KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

ÇİZGE VERİTABANLARINDA KAVRAM KEŞFİ

FURKAN GÖZ

KOCAELİ 2017

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

YÜKSEK LİSANS TEZİ

ÇİZGE VERİTABANLARINDA KAVRAM KEŞFİ

FURKAN GÖZ

Yrd. Doç. Dr. Alev MUTLU		
Danışman,	**Kocaeli Üniversitesi**	.....
Doç. Dr. Sevinç İLHAN OMURCA		
Jüri Üyesi,	**Kocaeli Üniversitesi**	.....
Yrd. Doç. Dr. Burcu YILMAZ		
Jüri Üyesi,	**Gebze Teknik Üniversitesi**	.....

Tezin Savunulduğu Tarih: 16.06.2017

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, kavram keşfi problemi için çizge veritabanı kullanarak yol bulma tabanlı bir yöntem geliştirilmiştir. Yöntem, farklı öğrenme problemlerinde kullanılan veri kümeleri ile test edilmiştir.

Yüksek lisans öğrenimim boyunca değerli birikimlerini benimle paylaşan, tezimin her aşamasında sorunlarımı dinleyerek, çalışmalarıma yön veren ve yoğun akademik yaşamında değerli zamanını her türlü problemimi çözmeye ayıran tez danışmanım saygıdeğer hocam Yrd. Doç. Dr. Alev MUTLU'ya teşekkürlerimi sunarım.

Akademik çalışmalarım sırasında, birçok aşamada beni her konuda dinleyen ve destekleyen Bilgisayar Mühendisliği Bölümü araştırma görevlilerine teşekkür ederim.

Hayatım boyunca bana güç veren en büyük destekçilerim, her aşamada sıkıntılarımı ve mutluluklarımı paylaşan sevgili babam Yüksel GÖZ, annem Bağdat GÖZ ve kardeşim Büşra GÖZ'e teşekkürlerimi sunarım.

Haziran – 2017 Furkan GÖZ

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR i
İÇİNDEKİLER ii
ŞEKİLLER DİZİNİ iii
TABLOLAR DİZİNİ iv
SİMGELER VE KISALTMALAR DİZİNİ v
ÖZET vi
ABSTRACT vii
GİRİŞ 1
1. GENEL BİLGİLER 4
1.1. Tez Çalışmasının Amacı ve Başlatma Sebebi 4
1.2. Tez Çalışmasının Katkıları 4
1.3. Literatür Taraması 5
1.4. Tezin Yapısı 9
2. TEMEL KAVRAMLAR 11
2.1. Kavram Keşfi 11
2.2. Çizge Veri Tabanı 13
3. YÖNTEM: ÇİZGE VERİ TABANI İLE KAVRAM KEŞFİ 19
3.1. Verilerin Modellenmesi 20
3.1.1. İki argümanlı ilişkilerin modellenmesi 21
3.1.2. N-argümanlı ilişkilerin modellenmesi 24
3.2. Aday Kavram Tanımlarının Bulunması ve Değerlendirilmesi 30
3.1.1. Hedefe farklı yollardan ulaşılabilen çizgede yol bulma 32
3.1.2. Hedefe farklı yollardan ulaşılamayan çizgede yol bulma 34
3.1.3. Aday kavram tanımlarının değerlendirilmesi 36
4. DENEYSEL SONUÇLAR 40
4.1. Veri Kümeleri 41
4.2. Sonuçlar 42
5. SONUÇLAR VE ÖNERİLER 53
KAYNAKLAR 55
KİŞİSEL YAYIN VE ESERLER 61
ÖZGEÇMİŞ 62

ŞEKİLLER DİZİNİ

Şekil 1.1. Kavram keşfi yöntemleri 5
Şekil 2.1. Yerel plato problemi 13
Şekil 2.2. NoSQL türleri 15
Şekil 2.3. Çizge veri tabanı çeşitlerine genel bakış 16
Şekil 2.4. Neo4j ile oluşturulmuş çizge örneği 17
Şekil 3.1. Yöntemin genel kodu 20
Şekil 3.2. Verilerin modellenmesi 21
Şekil 3.3. Kinship veri kümesi için örnek çizge 23
Şekil 3.4. Kinship veri kümesinin Neo4j'de örnek bir gösterim 24
Şekil 3.5. muta_atm çizge örneği gösterimi 27
Şekil 3.6. muta_atm çizge gösterimi 28
Şekil 3.7. muta_bond çizge örneği gösterimi 29
Şekil 3.8. d1'in alt çizge model örneği 29
Şekil 3.9. Mutagenicity d1 ilacı için Neo4j'de örnek alt çizge 30
Şekil 3.10. Kavram tanımlarının bulunması ile ilgili akış diyagramı 31
Şekil 3.11. Kinship için örnek yollar 32
Şekil 3.12. Örnek yollar ve yönleri 33
Şekil 3.13. Mutagenicity için örnek yollar 36
Şekil 3.14. Mutagenicity için sadeleştirilmiş örnek yollar 36
Şekil 3.15. Örnek yollar ve genelleştirilmiş halleri 37
Şekil 3.16. Örnek yollar ve açıkladığı ilişkiler 38
Şekil 4.1. Aday kuralın değerlendirilme aşaması 40
Şekil 4.2. Ölçeklenebilirlik 52

TABLOLAR DİZİNİ

Tablo 2.1. Mutagenicity örnek bilgiler 11
Tablo 2.2. Mutagenicity veri kümesi aday kavram tanımları örneği 12
Tablo 2.3. Mutagenicity veri kümesi kavram tanımları örneği 12
Tablo 3.1. İki argümanlı verilerin hedef örnekleri 21
Tablo 3.2. son için hedef örnekleri ve arka plan bilgileri 22
Tablo 3.3. Hedef örnekleri ve arka plan bilgileri 23
Tablo 3.4. N-argümanlı verilerin hedef örnekleri 24
Tablo 3.5. Lumo özelliğinin ayrıklaştırma sonucu 25
Tablo 3.6. Özelliklerin sınıflandırma sayıları 26
Tablo 3.7. Çok ilişkiye ait örnekler 26
Tablo 3.8. PTE ve Mutagenicity için oluşturulan yollar 34
Tablo 3.9. Gezilmesi gereken düğümler 35
Tablo 3.10. son(A,B) için aday kurallar 38
Tablo 3.11. son(A,B) için kavram tanımları 39
Tablo 4.1. Veri kümelerine bakış 41
Tablo 4.2. N-argümanlı verilerin hedef örnekleri 41
Tablo 4.3. Minimum destek ve güven değerleri 42
Tablo 4.4. Çizgelerin özellikleri ve oluşma zamanları 42
Tablo 4.5. Elti ilişkisini açıklayan kurallar 43
Tablo 4.6. Kinship ilişkileri ve kavram tanımlayıcıları 43
Tablo 4.7. Kinship veri kümesinde eşlerin durumu 45
Tablo 4.8. Dünür ilişkisini açıklayan kurallar 45
Tablo 4.9. Same Generation ilişkisini açıklayan kurallar 46
Tablo 4.10. Eastbound ilişkisini açıklayan kurallar 46
Tablo 4.11. Test sonuçları ve süreleri 47
Tablo 4.12. Karşılaştırmalı sonuçlar 48
Tablo 4.13. Mutagenecity çalışmasındaki örnek yollar 49
Tablo 4.14. Mutagenecity kavram tanımları 49
Tablo 4.15. PTE veri kümesi için kavram tanımlar 50
Tablo 4.16. Farklı uzunluktaki kurallar için metodun çalışma süreleri 50
Tablo 4.17. brother ve father için yol sayıları 51

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

NoSQL	:	Not Only SQL (İlişkisellikten Öte Sorgu Dili)
PTE	:	Predictive Toxicology Evaluation (Tahmini Toksikoloji Değerlemesi)
SQL	:	Structured Query Language (Yapılandırılmış Sorgu Dili)
TMP	:	Tümevarım Mantık Programlama

ÇİZGE VERİTABANLARINDA KAVRAM KEŞFİ

ÖZET

Çok ilişkili veri madenciliğinin en çok çalışılan problemlerinden birisi olan Kavram keşfi bir ilişkinin mantıksal tanımlamasının diğer ilişkileri kullanarak çıkarılmasıyla ilgilenir. Bu probleme genellikle Tümevarım Mantık Programlama ve çizge tabanlı yöntemlerle çözüm aranmaktadır.

Bu çalışmada kavram keşfi problemi çizge veri tabanları perspektifinden ele alınarak yol bulma tabanlı bir yöntem önerilmiştir. Verileri depolamak ve hedef ilişkileri açıklayan kavram tanımlarını bulmak için Neo4j çizge veri tabanı kullanılmıştır. Aynı zamanda kavram tanımlarının bulunmasında yöntemin performansını arttırmak için bazı tekniklere başvurulmuştur.

Araştırmanın deneysel sonuçları önerilen yöntemin farklı öğrenme problemlerinde kullanılan veri kümeleri için kavram tanımlarının bulunmasında başarılı olduğunu göstermektedir. Daha önce uygulanmış yöntemlerle kıyaslandığında benzer kavram tanımları bulunmuştur. Yöntemin kavram tanımlarının bulunmasında benzer kuralların bulunduğu sonuçların hedefleri kapsamada başarılı olduğu görülmektedir. Son olarak yöntemin farklı uzunluklu kavram tanımları için sonuçlar aranarak elde edilen sürelerle ölçeklenebilirliği tespit edilmiştir.

Anahtar kelimeler: Çizge Veri Tabanı, Kavram Keşfi, Neo4j, Yol Bulma.

CONCEPT DISCOVERY IN GRAPH DATABASE

ABSTRACT

Concept discovery is one of the most studied problems of multi-relational data mining. Concept discovery is concerned with inducing logical definitions of a relation by using other relations. This problem has long been studied from Inductive Logic Programming and graph-based methods.

In this work the problem of concept discovery is discussed from the perspective of graph databases. A path finding method is proposed for concept discovery in graph databases. Neo4j graph database is used to store data and find the concept descriptors that define the target relation. At the same time some techniques have been used to improve the performance of the method in the finding of concept descriptors for concept discovery.

Experimental results show that the proposed method is successful in the discovery of concept descriptors for data sets belonging to different learning problems. When compared with previously applied methods similar concept descriptors are found.

Keywords: Graph Database, Concept Discovery, Neo4j, Path Finding.

GİRİŞ

Veri toplama ve depolama araçlarının hem ucuzlaması hem de yaygın bir şekilde kullanılması ile birlikte organizasyonların arşivlediği veri miktarı artmıştır. Bu artan veri yığınlarının anlamlandırılması ve veri içindeki ilişkilerin bulunması ihtiyacı veri madenciliği kavramını ortaya çıkarmış ve günümüzde hem endüstride hem de akademide en çok çalışılan konulardan biri haline gelmiştir.

Geleneksel veri madenciliği yöntemleri genellikle işletim sistemin dosyalarında saklanan ve özellik – değer biçiminde tutulan verilerden bilgi çıkarmakla ilgilenir. Verilerin ilişkisel veri tabanlarında ve günümüzde de artan bir hızla çizge veri tabanlarında saklanması veri madenciliğinde yeni yaklaşımların ortaya çıkmasına neden olmuştur.

İlişkisel veri tabanlarından veri çıkarmak için iki farklı yaklaşım takip edilmiştir. İlk yaklaşımda ilişkisel veri modelinde saklanan veriler değer – özellik modeline dönüştürülerek klasik veri madenciliği yöntemleri kullanılmıştır [1]. Bu yaklaşımın temel kısıtı veri modeli dönüştürme aşamasındaki veri kaybıdır. İkinci yaklaşımda ise doğrudan ilişkisel modelde saklanan veriler üzerinde çalışabilecek yöntemlerin geliştirilmesi olmuştur [2]. Çizge veri madenciliği ise çizge modelinde saklanan veriden bilgi çıkarımı ile ilgilenmektedir. Alan yazınında ilişkisel verilerin kayıpsız olarak çizge ya da hiper-çizge (hyper-graph) modeline dönüştürülmesinde farklı yöntemler sunulmuştur [3]. Bağlı bileşenler, en sık görünen alt-çizge gibi çizge tabanlı yaklaşımlar çizge modelinde saklanan verilerden bilgi çıkarmak için uygulanmaktadır.

İlişkisel veri madenciliğinde en çok çalışılan konulardan biri kavram keşfidir. İlişkisel model çerçevesinde saklanan veride, kavram keşfi problemi hedef ilişki olarak adlandırılan bir ilişkinin tanımının modeldeki diğer ilişkiler cinsinden bulunmasıdır [4]. Özellikleri verilen kanserojen ve kanserojen olmayan kimyasallardan kanserojen olanlarının tanımının bulunması örnek bir kavram keşif problemidir.

Tümevaran Mantıksal Programlama (TMP) kavram keşfi probleminde yaygın olarak kullanılan yöntemlerden bir tanesidir [5]. TMP, makine öğrenmesi ve mantık programlamayı temel alarak arka plan bilgileri üzerinden hedef örnekleri açıklayan hipotezler bulmaya çalışır. Bu yaklaşımda yukarıdan aşağıya, aşağıdan yukarıya ve hibrit olacak şekilde üç farklı yaklaşımla öğrenme yapılır [10]. Yukarıdan aşağıya yaklaşımlarda tüm örnekleri açıklayan aşırı genel bir hipotezden başlanarak ve özelleştirme operatörleri kullanılarak sadece pozitif örnekleri açıklayan özel hipotezler üretilir. Aşağıdan yukarıya yaklaşımlarda sadece bir pozitif örneği açıklayan aşırı özel bir hipotezle başlanarak ve genelleştirme operatörleri kullanılarak tüm pozitif örnekleri açıklayan genel hipotezler üretilir. Hibrit yaklaşımlarda ise hem özelleştirme hem de genelleştirme operatörleri kullanılarak tüm pozitif örnekleri açıklayan hipotezler oluşturulmaya çalışılır [6].

Kavram keşfi için TMP tabanlı yaklaşımlarda bazı problemler bulunmaktadır. Bunlardan ilki TMP'de kaynak kullanımının ve öğrenmenin yavaş olmasıdır [7]. Bir diğeri ise öğrenmenin kalitesini etkileyen yerel plato problemidir [8]. Ayrıca ölçeklendirme de TMP'de görülen problemlerden bir tanesidir [9].

Kavram keşfinde çizge tabanlı yöntemler öncelikle yerel plato problemine çözüm getirmek için geliştirilmiştir. TMP tabanlı sistemler kavram tanımlarını her aşamada tanıma yeni bir ilişki ekleyerek - çıkararak ya da tanımda var olan bir ilişkinin argümanlarını genelleyerek - özelleştirerek oluşturur. Çizge tabanlı sistemler ise kavram tanımı oluştururken her aşamada bir ya da daha çok ilişki ekleyerek oluşturur. Çizge tabanlı yaklaşımların kavram keşfinin bu problemini çözebilmesi, ilişkisel modelde saklanan verinin çizge olarak gösterildiğinde verinin ve anlamının kaybolmaması ve de çizge algoritmalarının kavram keşfi için uyarlanabilir olması alan yazınında bu tür çalışmaların artmasına neden olmuştur.

Çizge tabanlı kavram keşif sistemleri iki ana gruba ayrılabilir. Yol bulma tabanlı olarak adlandırılan birinci grup yaklaşımlarda hedef örneğin argümanlarını bağlayan yollar bulunur. Eğer bu tür yollar her örnek için geçerli ise bu yolların genelleştirilmiş hali kavram tanımı olarak adlandırılır. Eş yapı bulma olarak adlandırılan ikinci grup çalışmalarda ise hedef örneği içeren al yapılar bulunur. Eğer

bu tür alt yapılar her örnek için geçerliyse bu yapılar kavram tanımı olarak kabul edilir.

Bu çalışmada çizge veri tabanlarında saklanan veriler için yol bulma tabanlı bir kavram keşfi önerilmektedir. Önerilen yaklaşımda verinin ilişkisel modelde saklanan veri kaynaklarından geldiği kabul edilerek bu tür verinin çizge olarak modellenmesi için bir yaklaşım sunulmuştur. Yol bulma tabanlı geliştirilen sistemde iki argümanlı ve n-argümanlı ilişkiler için farklı çizge gösterim modelleri kurulmuştur. İki argümanlı ilişkiler için düğümlerin ilişkinin argümanları, iki düğümü bağlayan kenarların ise bu iki düğüm arasında geçerli olan ilişkiyi temsil ettiği bir veri gösterim modeli sunulmuştur. n-argümanlı ilişkilerin çizge olarak modellenmesinde ise anlam kaybını önlemek için ilişkiler hiper-çizgeler olarak modellenmiştir. Kavram keşif ve değerlendirme sürecini hızlandırmak için n-argümanlı ilişkilerin kavram tanımlarının öğrenilmesinde iki argümanlı ilişkilerin tanımlarının öğrenilmesinde kullanılan tüketme yöntemi yerine alan uzamalarının bilgisine başvurularak yönlendirilmiş arama yöntemi kullanılmıştır.

Sunulan çalışmada verileri depolamak için Neo4j çizge veri tabanı kullanılmıştır. Kavram tanımlarının keşfinde ise Neo4j'nin sunduğu kütüphanelerden faydalanılarak geliştirilen yol bulma tabanlı algoritma kullanılmıştır. Yapılan deneylerde elde edilen sonuçlar alan yazınında raporlanan sonuçlarla benzerlik göstermektedir.

1. GENEL BİLGİLER

1.1. Tez Çalışmasının Amacı ve Başlatılma Sebebi

Öteden beri bilişim sistemlerinin kullanılmasıyla ihtiyaç duyulan bilginin keşfi ve çıkarımı yaygın bir çalışma konusu olmuştur. Gelişen teknoloji ile birlikte ortaya çıkan büyük veri bu konuya olan ilgiyi arttırmıştır. Ancak bilginin çıkarılması sırasında karşılaşılan bazı problemler bu alanda yeni yöntemlerin araştırılmasını gerekli kılmıştır. Bu problemlerden en önemlisi kavram keşfinde ölçeklenebilirliktir. Bu özellik veri miktarı arttıkça verinin büyüklüğüne bağlı olarak performansının maksimum seviyede korumayı ifade eder. TMP ile yapılan kavram keşfinde veri arttıkça performansın düştüğü bilinmektedir. Bunun dışında değişen veri hacminde verilerin işlenmesi, depolanması ve analizindeki güçlükler kavram keşfinde karşılaşılan diğer problemlerdir. Çizge veri tabanında veri modellemesinin ve veriye erişimin basit ve hızlı bir şekilde yapılabilmesi bu problemlerin çözümünde ana motivasyon kaynağı olmaktadır.

Bu çalışmada çizge tabanlı kavram keşfi için farklı veri kümelerinde arka plan bilgileri kullanılarak hedef ilişkilerin açıklanması amaçlanmıştır. Bu amaçla farklı formattaki verilerin çizgeye dönüştürülerek yeniden yapılandırılması düşünülmüştür. Bu sayede çizge verileri ile kavramların çıkarımı ve analizi mümkün olacaktır. Böylece çizge tabanlı yaklaşımın TMP'ye göre avantajlarının ve verimliliğinin ortaya konulması hedeflenmektedir. Bunun yanında farklı uzunluklu kuralların araştırılarak önerilen sistemin ölçeklenebilir olduğu ispatlanmak istenmektedir.

1.2.Tez Çalışmasının Katkıları

Bu çalışmada aşağıdaki katkıların sağlanması amaçlanmıştır.

Çizge veri tabanlarında kavram keşfinin yol bulma tabanlı bir yöntemle gerçekleştirilebilmesi.

Önerilen sistemin farklı özelliklere verilerde uygulanabilirliğinin gösterilmesi.

Performans anlamında daha önce çalışılmış kavram keşfi problemlerinde bir takım iyileştirmelerin yapılması.

n-argümanlı ilişki içeren veri kümelerinin modellenebilmesi.

