

## Özetleme Mekanizması Kullanılarak Bilgi Çizgesine Yeni Eklentiler

### *Novel Extensions to the Knowledge Graph Using the Hashing Mechanism*

Savaş TAKAN<sup>1</sup> 

<sup>1</sup>Ankara Üniversitesi, Mühendislik Fakültesi, Yapay Zeka ve Veri Mühendisliği Bölümü, Ankara, Türkiye

#### Öz

Bilginin doğasına ilişkin, onu şekillendiren çok çeşitli unsurlar bulunmaktadır. Örneğin güvenilirlik, tutarlılık, değişmezlik ve bağlam gibi mekanizmalar bunların başında gelir. Ancak söz konusu mekanizmaların bilgi çizgesinde temsil edilmesi oldukça yaygın bir problemidir. Çalışmamızda bu problemin çözümüne katkıda bulunmak amacıyla, bilginin karmaşık doğasına ilişkin güven, tutarlılık, değişmezlik ve bağlam gibi temel mekanizmalar, hashing teknolojisi kullanılarak bilgi çizgesine entegre edilmiştir. Çalışmamızda bu eklentiler, bilgi çizgesinden ayrı tutularak, yapıların işlevselliklerinin bozulmaması sağlanmıştır. Geliştirdiğimiz eklentiler sayesinde bir bilgi değiştiğinde onu etkileyen tüm bilgilerin otomatik güncellenmesi, belirsizlik, bilgiler arasında sıralama yapılamaması, bazı bilgilerin değişmez olarak tutulmaması ve bilgiler arasında hızlı bir karşılaştırmanın yapılamaması gibi yaygın bilgi çizgesi problemleri, örnek senaryolar üzerinden test edilerek çözüme kavuşturulmuştur. Çalışmamızın, bilgi çizgesinin iyileştirilmesine yönelik literatüre ve bilgi çizgesini kullanan yapay zeka yazılımlarının geliştirilmesine katkı sunması beklenmektedir.

**Anahtar Kelimeler:** Bilgi çizgesi, hashing, yapay zeka, yazılım mühendisliği, bilgi temsili

#### Abstract

There are various elements related to the nature of knowledge that shape it. For example, mechanisms such as reliability, consistency, invariance and context are among the main ones. However, representing these mechanisms in the knowledge graph is a common problem. In our work, in order to contribute to the solution of this problem, basic mechanisms related to the complex nature of information such as trust, consistency, immutability and context are integrated into the knowledge graph using hashing technology. In our work, these plugins are kept separate from the knowledge graph so that the functionality of the structures is not impaired. Thanks to the plugins we developed, common knowledge graph problems such as automatic updating of all the information that affects a piece of information when it changes, ambiguity, inability to sort information, inability to keep some information immutable, and inability to make a quick comparison between information are tested and solved through example scenarios. Our work is expected to contribute to the literature on knowledge graph improvement and to the development of artificial intelligence software that utilizes knowledge graphs.

**Keywords:** Knowledge graph, hashing, artificial intelligence, software engineering, knowledge representation

## I. GİRİŞ

Çok eski zamanlardan beri, bilgiyi edinmek, bilgiyi saklamak ve yönetmek insanlığın temel hedeflerinden biri olmuştur. Günümüzde gelişen teknolojiler sayesinde, bilgi çok hızlı bir şekilde çoğalmaktadır. Dolayısıyla bilgiyi işlemek, çıkarsamak ve kullanmak zorlaşmaktadır. Bu problemlerin çoğu, bilginin nasıl temsil edildiği ile ilintilidir. Bilgi temsili en yaygın kullanılan yöntemlerden biri, bilgi çizgesidir.

Bilgi çizgeleri, son on yılda yapay zekada önemli bir alan olarak öne çıkmıştır [31]. Bir bilgi çizgesini basitçe, bir çeşit semantiğe sahip, yönlendirilmiş, etiketlenmiş, çoklu ilişkisel bir çizge olarak tanımlamak mümkündür [19]. Anlamsal ağ olarak da bilinen bir bilgi çizgesi, gerçek dünya varlıklarının ve ilişkilerinin çizge ile temsilini ifade eder; nesnelere, olaylara, durumlara veya kavramlara ve bunlar arasındaki ilişkiyi gösterir. Bilgi çizgesi, bilgiyi saklamak ve bilgidен çıkarsamaları yapabilmek için önemli bir araçtır.

Son yıllarda, bilgi çizgeleri çok çeşitli alanlarda farklı amaçlar için yaygın olarak uygulanmakta ve buna paralel şekilde çok çeşitli alanlarla entegrasyonuna dair incelemeler bulunmaktadır. Bu incelemeler arasında dijital haber platformlarında haber üretimi, dağıtımı ve tüketimine yönelik anlamsal bilgi çizgelerinin oluşturulması [28], büyük bilgi çizgelerinin oluşturulmasında, heterojen bilgi kaynaklarının entegrasyonu ve Yapay Zeka (YZ) sistemlerinin daha açıklanabilir ve yorumlanabilir olması [31], imalat ve üretim alanında bilgi çizgesinin uygulanması, bilgi çizgesinde muhakeme teknolojileri [7], Anlamsal Web [32], makine öğrenimi, kural tabanlı öğrenme ve doğal dil işleme araç ve yaklaşımlarını uygulama [40] ve istatistiksel

modellerin büyük bilgi grafikleri üzerinde nasıl eğitilebileceği ve dünya hakkında yeni gerçekleri tahmin etmek için nasıl kullanılabilirliği [25] gibi çok çeşitli konular yer almaktadır.

Bilgi çizgesi, yapay zeka alanında bilgiyi saklamak için çok elverişli bir araç olsa da hangi alanda kullanılırsa kullanılsın bilgi çizgesine yönelik bazı önemli gereksinimler ve eksiklikler bulunmaktadır. Bu eksiklikleri, bir bilgi değiştiğinde onu etkileyen tüm bilgilerin otomatik güncellenmesi sorunu, bilgiler arasında sıralama yapılamaması, bazı bilgilerin değişmez olarak tutulmaması ve bilgiler arasında hızlı bir karşılaştırmanın yapılamaması şeklinde özetlemek mümkündür [19, 38, 27]. Çalışmamızda, söz konusu problemlerin çözümüne katkıda bulunmak amacıyla bilgi çizgesine güvenilirlik, tutarlılık, değişmezlik ve bağlam mekanizmaları entegre edilmiştir. Basit olması, uygulanması için bilgi çizgesine çok fazla müdahale gerektirmemesi ve yaklaşık bir müdahale ile güncelleme, değişmezlik, sıralama ve karşılaştırma gibi temel problemleri çözebilmesi dolayısıyla, bu eklentilerin geliştirilmesinde hashing mekanizmasından yararlanılmıştır.

