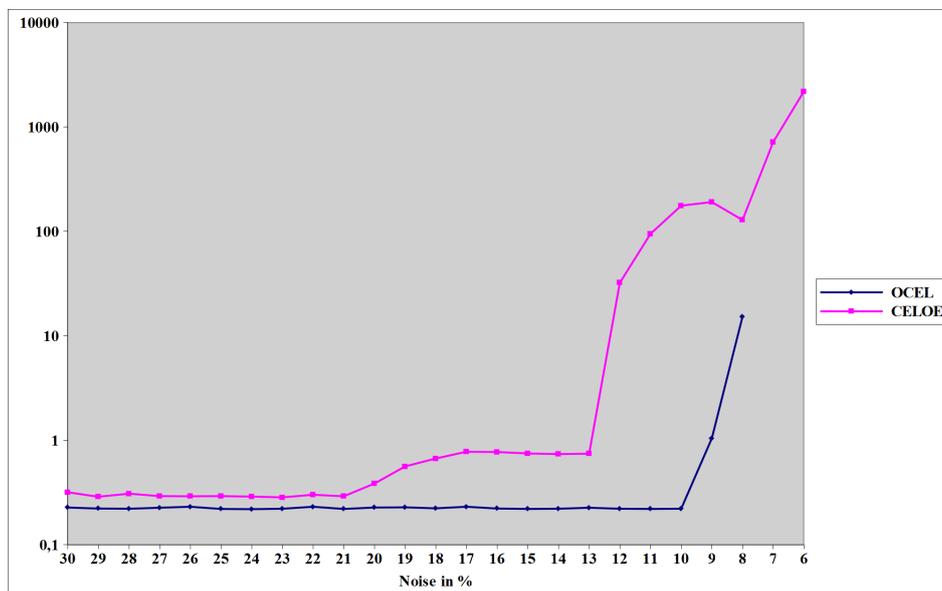


Abbildung 84: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse sechs der Wissensbasis Cardiography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass CELOE durchgängig eine höhere Laufzeit benötigt als OCEL. Beide Algorithmen beginnen mit einer geringen Laufzeit. Wenn der Noise-Parameter stark verringert wird steigen die Laufzeiten von CELOE stark an. Nur die Laufzeit von OCEL bleibt konstant.

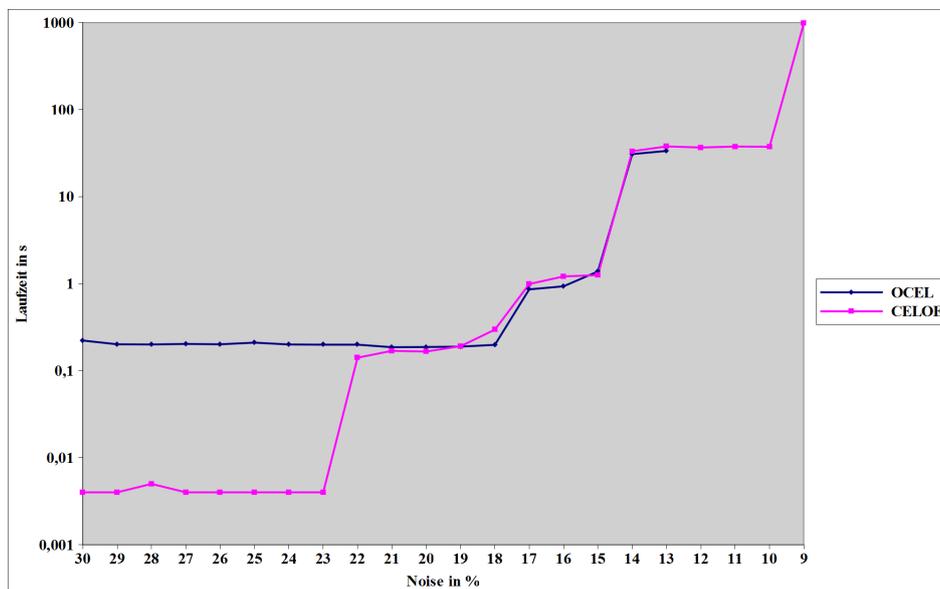


Abbildung 85: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse Normal der Wissensbasis Cardiotocography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass OCEL bis 23% Noise eine höher Laufzeit besitzt als CELOE. Im Anschluss daran ist die Algorithmuslaufzeit für beide Algorithmen gleich. Am ende hatte CELOE eine höhere Laufzeit, da es noch bis 9% Noise Ergebnisse produzierte

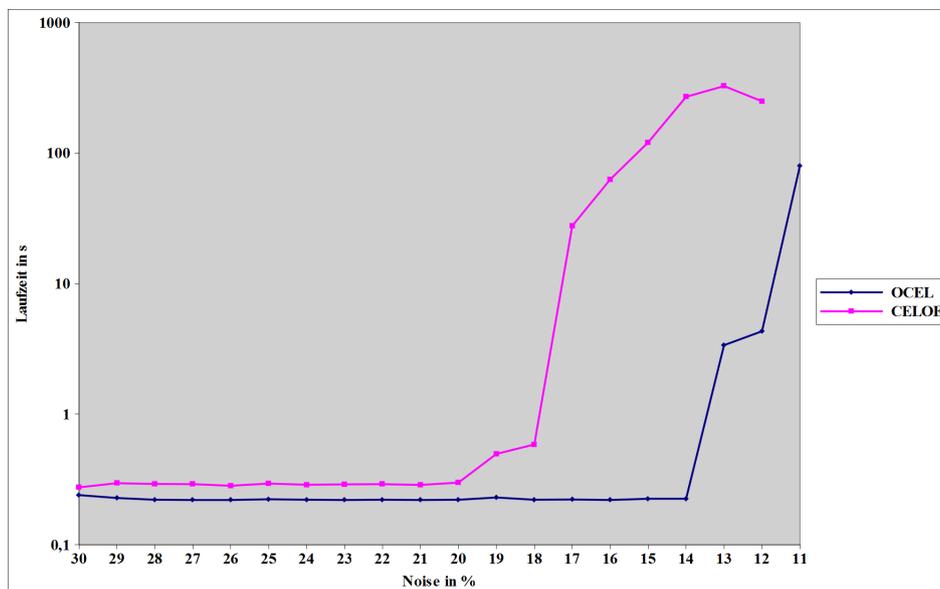


Abbildung 86: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse Suspect der Wissensbasis Cardiography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass CELOE teilweise höhere und teilweise niedrigere Laufzeiten als OCEL besitzt. Die Laufzeit von CELOE ist durchgängig höher als die von OCEL. Jedoch erreichen beide Algorithmen am Ende eine hohe Laufzeit.

