ADAPTİF ÖĞRENME TABANLI GÜVEN DEĞERİ KULLANILARAK BLOKZİNCİRİN ÖLÇEKLENMESİ

SCALING BLOCKCHAIN BY USING ADAPTIVE LEARNING BASED REPUTATION VALUE

AHMET BUĞDAY

DR. ÖĞR. ÜYESİ ADNAN ÖZSOY

Tez Danışmanı

PROF. DR. HAYRİ SEVER

Eş Danışman

Hacettepe Üniversitesi

Lisansüstü Egitim-Öğretim ve Sınav Yönetmeliğinin

Bilgisayar Mühendisliği Anabilim Dalı için Öngördüğü

DOKTORA TEZİ olarak hazırlanmıştır.

2020

Ailem için. . .

ÖZET

ADAPTİF ÖĞRENME TABANLI GÜVEN DEĞERİ KULLANILARAK BLOKZİNCİRİN ÖLÇEKLENMESİ

Ahmet BUĞDAY

Doktora,Bilgisayar Mühendisliği

Danışman: Dr. Öğr. Üyesi Adnan ÖZSOY

Eş Danışman: Prof. Dr. Hayri SEVER

Haziran 2020, 84 sayfa

Blokzincir teknolojisinin önündeki en büyük zorluklardan biri ölçeklenebilirlik problemidir. Uzlaşma algoritması seçimi, ölçeklenebilirlik probleminin pratik çözümü için kritik öneme sahiptir. Ölçeklenebilirliği artırmak için, Bizans Hata Toleransı (Byzantine Fault Tolerance - BFT) tabanlı yöntemler en yaygın şekilde uygulanmıştır. Bu çalışma, açık blokzincir ağında BFT tabanlı yöntemlerin kullanılmasına izin veren uzlaşma komitesini oluşturmak için İş Kanıtı (Proof of Work-PoW) yerine yeni bir model önermektedir. Önerilen modelde, çevrimiçi karar tabanlı öğrenme algoritması olan uyarlamalı çit yöntemi(adaptive hedge method) [1] kullanılarak uzlaşma komitesine katılmak isteyen düğümler için güven değeri hesaplanır ve uzlaşma komitesindeki düğümlerin zararlı olma ihtimalini azaltmak için uzlaşma komitesine yüksek güven değeri olan düğümler seçilir. Bu çalışmada uzlaşma komitesi oluşturulmasına odaklanıldığından, önerilen modeli daha etkin bir şekilde test etmek için blokzincir ağının simülasyonu kullanılmıştır. Test sonuçları, önerilen modelin (uzlaşma komitesinin oluşturulmasında makine öğrenmesinden faydalanan yeni bir yaklaşım) uzlaşma komitesine güven değeri yüksek olan düğümleri başarıyla seçtiğini göstermektedir. Ayrıca blokzincir

çalışmaları, son zamanlarda ölçeklenebilirlik problemini ele almak için blokzincirin parçalara ayrılmasına(sharding) odaklanmıştır. Parçalara ayırma yönteminde, blokzincir ağı küçük gruplara ayrılır. Daha kapsamlı bir ağ yerine, daha az düğüme sahip ağlar oluşturulur. Bu nedenle, ağdaki her düğümün güvenilir olması daha önemli hale gelir. Bu süreç için öğrenme temelli uyarlamalı yöntemlerin kullanılması, blokzincir parçalarının güvenli ve güvenilir kullanımına katkıda bulunacaktır. Her bir parçanın tüm blokzinciri bozma ve etkileme olasılığı azaltılacaktır. Modelimizi blokzincir ağını parçalara ayırmak için de uyguladık. Test sonuçları, güven değeri kullanımının parçaların güvenilirliğini artırdığını göstermiştir.

Anahtar Kelimeler: Uzlaşma komitesi, blokzincir, BFT, PBFT, çevrimiçi öğrenme, blokzincir parçalara ayırma

ABSTRACT

SCALING BLOCKCHAIN BY USING ADAPTIVE LEARNING BASED REPUTATION VALUE

Ahmet BUĞDAY
Doctor of Philosophy,Computer Engineering Department
Supervisor: Asst. Prof. Dr. Adnan ÖZSOY
Co-Supervisor: Prof. Dr. Hayri SEVER
June 2020, 84 pages

Scalability has become a challenging problem for blockchain technology. Consensus algorithm selection is critical for the practical solution of the scalability problem. Byzantine Fault Tolerance (BFT) based methods have been applied most commonly to increase scalability. We propose a new model for creating consensus committee which is not using Proof of Work (PoW) so that BFT-based methods could be used in public blockchain networks. In our model, we use an online, decision-theoretic, unsupervised learning algorithm which is called the adaptive hedge method [1]. For nodes wishing to join the consensus committee, the reputation value is calculated and nodes with a high reputation value are selected to the consensus committee to reduce the probability that the nodes in the consensus committee are harmful. Since this study focused on establishing a consensus committee, simulation of the blockchain network was used to test the proposed model more effectively. The test results show that the proposed model (a new approach that uses machine learning in the creation of a consensus committee) has successfully selected nodes with high reputation in the consensus committee. In addition, blockchain studies have recently focused on sharding the blockchain for solving the scalability problem. Sharding method divides the blockchain network into

small pieces. Networks with fewer nodes are created instead of a more extensive network. Therefore, it becomes more important for every node in the network to be reliable. Using adaptive learning-based methods for this process will contribute to the safe and reliable use of blockchain pieces. The probability of each piece breaking and affecting the entire blockchain will be reduced. We used our model to shard the blockchain network and we see that using reputation value increases shard's reliability in our test results.

Keywords: Consensus committee, the blockchain, BFT, PBFT, online learning, sharding

TEŞEKKÜR

TEŞEKKÜRLER ...

Her şeyden önce danışmanlarım Dr. Öğr. Üyesi Adnan ÖZSOY ve Prof. Dr. Hayri SEVER değerli tavsiye ve rehberliklerinden ötürü ayrıca bu tezin her aşamasında bilgi, deneyim, motivasyon ve teşvikleriyle beni destekledikleri için teşekkür ederim.

Ayrıca tez kurulu üyelerime Prof. Dr. Ali Aydın SELÇUK, Doç. Dr. Oğuz YAYLA, Prof. Dr. Süleyman TOSUN ve Dr. Öğr. Üyesi Kamer KAYA' ya bu tezi gözden geçirdikleri ve içgörülü yorumlar yaptıkları için teşekkür ederim.

Tez konusunun seçimindeki rehberlik ve çalışmalarımdaki destekleri için meslektaşlarıma, özellikle Dr. Serdar Murat ÖZTENER ve Dr. Yasin KAVAK'a, iyi dilekleri için tüm arkadaşlarıma teşekkür ederim. .