Bilgi çizgesinin her zaman tutarlı olması önemlidir [23]. Herhangi bir bilginin değişmesi durumunda bu tutarlılık kaybolabilir. Tutarlılığın yeniden sağlanması için, değişen bilgiye bağlı bütün bilgilerin değişmesi gerekmektedir. Çünkü, bir bilgiyi destekleyen unsurlarda oluşabilecek bir değişiklik, zincirleme bir etkiyle, o bilginin desteklediği tüm unsurların gerçekliğini sorgulanır hale getirir. Bilginin değişiminde tutarlılığı sağlamak açısından zaman faktörü önem taşır [37]. Ayrıca bilginin sürekli tutarlı bir şekilde tutulması, karmaşıklığın azaltılmasını sağlar [21]. Klasik bilgi yapısı, değişen bilgiyi neden-sonuç ve çıkarsama ile bulabilir. Fakat bu gibi yöntemlerin, herhangi bir damgalama ve takip özelliği olmadığı için, karmaşıktır ve değişmesi gereken bilginin gözden kaçırılabilmesine yol açabilir. Ayrıca, bu çıkarsamanın global olması durumunda, performans problemi oluşacaktır, lokal olduğunda ise değişimi yakalayamayacağı için, tutarsız bilgi dönecektir. Aynı zamanda, hatalı bilginin çıkarılması, yeni bilginin eklenmesi veya var olan bilginin değiştirilmesinde ciddi performans kayıpları oluşmaktadır.

Bilgi çizgesi ile ilgili diğer bir gereksinim ise bilgiler arası sıralamanın sağlanabilmesidir [30]. Bu gereksinimin sağlanabilmesi için çalışmamızda, bilgi çizgesine güvenilirlik mekanizması entegre edilmiştir. Buna göre bir bilgiyi destekleyen ne kadar fazla güvenilir unsur varsa, o bilgi o kadar güvenilir kabul edilir. Tam tersi durumda ise, söz konusu bilgi şüpheli olarak yorumlanır. Böylece bilgiler arasında sıralama mümkün hale gelir.

Bilgi çizgesinde diğer bir gereklilik iki bilginin karşılaştırılmasıdır. [42, 17]. Bu karşılaştırma işleminin çok hızlı bir şekilde yapılabilmesi önemlidir. Çalışmamızda bilgi çizgesine entegre ettiğimiz ve bağlam adını verdiğimiz hash mekanizması aracılığıyla,  $O(1)$  zamanda iki bilginin özdeşliği tespit edilebilmektedir. Bağlam, bir bilginin bağlamlarına bakılarak o bilgiye ait belirsizliğin giderilmesini sağlar. Örneğin Jaguar hem bir hayvan hem de bir programlama dilini ifade eder. Bir bilgi çizgesinde bunlardan hangisinin ifade edildiğine yönelik belirsizlik, geliştirdiğimiz bağlam eklentisi sayesinde, onu oluşturan bilgilerin karşılaştırılmasıyla giderilebilmektedir.

Bilgi çizgesine dair bir diğer önemli unsur, bilginin değişmez olabilmesidir [6, 4]. Örneğin bir kitabı satın alan ya da okuyan kişiler değişebilirken, kitabın adının ve yazarının değişmez olması gerekir. Yani bilgide, değişebilen unsurlar olduğu gibi, değişmeyen unsurlar da bulunmaktadır.

Çalışma kapsamında geliştirilen eklentilere neden ihtiyaç duyulabileceğini bir örnek ile açıklamak gerekirse, bir sistemde örneğin A, B'yi etkilesin, B, C'yi etkilesin, C de D'yi etkilesin. Bu durumda A'da meydana gelebilecek bir değişiklik anında güncellenmezse, mantıksal çıkarım yapılmak istenen bir durumda sonuçlar yanlış çıkacaktır. Dolayısıyla birbirine etki eden değerlerin anında güncellenmesi kritik önem taşımaktadır. Bu güncellenmenin başarılabilmesi için hash mekanizmasından yararlanılmıştır. Hash mekanizması sayesinde herhangi bir değişik durumda, lokal olarak bu değişikliğin anında fark edilmesi, tespit edilebilmesi ve böylece güncellenmesi sağlanabilmektedir.

Önerdiğimiz eklentide tutarlılık ile ilgili olarak, bilginin değiştiği anda işaretlenmesi ve arka planda istenildiği zaman çalışacak şekilde ilişkili bilgilerin güncellenmesi sağlanmıştır. Bilginin değişmez olmasını sağlayabilmek adına çalışmada, değiştirilemez ve değiştirilebilir verileri saklayacak bir yapı oluşturulmuştur. Çalışmada, bilginin güvenilirliği ile ilgili olarak, sistemde bir bilgi sıralaması geliştirilmiştir. Bağlam ile ilgili olarak, özetleme fonksiyonu sayesinde var olan bağlamlar için eşsiz hash değerleri verilmesi sağlanmıştır. Böylece iki bilginin farklı bağlamları arasında herhangi bir eşleşme sağlandığında, hızlı bir şekilde aynı bağlamlara sahip oldukları anlaşılabilir.

Çalışmada sırasıyla konuyla ilgili araştırmalara yer verilmiş ve ardından önerilen eklentilerin yöntemi açıklanmıştır. Daha sonra, eklentiler ayrıntılı şekilde açıklanmış, sonrasında avantaj ve dezavantajları ortaya konmuştur.

## II. İLGİLİ ÇALIŞMALAR

Bilgi çizgesinin konu edildiği çok geniş bir literatür mevcuttur. Konuyla ilgili öncelikle pek çok inceleme makalesi bulunmaktadır [8, 5, 9, 16, 12].

Literatürde, bilgi çizgesinde tutarlılık odaklı çeşitli araştırmalar bulunmaktadır. Bu çalışmalar içerisinde en fazla etki uyandıran eski bir araştırmada, bir ağdaki kısıtlamalar üzerine iki yeni tamamlayıcı özellik geliştirilmiştir [39]. Yazarlar, bu özelliklerin geri izleme aramasından önce ağın ön işleminden geçirilmesinin faydalı olup olmayacağına karar vermek için kullanılabileceğini öne sürer. Farklı bir araştırmada ise tutarlılık kontrolü için kullanılan araçların, Gen Ontolojisi'ndeki (GO) küçük tutarsızlıkları ve temsilindeki fazlalıkları azaltma fırsatı sağladığı tespit etmiştir [44]. Diğer bir çalışmada ise inanç değişimini ifade etmek için genel, tutarlılık tabanlı bir çerçeve sunulmuştur [13]. Bu çerçeve ile güncelleme ve silme gibi diğer inanç değişikliği işlemleri de ifade edilebilmektedir. Farklı bir araştırmada, olasılıksal bilgi tabanlarındaki tutarsızlık miktarını belirlemeyi amaçlayan bir ölçüm parametresi geliştirilmiştir [22]. Çalışmada tutarsızlığın ölçülmesi, bilgi tabanını tutarlı hale getirmek için gerekli olan ifadelerin kesinlik derecelerinde (yani bu makalede olasılıklar) minimum ayarlamaların dikkate alınmasıyla gerçekleştirilmiştir. Farklı bir araştırmada Mu, bir bilgi tabanındaki her bir formülün o tabanın tutarsızlığındaki sorumluluk derecesi için bir ölçüm önermiştir [23]. Bu ölçüm, bir bilgi tabanının minimum tutarsız alt kümeleri cinsinden verilmektedir.

