KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

TÜMEVARAN KAVRAM KEŞİF SİSTEMLERİ İÇİN TF-IDF TABANLI SEZGİSEL BİR YÖNTEM

CEMRE ONUR BAŞ

KOCAELİ 2020

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

TÜMEVARAN KAVRAM KEŞİF SİSTEMLERİ İÇİN TF-IDF TABANLI SEZGİSEL BİR YÖNTEM

CEMRE ONUR BAŞ

Dr. Öğr. Üyesi Alev MUTLU
Danışman, Kocaeli Üniversitesi .....

Doç. Dr. Sevinç İlhan OMURCA
Jüri Üyesi, Kocaeli Üniversitesi .....

Dr. Öğr. Üyesi Adem TUNCER
Jüri Üyesi, Yalova Üniversitesi .....

Tezin Savunulduğu Tarih: 27.01.2020

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, kavram keşfi problemi için tf-idf yöntemi ile sezgisel bir yöntem geliştirilmiştir. Yöntem, farklı öğrenme yöntemlerinde kullanılan veri kümeleri ile test edilmiştir.

Yüksek lisans öğrenimim boyunca bilgi ve tecrüberini benimle paylaşan, desteğini hiçbir zaman eksik etmeyen, her türlü problemde kendisine danışabildiğim saygıdeğer danışman hocam Dr. Öğr. Üyesi Alev MUTLU'ya teşekkürlerimi sunarım.

Tez çalışmam boyunca benden desteklerini eksik etmeyen, çıkmaza girdiğim zamanlarda destekleri ile arkamda duran arkadaşlarıma çok teşekkür ederim. Hayatım boyunca her türlü sıkıntımda arkadamda yer alan, varlığı ile rahatlatan, karşımıza çıkan her sıkıntıda beraber sırtlanarak üstesinden geldiğimiz ve her zamanda benim için yeri ayrı olacak olan sevgili annem Serpil ALPYÜREKOĞLU'na teşekkürlerimi sunarım.

Nisan – 2020 Cemre Onur BAŞ

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

ACCDB : Access Database (Access Veri Tabanı)
ACID : (Atomicity, Consistency, Isolation, Durability)
ALEPH : A Learning Engine for Proposing Hypotheses (Hipotez Önerisi için Öğrenme Motoru)
ALP : Abductive Logic Programming (Abdüktif Mantık Programlama)
API : Application Programming Interface (Uygulama Geliştirme Arayüzü)
C^2D : Condence-based Concept Discovery Method (Kurumsal Temelli Konsept Kesif Yöntemi)
CRIS : Concept Rule Induction System (Konsept Kural İndüksiyon Sistemi)
CQL : Cyhper Query Language
ÇİVM : Çok İlişkisel Veri Madenciliği
ILP : Inductive Logic Programming (Endüktif Mantıksal Programlama)
MIS : Model Inference System (Model Çıkarım Sistemi)
MRDM : Multi-Relational Data Mining (Çok İlişkili Veri Madenciliği)
NLP : Natural Language Processing (Doğal Dil İsleme)
NoSQL : non SQL (SQL dışı)
pCRIS : Parallel Concept Rule Induction System (Paralel Konsept Kural İndüksiyon Sistemi)
RDF : Resource Description Framework (Kaynak Açıklama Çerçevesi)
RDMS : Reliable Database Manager System (Güvenilir Veri Tabanı Yönetici Sistemi)
RLGG : Relative Least General Generalisations (Göreceli En Az Genellemeler)
TAL : Top-directed Abductive Learning (En Yönlendirilmiş Kaçırıcı Öğrenme)
TF-IDF : Term Frequency — Inverse Document Frequency
TMP : Tümevaran Mantıksal Programlama

TÜMEVARAN KAVRAM KEŞİF SİSTEMLERİ İÇİN TF-IDF TABANLI SEZGİSEL BİR YÖNTEM

ÖZET

Kavram keşif sistemleri, hedef ilişki olarak adlandırılan bir ilişkiyi bu ilişki ile doğrudan veya dolaylı olarak ilişkili arkaplan verisi olarak adlandırlan ilişkiler aracılığıyla tanımlayan modelleri arar. Tümevaran tabanlı kavram keşif sistemlerinde, sadece bir doğru hedef örneği açıklayan doymuş bir alt kuraldan başlanılarak ve yinelemeli olarak bu kuralı genelleştirilerek olabildiğince çok doğru hedef örneği ve olabildiğince az yanlış hedef örneğini açıklayan modeller oluşturulur. Fazla sayıda doğru hedef örneği olan veri kümelerinde, doymuş alt kural oluşturulacak hedef örneği seçmek, elde edilecek hipotezin kapsayıcılığını belirleyeceği için, önem arz etmektedir.

Bu çalışmada, tümevaran kavram keşif sistemlerinde doymuş alt kuralı oluşturmak için tf-idf tabanlı sezgisel bir yöntem önerilmektedir. Önerilen yöntemde veriler, hedef ilişki ve arkaplan verisi örneklerinin düğümleri, kenarların ise hedef ilişki örnekleri ile ilgili arkaplan örneklerini bağladığı bir çizge şeklinde temsil edilmektedir. Her hedef ilişki için biri doğru hedef örnekleri diğeri de yanlış hedef örnekleri modelleyen iki çizge şeklinde temsil edilmiştir. Her çizge örneklerin olası tüm genelleştirmeleri ile zenginleştirilmiştir. Her düğüm için tf-idf hesaplanmıştır. En yüksek tf-idf değerli düğümler birleştirilerek doymuş alt kural oluşturulmuştur. Elde edilen doymuş alt kural olabildiğince çok doğru hedef örneği ve olabildiğince az yanlış hedef örneği açıklayacak şekilde genelleştirilmiştir. Önerilen yöntemin başarısı 10-katlı çapraz doğrulama ile mutagenesis isimli biyokimyasal veri kümesi kullanılarak değerlendirilmiştir. Her katta, verilerin %90 eğitim %10'u da test için kullanılmıştır. Tablo 4.9 ve Tablo 4.12'de sonuçlara ait karmaşıklık matrisi verilmiştir.

Deney sonuçları 0,94 doğruluk, 0,96 hassasiyet ve 0,88 yanlış tahminleme oranına sahiptir. Elde edilen kavram tanımlarına incelendiğinde çok genel kuralların çözüm kümesine eklenmesini engelleyeci tedbirlerin alınmasını gerekliliği görülmüştür.

Anahtar Kelimeler: Kavram Keşfi, Sezgisel Yöntem, Tf-Idf.

A TF-IDT BASED HEURISTIC FOR BOTTOM-UP CONCEPT DISCOVERY SYSTEMS

ABSTRACT

Concept discovery systems look for patterns that explain a relation called target relation by means of its directly or indirectly related relations, called background knowledge. Bottom-up concept discovery systems start with building a saturated bottom clause and iteratively generalize it to cover as many positive target instances and as few negative instances as possible. In case of large population of positive target instances, choosing a target instance to build a bottom clause for becomes crucial as the bottom clause affects the overall coverage.

In this study, we propose a tf-idf based heuristic for building a bottom clause for bottom-up concept discovery systems. In the proposed method, data is represented as a graph where nodes represent facts and target instances and edges connect facts to target instances. For each target relation, two graphs, one representing the positive target instances and the other representing the negative target instances are built. Each graph is enhanced with all possible generalizations of the facts. For each node, tf-idf weight is calculated. To build a bottom clause, nodes with highest tf-idf weights are conjoined, i.e. logically anded, and the resulting bottom clause is generalized to cover as many positive target instances as possible and as few negative target instances as possible.

To evaluate performance of the proposed method a 10-fold experiment is conducted on a biochemical data set called mutagenesis. At each fold, 90% of the target instances are used to build the model and the remaining 10% of the instances are used to validate the model. In Table 4.9 and Table 4.12, we represent the confusion matrix. At each fold, a number of concept descriptors are obtained describing positive and negative target instances, and in validation step majority voting principle is employed.

The experimental results indicate 0.94 accuracy, 0.96 sensitivity, and 0.88 negative prediction rate. Analysis on the induced concept descriptors suggests to implement mechanism to prevent the inclusion of overly general concept descriptors in the solution set.

Keywords: Concept Discovery, Heuristic, Tf-Idf.

GİRİŞ

Verilerin varlık-ilişki modeli çerçevesinde birden fazla tabloda saklanmaya başlaması ile çok ilişkisel veri madenciliği (ÇİVM) kavramı ortaya çıkmıştır. Klasik veri madenciliği teknikleri sadece tek tabloda saklanan verilerden örüntüler çıkarmaya çalışırken ÇİVM algoritmaları birden fazla tabloda saklanan verilerden örüntüler çıkarmaya odaklanmıştır.

ÇİVM'nin en önemli problemlerinden biri de kavram keşfidir. Kavram keşfi, hedef ilişki olarak adlandırılan bir veri tabanı ilişkisinin arka plan verisi olarak adlandırılan diğer veri tabanı ilişkileri kullanılarak tanımının öğrenilmesi problemidir. Bu problem için ilk çözümler mantık programlama tabanlı olarak sunulurken son zamanlarda ise çizge teorisine dayalı yöntemler ön plana çıkmıştır.

Kavram keşfi için çizge tabanlı yöntemler, ilişkisel verinin çizgeler ile gösterilmesine ve çizge algoritmaları kullanılarak örüntülerin bulunmasına dayanmaktadır. İlişkisel verinin çizge ile modellenmesinde düğümler varlıkları kenarlar ise varlıklar arası ilişkiyi gösterir. Örüntülerin çıkarılması ise yol-bulma veya altçizge-bulma yaklaşımlarına dayanır. Yol-bulma tabanlı yaklaşımlarda hedef ilişkiye ait örneklerden biri ile başlayan sonlu uzunluklu yollar bulunur ve bu yollardan en çok hedef ilişki örneğini açıklayan yollar kavram tanımları olarak kabul edilir. Altçizge-tabanlı yaklaşımlarda ise hedef örnekleri içeren sık altçizgeler kavram tanımı olarak kabul edilir. Her iki yaklaşımda da temel problemlerinden biri örüntünün (yol veya altçizge) bulunması için hedef ilişkiye ait hangi örnekten başlanacağı problemidir.

Bu çalışma kapsamında kavram keşfi için altçizge-tabanlı sistemlere odaklanılmış ve başlangıç öreneğini seçmek için tf-idf tabanlı bir yaklaşım geliştirilmiştir. Önerilen yöntemde, iki farklı veri kümesi üzerinde deneyler yapılmıştır. Modelin son hali ile yapılan deney sonuçlarında pozitif veri kümelerinin yeterli bir başarıyla ayrıştırılması ve doğruluk (accuracy) değerlerinin yeterli çıkmasına ancak modelin aşırı uyum gösterme (overfitting) problemine maruz kaldığı tespit edilmiştir.

1. GENEL BİLGİLER

1.1. Tez Çalışmasının Amacı ve Başlatılma Sebebi

Teknolojinin gelişmesi ile beraber veriler önemli bir yer haline gelmeye başlamıştır. Bilgi ve bilgiye duyulan ihtiyacın artması sebebi ile de veri miktarında artış meydana gelmiş ve işlenmemiş bu veriler birer veri yığınları haline gelmeye başlamıştır. Bu da verilerin artık yönetilememesi gibi birtakım problemlere yol açmıştır. Bu problemler bazı alanlarda yeni yöntemlerin ortaya çıkmasına sebep olmuştur. Bu problemlerden bir tanesi de kavram keşfidir. Kavram keşfinde problemlerden birisi ölçeklenebilirliktir. Ölçeklenebilirlik veri miktarı ile ters olduğu için miktarın artması gibi durumlar ölçeklenebilirliğin azalmasına sebep olmaktadır.

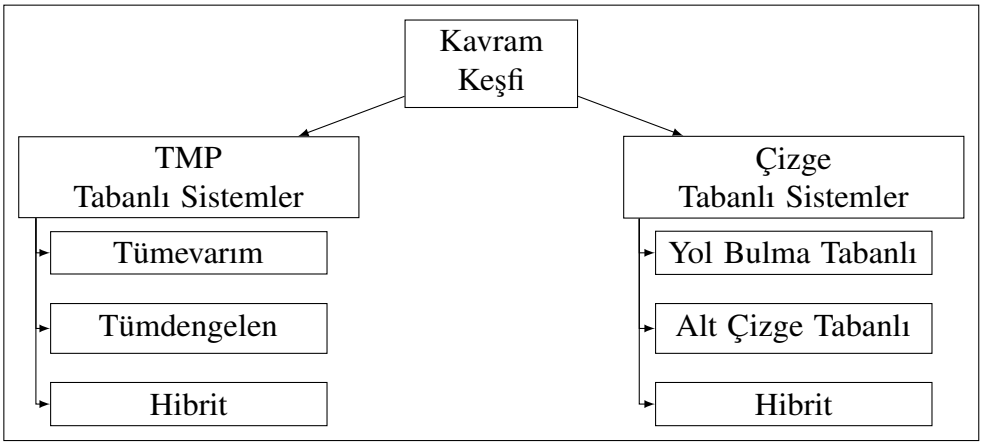
Bu çalışmada tf-idf ile çizge tabanlı sezgisel bir tümevarım yöntemi kullanılarak hedef veri tabanlarındaki ilişkilerde yer alan verilerin kullanım sıklığına göre yorumlama getirilerek hedef ilişkilerinin açıklanması amaçlanmıştır. Bu formatla ilişkiler klasik veri tabanından çizge tabanlı bir yapıya aktarılması düşünülmüştür. Bu sayede çok sistemli bu ilişkilerin birbirleri arasındaki yapıyı bağlamak ve yorumlamanın kolaylık sağlayacağı kanısına varılmıştır. Bunun yanında hedef ilişkileri destekleyen verilerin (düğümlerin) her iterasyonda elemine edilerek kuraldan çıkarılarak pozitif ve negatif olarak adlandırılan iki veri uzayının birbirinden özerklik olarak ayrılması hedeflenmiştir.

1.2. Tez Çalışmasının Katkıları

Bu çalışma aşağıda verilerinlere katkı sağlamayı amaçlamıştır.
Çizge tabanlı veri tabanı kullanılarak tf-idf ile sezgisel bir yöntemle gerçekleştirilmesi, Farklı sistemlerin farklı özellikleri ile önerilen yöntemin uygulanabilirliği, tf-idf yönteminin tümavarım yöntemi ile entegre edilebilirliği, Birtakım programlama tekniklerinin kullanılması ile performans olarak artış sağlaması ve arama uzayının daraltılması, Kavram keşfi problemine yeni bir yöntem önermesi

1.3. Literatür Taraması

Bu bölümde literatürde kavram keşfi ile ilgili yapılan çalışmalar sunulmuştur. Şekil 1.1'de kavam keşfinin sistematik olarak dağılımı verilmektedir.



Şekil 1.1. Kavram keşfi yöntemleri

Literatürde TMP tabanlı sistemlerde Kavram keşif problemine çözüm getirmek ve sınıflandırma yapmak için kullanılan yöntemlerden biri ILP (Inductive Logic Programming) sistemidir. Sınıflandırma da verilere göre genel kurallar oluşturulur ve ardından sınıflandırılmamış verileri gruplamak için kullanılır. Konsept keşfinde, sistem kullanıcılarına eğer varsa ilginç kurallar verilir. Çeşitli araştırma sınıfları, sezgisel tarama ve dil örüntüsü sınırlamaları kullanan çeşitli ILP tabanlı sistemler geliştirilmiştir. Geliştirilen bu sistemlerden en yaygın olarak bilinen yöntem LINUS yöntemidir. LINUS ilişkisel veri tabanında depolanan klasik özellik-değer ilişki içerisinde yer alan veriler ile LIP öğrenme sistemi içeren bir sistemdir. Burada hedef ILP problemi sınıfını önermeli forma dönüştürmek ve dönüştürülmüş öğrenme problemini bir nitelik-değer algoritmasıyla çözmektir. LINUS'un şu andaki dağılımı, ASSISTANT, NEWGEM ve CN2 özellik değer öğrenicilere arayüzler sunmaktadır. LINUS iki modda çalıştırılabilir. CLASS modunda çalışırken, gelişmiş bir özellik-değer öğrenicisine karşılık gelir. RELATION modunda, LINUS bir ILP sistemi gibi davranır.LINUS'un yanı sıra GOLEM, CIGOL, MIS, FOIL, PROGOL, ALEPH ve WARMR gibi sistemlerde vardır [1, 2].