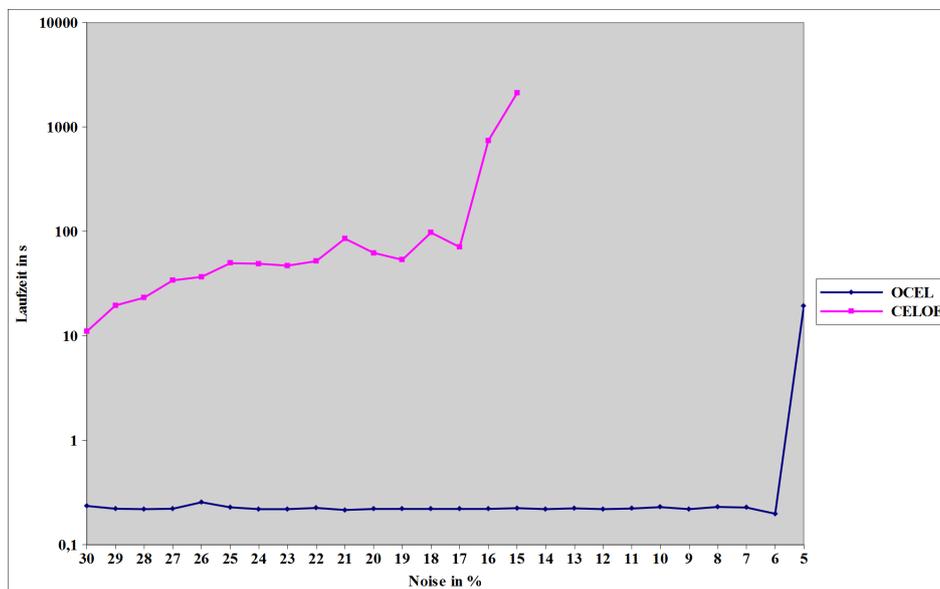


Abbildung 87: Die hier dargestellte Grafik zeigt die Laufzeiten in Abhängigkeit des Noise Parameters für die Klasse Pathological der Wissensbasis Cardiotocography. Dabei ist die y-Achse in logarithmisch dargestellt. Es ist zu sehen, dass CELOE eine durchgängig höher Laufzeit ausweist als OCEL. Wenn der Noise-Parameter stark verringert wird steigen die Laufzeiten stark an. Nur bei CELOE geschieht dies schneller und im höheren Maße als bei OCEL.

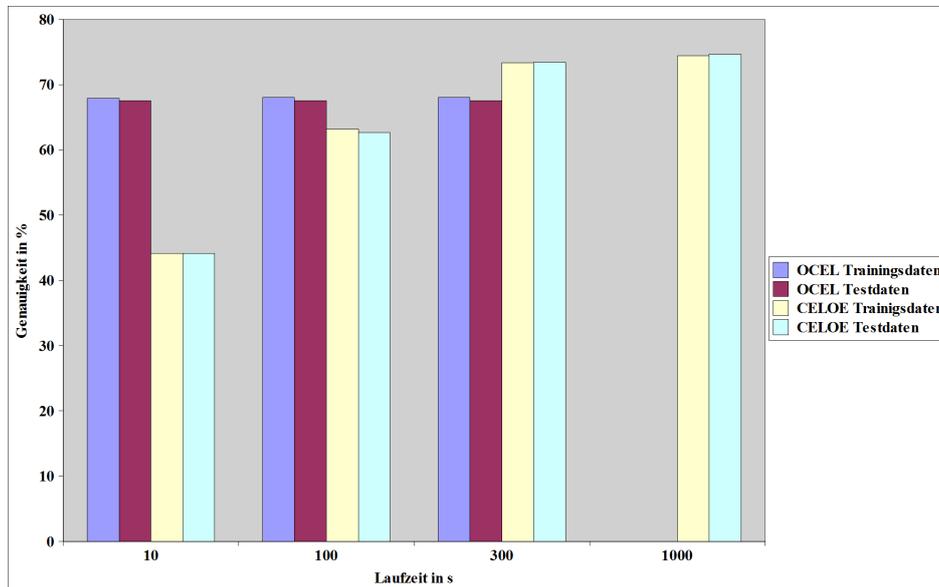
B.3 Grafiken für die fest eingestellte Laufzeit von 10, 100, 300 und 1000 Sekunden

Abbildung 88: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse eins der Wissensbasis Cardiotocography. Es ist zu sehen, dass es keine Unterschiede in der Genauigkeit bei OCEL gibt, egal welche Laufzeit eingestellt wurde. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryException* abgebrochen wurde. Bei CELOE sieht man, dass mit jeder Erhöhung der Laufzeit die Genauigkeit zunimmt.

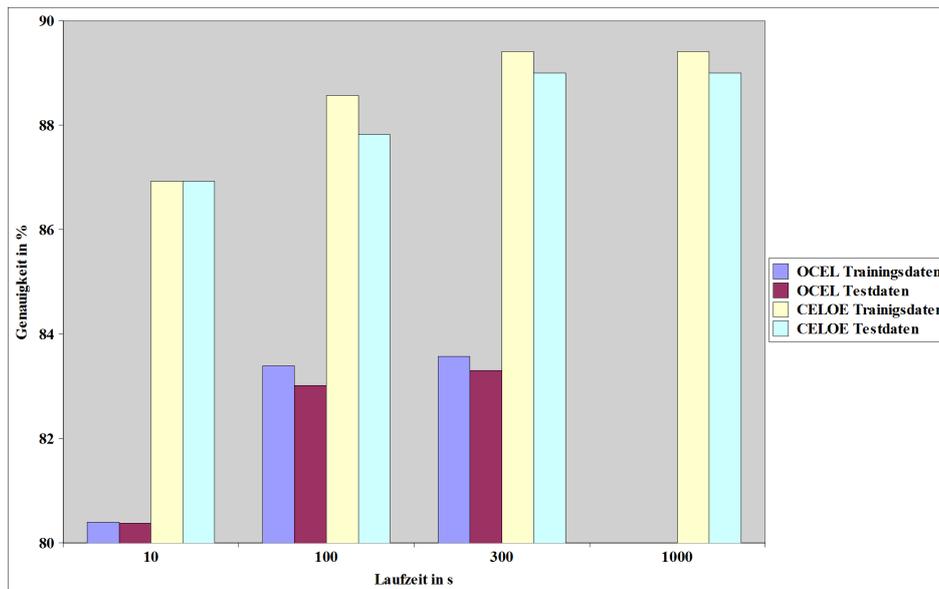


Abbildung 89: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse zwei der Wissensbasis CardiotoGRAPHY. Es ist zu sehen, dass OCEL bei der Erhöhung der Laufzeit von 10 Sekunden auf 100 Sekunden, eine Erhöhung der Genauigkeit stattfindet. Bei der Erhöhung der Laufzeit auf 300 Sekunden jedoch nicht. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryexception* abgebrochen wurde. Bei CELOE sieht man, dass mit der Erhöhung der Laufzeit auf 100 und 300 Sekunden, sich die Genauigkeit ebenfalls erhöht. Nur bei 1000 Sekunden gibt es keine Erhöhung der Genauigkeit mehr.

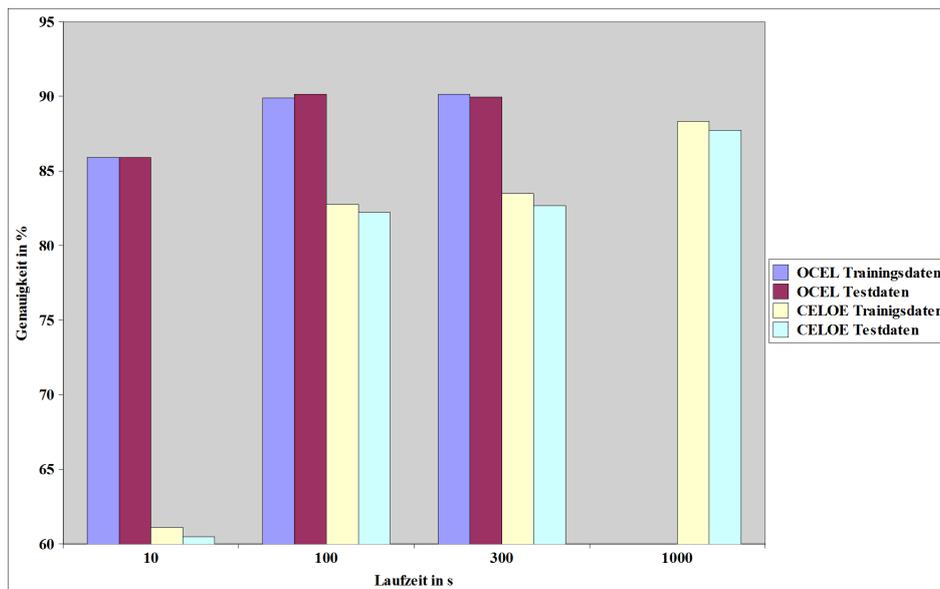


Abbildung 90: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse drei der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL bei der Erhöhung der Laufzeit von 10 Sekunden auf 100 Sekunden, eine Erhöhung der Genauigkeit stattfindet. Bei der Erhöhung der Laufzeit auf 300 Sekunden sinkt die Genauigkeit wieder etwas. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryexception* abgebrochen wurde. Bei CELOE sieht man, dass mit der Erhöhung der Laufzeit auf 100, sich die Genauigkeit ebenfalls erhöht. Bei 300 Sekunden Laufzeit findet keine Erhöhung der Genauigkeit statt, aber bei 1000 Sekunden wird die Genauigkeit wieder erhöht.