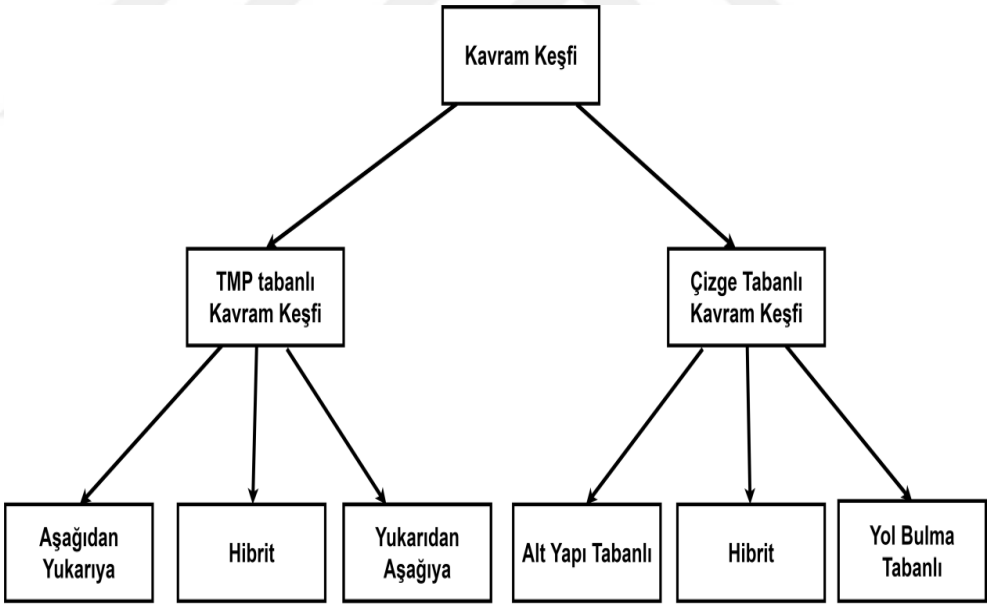
Yönteminve n-argümanlı ilişki içeren veri kümelerinde kavram tanımının yüksek doğruluk oranları ile yapılabilmesi.

Sistemin ölçeklenebilirliğinin gösterilmesi.

Bazı programlama teknikleri kullanılarak arama uzayının küçültülmesi ile performans artışının sağlanması.

1.3.Literatür Taraması

Bu bölümde kavram keşfi için literatür çalışması sunulmuştur. Şekil 1.1'de kavram keşfi problemi için kullanılan yaklaşımlar gösterilmiştir.



Şekil 1.1. Kavram keşfi yöntemleri

Literatürde TMP tabanlı Kavram Keşfi için aşağıdan yukarıya mantığı kullanılarak çeşitli çalışmalar bulunmaktadır. Bu çalışmalardan biri olan GOLEM yaklaşımında iç içe geçmiş iki döngü bulunmaktadır. Dış döngüde pozitif örnekleri kapsayan kurallar rastgele seçilir. İç döngüde ise açıklanmayan pozitif örneklerin için özel bir değer hesaplanır ve en yüksek kapsama alanına sahip kural bulunur. Ağ [11], uydu

[12] ve ilaçlar için yapı aktivite ilişkilerinin bulunması [13] gibi çeşitli problemlerde bu yöntem başarıyla uygulanmıştır [14]. Bu yaklaşımın eksikliği sadece pozitif örnekler için kullanılabiliyor olmasıdır.

CIGOL (logic kelimesinin tersi) TMP tabanlı Kavram Keşfi için aşağıdan yukarıya mantığını kullanan yöntemlerden bir diğeridir. Bu yöntem makine öğrenmesinde açıklama tabanlı öğrenme yaklaşımına benzemektedir. Tek örneklerin yoğun bir şekilde analizini gerçekleştirerek açıklanmamış kuralları tersine çevirme mantığına göre açıklamaya çalışır [15].

Kavram keşfi için yukarıdan aşağıya TMP yaklaşımı kullanılan çalışmalardan ALEPH kavram tanımının bulunmasında minimum destek ve güven değerleri ile arıtma operatörlerini kullanarak çalışır. Farklı tipteki problemlerde kullanılan ALEPH'te moleküler ve biyolojik yapıların ilişkilerini tanımlayan yapı aktivite bağlantılarının [16] kurulmasında önemli sonuçlar elde edilmiştir [17].

WARM ismi verilen bir başka çalışmada Apriori algoritması kullanılarak yukarıdan aşağıya TMP tabanlı bir sistem geliştirilmiştir. Bu yöntemde başlangıçta sık geçen sorgular, sık geçmeyen sorgular ve aday sorgular olmak üzere üç liste belirlenir. Uygun listeler algoritma içinde güncellenir. Temelde parametre olarak alınan hedef ilişkileri içinde barındıran en sık kurallar bulunur ve belirlenen güven değerini sağlayan ilişkilendirme kuralları çıkarılır [18].

Progol inverse entailment denen özel bir operatör kullanarak yukarıdan aşağıya TMP tabanlı bir yöntemdir. Arka plan bilgileri ve hedef örnekleri dışında mod işlemlerini de kullanır. Budama yöntemi ile en özel kuralları bulmaya çalışır [19].

Yukarıdan aşağıya TMP tabanlı bir çalışma olan FOIL'de maksimum sayıda özelliği açıklayan kurallar çözüm olarak alınmaya çalışır. Negatif örnekleri açıklayan kurallar azaltıp pozitif örnekleri açıklayan kuralları arttırmak için gain metriği kullanılır [20].

Kavram keşfi için SAHILP adlı çalışmada yukarıdan aşağıya TMP tabanlı yaklaşımı benzetilmiş tavlama algoritması kullanarak gerçekleştirilmiştir. Bu şekilde arama

uzayının daha etkili bir şekilde kullanılması amaçlanmaktadır. Bu yöntem öncelikli olarak aranan kuralları bulmada başarılı sonuçlar vermiştir [21].

MIS yukarıdan aşağıya yaklaşımını kullanan TMP tabanlı kavram keşfi yapar. Başlangıçta boş olan çözüm listesi örnekler okunarak doldurulur. Negatif örneklere benzeyen örnekler çözüm kümesine dahil edilmezken pozitif örneklere benzeyen örnekler çözüm kümesine dahil edilir. Çözüm kümesi tüm örnekler tamamlanana kadar devam eder [22].

LINUS çalışması yukarıdan aşağıya TMP tabanlı yöntemlerden biridir. Literatürde aynı türden verilen çalışmalardan farklı olarak verilerin özellik eşleşmesi formatında kullanılarak kavramların açıklanmasıdır. Üç aşamada gerçekleştirilen yöntemde önce ilişkiler uygun formata dönüştürülür. Çözüm aşaması gerçekleştirilir ve daha sonra veriler ilk formata dönüştürülerek yöntem tamamlanmış olur [23].

CRIS yukarıdan aşağıya ve aşağıdan yukarıya mantığını bir arada kullanan hibrit TMP tabanlı kavram keşfi yapan bir yöntemdir. Bu yöntemde kavram tanımlarını bulmak için aşağıdan yukarıya yaklaşımıyla hedef örneklerden rastgele birinin seçilmesiyle başlanır. Genelleştirilmiş kurallar bulunur ve gelişim tamamlanıncaya kadar yukarıdan aşağıya yaklaşımıyla kurallar özelleştirilir. Bu yöntemde ilişkisel kural madenciliği tekniklerinden faydalanılır [24].

TMP tabanlı yöntemlerde yerel plato, ölçeklenebilirlik ve öğrenmenin yavaş olması gibi problemler bulunmaktadır. Bu problemlere bir çözüm olarak kavram keşfi için çizge tabanlı yaklaşımlar ön plana çıkmıştır. Eş yapı ve yol bulma tabanlı yaklaşımlar çizge tabanlı kavram keşfi için literatürde karşılaşılan yöntemlerdendir. Eş yapı tabanlı yaklaşımlar olarak sınıflandırılan kavram keşif sistemlerinde ilk olarak çizgede benzer alt graflar araştırılır. Bulunan benzer alt yapılar tek bir düğüm noktası olacak şekilde temsil edilir. Yöntem tüm alt yapılar taranana kadar devam eder [25]. Yol bulma tabanlı yaklaşımlarda ise hedef örneklerin argümanları arasında önceden belirlenen bir uzunluğa sahip bulunan kurallar aday kavram tanımı olur. Bu kurallardan önceden belirlenen kriterlerin minimum değerlerini sağlayanlar kavram tanımları olarak alınır.

Eş yapı yaklaşımının kullanıldığı Subdue çizge tabanlı ilişkisel bir öğrenme sistemidir [26]. Subdue hedef örneklerin bulunduğu pozitif kuralları, sıkıştırma tabanlı bir yöntemle bulmaya çalışır. Bu sistemde bulunan kuralların birbirleriyle yakınlığını hesaplamak için çizge eşlemesi yapılır. Ayrıca birbirine benzeyen kuralları bulabilmek için minimum kodlama mantığına dayalı [27] minimum kural uzunluğu ve arka plan bilgilerinden faydalanılır [28]. Subdue, devre analizi [29], deprem aktiviteleri [30] ve kimyasal bileşik analizi [31] gibi problemlerde test edilmiştir.

Bir diğer çalışma olan SubdueCL çizge tabanlı ilişkisel kavram keşif sistemidir. Subdue sisteminin geliştirilmiş sürümü olarak görülebilen bu yöntemde Subdue'dan farklı olarak hem pozitif hem negatif örneklerden öğrenme yapılabilmektedir. SubdueCL'de Subdue yöntemindeki gibi sıkıştırma mantığı bulunmamaktadır. Bu yöntemin yerine pozitif ve negatif örnekleri ayrıştırma tekniği ile pozitif örneklerin tamamı açıklamaya çalışır ve bu şekilde en iyi kuralları bulmaya çalışır [32]. Moleküler biyoloji alanında karşılaşılan problemler için kullanılan veri setlerinde test edilmiştir [33].

Çizge madenciliğini daha etkin bir şekilde yapabilmek için geliştirilen Gaston çalışmasında hızlı başlangıç prensibi olarak adlandırılan ve aynı kuralı içeren yapıları kullanmaya dayalı bir yöntem geliştirilmiştir. Bu algoritmada sırasıyla en sık yollar, en sık boş ağaçlar ve döngüsel çizgeler aranarak kavram keşfi yapılır [34].

İlişkisel yol bulma tabanlı Richards'ın yaptığı çalışmada yerel maksimum ve yerel plato problemlerine çözüm önerilmektedir. Bu yöntem önceden belirli bir uzunlukta bulunan hedef örnekler arasında bağlantı kurmaya çalışır. Bu yöntemde rastgele pozitif örneklerden birini seçilerek başlanır. Seçilen örneğin argümanlarından bağlantılı düğümler genişletilerek farklı yollar bulunur. Bu yolların her birinin ayrı bir alt çizge olarak temsil edilmesiyle kesişen çizgeler aranır. Çizgeler arasında kesişim bulunursa belirlenen uzunlukta hedeflenen kavram bulunmuş olur. Kesişim bulunmazsa belirlenen uzunluk sınırını aşana kadar alt çizgeye düğüm eklenerek yol aramasına devam edilir [8].

Mutlu ve Karagöz tarafından sunulan hibrit çizge tabanlı kavram keşfi yaklaşımı yol bulma ve eş yapı tabanlı yöntemleri birleştirir. Bu yöntemde yol bulma tabanlı

yaklaşım çizge oluşturma aşamasında kullanılır. Yol bulma tabanlı yaklaşımı kavram tanımlarını bulmada kullanılmayan yöntemde aynı zamanda eş yapı tabanlı yaklaşımlarda bulunan sıkıştırma mantığını da kullanılmaz. Bu yaklaşımda kurallar bulunurken minimum güven ve destek değeri, arka plan bilgileri ve maksimum kural uzunluğu parametreleri esas alınır. Hibrit çizge tabanlı kavram keşfinde süreç kaynak ve hedef düğümlerin belirlenmesi ile başlar. Hedef örneklerin argümanlarından birincisi başlangıç düğümü ikincisi ise hedef düğümü olarak alınan yöntemde arka plan bilgileri kullanılarak sonsuz bir döngüye girmesine izin vermeden çizge düğüm ve düğümler arasındaki ilişkilerle genişletilir. Daha sonra düğümler birleştirilir ve kurallar bulunur. Belirlenen destek ve güven değerlerini geçemeyen kurallar iptal edilir [36].

Abay ve diğerleri kavram keşfi için Apriori algoritması ve graf veri tabanı kullanarak yol bulma ve bağlantılı kural madenciliği yöntemlerini birleştiren bir yöntem geliştirmişlerdir. Öncelikle oluşturulan grafta arka plan bilgileri kullanılarak hedef örneklerin parametreleri arasındaki tüm yollar bulunmaktadır. Bu aşamadan sonra bağlantılı kural madenciliğinin tekniklerinden faydalanarak kapsam, destek ve güven değerlerine bağlı olarak eleme işlemleri yapılır. Kalan kurallar kavram tanımları olarak alınır. Geliştirilen yöntem bu alanda yapılmış diğer çalışmalarla kıyaslanmış ve başarılı sonuçlar alınmıştır [25].

Bu tez çalışmasında ise kavram keşfi için çizge veri tabanı kullanarak yol bulma tabanlı bir yaklaşım önerilmiştir. Yöntemin tamamında graf veri tabanları ve özellikleri kullanılarak algoritmik bazı tekniklerle farklı öğrenme problemlerine ait veri setleri için uygulanabilir bir yöntem amaçlanmıştır. Yöntemin iki ya da n-argümanlı ilişkiler içeren veriler için ölçeklenebilir bir çözüm sunması beklenmektedir. Ayrıca bu alanda literatürde bulunan çalışmalarla kıyaslanarak ölçülebilir iyileşmeler beklenmektedir.

1.4. Tezin Yapısı

Bu tez çalışması 5 bölümden oluşmaktadır. Bölüm 1 tezin genel bilgilerinin bulunduğu kısımdır. Bu bölümde çalışmanın genel olarak tanıtılması, çalışmanın amacı ve başlatılma sebebi anlatılmaktadır. Literatür taraması da bu bölümde yapılmaktadır. Bölüm 2'de temel kavramlar verilmiştir. Bu bölümde kavram keşfi ve

çizge veri tabanları hakkında bilgiler verilmiştir. Bölüm 3'te tezin uygulaması olarak sunulan yöntemin açıklaması verilmiştir. Kavram keşfi için veri tabalarında yol tabanlı yaklaşımın nasıl kurulduğu detaylarıyla anlatılmıştır. Bölüm 4 ise çalışmanın deneysel sonuçlarından oluşmaktadır. Bu bölümde veri kümeleri için elde edilen kavram tanımları ve çalışma süreleri verilmiştir. Aynı zamanda sonuçlar alanında bulunan çalışmalarla kıyaslanmış ve algoritmanın ölçeklenebilirliği de burada araştırılmıştır. Çalışmanın son kısmı olan Bölüm 5 ise tez çalışmasında kullanılan yöntemin değerlendirildiği bölümdür.

2. TEMEL KAVRAMLAR

2.1. Kavram Keşfi

Veri madenciliği çalışma konularından biri veri hacimleri arasında tutulan daha önce öğrenilmemiş faydalı bilgilerin çıkarıldığı ve birçok alanda kullanılabilen bilgi keşfidir [37]. Bilgi çıkarımı yapılırken genellikle tek bir veri tablosu içeren veriler kullanılırken çok ilişkili veri madenciliği yaklaşımda birden fazla ilişki içeren bilgiler aranır. Kavram keşfi de çok ilişkili veri madenciliğinin uygulama alanlarından birisidir [38].

Kavram keşfi kısaca birbirleriyle bağlantısı bulunan nesneler arasında mantıksal ilişkilerin çıkarılması problemidir. Kavram keşfi problemlerinde ilişkisi verilen bilgilerin argümanları arasındaki bağlantıyı temsil eden hedef ilişki ve bu ilişkiyi açıklamak için arka plan bilgileri olarak adlandırılan diğer bilgiler kullanılmaktadır. Bu bilgiler kullanılarak hedef ilişkileri açıklayan kurallar bulunmaya çalışılır. Bulunan kurallar herhangi bir ölçümle doğrulanmadığı için aday kavram tanımları olarak alınır. Adayların belli yöntemlerle değerlendirilmesi ve belirlenen eşik değerlerini geçmesi sonucunda nihai kavram tanımları elde edilmiş olur.

Kavram keşif problemlerinde kullanılan veri kümelerinden bir tanesi olan Mutagenicity'de [40] mutajenik bileşikler açıklanmaya çalışılır. Bu veri kümesi için örnek hedef ilişkileri, arka plan bilgilerini gösteren örnekler Tablo 2.1'de verilmiştir.

Tablo 2.1. Mutagenicity örnek bilgiler

Hedef İlişki	**Arka Plan Bilgileri**
muta(d1,true)	ind1(d1,1) inda(d1,0) logp(d1,logp5) lumo(d1,lumo8) muta_bond(d1,d1_6,d1_10,1) muta_atm(d1,d1_6,c,22,ch8)

Tablo 2.1'de bulunan hedef örneği mutanın argümanlarından birincisinin d1, ikincisinin true olması d1 ilacının mutajenik olduğunu göstermektedir. İkinci

argümanı false alan bir örnek ise bir özelliğin mutajenik olmadığını gösterir. Bu örneğe göre kavram keşfi problemi hedef örnekte gösterilen d1 bileşiğinin mutajenik olup olmadığını tablonun ikinci sütununda verilen arka plan bilgileri ile açıklamaktır. Tablo 2.2'de arka plan bilgileri kullanarak bulunmuş bazı örnek aday kavram tanımları verilmiştir.

Tablo 2.2. Mutagenicity veri kümesi aday kavram tanımları örneği

Aday Kavram Tanımları	**Açıklanan Özellik Sayısı**
atm(A,B,h,3,ch15), atm(A,D,c,22,ch8), bond(A,D,B,1), inda(A,0), indV(A,1), (-0,122<=ch15<=-0,144), (-0,128<=ch8<=-0,106)	51
atm(A,B,h,3,ch14), atm(A,D,c,16,ch6), bond(A,D,B,1), indV(A,0), (-0,122<=ch15<=-0,144), (-0,128<=ch8<=-0,106)	3
atm(A,B,o,40,ch3), atm(A,D,n,38,ch21), bond(A,D,B,2), indV(A,1), (-0,494<=ch3<=-0,99), (0,553<=ch21<=0,793)	8
atm(A,B,c,22,ch7), atm(A,D,c,22,ch7), bond(A,D,B,7), inda(A,0), indV(A,1), (-0,167<=ch7<=-0,129)	9
atm(A,B,c,22,ch8), atm(A,D,c,22,ch8), bond(A,D,B,7), indV(A,1.0), (-0,128<=ch8<=-0,106)	85

Tablo 2.2'de sunulan aday kavram tanımlarının kavram tanımları olarak alınabilmesi destek değeri ve güven değeri olarak adlandırılan ölçüm değerlerinin her biri için belirlenen eşik değerlerini sağlaması gerekir [41].

Tablo 2.2'de bulunan Mutagenicity veri kümesi için destek ve güven değerlerinin hesaplanması sonucu eşik değerlerini sağlayan adaylardan kavram tanımları oluşturulmaktadır. Tablo 2.2 için ortaya çıkan kavram tanımları ile destek ve güven Tablo 2.3'de verilmiştir.