Son olarak, eğitim hayatım boyunca bana inandıkları için aileme derinden minnettarım. Beni her zaman en iyi dilekleriyle destekliyor ve teşvik ediyorlar.

İÇİNDEKİLER

Sayfa No

ÖZET i

ABSTRACT iii
TEŞEKKÜR v
İÇİNDEKİLER viii
ŞEKİLLER DİZİNİ x
ÇİZELGELER DİZİNİ xi
1. GİRİŞ 1
1.1 Motivasyon 1
1.2 Ana Katkılar 3
1.3 Tezin Organizasyonu 4
2. ARKAPLAN 5
2.1 Blokzincir Nedir? 5
2.2 Fikir Birliği Algoritmaları 8
2.2.1. İş Kanıtı (PoW) 8
2.2.2. Hisse Kanıtı (PoS) 9
2.2.3. Byzantine Fikir Birliği 9
2.2.4. Parçalama(Sharding) 10
3. İLGİLİ ÇALIŞMALAR 12
3.1 Parçalama Kullanmayan Yöntemler 14
3.1.1. Byzantine Fault Tolerance (BFT) 14
3.1.2. Practical Byzantine Fault Tolerance (PBFT) 15
3.1.3. ByzCoin 16
3.1.4. Tendermint 16
3.1.5. Solidus 17
3.1.6. Melez Fikir Birliği 18
3.2 Parçalama Kullanan Yöntemler 19
3.2.1. OmniLedger 19

3.2.2. Elastico 20
3.2.3. RsCoin 22
3.2.4. Aspen ile Servis Odaklı Parçalama 23
3.2.5. Chainspace: Parçalanmış Akıllı Sözleşmeler Platformu 24
3.2.6. PolyShard: Kodlu Parçalama Aynı Anda Doğrusal Ölçeklendirme Verimliliği ve Güvenliğini Sağlar 25
3.2.7. Doğal Parçalama ile Değer Aktarımı için Ölçeklendirilmiş Bir Blokzincir 26
3.2.8. Tüm Boyutların Parçalandığı Blokzincir(MultiVAC) 27
3.2.9. Blokzincir Sistemlerini Parçalama Yoluyla Ölçeklemeye Doğru 29
3.2.10. Harmony 30
3.2.11. RapidChain 31
3.2.12. Algorand 31
3.2.13. Zilliqa 31
3.3 GURU: Dağıtılmış Uzlaşma Protokolü için Evrensel İtibar Modeli 32
4. FİKİR BİRLİĞİ KOMİTESİ OLUŞTURMA MODELİ 33
4.1 Arkaplan 33
4.2 Model Genel Yapısı ve Adaptif Hedge Algoritması Temel İlkeleri 36
4.3 Özellik Seçimi 38
4.4 Adaptif Hedge Algoritmasının Uyarlanması 39
4.5 Komiteye Seçim 42
4.6 Saldırı Analizi 42
4.7 Deneysel Çalışmalar 43
4.7.1. Simülasyon Kurulumu 44
4.7.2. Ağırlıkların Güncellenmesi Testi 46
4.7.3. Aday Komitedeki Hatalı ve Dürüst Düğüm Sayısının Başarı Üzerindeki Etkisi Testi 48
4.7.4. Güvenilir Komitedeki Hatalı ve Dürüst Düğüm Sayısının Başarı Üzerindeki Etkisi Testi 50
4.7.5. Davranış Değişikliklerine Karşı Modelin Başarısının Testi 51

4.7.6. Düğüm Değiştirilecek Tur Sayısının Model Başarısı Üzerindeki Etkisi Testi ... 52
4.7.7. Güven Değeri Hesaplanırken Geçmiş Güven Değeri Sayısının Başarı Üzerindeki Etkisi ... 53
4.7.8. Örneklemden Seçim ... 54
4.7.9. Örneklem Katsayısının Başarı Üzerindeki Etkisi ... 56
4.7.10. Değiştirilecek Düğüm Sayısının Sistem Başarısı Üzerindeki Etkisi ... 56
4.7.11. Performans Kıyaslaması ... 57
5. PARÇALAMA MODELİ ... 59
5.1 Giriş ... 59
5.2 Arkaplan ... 59
5.3 Genel Yapı ve Temel İlkeler ... 61
5.4 İdeal Parça Sayısının ve Parçadaki Düğüm Sayısının Bulunması ... 62
5.5 Parçalara Atama Yöntemleri ... 62
5.6 Toplam Güven Değeri En Yakın Yöntemi ... 63
5.7 Parçalardan Seçim Yöntemi ... 64
5.8 Deneysel Çalışmalar ... 64
5.8.1. Dağıtım sonrası parça bilgileri ... 65
5.8.2. Düğüm sayısına göre dağılım varyansı ... 65
5.8.3. Hatalı düğüm oranı sabitken toplam düğüm sayısının dağılım üzerindeki etkisi ... 66
5.8.4. Parça sayısının dağılım üzerindeki etkisi ... 66
6. SONUÇ VE SONRAKİ ÇALIŞMALAR ... 71
KAYNAKLAR ... 73
ÖZGEÇMİŞ ... 84

ŞEKİLLER DİZİNİ

Sayfa No

2.1 Blokzincir Örnek Gösterimi 6
2.2 Ağ örnekleri 7

4.1 Sistemin çalışma adımları 37
4.2 Para Miktarı ve Cevap Türü Sabit, Cevap Süresi değişken..... 47
4.3 Cevap Süresi ve Cevap Türü sabit, Para Miktarı değişken 47
4.4 Bütün özellik değerleri sabit..... 48
4.5 Aday komitedeki düğüm ve hatalı düğüm sayısının seçim üzerindeki etkisi ... 49
4.6 Güvenilir Komitedeki Hatalı ve Dürüst Düğüm Sayısının Başarı Üzerindeki Etkisi Testi..... 50
4.7 Davranış Değişikliklerine Karşı Modelin Başarısı 51
4.8 Davranış Değişikliklerine Karşı Modelin Başarısı 52
4.9 Düğüm Değiştirilecek Tur Sayısının Model Başarısı Üzerindeki Etkisi 53
4.10 Geçmiş güven değeri sayısının seçim üzerindeki etkisi 54
4.11 Örneklemden Seçim-Aday düğüm sayısının sistem başarısı üzerindeki etkisi.. 55
4.12 Örneklemden Seçim-Güvenilir komitedeki düğüm sayısının sistemin başarısı üzerindeki etkisi..... 56
4.13 Örneklem Katsayısının Başarı Üzerindeki Etkisi..... 57
4.14 Değiştirilecek Düğüm Sayısının Başarı Üzerindeki Etkisi..... 58

5.1 Düğüm sayısına göre dağılım varyansı..... 67
5.2 Düğüm sayısına göre dağılım varyansı..... 68
5.3 Hatalı düğüm oranı sabitken 5 parça için dağılım varyansı 68
5.4 Hatalı düğüm oranı sabitken 7 parça için dağılım varyansı 69
5.5 Hatalı ve toplam düğüm sayısı sabitken (sırasıyla 165, 500) parça sayısının dağılım üzerindeki etkisi..... 69

5.6 Hatalı ve toplam düğüm sayısı sabitken (sırasıyla 33, 100) parça sayısının dağılım üzerindeki etkisi..... 70

ÇİZELGELER DİZİNİ

4.1 Başarı Oranı Karşılaştırması 58

5.1 Parça sayısı 3 iken düğüm dağılımları 65

5.2 Shard sayısı 5 iken düğüm dağılımları 66

1. GİRİŞ

Bu bölümde bu çalışmanın blokzincir literatüründeki hangi probleme çözüm olarak yapıldığı, yaptığımız katkılar ve tezin organizasyonu anlatılmıştır.