Konuyla ilgili daha güncel literatüre bakıldığında ise merkezi bir problem olan tutarlılık kontrolünün hesaplama karmaşıklığı ele alan çalışmaların [14] yanı sıra bir bilgi tabanındaki tutarsızlığın doğasını daha iyi anlamak amacıyla bilgi tabanı için tutarsızlığın ölçülmesine yönelik çizge tabanlı bir yaklaşımı içeren [24] çalışmalar ile karşılaşmıştır. Farklı bir güncel çalışmada ise inanç revizyon sürecine dair zorluklardan yola çıkılmaktadır [3]. Buna göre en önemli sorunlardan biri, dikkate alınacak K bilgi tabanının nasıl temsil edileceği ve yeni bilgilerin nasıl ekleneceğidir. Çalışmada,  $(K \ E \ (K \ *))$ 'nin ne zaman tutarsız olduğunu pratik bir şekilde belirlemeye olanak tanıyan algoritmik bir öneri geliştirilmiştir.

Tutarlılığın yanı sıra bağlam, güvenlik ve emniyet açısından kritik alanlardaki birçok modern uygulamanın merkezinde yer almaktadır. Farklı bir araştırmada, öge eşleşmesi, dilbilimsel eşdizim yaklaşımları ve wordnet anlamsal ağ mesafesi dahil olmak üzere dört farklı yöntem kullanılarak insan yorumlarının ifade benzerliği belirlenmiştir [36]. Bağlamı dahil eden yöntemin, vakaların %69'unda insan yanıtlayıcılarla aynı geometrik konfigürasyonu seçerek test edilen dört yöntem arasında en başarılı yöntem olduğu ifade edilmektedir. Bilgi çizgelerinde

bağlam konulu farklı bir araştırmada, mevcut bağlam farkındalıklı sistemlerde bilgi ve bağlamın resmi olarak entegre edilmemiş olmasından yola çıkılarak, bağlamsal muhakemeye ulaşmak için biçimsel bir yaklaşım geliştirilmiştir [1].

Literatürde, çizge teorisinde düğümlerin sıralaması konusunda pek çok çalışma bulunmaktadır [33, 26, 15, 11]. Fakat bilgi çizgesinde sıralamaya yönelik bir araştırmaya bilebildiğimiz kadarıyla rastlanmamıştır. Aynı zamanda veri yapıları özelinde değişmezlik konusu sıkça çalışılmış olsa da [10, 29, 35, 2], bilgi çizgesi özelinde değişmezlik odaklı bir çalışmaya rastlanmamıştır. Buna ek olarak, bilgi çizgesinde güvenilirlik ve sıralama odaklı çeşitli çalışmalar bulunsun da [34, 43, 18] bu çalışmalar, makalemizin konusu ile doğrudan ilintili değildir. Benzer şekilde sınırlı sayıda da olsa bilgi çizgesinde hashing [20, 41] odaklı çalışmalar bulunmaktadır ancak literatürdeki mevcut çalışmaların, makalemizde geliştirdiğimiz eklentiler ile ilişkisi bulunmamaktadır.

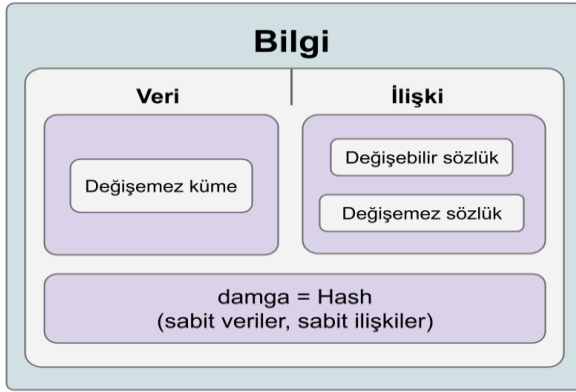
Görülebileceği üzere, literatürde bilgi çizgesine yönelik çalışmalar, çok geniş bir kapsamda ele alınmıştır. Genelde çalışmalarda, bilgi çizgesinin başka alanlara entegre edilmesine odaklanılmıştır. Literatürde bilgi çizgesinde tutarlılık odaklı çalışmalarda ise genellikle kompleks çözümler geliştirilmiştir. Literatürde yer alan bilgi çizgesinde bağlam odaklı sınırlı sayıda çalışmada ise genellikle, bilgi çizgesinin yapısında herhangi bir değişikliğe gidilmeden uygulama tabanlı çözümler geliştirilmiştir. Çalışmamızın, literatürdeki bilgi çizgesinde tutarlılık ve bağlam odaklı mevcut araştırmalardan farklı, söz konusu eklentilerin hash teknolojisi ile sağlamasıdır. Çünkü mevcut literatürde, hash teknolojisi kullanılarak bilgi çizgesine değişmezlik, tutarlılık, güvenilirlik ve bağlam gibi eklentilerin entegre edildiği çalışmalara rastlanmamıştır. Çalışmamızın literatüre temel katkısı, basit bir mekanizma ile (Hashing) bilgi çizgesine dört farklı özelliğin entegre edilebileceğini göstermesidir. Bu yönüyle çalışmamızın, bilginin daha iyi temsil edilmesi ve yaratılan çözümler konusunda literatüre ve bilgi çizgesini kullanan yapay zeka yazılımlarının geliştirilmesine katkı sunması beklenmektedir.

## III. YÖNTEM

Çalışmamızda, var olan bir bilginin nereden geldiği, kimler tarafından desteklendiği, desteklenme oranı, sıralama, değiştirilebilir veya değiştirilemez olması ve otomatik güncelleme işlemlerini gerçekleştirebilmek amacıyla bilgi çizgesine tutarlılık, bağlam, güvenilirlik ve değişmezlik mekanizmaları entegre edilmiştir. Bu eklentiler, literatürden farklı olarak, hashing mekanizması kullanılarak geliştirilmiştir. Bunun nedeni, oldukça basit bir mekanizma olan Hashing teknolojisinin, dört farklı özelliği kolaylıkla sağlayabilme imkanı sunmasıdır. Çalışmamızda, bilgi çizgesine tutarlılık, değişmezlik, güvenilirlik ve

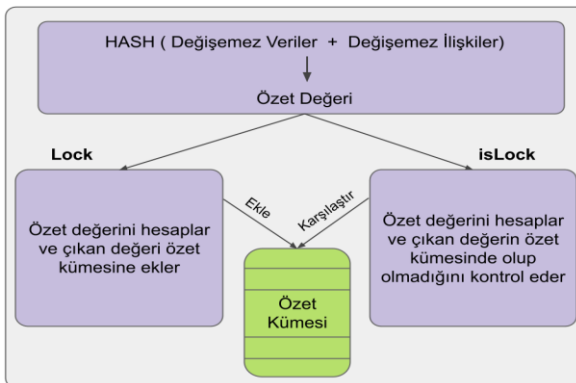
bağlam eklentisi sağlamak için “Knowledge” adını verdiğimiz bir model tasarlanmıştır.

Hashing mekanizması sayesinde, bilgi içerisindeki ilişkilerin ve verilerin değişip değişmediğini kontrol etmek mümkündür. Değişip değişmediği kontrol edilen ilişkilere sabit ilişki, kontrol edilmeyen ilişkilere değişken ilişki denir. Diğer yandan veriler ise her zaman kontrol edildiği için bilgi çizgesinde sabit olarak kabul edilirler. Şekil 1’de, geliştirdiğimiz bilgi çizgesi genel özellikleriyle gösterilmiştir.