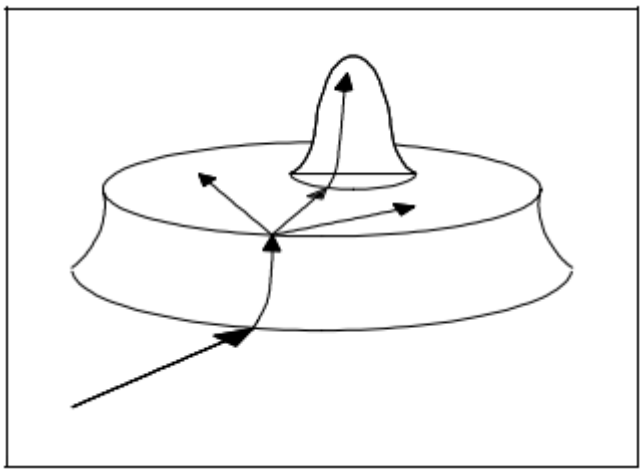
Tablo 2.3. Mutagenicity veri kümesi kavram tanımları örneği

Kurallar	**Destek Değeri**	**Güven Değeri**
atm(A,B,h,3,ch15), atm(A,D,c,22,ch8), bond(A,D,B,1), inda(A,0), indV(A,1), (-0,122<=ch15<=-0,144), (-0,128<=ch8<=-0,106)	0,408	1,0
atm(A,B,c,22,ch8), atm(A,D,c,22,ch8), bond(A,D,B,7), indV(A,1.0)	0,68	1,0

Tablo 2.3'e göre bazı kuralların destek ve güven değerlerini sağlayamaması bu kuralların çözüm kümesine dahil edilmemesi anlamına gelmektedir. Bulunan sonuçlarla kapsam değerini çıkarmak için bakılan açıklanan özellik sayı bu örnek

için toplamda 105 çıkmaktadır. Bu da 125 ilaç örneği için kapsam değerinin 0,84 olması anlamına gelmektedir.

Kavram keşfinde tümevarım mantık programlama ve çizge tabanlı yöntemler kullanılmaktadır. TMP'de yerel plato gibi bazı problemlerin bulunmasına karşın çizge tabanlı yaklaşımda bu tür problemlere çözüm sunulmaktadır. Şekil 2.1'de yerel plato problemi gösterilmiştir.



Şekil 2.1. Yerel plato problemi [8]

Çizge tabanlı yaklaşımda çizge algoritmaları kavram tanımlarını bulmak ve değerlendirmek için kullanılır. Çizge tabanlı kavram keşfinde eş yapı tabanlı ve yol bulma tabanlı iki farklı yaklaşımla çözüm sunmaktadır. Eş yapı tabanlı kavram keşfi yöntemleri ile hedef ilişkileri açıklayan kavram tanımlarının en sık ortaya çıkan yapıları bulunur [39]. Yol bulma tabanlı kavram keşfi yönteminde ise belirli uzunlukta yollar aranır ve hedef örnekleri açıklayan yollar kavram tanımları olarak alınır.

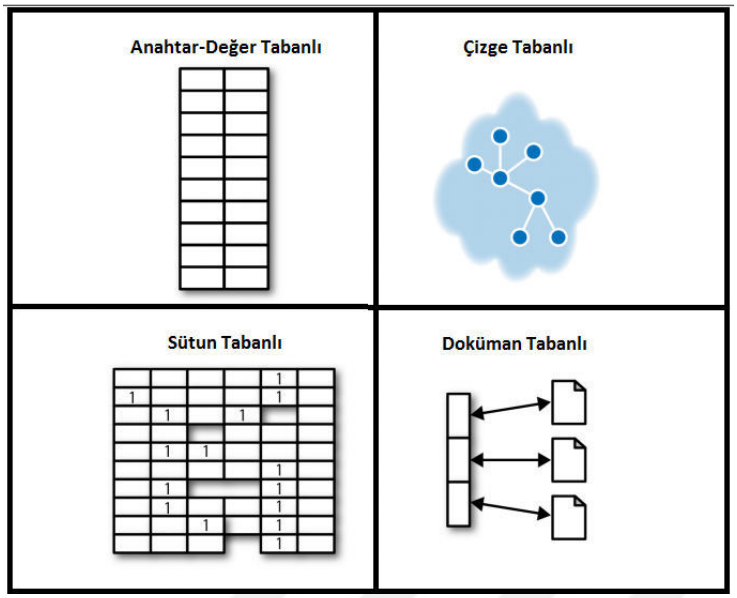
2.2. Çizge Veri Tabanı

Veri tabanı verilerin organizeli bir şekilde tablolar halinde ya da değişik formatlarda saklandığı, birbiriyle ilişkilendirildiği ve isteğe bağlı olarak üzerinde çeşitli sorguların yapılmasına imkân tanıyan bir ortam sunar. Veri tabanı türlerinden birisi olan ilişkisel veri tabanlarında veriler tablolar halinde tutulur. Bu türdeki veri tabanları ACID denilen bölünmezlik (atomicity), tutarlılık (consistency), izolasyon (isolation) ve dayanıklılık (durability) prensipleri üzerine inşa edilir [35]. Bir veri tabanı üzerinde güncelleme, ekleme ya da silme fonksiyonları en temel işlemlerdir.

Veri tabanı türlerinden biri olan ilişkisel veri tabanlarında her yeni bilginin eklenmesi satırlar halinde yapılır. Ancak bu tür veri tabanlarında var olmayan bir alana bilgi ekleme ihtiyacı duyulduğunda yeni bir alanın oluşturulması gerekir. Böyle bir alanı bir tabloya eklemek bazı durumlarda veri tabanlarındaki diğer tabloları da etkileyebilmektedir. Bu problem veri tabanlarında anomali problemi olarak ortaya çıkar. Anomalilerin giderilmesi için veri tabanı üzerinde normalizasyon işlemlerinin yapılması zaruri olmaktadır. Hatta sistemi yeniden tasarlamaya kadar götürebilecek durumlara da yol açabilir ki bu da maliyetin ciddi oranda artması anlamına gelmektedir. İlişkisel veri tabanlarında böyle bir durumla karşılaşmamak için tasarımın öncesinde planlamaların çok iyi yapılması gerekmektedir.

Bu tip problemlere bir çözüm olarak NoSQL veri tabanları ortaya çıkmıştır. NoSQL veri tabanları ilişkisel veri tabanlarının eksik olduğu noktalara alternatif çözüm önerileri sunmaktadır. Verilerin saklanabilmesi ve sorgulanabilmesi bu veri tabanı türünde oldukça kolaydır. Özellikle internet ile veri yoğunluğunun hızla artması sonucu ilişkisel veri tabanları üzerinde değişiklik yapmanın zorluğu NoSQL veri tabanlarını maliyet ve performans açısından çok daha avantajlı kılar. Bunun sebebi yeni bir alan eklemek için NoSQL veri tabanlarında tabloların yerine dinamik yapısal bileşenlerin kullanılmasıdır. NoSQL alan açma işlemini ilişkisel veri tabanlarının aksine doğrudan kendisi yapar. Bu veri tabanı türünün getirdiği en önemli avantajlardan birisi de verilerin yatay olarak ölçeklenebilmesidir. Yatay ölçekleme ile her bir bileşenin aralarında haberleşmesi sonucu ortaya çıkan toplam hesaplama yükü azaltılır ve performans etkinliği arttırılır [42].

NoSQL veri tabanları anahtar-değer tabanlı, doküman tabanlı, sütun tabanlı ve çizge tabanlı veri tabanı olmak üzere farklı türlere ayrılmaktadır. Şekil 2.2'de bu türlerin temsili bir gösterimi bulunmaktadır.



Şekil 2.2. NoSQL veri tabanı türleri [52]

Anahtar-değer tabanlı veri tabanları depolamayı map tablosu kullanarak anahtar değer eşleşmesi ile yapar. Map sayesinde veri hızlı bir şekilde aranır ancak veriler karmaşıklaştıkça bu türden bazı problemler çıkabilmektedir. Genellikle loglama ve arşivleme benzeri uygulamalarda kullanılır. Redis [43] ve Riak [44] anahtar-değer tabanlı veri tabanı türlerindendir. Doküman tabanlı veri tabanı türünde ise doküman kullanarak veri tutulur ve anahtar-değer depolamasına benzer bir yapıya sahiptir. Sorgulamanın daha verimli olduğu bu veri tabanı tipi genellikle web uygulamalarında kullanılır. CouchDB [45], MongoDB [46], OrientDB [47] bu tür veri tabanı sistemlerine örnek olarak gösterilebilir. Sütun tabanlı veri tabanlarında veri sütunlar halinde depolanır. Bu yönüyle yapısal anlamda SQL veri tabanlarına benzemektedir. Genellikle büyük verilerde dağıtık bir yapı ile kullanılır ve bu da veri tabanı işlemlerinde performansın artmasını sağlar. Cassandra [48] ve Hbase [49] bu tür veri tabanları için en bilinen örneklerdendir.

Çizge tabanlı veri tabanı sistemleri veriler ve arasındaki ilişkileri çizge teorisi temelinde düğüm ve kenar yapısı kullanarak depolayan veri tabanı çeşididir. Çizge teorisi bir problemin düğüm ve kenar ile modellenmesi ve bu modelin bir çizge yardımıyla gösterilmesi ilkesine dayanır. Bu teoriye göre çizge olarak modellenen bir problem, bu araçlarla çözülür ve daha sonra gerçek dünyaya uygulanır [50]. Çizgenin tanımı Denklem (2.1)'deki gibi yapılabilir;

$$G = (V, E) \tag{2.1}$$

Bu tanım çizgenin düğümler (V) ve kenarlardan (E) oluşan küme ile temsil edildiğini göstermektedir [51]. Çizgeler yönlü ya da yönsüz ya da ağırlıklı kenar ya da ağırlıksız kenar olarak sınıflandırılabilmektedir.

Çizge veri tabanları ile karmaşık ve büyük veriler kolaylıkla modellenir ve veride zamanla oluşan değişimlere kolaylıkla uyum sağlanır. Böylece sorgulamalar daha etkin bir şekilde yapılır. Hemen hemen her alanda kullanılabilen çizge veri tabanlarının uygulama alanlarından bazıları coğrafi bilgi sistemleri, tavsiye sistemleri, sosyal ağlar ve ulaşım ağlarıdır. Birçok çizge veri tabanı çeşidi bulunmaktadır. Şekil 2.3'de çizge veri tabanlarının çeşitlerini gösteren temsili bir resim bulunmaktadır.

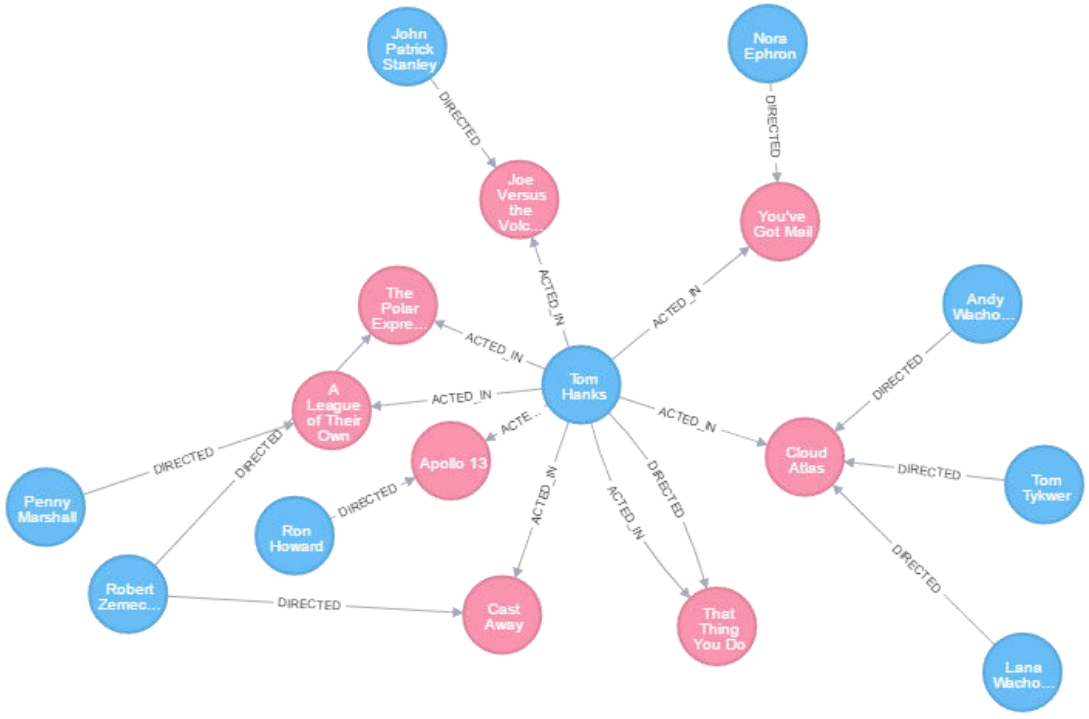


Şekil 2.3. Çizge veri tabanı çeşitlerine genel bakış [52]

Çizge veri tabanı türlerinden AllegroGraph [54] web uygulamaları geliştirmek amacıyla oluşturulmuştur. Bu veri tabanında SPARQL diliyle sorgulanan veriler özne, nesne ve yüklem üçlüleri şeklinde depolanır. Bir diğeri çizge veri tabanı olan Infinite Graph [56] Objectivity firması tarafından geliştirilen hem yerel bilgisayarlar hem de dağıtık mimariler üzerinden depolama imkânı sağlayan bir çizge veri tabanıdır. FlockDB ise [53] twitter için geliştirilmiş sosyal ağ çizgelerini saklayan çizge veri tabanıdır. Özel bir amaçla tasarlandığı için basit bir yapıya sahip olmasının yanı sıra diğer veri tabanlarından en önemli farkı giren ve çıkan ilişkilerin ayrı ayrı tutulmasıdır. Yine HyperGraphDB [57] bilgi yönetimi, yapay zekâ ve semantik web projeleri için tasarlanmış açık kaynak çizge veri tabanıdır. Verilerin bir kenarın birden çok düğüme bağlanabildiği hipergraf yapısı ile modellenmesine imkân tanır.

Trinity de [58] microsoft tarafından geliştirilmiş verilerin anahtar değer ilişki şeklinde saklanabildiği bulut üzerinde çalışan veri tabanı modelidir. Dex [59] büyük ağların hızlı analizine izin veren, C++, Java ve Python'la uyumlu, mobil platformlarda kullanılabilen çizge veri tabanı modelidir. Düşük maliyetle sorgulama ve analiz imkânı sağlar. Son olarak OrientDB [47] NoSQL veri tabanı türlerinden doküman tabanlı veri tabanları ile çizge veri tabanlarının özelliklerini taşıyan açık kaynak kodlu veri tabanı türüdür.

Bu çalışmada sunduğu kolaylıklar sebebiyle tercih edilen çizge veri tabanı türü Neo4j olmuştur. Neo4j Neo Technology tarafından geliştirilen ölçeklenebilir, hızlı, basit ve açık kaynak kodlu bir çizge veri tabanı türüdür. Java ile geliştirilmiş olmasına rağmen rest api sayesinde farklı dillerde kullanılabilen Neo4j aynı zamanda SQL sorgu diline benzer Cypher adında basit ve güçlü bir sorgulama diline sahiptir. Bu dil ile Neo4j'de karmaşık sorgular kolay ve etkin bir şekilde yapılabilmektedir. Neo4j diğer çizge veri tabanlarından farklı olarak ilişkisel veri tabanlarında uygulanan ACID'i de destekleyerek veri güvenliğini sağlar. Neo4j'de veriler çizge yapısında oluşturulduğundan düğüm ve kenarlar ile organize edilir. Şekil 2.4'de Neo4j ile oluşturulmuş bir çizge örneği gösterilmektedir.



Şekil 2.4. Neo4j ile oluşturulmuş çizge örneği

Bu veri tabanı ile veri modellemesi yapılırken Cypher sorgu dili kullanılarak büyük verilerin Neo4j'ye kolaylıkla aktarılması csv formatı üzerinden gerçekleştirilebilmektedir. Neo4j'ye aktarılan veriler belirlenen düğüm, ilişki, etiket ve özelliklerin tespit edilmesi ile çizgeye dönüştürülür. Çizgenin oluşması ile verinin analizi ya da veriden elde edilmek istenen bilgilerin çıkarılması süreci başlar. Bu aşamada da veri modellemesinde olduğu gibi Cypher dili kullanılabilmektedir. Lokal olarak kullanılabilen Cypher dilinin aynı zamanda gömülü olarak başka platformlarda kullanılabilmesi tercih edilen programlama dilinin sunduğu algoritmik kolaylıklardan da yararlanabilmeyi sağlar [55].

3. YÖNTEM: ÇİZGE VERİTABANI İLE KAVRAM KEŞFİ

Çalışmanın bu kısmında çizge tabanlı kavram keşfi için geliştirilen yöntem anlatılmaktadır. Bu yöntem çizgenin modellenmesi ve kavram tanımlarının bulunması aşamalarından oluşmaktadır.

İlk aşamada ilişkisel veri tabanında bulunan verileri çizgeye dönüştürmek için bazı ön işlemlerden geçirilmiştir. Sayısal verilerin ayrıklaştırılması ve çizge için gerekli düğüm, ilişki ve bunlara ait özelliklere uygun dosya formatlarında oluşturulması bu aşamanın adımlarını içine alır. İkinci aşamada ise hedef ilişkiler, arka plan bilgileri, minimum destek ve minimum güven değerleri kullanılmıştır. Arka plan bilgileri kullanılarak hedef ilişkiler açıklanır. Minimum destek ve güven değerleri ile de kavram tanımları elde edilerek kavram keşfi tamamlanır. Şekil 3.1'de geliştirilen yöntemin genel kodu verilmiştir.

```
Girdi: D: Düğümler H: Hedef Örnekler min_des: min destek değeri min_guv: min güven değeri
Çıktı: K: Kavram Tanımları
hedefOrnekleriListesi = Ø
cozumuOlanYolListesi = Ø
cozumOlmayanYolListesi = Ø
sonucKumesi = Ø
yolSayisi ← 0
for each Ni in N do
    hedefOrnekler = hedefOrnekleriBul(Ni)
    add hedefOrnekler to hedefOrneklerListesi
end for
for each Ti in T do
    yol = yolBul(Ti)
    genellestirilmisYol = genellestirilmisYolBul(yol)
    if genellestirilmisYol in cozumuOlanYolListesi then
        continue
    else if genellestirilmisYol in cozumuOlmayanYolListesi then
        continue
    else
        destekDegeri = destekDegeriHesapla(genellestirilmisYol)
        if destekDegeri < min_des then
            add genellestirilmisYol to cozumOlmayanYolListesi
        else
            add genellestirilmisYol and destekDegeri to cozumuOlanYolListesi
            guvenDegeri = guvenDegeriHesapla(genellestirilmisYol)
            if guvenDegeri > min_guv then
                fDeger = fdegeriBul(genellestirilmisYol)
                enYuksekFdeger = fDeger
                if enYuksekFdeger = fDeger then
                    add genellestirilmisYol,destekDegeri,guvenDegeri,fDeger to sonucKumesi
                    yolSayisi = yolSayisi + 1
                    kapsamDegeri +=  kapsamDegeriHesaplama(genellestirilmisYol)
                end if
            end if
        end if
    end if
end for
return sonucKumesi
```

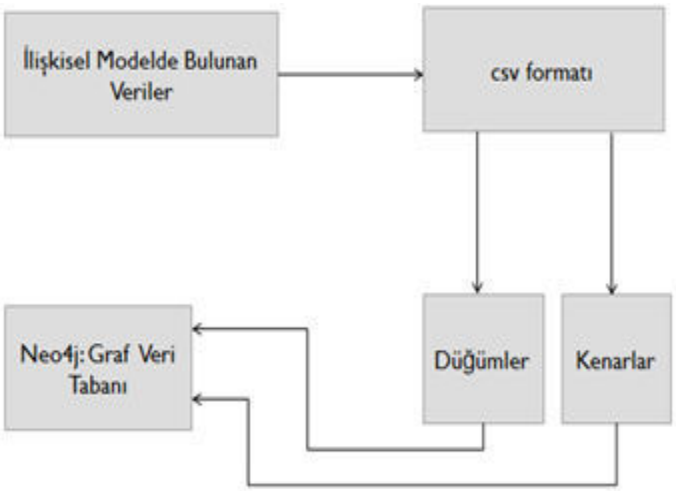
Şekil 3.1. Yöntemin genel kodu

3.1. Verilerin Modellemesi

Veri kümeleri genellikle prolog da ilişkisel modelde bulunur. Bu tür veri kümelerin çizge modeli olarak kullanılabilmesi için dönüştürülmesi gerekir. Bunun için öncelikle veriler arasındaki ilişki türleri ortaya konur. Bu biçimlerden biri olan tek argümanlı ilişkiler veri kümesinde herhangi bir argümanın bir özelliği olarak tanımlanır ve bir düğüm ile bu düğümün kendine dönen bir ilişkisi olarak gösterilir. İki argüman arasındaki ilişkiyi temsil eden iki argümanlı ilişkiler iki düğüm ve bu iki düğümün birbirine bağlandığı bir ilişkiden oluşmaktadır. n-argümanlı ilişkilerde ise iki varlığın bilgileri ve bunların arasındaki ilişkinin özellikleri tutulur.