FOIL olarak, GOLEM ampirik ILP sistemleri arasında bir "klasik" dir. Protein yapısı tahmini ve sonlu elemanlar ağı tasarımı gibi gerçek dünyadaki problemlere başarıyla uygulanmıştır. GOLEM, büyük veri kümeleriyle verimli bir şekilde baş edebilir. Örneğin, FOIL gibi tutarlı hipotezler için geniş bir hipotez alanı aramaktan kaçınır, ancak mevcut geçmiş bilgisine göre bir dizi pozitif örnek içeren benzersiz bir cümle kurar. İlke, Plotkin tarafından tanıtılan göreceli en az genel genellemelere (RLGG) dayanmaktadır. GOLEM, RLGG inşasını bir kaplama yaklaşımına dahil ediyor. Tek bir cümlenin uyarılması için, rastgele birkaç pozitif örnek çifti seçer ve sıralarını hesaplar. Bu hizmetler arasında, GOLEM en fazla sayıda pozitif örneği kapsayan ve negatif örneklerle uyumlu olanı seçer. Bu madde daha sonra genelleştirilir. GOLEM rastgele bir dizi pozitif örnek seçer ve bu örneklerin her birinin bölümlerini ve ilk yapım

aşamasında elde edilen maddeyi kurar. Yine, en geniş kapsama sahip rlgg, aynı işlem tarafından seçilir ve genelleştirilir. Genelleme işlemi, en iyi fıkranın kapsamı artmayı durdurana kadar tekrar edilir. GOLEM, ilgisiz değişmezleri kaldırarak indüklenen cümleleri azaltan bir postprocessing adımı uygular. Genel durumda, RLGG sonsuz sayıda değişmez içerebilir. Bu nedenle, GOLEM, RLGG uzunluğunun pozitif örnek sayısı ile polinom olarak büyümesini sağlayan arka plan bilgisi ve hipotez dili hakkında bazı kısıtlamalar getirmektedir. GOLEM'in temel bilgisinin temel gerçeklerden oluşması gerekir. Hipotez dili için belirleme kısıtlaması geçerlidir, yani, bir cümlenin baş değişkenlerinin değerleri için, beden değişmezlerinin argümanlarının değerleri benzersiz bir şekilde belirlenir. GOLEM'in hipotez dilinin karmaşıklığı, bir hipotez cümlesinde vücut değişkenlerinin sayısını ve derinliğini sınırlayan i ve j iki parametresi ile kontrol edilir. GOLEM, Golem functors ile Horn cümleleri öğrenir. GOLEM yalnızca olumlu örneklerden öğrenebilir. Olumsuz örnekler, işlem sonrası adımda yan tümce azaltmanın yanı sıra, kullanıcının isteğe bağlı olarak sağlayabileceği öngörüler için giriş / çıkış modu bildirimlerinde kullanılır. Gürültülü verilerle ilgilenmek için GOLEM, kullanıcının bir hipotez maddesinin kapsayabileceği maksimum negatif örnek sayısını tanımlamasını sağlayan bir sistem parametresi sağlar.

PROGOL, Stephen Muggleton'un bilgisayar biliminde kullanılan ters yerleştirme ve genelden özele aramayı bir birleştirme çizgesi ile birleştiren endüktif mantık programlaması uygulamasıdır. PROGOL, hipotez açıklama uzunluğuna karşı hataların açıklamasını dengelemek için "sıkıştırma ölçüsü" kullanarak gürültülü verilerle ilgilenir [3].

Ehud Shapiro'un MIS'i örneklerin olgulardan (facts) oluşan bir öğrenme ortamında öğrenimini gerçekleştirir. Buna karşın eksik bilgiler ile başa çıkabilmek için kullanıcıdan gerçek-değer ilişkisi için talepte bulunarak eksik bilgileri sorar ve bu eksik bilgilerin cevabı, MIS'in olumlu örneklerin ispatının yeniden oluşturulmasına izin verir. MIS, PROGOL'un aksine; A*'e benzer bir arama algoritma ile arama uzayında maksimum sıkıştırma garantisi verebilmektedir.

ALEPH sistemi, ILP'deki fikirleri keşfetmek için bir prototip olarak geliştirildi ve Prolog'da yazıldı [4]. Aleph, çeşitli ILP sistemlerinden gelen işlevleri kullanır: Progol, FOIL, FORS, Indlog, MIDOS, SRT, Tilde ve WARMR. ALEPH, karmaşık ifadeleri temsil etmeyi ve aynı anda yeni arka plan bilgisini kolayca dahil etmeyi sağlayan güçlü bir temsil diline sahiptir. Ayrıca kuralların oluşturulma sırasını seçmesine, değerlendirme işlevini ve arama sırasını değiştirmesine izin verir. Tüm bu özelliklere bağlı olarak ALEPH sistemi, tüm ILP araştırmacıları için güçlü bir kaynak olmasını sağlayan açık kaynaktır.

WARMR, Apriori kuralını arama sezgiselliği olarak kullanan açıklayıcı bir ILP sistemdir. Hem verileri hem de desenleri temsil etmek için veri günlüğünü (Datalog) kullanır. Datalog, özellikle tümdengelimli veritabanlarını uygulamak için özel olarak tasarlanmış bir mantık programlama dilidir. WARMR, kalıpların, birkaç tablonun bire-çok ve çoktan çoğa ilişkilerini yansıttığı yapılandırılmış veri hakkında bilgi keşfedebilir. Bu, standart veri madenciliği programlarında mümkün değildir. Arka plan bilgisi tek tip bir şekilde temsil edilir ve çoğu veri madenciliği ayarından farklı olarak, sık kullanılan modellerin keşfedilmesinde önemli bir role sahiptir [5].

TMP tabanlı sistemlerde ölçeklenebilirlik, yerel plato problemi ve yavaş öğrenme gibi problemler yer almaktadır. Bu problemlere çözüm getirmek için kavram keşfinde Çizge tabanlı sistemler geliştirilmiştir. Çizge tabanlı sistemlerde yol bulma ve alt çizge tabanlı sistemler literatürde yer alan yöntemlerdir. Alt çizge sistemlerinde kavram keşfinde hedef örnekte ilk olarak benzer alt graflar aranır. Bulunan benzer özellikli düğümler tek bir düğüm altında ortak olarak temsil edilir. Bu yöntem tüm alt graflar taranıncaya kadar devam eder. Yol bulma tablı yaklaşımlarda ise mevcut sistemin özellikleri arasında gezinerek önceden belirlenen bir uzunluk değerine sahip özellik kurala aday olarak gösterilir. Önceden belirlenen minimum değeri sağlayan bu aday kurallar karam tanımları olarak alınır.

Literatürde kavram keşfi problemi ile ilgili çeşitli çalışmalar yürütülmektedir. [6]'da kavram keşfinin ölçeklenebilirlik problemi için ILP tabanlı sistemlerde bir paralelleştirme yöntemi önerilmektedir. Bu sadece dil kalıplarındaki sınırlamaları bir kenara atarak zaman ve ölçeklenebilirlik anlamında performans açısından bir artış sağladığı savunulmuştur. Bu çalışmada paralel olmayan bir sistem olan CRIS (Konsept Kural İndüksiyon Sistemi)'in yüksek miktarda darboğaza neden olan sorgu işleme parçalarının paralelleştirilmesi gerçekleştirilmiştir. Ortaya çıkan yeni sistem pCRIS (Paralel Konsept Kural İndüksiyon Sistemi) olarak adlandırılır. CRIS, verilen hedef ilişkisine ve arka plan bilgisine göre sık ve güçlü konsept tanımları bulmak için ilişkisel birlik kuralı madenciliği kavram ve tekniklerini kullanan bir tahminci öğrenme sistemidir. CRIS'in temel özellikleri, doğrudan ilişkisel veritabanları üzerinde çalışmayı, mod bildirimleri gereksinimini ortadan kaldırmayı ve yeni arama alanı budama yöntemlerini ve ölçümlerini kullanmayı içerir. CRIS'de en çok zaman alan işlemler arama alanı oluşturma ve arama alanı değerlendirme ve budamadır. Arama alanı değerlendirme ve budama adımında olası cümlecikler SQL sorgularına çevrilir ve cümlelerin destek ve güven değerlerini hesaplamak için SQL sorguları çalıştırılır. Kullanıcı tanımlı eşik değerinden daha düşük destek değerine sahip üretilen tümceler, nadir olmayan tümceler olarak budanır. Sık kullanılan maddeler, güven değerlerine

dayanarak ayrıca değerlendirilirler: daha fazla araştırma için belirlenen aday maddelere güçlü olmayan maddeler eklenir ve olası çözümler setine güçlü maddeler eklenir. Arama alanı değerlendirme ve budama adımının en çok zaman alan kısmı SQL sorgularının yürütülmesidir. Budama süresi, SQL yürütmeleri için gereken süreye kıyasla önemsizdir. Arama alanı oluşturma adımı (x+1) 'in olası tümcelerini oluşturmak x'in birleştirilemez aday maddelerini anlamdırmak ile ilgilidir. CRIS'in paralel versiyonu, genel sistemin zaman verimliliğini artırmak için bu iki adımı paralelleştirir.

Hibrit bir yöntem ile çizge tabanlı yaklaşımı birleştiren karma bir yaklaşım sunulmaktadır. Önerilen yöntemde ilişkisel formatta girilen veriler çizge gösterimine dönüştürülür ve kavram tanımlayıcılarını bulmak için çizgeyi gezinir. Çizge geçişi ve budama, birliktelik kural madenciliği tekniklerine dayanarak yönlendirilir. Önerilen yöntem n-ary ilişkilerine sahip gelişmiş tekniklerinden ayırt edilmektedir, kuralları çıkarmak ve sayısal değerleri için yol bulma tabanlı sorgular kullanmaktadır [7].

Bir başka çalışmada Brave Induction adlı kavram öğrenimi için yeni bir mantıksal çerçeve sunmaktadır. Cesur indüksiyon indüksiyon için cesur çıkarım kullanır ve eksik bilgilerden öğrenmek için yararlıdır. Cesur indüksiyon, normalde endüktif mantık programlamasında kullanılan açıklayıcı indüksiyondan daha zayıftır ve clausal mantıkta genel bir kavram-öğrenme ortamı olan tatmin edilebilirlikten öğrenmekten daha güçlüdür. Yapılan çalışmada cesur indüksiyonun resmi özellikleri araştırılıp, sonra tam clausal teorilerde hipotezleri hesaplamak için bir algoritma geliştiririz. Ardından, çerçeveyi nonmonotonik mantık programlarında indüksiyona genişletilmektedir. Önerme teorileri üzerine indüksiyon için karar problemlerinin hesaplama karmaşıklığını analiz edilip sistem biyolojisinde gereksinim mühendisliği ve çok yönlü müzakere ile problem çözme örnekleri sunulmaktadır [8].

Bir başka belirtilen yöntemde monotonik olmayan ILP'ye ve TAL adlı uygulamasına yeni bir yaklaşım sunulmaktadır. TAL, ILP sistemlerinin tam tersine dayalı bazı sorunların üstesinden gelebilir ve ters teori ve hipotezlerin normal mantık programları olmasını sağlayan ilk yukarıdan aşağıya ILP sistemidir. Yaklaşım, bir ILP sorununu eşdeğer bir ALP ile eşleştirmeye dayanır. Bu, yerleşik ALP kanıt prosedürlerinin kullanılmasını ve daha zengin dil önyargılarının bütünlük kısıtlamaları ile belirtilmesini sağlar. Haritalama, bir ILP problemi için, üzerinde endüktif çözümleri hesaplamak için bir kaçıcı arama kullanılan bir ilkeli arama alanı sağlar [9].

Semantik Web ve Yaşam Bilimi çıkarım servis sistemi olan Prova'nın konu olarak yer aldığı bir başka çalışmada, karmaşık biyolojik ilişkiler gibi karmaşık Yaşam Bilimleri alanlarının çok ilişkisel veri madenciliği için mimarisi olan Prova'nın indüktif mantık

programlama (ILP) özellikleri açıklanmaktadır. Önerilen yeni tasarımda, kural tabanlı genelleme ve uzmanlaşma için tipik ILP çıkarım formalizmlerini uygular ve arama alanını ve genel düzeyini sınırlandırmak için, kapsamlı meta-veri temelli mantık ve yazılı mantık gibi etkileyici mantık temelli formalizmlerle birleştirmektedir. Çok ilişkisel veri madenciliği için ILP'nin yüksek ifade edilebilirliğini ve esnekliğini korumakta ve ILP'nin yayınlanmış çok büyük ve dağınık heterojen miktardaki verilerle karşı karşıya kaldığında ILP'nin bilinen hesaplama ve mantıksal sorunlarını aşmaya çalışmaktadır [10].

Tf-idf, bir kelimenin doküman setindeki bir doküman için ne kadar önemli olduğunu gösteren bir ölçümdür. Otomatik metin düzenleme gibi birçok önemli alanda kullanımı vardır ve Doğal Dil İşleme (NLP) için makine öğrenme algoritmalarındaki kelimeleri puanlamak için çok kullanışlıdır. Literatürde tf-idf ile ilgili çeşitli çalışmalar bulunmaktadır.

Anahtar sözcük çıkarma üzerine yeni bir yöntem sunulan başka bir çalışmada geleneksel tf-idf algoritmasında çıkartılan anahtar kelimeler, çoğunlukla kelime sıklığına göre hesaplanır. Daha az tekrarlanan diğer özellik kelimelerinin önemi ve makalenin altındaki okuyucuların yorumları dikkate alınmaz. Bu problemleri hedef alan önerlen yöntemde geleneksel tf-idf algoritmasını geliştirir, konuşmanın bir kısmını ve okuyucunun yorumunu etki faktörü olarak ekler ve algoritmanın doğruluğunu artırmak için tf-idf'nin ağırlığını yeniden hesaplar. Sonuçlara bakıldığında tf-idf algoritmasının (önerilen yöntemdeki), geleneksel tf-idf ile karşılaştırıldığında doğruluk, geri çağırma hızı, F1, MacAvg_P, MacAvg_R ve MacAvg_F1 açısından önemli ölçüde geliştiğini göstermektedir [11].

Diğer bir çalışma da ise alan bilgisine dayalı bir yöntem önerilmektedir. Önerlien yöntem ile tf-idf algoritmasında iyileştirmeler yapılmaktadır. Önerilen yöntem yasal alan uygulamalarını temel alarak metin özelliği çıkarımında etki alanına ilişkin anahtar kelimelere atanan makul ağırlığı elde etmek için tf-idf algoritmasında iyileştirmeler yapmaktadır [12].

Bilgisayardan otomatik metin soyutlamanın doğrulunu sağlayabilmek için TextRank ile tf-idf algoritmasını birleştiren ve geliştiren yorum faktörüne, pozisyon faktörüne ve konuşma kısmı faktörünün ağırlıklandırmasına dayalı bir ağırlıklandırma yöntemi tasarlanmıştır. Örnerilen yöntemde okuyucunun yorumlarını bir yorum faktörü olarak sunmaya, orijinal metne çok fazla dikkat etmemenin eksikliklerini tamamlamaya ve okuyucunun yorumlarını görmezden gelmeye odaklanır [13].

Bir başka çalışma da TA Tf-idf adı verilen rafine bir algoritma sunulmaktadır. Önerilen algoritma zaman dağılımı bilgisine ve kullanıcının dikkatine dayanarak sıcak terimler bulmayı hedeflemektedir. Ayrıca, ilgili çalışmada Çince kelime segmentasyon algoritmasıyla ayrılan yeni terimler ve birleşik terimler üretmek için bir yöntem öne sürülmüştür [14].

Bu tez çalışmasında ise tf-idf ile sezgisel bir yöntem önerilmiştir. Yöntemin her iterasyonunda hedef ilişkilerin frekans karşılaştırması yapılarak (y+1). iterasyon için minumum destek ve güvenilirlik değerlerine bakılarak çıkarımlarda bulunulmuştur. Bu sayede anlamlandıramayan veri yığınının önüne geçilerek pozitif ve negatif veri uzayı olarak kabul edilen iki ayrı veri kümesinin birbirlerinden bağımsız olarak özerkleşip ayrılmalarına olanak sağladığı savunulmaktadır.