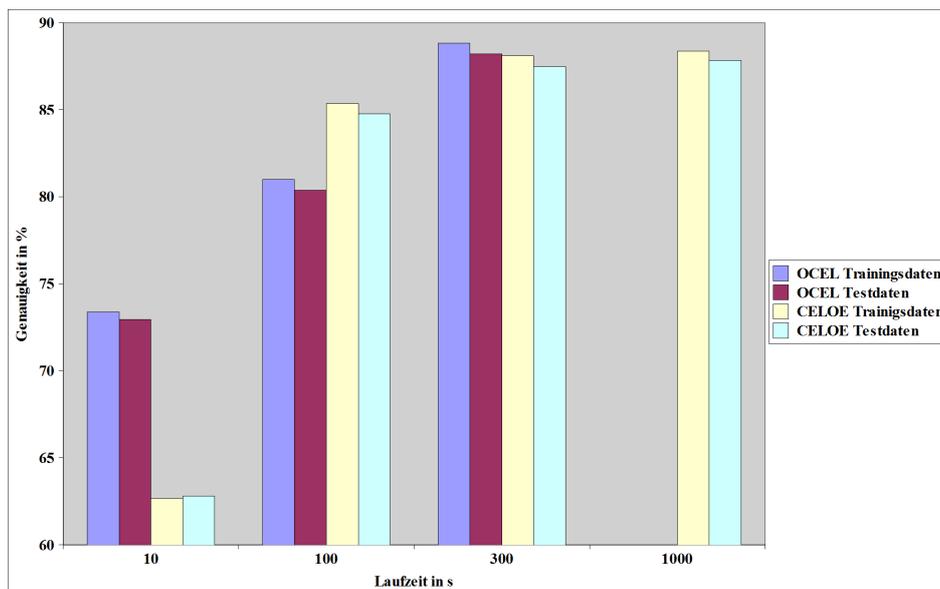


Abbildung 91: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse vier der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL bei jeder Erhöhung der Laufzeit auch die Genauigkeit erhöht wird. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryexception* abgebrochen wurde. Bei CELOE sieht man, dass sich mit der Erhöhung der Laufzeit bis auf 300 Sekunden, jedes mal die Genauigkeit erhöht. Nur bei der Erhöhung der Laufzeit auf 1000 Sekunden bleibt die Genauigkeit gleich.

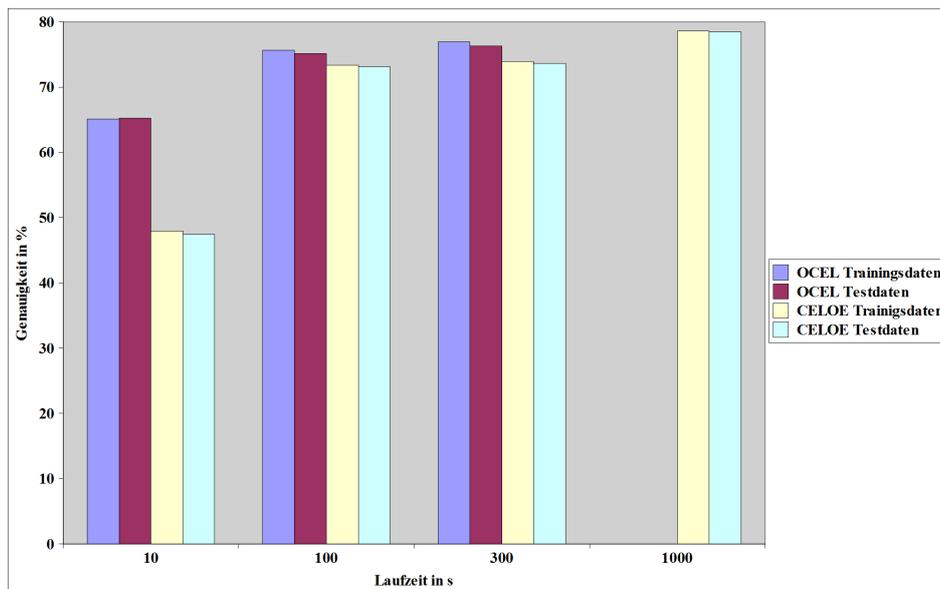


Abbildung 92: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse fünf der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL bei der Erhöhung der Laufzeit von 10 Sekunden auf 100 Sekunden, eine Erhöhung der Genauigkeit stattfindet. Bei der Erhöhung der Laufzeit auf 300 Sekunden bleibt die Genauigkeit gleich. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryexception* abgebrochen wurde. Bei CELOE sieht man, dass mit der Erhöhung der Laufzeit auf 100, sich die Genauigkeit ebenfalls erhöht. Bei 300 Sekunden Laufzeit findet keine Erhöhung der Genauigkeit statt, aber bei 1000 Sekunden wird die Genauigkeit wieder erhöht.

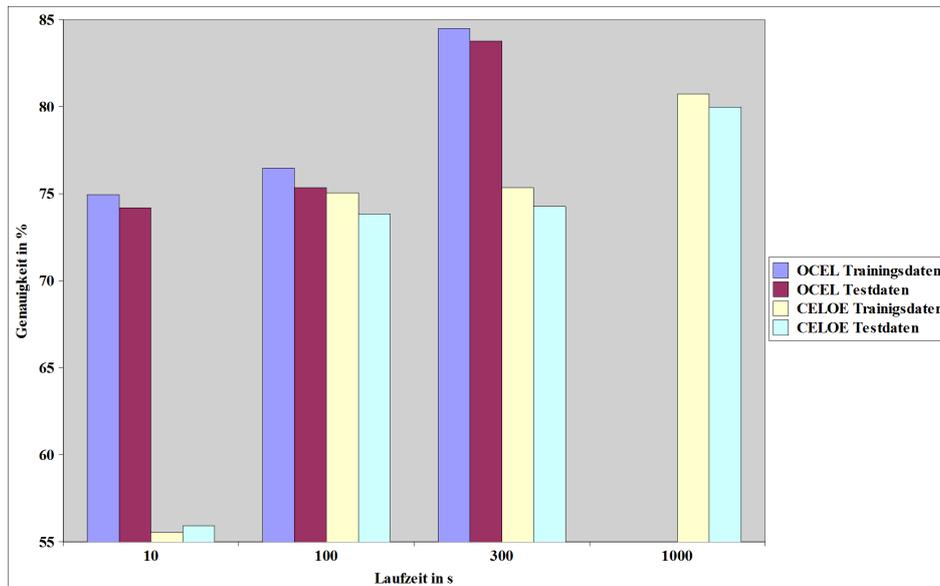


Abbildung 93: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse sechs der Wissensbasis Cardiotoxicography. Es ist zu sehen, dass OCEL bei jeder Erhöhung der Laufzeit eine Erhöhung der Genauigkeit stattfindet. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryException* abgebrochen wurde. Bei CELOE sieht man, dass mit der Erhöhung der Laufzeit auf 100, sich die Genauigkeit ebenfalls erhöht. Bei 300 Sekunden Laufzeit findet keine Erhöhung der Genauigkeit statt, aber bei 1000 Sekunden wird die Genauigkeit wieder erhöht.