1.1 Motivasyon

Kripto para birimlerine artan ilgiyle birlikte, arkalarındaki blokzincir teknolojisi son yıllarda popülerlik kazanmıştır. Blokzinciri, ağdaki katılımcılar tarafından doğruluğu garanti edilen değiştirilemez herkese açık kayıtlardan oluşan bir veri yapısı olarak tanımlayabiliriz[2]. Blokların geçerli olması için, blokzincir ağındaki düğümlerin geçerliliği konusunda bir fikir birliğine varmaları gerekir. Sonuç olarak, işlemleri onaylama konusunda güvenilir bir üçüncü taraf otoritesine gerek yoktur. İşlem sayısı arttıkça, fikir birliği algoritmasının karmaşıklığı ve sınırlı blok büyüklüğü nedeniyle ölçeklenebilirlik problemi ortaya çıkar. En iyi bilinen fikir birliği algoritması, saldırganın çok sayıda kimlik oluşturarak ağı ele geçirme(Sybil)[3] saldırılarına karşı dayanıklı bir yöntem olan İş Kanıtı (PoW)[4] yöntemidir. Blokzincirde kullanılan PoW yöntemi ise *Hashcash*[5] yöntemidir. Blokzincir fikir birliği problemi için tek veya kesin bir çözüm yoktur. Blokzincir için güvenli ve pratik fikir birliği algoritmaları tasarlamak için daha fazla çalışma gereklidir. [6] 'de yoğun şekilde tartışıldığı üzere, PoW gibi ana blokzincir fikir birliği algoritmaları ve farklı uygulamaları (örneğin, Bitcoin [2] ve Ethereum [7]), fikir birliği problemine bir çözüm sunsalar da, yüksek enerji tüketimi ve uzun mutabakat süreleri gibi büyük dezavantajlara sahiptir.

Bizans Hata Toleranslı (BFT) bazlı yöntemler de ölçeklenebilirlik problemini çözmek için PoW'a alternatif olarak fikir birliği oluşturmak için yaygın olarak kullanılmaktadır. BFT yönteminin amacı, zararlı düğümlere sahip bir ortamda sistemin bozulmaya karşı sürekliliğini sağlamaktır [8]. BFT tabanlı yöntemlerde mutabakat aşaması, bu tür PoW yaklaşımlarından daha az zamanda ve daha düşük maliyetlerle gerçekleştirilebilir. Bir blokta fikir birliğine

ulaşmak için BFT temelli yöntemlerde, önce fikir birliği komitesi oluşturulmalıdır. Düğümlerin güvenilirliğinin bilinmediği herkese açık blokzincir ortamında, bu komiteyi oluşturmak zordur. Ayrıca, bir ağdaki tüm düğümler, performans problemi nedeniyle fikir birliği komitesi olarak kullanılamaz.

Genellikle, BFT tabanlı yaklaşımlar ağdaki düğümlerin bir alt kümesini kullanır. Bunun bir sonucu olarak alt küme seçimi, üzerinde çalışılması gereken bir sorun olarak ortaya çıkar. Bu çalışmada bu soruna bir çözüm olarak karar tabanlı çevrimiçi öğrenme algoritması kullanmayı öneriyoruz. Düğüm güven değerlerini özelliklere göre hesaplıyor ve fikir birliği komitesi seçiminde kullanıyoruz. Modelimizi değerlendirmek ve doğruluğunu test etmek için bir blokzincir simülasyon ortamı oluşturduk ve kullandık. Dokuz farklı test yaptık. Sonuçlara göre, önerdiğimiz model fikir birliği komitesini yüksek güven değerine sahip olan düğümlerden oluşturmaktadır ve modelimiz davranış değişikliklerine Şekil 4.7 ve 4.8' de görüldüğü üzere başarılı bir şekilde adapte olmaktadır. Seçilecek düğüm sayısına bağlı olarak, kötü niyetli aday düğümlerin sayısı yüzde 50'den fazla olsa bile, güven değeri yüksek olan düğümler seçilmektedir.

Ölçekleme probleminin çözümü için son zamanlarda blokzincir parçalama(sharding) yöntemi çokça başvurulan bir yöntem olmuştur. Blokzincir parçalama yöntemini; parçaların yönetilmesini kolaylaştırmak ve isteklere daha hızlı cevap vermek için bir bütünün farklı parçalara bölünmesi olarak ifade edebiliriz. Parçalama yöntemi, veri tabanındaki yükü azaltmak ve performans iyileştirmeleri elde etmek için veri tabanı yönetim sistemlerinde uzun zamandan beri kullanılmaktadır; fakat blokzincir dünyasında yenidir. Parçalama yöntemi kullanarak aşağıdaki faydalar elde edilmek istenir:

- Ana blokzincir ağındaki düğümler, her bir parçanın kendi blokzincirini çalıştırmasına izin veren N parçaya bölünmüştür. Bu sayede işlem sayısında artış sağlanır.
- Ağdaki bütün düğümlerin birbiriyle haberleşmesi yerine sadece ilgili parçadaki düğümlerle haberleşmesi sağlanarak ağ iletişimi etkinliği sağlanır.

- Bir düğüm ağa katılmak için ağdaki bütün blokları indirmek yerine sadece ilgili parçadaki blokları indirerek depolama etkin kullanılmış olur.

Blokzincir parçalama yöntemi uygulanırken ağdaki düğümler parçalara ayrıldığından her bir parçanın güvenilir olması için parçaların yeterince büyük olmasına dikkat edilmelidir ya da düğümlerin seçimine daha fazla önem verilmelidir. Literatürdeki çalışmalarda atama işleminde yönelim tahminine dayanıklı rasgelelik (Verifiable Random Functions (VRF)[9]) kullanılır. Diğer yöntemlerden farklı olarak bu çalışmada elde edilen güven değerleri kullanılarak düğümlerin parçalara atanması gerçekleştirilir. Bu sayede parçalardaki hatalı düğüm sayısı minimumda tutulur.