Verilerin yapısı analiz edildikten sonra verilerin çizge modeline aktarılabilmesi için doğru bir şekilde dosyaların tasarlanması gerekmektedir. Bu dosyalardan düğüm, ilişki ve bunların özellliklerini içeren bilgiler okunarak çizgeye dönüşüm gerçekleştirilir.

Veriler csv formatında java programlama dili kullanılarak Neo4j graf veri tabanına dönüştürülmüştür. Şekil 3.2'de bu aşamanın akış diyagramı verilmiştir.



Şekil 3.2. Verilerin modellenmesi

3.1.1. İki argümanlı ilişkilerin modellenmesi

İki argümanlı ilişkilerde veriler arasındaki ilişki yönlü ya da yönsüz olabilir. Bir ilişkinin yönlü olması o ilişkinin tek taraflı geçerli olduğunu, yönsüz olması ise verilen ilişkinin çift taraflı geçerli olduğunu göstermektedir. Bu çalışmada kullanılan veri kümelerinde varlıklar arasındaki ilişkiler yönlü olup aynı ilişkiyi içeren argümanların açıklanması için iki farklı ilişki kullanılır. Kavram keşfi için iki argümanlı ilişkilerden oluşan veri kümelerinde açıklanmak istenen hedef örnekleri Tablo 3.1'de gösterilmiştir.

Tablo 3.1. İki argümanlı verilerin hedef örnekleri

Veri Kümesi	**Hedef**	**Hedef Örneği**
Elti	Elti	elti(kişi,kişi)
Dünür	Dünür	dunur(kişi,kişi)
Kinship	Baba	father(kişi,kişi)
	Anne	mother(kişi,kişi)

Tablo 3.1.(Devam) İki argümanlı verilerin hedef örnekleri

Veri Kümesi	**Hedef**	**Hedef Örneği**
Kinship	Eş (erkek)	husband(kişi,kişi)
	Eş (kadın)	wife(kişi,kişi)
	Kardeş (kız)	sister(kişi,kişi)
	Kardeş (erkek)	brother(kişi,kişi)
	Amca, Dayı, Enişte	uncle(kişi,kişi)
	Hala, Teyze, Yenge	aunt(kişi,kişi)
	Çocuk (erkek)	son(kişi,kişi)
	Çocuk (kız)	daughter(kişi,kişi)
	Yeğen (erkek)	nephew(kişi,kişi)
	Yeğen (kız)	niece(kişi,kişi)
Same Generation	Gen	gen(kişi,kişi)

Veri kümelerinin çizge ile gösterilmesi için düğüm ve kenarların tespit edilmesi gerekmektedir. Çalışmada kullanılan Kinship veri kümesi için düğüm, hedef örneklerini ve arka plan bilgilerini içeren kişileri kenar ise iki kişi arasındaki bağlantıyı tanımlayan ilişkileri ifade eder. Kinship'te açıklanmak istenen ilişki son olduğunda veriler arasındaki tüm "son" ilişkileri hedef örnekler olurken diğer ilişkileri içeren veriler arka plan bilgileri olmaktadır. Tablo 3.2'de son için hedef örnekleri ve arka plan bilgileri gösterilmiştir.

Tablo 3.2. son için hedef örnekleri ve arka plan bilgileri

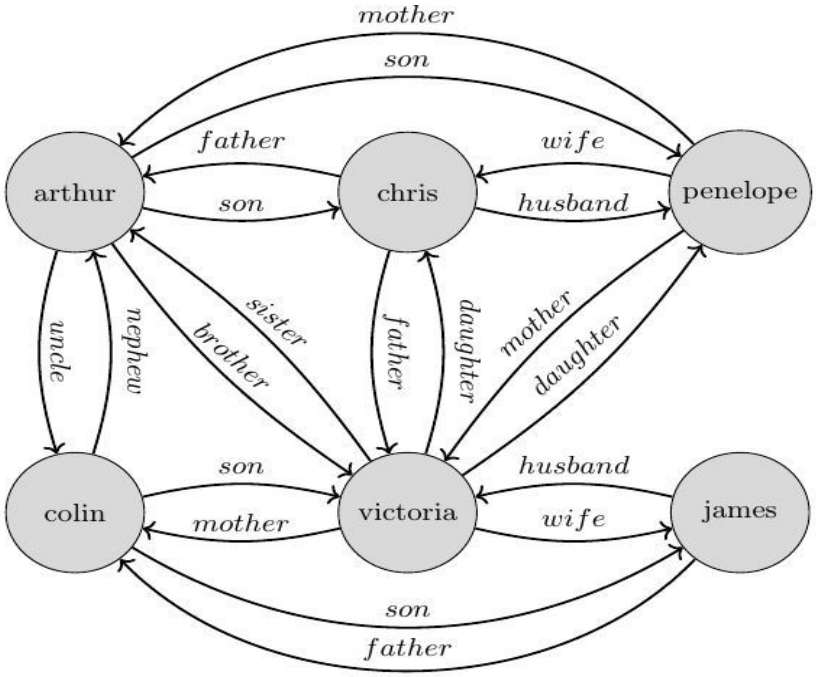
Hedef Örnekleri	**Arka Plan Bilgileri**
son(arthur,chris)	father(kişi,kişi)
son(arthur,penelope)	mother(kişi,kişi)
son(james,andrew)	husband(kişi,kişi)
son(james,christine)	wife(kişi,kişi)
son(colin,victoria)	sister(kişi,kişi)
son(colin,james)	brother(kişi,kişi)
son(emilio,roberto)	uncle(kişi,kişi)
son(emilio,maria)	aunt(kişi,kişi)
son(marco,pierro)	son(kişi,kişi)
son(marco,francesca)	daughter(kişi,kişi)
son(alfonso,lucia)	nephew(kişi,kişi)
son(alfonso,marco)	niece(kişi,kişi)

Tablo 3.2'de "son" ilişkisini içeren 12 hedef örneği görülmektedir. Ayrıca Tablo 3.3'de Arthur ile Penelope arasındaki örnek arka plan bilgileri gösterilmiştir.

Tablo 3.3. Hedef örnekleri ve arka plan bilgileri

Hedef Örnek	Arka Plan Bilgileri
son(arthur,penelope)	father(chris,arthur)
	husband(chris,penelope)
	wife(penelope,chris)
	uncle(arthur,colin)
	nephew(colin, arthur)
	sister (victoria, arthur)
	brother(arthur, victoria)
	father(chris,victoria)
	daughter(victoria, chris)
	daughter(victoria, penelope)
	mother(penelope, victoria)

Elti, Kinship ve Same Generation gibi iki argümanlı ilişkileri içeren veri kümeleri Java'da Neo4j aracılığıyla csv formatından çizge veri tabanı haline getirilmiştir. Şekil 3.3'de örnek bir çizge gösterilmiştir.



Şekil 3.3. Kinship veri kümesi için örnek çizge

Şekil 3.4'de Kinship veri kümesi için Neo4j'de çizge verilmiştir. Buna göre Kinship veri kümesinin iki bağımsız çizge içerdiği görülmektedir.



Şekil 3.4. Kinship veri kümesinin Neo4j'de örnek bir gösterim

3.1.2. N-argümanlı ilişkilerin modellenmesi

N-argümanlı ilişkilerde veriler iki argümandan daha fazla bağlantının olabileceği ilişkilere sahiptir. Bu tür ilişkilerde iki argüman arasındaki bağlantı ve bu bağlantı sonucu ortaya çıkan bazı özellikler bulunur. N-argümanlı ilişkilerin modellenmesinde bazı ön işlemler gerekmektedir. Tablo 3.4'te bu çalışmada yer alan n-argümanlı verilerin hedef örnekleri gösterilmiştir.

Tablo 3.4. N-argümanlı verilerin hedef örnekleri

Veri Kümesi	**Hedef**	**Hedef Örneği**
Eastbound	Doğu Yönlü Giden Araba	eastbound(araba)
Mutagenicity	Mutajenik Bileşik	muta(A,doğru)
PTE	Kanserojen İlaç	pte(A,doğru)

N-argümanlı ilişkilerde sayısal ve bazı sabitler bulunabilmektedir. Bu bilgilerin ayrıklaştırılması ve bilgi kaybı olmadan doğru şekilde ifade edilebilmesi önem taşımaktadır. Bunun için çok sayıda veri ayrıştırma yöntemleri bulunmaktadır. Gruplama yöntemi bu yöntemlerden birisidir. Bu yöntemde amaç sayısal verilerin daha anlamlı hale getirilmesidir.

Gruplama yöntemi bir özelliğin sayısal değerleri arasındaki eşsiz değerlerin tespit edilmesi ile başlar. Bu değerler küçükten büyüğe veya tersi yönde sıralanır. Daha sonra hedef veri kümesine göre veriler eşit aralıklara bölünür. Burada bölünme miktarı en küçük destek değerlerini sağlayacak şekilde belirlenmiştir. Çalışmada kullanılan Mutagenicity veri kümesindeki logp, lumo, inda, ind1 değerleri ile PTE ve Mutagenicity veri kümelerinde ortak bulunan charge değerleri sayısal veriler içeren değerlerdir.

Bunlardan inda ve ind1 özellikleri iki değere (0 ya da 1) sahiptir. Bu durum var ya da yok şeklinde sadece iki durumu ifade ettiği için herhangi bir ayrıklaştırma işlemine gerek duyulmamıştır.

Veri kümelerinde bulunan logp, lumo ve charge özellikleri ise ondalıklı değerler içermektedir. Bu özelliklerin karşıladığı değerler için veri kümelerinin destek değerlerine göre ayrıklaştırma işlemi yapılmıştır. Mutagenicity veri kümesinde 230 ilaç bulunduğundan 23, PTE veri kümesinde ise 340 ilaç bulunduğundan aynı teknikle 34 aralıklı parçalara bölünmüştür. Mutagenicity veri kümesi için lumo özelliğinin ayrıklaştırma sonucu Tablo 3.5'de gösterilmiştir.

Tablo 3.5. Lumo özelliğinin ayrıklaştırma sonucu

Lumo adı	**En küçük değeri**	**En büyük değeri**
lumo1	-3,768	-2,294
lumo2	-2,292	-2,138
lumo3	-2,113	-1,845
lumo4	-1,834	-1,698
lumo5	-1,691	-1,585
lumo6	-1,57	-1,447
lumo7	-1,443	-1,293
lumo8	-1,289	-1,144
lumo9	-1,125	-0,868
lumo10	-0,857	-0,529

Tablo 3.5'ten 10 farklı lumo değerinin oluştuğu gözlemlenmektedir. Bu şekliyle çizgede 10 düğüm oluşturulduğunda bütün lumo değerleri temsil edilmiş olmaktadır. Oluşan yeni lumo düğümlerinin en küçük ve en büyük değerleri onun özellikleri olarak saklanır. Böylece herhangi bir kayıp olmadan istenebilecek herhangi bir aralık

değerine erişim sağlanabilmektedir. Tablo 3.6'da diğer özelliklerin aynı yöntem kullanılarak ayrıklaştırılması sonucu oluşan düğüm sayıları verilmiştir.

Tablo 3.6. Özelliklerin sınıflandırma sayıları

Özellik	**Adet**
Mutagenicity logp	6
Mutagenicity lumo	10
Mutagenicity charge	23
PTE charge	32

N-argümanlı ilişkileri çizgede temsil etmek iki argümanlılardan daha zordur. Bu konuyla ilgilenen hipergraf olarak adlandırılan çizge yönteminde bir kenar çok sayıda düğüme bağlanır. Hipergrafları normal çizge ile temsil etmek mümkündür. Hipergraf içeren düğümlerin arasındaki ilişkileri ikili hale getirerek bu sağlanabilir. Ancak böyle bir durumda bazen bilgi kaybı ortaya çıkabilmektedir. Bunun için fazladan düğüm oluşturarak bu problemin çözümü sağlanabilir [60].

Hipergraf yapı, Eastbound veri kümesinde olduğu gibi tek ya da Mutagenicity ve PTE veri kümesindeki gibi çoklu ilişkiye sahip olabilir. Tekli ilişkiler içeren Eastbound veri kümesinde closed, jagged, open_car, tr_double, tr_long, tr_short özellikleri bulunur. Bu özellikler ilgili verinin kendisi ile ilişkili olduğu gösterilerek düğümün kendisine dönmesi ile oluşturulur.

Mutagenicity veri kümesinde muta_atm özelliği ile PTE veri kümesinde pte_atm özelliği çoklu ilişkiye sahiptir. Tablo 3.7'de bu özelliklerin bir kaç örneği verilmiştir.

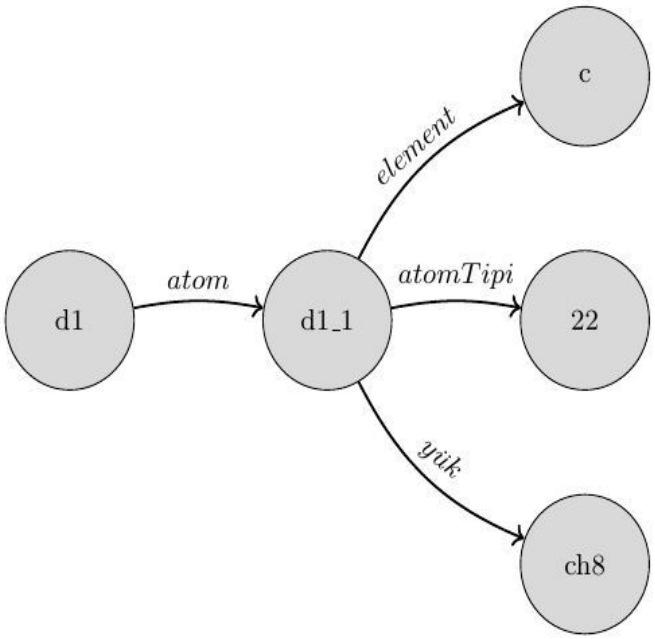
Tablo 3.7. Çok ilişkiye ait örnekler

Muta	**pte**
muta_atm(d1,d1_1,c,22, -0,117)	pte_atm(d1,d1_1,c,22, -0,133)
muta_atm(d102,d102_7,h,3, 0,141)	pte_atm(d121,d121_24,h,3, 0,137)
muta_atm(d156,d156_19,o,40, -0,405)	pte_atm(d152,d152_14,c,14, 0,617)
muta_bond(d1,d1_11,d1_17,7)	pte_bond(d1,d1_14,d1_22,2)
muta_bond(d12,d12_2,d12_8,1)	pte_bond(d1,d10_2,d10_8,1)

Tablo 3.7'de görülen örneklerden Mutagenicity veri kümesi için muta_atm özelliğini içeren birinci örneğe bakıldığında beş argümana sahip olduğu görülmektedir. Ancak

hipergraf yapısına uyan bu yapıyı modelleyebilmek için öncelikle beş argümanlı bu ilişkinin neyi ifade ettiğini anlamak gerekmektedir.

Birinci parametrede d1 ilacı görülmektedir. İkinci parametrede ise birinci parametrede bulunan d1 ilacını oluşturan atomlardan biri olan d1_1 atomu bulunmaktadır. Bu atom, d1 ilacının bir yapı taşı olduğundan dolayı ilaç ile atom arasında tek yönlü bir ilişki kurulabilir. Diğer üç parametrede ise c elementini, atom tipi 22'yi ve 0,117 yük değeri görülmektedir. Bu üç özellik d1 ilacının, d1_1 atomu ile olan ilişkisi sonucu ortaya çıkan değerlerdir. Bu sebeple d1_1 atomunun bu üç değerle ayrı ayrı ilişkisi var olduğu söylenebilir. Yük değeri yapılan ayrıklaştırma yönteminden dolayı ch8 sınıfına girmektedir. Şekil 3.5'de bu örneğin çizgesi görülmektedir.

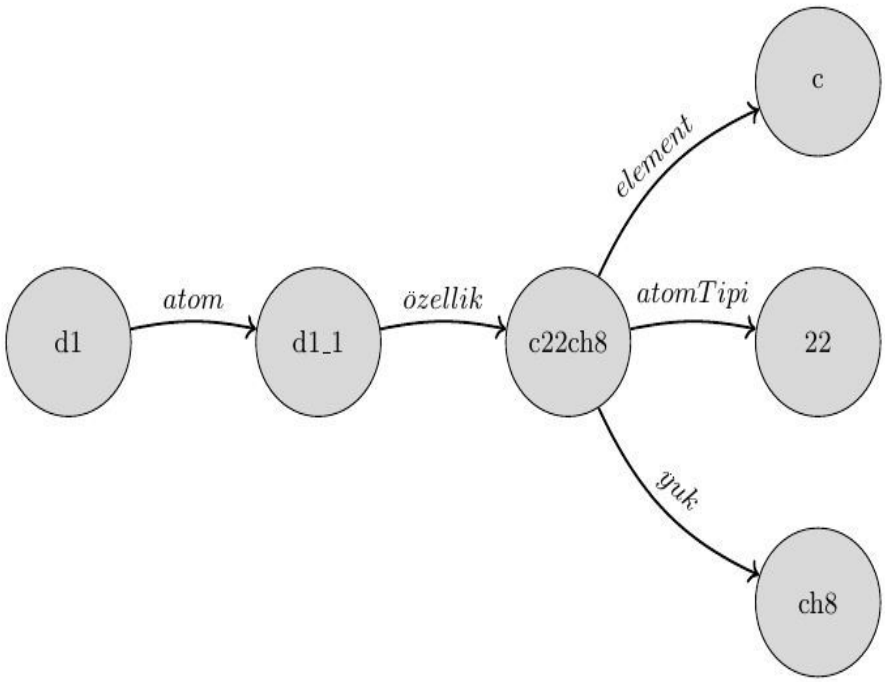


Şekil 3.5. muta_atm çizge örneği gösterimi

Şekil 3.5'te gösterilen beş argümanlı ilişki bir ilacın aynı atomla birden fazla bağlantısı olduğu durumlarda sorun çıkartabilir. Bu o ilişkinin yük, element ve atom tipi parametrelerinde değişiklik gösterebilmesinden kaynaklanmaktadır. Belirtilen üç özellikten birisi değişse bile aynı atoma farklı değerlerle bir ilişki kurulacağından hangi bağlantıdan hangi ilişkinin olduğunu tespiti zor olacaktır.

Aynı zamanda element, atom tipi ve yük üçlüsü tek bir ilişkiden meydana geldiği için bu üçlünün ortak şekilde temsil edilmesi yol bulma algoritmasında kolaylık

sağlayacaktır. Bu sebeplerden ötürü bu yöntemde yeni bir düğüm oluşturup araya bir katman koyulmuştur. Böylece bir durumda temsil edilmek istenen özellik ve ilişkiler bilgi kaybı olmadan doğru şekilde kullanılmaktadır. Şekil 3.6'de çözümün gösterimi bulunmaktadır.