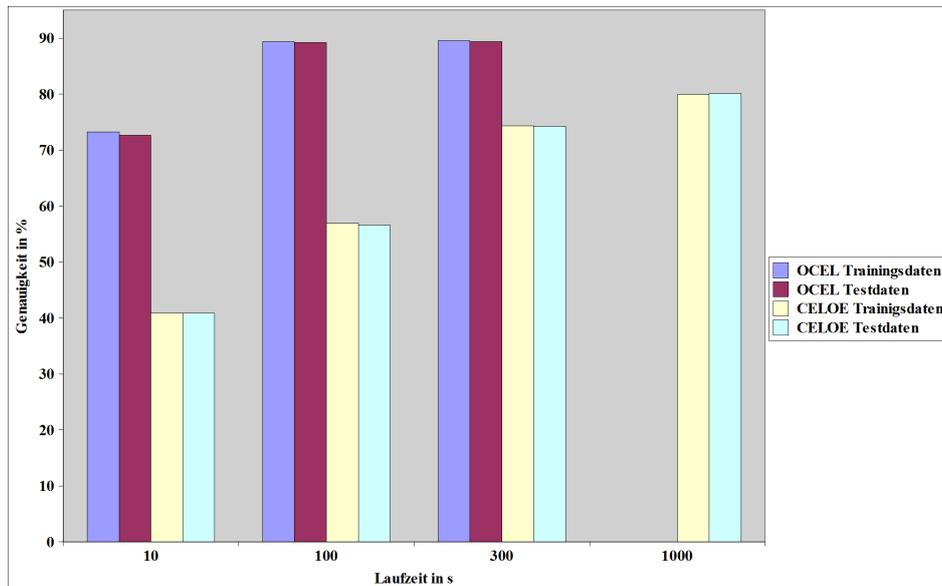


Abbildung 94: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse sieben der Wissensbasis Cardiotoctography. Es ist zu sehen, dass OCEL bei der Erhöhung der Laufzeit von 10 Sekunden auf 100 Sekunden, eine Erhöhung der Genauigkeit stattfindet. Bei der Erhöhung der Laufzeit auf 300 Sekunden bleibt die Genauigkeit gleich. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryexception* abgebrochen wurde. Bei CELOE sieht man, dass mit jeder Erhöhung der Laufzeit auch die Genauigkeit erhöht wird.

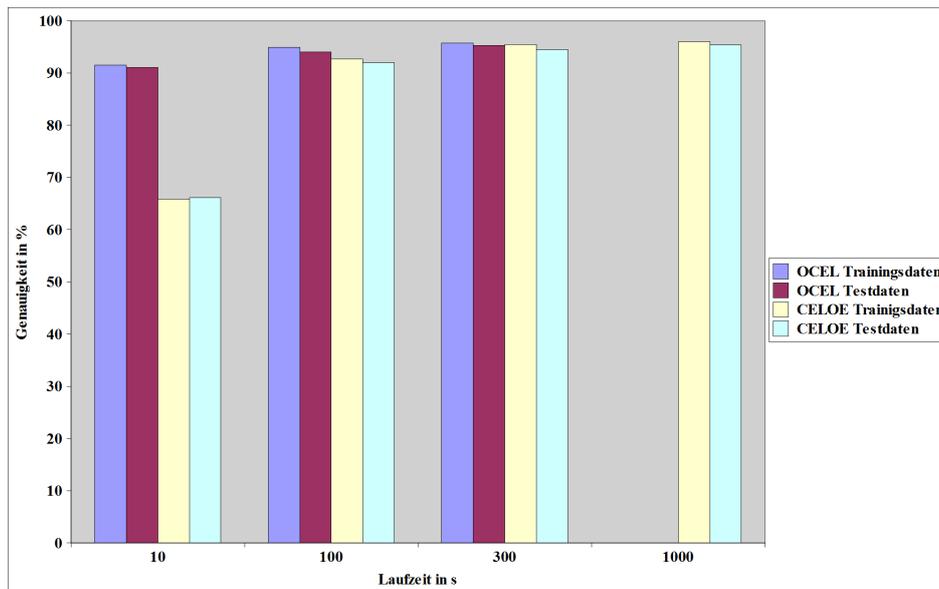


Abbildung 95: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse sieben der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL bei der Erhöhung der Laufzeit von 10 Sekunden auf 100 Sekunden, eine Erhöhung der Genauigkeit stattfindet. Bei der Erhöhung der Laufzeit auf 300 Sekunden bleibt die Genauigkeit gleich. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryexception* abgebrochen wurde. Bei CELOE sieht man, dass sich bis zur Erhöhung der Laufzeit auf 300 Sekunden die Genauigkeit ebenfalls erhöht. Nur bei einer Erhöhung auf 1000 Sekunden bleibt die Genauigkeit gleich.

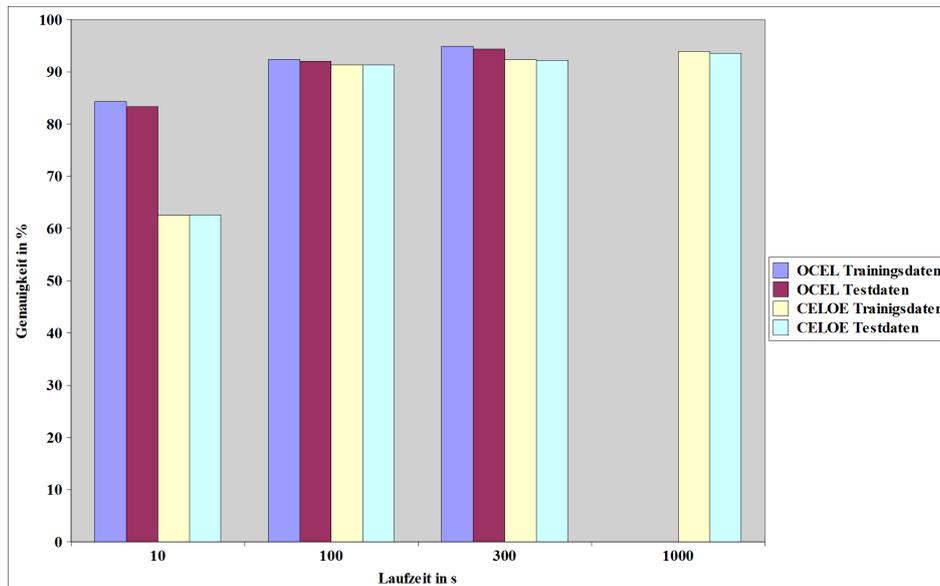


Abbildung 96: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse fünf der Wissensbasis Cardiotocography. Es ist zu sehen, dass OCEL bei jeder Erhöhung der Laufzeit eine Erhöhung der Genauigkeit stattfindet. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryexception* abgebrochen wurde. Bei CELOE sieht man, dass mit der Erhöhung der Laufzeit auf 100, sich die Genauigkeit ebenfalls erhöht. Bei 300 Sekunden Laufzeit findet keine Erhöhung der Genauigkeit statt, aber bei 1000 Sekunden wird die Genauigkeit wieder erhöht.