Şekil 1. Etiket yapısının genel gösterim

Bilgi çizgesinde değişmezlik kontrolü, hash mekanizması ile sağlanmıştır. Burada hash, değişmez ilişkilerin ve verinin özetini ifade eder. İstenildiği zaman hash hesaplanır ve hash kümesine eklenir. Geliştirdiğimiz bilgi yapısında bu işlem, lock() fonksiyonu ile gerçekleştirilmektedir. Daha sonra, bilgide herhangi bir değişiklik olup olmadığını kontrol etmek için yeni bir hash değeri hesaplanır ve hash kümesinin içerisindeki eski hash değerleri ile karşılaştırılır. Bu andan sonra, bilginin ilişkilerinde veya verilerinde bir değişiklik gerçekleşirse, farklı hash değeri çıkacağından, bilginin değişip değişmediği ve değişmişse bulunduğu pozisyon otomatik olarak tespit edilebilir. Çıkan sonuçlar eşitse yapının değiştirilmediği; sonuçlar eşit değilse yapının değiştirildiği anlamına gelir. Bu işlem, isLock() fonksiyonu ile gerçekleştirilmektedir. Bilginin Lock ve islock durumunun genel yapısı Şekil 2’de gösterilmiştir.



Şekil 2. Lock ve Unlock durumu

Örnek hash bulma formülü aşağıdaki gibidir. Burada i değeri, n tane sabit veriden kaçınıcısının ifade edildiğini gösterir. j değeri ise, m tane sabit ilişkiden kaçınıcısının ifade edildiğini gösterir.

$$hash\left(\sum_{i=0}^n \text{sabit veri}_i + \sum_{j=0}^m \text{sabit ilişki}_j\right)$$

**Formül (1).** Bilgi hash değerinin hesaplanması

Bir bilginin silinmesi, değiştirilmesi, yaratılması ve eklenmesi durumunda, ona bağlı olan bilgiler, tutarlılığı korumak adına Algoritma 1 sayesinde yeniden hesaplanarak güncellenir. Buna paralel olarak, Algoritma 1 sayesinde, bilgi çizgesinde bir değişiklik olup olmadığı tespit edilebilir. Algoritma 1 sayesinde, düşük karmaşıklık ile değişikliklerin bilgi çizgesindeki tüm noktalara taşınması sağlanmaktadır. Ayrıca, bilgi çizgesinde meydana gelen değişikliklerin nerede gerçekleştiği de yine aynı algoritma ile bulunabilmektedir. Algoritmada güncellemeler, bilgi çizgesindeki değişmez ilişkiler üzerinden gerçekleşir. Yani değişken ilişkiler hesaba katılmaz. Bu algoritma, derinlik ilk arama, dinamik programlama ve topolojik sıralama kullanılarak geliştirilmiştir.

```
def update(graph, parent, weight_cost, start_edge):
    stack = [start_edge]
    visited = []
    while stack:
        edge = stack.pop()
        visited.append(edge)
        for neighbor_edge in graph.neighbors(edge):
            if neighbor_edge not in visited:
                visited.append(neighbor_edge)
            if edge not in parent:
                parent[neighbor_edge] = edge
                weight_cost[neighbor_edge] = weight_cost[edge] + graph[neighbor_edge][weight]
                stack.append(neighbor_edge)
def run(graph, start_edge):
    weight_cost = defaultdict(int, {edge: 0 if start_edge else -sys.maxsize for edge in graph.edges})
    weight_cost[start_edge] = 0
    while len(required) > 0:
        update(graph, parent, weight_cost, start_edge)
```

**Algoritma 1.** Değişikliklerin bilgi çizgesinde güncellenmesi için kullanılan pseudo kod

Bilgi çizgesi çoklu geçişli döngüsel bir çizge olduğundan, bütün geçişlerin gezilebilmesi için düğümlerle kenarlar yer değiştirilmiştir. Böylece tüm kenarların gezilebilmesi sağlanmıştır. Bu sayede tüm sistem O(E) karmaşıklığı ile gezilmiştir. Sonuç olarak lineer bir karmaşıklık ile bütün sistem güncellenebilmektedir. Bilgi çizgesinde güncelleme, kullanıcının verdiği derinlik parametresine göre belirlenebilir ve böylece kaç birim derinliğin güncellenebileceği kullanıcı tarafından oluşturulabilir.

#### IV. GELİŞTİRİLEN EKLENTİLER

Bu bölümde, bilgi çizgesi için geliştirdiğimiz her bir eklenti açıklanmış ve örnek senaryolarla test edilerek işlevsellikleri ispatlanmıştır. Böylece, geliştirdiğimiz bilgi çizgesi eklentilerinin çok çeşitli yazılım süreçlerinde kullanılabilir olduğu gösterilmiştir.

#### 4.1. Değişmezlik (Kontrol Edilmeyen İlişkiler)

Kontrol edilmeyen ilişkiyi göstermek amacıyla, aşağıda beş farklı bilgi oluşturulmuştur. Burada A bilgisinin, k ve j vasıtasıyla C ve B ile dışı doğru ilişkisi bulunmaktadır. B, C, j, k bilgilerinin ise bu aşamada herhangi bir ilişkisi yoktur. Bu beş bilgiyi oluşturmak için kullanılan JSON formatı aşağıdaki gibidir:

```
k: {'yap'}, Null, Null, Null}
j: {'sat'}, Null, Null, Null}
B: {'Müşteri 1'}, Null, Null, Null}
C: {'Resim 1'}, Null, Null, Null}
A: {'Ressam'}, {k: C}, {j: B}, Null}
```

Yukarıda gösterilen aşamada, lock fonksiyonu çalıştırılmamıştır. Dolayısıyla herhangi bir değişmezlik mekanizması aktive edilmemiş, yani hash değerleri girilmemiş (Null) olarak gösterilecektir. Yukarıdaki JSON gösteriminde dört tane değer bulunmaktadır. Bunlardan ilki kontrol edilen veriler, ikincisi kontrol edilen ilişkiler, üçüncüsü kontrol edilmeyen ilişkiler, dördüncüsü ise hash değeridir. Aşağıdaki komut ile A bilgisinin lock fonksiyonu çağrılarak, sistem kilitlemiş ve böylece değiştirilemez yapılmıştır. A bilgisinin lock fonksiyonunu çağırma komutu aşağıdaki gibidir:

```
A.lock()
```

Lock fonksiyonu uygulandıktan sonraki yapının JSON format görünümü aşağıdaki gibidir. Burada dikkat edilmesi gereken nokta, hash değerlerinin girilmiş olmasıdır. C ve k bilgileri, A bilgisine bağlı olduğu için, A bilgisi değiştirilmez hale getirildiğinde, bu bilgiler de de değiştirilmez hale gelir. Öte yandan j ve B bilgileri kontrol edilmediğinden (değişken ilişkiye sahip olduğundan) dolayı, bu bilgiler sabitlenmez ve hash değeri Null olarak kalmaya devam eder. Bu durum, Formül 1'de rahatlıkla görülebilir. Ayrıca A bilgisi, Formül 1'deki hesaplama ve Algoritma 1 ile k ve C'de bilgilerin değişip değişmediğini, değişmişse hangi bilginin değiştiğini tespit edebilir.

```
k: {'yap'}, Null, Null, {3e4d}
j: {'sat'}, Null, Null, Null}
B: {'Müşteri 1'}, Null, Null, Null}
C: {'Resim 1'}, Null, Null, {13sa}
A: {'Ressam'}, {k: C}, {j: B}, {12ew}}
```