1.4. Tezin Yapısı

Bu tez çalışması 5 bölümden oluşmaktadır. Bölüm 1 problemin tanımı ve tezin genel bilgilerinin bulunduğu kısımdır. Bu bölümde çalışmanın genel olarak tanıtılması, çalışmanın amacı, başlatılma sebebi, literatür çalışması anlatılmaktadır. Bölüm 2'de temel kavramlar açıklanmıştır.Bu bölümde kavram keşfi ve önerilen yöntem olan tf-idf hakkında bilgiler verilmiştir. Bölüm 3 tezin uygulaması olarak sunulan tf-idf yöntemi açıklanmıştır. Kavaram keşfi için tf-idf yapısının nasıl kurulduğu algoritma ve sözde kodlar yardımıyla anlatılmıştır. Bölüm 4'te ise çalışmanın deneysel sonuçlarından oluşmaktadır. Bu bölümde veri kümeleri için elde edilen kavram tanımları ve çalışma süreleri verilmişltir. Ayrıca sonuçlar alanında bulunan çalışmalarla kıyaslanmış ve yorumlanmıştır, algoritmanın performans ve ölçeklenebilirliği hakkında araştırma sonuçları bu bölümde açıklanmıştır. Çalışmanın son kısmı olan Bölüm 5'te ise önerilen yöntemin deneysel sonuçlarına göre genel değerlendirmelere yer verilmiştir.

2. TEMEL KAVRAMLAR

2.1. Kavram Keşfi

Veri madenciliği çalışma konularından biri anlamdırmamış veri yığınlarından faydalı bilgilerin çıkarıldığı ve birçok alanda da kullanılabilen bilgi keşfidir. Veri yığınlarının anlamdırılması sırasında geleneksel tek bir veri tablosundan ilişki kurmak yerine çok ilişkili veri madenciliği ile bu işi yürütmektedir [15].

Kavram keşif probleminde ilişkisi verilen bilgilerin argümanlar ile arasındaki bağlantıyı temsil eden hedef ilişki ve bu ilişkiler arasındaki ilişkiyi açıklama için arka plan bilgisi olarak adlandırılan bilgiler kullanılmaktadır. Bu bilgiler kullanılarak hedef ilişkiler bulunmaya çalışır. Bulunan bu ilişkiler herhangi bir ölçüm sistemi tarafından onaylanmadığı için aday kavram tanımı olarak gösterilir. Aday tanımlar berlirli eşik değerlerini geçmesi sonucunda kurala dahil edilmiş kavram tanımları haline gelmiş olur.

Kavram keşfi problemlerinde kullanılan veri tabanlarından birisi olan Mutagenicity'de biyokimyasal tabanlı mutajenik bileşiklerin kanserojen etkileri araştırılmaya çalışılır. Bu veri kümesi için örnek hedef ilişki ve arka plan bilgilerini gösteren örnekler Tablo 2.1'de verilmiştir. Tablo 2.1'de hedef örnekte mutanın argümanlarından birinicisi olan d1, ikinci olan true olması d1 ilacının mutajenik (kanserojen etkisi olan) olarak temsil edilmektedir.

Tablo 2.1. Mutagenicity örnek bilgiler

Hedef İlişki	**Arka Plan Bilgileri**
muta(d1,true)	ind1(d1,1) inda(d1,0) lumo(d1,lumo1) logp(d1,logp1) muta_bond(d1,d1_6,d1_10,1) muta_atm(d1,d1_6,c,22,charge1)

İkinci argümanı false olan bir ilaçta ise ilgili ilacın mutajenik olmayan (kanserojen etkisi olmayan) olarak temsil edildiği görülmektedir. Tablo 2.2'de arka plan bilgileri kullanılarak bulunmuş aday kavram tanımlarına ait açıklanan özellik sayısının değerleri verilmiştir. Mutagenicity veri kümesine göre elde edilen kavram tanımlarında açıkça farklı değerlikli sonuçlar elde edildiği ve elde edilen kavram tanımlarının birden fazla

özelliğe açıklık getirdiği görülmektedir. Tablo 2.2'de yer alan kavram tanımları, Mutagenicity veri kümesinin örnek bir eğitim verisinin sonuçlarının bir kısmını yansıtmaktadır.

Tablo 2.2. Mutagenicity veri kümesindeki aday kavram tanımları örneği

Aday Kavram Tanımları	**Açıklanan Özellik Sayısı**
atom : c bond : 1 lumogroup : lumo2[min :-2.306 , max : -2.155]	21
atom : c bond : 1 lumogroup : lumo1[min :-3.768 , max : -2.338]	18
charge : 22 bond : 1 lumogroup : lumo1[min :-3.025 , max : -2.005]	13
charge : 22 bond : 1 lumogroup : lumo4[min :-1.665 , max : -1.59]	12

Tablo 2.2'de yer almakta olan aday kavram tanımlarının belirli ölçütler uygulandıktan sonra eşik değerlerini sağlaması halinde kavram tanımı olarak yer alacaklardır.

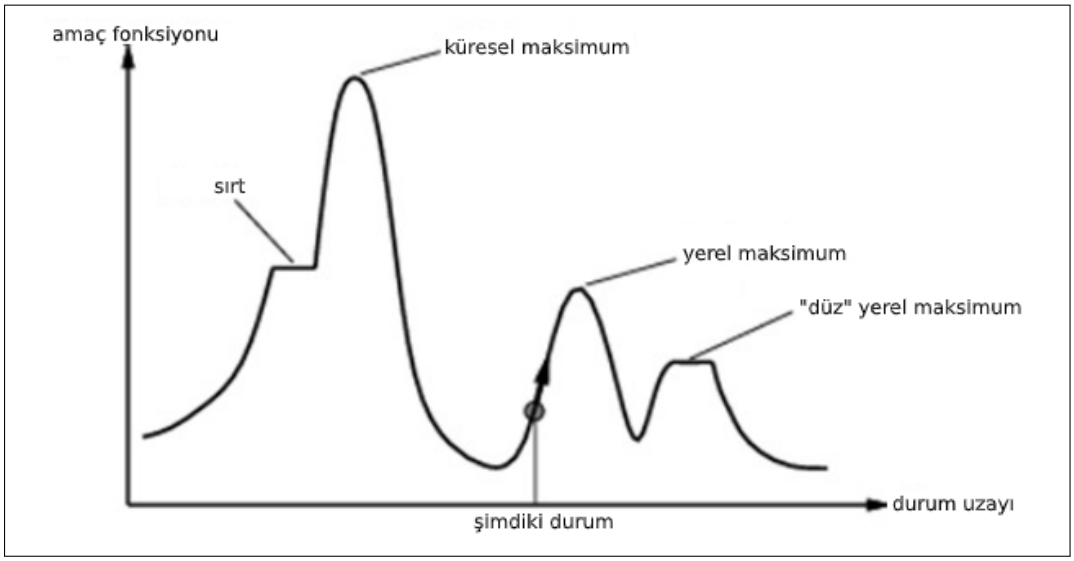
Tablo 2.2'de bulunan Mutagenicity veri kümesinin minimum destek (support) ve güven (confidence) olarak adlandıran ölçütler ile kontrolleri yapılarak hedef özelliklerin ortaya çıkması sağlanır. Tablo 2.3 için yapılan ölçütler sonucu kavram tanımlarında ortaya çıkan ölçüt değerleri verilmektedir.

Tablo 2.3. Mutagenicity veri kümesindeki kavram tanımları örneği

Kurallar	**Minimum Destek Değeri**	**Güven Değeri**
atom : c bond : 1 lumogroup : lumo2[min :-2.306 , max : -2.155]	0.168	1.0
atom : c bond : 1 lumogroup : lumo1[min :-3.768 , max : -2.338]	0.173	1.0
atom : c bond : 1 lumogroup : lumo4[min :-1.538 , max : -1.894]	0.082	0.6

Bu tabloya göre bazı aday kurallar minimum destek deveri ve güven değerinin altında kalması bu kuralların kavram kuralı olarak dahil edilmemesi ve elenmesi anlamına gelmektedir. Yapılan deneylerin birinde 127 ilaç örneğinin 121'i bir diğerinde ise 64'ü açıklanmıştır. Bu iki deney sonucunun kendilerine göre artı ve eksi yönleri bulunmaktadır. Deney yapılırken kullanılan giriş argümanları ve çıkış sonuçları Bölüm 4'ta delaylı olarak anlatılmaktadır.

Kavram keşfinde en istenmedik problem yerel plato problemi olarak isimlendirilen problemdir. Bu problemde önerilen model belirli bir aşamaya geldikten sonra mevcut sistemde ne kadar fazla aday kavram tanımı eşik değerini geçip kavram tanımı haline gelsede Mutagenicity veri tabanından örnek verilirse pozitif ve negatif değerler özerklik olarak birbirinden ayrılamamaktadır. Aday kavram tanımlarına yeni bir argüman dahil edilmesine rağmen sisteme bir katkısı görülmediği anlaşıldığında sistemin yerel plato problemi içerisine girdiği anlamına gelmektedir. Şekil 2.1'de bu problem çizgesel olarak resmedilmiştir.



Şekil 2.1. Yerel plato problemi grafiği

Çizge tabanlı yaklaşımda kavram keşfinde kullanılan bir yöntemdir. Genellikle yol bulmak ya da alt çizge tabanlı sistemler olarak kullanılır. Alt yapı tabanlı sistemlerde hedef ilişkileri açıklayan birbirlerine benzer ortak alt çizgeler tespit edilmeye çalışır [16]. Yol bulma tabanlı sistemlerde ise belirli uzunlukta yollar aranır ve hedef ilişkileri açıklayan yollar kavram tanımı olarak sisteme dahil edilir.

2.2. Çizge Veri Tabanı

Veri tabanları verilerin sistematik, düzenli ve belirli bir hiyerarşi ile depolanmasına olanak tanıyan verileri bir tablo halinde tutulduğu sistemlerdir. Veri tabanları çeşitli sorguların yapılmasına imkan sunar. Geleneksel olarak veri tabanları ACID (bölünmezlik, tutarlılık, izolasyon, dayanıklılık) adı verilen işlemleri desteklemek için normalleştirildiği ilişkisel modelle tasarlanmıştır. Veri normalleştirme işlemi, veri tabanındaki yinelenen verileri kaldırır. Veri normalleştirmenin amacı veri tutarlılığını korumaktır. İlişkisel model, verileri birçok tabloya ayırarak ACID işlemlerini zorlar. İlişkisel modeller incellendiğinde tutarlılığı garanti etmek için ağır veri normalizasyonu uygular. İlişkisel modelin tasarım motivasyonlarından biri, sıralı hızlı erişim sağlamaktır [17]. Saklanan veriler arasında karmaşık ilişkiler oluşturulmasına ihtiyaç duyulduğunda sorunlar ortaya çıkar. İlişkiler ilişkisel modelle analiz edilebilse de, birçok tablo üzerinde birçok farklı öznitelik üzerinde birçok birleştirme işlemi gerçekleştiren karmaşık sorgular gereklidir. İlişkisel modellerle çalışırken, yabancı anahtar kısıtlamaları ve ilişkileri alırken ek ek yüke neden olurken de dikkate alınmalıdır. Şekil 2.2'de bu türlerin temsili bir gösterimi gösterilmektedir.

Şekil 2.2. Nosql veri modelleri

Bu tip problemlerin üstesinden gelebilmek için NoSQL veri tabanı ortaya çıkmıştır. NoSQL ilişkisel veri tabanında kullanılan tablosal ilişkilerin dışında modellenmiş verilerin depolanması ve alınması için bir mekanizma sağlar. NoSQL veritabanları büyük veri ve gerçek zamanlı web uygulamalarında giderek daha fazla kullanılmaktadır [18]. NoSQL yaklaşımı ilişkisel veri tabanı problemlerine:

- Tasarım basitliği
- Daha yüksek ölçeklenebilirlik
- Dağıtılmış bir bilgi işlem sistemi
- Düşük Maliyet
- Daha esnek bir şema
- Karmaşık olmayan bir ilişki yapısı
- Daha basit ölçeklendirme (yatay olarak)
- Kullanabilirlik üzerine daha hassas kontrol
- Nesne-ilişkisel arası uyumsuzluğu sınırlama

gibi çözümler getirmektedir [19]. NoSQL modeli ilişkisel değildir ve "dağıtılmış" bir veri tabanı sistemi kullanır. Bu ilişkisel olmayan sistem hızlıdır, verileri organize etmek için geçici bir yöntem kullanır ve yüksek hacimli farklı veri türlerini işler. Sadece yapılandırılmış ve yapılandırılmamış verileri işlemekle kalmaz. Aynı zamanda yapılandırılmamış büyük verileri de çok hızlı bir şekilde işleme kabiliyetine sahiptir. Bunların yanısıra NoSQL'in bazı dezantajları da bulunmaktadır. Bunlar:

- RDBMS'deki gibi İlişkisel veri yapısı mevcut değildir.
- RDBMS'de yapılan uygulamaların NoSql sistemlerine taşınması zahmetlidir.

- Transaction kavramı bulunmadığından veri kaybı söz konusu olabilir. Bundan dolayı finansal uygulamalarda tercih edilmezler.
- Veri güvenliği konusunda RDBMS kadar gelişmiş değildir.
- Doküman ve profesyonel destek konusundan eksiklikleri olabilir.

gibi maddeler yer almaktadır. NoSQL sistemleri Döküman tabanlı (genelde binary json formatlı verilerin tutulduğu sistemlerdir. mongoDB bu yapıdadır), Sütun Ailesi (anahtar değer ikilisinden oluşur. her değerde anahtar değer kümeleri bulunabilir), Çizge (Coğrafi bilgi sistemlerine uygun tasarlanmış modellerdir), Anahtar-değer (Anahtar-değer ikililerinin tutulduğu veri modeli. Redis en ünlülerindendir.) gibi farklı türlerde ayrılmaktadır. Döküman tabanlı sistemler yarı yapılandırılmış belge odaklı verileri depolar ve yönetir.

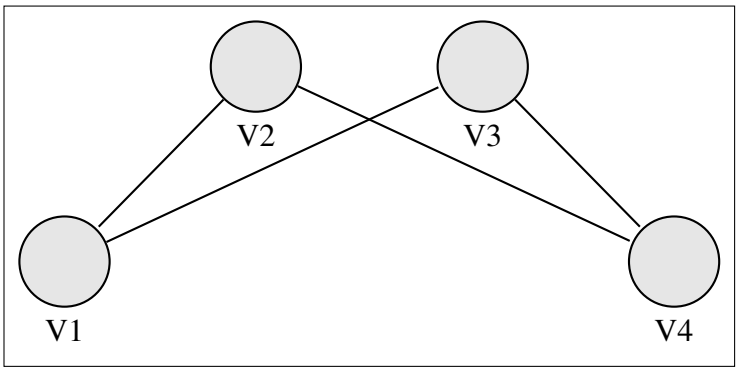
Döküman tabanlı sistemler sorguları hızlı ve kolay hale getiren güçlü bir sorgu motoru ve dizin oluşturma denetimleriyle birlikte gelir. Mongo DB ve Amazon Dynamo DB belirli bir nesne için tüm bilgileri veri tabanında depolar ve depolamadaki her nesne diğerlerinden oldukça farklı olabilir. Bu yaklaşım, nesneleri veri tabanıyla eşlemeyi kolaylaştırır ve web programlama uygulamaları için belge depolamayı çok ilgi görülebilir hale getirir.

Sütun tabalı sistemler ilişkisel veri tabanı sistemlerinden oldukça farklıdır. Veriler satırlar yerine sütunlar halinde saklanır. Bu değişiklik, satırdan sütuna, büyük miktarda verilerin tek bir sütunda depolandığında sütun veritabanlarının performansının artmasını olanak tanır. Cloudera, Cassandra, ve HBase (Hadoop tabanlı) gibi sistemler sütun tabanlı sistemlere örnek olarak gösterilebilir.

Anahtar-değer tabanlı sistemlerde veri tabanlasrı depolamayı map tablosu kullanarak anahtar-değer eşleşmesini yapar. Tüm erişim birincil anahtar kullanılarak yapılır. Anahtar-değer depoları, veri öğeleri arasında karmaşık ilişkiler olduğunda veya verilerin birincil anahtardan başka bir sorgu tarafından sorgulanması gerektiğinde yararlı değildir. Riak, Berkeley DB, and Aerospike gibi sistemler sütun tabanlı sistemlere örnek olarak gösterilebilir.