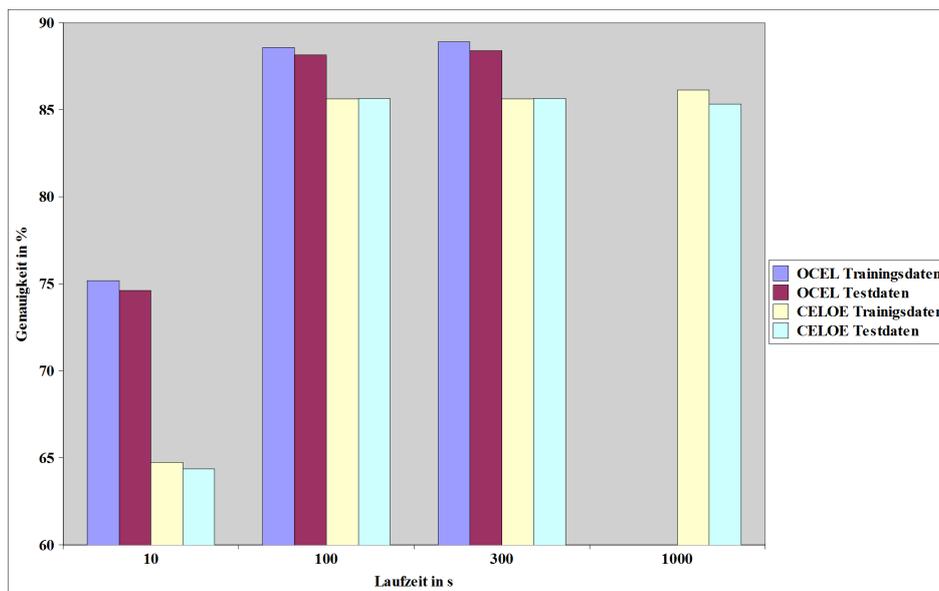


Abbildung 97: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse fünf der Wissensbasis Cardiotoagraphy. Es ist zu sehen, dass für beide Algorithmen bei der Erhöhung der Laufzeit von 10 Sekunden auf 100 Sekunden, eine Erhöhung der Genauigkeit stattfindet. Bei der Erhöhung der Laufzeit auf 300 Sekunden bleibt die Genauigkeit gleich. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryexception* abgebrochen wurde. Bei CELOE bewirkt das Erhöhen der Laufzeit auf 1000 Sekunden eine Steigerung der Trainingsgenauigkeit aber eine Verschlechterung der Testgenauigkeit.

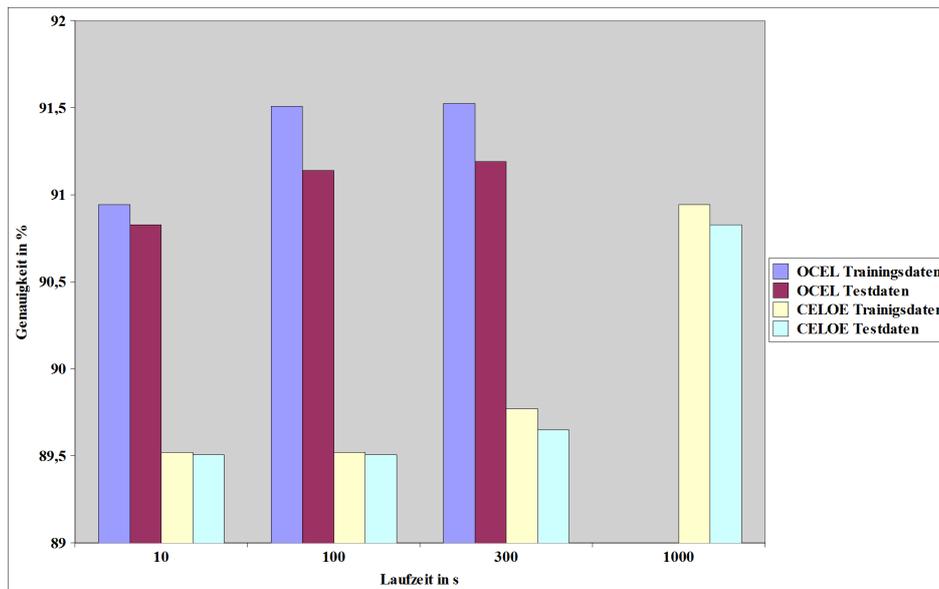


Abbildung 98: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse Normal der Wissensbasis Cardiotocography. Es ist zu sehen, dass für OCEL, bei der Erhöhung der Laufzeit von 10 Sekunden auf 100 Sekunden, eine Erhöhung der Genauigkeit stattfindet. Bei der Erhöhung der Laufzeit auf 300 Sekunden bleibt die Genauigkeit gleich. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryexception* abgebrochen wurde. Bei CELOE bewirkt das Erhöhen der Laufzeit auf 100 Sekunden gar nichts. Die Genauigkeit bleibt gleich. Erst bei der Erhöhung der Laufzeit auf 300 und 1000 Sekunden erhöht sich auch die Genauigkeit des Algorithmus.

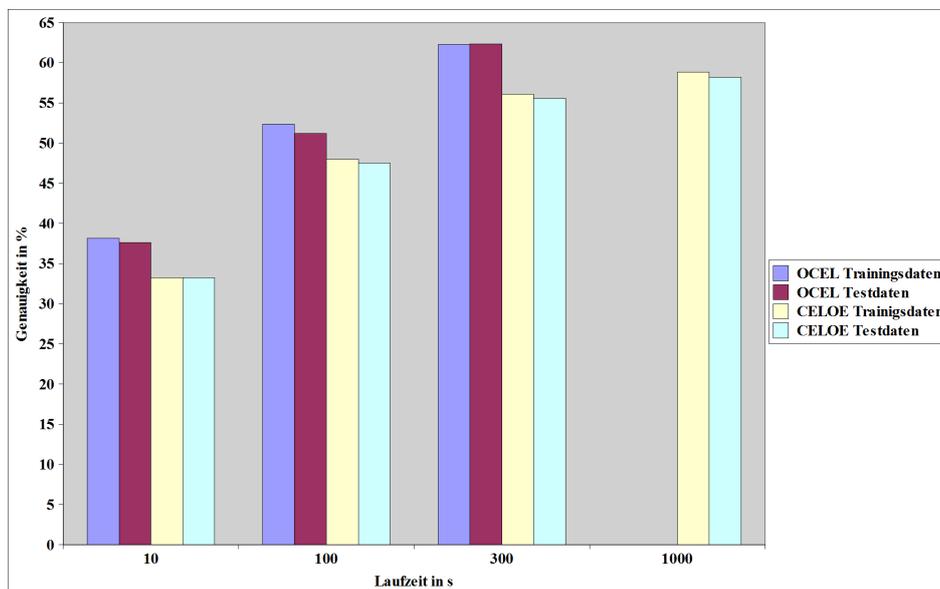


Abbildung 99: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse Suspect der Wissensbasis Cardiotocography. Es ist zu sehen, dass für beide Algorithmen bei jeder Erhöhung der Laufzeit auch die Genauigkeit erhöht wird. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryException* abgebrochen wurde.

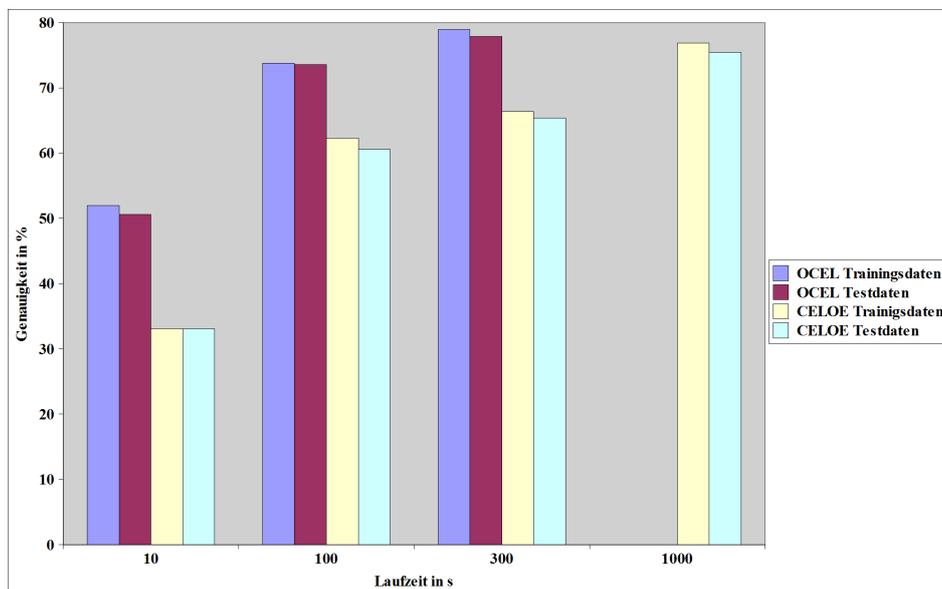


Abbildung 100: Die hier gezeigte Grafik zeigt die Genauigkeit der Lernergebnisse in Bezug auf den eingestellten Laufzeit Parameter für die Klasse Pathological der Wissensbasis Cardiotocography. Es ist zu sehen, dass für beide Algorithmen bei jeder Erhöhung der Laufzeit auch die Genauigkeit erhöht wird. Das Fehlen der Säulen bei 1000 Sekunden Laufzeit für OCEL liegt daran, dass der Algorithmus mit einer *OutOfMemoryexception* abgebrochen wurde.