İstendiğinde, değişken ilişki ile bağlı yeni bir bilgi eklenebilir ve bu durumda lock fonksiyonu çalışsa bile hash değeri değişmez. Bu durum tasarımsal esneklik sağlar. Çünkü bazı ilişkiler sabit iken bazıları değişkendir. Örneğin bir eserin ressamı sabit iken, bu eseri satın alan müşteriler değişkendir. Bu yapıyı bilgi çizgesinde oluşturabilmek zordur. Aşağıda, kontrolünü sağlamadığımız (değişken) ilişkiler değişse bile hash değerlerinin değişmediği JSON formatında gösterilmiştir:

```
k: {'yap'}, Null, Null, {3e4d}
j: {'sat'}, Null, Null, Null}
B: {'Müşteri 2'}, Null, Null, Null}
C: {'Resim 1'}, Null, Null, {13sa}
A: {'Ressam'}, {k: C}, {j: B}, {12ew}}
```

İstendiğinde değişmez ilişkiye yeni bilgi eklenirse, hash değerleri yeniden oluşturulur ve bu yeni değerler eski değerlerle karşılaştırıldığında, yeni oluşturulan hash değerlerinin eski hash değerlerinden farklı olduğu görülecektir. Burada dikkat çekmek istediğimiz nokta, yukarıdaki örneğe yeni bir D bilgisi eklendiğinde A bilgisinin hash değerinin değişeceği. Bu durum, aşağıda JSON formatında gösterilmiştir:

```
k: {'yap'}, Null, Null, {3e4d}
j: {'sat'}, Null, Null, Null}
B: {'Müşteri 2'}, Null, Null, Null}
C: {'Resim 1'}, Null, Null, {13sa}
D: {'Resim 2'}, Null, Null, {12de}
A: {'Ressam'}, {k: C, k: D}, {j: B}, {1saw}}
```

Kontrol edilmeyen ilişkilere yönelik çeşitlilik, bilgi çizgesinde bulunmamaktadır. Bu da tasarımsal manipülasyonları önemli ölçüde etkilemektedir. Örneğin, j ve B değişken ilişkilere sahip olmasaydı, müşteri 1 bilgisi sistemde durmaya devam edecekti. Ayrıca herhangi bir değişiklik durumunda tüm sistemin hash değerlerinin yeniden hesaplanması gerekecekti. Bunun dışında bilgi çizgesinde lock ve isLock fonksiyonları bulunmadığı için, sistem ancak manuel şekilde sabitlenebilmekte ya da en baştan oluşturulmaktadır. Bu da ciddi zaman ve alan kayıplarına yol açabilmektedir.

#### 4.2. Güvenilirlik

Güven eklentisi, bilgi çizgesindeki değişmez ilişkilerin toplamından meydana gelir. Güven mekanizması bu yönüyle, bilgilerin sıralanmasını sağlar. Yani, güven değeri yüksek olan bilgi, daha güvenlidir ve üst sıralarda yer alır.

Geliştirilen güvenilirlik eklentisinde, bir bilginin güvenilirliği, o bilginin sabit ilişkilerinin sayısı ile ilintilidir. Yani bir bilgi, diğer bilgilerle ne kadar fazla değiştirilemez ilişkiye sahipse, o kadar güvenilir kabul edilir. Eğer bir bilginin hiçbir sabit ilişkisi bulunmuyorsa, o bilgiye şüpheli bilgi denir. Aşağıdaki örnekte hiçbir sabit ilişkisi bulunmayan C bilgisinin JSON formatında gösterimine yer verilmiştir. Burada C'nin güvenilirlik değeri 0'dır.

```
C: {'Resim 1'}, Null, Null, {13sa}
```

Aşağıdaki örnekte ise birden fazla sabit ilişkisi bulunan Z bilgisinin JSON formatında gösterimine yer verilmiştir. Burada Z'nin güvenilirlik değeri 2'dir. Güvenilirlik açısından kullanıcının derinlik parametresi girmesi durumunda, hesaplamalar söz

konusu derinliğe kadar yapılır. Örneğin Z'nin derinlik parametresi 1 verilirse, güvenilirlik değeri de 1 çıkacaktır. Bu özellik, zaman ve mekan karmaşıklığını önemli ölçüde azaltmak amacıyla geliştirilmiştir.

k: {{'yap'}, Null, Null, {3e4d}}  
j: {{'kullan'}, Null, Null, {2d5S}}  
l: {{'sat'}, Null, Null, Null}  
M: {{'Müşteri 3'}, Null, Null, Null}  
X: {{'Boya'}, Null, Null, {23ft}}  
Y: {{'Resim'}, {j: X}, Null, {feSa}}  
Z: {{'Ressam'}, {k: Y}, {l: M}, {des3}}

Bilgi çizgesinde, geliştirdiğimiz anlamda pratik ve basit bir güvenilirlik mekanizması bulunmamaktadır. Bu nedenle güvenilirliğin derecelendirilmesi de mümkün değildir. Bu da güven eksenli sıralama mekanizmasını engellemektedir. Diğer yandan, geliştirdiğimiz güvenilirlik mekanizması, bilgi çizgesine pratik ve basit bir şekilde uygulanabilmekte, böylece gerekli hallerde güven odaklı sıralama ihtiyacı rahatlıkla giderilebilmektedir.

#### 4.3. Tutarlılık

Bu bölümde, bilgi çizgesinde tutarlılık eklentisi, örnek bir senaryo vasıtasıyla açıklanmıştır. Burada öncelikle beş adet bilgi yaratılmıştır. Bu bilgiler yaratıldığı anda, sabit ya da değişken hiçbir ilişkileri bulunmamaktadır. Aşağıda, bilgilerin yaratımı JSON formatında gösterilmiştir.

Knowledge1: {'36 yaşındaki bir adam eski nişanlısını bıçaklayarak öldürdü.', Null, Null, Null}  
Knowledge2: {'Eski nişanlısını bıçaklayarak öldüren adam hakkında 23 yıl hapis cezası istendi.', Null, Null, Null}  
Knowledge3 :{'Eski nişanlısını bıçaklayarak öldüren adam ilk duruşmada iyi halde serbest bırakıldı.', Null, Null, Null}  
Knowledge4 :{'Kadın hakları aktivistleri, bu kararı mahkeme önünde protesto etti.', Null, Null, Null}  
Knowledge5: {'Feminizm yaygınlaşıyor.', Null, Null, Null}

Bilgiler yaratıldıktan sonra, aralarında neden sonuç ilişkisi kurulur. Eğer bir bilginin herhangi bir ilişkisi bulunmuyorsa bu bilgi güvenilir değildir. Örneğin aşağıdaki komutlarda, beşinci bilginin nedeni dördüncü, dördüncünün nedeni üçüncü, üçüncünün nedeni ikinci, ikincinin nedeni ise birinci bilgi arasındaki ilişkilerdir. Dördüncü bilginin yok olması, beşinci bilginin güvenilirliğini ortadan kaldıracak ve bu bilgi şüpheli hale getirecektir. Aşağıda, bilgilerin neden sonuç ilişkileri girildikten sonra, bilgiler arasındaki ilişkilerin lock fonksiyonu ile kilitlenmiş hali, JSON formatında gösterilmiştir.