1.2 Ana Katkılar

Bu tezin odak noktası blokzincirin ölçekleme problemi olmuştur. Bu problemi incelediğimizde, problemin altında yatan temel sebebin fikir birliği algoritmasının zaman maliyetli ve karmaşık olmasının olduğunu fark ettik. Yaptığımız araştırmalara göre fikir birliği algoritmalarından BFT tabanlı algoritmaların ve parçalama yönteminin bu problemin çözümünde kullanıldığını gördük. Bu yöntemlerin blokzincire uyarlanabilmesi için farklı fikir birliği komitesinin seçimi ve düğümlerin parçalara atanması gibi yeni problemlerin ortaya çıktığını gördük. Bu çalışmamızda yeni problemlere çözüm önerdik. Özetle aşağıdaki gibi katkılar gerçekleştirdik:

- Fikir birliği komitesi oluşturulmasında çevrimiçi karar tabanlı adaptif makine öğrenmesi kullanımı
- Blokzincir ağındaki düğümlere güven değeri atayarak düğümlerin dürüst davranmaya teşvik edilmesi
- Düğümlerin parçalara atanmasında güven değerinin kullanılması
- Parçaların bozulma olasılığının azaltılması

1.3 Tezin Organizasyonu

Tezin geri kalanı şu şekilde düzenlenmiştir: 2. bölüm, blokzincir ile ilgili detaylı bilgileri içermektedir. 3. bölümde, ilgili çalışmaların detayları anlatılmıştır. Bölüm 4'te ise fikir birliği komitesi seçiminde önerilen model tasarımı ve algoritması açıklanmaktadır. 5. bölüm fikir birliği komitesi seçimine ait deneysel çalışmalar ve sonuçlar hakkında ayrıntılı bilgi verir. Bölüm 6'da 4. bölümde anlatılan algoritmanın blokzincir parçalamasına uyarlanması çalışmaları yer almaktadır. Son olarak, sonuç bölümü ve gelecekteki çalışmalar sunulmaktadır.

2. ARKAPLAN

Bu bölümde blokzincir ile ilgili temel bilgiler anlatılacaktır.

2.1 Blokzincir Nedir?

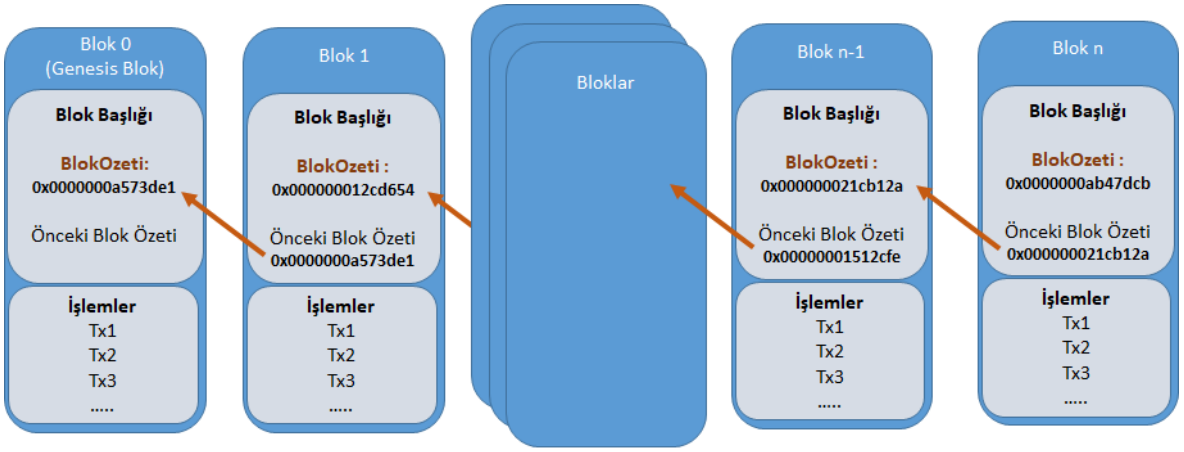
Blokzincir; işlem hareket bilgilerini, blok oluşturulma zamanını ve önceki bloğun kriptografik özet (hash) bilgisini içeren blokların birbirine bağlanmasıyla elde edilen bir veri yapısıdır. Bir blok iki temel alana ayrılmıştır. Bunlar blok başlığı ve blok detayı olarak ifade edilir. Bloğun daha kolay doğrulanabilmesi için blok başlığı bilgisi kullanılır. Blok başlığı bloğu tanımlayan meta veri olarak kabul edilir. Blok başlığında blok özet değeri, önceki blok özet değeri, Merkle kök değeri ve oluşturulma zamanı gibi temel bilgiler yer alır. Her bir öğenin tanımı aşağıda verilmiştir.

- **Özet Değeri:** Kriptografik özet algoritması bloğa uygulanarak bir değer elde edilir ve bu değer bloğun kimliğidir.
- **Önceki Blok Özet Değeri:** Önceki bloğun kimlik bilgisidir. Bu değer sayesinde blokzincirin tutarlılığı değiştirilemez olması garanti edilir. Şekil 2.1 de gösterildiği üzere önceki blok özet değeri ilk bloğa kadar gider. Bu sayede blokların değiştirilmesi engellenmiş olunur. Blokzincirdeki kayıtları değiştirebilmek için yayınlanan bloklarla birlikte önceki blok özet bilgilerini de güncellemek gerekir. Bu işlemi yapabilmek için oldukça fazla işlem gücü gerekmektedir. Bu nedenle işlem gücünün çoğunluğu tek bir yerde toplanmadıkça blokzincirdeki kayıtlar değiştirilemez.
- **Merkle kök değeri:** Blok içindeki işlem bilgileri blok başlığında tutulmaz. Fakat bütün bloğa ihtiyaç duymadan istenilen hareketin blokta olup olmadığının doğrulanmasına ihtiyaç vardır. Bu nedenle Merkle ağacı [10–14] oluşturulur ve kök değeri blok başlığında saklanarak istenilen harekete ulaşılabilmesine imkan tanınır.
- **Oluşturulma zamanı:** Bu değer bloğun oluşturulduğu zamanı gösterir. Bu bilgi iki kez harcamanın (double-spending) önlenmesinde kullanılır.