Çizge tabanlı sistemlerde ise çizge teorine dayanır ve çizge olarak görüntülenebilir verilerle iyi çalışır. İlişkisel veri tabanındaki ilişkilerin aksine satır verisi düğüm, veriler arasındaki ilişkiler ise kenar olarak temsil edilmektedir. Çizge tabanlı sistemlerde ilişkisel veri tabanlarından çok daha esnek, dinamik ve daha düşük maliyetlidir. Bir çizge göz önüne alındığında, genellikle setin her bir elemanı tarafından düzlemdeki bir nokta ve her bir kenar bir çizgi parçası ile temsil edilen diyagramatik olarak ifade etmek

yararlıdır. Şekil 2.3'te V = {V1, V2, V3, V4} düğümlerini E = {V1V2, V1V3, V2V4, V3V4} ise bu düğümler arasındaki ilişkileri temsil etmektedir.



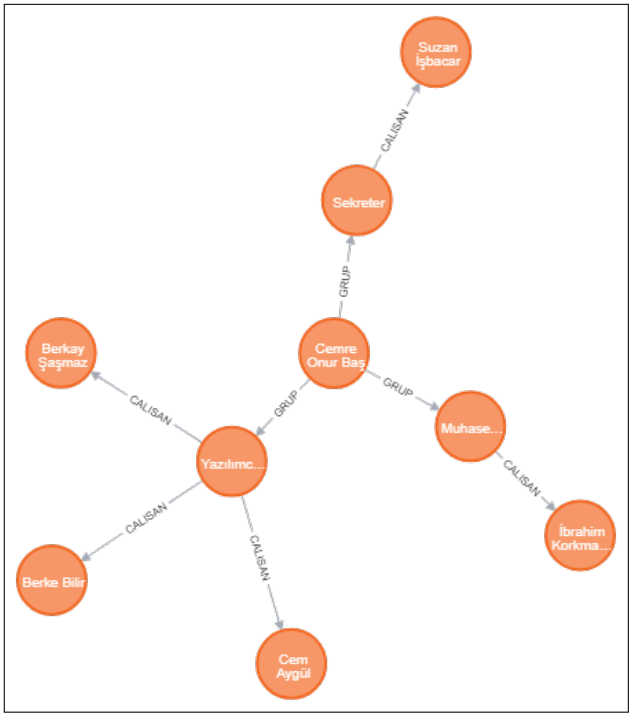
Şekil 2.3. Örnek çizge veri tabanı gösterimi

Çizge veri tabanlı sistemlerde Şekil 2.4'te görüldüğü gibi birçok sistem yer almaktadır.

Şekil 2.4. Çizge veri tabanı çeşitlerinin gösterimi

Bu sistemlerden bir tanesi olan AllegroGraph'ta web uygulamaları geliştirmek amacıyla ortaya çıkmıştır. AllegroGraph, SPARQL adı verilen çizge tabanlı sorgu dilini kullanır. Bu sorgu dilinde özne veya kaynak, yüklem veya özellik ve nesne veya özellik değeri şeklide veriler üçlü olarak RDF'de saklanır. SPARQL'in ana görevi, temel RDF grafiğinde konu, yüklem ve nesneden oluşan modelle eşleşmektir [20]. InfiniteGraph yüksek oranda bağlı büyük veri kümelerinde yararlı ve genellikle gizli ilişkiler bulmak için kullanılmaktadır. InfiniteGraph platformlar arası, ölçeklenebilir, bulut özelliklidir ve çok yüksek iş hacmini kaldıracak şekilde tasarlanmıştır [21]. FlockDB, destek depolama alanı olarak hala MySQL'e güvenen dağıtılmış depolama özellikleri sunar; grafik kenarlarını olabildiğince hızlı sorgulayıp oluşturduğu veya güncellediği düşünülmektedir. Twitter için geliştirilmiş bir sosyal ağ grafik desteği olan veri tabanıdır. HyperGraphDB, yapay zeka, biyoinformatik ve doğal dil işleme gibi son derece karmaşık, büyük ölçekli bilgi sunum uygulamaları için evrensel bir veri modeli olarak

tasarlanmış gömülü, işlemsel bir veri tabanıdır [22]. Trinity ise Microsoft tarafından geliştirilen dağıtılmış bir bellek bulutu üzerinde genel amaçlı bir grafik motorudur. Trinity optimize edilmiş bellek yönetimi ve ağ iletişimi sayesinde hızlı grafik keşfinin yanı sıra verimli paralel hesaplamayı da destekler.Trinity TSL olarak adlandırılan üst düzey bir tanımlama dili sağlar ve bu da genel amaçlı grafik yönetimi ve bilgi işlem için büyük kullanım kolaylığı sağlar [23]. Çizge veri tabanlarının ortaya koyduğu farklı zorluklar arasında, tamamen belleğe sığmayan büyük çizgeleri temsil etmek ve işlemek için etkili bir yol bulmak hala çözülmemiş bir sorundur. Bu problemlere çözüm getirmek amacıyla Dex geliştirilmiştir. Bitmap'lere ve diğer yapılara dayanan yüksek performans sunan bir çizge veri tabanıdır. C++, Java ve Python dilleri ile uyumludur [24]. Bir diğer çizge veri tabanı olan OrientDB ise döküman tabanlı sistem ile çizge tabanlı sistemi birleştiren çoklu model sistemidir [25]. Bu çalışmada sunduğu kolaylıklar ve performanslardan dolayı tercih edilen çizge veri tabanı türü Neo4j'dir. Neo Technology tarafından geliştirilen hızlı, ölçeklenebilir, basit ve açık kaynak kodlu bir çizge veri tabanıdır. CQL ismini verdiği sql benzeri Cypher sorgu dilini kullanmakadır. Java Scala programlama dilleri ile yazılmıştır. Neo4j'de çizge veri tabanındaki bir düğüm ve kenarın istediği kadar niteliğe sahip olması mümkündür. Düğümler ve kenarlar etiketlenebilir. Neo4j'de bağlantı kurmak için Windows CMD üzerinden Neo4j Shelle'e bağlanarak, Rest API üzerinden veya Java API'si üzerinden, beşincisi Gremlin graf dili plug-in'i vasıtasıyla veri tabanına bağlantı kurulabilir. Şekil 2.5'te Neo4j ile oluşturulmuş bir çizge veri tabanı örneği verilmiştir.



Şekil 2.5. Neo4j ile oluşturulmuş çizge veri tabanı örneği

Bu veri tabanının modellemesi yapılırken Cypher sorgu dili kullanılarak kolaylıkla Neo4j'de oluşturulacağı gibi csv gibi dosya formatları yardımıyla da veri tabanı içe aktarılarak veri tabanının çizgesel ortamda oluşturulması sağlanabilir. Neo4j'ye aktarılan satır veriler birer düğüm olarak temsil edilmekte ve bu düğümler arasındaki ilişkiler kenar olarak tutulmaktadır. İlgili veri tabanında düğümler arası ilişkiler etiketlenerek anlaşılırlığı kolay ve basit hale getirmektedir. Lokal olarak kullanılabilen Neo4j database i Cloud gibi ortamlarda da tutularak uzaktan erişilebilirite açısından kolaylık sağlamaktadır.

3. ÖNERİLEN YÖNTEM: TF-IDF TABANLI SEZGİSEL YÖNTEM İLE KAVRAM KEŞFİ

Çalışmanın bu kısmında tf-idf tabanlı tümevaran kavram keşfi için geliştirilen sezgisel yöntem anlatılmaktadır. Şekil 3.1'de gösterilen bu yöntem çizge veri tabanının modellenmesi, kavram tanımlarının bulunması, algoritmanın açıklanması ve önerilen yöntemde kullanılan araçların tanımlanmasından oluşmaktadır.

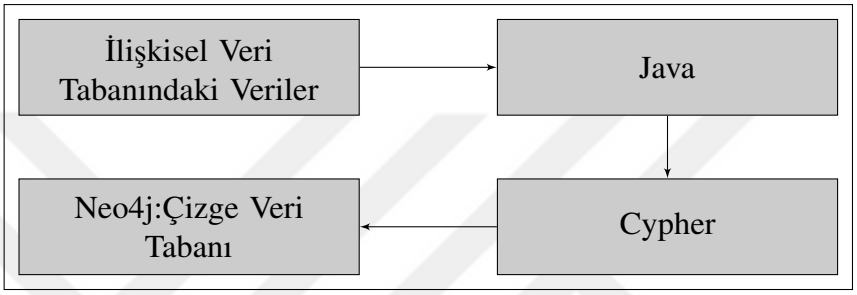
```
Girdi: A: nitelikler, ms: minimum destek değeri, mc: minimum güven değeri,
       N: tespit edilemeyen hedef örnekler
Çıktı: S: kavram tanımlayıcı dizisi
/* pozitifnegatifkontrol : pozitif veya negatif iin kural
   tespitini ayıran kontrol değişkenidir.                    */
Kurallar = [];
foreach a in A do
    if N >= 5 then
        break;
    else
        cizgeyiTekrarOlustur(tabloismi, veritabaniismi, pozitifnegatifkontrol);
        rules = kurallariBul(pozitifnegatifkontrol, 0, a);
        sp = minimumDestegiHesapla(rules);
        cnf = minimumGuveniHesapla(rules);
        if sp >= ms and cnf >= mc then
            S.ekle(c);
            hedefOrneklerinSistemdenCikarilmasi();
        end
    end
    foreach r in S do
        Kurallar.ekle(c);
    end
end
```

Şekil 3.1. Önerilen yöntemin genel kodu

İlk aşamada ilişkisel veri tabanında tutulan verilerin çizge ortamına geçirilmesi gibi işlemler gerçekleştirilmiştir. ikinci aşamada ise hedef ilişkiler, arka plan bilgileri, minimum destek ve güven değerleri hesap edilerek Java programlama dilinde algoritmanın modellenmesi gerçekleştirilmiştir.

3.1. Verilerin Modellenmesi

Veri kümeleri çizge veri tabanına aktarılmadan önce ilişkisel veri tabanında prologta bulunur. Veri kümelerinin çizge veri tabanına aktarılıp kullanılabilmesi gerekmektedir. Bu işlem için Java programlama dilinde Cypher sorguları ile yazılmış Şekil 3.1'da gösterilen cizgeyiTekrarOlustur() isimli fonksiyonda bu işlemler yürütülmektedir. İlgili fonksiyon kendi içerisinde ACCDB'de yer almakta olan ilişkisel veri kümelerini sorgular aracılığıyla çizge veri tabanına çekmekte ve ilişkisel veri tabanında dağıtık bir halde olan sayısal ifadeler kümelendirilerek her bir düğümün nitelikleri arasına dahil edilmektedir. Şekil 3.2'de algoritmanın akış diyagramı verilmiştir.



Şekil 3.2. Verilerin modellenmesi

Şekil 3.3'de görülen sözde kodda çizge veri tabanının ilişkisel veri tabanından aktarılma aşamaları verilmiştir.

```
1 Function cizgeyiTekrarOlustur(tabloismi, veritabaniismi,
    pozitifnegatifkontrol):
      /* blackList, kavram tanımlarının tespiti ardından elenen
         ilaç örneklerinin tutulduğu liste bir sonraki
         iterasyonda ölçümlere dahil edilmemesi için          */
      blackList = [];
      genelNodeларıOlustumaBirbirineBaglama(veritabaniismi,
       pozitifnegatifkontrol);
      sayisalVerileriAyriklastir(blackList);
      cizgeVeriTabanindaIliskilerinBaglanmasi(rules, blackList,
       pozitifnegatifkontrol, tabloismi, veritabaniismi);
```

Şekil 3.3. Çizge veri tabanının oluşturulması ile ilgili sözde kod

Belirtilen fonksiyonda örnerilen sistemde geri dönüşümlü olarak çizge ortamında yer alan tabloların yok edilip ardından yeniden oluşturulması aşaması anlatılmaktadır. Bu aşamada Bölüm 3.1.1'de detaylı olarak anlatılacak olan Mutagenecity ve PTE veri tabanlarının veri kümelerinde sayısal olarak ifade edilen muta_atm, pte_atm, muta_lumo, muta_logp, pte_atm_min_charge, pte_atm_max_charge gibi ilişkisel tablolarında yer almakta olan veriler ayrıklaştırılması işlemleri gerçekleştirilmektedir. Bu sayede çizge

veri tabanındaki kenar (edge) sayısının azalması ve performans olarak sorgu hızlarının artması planlanmıştır. Bu aşamalardan sonra node ismi verilen düğümler oluşturularak ilişkisel anlamda birbirine bağlanması sağlanmıştır. Şekil 3.4'da örnek bir veri kümesinin ayrıştırılması ile ilgili sözde kodu verilmiştir.

```
Function sayisalVerileriAyriklastir(blackList):
    /* _minsupport= 0.1, Önerilen sistemde kullanılan minimum
       destek değeridir                                        */
    getNodeList = tekrarsizTumVerileriGetir(blackList);
    sorguCiktisi = executeQuery('SQL getNodeList te yer alan eşsiz toplam
     verilerinin getirilmesi');
    while sorguCiktisi.next() do
        charge_toplam_verisayisi = (int) (sorgudangelenToplamDeger *
         _minsupport);
    end
    ozellikListesi = [];
    sorguCiktisi = executeQuery('SQL olarak tekrarsız verilerin listelenmesi');
    while sorguCiktisi.next() do
        ozellikListesi.ekle(sorgudangelenDeger);
    end
    for i = 0 to ozellikListesi.uzunluk do
        if mygroup.getir(i).grupsayisi < charge_toplam_verisayisi then
            mygroupDegerleriniGuncelle();
        else
            mygroup.ekle(grupismi, ozellikListesi.getir(i), minimumdeger,
             maximumdeger, 1);
        end
    end
```

Şekil 3.4. İlişkisel veri tabanındaki veri kümelerinin ayrıklaştırılması örneği

Burada veri kümesine göre özel sorgularla veri kümesinin tekrarsız veri kümeleri küçükten büyüğe sıralanarak sistematik olarak gruplandırılması işlemleri gerçekleştirilmiştir. Bu işlem sırasında bir önceki iterasyonda tespit edilmiş kavram tanımlarında elenen örnek ilaçların ölçüme katılmaması için sorguda dahil edilmemiştir. Benzer şekilde ilgili diğer veri kümelerinde sıralı işlemler gerçekleştirilmektedir. Oluşturulan grupların maximum üye sayısı önerilen yöntemde tercih edilen minimum destek değerinin örnek veri kümesinin tekrarsız toplam veri sayısı ile çarpımı olarak referans edilmiştir. Bu tercih ile ilgili yaklaşım ve sonuçlar Bölüm 4'ta detaylı olarak anlatılmaktadır.

3.1.1. İlişkisel veri kümelerinin ayrıklaştırılması

İlişkisel veri tabanında sayısal olarak ifade edilen verilerin bilgi kaybını azaltmak ve performans sağlamak için veri ayrıştırılma yöntemi tercih edilir. Bu çalışmada gruplama

yöntemi tercih edilmiştir. Bu işlem için çok sayıda veri ayrıştırma yöntemi bulunmaktadır. Gruplama yönteminde sayısal veriler daha anlamlı veriler haline getirilir. Bu yöntem sayısal değerlikteki verilerin tekrarsız, sıralı ve eşsiz (unique) değerlerin tespit edilmesi ile başlar. Bu değerler küçükten büyüğe veya tam tersi sıralanma işleminin ardından belirlenen ölçütler doğrultusunda gruplandırılarak anlamlı hale gelmesi amaçlanır. Bu gruplama sırasında en belirgin ölçüt minimum destek değeridir. Bu çalışmada sayısal olarak gruplanan veri kümeleri muta_atm, pte_atm, pte_atm_min_charge, pte_atm_max_charge tablolarında yer alan charge değerleri ve muta_lumo, muta_logp tablolarında yer alan lumo ve logp değerleridir.

İlişkisel veri kümelerinde depolanan lumo, logp ve charge değerleri ondalıklı değerlerdir. Minimum destek değerine göre yapılan bu gruplama işlemi sonucu 230 verisi bulunan Mutagenicity veri kümesinin 11 farklı her bir üyesinde 23 ilaç örneğinin olduğu, 340 verisi bulunan PTE veri kümesinin ise 11 farklı her birinin 34 ilaç örneğinin bulunduğu gruplar haline gelmiştir. Tablo 3.1'te bu gruplama işleminin lumo özelliğine göre sonuçları gösterildiği ve birbirinden farklı 10 grubun oluştuğu gözlemlenmektedir.