Eylem1: {'neden', Null, Null, {12fK}}  
Knowledge1: {'36 yaşındaki bir adam eski nişanlısını bıçaklayarak öldürdü.', Null, Null, {76Tf}}

Knowledge2: {'Eski nişanlısını bıçaklayarak öldüren adam hakkında 23 yıl hapis cezası istendi.', {Eylem1: Knowledge1}, Null, {23wS}}  
Knowledge3 :{'Eski nişanlısını bıçaklayarak öldüren adam ilk duruşmada iyi halde serbest bırakıldı.', {Eylem1: Knowledge2, Eylem1: Knowledge1}, Null, {23dS}}  
Knowledge4 :{'Kadın hakları aktivistleri, bu kararı mahkeme önünde protesto etti.', {Eylem1: Knowledge3}, Null, {P3se}}  
Knowledge5: {'Feminizm yaygınlaşıyor.', {Eylem1: Knowledge4}, Null, {wq2}}

Bilginin kendisinde ya da onu destekleyen sabit bilgilerin herhangi birinde bir hata ya da değişiklik olduğunda model, değişikliğin kaynağını bularak, o kaynağı bağlamdan çıkarır ve o kaynak ile ilişkili olan tüm bilgileri, derinlik parametresine bağlı olarak günceller. Böylece sistemde tutarlılık sağlanmış olur.

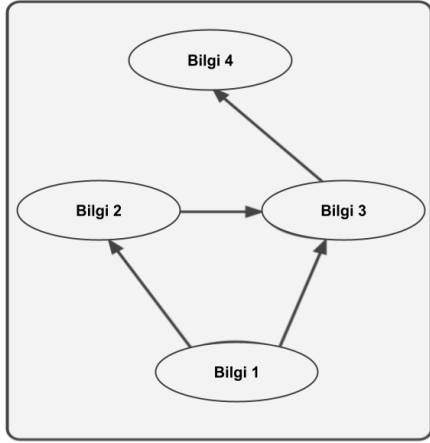
Geliştirdiğimiz eklentideki tutarlılık anlayışı hem bilgi çizgesinin kopyasındaki değişikliğe hem de bilgi çizgesi içerisindeki değişikliklere odaklanır. Bilgi çizgesinin içerisinde herhangi bir bilgide meydana gelebilecek bir değişiklik, bilgi çizgesindeki her bir bilginin güncellenmesine neden olacağı gibi istendiğinde kopyalarındaki değişikliklerin de güncellenmesini mümkün kılar.

#### 4.4. Bağlam

Aşağıda bir bilginin bağlamını açıklayabilmek için dört tane ilişki oluşturulmuştur.

Eylem1: {{'neden'}, Null, Null, Null}  
Knowledge1: {{'veri1'}, Null, Null, Null}  
Knowledge2: {{'veri2'}, {Eylem1:Knowledge1}, Null, Null}  
Knowledge3: {{'veri3'}, {Eylem1:Knowledge1, Eylem1:Knowledge2}, Null, Null}  
Knowledge4: {{'veri4'}, {Eylem1:Knowledge3}, Null, Null}

Aşağıda, yukarıdaki bilgilerin aralarında oluşturdukları bağlam ifade edilmiştir. Burada Knowledge1'in bağlamı yoktur fakat modelde varlığını sürdürür. Bu, Knowledge1'i doğrulayacak herhangi bir değişmez ilişkinin bulunmadığı anlamı taşımaktadır. Değişmez bir ilişkisi bulunmayan ve dolayısıyla normal şartlarda güvenilir olmayan herhangi bir bilginin güvenilir kabul edilip edilmeyeceği, bilgi çizgesini yaratan kişinin inisiyatifine bağlıdır.



Şekil 3. Bilginin bağlam yapısı

Şekil 3'te görüleceği gibi, Knowledge4'ün bağlamı içerisinde Knowledge1, Knowledge2, Knowledge3 bulunmaktadır. Bağlamı hesaplarken hash değeri, çizgeye özgü olacağından, yukarıdaki şekilde Knowledge4'ün bağlamı Knowledge1, Knowledge2, Knowledge3'e ve aralarındaki ilişkiye özgü olacaktır. Şekil 3'te de görülebileceği üzere, hash kümesinde tutulan değerler aynı zamanda bağlamı belirler. Yani bir bilginin birden çok bağlamı olabileceği için istenilen özetler hash kümesine atılarak, o bilginin bağlamlarını oluşturmak mümkündür. Daha sonra bir bilginin, başka bir bilginin bağlamı ile uyumlu olup olmadığı, bu hash değerlerine bakılarak belirlenebilir. Böylece, bilgiye dair pek çok belirsizlik ortadan kaldırılmış olur.

Geliştirdiğimiz eklenti, bilgi çizgesinde bilgi karşılaştırması yapabilmek için bağlam mekanizmasını destekler. Bu sayede, bilgi çizgesinde bilgilerin kolaylıkla karşılaştırılabilmesi mümkün hale gelir. Gerçek yaşamda olduğu gibi bir bilginin değeri, pek çok farklı bağlama göre değişiklik gösterebilir. Geliştirdiğimiz eklentide bu durum rahatlıkla gerçekleştirilebilmektedir.

## V. DEĞERLENDİRME

Bu bölümde, bilgi çizgesi için geliştirdiğimiz eklentilerin avantaj ve dezavantajlar ortaya koyulmuştur. Değişmezlik eklentisinin sağladığı avantaj, bilgi çizgesi içerisindeki bilgilerin değişmiş/değişmemiş olarak damgalanması ve böylece hangi bilgilerin değiştiğinin kolaylıkla tespit edilebilmesidir. Veri yapısı konusunda değişmezlik odaklı çok çeşitli çalışmalar bulunsa da bilgi çizgesi odağında değişmezlik konulu çalışmaya rastlanmamıştır.

Değişmezlik eklentisi, bilgilerin kolaylıkla güncellenmesini sağlayarak bilgi çizgesinin tutarlı kalmasına katkıda bulunur. Bu katkı, çalışmamızda tutarlılık eklentisi olarak tanımlanmıştır. Tutarlılık eklentisi, Algoritma 1 sayesinde tüm değişmiş bilgiler

üzerinde gezinerek, bu bilgilerin hızlı bir şekilde güncellenmesini sağlar. Bu fonksiyon, bir bilgi değiştiğinde otomatik olarak çalıştırılır ve etki ettiği tüm bilgileri, söz konusu değişime bağlı olarak günceller. Bilgi çizgesinde tutarlılığa yönelik çok çeşitli çalışmalar bulunmaktadır. Ancak bu çalışmaların hiçbirinde hash mekanizması kullanılmamıştır. Aynı zamanda, literatürdeki çalışmaların hemen hemen hepsi oldukça karmaşık prosedürler içermektedir.

Güvenirlilik eklentisi sayesinde, bilgiler arasında sıralama gerçekleştirilir. Bilgilerin sıralanması, iki ya da daha fazla bilgi arasındaki önemi ortaya koymaktadır. Bilgi çizgesinde bilgilerin, önem derecesine göre sıralanması, bilgiler arası kıyas yapabilmeye avantajı ve esnekliği sunmaktadır. Literatürde, bilgi çizgesinde güvenirlilik ile ilgili az sayıda araştırmanın hiçbirinde hash mekanizması kullanılmamıştır. Aynı zamanda, mevcut çalışmaların hemen hemen hepsi yine oldukça karmaşık prosedürler içermektedir.