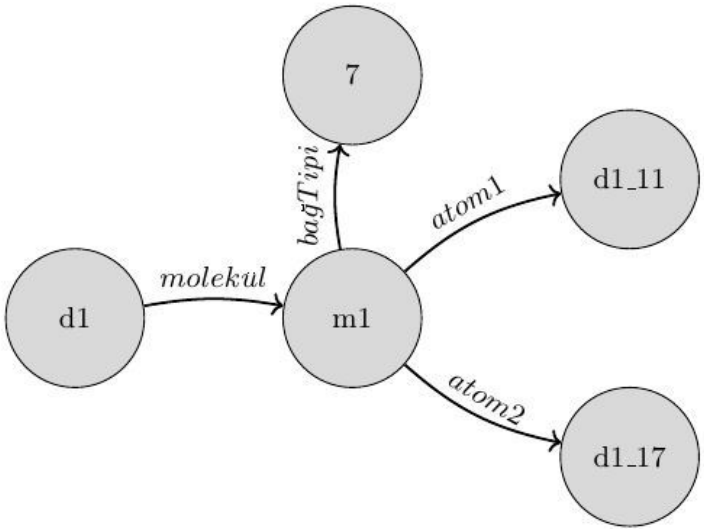


Şekil 3.6. muta_atm çizge gösterimi

Tablo 3.7'de muta_bond örneğine bakıldığında ise dört parametre görülmektedir. Burada d1 ilacının ilk parametre olduğu, ikinci ve üçüncü parametrede ise d1_11 ve d1_17 gibi iki farklı atomun bulunduğu görülmektedir. Dördünce parametrede ise bağ tipi özelliği (7) bulunmaktadır.

Burada d1 ilacının d1_11 ve d1_17 atomları ile bir bağlantısı bulunmaktadır. Aynı şekilde d1_11 ile d1_17 arasında bir bağlantı görülmekte ve bu bağın özelliği olarak da 7 bağ tipi olarak oluşmaktadır.

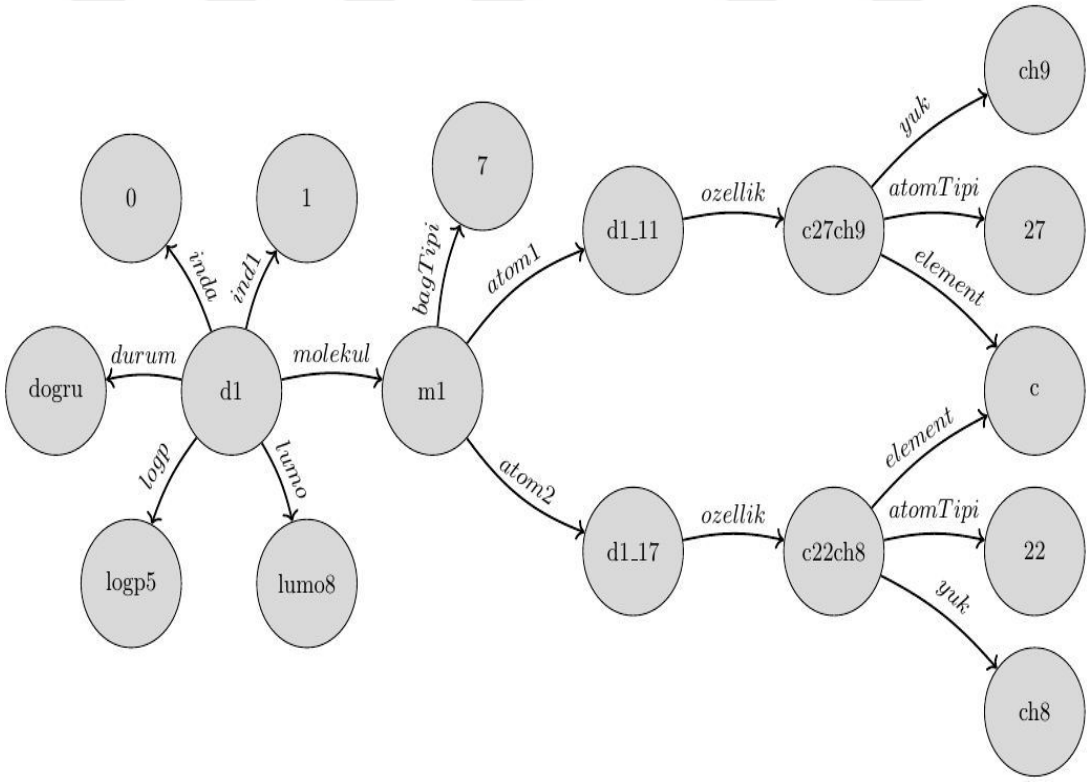
d1 ilacının birden fazla bağlantısı olduğu görülmektedir. Bu sebeple bu özelliklerin ve bağın anlamlı kalabilmesi için d1 ilacı ile d1 ilacının bağlantısı olan atomların arasında bir katman koymak mantıklıdır. Bu katman d1_11 ile d1_17 arasındaki bağı temsil ettiğinden bağ tipinin bu katman ile bir bağlantı kurması gerekir. Şekil 3.7'de bu çözümün gösterimi bulunmaktadır.



Şekil 3.7. muta_bond çizge örneği gösterimi

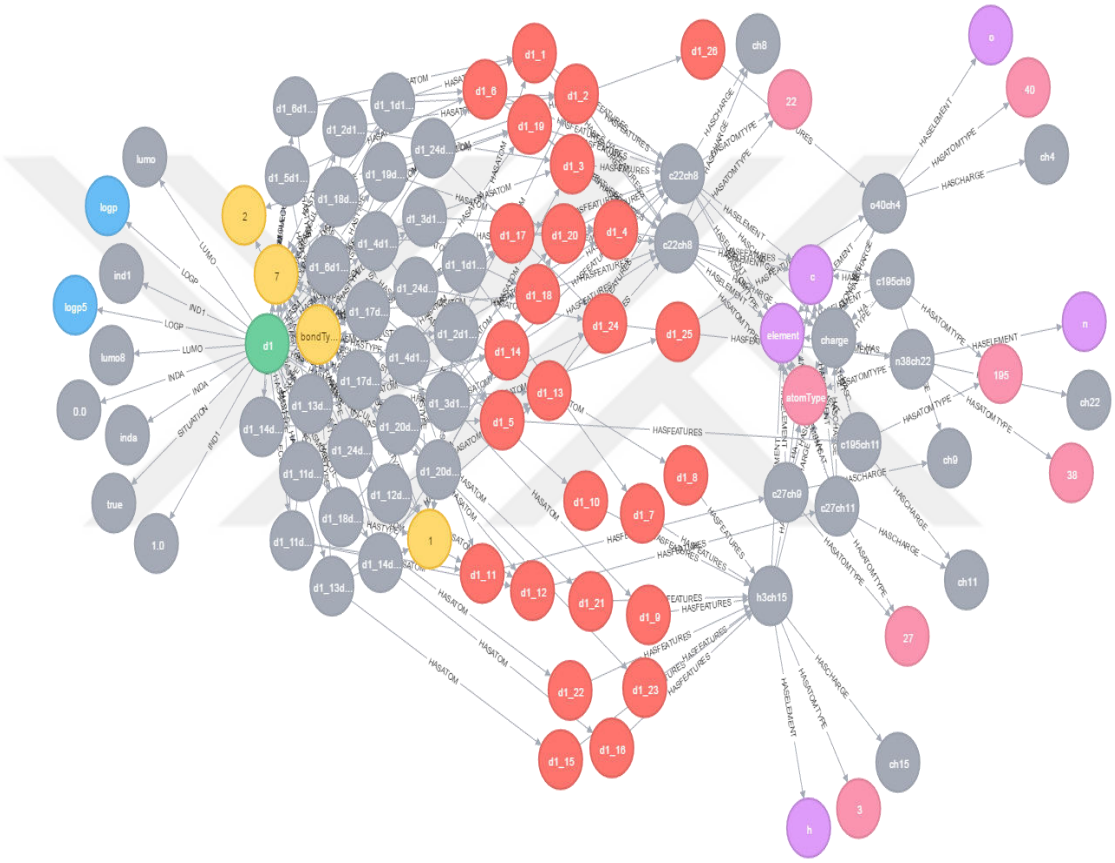
PTE veri kümesi için pte_atm muta_atm ile ve pte_bond ise muta_bond ile aynı yöntem uygulanarak modellenmiştir.

d1 ilacı Şekil 3.5'de gösterilen m1 gibi 28 bağdan oluşmaktadır. Şekil 3.8'da muta veri kümesinin d1 ilacı için bir alt çizge gösterimi bulunmaktadır.



Şekil 3.8. d1'in alt çizge model örneği

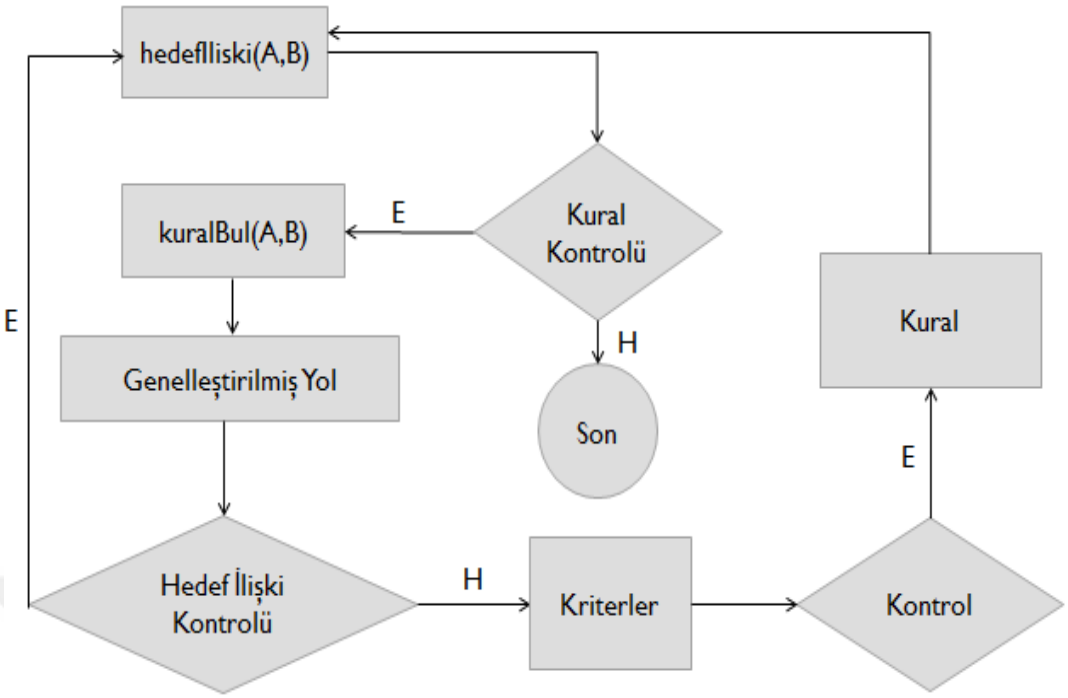
Sonuç olarak n-argümanlı ilişkiler için öncelikle ayrıklaştırılması gereken sayısal veriler sınıflandırılmıştır. Daha sonra tekli ilişkiler kendi içinde modellenmiş ve ikili ilişkiler kurulmuştur. N-argümanlı ilişki içeren özellikler için de ayrı ayrı ilişkiler belirlenmiştir. Düğüm ve kenarlar csv formatına dönüştürüldükten sonra Java'da çizge veri tabanına modellemesi gerçekleştirilmiştir. Şekil 3.9'da Neo4j'nin ara yüzünde Mutagenicity veri kümesinin d1 ilacı için oluşturulmuş bir çizge örneği verilmiştir.



Şekil 3.9. Mutagenicity d1 ilacı için Neo4j'de örnek alt çizge

3.2. Aday Kavram Tanımlarının Bulunması ve Değerlendirilmesi

Adayların bulunması ve bulunan adayların belirlenen kriterlerle değerlendirilmesi sonucu ayıklanması ve nihai kavram tanımlarının tespit aşamasıdır. Bu aşamada Java'da Neo4j kütüphanesinin sunduğu özellikler kullanılmıştır. Şekil 3.10'da kavram tanımlarının bulunması ile ilgili akış diyagramı verilmiştir.

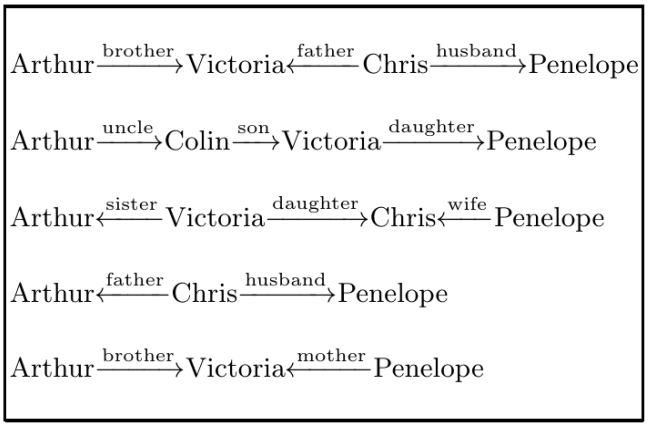


Şekil 3.10. Kavram tanımlarının bulunması ile ilgili akış diyagramı

Aday kavram tanımlarının bulunması için gezinme yöntemi ve yol bulma algoritması birlikte kullanılmıştır. Belirlenen parametrelerle çalışan bu algoritmada öncelikle başlangıç ve bitiş düğümleri belirlenir ve listeye eklenir. Listeden rastgele seçim yapılır.

Başlangıç düğümünden başlayan yöntem bitiş düğümüne ulaşıncaya kadar gezinmeye başlar. Gezinme sığ öncelikli arama ya da derin öncelikli arama ile sağlanabilir. İki algoritmada aynı hesaplama ve arama karmaşıklığına sahip olmasına rağmen çözüm kök düğüme yakın olması beklendiğinde sığ öncelikli arama tercih edilmelidir [61]. Bu sebeple yöntemde kavram keşfinde genellikle en kısa uzunluklu yollar arandığından sığ öncelikli arama tercih edilmiştir.

Şekil 3.3'de son(arthur, penelope) hedef örneği için Arthur başlangıç düğümü Penelope hedef düğümü olmaktadır. Bu hedef düğüme farklı yollardan birden fazla yol izleyerek gidilebilmektedir. Şekil 3.11'de gidilebilecek bazı örnek yollar gösterilmiştir.



Şekil 3.11. Kinship için örnek yollar

Şekil 3.8'de görülen muta(d1,dogru) düğümleri hedef örneğidir. Ancak doğru düğümüne d1 dışında herhangi bir düğümden ulaşılamamaktadır. Bu sebeple yol bulma yöntemi hedef düğümün diğer düğümlerle ulaşılamadığı bu tip veri kümeleri için değişiklik göstermektedir. Bu sebeple bu aşamada yöntem hedefe doğrudan yol olan ve olmayan şeklinde iki farklı yöntemle ilerlemektedir.

3.2.1. Hedefe farklı yollardan ulaşılabilen çizgede yol bulma

Hedef düğüme farklı yollardan ulaşılabildiği Elti, Kinship, Gen ve Eastbound veri kümelerinde uygulanan yöntem bu başlık altında anlatılmaktadır. Yol bulmada kullanılan algoritmada ilk olarak hedef örneklerinden başlangıç ve bitiş düğümleri bir listeye eklenir. Listeden düğümler seçilerek belirlenen bazı parametreler kullanılarak arama işlemi başlar.

Bu parametrelerden derinlik kural uzunluğunun belirlenmesi için kullanılır. Burada kavram tanımlarına ulaşmak için hedefe düğüme en az iki düğümde ulaşılabileceği için derinlik 2 olarak belirlenmiştir. 2 uzunluklu kurallarla hedefe ulaşılamadığında derinlik arttırılmaktadır. Bu parametrenin kullanılma amacı bir kuralı en iyi yollarla açıklamaktır.

Düğümlerin birbirleriyle kurabileceği bağlantıların adlarını içeren ilişkiler arama parametresi olarak girilir. Her bir derinlikte bu ilişkiler aranarak düğümler arasındaki bağlantı bulunmuş olur. Arama ilişkilerin yön parametresine bakılarak belirlenir. Giden, gelen ya da her ikisinin de seçilebildiği yön özelliği ile algoritmanın ihtiyaca

göre tek ya da çift yönlü arama yapılabilmesi sağlanır. Şekil 3.12'de yönlerle ilgili örnek yollar gösterilmiştir.

$\text{Victoria} \xleftarrow{\text{father}} \text{Chris} \xleftarrow{\text{wife}} \text{Penelope}$	sol yön: 2 (tek yön)
$\text{Chris} \xrightarrow{\text{father}} \text{Victoria} \xrightarrow{\text{wife}} \text{James}$	sağ yön: 2 (tek yön)
$\text{Arthur} \xrightarrow{\text{son}} \text{Chris} \xleftarrow{\text{wife}} \text{Penelope}$	sağ yön: 1/sol yön: 1 (çift yön)
$\text{Arthur} \xrightarrow{\text{brother}} \text{Victoria} \xrightarrow{\text{daughter}} \text{Chris} \xleftarrow{\text{wife}} \text{Penelope}$	sağ yön: 2/sol yön: 1 (çift yön)

Şekil 3.12. Örnek yollar ve yönleri

Şekil 3.12'de "son" ilişkisinin farklı yönler kullanarak bulunabileceği görülmektedir. Kural aramasını tek yönlü yapmak bu kurallardan bazılarının bulunamamasına sebep olacağından yöntemde çift yönlü kural araması yapılmıştır.

Yöntemde bir başka parametre olan eşsizlik parametresi bulunmaktadır. Bu parametre bir kuralın içerdiği bir düğümün ya da ilişkinin benzersiz olarak kullanılabilmesi sağlanır. Düğümlerin eşsiz olması bulunan bir yol için kullanılan bir düğümün, ilişkilerin eşsiz olması kullanılan ilişkinin bir daha seçilmeyeceği anlamına gelmektedir. Bu parametre veri kümelerinin yapısına bağlı olarak değişmektedir.

Elti, Kinship ve Gen veri kümelerinde tüm ilişkiler ikili olduğu için düğümlerin tekrar etmemesi mantıklıdır. Bu tür veri kümelerinde eşsizlik parametresi olarak düğüm seçilmiştir. İlişki seçildiği bir durumda bir düğümden çıkan yol tekrar kendisine dönebileceği için sonsuz döngü problemine sebep olur.

Eastbound veri kümesi tek ve iki argümanlı ilişkilerden oluşmaktadır. Tekli oluştuğu durumlarda eşsizlik parametresi düğüm seçilirse bazı düğümlerin kendisi ile bağlantı kurması gerektiğinde problem ortaya çıkar. Bu sebeple bu veri kümesinde eşsizlik parametresi olarak ilişki seçilmiştir.

Belirtilen parametre ve yöntemlerle hedef örneklerinden rastgele seçilen başlangıç düğümünden hedef düğümü ulaşınca bir yol bulunmuş olur. İlgili hedef düğümlerden

diğer yollardan aynı şekilde yol bulma işlemi devam eder ve aday kavram tanımları belirlenir.

3.2.2. Hedefe farklı yollardan ulaşılamayan çizgede yol bulma

Hedef düğüme farklı yollardan ulaşılamadığı PTE ve Mutagenicity veri kümelerinde uygulanan yöntem bu başlık altında anlatılmaktadır. Bu tür veri kümelerinde hedef düğüm olarak iki durum söz konusudur ve bu düğümlere başlangıç düğümleri dışında ulaşılamamaktadır. Bundan dolayı yol bulmada kullanılan algoritmada her iki durum içinde düğümler belirlenir ve ayrı listelere eklenir.

Tablo 3.8'de oluşturulan listelerle ilgili bilgiler paylaşılmıştır. Bu listedeki düğümler ile bilgiler güven ve destek değerlerinin hesaplamalarında kullanılmaktadır.