Tablo 3.1. Lumo veri kümesinin ayrıklaştırılması sonucu

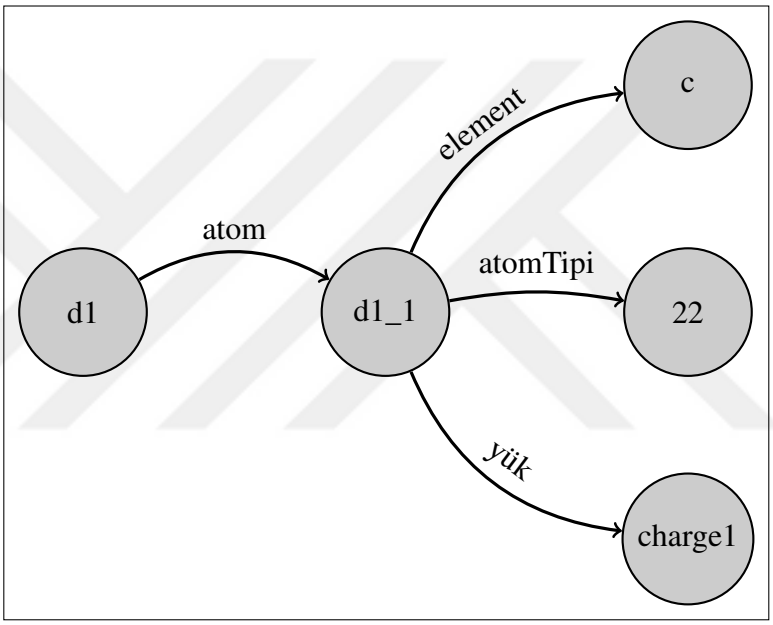
Lumo Adı	**En küçük Değeri**	**En Büyük Değeri**
lumo1	-3,768	-2,31
lumo2	-2,306	-2,155
lumo3	-2,149	-1,918
lumo4	-1,889	-1,748
lumo5	-1,742	-1,61
lumo6	-1,607	-1,499
lumo7	-1,492	-1,387
lumo8	-1,37	-1,256
lumo9	-1,254	-1,102
lumo10	-1,092	-0,798
lumo11	-0,746	-0,529

Bu şekilde çizge veri tabanında lumo için 10 farklı düğüm oluşturulmakta ve hedef ilaç verileri bu düğümler ile etiketlenmektedir. Oluşturulan her düğümün en küçük ve en büyük değerleri düğümün özelliği olarak saklanmaktadır. Böylece bir kayıp olmaksızın veriler birbiri ile eşleştirilebilecektir. Mutagenicity ve PTE veri kümelerinde gruplaştırma sonucu oluşan grupların sayısı aynı olarak gözükmesine rağmen grupların üye sayıları farklılıklar göstermekte olduğu gözlemlenmektedir. Tablo 3.2'de diğer özelliklerin gruplandırılması sonucu ortaya çıkan grupların üye sayıları verilmektedir. Tablodaki ayrıklaştırma işlemle birlikte düğüm sayısında azaltmalara gidilerek basitleştirme adımına gidilir.

Tablo 3.2. Özelliklerin sınıflandırılma sayıları

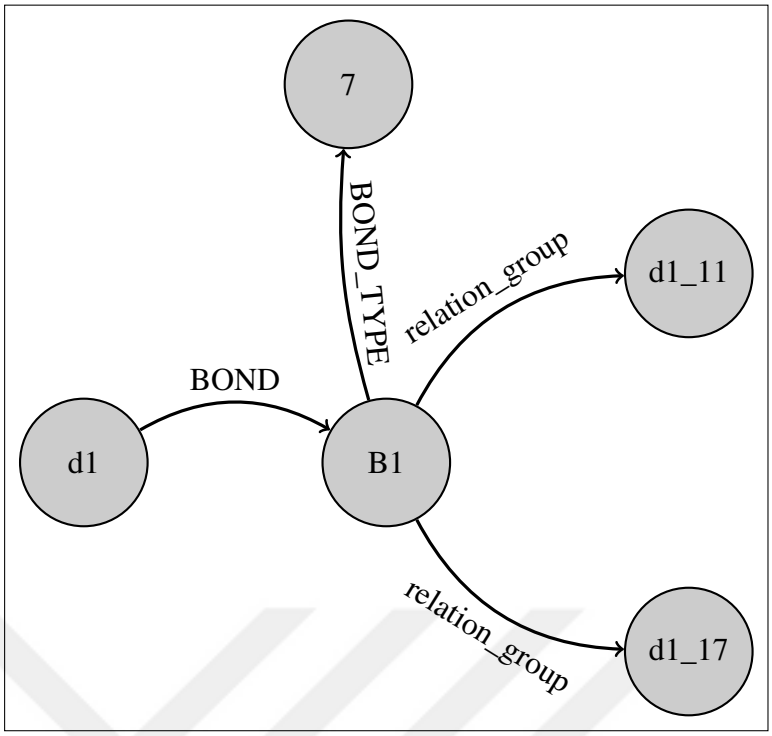
Özellik	**Adet**
Mutagenicity logp	11
Mutagenicity lumo	11
Mutagenicity charge	11
PTE charge	11
PTE mincharge	11
PTE maxcharge	11

Gruplaştırma adımlarının ardından Şekil 3.5'e bakıldığından birinci düğümde d1 ilacı görülmektedir.



Şekil 3.5. muta_atm ilişkisel veri kümesinin çizge örnek gösterimi

İkinci düğümde ise birinci düğüme bağlı bir şekilde olan d1 ilacını oluşturan atomlardan biri olan d1_1 atomu gözükmektedir. Bu atom, d1 ilacı tarafından oluşturulan bir yapı taşı olduğundan tek yönlü bir ilişki kurulmaktadır. d1_1 tarafından oluşturulan diğer 3 düğüme bakıldığında ise bunlar atom özelliğini temcil eden c elementi, atom'un tipini temsil eden 22 özelliği ve ilgili atomun charge grubunu temsil eden charge1 düğümleridir. Şekil 3.6'da görüldüğü gibi d1 isminde bir ortak düğüm kullanılmaktadır.

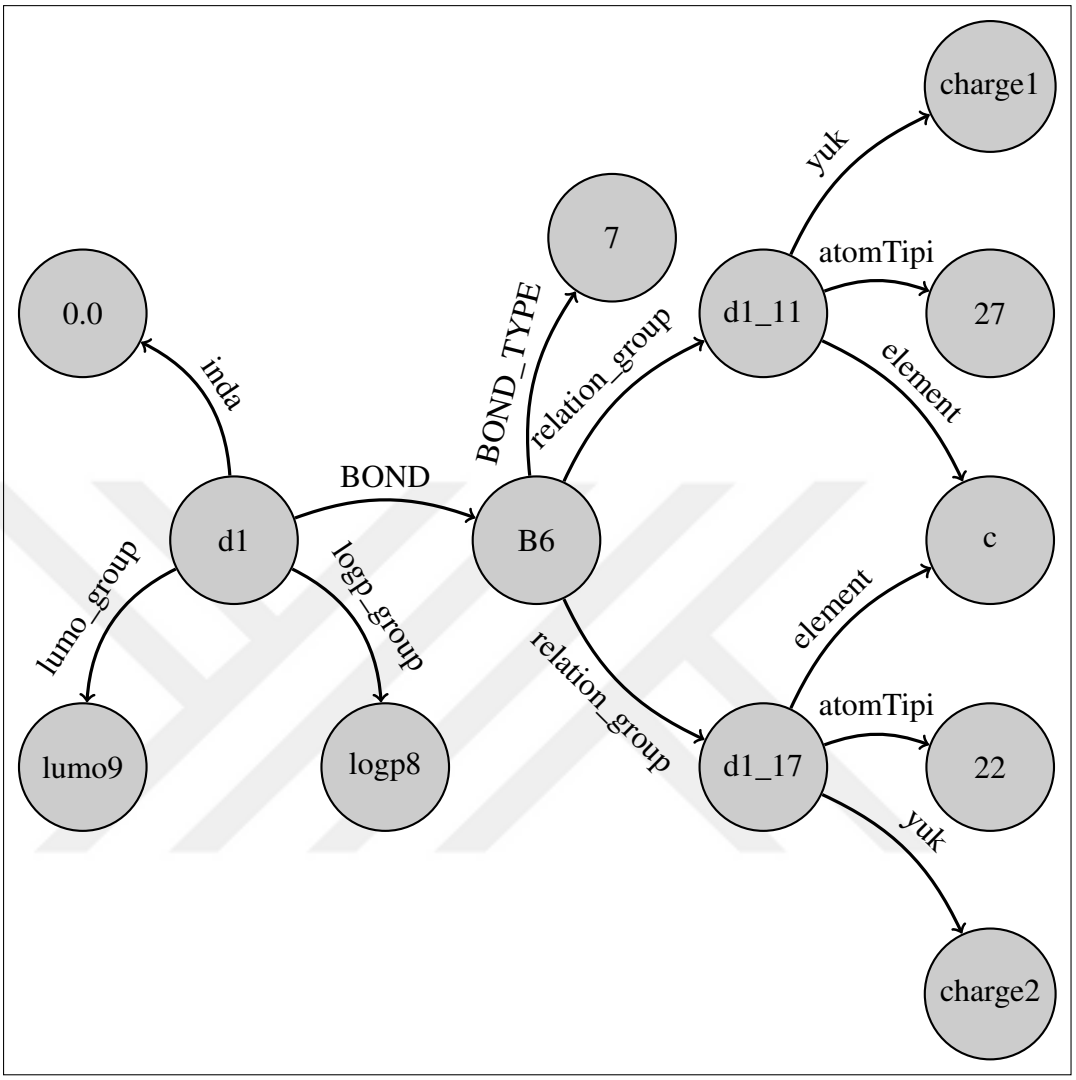


Şekil 3.6. muta_bond ilişkisel veri kümesinin çizge örnek gösterimi

Bu düğüm önerilen sistemlerde d1 ilacının tespiti sırasında d1_11 ve d1_17 atomlarının özelliklerine ulaşmayı ve ölçütlere uygunluğunun denetlenmesini sağlamaktadır. Burada d1 ilacının d1_11 ve d1_17 atomları ile bağlantısı olduğu görülmektedir. Aynı zamanda d1_11 atomu ile d1_17'nun c atom bilgisi ortak olmasına rağmen atom tipi ve charge gibi özelliklerinin birbirinden farklılıklar gösterdiği görülmektedir.

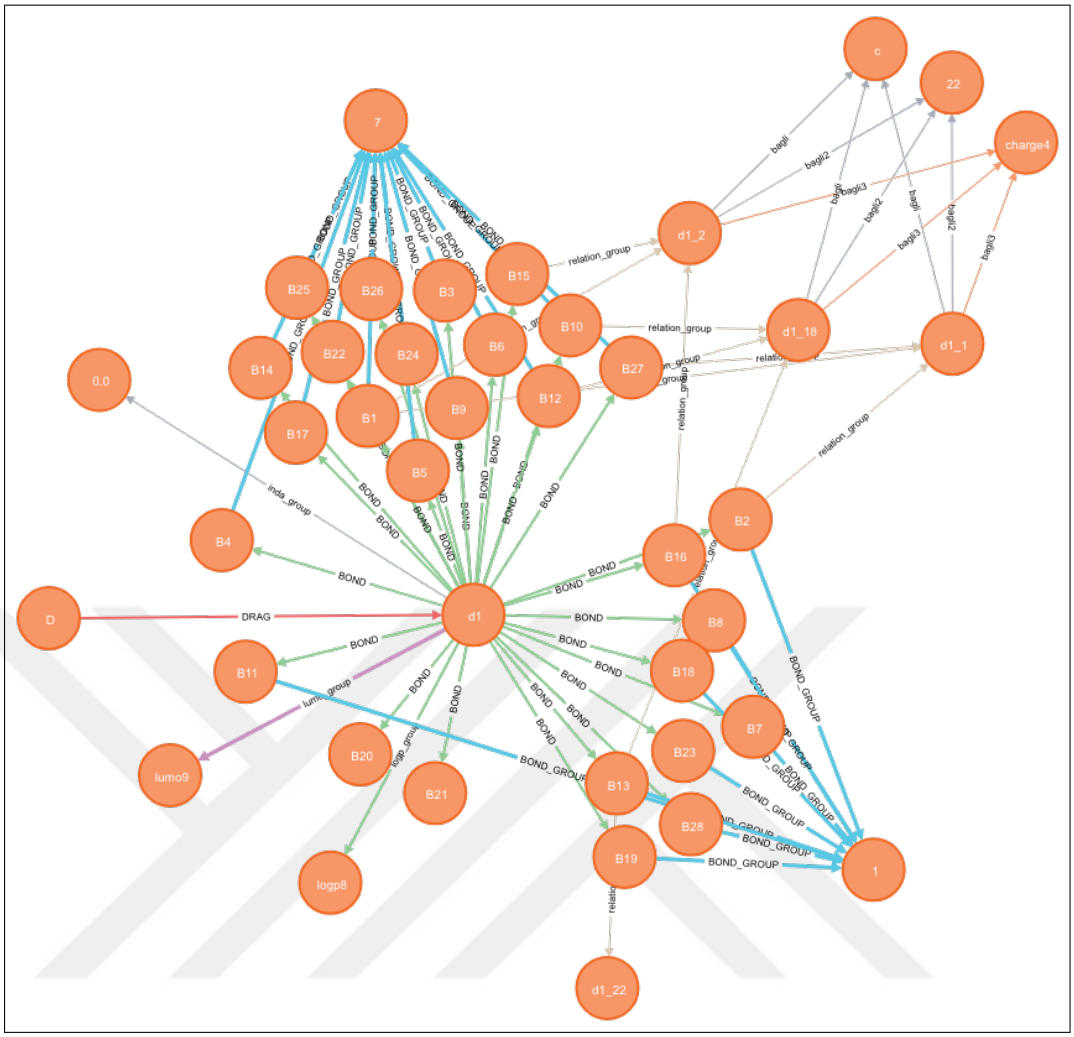
Bu sebeple bu özelliklerin ve bağın anlamlı birer bütün halinde kalabilmesi için bu düğümlerin arasına genel ifade olarak temsil edilen birer bağlayıcı düğüm koyulmasının anlamlı olacağı düşünülmektedir. Eklenen bu bağlayıcı düğümün sistemin ilk iterasyonlarında çalışma performansını etkilemesinde bir etken oluşturacağı düşünülmekte ancak sonraki iterasyonlarda düğüm sayılarındaki gözle görülür azalmanın ardından bu performans farkının azaldığı yapılan deney ve incelemelerde ölçütlere yansımıştır. Performans ölçütü olarak algoritmanın çalışmasını etkileyen birden fazla opsiyonel argüman yer almaktadır. Daha önce yapılan deneylerde, modelin farklı formları, farklı ayrıklaştırılma adımları ve farklı arabirim tercihleri üzerinde durulmuş ve nihai sonuç olarak belirtilen sistem ve modelleme teknikleri bu çalışmada tercih edilmiştir. Bu çalışmada kullanılan diğer modelleme teknikleri ve tercih edilen arabirim ve yaklaşımlar ilerleyen bölümlerde detaylı bir şekilde anlatılmıştır.

Şekil 3.7'de görüldüğü gibi sonuç olarak sayısal içerikli veriler ayrıştırılarak sınıflandırılmıştır.



Şekil 3.7. d1'in çizge veri tabanı model örneği

Daha sonra eşsiz ve tekli veriler halinde kendi içerisinde modellenmiş ve ilişkileri oluşturulmuştur. Mutagenicity'ye ait d1 verisine ait bu modelde sadece d1_11 ve d1_17 bondlarına ait o anki gruplandırılmış verilerine ait bağ ilişkileri gözlemlenmektedir. Her iterasyonda kalan hedef özelliklere göre lumo, logp ve charge gibi özelliklerin frekans değerlerinin minimum ve maksimum aralıklarında değişmeler meydana gelmekte ve toplam grup sayılarında ve bunların hedef özelliklerle olan ilişkilerinin değişmesine yol açmaktadır. Iterasyonla ters orantılı olarak ilerleyen grup ve bağ sayısı, ileri iterasyonlarda daha da azalmasına ve algoritma süresinde azalmalara meydana gelmesine yol açmaktadır. Şekil 3.8'de Mutagenicity veri kümesinin d1 ilacı için nihai halinin örnek gösterimi yer almaktadır.