Bağlam eklentisi sayesinde iki bilgi arasında karşılaştırma yapılabilir. Bağlam eklentisi, bilgilerin hash değerlerine bakılarak, bu bilgilerin aynı olup olmadığının anlaşılmasını sağlar. Hash kümesi sayesinde bilgi, birden fazla bağlama sahip olur ve yine hash değeri sayesinde bilginin hangi bağlamda bulunduğu tespit edilir. Bu durum bilgi çizgesine esneklik ve soyutlama avantajı sağlar. Ayrıca Hash algoritması ile karşılaştırma yapıldığından dolayı zaman karmaşıklığı  $O(1)$ 'dir. Literatürde, bilgi çizgesinde bağlam ile ilgili çeşitli araştırmalar bulunsa da bu araştırmaların hiçbirinde hash mekanizması kullanılmamıştır. Aynı zamanda, mevcut çalışmaların hemen hemen hepsi yine oldukça karmaşık prosedürler içermektedir.

Çalışmamızda, bilgi çizgesi için geliştirilen dört eklentinin dezavantajı, hash mekanizmasından kaynaklanan, bilgi çizgesindeki bilgilerin bağlı olduğu tüm bilgilerin hash değerlerinin tutulmasıdır. Burada, bilginin  $N$  tane bağı varsa,  $N \times (256 \text{ Byte})$  uzunluğunda hash değeri saklanır. Bu durum alan karmaşıklığını az da olsa artırır. Diğer bir husus, güncelleme fonksiyonunun  $O(E)$  karmaşıklığındaki çalışma zamanıdır. Güncelleme, çap parametresi ile hem azaltılabilir hem de artırılabilir. Bu durum, karmaşıklık üzerinde önemli bir etkiye sahiptir. Güncelleme karmaşıklığının çap parametresi ile azaltılabilmesi, sığ öncelikli arama (Breadth First Search)'daki derinliği ifade etmektedir. Daha az derinlikte olması, karmaşıklığın yanı sıra bilgi miktarını da azaltacağı için belirli bir denge yakalanması gerekli olmaktadır. Buradaki karmaşıklık, Breadth First Search'deki karmaşıklık ile aynıdır. Burada  $O(E)$  karmaşıklığındaki  $E$ , kenar sayısını ifade etmektedir.

Geliştirdiğimiz eklentilerin, bilgi çizgesine sağladığı katkılar göz önüne alındığında, dezavantaj olarak ifade edilebilecek bu iki hususun göz ardı edilebileceği düşünülmektedir.

## VI. SONUÇ

Çalışmamızda, var olan bir bilginin nereden geldiği, kimler tarafından desteklendiği, desteklenme oranı, sıralama, değiştirilebilir veya değiştirilemez olması ve otomatik güncelleme işlemlerini gerçekleştirebilmek amacıyla bilgi çizgesine tutarlılık, bağlam, güvenilirlik ve değişmezlik mekanizmaları modüler bir biçimde entegre edilmiştir. Bu eklentilerin geliştirilmesinde hashing mekanizması kullanılmıştır. Bunun nedeni, oldukça basit bir mekanizma olan Hashing teknolojisinin, dört farklı özelliği kolaylıkla sağlayabilme imkanı sunmasıdır. Çalışmamızda, bilgi çizgesine tutarlılık, değişmezlik, güvenilirlik ve bağlam eklentisi sağlamak için “Knowledge” adını verdiğimiz bir model tasarlanmıştır.

Önerdiğimiz eklentilerin ilki olan değişmezlik, bir bilgi değiştirilemez yapıldığında, onunla ilintili bütün bilgilerin değiştirilemez olmasını sağlar. Bu sayede bilgi güvenilirliği garanti altına alınır. Herhangi bir değişiklik olduğunda, hash bilgisi değişeceğinden, söz konusu değişikliğin nerede olduğu hemen tespit edilir. Güvenilirlik düzeyi, bilgiyi destekleyen güvenilir bilgilerin miktarı ile ilgilidir. Bu sayede bilgilerin güvenilirlik derecesine göre sıralanması sağlanır. Tutarlılık, bilgi çizgesinde herhangi bir değişiklik olduğunda, bu değişiklikten etkilenen tüm bilgilerin anında güncellenmesini ifade eder. Bağlam ise, bir bilgi ile ilgili tüm bilgilerden ve bunların arasındaki ilişkilerden oluşur. Farklı bağlamlar hesaplanarak, bir bağlam dizisi içine kaydedilir ve bilgilerin bağlam dizisine bakılarak, diğer bağlamlarla uygunlukları kontrol edilebilir.

Geliştirdiğimiz eklentiler sayesinde, bilgi çizgesine ek özellikler kazandırılarak, bilginin daha kapsamlı şekilde yansıtılması sağlanmıştır. Eklentilerin, bilgi çizgesinin iyileştirilmesine yönelik literatüre ve bilgi çizgesini kullanan yapay zeka yazılımlarının geliştirilmesine katkı sunması beklenmektedir. Daha geniş anlamda ise çalışmamızın, bilgi temsiline ihtiyaç duyulan yazılımların geliştirilmesinin yanı sıra, bilginin her alanda kullanılan bir yapı olması dolayısıyla, bilgi ile ilgili çok geniş bir alana katkı sunması beklenmektedir. İleriki çalışmalarda, bir tasarı olarak geliştirilen eklentilerin kapsamlı plot uygulamalarının gerçekleştirilmesi planlanmaktadır.

## KAYNAKLAR

- [1] Alsaig, A., Alagar, V., & Nematollaah, S., Contelog: A declarative language for modeling and reasoning with contextual knowledge. *Knowledge-Based Systems*, 207, 106403, (2020).
- [2] Balakrishnan, D., Ziarek, L., & Kennedy, O., Fluid data structures. *In 17th ACM SIGPLAN International Symposium on Database Programming Languages*, 3–17., (2019).
- [3] Bello López, P., & De Ita Luna, G., An Algorithm to Belief Revision and to Verify Consistency of a Knowledge Base. *IEEE Latin America Transactions*, 19(11), 1867–1874, (2021).
- [4] Besançon, L., Da Silva, C. F., Ghodous, P., & Gelas, J.-P., A Blockchain Ontology for DApps Development. *IEEE Access*, 10, 49905–49933, (2022).
- [5] Cambria, E., Ji, S., Pan, S., & Yu, P. S., Knowledge graph representation and reasoning. *Neurocomputing*, 461, 494–496, (2021).
- [6] Cano-Benito, J., Cimmino, A., & García-Castro, R., Toward the Ontological Modeling of Smart Contracts: A Solidity Use Case. *IEEE Access*, 9, 140156–140172, (2021).
- [7] Chen, X., Jia, S., & Xiang, Y., A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141, 112948, (2020).
- [8] Chen, X., Xie, H., Li, Z., & Cheng, G., Topic analysis and development in knowledge graph research: A bibliometric review on three decades. *Neurocomputing*, 461, 497–515, (2021).
- [9] Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., & Duan, Z., Knowledge Graph Completion: A Review. *IEEE Access*, 8, 192435–192456, (2020).
- [10] Chowdhury, M. J. M., Colman, A., Kabir, M. A., Han, J., & Sarda, P., Blockchain Versus Database: A Critical Analysis. *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 1348–1353, (2018).
- [11] Christoforou, E., Nordio, A., Tarable, A., & Leonardi, E., Ranking a Set of Objects: A Graph Based Least-Square Approach. *IEEE Transactions on Network Science and Engineering*, 8(1), 803–813, (2021).
- [12] Dai, Y., Wang, S., Xiong, N. N., & Guo, W., A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks. *Electronics*, 9(5), 750, (2020).
- [13] Delgrande, J. P., & Schaub, T., A consistency-based approach for belief change. *Artificial Intelligence*, 151(1), 1–41, (2003).
- [14] Grant, J., Molinaro, C., & Parisi, F., Probabilistic spatio-temporal knowledge bases: Capacity