Bir bloğun blokzincire eklenmesi için ağdaki düğümlerin blok üzerinde uzlaşmaya varması gerekir. Kullanılan uzlaşma algoritmasına göre örneğin PoW kullanıldığı bazı durumlarda aynı anda farklı bloklar yayınlanabilir. Hangi bloğun ve zincirin geçerli olduğunu sonraki bloklar belirler. En uzun zincir doğru kabul edilerek yeni bloklar bu zincire eklenir. Bunun gibi farklı zincir oluşması durumuna *çatallanma* denir. Blokzincirde üç farklı nedenden ötürü çatallanma oluşabilir:

- **Geçici çatallanma:** Aynı anda birden fazla blok oluşmasından kaynaklı birden fazla blokzincir oluşması durumuna *geçici çatallanma* (fork) denir.
- **Yumuşak çatallanma (soft fork):** Blokzincirde kullanılan yazılımda değişiklik yapıldığı zaman geçmişteki bloklar da doğru kabul ediliyorsa buna *yumuşak çatallanma* denir.
- **Sert çatallanma (Hard fork):** Blokzincirde kullanılan yazılımda değişiklik yapıldığı zaman geçmişteki bloklar doğru kabul edilmiyor ise *sert çatallanma* durumu oluşur.

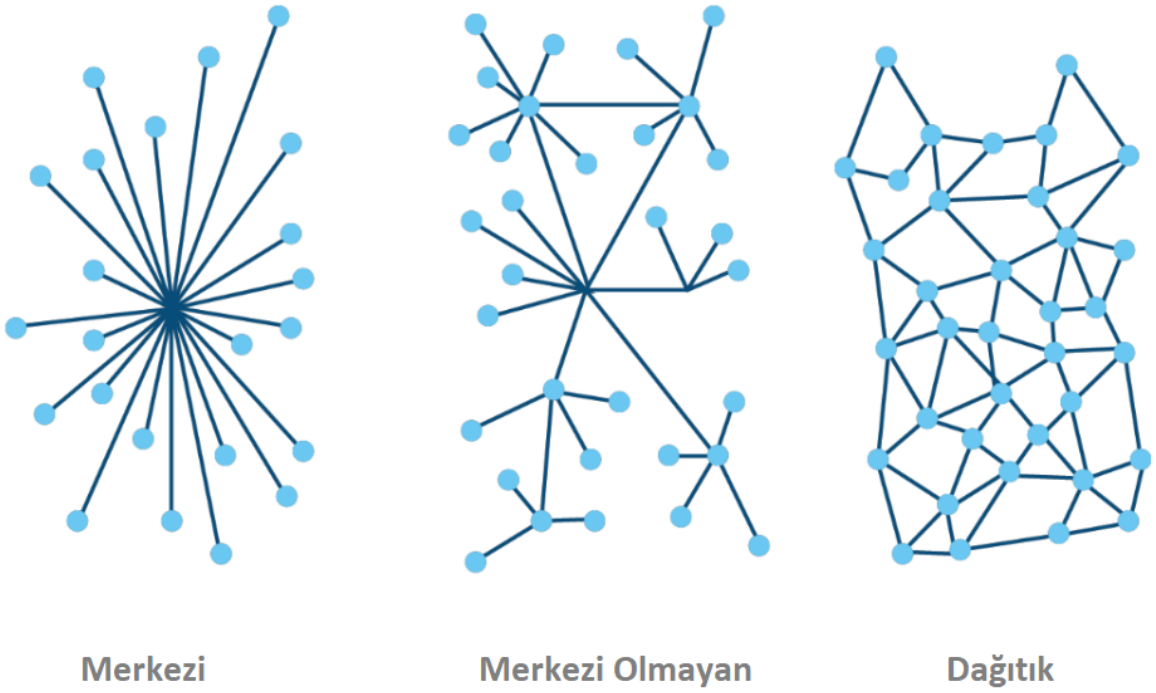
Yumuşak ve sert çatallanma durumlarında önceki blokzincir yaşamına devam edebilir ve yeni blokzincirlerde yeni kurallara uygun olarak yaşamına devam eder.



Şekil 2.1: Blokzincir Örnek Gösterimi

Blokzincir ağında işlemlerin doğrulanması için güvenilir bir merkezi otoriteye ihtiyaç yoktur. İşlemlerin doğruluğuna ağdaki düğümler karar vermektedir. Alışıldık istemci-sunucu mimarisinde merkezi bir yere bağlanılarak işlemler gerçekleştirilirken blokzincirde ise dağıtık

mimariye uygun her istemci doğrudan birbiriyle iletişime geçebilir ve işlem yapabilir. Şekil 2.2'de örnek ağ yapıları gösterilmektedir. Klasik veri tabanlarında her veri, merkezi veya merkezi olmayan sunucularda tutulabilirken blokzincirde ise bilgilerin tamamının birer kopyası ağdaki tüm düğümlerde bulunur. Bir düğümdeki veri bozulsa bile diğer düğümlerden bilgileri kurtarabilir.



Şekil 2.2: Ağ örnekleri

Blokzincirin temelde iki farklı türü vardır. Bunlar açık blokzincirler ve özel blokzincirlerdir.

- **Açık Blokzincirler:** Açık blokzincir ağlarında ağa katılmak isteyen her kullanıcı izin almadan ağa katılabilir. Ağdaki düğümler herhangi bir kısıtlama olmadan kurallara uygun olduğu sürece geçmiş kayıtlara erişme, yeni kayıt oluşturma, blok doğrulama gibi

ağdaki bütün işlemleri gerçekleştirebilir. Bitcoin, Ethereum gibi blokzincir kullanan kripto paralar açık blokzincirlere örnek olarak verilebilir.

- **Özel Blokzinciler:** Özel blokzincir ağlarında ağa katılmak izne bağlıdır. Özel blokzincirler genelde birbiriyle ilişkili kuruluşların kendi arasında oluşturduğu blokzincirlerdir. Blokzincirin güvenlik seviyesini ve blokzincir ağına katılma kurallarını blokzinciri kuran kuruluş belirlemektedir. Daha kısıtlı kullanıcılara açık olduğu için merkezileşme riski vardır.

2.2 Fikir Birliği Algoritmaları

Bir bloğun blokzincire eklenmesi için ağdaki düğümlerin o blok üzerinde fikir birliğine varması gerekir. Blokzincir dünyasında farklı fikir birliği algoritmaları kullanılmaktadır.