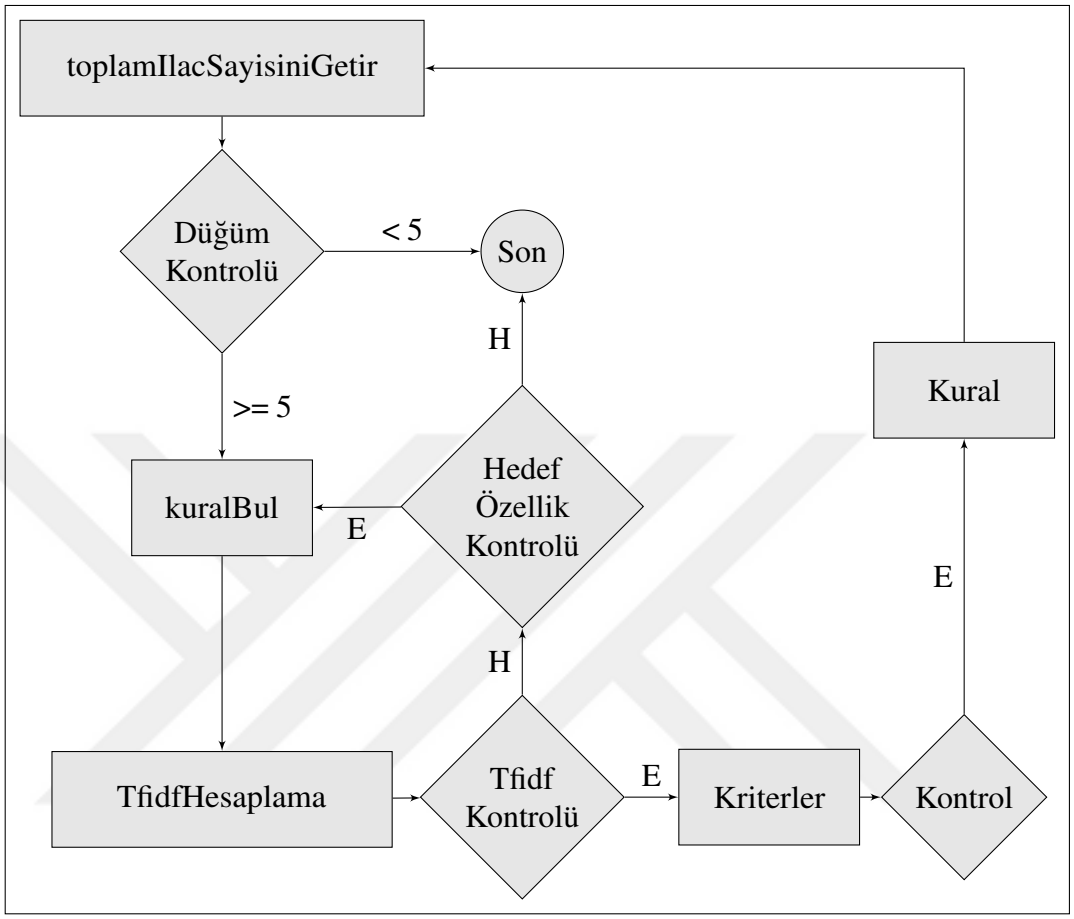
Şekil 3.8. Mutagenicity d1 ilacı için Neo4j'deki örnek çizge gösterimi

Mutagenicity'ye ait ilgili modelde c atomu gibi bazı düğüm bilgilerinin ortak olarak birtakım ilişkilerde görüldüğü gözlemlenmektedir. Şekilde görülmekte olan örnek olarak d1 düğümünün diğer tablo ilişkileri ile oluşturulan yapısal bir bütününün sadece bir parçasıdır. Sistemin geneli düşünüldüğünde fazlaca kenar ve düğümden oluşan bir yapıya sahip olunduğu görülmektedir.

3.1.2. Aday kavram tanımlarının oluşturulması

Adayların bulunması ve bulunan adayların daha önceden belirlenen kriterlere göre değerlendirimesi ve nihai sonuçlar için gerekli tespitlerin yapılması aşamasıdır. Bu aşamada Java'da tf-idf yönteminin faydalarından ve Neo4j kütüphanesinin sunduğu avantajlardan yararlanılmıştır. Aday kavram tanımlarının bulunması ve nihai sonuca ulaşmasında yardımcı olan tespit sistemlerinin akış diyagramı yardımıyla gösterilmektedir. Başlangıç durumu olarak gösterilen toplamIlacSayisiniGetir isimli olay öncesinde Neo4j'nin Cypher sorguları ile bir takım işlemler gerçekleştirilmekte ve sonuçlar sonraki olaylara aktarılıp değerlendirilmektedir. Bu işlemler tarafımız

tarafından belirlenmekte olan bir sonlandırıcı paremetrenin altına ulaşıncaya kadar özyinelemeli olarak devam etmektedir. Şekil 3.9'da gösterilen belirli kontroller ile çalışan bu algoritmada iki adet sonlandırma kriteri bulunmaktadır.



Şekil 3.9. Kavram tanımlarının bulunması ile ilgili akış diyagramı

Bu kriterlerden biri önerilen yöntem olan tf-idf değerinin hesaplanması işleminin ardından aday kavramların tanımlanmasına yetecek hedef özellik kalmadığında sonlandırılması, diğeri ise örnek ilaç sayısının belirlenen ölçütün altında olması durumunda gerçekleşmektedir. Aday kavram tanımların bulunması için tf-idf yöntemi kullanılmıştır. Eklemeli olarak devam eden hedef ilişkiler tf-idf değerleri hesaplandıktan sonra kurala uygunlukları hesaplanır ve verilen ölçütleri uyması koşulunda kural olarak alınmaktadır.

Tf-idf, Term Frequency ve Inverse Document Frequency kelimelerinin bir araya gelmesidir. Term frequency belirli bir döküman içerisinde geçen terimin ağırlıkları hesaplanır. Inverse Document Frequency ise birden fazla dokümanda kelimenin veya verinin geçme sayısını bularak bu ifadelerin terim olup olmadığını olduğu anlamaya çalışır.

$$TF(p_i, G_j) = \frac{G_j \text{ grubundaki } p_i \text{ sayıyısı}}{G_j \text{ grubundaki p sayıyısı}} \tag{3.1}$$

$$IDF(p_i, G_j) = \frac{G_j \text{ grubundaki } p_i \text{ sayıyısı}}{\text{Toplam } p_i \text{ sayıyısı}} \tag{3.2}$$

$$TF - IDF = TF(p_i, G_j) x IDF(p_i, G_j) \tag{3.3}$$

Denklem 3.1, 3.2 ve 3.3'te tf-idf yaklaşımının formülleri yer almaktadır. Bu formulleri incelediğimizde TF formulündeki ilgili pozitif veya negatif verinin ilgili pozitif veya negatif gruptaki frekans değeri ile ilgili pozitif veya negatif verinin tüm gruplardaki toplam değerlerinin çarpımları sonucu elde edilmektedir. Bu çalışmada tf-idf yöntemi PositiveGraph ve NegativeGraph isminde iki ayrı çizge veri tabanı bulunmaktadır.

Şekil 3.9'da gösterilmekte olan TfidfHesaplama bloğunda bu iki ayrı çizge veri tabanı formulasyona yedirilerek hesaplanmaktadır. Bu işlemler sırasında Cypher sorgu dilinin avantajlarından yararlanılmıştır. Her iterasyonda ilişkisel veri tabanının özellik uzayı kadar tekrarlı olarak kurala tf-idf değerinin hedef ilişkinin tf-idf değerinden büyük olma durumunda kurala dahil edilmektedir.

İlgili denklemler incelendiğinde G_j olarak adlandırılan grubun pozitif ve negatif veri kümelerini tespit etmekte, p_i sayısının ise bu veri kümelerinde yer almakta olan hedef ilişkiye ait pozitif veya negatif veri kümesindeki eleman sayısını temsil etmektedir. Fonksiyon içerisinde yer almakta olan toplamPozitifDugumSayisi ve toplamNegatifDugumSayisi fonksiyonları ilgili p_i değerinin bulunmasını sağlamakta, PozitifTfDegeri, PozitifIdfDegeri gibi fonksiyonlar ile de hedef ilişki için tf ve idf değerleri hesaplanmaktadır. Bu işlemlerin ardından hedef değerler TfidfHesapla fonksiyonuna gönderilerek tf-idf değerinin hedef ilişki için bulunması sağlanmaktadır.

Tf-idf değerlerinin hedef ilişki için bulunduktan sonra mevcut iterasyonda maksimum değere sahip tf değerinden yüksek değerlikte olup olmadığı kontrol edilir ve ardından ilgili ölçütlerinin sağlanması şartı ile hedef ilişki mevcut sistemde aday kavram olarak tanımlanır. Bu işlemlerin sonucunda ilgili veri kümesinde aday kavram tarafından tespit edilen düğümler bir sonraki iterasyondaki değerleri etkilememesi için sistemden çıkarılır. Bu işlem Cypher sorgu dilinde gerçekleştirilir. Ardından sonraki iterasyon için kurallariBul fonksiyonu özyinelemeli olarak tekrardan çağrılır. Şekil 3.10'de özyinelemeli olan kurallariBul adı verilen fonksiyonda mevcut veri tabanının pozitif ve negatif üzerine kuralların tespit edilmesini sağlayan sözde kod bilgisi görülmektedir.

Mevcut fonksiyon tf-idf değeri en yüksek oluncaya kadar kural tablosuna her iterasyonda yeni bir özellik yapıp ölçütlere uygunluğu kontrol edilmektedir.

```
1 Function kurallariBul(pozitifnegatifkontrol, currentPropertyIndex, nitelik):
    /* _maxTfIdf = -999, Tf-idf değeri en büyük olanı almak
       için gerekli kontrol mekanizması                          */
    /* totalProperty veri kümelerinin toplam hedef özellik
       sayısı                                                    */
    rules = [];
    for i = currentPropertyIndex to totalProperty do
        rules = kuralBul(rules);
        toplam_p = 0;
        toplam_n = 0;
        tf_p = 0;
        tf_n = 0;
        idf_p = 0;
        idf_n = 0;
        tfidf = 0;
        if pozitifnegatifkontrol then
            toplam_p = toplamPozitifDugumSayisi(nitelik);
            tf_p = PozitifTfDegeri(rules, toplam_p, nitelik);
            idf_p = PozitifIdfDegeri(rules, toplam_p, nitelik);
        else
            toplam_n = toplamNegatifDugumSayisi(nitelik);
            tf_n = NegatifTfDegeri(rules, toplam_n, nitelik);
            idf_n = NegatifIdfDegeri(rules, toplam_n, nitelik);
        end
        tfidf = TfidfHesapla();
        if tfidf < _maxTfIdf then
            _maxTfIdf = tfidf;
        else
            currentPropertyIndex = i;
            kurallariBul(pozitifnegatifkontrol, currentPropertyIndex);
        end
    end
    return rules;
```

Şekil 3.10. Aday Kavramların tf-idf yöntemi ile tespit edilmesi

3.1.3. Tanımların hedef sisteme dahil edilmesi

Bölüm 3.1.2'de önerilen yöntemde tespit edilen aday kavramlarının tf-idf yöntemi ile nihai sonuç olan kavram tanımları haline gelmeden önceki son aşamaları gösterilmiştir. Bu adımda ise kavram tanımların tespit edilmesi ve hedef sistemin x. iterasyonunda yer almakta olan sistemin (x+1). iterasyon için gerçekleştirilen adımlar incelenmiştir. Şekil 3.11'de kavram tanımlarının tespit edilmesi ardından örnek ilaçların sonraki iterasyon

için kaldırılması ardından çizge veri tabanının tekrardan oluşturulması aşamaları gösterilmektedir.

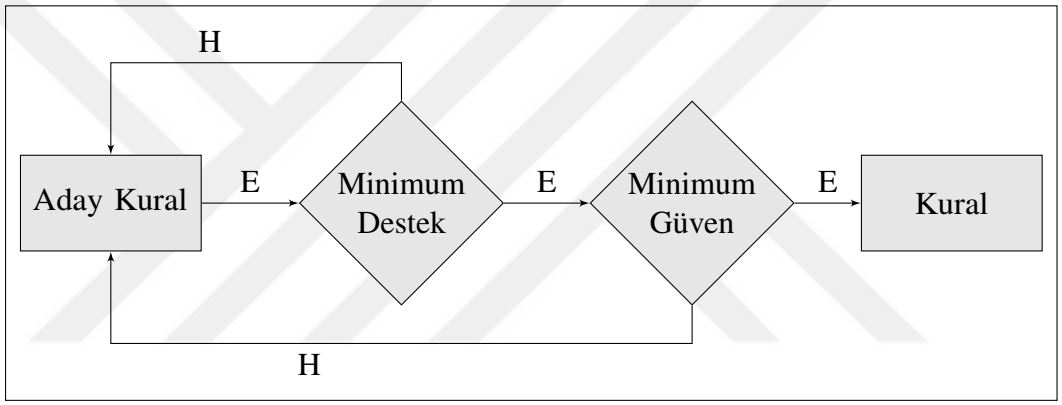
```
Function cizgeVeriTabanindaIliskilerinBaglanmasi(rules, blackList,
 pozitifnegatifkontrol, tabloismi, veritabaniismi):
    /* ms : Minimum destek değeri, mc : Minimum Güven Değeri  */
    pozitifListe = [];
    negatifListe = [];
    minsp = minSupportHesapla(rules, pozitifnegatifkontrol);
    confidence = confHesapla(rules, pozitifnegatifkontrol);
    for i = 0 to pozitifListe do
        if minsp >= 0.1 and confidence >=0.7 then
            blackList.ekle(pozitifListe[i]);
            pozitifIlacicizgetenSil(pozitifListe[i]);
        end
    end
    for j = 0 to negatifListe do
        if minsp >= ms and confidence >=mc then
            blackList.ekle(negatifListe[j]);
            negatifIlaciçizgetenSil(negatifListe[j]);
        end
    end
    sayisalVerileriAyriklastir(blackList);
    cypherSorgulariIleCizgeyiTekrarOlustur(tabloismi, veritabaniismi);
```

Şekil 3.11. Minimum destek ve güven değerlerine göre çizge veri tabanının yeniden oluşturulması ile ilgili sözde kod

İlgili şekilde minimum destek ve minimum güven değerlerinin önerilen sistemde önceden belirlenmiş olan kriterlerine uyması durumunda mutajen olan örnek ilaçlar ile mutajen olmayan örnek ilaçların benzerliklerini ortadan kaldırmak amaçlı sistemden çıkarılması gerekir. Belirlenen ölçütlerin hesaplanmasının ardından mevcut çizge veri tabanından ilgili örnek düğümlerin (ilaçların) silinmesi gerçekleştirilir. Bu aşamadan sonra Bölüm 3.1.1'de anlatılan ayrıklaştırma işleminin tekrardan uygulanır. Bu işlemler Şekil 3.9'da detaylı bir şekilde gösterilen kriterler sağlandığı süreç boyunca özyinelemeli olarak devam eder.

4. DENEYSEL SONUÇLAR

Önerilen yöntemin performansını değerlendirmek için farklı veri kümelerinde çeşitli deneyler gerçekleştirilmiştir. Bu bölümde ilk olarak önerilen yöntemde kullanılan veri modeli anlatılmaktadır. Daha sonra farklı veri kümelerinde gerçekleşen deneylerin sonuçları öğrenme eğilimleri ile beraber tartışılmaktadır. Deney sonuçları elde edilirken minimum destek ve güven değerleri ilgili sonuçların çıkmasında etkili olmuştur. Şekil 4.1'de aday kuralların birer kural olması için gerekli olan aşamaların akış diyagramı verilmiştir.



Şekil 4.1. Aday kuralların değerlendirilmesi aşaması

Minimum değeri, bulunan bir aday kuralın hedef örneklerden (pozitif veya negatif olanlar) kaçını açıklayabildiğini gösteren bir parametredir. Bir aday kuralın açıkladığı örnek sayısının ilgili veri kümesinin toplam hedef örnek sayısına bölünerek hesaplanır.

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

Denklem 4.1'de minimum destek formülü verilmiştir. Güven değeri ise bulunan bir aday kuralı açıklayan hedef örnek sayısının hedef olan ve olmayan toplam örnek sayısına bölümü ile hesaplanan bir parametredir. İlgili parametre tespit edilen aday kuralın hedef örneklere ve sisteme ne kadar güven sağladığını ortaya koymaktadır.