Literaturverzeichnis

Literatur

- [BK94] Ivan Bratko and Ross D. King. Applications of inductive logic programming. *SIGART Bulletin*, 5(1):43–49, 1994.
- [Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, oct 2001.
- [BW04] M. Bauer and K. Weingard. Konventionelle und mr-mammographie. *Der Gynäkologe*, 37:1121–1134, 2004. 10.1007/s00129-004-1624-6.
- [DLdCD⁺91] Asim Kumar Debnath, Rosa L. Lopez de Compadre, Gargi Debnath, Alan J. Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry*, 34(2):786–797, 1991.
- [FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting,. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997.
- [FW98] Eibe Frank and Ian H. Witten. Generating accurate rule sets without global optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98*, pages 144–151, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [HKRS08] Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, and York Sure. *Semantic Web Grundlagen*. Springer Verlag, 2008.
- [JL95] George John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann, 1995.
- [Jos96] J. Jossinet. Variability of impedivity in normal and pathological breast tissue. *Medical and Biological Engineering and Computing*, 34:346–350, 1996. 10.1007/BF02520002.
- [KDM04] J. Karmrodt, M. David, and K. Markstaller. Elektrische impedanztomographie (eit). *Notfall + Rettungsmedizin*, 7:572–576, 2004. 10.1007/s10049-004-0703-5.
- [Klo03] Th. Klockgether. Medikamentöse behandlung der idiopathischen parkinson-krankheit. *Der Nervenarzt*, 74:s12–s21, 2003. 10.1007/s00115-003-1483-8.

- [Koh95] Ron Kohavi. The power of decision tables. In Nada Lavrac and Stefan Wrobel, editors, *Machine Learning: ECML-95*, volume 912 of *Lecture Notes in Computer Science*, pages 174–189. Springer Berlin / Heidelberg, 1995.
- [Kra07] R. Kramme. Elektrokardiographen (ekg-geräte). In Rüdiger Kramme, editor, *Medizintechnik*, pages 87–96. Springer Berlin Heidelberg, 2007.
- [Leh09] Jens Lehmann. DL-learner: Learning concepts in description logics. volume 10, pages 2639–2642, 2009.
- [Leh10] Jens Lehmann. *Learning OWL Class Expressions*. PhD thesis, University of Leipzig, 2010. PhD in Computer Science.
- [LH10] Jens Lehmann and Pascal Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 78:203–250, 2010.
- [LMR⁺07] M.A Little, P. E. McSharry, S.J. Roberts, D.A.E. Costello, and I.M. Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine*, 6(23), 2007.
- [MBL08] Dinora A. Morales, Endika Bengoetxea, and Pedro Larra naga. Selection of human embryos for transfer by bayesian classifiers. *Comput. Biol. Med.*, 38(11-12):1177–1186, 2008.
- [Mot99] H.J. Motulsky. *Analyzing Data with GraphPad Prism*. GraphPad Software Inc., 1999.
- [PSS⁺06] K. Possinger, P. Schmid, H. J. Schmoll, K. Höffken, R. Kreienberg, and J. Dunst. Mammakarzinom der frau. In Hans-Joachim Schmoll, Klaus Höffken, and Kurt Possinger, editors, *Kompendium Internistische Onkologie*, pages 4215–4331. Springer Berlin Heidelberg, 2006.
- [QAY⁺02] Yinsheng Qu, Bao-Ling Adam, Yutaka Yasui, Michael D. Ward, Lisa H. Cazares, Paul F. Schellhammer, Ziding Feng, O. John Semmes, and Jr. Wright, George L. Boosted Decision Tree Analysis of Surface-enhanced Laser Desorption/Ionization Mass Spectral Serum Profiles Discriminates Prostate Cancer from Noncancer Patients. *Clin Chem*, 48(10):1835–1843, 2002.
- [RKJH04] P.W. Rieck, J. Kriegsch, C. Jaeckel, and C. Hartmann. Effekt von suramin auf proliferation und migration von linsenepithelzellen in vitro. *Der Ophthalmologe*, 101:73–79, 2004. 10.1007/s00347-003-0908-x.

- [SMS⁺08] A. Strzelczyk, J.C. Möller, M. Stamelou, A. Matusch, and W.H. Oertel. Atypische parkinson-syndrome. *Der Nervenarzt*, 79:1203–1222, 2008. 10.1007/s00115-008-2559-2.
- [SR03] U. Sommer and H. Reichmann. Früherkennung des parkinson-syndroms. In *Der Orthopäde*, pages 82–84. Springer Verlag, 2003. 10.1007/s00132-002-0385-y.
- [Sri04] A. Srinivasan. *The Aleph Manual*, 2004. <http://www.comlab.ox.ac.uk/activities/machinelearning/Aleph/>.
- [TASM08] Kazuhisa Tsunoyama, Ata Amini, Michael J. E. Sternberg, and Stephen H. Muggleton. Scaffold hopping in drug discovery using inductive logic programming. *Journal of Chemical Information and Modeling*, 48(5):949–957, 2008. PMID: 18457387.
- [WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, San Francisco, CA, 2nd edition, 2005.

Abbildungsverzeichnis

1	Layer Cake des Semantik Web	20
2	Einfacher RDF Graph	20
3	„Generate and Test“ Ansatz des DL-Learners	34
4	Darstellung der Architektur des DL-Learners.	36
5	Ablaufbeschreibung der Konvertierung der Mutagenesis Wissensbasis	41
6	Ergebnisse der Lernprozesse für Mutagenesis 1 anhand des Noise Parameters.	42
7	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für den ersten Teil der Mutagenesis Wissensbasis.	45
8	Ergebnisse der Lernprozesse für Mutagenesis 1 anhand des Laufzeit Parameters.	46
9	Ergebnisse der Lernprozesse für Mutagenesis 2 anhand des Noise Parameters	47
10	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für den zweiten Teil der Mutagenesis Wissensbasis.	50
11	Ergebnisse der Lernprozesse für Mutagenesis 2 anhand des Laufzeit Parameters	51
12	Darstellung der Strukturformel von Suramin	52
13	Ablaufbeschreibung der Konvertierung der Suramin Wissensbasis	54
14	Ergebnisse der Lernprozesse für Suramin anhand des Noise Parameters.	56
15	Laufzeit der Lernprozesse für Suramin in Abhängigkeit des Noise Parameters.	59
16	Ergebnisse der Lernprozesse für Suramin anhand des Laufzeit Parameters.	60
17	Ablaufbeschreibung der Konvertierung der Parkinsons Wissensbasis	63
18	Ergebnisse der Lernprozesse für Parkinson anhand des Noise Parameters.	65
19	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Parkinson Wissensbasis.	69
20	Ergebnisse der Lernprozesse für Parkinson anhand des Laufzeit Parameters.	70
21	Ablaufbeschreibung der Konvertierung der Heart Wissensbasis	73
22	Ergebnisse der Lernprozesse für Heart anhand des Noise Parameters	75
23	Laufzeit der Algorithmen OCEL und CELOE in Abhängigkeit des eingestellten Noise Parameter für die Heart Wissensbasis	79
24	Ergebnisse der Lernprozesse für Heart anhand des Laufzeit Parameters	80