2.2.1. İş Kanıtı (PoW)

En çok bilinen fikir birliği algoritması PoW algoritmasıdır. Bu algoritmada bloğa bir değişken eklenerek yeni bir kriptografik özet değeri oluşturulur. Oluşan kriptografik özet değeri zorluk derecesini karşılıyorsa PoW çözümü bulunmuş demektir. Denklem 1, örnek bir PoW formülünü göstermektedir. Bu formülde *H* kriptografik özet fonksiyonunu, n değişkeni ve D zorluk derecesini ifade etmektedir. Eşitsizlik doğrulanıncaya kadar değişken değer değiştirilerek yeniden kriptografik özet alınır. Eşitsizlik doğrulanınca blok ve değişken değer bütün ağa gönderilir. Ağdaki düğümler bu bilgileri kontrol eder ve doğruysa kendi blokzincirlerine eklerler. Zorluk derecesi arttıkça değişkeni bulmak için daha fazla işlem yapılması gerekir. PoW işlem maliyeti ve enerji gereksinimi çok yüksek olan bir yöntemdir. Bu nedenle daha az maliyetli fakat güvenli ve güvenilir fikir birliği yöntemleri araştırılmaktadır.

$$H(blok\_ozet||n) <= 2^{\alpha} - D \tag{1}$$

2.2.2. Hisse Kanıtı (PoS)

PoW alternatifi olarak geliştirilen bu yöntemde blok doğrulaması yapabilmek için hisseye (kripto para) sahip olmak gerekir. Bu yöntemin temel prensibi; hissesi fazla olan kişiler doğal olarak güvenilir kişilerdir ve sistemin bozulmaması, kurallara uygun olarak işlemesi için çalışırlar şeklindedir. Hisse miktarıyla madencilik gücü orantılıdır. Örneğin bir doğrulayıcı sistemdeki toplam hisse değerinin % 5'ine sahipse en fazla yayınlanan blokların % 5'ini yayınlayabilir. Hisse kanıtı ilk olarak [15] çalışmasında kullanılmıştır. Sonrasında [16–18] çalışmaları da PoS yöntemini kullanmışlardır. Blok yayınlayacak düğüm seçiminde sadece hisse miktarına bakmak, sistemi en çok hisseye sahip kişinin tekeline dönüştüreceği için hisse miktarıyla birlikte para yaşı, en düşük özet değeri gibi farklı özelliklere bakmayı da gerektirir. Bu fikir birliği yönteminde kaybedecek bir şey yok(nothing at stake) [19] saldırısına karşı savunmasızdır. Blokzincirde çatallanma olduğunda çatallanmanın çözülmesi için sonraki düğümlere bakılır. Bu saldırıya göre düğümler çatallanmanın çözümüyle ilgilenmez. Bunun yerine kazancını artırabilmek için ve kendi hisse miktarını etkilemediği için oluşan çatalların hepsinde blok yayınlamaya devam eder. Bu saldırıyı engellemek için sonraki çalışmalar düğümün hissesinden ceza kesmek gibi farklı yöntemler geliştirmişlerdir.

2.2.3. Byzantine Fikir Birliği

Diğer bir fikir birliği alternatif yöntemi BFT tabanlı yöntemlerdir. Bu yöntemlerde yayınlanacak bloğa bir kişi yerine bir grup tarafından karar verilir. Karar veren grup içinde hatalı davranan düğümlerin olduğu kabul edilir ve önerilen yöntemin bu şartlar altında çalışması sağlanır. Bu yöntemlerde blok yayınlayan gruptaki hatalı düğüm sayısı grup düğüm sayısının 1/3'ünden fazlaysa yöntemler doğru çalışmazlar. Bu yöntemlerde lider düğüm bir öneride bulunur ve diğer düğümler bu öneriyi oylayarak önerinin kabul edilip edilmeyeceğine karar verilir. Bizans fikir birliği için PBFT yöntemi [20] ilk önerilen yöntem olmuştur. PBFT yönteminde düğüm sayısı arttıkça sistemin performansı düşmektedir. Daha sonra iletişim

maliyetini düşürerek sistem performansını artıracak HoneyBadger[21], BFT-Smart[22] gibi yöntemler geliştirilmiştir.

2.2.4. Parçalama(Sharding)

Bir bütünü farklı parçalara bölerek parçaları daha kolay yönetme ve daha hızlı işlem yapabilme özelliği kazandırma işlemine parçalama(sharding) denir. Parçalama yöntemi; veri tabanı yönetim sistemlerinde veri tabanı üzerindeki yükü azaltmak için ve performans iyileştirmelerinde uzun süredir kullanılmaktadır. Yatay ölçekleme olarak da bilinmektedir. Parçalama için müşteri tablosunu coğrafi bölgelere göre bölmek bir örnek olarak gösterilebilir. Her bölge için farklı veri tabanları oluşturularak diğer bölgelerden etkilenmeden çalışması sağlanır.

Blokzincir dünyasında parçalama ise bir düğüm sadece bulunduğu parçadaki bilgileri tutmakla yükümlü olacaktır. Düğüm, bütün blokzinciri bilmek yerine sadece ilgilendiği parçaların bilgisine sahip olacaktır. Bu sayede depolama ve işlem yükü bakımından ölçekleme sağlanacaktır. Blokzincir parçalara ayrılırken blokzincirin temel ilkeleri olan merkezi olmama ve güvenli olma özelliklerinin kaybedilmemesine özellikle dikkat edilmelidir. Parçalama sadece hareketlerin bölünmesiyle sınırlı kalmamalıdır. Hareketlerle birlikte veri saklama ve ağ iletişiminin de bölünmesi birlikte düşünülürse blokzincir ticari ürünlerle yarışabilir hale gelir. Hareketler farklı parçalara bölünerek eş zamanlı işlenmesi sağlanır. Bu sayede birim zamanda işlenen hareket sayısı artar. Parça sayısı arttıkça eş zamanlı işlenen hareket sayısı da artar. Ana blokzincir ağındaki düğümler K alt ağlara bölünerek hareketlerin paralel işlenmesine izin verilir. Alt blokzincirlerde oluşturulan blokların ana blokzincire eklenmesi veya ana blokzincirle alt blokzincirlerin iletişimde olması sağlanarak blokzincir bütünlüğü sağlanmaya çalışılır. Her bir parça kendi içinde blok doğrulamasını ve yayınlamasını yapar. Parça içi iletişim genel blokzincir işleyişinden farklı değildir fakat parçalar arası iletişim için farklı yöntemlerin oluşturulması gerekmektedir.

Her düğüme her hareketi işlemeyi bıraktırdığınızda, geçersiz işlemlerin bir kötü amaçlı düğüm kümesi tarafından kaydedilebileceği ve denetlenmeden kalma riskini teorik olarak yükseltirsiniz. Ağdaki düğüm sayısı arttıkça, ağı hareketlerin işlenmesi yükünü paylaşabilecek parçalara ayırma kabiliyeti artar ve aynı zamanda her bir parça güvenilir uzlaşma için yeterince büyük kalır. Her blokzincir kendi içinde uzlaşma sağlayarak blok yayınlaması yaparken blokzincirler arası iletişim için farklı blokzincirler irtibata geçirilir. Bu sayede işlem ve depolama yükü bölünmüş olur. Fakat blokzincirlerin iletişimi protokollerinin farklılığından dolayı daha zor olacaktır. Bu nedenle blokzincirler üstü bir blokzincir kurulması gerekir ki bu blokzinciri parçalara ayırmaktan daha zor olacaktır. Çünkü farklı kültüre sahip blokzincirleri aynı masa etrafında toplayıp birlikte çalışmaya ikna etmek gerekir.