Tablo 3.8. PTE ve Mutagenicity için oluşturulan listeler

Düğüm	**Liste**	**Liste Boyutu**
Mutagenicity İlaç Düğümü	Doğru ilaçlar	125
	Yanlış İlaçlar	63
PTE İlaç Düğümü	Doğru ilaçlar	163
	Yanlış İlaçlar	136

Yön parametresi olarak tek yön seçilmiştir. Buna göre yön, bulunulan düğümden gidilebilecek sonraki düğüme doğrudur. Bu veri kümeleri tam bir yol bulmadan bahsedilemeyeceğinden derinlik parametresi için 1 seçilmiştir. Derinlik bir seçildiği için bir düğümden en fazla bir düğüme gidilebilmektedir. Ulaşılan düğüm için bu, bir sonraki düğüme gidiş anlamına gelmektedir. Böylece birer birer gezinme yöntemi ile yollar birbirine eklenmektedir.

Doğru ilaçlar için bir çözüm arandığında bu yöntem doğru düğümler listesinden rastgele bir düğüm seçmekte ve gezinme yöntemi ile gidebileceği bütün yolları birbirine eklenmektedir. Yol herhangi bir seçenek kalmayana kadar devam eder. Diğer bir ifadeyle herhangi bir düğümün çocuğu yoksa o gezinme tamamlanmış olur.

Herhangi bir düğümde yol ikiye ya da daha fazla parçaya bölünüyorsa bölünen düğüm bilgisi saklanır ve yollardan birisi seçilir. Yol tamamlandığında saklanan düğüm bilgisi ile diğer yoldan devam edilir. Bu şekilde yollar bitene kadar algoritma devam eder. Tüm yollar bittiğinde sonlanan düğüm bilgileri elde edilir ve yollar

bulunmuş olur. Şekil 3.6'da d1 ilacı için verilen alt çizgeye göre algoritmanın tamamlanması için Mutagenicity veri kümesi için ulaşılması gereken düğüm isimleri Tablo 3.9'de verilmiştir.

Tablo 3.9. Gezilmesi gereken düğümler

Mutagenicity Düğümleri	**Örnek Değeri**
logp	logp5
lumo	lumo8
inda	0
ind1	1
bagTipi	7
element	c
atomTipi	22
yuk	ch8

Şekil 3.8 ile Tablo 3.9'da verilen bilgilere örnek bir yol: muta(d1,doğru): muta_atm(d1,d1_11,c,22,ch8), logp(d1,logp5), lumo(d1, lumo8), inda(d1,0), ind1(d1,1), muta_bond(d1,d1_11,d1_17,7) şeklinde olmaktadır.

Bu örnek kuralda çocuk düğümü olmayan bütün özelliklerin kullanıldığı bir yol bulunmuştur. Bu kurallardan bazılarını kullanmak daha iyi kavram tanımlarının bulunmasına engel olabileceği için bazı özelliklerin çıkarılabilmesi gerekmektedir.

Buna bir çözüm olarak bazı kuralları elimine edebilmek için çocuk düğümü olmayan her bir özellik için yeni bir düğüm oluşturulmuştur. Oluşturulan düğümlerin isimleri özelliklerin genel ismiyle adlandırılmış ve tüm ilgili düğümlerle arasında bir bağlantı kurulmuştur. Böylece genel isimleri içeren bir yol bulunulduğunda bu özellikler bütün düğümlerde ortak olarak yer aldığından kurala dâhil edilmemiş olur. Şekil 3.13'ten bu şekliyle çıkartılabilecek örnek kurallar verilmiştir.

```
muta_atm(d1,d1_11,element,atomTipi,yuk), logp(d1,logp), lumo(d1, lumo), inda(d1,0),
ind1(d1,ind1),  muta_bond(d1,d1_11,d1_17,7)

muta_atm(d1,d1_11,c,22,ch8), logp(d1,logp5), lumo(d1, lumo8), inda(d1,inda),
ind1(d1,ind1), muta_bond(d1,d1_11,d1_17,7)

muta_atm(d1,d1_11,element,22,ch8), logp(d1,logp5), lumo(d1, lumo), inda(d1,0),
ind1(d1,1),  muta_bond(d1,d1_11,d1_17,bagTipi)
```

Şekil 3.13. Mutagenicity için örnek yollar

Şekil 3.13'te verilen örnek yollardan birincisine bakıldığında element, atomTipi, yuk, logp, lumo ve ind1 değerleri, ikincisine bakıldığında inda, ind1 değerleri ve üçüncü örneğe bakıldığında ise lumo, bagTipi değerleri sonradan oluşturduğumuz düğüm ilişkilerini parametre olarak aldığı görülmektedir. Bu da bu düğümlerin kuralları etkilemediğini göstermektedir. Şekil 3.14'de gereksiz özellikler çıkarıldığında oluşan yeni yollar görülebilir.

```
inda(d1,0), muta_bond(d1,d1_11,d1_17,7)

muta_atm(d1,d1_11,c,22,ch8), logp(d1,logp5), lumo(d1, lumo8),
muta_bond(d1,d1_11,d1_17,7)

muta_atm(d1,d1_11,element,22,ch8), logp(d1,logp5), inda(d1,0), ind1(d1,1),
muta_bond(d1,d1_11,d1_17,bagTipi)
```

Şekil 3.14. Mutagenicity için sadeleştirilmiş örnek yollar

3.2.3. Aday kavram tanımlarının değerlendirilmesi

Aday kavram tanımları olarak adlandırılan yolların bulunması sonucunda değerlendirme aşamasına geçilmektedir. Değerlendirme destek ve güven değerleri kullanarak yapılmaktadır. Destek değeri hedef örneği açıklayan bir yolun açıkladığı hedef örnek sayısı ile ilgili, güven değeri ise hedef örnek dışındaki açıklanan yollarla ilgili bir ölçümdür.

Her aday için literatürde geçen destek ve güven değerlerini sağlayan kurallar kavram tanımları olarak alınmış olur. Destek ve güven değerlerini sağlanmıyorsa adaylar elenmiş olur.

Algoritmanın performansını arttırmak için hash listeleri kullanılmıştır. Bu listeler eşik değerlerini geçenler için kavram tanımları hash listesi, geçemeyen yollar için elenen aday kavram tanımları listeleridir. Bulunan yollar genelleştirilerek uygun listelerde kullanılmıştır. Genelleştirilmiş hallerinin kullanılma sebebi bir hedef örneği için bulunan bir yolun başka biri için tekrardan aranmasını engellemektir. Şekil 3.15'te Kinship veri kümesi için "son" ilişkisini açıklayan iki aday kavram tanımları ve genelleştirilmiş halleri verilmiştir.

$\text{Colin} \xleftarrow{\text{mother}} \text{Victoria} \xrightarrow{\text{wife}} \text{James}$	$\text{A} \xleftarrow{\text{mother}} \text{B} \xrightarrow{\text{wife}} \text{C}$
$\text{Arthur} \xleftarrow{\text{mother}} \text{Penelope} \xrightarrow{\text{wife}} \text{Chris}$	$\text{A} \xleftarrow{\text{mother}} \text{B} \xrightarrow{\text{wife}} \text{C}$

Şekil 3.15. Örnek yollar ve genelleştirilmiş halleri

Şekil 3.15'de aynı ilişkileri barındıran farklı hedef örnekleri için bulunan kuralların genelleştirilmiş hallerinin aynı olduğu görülmektedir. Bu kurallardan herhangi birisi bulunulduğunda ilgili kuralın destek ve güven değerleri hesaplanır. Bu belirtilen destek ve güven değerlerini sağlayan bir kuralsa genelleştirilmiş yol kavram tanımları hash listesine destek ve güven değerleri ile birlikte alınır. Belirtilen değerleri sağlamıyorsa elenen kavram tanımları hash listesine eklenir ve başka bir kavram tanımlarının aranmasına geçilir.

Daha sonra Şekil 3.12'deki diğer kural bulunulduğunda genelleştirilmiş hali iki hash listesinde de aranır. Kural daha önce bu listelerden birine eklendiği için herhangi bir hesaplama yapmadan diğer kuralların aranmasına geçilir. Hash liste kullanılmasının sebebi aramanın hash listede normal listeden daha hızlı yapabilmesidir.

Kinship'de "son" hedef ilişkisi için toplam 88 adet aday yol bulunmaktadır. Ancak bu adaylar genelleştirilmiş formda düşünüldüğünde tekrarlı yollar görülmektedir. Genelleştirilmiş halleriyle toplamda eşsiz 16 aday kavram tanımları bulunmaktadır. Tablo 3.10'da bu yollar verilmiştir.

Tablo 3.10. son(A,B) için aday kurallar

Aday Kavram Tanımları
sister(C,A), daughter(C,B)
sister(C,A), mother(B,C)
brother(A,C), daughter(C,B)
brother(A,C), mother(B,C)
father(C,A), wife(B,C)
father (C,A), husband(C,B)
sister(C,A),father(B,C)
brother(A,C), father(B,C)
mother(C,A), wife(C,B)
mother (C,A), husband(B,C)
nephew(A,C), sister(B,C)
nephew (A,C), brother(C,B)
uncle(C,A),sister(B,C)
uncle (C,A), brother (C,B)
aunt(C,A),sister (C,B)
aunt(C,A), brother (B,C)

Adayların genelleştirilmiş halleri değerlendirilirken yolların yönleri de önemlidir. Aynı ilişkileri içeren yollar yönler değiştiğinde farklı ilişkileri açıklayabilmektedir. Bu sebeple yönlerin önemsenmemesi durumunda destek ve güven değerlerinin değişmesi ile ya yanlış kavram tanımları elde edilmiş olacak ya da alınan kavram tanımlarının destek ve güven değerleri tutarsız olacaktır. Şekil 3.16'da örnek olarak Kinship veri kümesinin farklı yönler kullanıldığında açıklanan ilişkinin değiştiği bir durum gösterilmektedir.

$\text{Arthur} \xrightarrow{\text{brother}} \text{Victoria} \xrightarrow{\text{mother}} \text{Colin}$	uncle
$\text{Colin} \xleftarrow{\text{brother}} \text{Charlotte} \xleftarrow{\text{mother}} \text{Victoria}$	son

Şekil 3.16. Örnek yollar ve açıkladığı ilişkiler

Şekil 3.16'daki yolların içerdiği ilişkilerin aynı olduğu görülmektedir. Bu iki yolun açıkladığı ilişki birinci örnekte uncle olurken ikinci örnek için son olmaktadır. Bu örnekler yönsüz olarak değerlendirilmeye alınacak olursa son ilişkisini açıklamak için uncle ilişkisini açıklayan kural destek değerini etkilemese de güven değerini değiştirir. Bu da alınacak kuralların değişmesi anlamına gelir.

Yöntemde kinship veri kümesi için destek değeri 0,1 güven değeri 0,7 olarak alınmıştır. “son” ilişkisi açıklayan benzersiz 16 aday kavram tanımdan değerlendirme kriterlerini geçemeyen yollar elendiğinde kural sayısı 4’e düşmektedir. Tablo 3.11’de destek ve güven değerleri sonucunda elde edilen kavram tanımları verilmiştir.

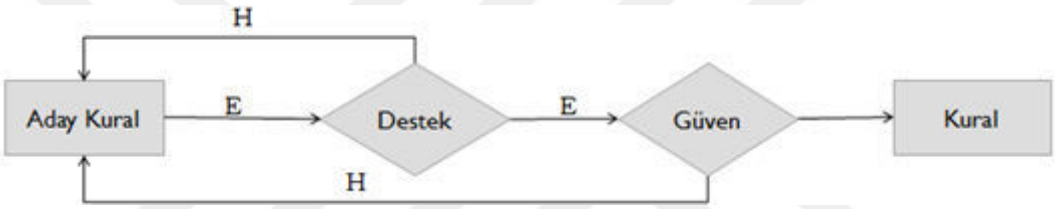
Tablo 3.11. son(A,B) için kavram tanımları

Aday Kavram Tanımları
sister(C,A); daughter(C,B)
sister(C,A); mother(B,C)
brother(A,C); daughter(C,B)
brother(A,C); mother(B,C)

Kavram tanımları bulunduktan sonra bulunan kuralların hedef düğümlerin ne kadarını açıkladığına bakılır. Böylece bulunan kuralların açıklamadığı hedef düğümler tespit edilmiş olur. Tablo 3.11’da verilen kavram tanımları “son” ilişkisi içeren tüm hedef örnekleri açıkladığı görülmüştür.

4. DENEYSEL SONUÇLAR

Önerilen çalışmanın performansını değerlendirmek için farklı öğrenme problemlerinde kullanılan veri kümelerinde çeşitli deneyler gerçekleştirilmiştir. Bu bölümde ilk olarak deneyde kullanılan veri kümeleri anlatılmaktadır. Daha sonra veri kümelerinde gerçekleştirilen deneylerin sonuçları tartışılmaktadır. Deney sonuçları elde edilirken destek ve güven değerleri kullanılmıştır. Ayrıca önerilen sistem literatürde geçen çeşitli uygulamalarla kıyaslanmaktadır. Şekil 4.1'de aday kuralların kural olması için gerekli aşamaların akış diyagramı verilmiştir.



Şekil 4.1. Aday kuralın değerlendirilme aşaması

Destek değeri, bulunan bir kuralın hedef örneklerden ne kadarını açıkladığını gösteren parametredir. Bir kuralın açıkladığı hedef örnek sayısının toplam hedef örnek sayısına oranı ile bulunur. Denklemin (4.1)'de formülü verilmiştir;

$$\text{destek değeri} = \frac{\text{açıklanan hedef örnek sayısı}}{\text{toplam hedef örnek sayısı}} \tag{4.1}$$

Güven değeri, bulunan bir kuralın pozitif hedef örneklerin dışındaki açıkladığı kurallarla ilgilenen bir parametredir. Pozitif hedef örnekleri dışında açıklanan örneklerin fazla olması değeri düşük olmasını sağlar. Bir kuralın açıkladığı hedef örnek sayısının toplam açıkladığı örnek sayısına oranı ile bulunur. Denklem (4.2)'de formülü verilmiştir;

$$\text{güven değeri} = \frac{\text{açıklanan hedef örnek sayısı}}{\text{toplam açıklanan örnek sayısı}} \tag{4.2}$$

Belirlenen ölçüm değerleri elde edilen kavram tanımlarının geçerlilik oranını tespit etmek kuralların başarı oranını görmek için önemlidir. Bu ölçüm ancak kapsam değeri ile yapılabilir. Bulunan tüm kurallarda açıklanan özellik sayısının,

açıklanmak istenen özellik sayısına oranı ile bulunur. Kapsam değerinin 100 olması tüm özelliklerin açıklandığını, 0 olması ise hiçbir özelliğin açıklanmadığını ifade etmektedir. Denklem (4.3)'te formülü yer almaktadır;

$$\text{kapsam değeri} = \frac{\text{açıklanan özellik sayısı}}{\text{açıklanmak istenen özellik sayısı}} \tag{4.3}$$

4.1. Veri Kümeleri

Bu çalışmada 7 farklı veri kümesi kullanılmıştır. Tablo 4.1'de kullanılan veri kümelerinin tanımı verilmiştir.

Tablo 4.1. Veri kümelerine bakış

Veri Kümesi	**Tanım**
Elti	Elti ilişkisini açıklayan veri kümesi
Dünür	Dünür ilişkisini açıklayan veri kümesi
Kinship	Birçok ilişkiyi açıklayan veri kümesi
Same Generation	Gen ilişkisini açıklayan veri kümesi
Eastbound	Doğuya giden araçları açıklayan veri kümesi
Mutagenicity	Mutajenik bileşikleri öğrenmeyi amaçlayan veri kümesi
PTE	Karserojen ilaçları öğrenmeyi amaçlayan veri kümesi

Tablo 4.2'de bu veri kümelerinin özellikleri verilmiştir.

Tablo 4.2. N-argümanlı verilerin hedef örnekleri

Veri Kümesi	**İlişki Sayısı**	**Özellik Sayısı**
Elti	9	224
Dünür	9	224
Kinship	12	744
Same Generation	2	408
Eastbound	12	196
Mutagenicity	12	53697
PTE	15	56879

Kavram tanımlarını elde etmek için destek ve güven değerlerinden faydalanılmıştır. Bu çalışmada kullanılan veri kümeleri için kullanılan minimum destek ve güven değerleri Tablo 4.3'de verilmiştir.

Tablo 4.3. Minimum destek ve güven değerleri

Veri Kümesi	Destek Değeri	Güven Değeri
Elti	0,3	0,7
Dünür	0,2	0,3
Kinship	0,1	0,7
Same Generation	0,3	0,6
Eastbound	0,1	0,6
Mutagenicity	0,1	0,7
PTE	0,1	0,7

Tablo 4.3'de verilen destek ve güven değerleri literatürden alınmıştır. Testler 2,5 Ghz işlemci ve 4 Gb belleğe sahip bir bilgisayarda yapılmıştır.

4.2. Sonuçlar

Önerilen yöntem çizgenin modellenmesi ile başlanır. Uygulamada kullanılan veri kümeleri ilişkisel modelde bulunduğundan uygun formatta düzenlenmesi gerekmektedir. Bunun için öncelikle düğüm ve kenar bilgileri tespit edilmiştir. Daha sonra düzenlenen dosyaların Neo4j kütüphanelerinden faydalanarak Java'da çizgeleri oluşturulmuştur. Tablo 4.4'de her bir veri kümesi için çizge oluşturma zamanları ile veri kümelerinde oluşan düğüm ve kenar sayıları verilmiştir.

Tablo 4.4. Çizgelerin özellikleri ve oluşma zamanları

Veri Kümesi	Düğüm Sayısı	Kenar Sayısı	Çizge Oluşma Süresi (sn)
Elti	47	224	6,51
Dünür	24	128	5,60
Kinship	24	112	5,41
Same Generation	47	408	5,93
Eastbound	55	183	6,25
Mutagenicity	12650	35776	503
PTE	19363	27132	840

Tablo 4.4'de düğüm ve kenar sayıları birbirine yakın olan veri kümeleri için çizge oluşma sürelerinin yakın olduğu düğüm ve kenar sayısı arttıkça da çizge oluşma sürelerinin arttığı görülmektedir.

Elti veri kümesi gerçek verilerle oluşturulmuş akrabalık ilişkilerini içeren veri kümesidir. Türkçede iki erkek kardeşin eşleri arasındaki ilişkiyi tanımlar. elti(A,B) ilişkisini açıklayan aday kavram tanımlarından belirlenen destek ve güven değerlerini geçen 8 kural bulunmuştur. Tablo 4.5'de bulunan kurallar verilmiştir.

Tablo 4.5. Elti ilişkisini açıklayan kurallar

Kurallar	**Destek Değeri**	**Güven Değeri**
wife(A,C),brother(D,C),wife(B,D)	1,0	1,0
wife(A,C),brother(D,C),husband(D,B)	1,0	1,0
wife(A,C),brother(C,D),wife(B,D)	1,0	1,0
wife(A,C),brother(C,D),husband(D,B)	1,0	1,0
husband(C,A),brother(D,C),wife(B,D)	1,0	1,0
husband(C,A),brother(D,C), husband(D,B)	1,0	1,0
husband(C,A),brother(C,D),wife(B,D)	1,0	1,0
husband(C,A),brother(C,D), husband (D,B)	1,0	1,0

Tablo 4.5'te "wife", "husband" ve "brother" ilişkilerinden oluşan kurallar görülmektedir. "brother" için brother(A,B) ilişkisi, brother(B,A)'ya eşit olduğundan kuralların bazıları simetrik çıkmıştır.

Kinship veri kümesi birçok aileden alınan verilerle oluşturulmuş bir veri kümesidir. 24 kişinin bulunduğu bu veri kümesi 2 aile grubundan oluşmaktadır. Kinship veri kümesi 12 ilişki içermektedir. Bu ilişkilerin hepsi ayrı ayrı açıklanmak istenmiştir. Tablo 4.6'da ölçütleri sağlayan ilişkiler ve kurallar verilmiştir.