25	Ablaufbeschreibung der Konvertierung der Mammographic Wissensbasis	84
26	Ergebnisse der Lernprozesse für Mammographic Masses anhand des Noise Parameters	86
27	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Mammographic Masses Wissensbasis.	90
28	Das Ergebnis des Lernprozesses in Abhängigkeit des eingestellten Laufzeitparameter	91
29	Ablaufbeschreibung der Konvertierung der Breast Tissue Wissensbasis	94
30	Ergebnisse der Lernprozesse für alle Klassen der Wissensbasis Breast Tissue anhand des Noise Parameters.	96
31	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für alle Klassen der Wissensbasis Breast Tissue.	99
32	Ergebnisse der Lernprozesse für alle Klassen der Wissensbasis Breast Tissue anhand des Laufzeit Parameters.	100
33	Ablaufbeschreibung der Konvertierung der Cardiotocography Wissensbasis	111
34	Ergebnisse der Lernprozesse für das zehn Klassen Lernproblem der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	113
35	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für das zehn Klassen Lernproblem der Wissensbasis Cardiotocography.	116
36	Ergebnisse der Lernprozesse für das zehn Klassen Lernproblem der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	116
37	Ergebnisse der Lernprozesse für das drei Klassen Lernproblem der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	118
38	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für das drei Klassen Lernproblem der Wissensbasis Cardiotocography.	120
39	Ergebnisse der Lernprozesse für das drei Klassen Lernproblem der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	120
40	Ergebnisse der Lernprozesse für alle gelernten Klassen anhand des Noise Parameters.	135
41	Ergebnisse der Lernprozesse für die Klasse Carcinoma der Wissensbasis Breast Tissue anhand des Noise Parameters.	139
42	Ergebnisse der Lernprozesse für die Klasse Fibro-adenoma der Wissensbasis Breast Tissue anhand des Noise Parameters.	140
43	Ergebnisse der Lernprozesse für die Klasse Mastopathy der Wissensbasis Breast Tissue anhand des Noise Parameters.	141

44	Ergebnisse der Lernprozesse für die Klasse Glandular der Wissensbasis Breast Tissue anhand des Noise Parameters.	142
45	Ergebnisse der Lernprozesse für die Klasse Connective der Wissensbasis Breast Tissue anhand des Noise Parameters.	143
46	Ergebnisse der Lernprozesse für die Klasse Adipose der Wissensbasis Breast Tissue anhand des Noise Parameters.	144
47	Ergebnisse der Lernprozesse für die Klassen Fibro-adenoma, Mastopathy und Glandula der Wissensbasis Breast Tissue anhand des Noise Parameters.	145
48	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse Carcinoma der Wissensbasis Breast Tissue.	146
49	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse Fibro-adenoma der Wissensbasis Breast Tissue.	147
50	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse Mastopathy der Wissensbasis Breast Tissue.	147
51	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse Glandular der Wissensbasis Breast Tissue.	148
52	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse Connective der Wissensbasis Breast Tissue.	148
53	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse Adipose der Wissensbasis Breast Tissue.	149
54	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klassen Fibro-adenoma, Mastopathy und Glandula der Wissensbasis Breast Tissue.	149
55	Ergebnisse der Lernprozesse für die Klasse Carcinoma der Wissensbasis Breast Tissue anhand des Laufzeit Parameters.	150
56	Ergebnisse der Lernprozesse für die Klasse Fibro-adenoma der Wissensbasis Breast Tissue anhand des Laufzeit Parameters.	151
57	Ergebnisse der Lernprozesse für die Klasse Mastopathy der Wissensbasis Breast Tissue anhand des Laufzeit Parameters.	151
58	Ergebnisse der Lernprozesse für die Klasse Glandular der Wissensbasis Breast Tissue anhand des Laufzeit Parameters.	152
59	Ergebnisse der Lernprozesse für die Klasse Connective der Wissensbasis Breast Tissue anhand des Laufzeit Parameters.	152

60	Ergebnisse der Lernprozesse für die Klasse Adipose der Wissensbasis Breast Tissue anhand des Laufzeit Parameters.	153
61	Ergebnisse der Lernprozesse für die Klassen Fibro-adenoma, Mastropathy und Glandula der Wissensbasis Breast Tissue anhand des Laufzeit Parameters.	153
62	Ergebnisse der Lernprozesse für die erste Klasse der Wissensbasis Carditocography anhand des Noise Parameters.	154
63	Ergebnisse der Lernprozesse für die zweite Klasse der Wissensbasis Carditocography anhand des Noise Parameters.	155
64	Ergebnisse der Lernprozesse für die dritte Klasse der Wissensbasis Carditocography anhand des Noise Parameters.	156
65	Ergebnisse der Lernprozesse für die vierte Klasse der Wissensbasis Carditocography anhand des Noise Parameters.	157
66	Ergebnisse der Lernprozesse für die fünfte Klasse der Wissensbasis Carditocography anhand des Noise Parameters.	158
67	Ergebnisse der Lernprozesse für die sechste Klasse der Wissensbasis Carditocography anhand des Noise Parameters.	159
68	Ergebnisse der Lernprozesse für die siebte Klasse der Wissensbasis Carditocography anhand des Noise Parameters.	160
69	Ergebnisse der Lernprozesse für die achte Klasse der Wissensbasis Carditocography anhand des Noise Parameters.	161
70	Ergebnisse der Lernprozesse für die neunte Klasse der Wissensbasis Carditocography anhand des Noise Parameters.	162
71	Ergebnisse der Lernprozesse für die zehnten Klasse der Wissensbasis Carditocography anhand des Noise Parameters.	163
72	Ergebnisse der Lernprozesse für die Klasse Normal der Wissensbasis Carditocography anhand des Noise Parameters.	164
73	Ergebnisse der Lernprozesse für die Klasse Suspect der Wissensbasis Carditocography anhand des Noise Parameters.	165
74	Ergebnisse der Lernprozesse für die Klasse Pathological der Wissensbasis Carditocography anhand des Noise Parameters.	166
75	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse eins der Wissensbasis Carditocography.	167

76	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse zwei der Wissensbasis Cardiotocography.	168
77	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse drei der Wissensbasis Cardiotocography.	169
78	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse vier der Wissensbasis Cardiotocography.	170
79	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse fünf der Wissensbasis Cardiotocography.	171
80	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse sechs der Wissensbasis Cardiotocography.	172
81	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse sieben der Wissensbasis Cardiotocography.	173
82	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse acht der Wissensbasis Cardiotocography.	174
83	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse sechs der Wissensbasis Cardiotocography.	175
84	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse sechs der Wissensbasis Cardiotocography.	176
85	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse Normal der Wissensbasis Cardiotocography.	177
86	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse Suspect der Wissensbasis Cardiotocography.	178
87	Laufzeit der Algorithmen CELOE und OCEL in Abhängigkeit des Noise Parameters für die Klasse Pathological der Wissensbasis Cardiotocography.	179
88	Ergebnisse der Lernprozesse für die erste Klasse der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	180
89	Ergebnisse der Lernprozesse für die zweite Klasse der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	181
90	Ergebnisse der Lernprozesse für die dritte Klasse der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	182
91	Ergebnisse der Lernprozesse für die vierte Klasse der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	183

<i>LISTINGS</i>	201
92 Ergebnisse der Lernprozesse für die fünfte Klasse der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	184
93 Ergebnisse der Lernprozesse für die sechste Klasse der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	185
94 Ergebnisse der Lernprozesse für die siebte Klasse der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	186
95 Ergebnisse der Lernprozesse für die achten Klasse der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	187
96 Ergebnisse der Lernprozesse für die fünfte Klasse der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	188
97 Ergebnisse der Lernprozesse für die fünfte Klasse der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	189
98 Ergebnisse der Lernprozesse für die Klasse Normal der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	190
99 Ergebnisse der Lernprozesse für die Klasse Suspect der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	191
100 Ergebnisse der Lernprozesse für die Klasse Pathological der Wissensbasis Cardiotocography anhand des Laufzeit Parameters.	192