Veri tabanlarında sıkça kullanılan bu modelin blokzincirin ölçeklenebilirlik problemine de çözüm olup olmayacağı üzerine araştırmalar son zamanlarda oldukça artmıştır[23]. Blokzincirde parçalama kullanabilmek için çözülmesi gereken bazı problemler vardır.

- Ağa katılmak isteyen düğümlerin parçalara ve fikir birliği komitesine atanması için bir yöntem oluşturulmalıdır.
- Parçalara ayırma yapısında hareketler farklı komiteler üzerinde paralel bir şekilde işlenmeye çalışılır. Eğer bir hareket başka bir harekete bağlıysa diğer bir ifadeyle bir hareketin doğru kabul edilmesi için başka bir hareketin gerçekleşmesi gerekiyorsa bu hareketlerin paralel olarak işletilmemesi gerekir. Aslında sıralı işletilince hata almayacak bir hareket paralel işletilince hata alacaktır. Bu durum sistemin tutarlılığını bozacağı için parçalamada bu soruna dikkat edilmeli ve bu sorun çözülmelidir.
- Parçalar arasındaki iletişimin sağlanması, farklı parçalar arasındaki transferlerin yönetilmesi, bir parçanın bozulması düşünülmeden yapılan tasarım bütün blokzinciri etkileyebilir.

3. İLGİLİ ÇALIŞMALAR

Üye sayısının bilinmediği veya üye sayısının yüksek olduğu blokzincir ağlarında bir fikir birliği komitesi oluşturmak çok önemlidir. Bitcoin ve türevleri gibi blok yayınlanmasının PoW ile yapıldığı ağlarda fikir birliği komitesi ağdaki tüm düğümler olarak kabul edilir. PoW, [6] referansı ile verilen çalışmada belirtildiği gibi enerji ve işlem süresi açısından çok maliyetlidir. Bu maliyetleri azaltmak için [7, 15] çalışmalarında PoW yerine PoS kullanılmıştır. Ancak, PoS'un bağımsız kullanımı güvenlik için yeterli değildir (tehlikede olan hiçbir şey yoktur [19]). Literatürde, genelde fikir birliği komitesinde, sistemdeki tüm düğümler yerine ağın alt kümesini kullanılır. Birçok çalışmada [24–33] blok doğrulaması için PoW ve PoS'a alternatif yöntemler kullanmaktadır. Bazı çalışmalarda [24–29] alt kümeye katılmak isteyen düğümler PoW [13-18] çözerek bilgilerini kanıtlamaktadır. ByzCoin [24] komiteye katılmak isteyen düğümlerden belirli bir süre boyunca PoW çözmesini ister. Bu süre sonunda komite yenilenirken PoW çözen düğümlere çözümlerine göre oy hakkı vererek komiteye dahil eder. Blok yayınlamada PoW yerine PBFT kullanılır. PoW komiteye düğüm seçimi için kullanılır. OmniLedger [25] çalışması ByzCoin temellidir. ByzCoin'in blokzincire parçalama eklenmiş bir versiyonu olarak düşünülebilir. Komite seçimi ByzCoin ile aynıdır. Elastico [26] açık blokzincir ağında PBFT kullanılmasını önermektedir. Bu yöntemde komiteye katılmak isteyen düğümlerin kimlik bilgileriyle uyumlu PoW çözümünü sunmaları beklenmektedir. Blok doğrulaması dönemlere bölünür ve her dönemin sonunda komite yenilenir. Solidus [27], ByzCoin benzeri bir komite seçimi yapar. Diğer çalışmalardan farklı olarak, komitedeki düğümlerin doğru işleyişini teşvik etmek için yöntemler ekler.

Özel blokzincir ağları için yapılan çalışmalarda [30, 31] uzlaşma seçimi statik olarak yapılmıştır çünkü sistemdeki kullanıcı bilgileri bilinmektedir. RsCoin [30]'da ağın yönetiminden bir merkez bankasının sorumlu olduğu varsayılmaktadır. Tendermint [31], komiteye katılabilmek için tanımlayıcı bilgileri olan belirli bir miktar mevduat gerektirir. Hatalı davranışlar sergileyen düğümlerin başlangıçta bıraktığı depozitten ceza kesilebilir. Ağdaki düğüm bilgisi Ripple [32] gibi önceden bilindiği durumlarda, fikir birliği komitesi başlangıçta her

düğüm için belirlenir ve uygulamada değişmez. Statik seçim yapıldığında veya komite değişmediğinde, blokzincir merkezileşmeye doğru hareket eder. Bu durum, blokzincirin merkezi olmayan mimarisiyle eşleşmemektedir. Stellar [33] her bir düğümün kendi güvenilir komitesini seçtiği bir yöntem önerir. Komite seçimi düğümlere bırakılmıştır. Doğru blokları elde etmek için düğümlerin doğru yapılandırmayı oluşturması gerekir.

P2P ağları üzerinde bir düğümün itibarını bulmak için çalışmalar kapsamlı bir şekilde yapılmıştır [34–39], ancak doğrudan blokzincir alanı ile ilgili değillerdir. Uzlaşma komitesinin seçimi için GURU olarak adlandırılan bir çalışma [40] yapılmıştır. GURU çalışmasında da bizim çalışmamıza benzer bir şekilde tam bir blokzincir işlemi yerine sadece komite seçimi yöntemi önerilmiştir. GURU çalışmasında güven değeri hesaplanmasında sadece blok doğrulama sonucuna bakılırken bizim çalışmamızda blok doğrulama sonucuna ilaveten hisse miktarı ve cevap süresi de kullanılır. Böylece güven değeri hesaplamada daha doğru sonuçlar elde edilir. Ayrıca, modelimizde kullanılan özellik vektörü yeni özelliklerle kolayca genişletilebilir. GURU'nun modeli herhangi bir akıllı güven değeri hesaplama yaklaşımı içermezken, modelimizde uzlaşma komitesini oluşturmak için güven değeri hesaplamada gözetimsiz bir çevrimiçi öğrenme şeması kullanılır. Pencere tabanlı (geçmiş güven değerlerinin de hesaba katılması) öğrenme modelimiz, GURU'daki birikimsel yaklaşımın aksine, dürüstten kötü amaçlıya ve kötü amaçlıdan dürüste davranış değişikliği olan düğümlerin seçimi için daha sağlam, güvenli ve daha adildir. Bunlara ek olarak önerdiğimiz model sayesinde her düğüme bir güven değeri atayarak düğümler dürüst davranmaya teşvik edilir. Düğümler birbiriyle işlem yaparken güven değerine bakarak daha dikkatli davranabilir veya akıllı sözleşmeler sayesinde güven değerine göre bazı kısıtlamalar yapılabilir.