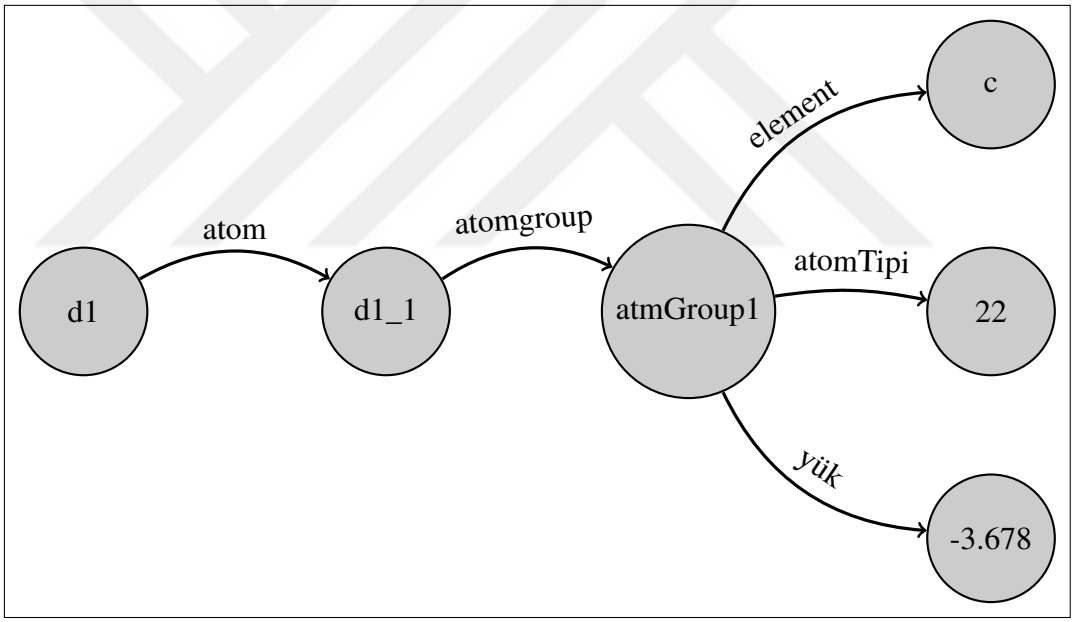
$$\text{güven değeri} = \frac{\text{açıklanan hedef örnek sayısı}}{\text{toplam açıklanan örnek sayısı}} \tag{4.2}$$

Denklem 4.2'de minimum güven değerinin formülü verilmiştir. Bu çalışmada yapılan testler sonucu kavram tanımlarında tespit edilen hedef örneklerin oy çokluğuna (majority voting) göre karmaşıklık matrisleri oluşturulmuştur. Mutagenicity veri kümesine göre

10-katlı, PTE veri kümesine göre ise 1-katlı çapraz doğrulama olarak gerçekleştirilen deneylerde elde edilen karışıklık matrisleri kullanarak oylama işlemi gerçekleştirilmiştir.

4.1. Veri Modelleri

Bu bölümde nihai sonuca ulaşmadan önce yapılan diğer veri modellerinin deneysel çalışmaları anlatılmaktadır. Bölüm 3.1.1'de anlatılan ilişkisel veri kümelerinin ayrıklaştırılması aşamasında daha önce Şekil 3.5'te gösteilen çizge veri modelinden önce farklı bir model ile testler gerçekleştirilmiştir. Gösterilen şekillerde çizge veri modelinde önceki modele ait muta_atm ilişkisel veri kümesinin diyagramı verilmiştir. Sayısal olarak ifade edilen verilerin ayrıklaştırılması aşamasında ortak değerliklere sahip hedef özelliklerin bir arada gruplanması işlemlerinin yürütürüldüğü Şekil 4.2'te örnek ilaçtan oluşan atomun tek bir gruplaşmış olan eşsiz düğüme bağlı olduğu görülmektedir. Muta_atm'de görülen bu veri modeli aynı şekilde pte_atm veri kümesine de uygulanmıştır.



Şekil 4.2. muta_atm ilişkisel veri kümesinin farklı çizge örnek gösterimi

4.2. Kullanılan Ölçütler

Bölüm 3.1.1'da detaylı olarak anlatılan minimum destek değerine göre gerçekleştirilen ayrıklaştırma adımı yerine 0.3 olarak kabul edilen bir gruplaştırma adımı denenmiştir. Bu deney sonucu muta_lumo'nun lumo değerleri için ortaya çıkan ayrıklaştırma grupları ve bu grupların en küçük ve en büyük değerleri elde edilmiştir.

Bu işlemin ardından en büyük ve en küçük değerin arasındaki sayısal farkı fazla olan daha az grup oluştuğu görülmektedir. 0.1 olarak kabul edilen minimum destek değerinin

yerine 0.3 olarak kabul edilmesinin tespit edilen kurallara ve karmaşıklık matrislerinde ortaya çıkan doğruluk (accuracy) değerlerinde de etkisi olmuştur. Bölüm 4.4'te detaylı bir şekilde bu sonuçlar paylaşılmıştır. Tablo 4.1'de muta_lumo tablosunun farklı ölçütler kullanılarak elde edilen ayrıklaştırma grupları ve bu grupların minimum, maksimum değerleri verilmiştir.

Tablo 4.1. Farklı ölçütlerde muta_lumo'nun lumo ayrıklaştırılması sonucu

Lumo Adı	**En Küçük Değeri**	**En Büyük Değeri**
lumo1	-3,768	-1,889
lumo2	-1,88	-1,491
lumo3	-1,488	-1,069
lumo4	-1,056	-0,529

4.3. Veri Kümeleri

Bu çalışmada 2 farklı veri kümesi kullanılmıştır. Tablo 4.2'da kullanılan farklı veri kümelerinin tanımları verilmiştir. Kavram tanımlarını elde etmek için minimum destek ve güven değerlerinden faydanılmıştır.

Tablo 4.2. Farklı veri kümelerinin tanımları

Veri Kümesi	**Tanım**
Mutagenicity	Mutajenik bileşikleri öğrenmeyi amaçlayan veri kümesi
PTE	Kanserojen ilaçları öğrenmeyi amaçlayan veri kümesi

Tablo 4.3. Farklı veri kümelerinin minimum destek ve güven değerleri

Veri Kümesi	**Minimum Destek Değeri**	**Minimum Güven Değeri**
Mutagenicity	0,1	0,7
PTE	0,1	0,65

Bu çalışmada farklı veri kümeleri için kullanılan minimum destek ve güven değerleri Tablo 4.3'de verilmiştir. Gerçekleştirilen testler 2 Ghz İşlemci ve 8Gb belleğe sahip SSD'li bir bilgisayarda yapılmıştır. Yapılan testlerde Mutagenicity veri kümesi için 10 farklı test verisi üzerinde çalışılmıştır. Yerel veri tabanının kullanıldığı bu çalışma da disk arabiminin yazma-okuma değerleri algoritmanın performansını etkilediği gözlenmiştir. Daha önce yapılan çalışmalarda HDD disk arabirimi olarak seçilmiştir. Ancak yapılan sonuçlarda bekleme sürelerinin arttığı gözlemlenmiştir. Bu çalışmanın sonuçları Bölüm 4.4'te detaylı bir şekilde anlatılmaktadır. Tablo 4.4'te karmaşıklık matrisinin gösterimi verilmiştir.

Tablo 4.4. Karmaşıklık matrisi gösterimi

Gerçek	**Tahmin**		
		Pozitif	Negatif
	Pozitif	Olumlu Pozitif (TP)	Olumsuz Negatif (FN)
	Nagatif	Olumsuz Pozitif (FP)	Olumlu Negatif (TN)

Karmaşıklık matrisin oluşturulmasından önce kavram tanımlarında tespit edilen ilaçların oylaması yapılmıştır. Tablo 2.1'de ifade edilen, ikinci parametresi true olan d1 ilacının mutajenik olarak temsil edildiği için Gerçek (Actual) ifadesi Pozitif'tir. Örnek olarak testler sonucu d1 ilacının 4 pozitif, 3 negatif kavram tanımı tarafından tespit edilmesi ile çoğunlukçu oylama sonucu olarak Tahmin değerinin Pozitif'tir. Gerçek değeri Pozitif, Tahmin değeri Pozitif olan d1 ilacı, Olumlu Pozitif (TP) olarak adlandırılmaktadır. Önerilen sistemin etkinliğini azalttığı için Olumsuz Pozitif (FP) ve Olumsuz Negatif (FN) durumları istenmeyen durumlardır.

4.4. Sonuçlar

Bölüm 4.2'de yer alan ölçüt değerine göre yapılan testlerde tespit edilen pozitif kuralların destek ve güven değerleri verilmektedir. Bu işlem Mutagenicity veri kümesinde yer alan muta1_train veri setine göre eğitilen önerilen modelin muta1_test veri kümesinde test edilmektedir. Önerilen yöntem ilişkisel veri kümelerinin çizge veri tabanına dönüştürülmesi ile başlanır. Bu işlem için önce ayrıklaştırma denilen işlem gerçekleştirilir. Daha sonra özellik-değer ilişkisinde olan veri veri seti ve bu veri setleri arasındaki ilişkiler birer kenar ve düğüm olarak çizge veri tabanına aktarılır. Bu işlem için Java'da Neo4'nin imkan verdiği Cypher sorguları ile gerçekleştirilir. Tablo 4.5'te testlerde kullanılan farklı veri kümeleri için çizge veri tabanına dönüştürülme süreleri verilmiştir. Tablo 4.5'de düğüm sayısı arttıkça oluşturulma süresinin uzağıdı gözlenmektedir.

Tablo 4.5. Farklı veri kümelerinin çizge veri tabanı oluşturma süreleri

Veri Kümesi	**Düğüm Sayısı**	**Kenar Sayısı**	**Çizge Veri Tabanı Oluşturma Süresi (sn)**
Mutagenicity	12650	35776	366
PTE	19363	27132	808

Tablo 4.6'te Bölüm 4.2'de yer alan ölçüt değerine göre yapılan testlerde tespit edilen pozitif kuralların destek ve güven değerleri verilmektedir.

Tablo 4.6. Farklı ölçüt değerlerinde mutagenicity veri kümesi pozitif kurallarının destek ve güven değerleri

Kurallar	**Destek Değeri**	**Güven Değeri**
atom : c bond : 1 lumogroup : lumo1[min :-3,768 , max : -1,889]	0,472	0,92
charge : 22 bond : 1 lumogroup : lumo1[min :-3,025 , max : -1,602]	0,393	0,81
charge : 22 bond : 1 lumogroup : lumo2[min :-1,488 , max : -1,266] logpgroup : logp3[min :3,12 , max : 4,83]	0,225	1,0
charge : 22 bond : 1 lumogroup : lumo1[min :-3,025 , max : -1,503] logpgroup : logp3[min :3,06 , max : 5,06]	0,29	1,0
charge : 22 bond : 1 lumogroup : lumo2[min :-1,474 , max : -1,246] logpgroup : logp3[min :2,9 , max : 5,06]	0,272	1,0
chargegroup : charge2[min :-0,13 , max : 0,109] bond : 7 lumogroup : lumo3[min :-1,213 , max : -0,93] logpgroup : logp3[min :2,83 , max : 5,06]	0,25	1,0
chargegroup : charge2[min :-0,13 , max : 0,109] bond : 1 lumogroup : lumo2[min :-1,491 , max : -1,256] logpgroup : logp4[min :5,06 , max : 7,13]	0,166	1,0
chargegroup : charge2[min :-0,13 , max : 0,109] bond : 1 lumogroup : lumo3[min :-1,228 , max : -0,923] logpgroup : logp2[min :1,8 , max : 2,72]	0,2	1,0
chargegroup : charge2[min :-0,13 , max : 0,109] bond : 1 lumogroup : lumo2[min :-1,491 , max : -1,256] logpgroup : logp1[min :-0,47 , max : 1,77]	0,125	1,0
chargegroup : charge2[min :-0,13 , max : 0,109] bond : 1 lumogroup : lumo3[min :-1,228 , max : -0,923] logpgroup : logp1[min :-0,47 , max : 1,77]	0,142	1,0
charge : 22 bond : 1 lumogroup : lumo2[min :-2,055 , max : -1,845]	0,179	0,85

Bu işlem Mutagenicity veri kümesi veri setine göre eğitilen önerilen modelin test edilmesi sonucu tespit edilen kuralları listelenmektedir. Tablo 4.7'da veri modelinin nihai halinde yapılan testlerdeki kuralların destek ve güven değerleri verilmiştir.

Tablo 4.7. Nihai modeldeki mutagenicity veri kümesi pozitif kurallarının destek ve güven değerleri

Kurallar	**Destek Değeri**	**Güven Değeri**
atom : c bond : 1 lumogroup : lumo2[min :-2,306 , max : -2,155]	0,168	1,0
atom : c bond : 1 lumogroup : lumo1[min :-3,768 , max : -2,338]	0,173	1,0
charge : 22 bond : 1 lumogroup : lumo1[min :-3,025 , max : -2,005]	0,151	1,0
charge : 22 bond : 1 lumogroup : lumo4[min :-1,665 , max : -1,59]	0,164	0,85

Tablo 4.8'de Bölüm 4.2'de yer alan ölçüt değerine göre yapılan testlerde tespit edilen kuralların destek ve güven değerleri verilmiştir.

Tablo 4.8. Farklı ölçüt değerlerinde mutagenicity veri kümesinin sonuçları

		Gerçek Değerler	
		Pozitif	**Negatif**
Tahmini Değerler	**Pozitif**	110	28
	Negatif	15	35

Tablo 4.9'de veri modelinin nihai halinde yapılan testlerdeki kuralların destek ve güven değerleri verilmiştir.

Tablo 4.9. Nihai modeldeki mutagenicity veri kümesinin sonuçları

		Gerçek Değerler	
		Pozitif	**Negatif**
Tahmini Değerler	**Pozitif**	48	2
	Negatif	2	15

Yukarıda yer almakta olan tablolar incelendiğinde önceki ölçüt değerlerine ait tespit edilen kuralların destek değerlerinin daha iyi olduğu gözlemlenmesine rağmen karmaşıklık matrisleri karşılaşıldığında önceki ölçüt değerinde yer alan 15 FP ve 28 FN değerlerinin 2'ye indirildiği gözlenmektedir.

$$\text{doğruluk(accuracy)} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.3}$$

Denklem 4.3'te doğruluk ölçütünün formülasyonu verilmektedir. İlgili denklemdeki formulasyona göre yapılan deneyin karmaşıklık matrisleri karşılaştırıldığında, mutagenicity veri tabanı için daha önce yer almakta olan ölçüt değerinde doğruluk değeri 0,77 iken önerilen modelin nihai halinde 0,94 olduğu görülmektektedir. Bu deney sonrası doğruluk değerinde ciddi bir performans artışı olmuştur. PTE ilaçların kanserojen olup olmadığı ile ilgilenen ve ilaçların bilgilerini tutan veri kümesidir [14, 26]. Mutagenecity veri kümesi ailesi için elde edilen 10-katlı çapraz doğrulama sonucu elde edilen karışıklık matrislerinin hedef özelliklerinin tespit edilmesi açısından ne kadar sağlıklı olduğunun tespit edilebilmesi için doğruluk tablosu 10-katlı olarak Mutagenecity veri kümesinde, 1-katlı olarak ta PTE veri kümesinde uygulanmıştır. Daha önce nihai modelin farklı formlarında yapılan deneylerde Mutagenecity ve PTE veri kümesi için bu sonuçlar literatürde yapılan araştırma ve gelişmelerin arkasında kaldığı tespit edilmekteydi. Yapılan ayrıklaştırma ve gruplaştırma adımlarının, deney sonuçlarına bakıldığında bu çalışmada deneylerde kullanılan veri kümeleri için gerekli olduğu gözlemlenmektedir. Tablo 4.10'da PTE veri kümesinin Bölüm 4.2'de yer alan ölçüt değerine göre yapılan testlerde tespit edilen pozitif kuralların destek ve güven değerleri verilmiştir.