Listings

1 Hier ist der Aufbau einer ARFF Datei dargestellt. Zuerst wird der Relationsname mit <i>@relation</i> definiert und im Anschluss daran werden die Attribute mit ihren möglichen Ausprägungen mit <i>@attribute</i> festgelegt. Mit <i>@data</i> wird signalisiert, dass nur noch Instanzdaten folgen. Jede Instanz wird auf einer neuen Zeile dargestellt und die Attributsausprägungen werden hintereinander, durch Komma getrennt angegeben. . .	34
2 Prolog Datei der Mutagenesis Wissensbasis. Hier sind die Atom und Atombindungen dargestellt. Die Blöcke 3 bis 7 sind für die einzelnen Eigenschaften die oben aufgezählt sind.	39

Tabellenverzeichnis

1 Manchester OWL Syntax Beispiele mit Erklärung	23
-----------------------------------------------------------	----

2	Höchste Testgenauigkeiten und die dazu entstandenen Laufzeiten und Konzeptlängen für Mutagenesis 1.	43
3	Signifikanzergebnisse des ersten Teils der Mutagenesis Wissensbasis.	43
4	Beste Konzepte der Algorithmen OCEL, CELOE und Aleph, für den ersten Teil der Wissensbasis Mutagenesis	44
5	Signifikanzergebnisse der Wissensbasis Mutagenesis 1 bei fest eingestellter Laufzeit.	48
6	Lernergebnisse von Mutagenesis 2	48
7	Signifikanzergebnisse des zweiten Teils der Mutagenesis Wissensbasis.	49
8	Beste Konzepte der Algorithmen OCEL, CELOE und Aleph, für den zweiten Teil der Wissensbasis Mutagenesis	49
9	Höchste Testgenauigkeiten und die dazu entstandenen Laufzeiten und Konzeptlängen für die Wissensbasis Suramin.	57
10	Signifikanzergebnisse der Suramin Wissensbasis.	57
11	Beste Konzepte der Algorithmen OCEL, CELOE und Aleph, für die Wissensbasis Suramin.	58
12	Höchste Testgenauigkeiten und die dazu entstandenen Laufzeiten und Konzeptlängen für die Wissensbasis Parkinsons.	66
13	Signifikanzergebnisse der Parkinsons Wissensbasis.	67
14	Beste Konzepte der Algorithmen OCEL, CELOE und Weka, für die Wissensbasis Parkinsons.	68
15	Signifikanzergebnisse der Wissensbasis Parkinsons bei fest eingestellter Laufzeit. . .	70
16	Höchste Testgenauigkeiten und die dazu entstandenen Laufzeiten und Konzeptlängen für die Wissensbasis Heart.	76
17	Signifikanzergebnisse der Heart Wissensbasis.	77
18	Beste Konzepte der Algorithmen OCEL, CELOE und Weka, für die Wissensbasis Heart.	78
19	Signifikanzergebnisse der Wissensbasis Heart bei fest eingestellter Laufzeit.	81
20	Höchste Testgenauigkeiten und die dazu entstandenen Laufzeiten und Konzeptlängen für die Wissensbasis Mammographic.	87
21	Signifikanzergebnisse der Mammographic Wissensbasis.	88
22	Beste Konzepte der Algorithmen OCEL, CELOE und Weka, für die Wissensbasis Mammographic.	89

23	Signifikanzergebnisse der Wissensbasis Mammographic bei fest eingestellter Laufzeit.	90
24	Höchste durchschnittliche Testgenauigkeiten, für die Wissensbasis Breast Tissue, die aus den Einzeltests entstanden ist.	97
25	Signifikanzergebnisse der Breast Tissue Wissensbasis.	98
26	Signifikanzergebnisse der Wissensbasis Breast Tissue bei fest eingestellter Laufzeit. .	101
27	Beste Konzepte der Algorithmen OCEL und CELOE, für die Klasse Carcinoma der Wissensbasis Breast Tissue.	101
28	Beste Konzepte der Algorithmen OCEL und CELOE, für die Klasse Fibro-Adenoma der Wissensbasis Breast Tissue.	102
29	Beste Konzepte der Algorithmen OCEL und CELOE, für die Klasse Mastropathy der Wissensbasis Breast Tissue.	103
30	Beste Konzepte der Algorithmen OCEL und CELOE, für die Klasse Glandular der Wissensbasis Breast Tissue.	104
31	Beste Konzepte der Algorithmen OCEL und CELOE, für die Klasse Connective der Wissensbasis Breast Tissue.	105
32	Beste Konzepte der Algorithmen OCEL und CELOE, für die Klasse Adipose der Wissensbasis Breast Tissue.	106
33	Beste Konzepte der Algorithmen OCEL und CELOE, für die Zusammenfassung der Klassen Fibro-Adenoma, Mastropathy und Glandular der Wissensbasis Breast Tissue.	107
34	Höchste durchschnittliche Testgenauigkeiten für die Wissensbasis Carditocography, die aus den Einzeltests der zehn Klassen entstanden ist.	114
35	Signifikanzergebnisse des drei Klassen Lernproblem der Wissensbasis Cardiotocography.	115
36	Signifikanzergebnisse des zehn Klassen Lernproblem der Wissensbasis Cardiotocography bei fest eingestellter Laufzeit.	117
37	Höchste durchschnittliche Testgenauigkeiten für die Wissensbasis Carditocography, die aus den Einzeltests der drei Klassen entstanden ist.	119
38	Signifikanzergebnisse des drei Klassen Lernproblem der Wissensbasis Cardiotocography.	121
39	Signifikanzergebnisse des drei Klassen Lernproblem der Wissensbasis Cardiotocography bei fest eingestellter Laufzeit.	122

40	Beste Konzepte der Algorithmen OCEL und CELOE, für die erste Klasse der Wissensbasis Cardiotocography.	122
41	Beste Konzepte der Algorithmen OCEL und CELOE, für die zweite Klasse der Wissensbasis Cardiotocography.	123
42	Beste Konzepte der Algorithmen OCEL und CELOE, für die dritte Klasse der Wissensbasis Cardiotocography.	124
43	Beste Konzepte der Algorithmen OCEL und CELOE, für die vierte Klasse der Wissensbasis Cardiotocography.	125
44	Beste Konzepte der Algorithmen OCEL und CELOE, für die fünfte Klasse der Wissensbasis Cardiotocography.	126
45	Beste Konzepte der Algorithmen OCEL und CELOE, für die sechste Klasse der Wissensbasis Cardiotocography.	127
46	Beste Konzepte der Algorithmen OCEL und CELOE, für die siebte Klasse der Wissensbasis Cardiotocography.	128
47	Beste Konzepte der Algorithmen OCEL und CELOE, für die achte Klasse der Wissensbasis Cardiotocography.	129
48	Beste Konzepte der Algorithmen OCEL und CELOE, für die neunte Klasse der Wissensbasis Cardiotocography.	130
49	Beste Konzepte der Algorithmen OCEL und CELOE, für die zehnte Klasse der Wissensbasis Cardiotocography.	131
50	Beste Konzepte der Algorithmen OCEL und CELOE, für die Klasse Normal der Wissensbasis Cardiotocography.	132
51	Beste Konzepte der Algorithmen OCEL und CELOE, für die Klasse Suspect der Wissensbasis Cardiotocography.	133
52	Beste Konzepte der Algorithmen OCEL und CELOE, für die Klasse Patologic der Wissensbasis Cardiotocography.	134

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Leipzig, den

Christian Kötteritzsch