Tablo 4.6. Kinship ilişkileri ve kavram tanımları

İsim	**Kurallar**	**İsim**	**Kurallar**	**Destek Değeri**	**Güven Değeri**
aunt	wife(A,C),uncle(C,B)	uncle	wife(C,A),aunt(C,B)	1,0	1,0
	husband(C,A), uncle(C,B)		husband(A,C),aunt(C,B)	1,0	1,0
brother	son(A,C),daughter(B,C)	sister	daughter(A,C),son(B,C)	1,0	1,0
	son(A,C),mother(C,B)		daughter (A,C),mother(C,B)	1,0	1,0
	son(A,C),father(C,B)		daughter (A,C),father(C,B)	1,0	1,0
	mother(C,A),daughter(B,C)		mother(C,A),son (B,C)	1,0	1,0
	father(C,A),daughter(B,C)		father(C,A), son(B,C)	1,0	1,0

Tablo 4.6.(Devam) Kinship ilişkileri ve kavram tanımları

İsim	Kurallar	İsim	Kurallar	Destek Değeri	Güven Değeri
daughter	sister(A,C),son(C,B)	son	sister(C,A),daughter(C,B)	1,0	1,0
	brother(C,A),son(C,B)		brother(A,C),daughter(C,B)	1,0	1,0
father	wife(C,A),mother(C,B)	mother	wife(A,C),father(C,B)	1,0	1,0
	husband(A,C),mother(C,B)		husband(C,A),father (C,B)	1,0	1,0
husband	daughter(C,A),mother(B,C)	wife	daughter (C,A),father (B,C)	0,6	1,0
	son(C,A),mother(B,C)		son(C,A),father(B,C)	0,6	1,0
	father(A,C),daughter(C,B)		mother(A,C),daughter(C,B)	0,6	1,0
	father(A,C),mother (B,C)		mother(A,C),father(B,C)	0,6	1,0
	father(A,C),son(C,B)		mother(A,C),son(C,B)	0,6	1,0
	niece(C,A),aunt(B,C)		niece(C,A),uncle(B,C)	0,4	1,0
	nephew(C,A),aunt(B,C)		nephew(C,A),uncle(B,C)	0,4	1,0
	uncle(A,C),niece(C,B)		aunt(A,C),niece(C,B)	0,4	1,0
	uncle(A,C),aunt(B,C)		aunt(A,C),uncle(B,C)	0,4	1,0
	uncle(A,C),nephew (C,B)		aunt(A,C),nephew (C,B)	0,4	1,0
nephew	sister(C,A),niece(C,B)	niece	sister(A,C),nephew (C,B)	1,0	1,0
	Brother(A,C),biece(C,B)		Brother(C,A),nephew (C,B)	1,0	1,0

Tablo 4.6'da "husband" ve "wife" ilişkilerinde kavram tanımlarının sayısının çok çıkması tüm hedef düğümleri açıklayabilmek için kuralların yetmemiş olmasından kaynaklanmaktadır. Diğer ilişkilerde destek değeri 1 çıkmasına rağmen bu ilişkilerde bulunan kavram tanımlarının destek değerleri verilen en küçük destek değerini 0,6 ile sağlamasına rağmen 1 olmamaktadır. Bu kavram tanımları olarak bulunan cümlelerin bütün hedef düğümlerini açıklayamadığı anlamına gelmektedir. Bu iki açıklanmak istenen ilişki için ilk 5 kavram tanımlarına bakıldığında son, daughter, mother ve father gibi çocuklu ailelerde bulunan ilişkiler bulunmaktadır.

Kinship veri kümesinde çocuğu olmayan eşler bulunduğundan başka kavram tanımlarına ihtiyaç duyulmuştur. Bu kişiler için nephew, niece, aunt, uncle ilişkilerini içeren kavram tanımlarını ortaya çıkmaktadır. Eşli kişilerde de bu ilişkiler bulunmadığından bu ilişkileri içeren kavram tanımlarının destek değeri 0,4 olmaktadır. Tablo 4.7'de çocuklu ve çocuksuz kişilerin listesi verilmiştir.

Tablo 4.7. Kinship veri kümesinde eşlerin durumu

Çocuklu Eşler	**Çocuksuz Eşler**
Chris-Penelope	Arthur-Margaret
Andrew-Christine	Charles-Jennifer
James-Victoria	Emilio-Gina
Roberto-Maria	Tomaso-Angela
Pierro-Francesca	
Marco-Lucia	

Dünür veri kümesi Kinship veri kümesinin güncellenerek Türkçede evli eşlerin anne ya da babaları arasındaki ilişkiyi açıklayan dünür ilişkisinin eklenmiş halidir. Tablo 4.8'de Dünür veri kümesi için bulunan kavram tanımları verilmiştir.

Tablo 4.8. Dünür ilişkisini açıklayan kurallar

Kurallar	**Destek Değeri**	**Güven Değeri**
daughter(C,A), wife (C,D),son(D,B)	1,0	1,0
daughter(C,A), husband (D,C),son(D,B)	1,0	1,0
son(C,A), wife (C,D), daughter (D,B)	1,0	1,0
son (C,A), husband (D,C), daughter (D,B)	1,0	1,0

Tablo 4.8'de bulunan görülen kavram tanımlarının her birinin ayrı ayrı destek değerleri 0,5 çıkmıştır. Bu değerin anlamı bulunan kavram tanımlarının her biri dünür ilişkisi bulunan hedef örneklerden ancak yarısını açıklayabildiğini göstermektedir. Dünür ilişkisi evli kişilerin aileleri arasındaki ilişki olduğundan kurallarda bulunan daughter ile son ilişkisi bunun sebebidir. İlk iki kural daughter ilişkisi ile başladığından A terimi eşlerden kadın olanın ailesi olmak zorundadır. Bu sebeple bu kavram tanımları sadece kız tarafının erkek tarafı ile olan ilişkilerini açıklamaktadır. Son iki kavram tanımları ise son ilişkisini içerdiği için ve aynı zamanda A terimi erkek eşin ailesi olması gerektiğinden sadece erkek tarafının kız tarafı ile olan ilişkisini açıklamaktadır.

Same Generation veri kümesi aynı genden olan insanların tespiti için oluşturulmuş bir veri kümesidir. Toplam 2 ilişki ve 408 düğüm barındırmaktadır. Same Generation veri kümesinde 344 gen ilişkisi bulunmaktadır. Same Generation Elti, Kinship ve Dünür veri kümelerine bakıldığında açıklanmak istenen hedef örnek sayısı çok daha

fazla olan bir veri kümesi olmasından dolayı deneyler açısından önemlidir. Tablo 4.9'da Same Generation veri kümesi için bulunan kavram tanımları verilmiştir.

Tablo 4.9. Same Generation ilişkisini açıklayan kurallar

Kurallar	**Destek Değeri**	**Güven Değeri**
sg(C,A), sg (C,D),sg(B,D)	1,0	1,0
sg(C,A), parent (D,C), parent (D,B)	0,88	1,0
parent (A,C), sg (D,C), parent (B,D)	0,94	1,0
parent (C,A), parent (C,D),sg(D,B)	0,88	1,0

Birinci kavram tanımı tüm hedef örnekleri açıklarken diğerleri açıklayamamaktadır. Birincisinde sadece gen ilişkisi bulunduğundan ve bu ilişkiler veride 3 kişiden az gen ilişkili bir grup olmadığından tüm hedef örnekleri kapsamaktadır. İkinci ve dördüncü kural ise parent ilişkisi olmayan ailelerden ve iki çocuklu olmayan ailelerden dolayı tüm örnekleri açıklayamaz. Üçüncü kavram tanımı da parent ilişkisi olmayan kişileri açıklayamaz.

Eastbound arabalar hakkında bilgi tutan veri kümesidir. Bu arabaların yük, şekil gibi özellikleri arka plan bilgisi olarak bulunmaktadır. Bu veri kümesinde amaç bir aracın doğuya gidip gitmediğinin bulunmasıdır. Tablo 4.10'da kurallar verilmiştir.

Tablo 4.10. Eastbound ilişkisini açıklayan kurallar

Kurallar	**Destek Değeri**	**Güven Değeri**
has_car(A,B),short(B),closed(B)	1,0	1,0
has_car(A,B), closed(B)	1,0	0,71
has_car(A,B), double(B)	0,4	0,66

Eastbound veri kümesi için Tablo 4.10'da görülen 3 uzunluklu ilk kural destek ve güven değerleri ile tüm örnekleri açıklamaktadır. İkinci ve üçüncü örnek eastbound veri kümesi için iki uzunluklu sonuçlar için çıkan kurallardır. Bu kurallardan birincisi tüm doğuya giden araçları açıklamaktadır. Ancak bu kural aynı zamanda bazı batıya giden araçlarda da bulunduğu için güven değeri düşük çıkmıştır. Üçüncü kural ise hem tüm hedef örnekleri açıklayamamakta hem de bazı batıya giden araçları açıklamaktadır. Bu sebeple çok iyi bir kural değildir. Eastbound veri kümesi için Kavurucu ve arkadaşları tarafından yapılmış olan çalışma [63] ile benzerlik göstermektedir.

Sonuçlar maksimum kavram tanımı uzunluğu (K.T.), K.T. sayısı, kapsam değeri ve süre açısından kıyaslanmıştır. Tablo 4.11'de bu çalışmada veri kümeleri için ortaya çıkan deneysel sonuçlar gösterilmektedir.

Tablo 4.11. Test sonuçları ve süreleri

Veri Kümesi	**Hedef İlişki**	**Maksimum K.T Uzunluğu**	**K.T. Sayısı**	**Kapsam (%)**	**Süre (saniye)**
Elti	elti	3	8	100	0,243
Dünür	dunur	3	4	100	0,230
Kinship	aunt	2	2	100	0,177
	brother	2	5	100	0,208
	daughter	2	2	100	0,191
	husband	2	10	100	0,201
	father	2	2	100	0,193
	mother	2	2	100	0,196
	nephew	2	2	100	0,194
	niece	2	2	100	0,180
	sister	2	5	100	0,218
	son	2	2	100	0,204
	uncle	2	2	100	0,176
	wife	2	10	100	0,192
Same Generation	sg	3	4	100	7,287
Eastbound	eastbound	3	3	100	0,263

Tablo 4.11'de sunulan sonuçlardan yöntemin test edildiği tüm veri kümeleri için kapsam değerinin 100 çıktığı görülmektedir. Bu sonuç açıklanmak istenen tüm hedef örneklerinin bulunan kurallar ile açıklanabildiğini göstermektedir.

Elti, Dünür, Same Generation ve Eastbound veri kümeleri için 3 uzunluklu kurallar ile diğer veri kümeleri için de 2 uzunluklu kurallar hedef örnekleri açıklamak için yeterli olmuştur. Kavram tanımlarının sayısının husband ve wife hedef ilişkilerindeki daha önceden bahsedilmiş olan problemden dolayı 10 kural bulunduğu görülmektedir.

Deneylerin elde edilme sürelerine bakıldığında benzer uzunlukta kurallar bulunduğundan sürelerin birbirine çok yakın olduğu görülmektedir. Same Generation veri kümesi için bu süresinin yüksek çıkmasının sebebi hedef örneklerin sayısının

çok fazla olmasıdır. Dolayısıyla her bir hedef örnek için yöntem çalıştığından süre fazla çıkmaktadır.

Yöntem son yapılan çizge tabanlı kavram keşfi çalışmaları ile kıyaslandığında sunulan yöntem Gao ve arkadaşlarının [65] tarafından gerçekleştirilen yöntemle Kinship veri kümesi için çalışma zamanı yakın çıkmıştır. Aynı sistemle kıyaslandığında daha kısa kavram tanımları bulunmuştur. Bu sonuç kısa kavram tanımlarının yorumlanması daha kolay olduğu için önemli bir sonuçtur. Bir diğer çalışma [66] ile yöntem kıyaslandığında Elti veri kümesi için yakın sonuçlar çıkarken Kinship veri kümesi için yöntemin çalışma süresi çok daha başarılıdır. Kapsam değeri ise iki yöntem ile de aynı çıkmıştır. Yöntem Nazmiye ve arkadaşları [67] ile yapılan çalışma ile kıyaslandığında Elti ve Same Generation veri kümeleri için çalışma süreleri daha kısa çıkmıştır. Ayrıca Same Generation ve Kinship veri kümesi için kapsam değeri daha yüksek çıkmıştır. Tablo 4.15'de karşılaştırmalı sonuçlar verilmiştir.

Tablo 4.12. Karşılaştırmalı sonuçlar

Data set	Sistem	Çalışma Zamanı (s)	Kapsam Değeri (%)
Elti	Yöntem	0,243	100
	[70]	-	-
	[71]	2	100
	[72]	4	100
Kinship	Yöntem	0,191	100
	[70]	0,027	-
	[71]	1,23	100
	[8]	-	99
Same Generation	Yöntem	7,287	100
	[70]	-	-
	[71]	-	-
	[8]	24	96

Mutagenicity veri kümesi 230 bileşiğin mutajenikliğini ve bu bileşiklerin özelliklerini içeren veri kümesidir [40]. Bir ilacın mutajenik olması o ilacın DNA gibi canlı organizmaların mutasyona sebep olabileceğini göstermektedir [68]. Bu problem organik kimya alanıyla ilgilidir. Nitroaromatik olarak geçen bu bileşikler egzoz dumanlarında ve bazı endüstriyel bileşiğin sentezinde ara madde olarak bulunur [58].

Bu çalışmada 230 bileşikten regresyon dostu olan 188 ilaç kullanılmıştır. Mutagenecity veri kümesi ile ilgili yapılmış çalışmada bulunan bazı kurallar Tablo 4.13'de verilmiştir.

Tablo 4.13. Mutagenecity çalışmasındaki örnek yollar [72]

Kurallar	**Güven değeri**
atm(A,B,c,10,C), atm(A,D,c,22,E), bond(A,D,B,1)	0,3
atm(A,B,c,27,C), bond(A,D,E,1), bond(A,B,E,7),	0,58
atm(A,B,o,40,C), atm(A,D,n,32,C)	0,08
atm(A,B,o,40,-0,378)	0,13
atm(A,B,c,195,C)	0,1

Mutagenicity veri kümesi için tanımlanan kurallara göre kavram tanımları bulunmuştur. Tablo 4.14'de Mutagenicity için bulunan bazı kurallar gösterilmiştir.

Tablo 4.14. Mutagenicity kavram tanımları

Kurallar	**Destek Değeri**	**Güven Değeri**
atm(A,B,h,3,ch14)	0,16	0,77
atm(A,B,c,22,ch8)	0,83	0,736
atm(A,B,h,3,ch15), lumo(A,lumo2)	0,1	1
atm(A,B,h,3,ch15), inda(A,0.0),indV(A,1.0)	0,42	0,98
atm(A,B,h,3,ch15), atm(A,D,c,22,ch8), bond(A,D,B,1), inda(A,0.0),indV(A,1.0)	0,41	1
atm(A,B,c,22,ch8), atm(A,D,c,22,ch8), bond(A,D,B,7), indV(A,1.0)	0,68	1
atm(A,B,o,40,ch4), atm(A,D,n,38,ch22), bond(A,D,B,2)	0,69	0,70
atm(A,B,h,3,ch16), atm(A,D,c,22,ch8), bond(A,D,B,1)	0,35	0,83
atm(A,B,c,29,ch11), atm(A,D,c,29,ch11), bond(A,D,B,1),inda(A,0.0),indV(A,1.0)	0,16	0,95
atm(A,B,o,40,ch4), atm(A,D,n,38,ch23), bond(A,D,B,2), inda(A,0.0),indV(A,1.0)	0,13	1
atm(A,B,element,atomType,ch4), bond(A,D,E,2), bond(A,B,E,1), (-0,398<=ch4<=-0,376)	0,872	0,741

Mutagenesis çalışmasında bulunan Tablo 4.13'deki kurallar bu tez çalışmasında bulunan Tablo 4.14'de bulunan kurallara kıyaslandığında benzer yapıda kuralların bulunduğu görülmektedir. Ayrıca Mutagenicity veri kümesinde bulunan farklı özellikleri içeren farklı kuralların da bulunduğu Tablo 4.12'de görülmektedir. Bu

çalışmada Mutagenicity veri kümeleri içim bulunan kavram tanımları bu veri kümesi için tüm hedef örnekleri açıklamaktadır.

PTE ilaçların kanserojen olup olmadığı ile ilgilenen ve ilaçların bilgilerini tutan veri kümesidir [69-71]. Kullanılan sürümünde 298 ilaç bulunmaktadır. Bu ilaçlardan 163'ü kanserojen olup geri kalanında kanserojen özelliği bulunmamaktadır. PTE veri kümesi için tanımlanan kurallara göre kavram tanımları bulunmuştur. Tablo 4.15'de PTE için bulunan bazı kurallar gösterilmiştir.

Tablo 4.15. PTE veri kümesi için kavram tanımlar

Kurallar	**Destek Değeri**	**Güven Değeri**
atm(A,B,h,atomType,charge),has_property(A,cytogen_ca,property2), ind(A,di10,1)	0,140	0,703
atm(A,B,c,atomType,charge),has_property(A, salmonella,property2), ind(A,indp1,1)	0,111	0,833
atm(A,B,element,atomType,ch12),has_property(A,cytogen_sce,property2), ind(A,indp1,1),(-0,159<=ch12<=-0,126)	0,104	0,875
atm(A,B,element,22,charge),has_property(A, salmonella,property2), ind(A,di10,1)	0,126	0,739
atm(A,B,element,atomType,ch22), atm(A,D,element,32,charge), (0,191<=ch22<=0,227)	0,245	0,702

PTE veri kümesi için minimum destek ve güven değerini aşan kurallardan bazıları Tablo 4.15'de görülmektedir. Bulunan kurallar Mutagenesis [72] çalışmasında bulunan kurallarla benzer yapıya sahiptir. PTE için bulunan kurallar tüm hedef örneklerini başarılı bir şekilde açıklamaktadır.

Sunulan yöntemin ölçeklenebilirliğini test etmek için yöntem Kinship veri kümesi için farklı hedef örneklerini 2 uzunluklu kurallardan başlayarak 9 uzunluklu kurallara kadar çalıştırılmıştır. Tablo 4.16'da test edilen hedef ilişkiler için kavram tanımlarının bulunması için metodun çalışma süreleri verilmiştir.