Tablo 4.10. Farklı modeldeki PTE veri kümesinin pozitif sonuçları

Kurallar	**Destek Değeri**	**Güven Değeri**
atom : c bond : 1 mincharge : mincharge1[min :-0,812, max : -0,638] maxcharge : maxcharge1[min :0,052 , max : 0,376]	0,14	0,67
atom : c bond : 1 mincharge : mincharge1[min :-0,812, max : -0,626] maxcharge : maxcharge3[min :0,697, max : 1,0]	0,17	1,0
chargegroup : charge2[min :-0,19 , max : 0,13] bond : 1 mincharge : mincharge2[min :-0,63 , max : -0,43] maxcharge : maxcharge2[min :0,423 , max : 0,632]	0,16	1,0
chargegroup : charge2[min :-0,19 , max : 0,13] bond : 1 mincharge : mincharge4[min :-0,173 , max : -0,048]	0,28	1,0
chargegroup : charge2[min :-0,19 , max : 0,13] bond : 1 mincharge : mincharge3[min :-0,512 , max : -0,273] maxcharge : maxcharge1[min :0,052 , max : 0,45]	0,18	1,0
chargegroup : charge2[min :-0,19 , max : 0,13] bond : 1 mincharge : mincharge2[min :-0,641 , max : -0,536] maxcharge : maxcharge2[min :0,45 , max : 0,72]	0,28	1,0
chargegroup : charge2[min :-0,19 , max : 0,13] bond : 1 mincharge : mincharge1[min :-0,812 , max : -0,642] maxcharge : maxcharge1[min :0,052 , max : 0,431]	0,4	1,0

Tablo 4.11'de Bölüm 4.2'de yer alan ölçüt değerine göre PTE veri kümesinde yapılan testlerde tespit edilen kuralların destek ve güven değerleri verilmiştir.

Tablo 4.11. Farklı ölçüt değerlerinde PTE veri kümesinin sonuçları

		Gerçek Değerler	
		Pozitif	**Negatif**
Tahmini Değerler	**Pozitif**	19	17
	Negatif	1	2

Tablo 4.12'de veri modelinin nihai halinde PTE veri kümesinde yapılan testlerdeki kuralların destek ve güven değerleri verilmiştir.

Tablo 4.12. Nihai modeldeki PTE veri kümesinin sonuçları

		Gerçek Değerler	
		Pozitif	**Negatif**
Tahmini Değerler	**Pozitif**	5	3
	Negatif	1	9

Yukarıda yer almakta olan PTE veri kümesine ait önceki ölçüt değeri ile nihai ölçüt değerinin sonuçları arasında farklar olduğu görülmektedir. Karmaşıklık matrisleri

karşılaşıldığında önceki ölçütte yer alan 17 FP değerinin 3'e indirildiği FP değerinin ise aynı kaldığı gözlenmektedir. Verilen karmaşıklık matrisleri incelendiğinde PTE veri tabanı için daha önce yer almakta olan ölçüt değerlerinde doğruluk değeri 0,51 iken 0,77 olduğu görülmektedir. Bu deney sonrası farklı veri kümerinin sonuçlarını incelendiğinde ortaya çıkan doğruluk değerlerinin hepsinde artış söz konusudur.

5. SONUÇLAR VE ÖNERİLER

Kavram keşfi veri kümelerinde yer alan veri ilişkilerinden yola çıkarak mantıksal çözümlerle çıkarımda bulunmayı amaçlayan bir problemler bütünüdür. Literatürde Kavram keşfi problemleri ile ilgili çeşitli yaklaşımlar ve önermeler mevcuttur.

Bu çalışmada ise kavram keşfi probleminde tf-idf dahil edilerek sezgisel bir yöntem önerilmektedir. Önerilen yöntem ilişkisel veri kümesinin modellenerek grafik veri kümesi haline getirilmesi ve kavram tanımlarının bulunması aşamalarından oluşmaktadır.

İlişkisel veri kümelerinin çizge veri kümesine dönüştürülme aşamaları Neo4j'den ve onun sağladığı Cypher dilinden yararlanarak gerçekleştirilmiştir. Kavram tanımlarının ortaya çıkması için ise çalışmada önerilen yöntem olan tf-idf yöntemi, sayısal içerikli hedef özelliklerin ayrıştırılması gibi işlemler Java programlama dili ile gerçekleştirilmiştir. Bu programlama aşamasında da Neo4j Maven'den yararlanılmıştır.

Önerilen yöntem literatürde Terim Sınıflandırma olarak adlandırılan yaklaşımda tercih edilen tf-idf yönteminin grafik ortamlı veri kümelerinin kavram tanımlarının tespit edilmesine yönelik sezgisel bir yöntemde kullanılmasından dolayı diğer yöntemlerden farklıdır.

Deneysel sonuçlar önerilen yöntemin farlı veri kümelerinde ve bu veri kümelerinin farklı ölçüt değerilerinde kavram tanımlarının keşfinde farklı sonuçlara ulaşıldığı görülmektedir. Daha önce yapılan deneyler ile kıyaslandığında doğruluk yüzdesinin diğer yaklaşımlara oranla iyileştirildiği görülmektedir. Önerilen çalışmada kullanılan minimum destek ve güven değerinin Tablo 4.3'te yer alan değerlerine göre yapılması önerilmektedir.

Çalışmanın minimum destek değerine sahip ayrıklaştırma grupları ile yapılan deneylerde doğruluk yüzdesinde artış sağlanmasına rağmen tespit edilen hedef ilaç sayısında azalma olduğu görülmekte ve aşırı uyum gösterme (overfitting) oluşmasına sebep olmaktadır. Bu sebeple bu çalışmanın devamı olarak yapılacak çalışmalarda önerilen ölçüt değerleri uygulandığında tf-idf yönteminin bahsedilen problemlere çözüm araması hedeflenmektedir.

KAYNAKLAR

[1] Džeroski S., Lavrač N., Learning Relations from Noisy Examples: An Empirical Comparison of LINUS and FOIL, L. Birnbaum and G. Collins, 1991, 399–402.

[2] Džeroski S., Lavrač N., *Inductive Logic Programming: Techniques and Applications*, Ellis Horwood, 1994.

[3] Muggleton S., Inverse Entailment and Progol, *New Generation Computing, Special issue on Inductive Logic Programming*, 1995**, 13**(3-4), 245–286.

[4] Conceição J. P. D., The Aleph System Made Easy, 2008.

[5] King R. D., Srinivasan A., Dehaspe L., Warmr: A Data Mining Tool for Chemical Data, 2001**, 15**(2), 173–81.

[6] Mutlu A., Senkul P., Kavurucu Y., Improving the scalability of ILP-based multi-relational concept discovery system through parallelization, *Knowledge-Based Systems*, 2012**, 27**(1), 352–368.

[7] Abay N. C., Mutlu A., Karagoz P., A Graph-Based Concept Discovery Method for n-Ary Relations, vol. 9263, 2015, 391–402, DOI: 10.1007/978-3-319-22729-0_30.

[8] Sakama C., Inoue K., Brave induction: a logical framework for learning from incomplete information, *Machine Learning*, 2009**, 76**(1), 3–35.

[9] Corapi D., Russo A., Lupu E., Inductive Logic Programming as Abductive Search, *Technical Communications of the 26th International Conference on Logic Programming*, Editors: Manuel Hermenegildo, Torsten Schaub, vol. 7, Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2010, 54–63, DOI: 10.4230/LIPIcs.ICLP.2010.54.

[10] Paschke A., Schroeder M., Inductive Logic Programming for Bioinformatics in Prova, 2007.

[11] Guan X., Li Y., Gong H., Improved TF-IDF for We Media Article Keywords Extraction, vol. 1302, IOP Publishing, Ağustos 2019, p. 032003, DOI: 10.1088/1742-6596/1302/3/032003.

[12] Wang Y., Zhang D., Yuan Y., Liu Q., Yang Y., Improvement of TF-IDF Algorithm Based on Knowledge Graph, *2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA), IEEE*, 2018, 19–24.

[13] Guan X., Li Y., Zeng Q., Zhou C., An Automatic Text Summary Extraction Method Based on Improved TextRank and TF-IDF, *IOP Conference Series: Materials Science and Engineering*, vol. 563, 4*, IOP Publishing*, 2019, p. 042015.

[14] Zhu Z., Liang J., Li D., Yu H., Liu G., Hot topic detection based on a refined TF-IDF algorithm, *IEEE Access*, 2019**,** **7**(1), 26996–27007.

[15] Iğde M., Kavurucu Y., Mutlu A., Graph Representation of Relational Database for Concept Discovery, *Procedia-Social and Behavioral Sciences*, 2015**,** **195**(1), 1981–1989.

[16] Matsuda T., Horiuchi T., Motoda H., Washio T., Graph-Based Induction for General Graph Structured Data and Its Applications, *Transactions of the Japanese Society for Artificial Intelligence*, 2001**,** **16**(4), 363–374.

[17] Codd E. F., A relational model of data for large shared data banks, *Communications of the ACM*. 1970**,** **6**(13), 377–387.

[19] Leavitt N., Will NoSQL databases live up to their promise?, *Computer*, 2010**,** **43**(2), 12–14.

[20] Sen S., Agrawal A., Rathi A., Dutta A., Dutta B., An analytical approach for query optimization based on hypergraph, *2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), IEEE*, 2015, 1–6.

[21] Fulton S., *The Other Non-SQL Alternative: Infinite Graph 2.0*, 2011.

[22] Iordanov B., HyperGraphDB: A Generalized Graph Database, *Proceedings of the 2010 International Conference on Web-age Information Management*, WAIM Í10, Jiuzhaigou Valley, China: Springer-Verlag, 2010, 25–36, ISBN: 3-642-16719-5, 978-3-642-16719-5.

[23] Shao B., Wang H., Li Y., Trinity: A Distributed Graph Engine on a Memory Cloud, *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD Í3, New York, New York, USA: ACM, 2013, 505–516, ISBN: 978-1-4503-2037-5, DOI: 10.1145/2463676.2467799.

[24] Martínez-Bazan N., Muntés-Mulero V., Gómez-Villamor S., Nin J., Sánchez-Martinez M. A., Larriba-Pey J. L., Dex: High-performance Exploration on Large Graphs for Information Retrieval, *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM Ó7, Lisbon, Portugal: ACM, 2007, 573–582, ISBN: 978-1-59593-803-9, DOI: 10.1145/1321440.1321521.

[25] Messaoudi C., Fissoune R., Badir H., A performance evaluation of NoSQL databases to manage proteomics data, *International Journal of Data Mining and Bioinformatics*, 2018**,** **21**(1), 70–89.

[26] Srinivasan A., King R. D., Bristol D. W., An Assessment of ILP-Assisted Models for Toxicology and the PTE-3 Experiment, *Proceedings of the 9th International Workshop on Inductive Logic Programming*, ILP Ó9, Berlin, Heidelberg: Springer-Verlag, 1999, 291–302, ISBN: 3-540-66109-3.

[27] Kavurucu Y., Mutlu A., Ensari T., Graph-based concept discovery in multi relational data, *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence), IEEE*, 2016, 274–278.

[28] Achenbach T. M., Plato's problems and Plato's problem, *Language & Communication*, 2003**, 23**(8), 81–91.

[29] de Wolf R. M., Contributions to Inductive Logic Programming, Doktora Tezi, Erasmus Universiteit, 1996.

[30] Zelle J. M., Mooney R. J., Konvisser J. B., "Combining top-down and bottom-up techniques in inductive logic programming", *Machine Learning Proceedings 1994*, Elsevier, 1994, 343–351.

[31] Blum C., Roli A., Sampels M., *Hybrid Metaheuristics*, 6th International Workshop, 2009.

[32] Kavurucu Y., Senkul P., Toroslu I. H., A Comparative Study on ILP-based Concept Discovery Systems, *Expert Syst. Appl.* Eylül 2011**, 38**(9), 11598–11607.

[33] Cropper A., Muggleton S. H., Learning Efficient Logic Programs, *Mach. Learn.* Temmuz 2019**, 108**(7), 1063–1083.

[34] Han J., Haihong E., Le G., Du J., Survey on NoSQL database, *2011 6th international conference on pervasive computing and applications, IEEE*, 2011, 363–366.

[35] Kavurucu Y., Senkul P., Toroslu I. H., ILP-based Concept Discovery in Multi-relational Data Mining, *Expert Syst. Appl.* Kasım 2009**, 36**(9), 11418–11428.

[36] Kavurucu Y., Senkul P., Toroslu I. H., Confidence-based concept discovery in multi-relational data mining, *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2008.

[37] 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 (IISA)*, 2015.

[38] Toprak S. D., Senkul P., Kavurucu Y., Toroslu I. H., A New ILP-based Concept Discovery Method for Business Intelligence, *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, ICDEW Ó7, Washington, DC, USA: IEEE Computer Society, 2007, 962–969, ISBN: 978-1-4244-0831-3, DOI: 10.1109/ICDEW.2007.4401092.

[39] Fonseca N. A., Silva F., Costa V. S., Camacho R., A pipelined data-parallel algorithm for ILP, *IEEE International Conference on Cluster Computing (CLUSTER 2005)*, 2005, DOI: 10.1109/CLUSTR.2005.347059.

[40] Mutlu A., Karagoz P., Kavurucu Y., *A Counting-Based Heuristic for ILP-Based Concept Discovery Systems*, LNCS, 2013.

[41] Robinson I., Webber J., Eifrem E., Graph Databases, 2013.

[42] Djoko S., Cook D. J., Holder L. B., *Analyzing the Benefits of Domain Knowledge in Substructure Discovery*, 1995.

[43] Gonzalez J. A., Holdera L. B., Cook D. J., Graph Based Concept Learning, *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, AAAI Press, 2000, p. 1072, ISBN: 0262511126.

[44] Ade H., Denecker M., AILP Abductive Inductive Logic Programming, *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., 1995, 1201–1207, ISBN: 1-55860-363-8.

[45] Han J., Kamber M., Pei J., Data Mining Trends and Research Frontiers, 585–631, 2012.

[46] Fonseca N. A., Silva F., Camacho R., Strategies to Parallelize ILP Systems, *Inductive Logic Programming*, Editors: Stefan Kramer, Bernhard Pfahringer, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, 136–153, ISBN: 978-3-540-31851-4.

[47] Biggs N., Lloyd E. K., Wilson R. J., Graph theory 1736-1936, *The Mathematical Gazette*, 1987**, 71**(456), 177–177.

[48] Paik J. H., A novel TF-IDF weighting scheme for effective ranking, *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013, 343–352.

[49] Wu H. C., Luk R. W. P., Wong K. F., Kwok K. L., Interpreting tf-idf term weights as making relevance decisions, *ACM Transactions on Information Systems (TOIS)*, 2008**, 26**(3), 1–37.

[50] Srinivasan A., King R. D., Muggleton S. H., Sternberg M. J., Carcinogenesis predictions using ILP, *Inductive Logic Programming*, Editors: Nada Lavrač, Sašo Džeroski, Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, 273–287, ISBN: 978-3-540-69587-5.

[51] Fonseca N. A., Silva F., Camacho R., Strategies to parallelize ILP systems, *International Conference on Inductive Logic Programming, Springer*, 2005, 136–153.

KİŞİSEL YAYIN VE ESERLER

[1] **Baş C. O.**, Tümevaran Kavram Keşif Sistemleri İçin Tf-Idf Tabanlı Sezgisel Bir Yöntem, *Avrasya 5. Uluslararası Uygulamalı Bilimler Kongresi*, Türkiye, Adana, 2019, ISBN: 978-625-7029-47-6.

ÖZGEÇMİŞ

Cemre Onur Baş 1993 yılında Mersin'de doğdu. İlk, orta ve lise öğrenimini Mersin'de tamamlandı. 2011 yılında girdiği Kocaeli Üniversitesi Mühendislik Fakultesi Bilgisayar Mühendisliği Bölümü'nden 2015 yılında mezun oldu. Aynı yıl içerisinde Kocaeli Üniversitesi Müdendislik Fakultesi Bilgisayar Mühendisliği'nde Yüksek Lisans eğitimine başladı. 2015 yılında Gebze Organize Sanayi Bölgesinde Teknopark'ta WhiteCAD Technologies isimli bir yazılım şirketinde Bilgisayar Mühendisi pozisyonunda işe başladı. 2019 senenesinde Sakarya'da askerlik görevini gerçekleştirdi. 4 senelik iş deneyimin ardından blunduğu iş yerinden ayrılarak 2019 yılının Nisan ayında SAMM Technologies şirketinde ARGE Mühendisi olarak işe başladı.