- constraints, count queries, and consistency checking. *International Journal of Approximate Reasoning: Official Publication of the North American Fuzzy Information Processing Society*, 100, 1–28, (2018).
- [15] Huang, Y., Zhang, L., Yang, X., Chen, Z., Liu, J., Li, J., & Hong, W., An Efficient Graph-Based Algorithm for Time-Varying Narrowband Interference Suppression on SAR System. *IEEE Transactions on Geoscience and Remote Sensing*, 59(10), 8418–8432, (2021).
- [16] Issa, S., Adekunle, O., Hamdi, F., Cherfi, S. S.-S., Dumontier, M., & Zaveri, A., Knowledge Graph Completeness: A Systematic Literature Review. *IEEE Access*, 9, 31322–31339, (2021).
- [17] Jabla, R., Khemaja, M., Buendia, F., & Faiz, S., Automatic Rule Generation for Decision-Making in Context-Aware Systems Using Machine Learning. *Computational Intelligence and Neuroscience*, 2022, 5202537, (2022).
- [18] Jiang, S., Liu, Y., Zhang, Y., Luo, P., Cao, K., Xiong, J., Zhao, H., & Wei, J., Reliable Semantic Communication System Enabled by Knowledge Graph. *Entropy*, 24(6), (2022).
- [19] Kejriwal, M., Knowledge Graphs: A Practical Review of the Research Landscape. *Information. An International Interdisciplinary Journal*, 13(4), 161, (2022).
- [20] Khan, N., Ma, Z., Yan, L., & Ullah, A., Hashing-based semantic relevance attributed knowledge graph embedding enhancement for deep probabilistic recommendation. *Applied Intelligence, Dordrecht, Netherlands*, 53(2), 2295–2320, (2023).
- [21] Liberatore, P., & Schaerf, M., Belief Revision and Update: Complexity of Model Checking. *Journal of Computer and System Sciences*, 62(1), 43–72, (2001).
- [22] Muiño, D. P., Measuring and repairing inconsistency in probabilistic knowledge bases. *International Journal of Approximate Reasoning: Official Publication of the North American Fuzzy Information Processing Society*, 52(6), 828–840, (2011).
- [23] Mu, K., Responsibility for inconsistency. *International Journal of Approximate Reasoning: Official Publication of the North American Fuzzy Information Processing Society*, 61, 43–60, (2015).
- [24] Mu, K., Measuring inconsistency with constraints for propositional knowledge bases. *Artificial Intelligence*, 259, 52–90, (2018).
- [25] Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E., A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE*, 104(1), 11–33, (2016).
- [26] Nirmala, P., & Nadarajan, R., Cumulative centrality index: Centrality measures based ranking technique for molecular chemical structural graphs. *Journal of Molecular Structure*, 1247, 131354, (2022).
- [27] Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., & Taylor, J., Industry-scale Knowledge Graphs: Lessons and Challenges. *ACM Queue: Tomorrow's Computing Today*, 17(2), 48–75, (2019).
- [28] Opdahl, A. L., Al-Moslmi, T., Dang-Nguyen, D.-T., Gallofré Ocaña, M., Tessem, B., & Veres, C., Semantic Knowledge Graphs for the News: A Review. *ACM Comput. Surv.*, 55(7), 1–38, (2022).
- [29] Ozdayi, M. S., Kantarcioglu, M., & Malin, B., Leveraging blockchain for immutable logging and querying across multiple sites. *BMC Medical Genomics*, 13(Suppl 7), 82, (2020).
- [30] Porebski, S., Evaluation of fuzzy membership functions for linguistic rule-based classifier focused on explainability, interpretability and reliability. *Expert Systems with Applications*, 199, 117116, (2022).
- [31] Rajabi, E., & Etminani, K., Knowledge-graph-based explainable AI: A systematic review. *Journal of Information Science and Engineering*, 01655515221112844, (2022).
- [32] Ryen, V., Soyulu, A., & Roman, D., Building Semantic Knowledge Graphs from (Semi-)Structured Data: A Review. *Future Internet*, 14(5), 129, (2022).
- [33] Sciriha, I., & da Fonseca, C. M., On the rank spread of graphs. *Linear and Multilinear Algebra*, 60(1), 73–92, (2012).
- [34] Seo, S., Oh, B., & Lee, K.-H., Reliable Knowledge Graph Path Representation Learning. *IEEE Access*, 8, 32816–32825, (2020).
- [35] Stančić, H., & Bralić, V., Digital Archives Relying on Blockchain: Overcoming the Limitations of Data Immutability. *Computers*, 10(8), 91, (2021).
- [36] Stock, K., & Yousaf, J., Context-aware automated interpretation of elaborate natural language descriptions of location through learning from empirical data. *International Journal of Geographical Information Science: IJGIS*, 32(6), 1087–1116, (2018).
- [37] Terenziani, P., Integrated temporal reasoning with periodic events. *Computational Intelligence. An International Journal*, 16(2), 210–256, (2000).
- [38] Troussas, C., & Krouska, A., Path-Based Recommender System for Learning Activities Using Knowledge Graphs. *Information. An International Interdisciplinary Journal*, 14(1), 9, (2022).
- [39] Van Beek, P., & Dechter, R., Constraint tightness and looseness versus local and global consistency. *Journal of the ACM*, 44(4), 549–566, (1997).
- [40] Verma, S., Bhatia, R., Harit, S., & Batish, S., Scholarly knowledge graphs through structuring scholarly communication: A review. *Complex & Intelligent Systems*, 1–37, (2022).

- 
- [41] Wang, H., Shang, Y., & Qiao, X., The Integrated Organization of Data and Knowledge Based on Distributed Hash. *2020 IEEE International Conference on Knowledge Graph (ICKG)*, 243–250, (2020).
- [42] Wu, W., Zhu, Z., Zhang, G., Kang, S., & Liu, P., A reasoning enhance network for multi-relation question answering. *Applied Intelligence*, 51(7), 4515–4524, (2021).
- [43] Yang, M., Chen, K., Sun, S., Han, Z., Kong, L., & Meng, Q., A Pattern Driven Graph Ranking Approach to Attribute Extraction for Knowledge Graph. *IEEE Transactions on Industrial Informatics*, 18(2), 1250–1259, (2022).
- [44] Yeh, I., Karp, P. D., Noy, N. F., & Altman, R. B., Knowledge acquisition, consistency checking and concurrency control for Gene Ontology (GO). *Bioinformatics*, 19(2), 241–248, (2003).