Tablo 4.16. Farklı uzunluktaki kurallar için yöntemin çalışma süreleri

	Süre (saniye)							
Hedef ilişki /kural uzunluğu	2	3	4	5	6	7	8	9
aunt	0,177	0,455	0,992	2,032	5,750	18,584	41,069	79,672
brother	0,208	0,357	0,805	1,920	4,610	11,020	25,623	43,229
daughter	0,191	0,459	1,021	2,387	7,886	20,463	48,895	75,284

Tablo 4.16.(Devam) Farklı uzunluktaki kurallar için yöntemin çalışma süreleri

	Süre (saniye)							
Hedef ilişki /kural uzunluğu	2	3	4	5	6	7	8	9
husband	0,201	0,462	1,237	2,463	8,067	19,367	50,498	95,750
father	0,193	0,404	1,032	2,029	5,730	18,278	45,273	98,016
mother	0,196	0,463	0,998	2,240	5,870	17,942	44,581	95,999
nephew	0,194	0,467	0,983	2,114	6,944	18,177	41,689	84,410
niece	0,180	0,573	0,965	2,170	5,995	18,134	42,533	86,698
sister	0,218	0,474	0,898	1,988	4,895	11,556	30,463	49,765
son	0,204	0,517	0,993	2,388	7,318	19,607	43,057	76,659
uncle	0,176	0,447	0,917	1,878	5,461	16,147	39,909	72,537
wife	0,192	0,456	1,131	2,358	7,772	19,056	47,147	93,320

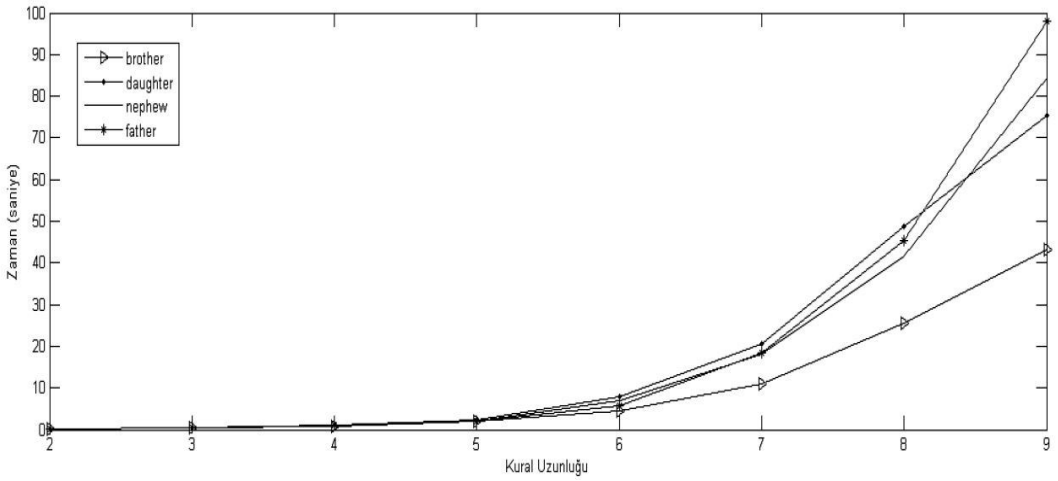
Tablo 4.16'ye bakıldığında kural uzunluğu düşük olan hedef ilişkilerde süreler birbirine yakın çıkarken kural uzunluğu arttıkça hedef ilişkiler arasında süreler açısından belirgin bir artış görülmektedir. Bunun sebebi kavram tanımlarını bulmak için kullanılan yöntemin uzunluk sayıları arttıkça daha fazla yol bulmasından kaynaklanmaktadır. Kinship veri kümesi için 9 uzunluklu yollar için en kısa süren brother ve en uzun süren father hedef ilişkilerinin 2 ve 9 uzunluklu yöntemin işlediği yol sayıları Tablo 4.17'de verilmiştir.

Tablo 4.17. brother ve father için yol sayıları

	Yol sayısı	
Hedef ilişki /kural uzunluğu	2	9
brother	112	90112
father	88	241920

Tablo 4.17'e bakıldığında 2 uzunluklu yolların birbirine yakın olduğu için sürelerde yakın çıkmıştır. 9 uzunluklu yollara bakıldığında yol sayısı yaklaşık 3 katına çıkmasına rağmen sürede yaklaşık 3 kat artmıştır. Bu da yöntemin ölçeklenebilir olduğunu göstermektedir.

Yöntemin ölçelenebilirliğini göstermek amacıyla Tablo 4.16'da bulunan tabloya göre seçilen brother, daughter, nephew ve father hedef ilişkileri için çizdirilen grafik Şekil 4.2'de bulunmaktadır.



Şekil 4.2. Ölçeklenebilirlik

5. SONUÇLAR VE ÖNERİLER

Kavram keşfi veri kümesinde veriler arasındaki açıklanmak istenen bir ilişkinin diğer ilişkileri kullanarak mantıksal çıkarımlarının yapılması problemidir. Literatürde kavram keşfi problemlerine tümevarım mantıksal programlama ve çizge tabanlı yöntemler kullanılmıştır.

Bu çalışmada ise kavram keşfi problemine çizge veri tabanı kullanarak yol bulma tabanlı bir yaklaşım tercih edilmiştir. Bu yaklaşım verilerin modellenmesi ve kavram tanımlarının bulunması aşamalarından oluşmaktadır.

Genellikle ilişkisel veya prolog formatında bulunan veri kümelerinin çizge yapısına dönüştürülmesi Neo4j kütüphanelerinden faydalanarak basit ve etkili bir şekilde yapılmıştır. Kavram tanımlarının tespitinde ise Java ve Neo4j kullanılarak geliştirilmiş yöntemin performansını arttırmak için hash tabanlı bazı algoritma tekniklerinden yararlanılmıştır.

Yöntem çizge veri tabanları tarafından sağlanan işlevlerle gerçeklendiğinden diğer yöntemlerden farklıdır. Aynı zaman yöntem hem iki ilişkili hem de çok ilişkili veri kümeleri için çözüm sunmaktadır. İki türden ilişkiler için de yol bulma tabanlı yöntemle kavram tanımları aranırken veri kümelerindeki farklılıklardan dolayı farklı problemlerde farklı stratejiler uygulanmıştır.

Deneysel sonuçlar önerilen yöntemin farklı öğrenme problemlerine ait veri kümeleri için kavram tanımlarının keşfinde başarılı olduğunu göstermektedir. Daha önce uygulanmış yöntemlerle kıyaslandığında benzer ve yorumlaması daha kolay olan kısa uzunluklu kavram tanımları bulunmuştur. Bulunan kavram tanımları için yöntemin çalışma süresinin önceki çalışmalarla kıyaslandığında iyileştiği görülmüştür. Ayrıca bulunan sonuçların hedefleri kapsamada başarılı olduğu görülmektedir. Son olarak yöntemin farklı uzunluklu kavram tanımları için sonuçlar aranarak elde edilen sürelerle ölçeklenebilir olduğu gösterilmiştir.

Çalışmanın n-argümanlı ilişkilerin modellenmesi ve kavram tanımlarının tespit edilmesinde başarılı olduğu görülmesine rağmen performans olarak geliştirilmesi gerekmektedir. Ayrıca bulunan n-argümanlı sonuçlar için belirli taslaklar dışında kuralları tespit etmede yetersiz kaldığı söylenebilir. Bu sebeple bu çalışmanın devamı olarak yapılacak çalışmalarda n-argümanlı ilişkilerin bahsedilen problemlerine çözüm aranması hedeflenmektedir.

KAYNAKLAR

[1] Kramer S., Lavrač N., Flach P., Propositionalization Approaches to Relational Data Mining, Editors: Džeroski S., Lavrač N., *Relational Data Mining*, Springer-Verlag Berlin Heidelberg, New York, 262-291, 2001.

[2] Džeroski S., Multi-Relational Data Mining: An Introduction, *ACM SIGKDD Explorations Newsletter*, 2003, **5**(1), 1-16.

[3] Blockeel H., Witsenburg T., Kok J., Graphs, Hypergraphs and Inductive Logic Programming, *5th International Workshop on Mining and Learning with Graphs*, Florence, Italy, 1-3 August 2007.

[4] Quinlan R., Learning Logical Definitions from Relations, *Machine Learning*, 1990, **5**(3), 239-266.

[5] Muggleton S., Inductive Logic Programming, *New Generation Computing*, 1991, **8**(4), 295-318.

[6] Muggleton S., De Raedt L., Inductive Logic Programming: Theory and Methods, *The Journal of Logic Programming*, 1994, **19/20**(1), 629-679.

[7] Fonseca N., Costa V. S., Silva F., Camacho, R., On The Implementation of an ILP System with Prolog, *Universidade do Porto*, DCC-2003-03, 1-14, 2003.

[8] Richards B. L., Mooney R. J., Learning Relations by Path-finding, *Tenth National Conference on Artificial Intelligence,* California, USA, 12-16 July 1992.

[9] Fonseca N. A., Srinivasan A., Silva F. M. A., Camacho R., Parallel ILP for Distributed-Memory Architectures, *Machine Learning*, 2009, **74**(3), 257-279.

[10] Bergadano F., Gunetti D., Learning Relations: Basing Top-Down Methods on Inverse Resolution, Editor: Torasso P., *In Advances in Artificial Intelligence*, 1st ed., Springer Berlin Heidelberg, Germany, 190-201, 1993.

[11] Dolsak B., Muggleton S., The Application of Inductive Logic Programming to Finite Element Mesh Design, *Inductive Logic Programming*, Tokyo, Japan, 6-7 June 1992.

[12] Feng C., Inducing Temporal Fault Dignostic Rules from a Qualitative Model, *Eighth International Workshop on Machine Learning*, Illinois, USA, 28-30 June 1991.

[13] King R. D., Muggleton S., Lewis R. A., Sternberg M. J., Drug Design by Machine Learning: The Use of Inductive Logic Programming to Model The Structure-Activity Relationships of Trimethoprim Analogues Binding to

Dihydrofolate Reductase, *Proc Natl Acad Sci USA*, 1992, **89**(23), 11322-11326.

[14] Muggleton S., Feng C., Efficient Induction of Logic Programs, *First Conference on Algorithmic Learning Theory*, Tokyo, Japan, 8-10 October 1990.

[15] Muggleton S., Buntine W., Machine Invention of First Order Predicates by Inverting Resolution, *5th International Workshop on Machine Learning*, Michigan, USA, 12-14 June 1988.

[16] Hulzebos E., Janssen P., Maslankiewicz L., Meijerink M., Muller J., Pelgrom S., Verdam L., Vermeire T., The Application of Structure-Activity Relationships in Human Hazard Assessment: A First Approach, *The Dutch National Institute for Public Health and the Environment*, RIVM-601516008, 1-69, 2001.

[17] Srinivasan A., The ALEPH Manual, Oxford University, http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/aleph/ (Ziyaret tarihi: 12 Mayıs 2017).

[18] King R. D., Srinivasan A., Dehaspe L., WARMR: A Data Mining Tool for Chemical Data, *Journal of Computer-Aided Molecular Design*, 2001, **15**(2), 173–181.

[19] Muggleton S., Inverse Entailment and Progol, *New Generation Computing*, 1995, **13**(3), 245–286.

[20] Quinlan J. R., Learning Logical Definitions from Relations, *Machine Learning*, 1990, **5**(3), 239–266.

[21] Serrurier M., Prade H., Introducing Possibilistic Logic in ILP for Dealing with Exceptions, *Artificial Intelligence*, 2007, **171**(16-17), 939–950.

[22] Shapiro E. Y., *Algorithmic Program Debugging*, MIT Press Cambridge, USA, 1983.

[23] Džeroski S., Lavrač N., Grobelnik M., Learning Nonrecursive Definitions of Relations with Linus, Editor: Kodratoff Y., *Machine Learning*, 1st ed., Springer-Verlag Berlin Heidelberg, Berlin, 265–281, 1991.

[24] Kavurucu Y., Senkul P., Toroslu I. H., Confidence-Based Concept Discovery in Multi-Relational Data Mining, *International Multi Conference of Engineers and Computer Scientists*, Hong Kong, 19-21 March 2008.

[25] Abay N. C., Improving The Efficiency of ILP-based and Graph-based Concept Discovery Systems, Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2014, 383306.

[26] Chittimoori R. N., Holder L. B., Cook D. J., Applying the Subdue Substructure Discovery System to the Chemical Toxicity Domain, *Twelfth*

International Florida Artificial Intelligence Research Society Conference, Florida, USA, 1-5 May 1999.

[27] Rissanen J., *Stochastic Complexity in Statistical Inquiry,* World Scientific Publishing Company, New Jersey, 1989.

[28] Cook D., Holder L. B., Substructure Discovery Using Minimum Description Length and Background Knowledge, *Journal of Artificial Intelligence Research*, 1994, **1**(1), 231- 255.

[29] Djoko S., Cook D. J., Holder L. B., Analyzing the Benefits of Domain Knowledge in Substructure Discovery, *First International Conference on Knowledge Discovery and Data Mining*, Montreal, Canada, 20-21 August 1995.

[30] Gonzalez J. A., Holder L. B., Cook D. J., Structural Knowledge Discovery Used to Analyze Earthquake Activity, *Thirteenth International Florida Artificial Intelligence Research Society Conference*, Florida, USA, 22-24 May 2000.

[31] Su S., Cook D. J., Holder L. B., Application of Knowledge Discovery to Molecular Biology: Identifying Structural Regularities in Proteins, *4th Pacific Symposium on Biocomputing*, Hawai, USA, 4-9 January 1999.

[32] Gonzalez J. A., Holder L. B., Cook D. J., Graph-Based Concept Learning, *Seventeenth National Conference on Artificial Intelligence*, Austin, USA, 30 July-3 August 2000.

[33] Cook D. J., Holder L. B., Su S., Maglothin R., Jonyer I., Structural Mining of Molecular Biology Data, *Engineering in Medicine and Biology Magazine*, 2001, **20**(4), 67-74.

[34] Nijssen S., Kok J. N., A Quickstart in Frequent Structure Mining Can Make A Difference, *Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* Seattle, USA, 22-25 August 2004.

[35] Little M., Transactions and Web services, *Commun. ACM*, 2003, **46**(10), 49-54.

[36] Mutlu A., Karagoz P., A Hybrid Graph-Based Method for Concept Rule Discovery, Editors: Bellatreche L., Mohania M. K., *In Data Warehousing and Knowledge Discovery*, Springer Berlin Heidelberg, Berlin, 327-338, 2013.

[37] Han J., Kamber M., *Data Mining Concepts and Techniques*, 2nd ed., Morgan Kaufmann Publishers, San Francisco, 2006.

[38] Džeroski S., Multi-Relational Data Mining: An Introduction, *SIGKDD Explorations*, 2003, **5**(1), 1-16.

[39] Matsuda T., Motoda H., Washio T., Graph-Based Induction and Its Applications, *Advanced Engineering Informatics*, 2002, **16**(2), 135-143.

[40] Srinivasan A., Muggleton S., Sternberg M. J. E., King R. D., Theories for Mutagenecity: A Study of First-Order and Feature Based Induction, *Artificial Intelligence*, 1996, **85**(1-2), 277-299.

[41] Mutlu A., Senkul P., Kavurucu Y., Improving The Scalability of Ilp-Based Multi-Relational Concept Discovery System through Parallelization, *Knowl.-Based Syst.*, 2012, **27**(1), 352-368.

[42] Han J., Haihong E., Le G., Du J., Survey on NoSQL Database, *6th International Conference on Pervasive Computing and Applications*, Port Elizabeth, South Africa, 26-28 October 2011.

[43] Carlson J. L., *Redis in Action*, 1st ed., Manning Publication, New York, USA, 2013.

[44] Riak KV NoSQL Database, Riak KV Enterprise, USA, http://docs.basho.com/riak/kv/2.2.3/ (Ziyaret tarihi: 20 Mayıs 2017).

[45] Lennon J., *Beginning CouchDB*, Apress Publisher, New York, USA, 2009.

[46] Chodorow K., Dirolf M., *MongoDB: The Definitive Guide*, O'Reilly Media Publisher, USA, 2010.

[47] OrientDB: The World's First Distributed Multi-Model NoSQL Database with a Graph Database Engine, OrientDB Ltd, London, UK, http://orientdb.com/orientdb/ (Ziyaret tarihi: 23 Mayıs 2017).

[48] Lakshman A., Avinash M., Cassandra, A Decentralized Structured Storage System, *ACM SIGOPS Operating Systems Review*, 2010, **44**(2), 35-40.

[49] HBase: Big Table Like Structured Storage for Hadoop HDFS, Hadoop PMC, http://wiki.apache.org/hadoop/Hbase/ (Ziyaret Tarihi: 25 Mayıs 2017).

[50] Seker S. E., Çizge Teorisi (Graph Theory), *YBS Ansiklopedi*, 2015, **2**(2), 17-29.

[51] Biggs N., Lloyd E., Wilson R., *Graph Theory 1736-1936*, 1st ed., Oxford University Press, 1976.

[52] Robinson I., Webber J., Eifrem E., *Graph Databases,* 2nd ed., O'Reilly, USA, 2015.

[53] Kallen N., Pointer R., Kalucki R., Ceaser E., FlockDB, https://github.com/twitter-archive/flockdb/ (Ziyaret tarihi: 30 Mayıs 2017).

[54] AllegraGraph, Franzinc, https://franz.com/agraph/allegrograph/ (Ziyaret tarihi: 30 Mayıs 2017).

[55] Neo4j: The Worlds Leading Graph Database, Neo Technology, https://neo4j.com/ (Ziyaret tarihi: 15 Haziran 2017).

[56] InfiniteGraph, Objectivity, http://www.objectivity.com/products/infinitegraph (Ziyaret tarihi: 29 Mayıs 2017).

[57] HyperGraphDB, http://www.hypergraphdb.org/ (Ziyaret tarihi: 29 Mayıs 2017).

[58] Trinity: A Distributed Graph Engine on A Memory Cloud, Microsoft, https://www.microsoft.com/en-us/research/publication/trinity-a-distributed-graph-engine-on-a-memory-cloud/ (Ziyaret tarihi: 29 Mayıs 2017).

[59] Dex: Scalable High-Performance Graph Database, Sparsity Technologies, http://www.sparsity-technologies.com/ (Ziyaret tarihi: 29 Mayıs 2017).

[60] Needham M., Neo4j: Modelling Hyper Edges in A Property Graph, http://www.markhneedham.com/blog/2013/10/22/neo4j-modelling-hyper-edges-in-a-property-graph/ (Ziyaret tarihi: 1 Haziran 2017).

[61] Jiang B., Traversing Graphs in A Paging Environment, BFS or DFS?, *Inf. Process. Lett.,* 1991, **37**(3), 143-147.

[62] Larson J., Michalski R. S., Inductive Inference of VL Decision Rules, *Sigart Newsletter,* 1977, **63**(1), 38-44.

[63] Kavurucu Y., An ILP-Based Concept Discovery System for Multi-Relational Data Mining, Doktora Tezi, Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2009, 255312.

[64] Leiva H. A., MRDTL: A Multi-Relational Decision Tree Learning Algorithm, Master's Thesis, Iowa State University, Iowa, USA, 2002.

[65] Gao Z., Zhang Z., Huang Z., Extensions to The Relational Paths Based Learning Approach RPBL, *First Asian Conference on Intelligent Information and Database Systems*, Quang Binh, Vietnam, 1-3 April 2009.

[66] Abay N. C., Mutlu A., Karagoz P., A Path-Finding Based Method for Concept Discovery in Graphs, *6th International Conference on Information, Intelligence, Systems and Applications*, Corfu, Greece, 6-8 July 2015.

[68] Fayyad U., Piatetsky-Shapiro G., Smyth P., From Data Mining to Knowledge Discovery in Databases, *AI magazine*, 1996, **17**(3), 1-37.

[69] Srinivasan A., King R. D., Bristol D. W., An Assessment of ILP-Assisted Models for Toxicology and the PTE-3 Experiment, Editors: Džeroski S., Flach P., *Inductive Logic Programming*, Springer-Verlag Berlin Heidelberg, Berlin, 291-302, 1999.

[70] Srinivasan A., King R. D., Muggleton S., Sternberg M., The Predictive Toxicology Evaluation Challenge, *Fifteenth International Joint Conference on Artificial Intelligence,* Nagoya, Japan, 23-29 August 1997.

[71] Srinivasan A., King R. D., Muggleton S., Sternberg M., Carcinogenesis Predictions Using ILP, Editors: Džeroski S., Lavrač N., *Inductive Logic Programming*, Springer-Verlag Berlin Heidelberg, Berlin, 273–287, 1997.

[72] Srinivasan A., King R. D., Muggleton S., Sternberg M., Mutagenesis: ILP Experiments in A Non-determinate Biological Domain, *4th International Workshop on Inductive Logic Programming*, Bonn, Germany, 12-14 September 1994.

KİŞİSEL YAYIN VE ESERLER

Göz F., Mutlu A., Concept Discovery in Graph Databases: A Case Study with Neo4j, *12th International Conference on Hybrid Artificial Intelligence Systems*, La Rioja, Spain, 21-23 June 2017.

ÖZGEÇMİŞ

Furkan Göz 1991 yılında Kocaeli'de doğdu. Lise öğrenimini İstanbul Köy Hizmetleri Anadolu Lisesi'nde tamamladı. 2009 yılında girdiği Anadolu Üniversitesi Bilgisayar Mühendisliği Bölümü'nden hazırlık eğitimi ile birlikte 2014 yılında mezun oldu. Aynı yıl Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü'nde yüksek lisans eğitimine başladı. Yüksek lisans eğitiminde çizge veri tabanları konusunda çalışmaları bulunmaktadır. Ayrıca, 2014 yılında Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde başladığı araştırma görevlisi görevini halen sürdürmektedir.