Alt bölümlerde blokzincir ağlarında fikir birliği komitesi ve blokzincir parçalama kullanan çalışmaların özetleri anlatılmıştır.

3.1 Parçalama Kullanmayan Yöntemler

Bu bölümde parçalama kullanmayan yöntemler anlatılmıştır. Yöntemlere geçmeden önce temel sayılabilecek Bizans Hata Toleransının ne olduğu kısaca anlatılacaktır.

3.1.1. Byzantine Fault Tolerance (BFT)

BFT yönteminin amacı zararlı düğümlerin olduğu bir ortamda bozulmalara karşı sistemin tutarlılığının devam edebilmesinin sağlanmasıdır. Yanlış davranma bilinçli bir şekilde de olabilir saldırılarla düğümün ele geçirilmesi sonucu da olabilir. BFT algoritması bu tür durumlara karşı dayanıklı olmak zorundadır. Bizans generalleri problemlerine çözüm olarak ortaya atıldığı için bu isimle adlandırılmıştır. Bizans generalleri probleminde [8]; generaller kaleyi kuşatmışlardır ve başarılı olabilmeleri için aynı anda saldırmaları gerekmektedir. Fakat aralarında hain(ler) olduğu bilinmektedir. Buradaki amaç bilinen haine rağmen saldırının başarılı olmasıdır. Bu metafor bilgisayar ağlarında yaşanan problemi ifade etmektedir. Şöyle ki; bir konu hakkında birlikte karar verileceği zaman sistemin tutarlılığını bozmak isteyen düğümlerle birlikte doğru karar verilmeye çalışılmaktadır. Örneğin blokzincire yeni bir blok ekleneceği zaman zararlı düğümler sistemi sabote etmeye çalışabilir. BFT çözümü için genel bir çözüm bulunmamaktadır. Farklı alternatifleri mevcuttur. Örneğin Bitcoinde kullanılan PoW da bir çeşit BFT çözümü olarak kabul edilir. Oylama yöntemi ise PoW'a göre daha kısa sürede ve daha az maliyetle gerçekleştirilebilen bir çözüm yöntemidir. Birim zamanda geçerlilik kazanan hareket sayısını artırmak için tercih edilen yöntem oylama yöntemidir. Oylama yönteminde uzlaşma grubuna bir öneri sunulur ve bu gruptaki belirli çoğunluk kabul oyu verirse öneri geçerlilik kazanır. Bu yaklaşımda hatalı düğüm sayısının toplam düğüm sayısına oranının %33'ten az olduğu kabul edilmiştir.

3.1.2. Practical Byzantine Fault Tolerance (PBFT)

PBFT, BFT probleminin çözümü için geliştirilmiş asenkron ağda güvenlik (safety) ve canlılık (liveness) sağlayabilen bir yöntemdir [20]. Güvenlik; istekler tam sıralı bir yapıdadır. Canlılık; istemci mutlaka gönderdiği her istem için bir cevap alır. Ağdaki her düğüm için bir görünüm (view) tanımı vardır. Sırasıyla her düğüm uzlaşma grubunun yöneticisi olur. PBFT 5 adımdan oluşan bir işleyişe sahiptir:

1. İstem (Request): Bu aşamada istemci istemini yöneticiye gönderir. İstemi alan yönetici mesajı doğrular ve bir sıra numarası atar.
2. Ön Hazırlık (pre-prepare): Yönetici uzlaşma grubundaki bütün düğümlere ön hazırlık mesajını gönderir. Bu sayede mesajın doğrulaması yapılır ve sıra numarası alınır.
3. Hazırlık (prepare): Uzlaşma grubundaki bütün düğümler birbirine hazırlık mesajını gönderir. Bu sayede bütün kopyaların sıra numarası üzerinde uzlaşılır.
4. İşleme (commit): $2f$ (f hatalı olabilecek düğüm sayısı) adet hazırlık mesajını alan düğümler işleme mesajını gönderir. İşleme mesajının gönderilmesi sıralama ve mesaj uygunluk mutabakatının sağlandığı anlamına gelir.
5. Cevap (Reply): Aktif düğümler istemi gönderen düğüme cevap gönderirler. Bu sayede yöneticinin istek ile cevap arasında bozulduğu durumun atlanması sağlanmış olur.

Yöneticinin hatalı davrandığı durumlara çözüm olarak istemi gönderen düğüm zaman aşımı süresi kullanır. Bu süre aşılırsa istem bütün düğümlere gönderilir. İstemi alan düğüm, daha önce istemi almışsa bu istemi göz ardı eder. Eğer istem daha önce alınmamışsa ikinci bir zamanlayıcı başlatır. İkinci zamanlayıcının süresi içinde istem gelmezse yönetici değiştirme sürecini başlatır. Yönetici değişikliği süresince yeni istem kabul edilmez.

3.1.3. ByzCoin

ByzCoin[24], Bizans uzlaşma protokolü kullanarak hareketlerin saniyeler içinde geçerli olmasını sağlayan bir yöntem önermiştir. Bu çalışmada PBFT yönteminin açık üyeliği ve çok sayıda düğümü desteklemesi için PBFT yöntemine uyarlamalar yapılmıştır. ByzCoin PBFT'yi yüzlerce binlerce imzayı etkin bir şekilde birleştirebilen Collective Signing (CoSi [41]) ile birlikte kullanmıştır. CoSi hem PBFT turlarını hem de maliyetlerini düşürmektedir. CoSi consensus yöntemi değildir. PBFT'nin hazırlama ve işleme adımlarının ölçeklenebilir olmasını sağlamaktadır. ByzCoin uzlaşma gruplarını önceki doğrulanan bloklar çerçevesini kullanarak dinamik bir şekilde oluşturur. Daha önce blok doğrulaması yapmış madencilere kriptografik özet güçleri oranında oy hakkı veya pay verir. Ağdaki düğümler ağaç şeklinde tutularak iletişimin daha hızlı olması planlanmıştır. Ağaçtaki kök düğüm Bizans uzlaşmasını yönetir. Ağaçtaki bir düğümde problem olursa alttaki düğümlerle iletişim kesilebileceği için düz iletişim yöntemi de vardır. ByzCoinde blok üzerinde fikir birliğinin sağlanmasından ve düğümler arasındaki iletişimden o anki lider sorumludur. Kötü niyetli bir lider uzlaşma grubundan bazı düğümleri dışlayabilir. Uzlaşma grubundaki kötü niyetli düğüm sayısı toplam düğüm sayısının üçte birinden (%33) fazlaysa ByzCoin yöntemi çalışmaz. Bu oran Bitcoinde %51'dir. Fikir birliği komitesi oluşturulurken PoW yöntemi kullanılmaktadır. ByzCoin, dezavantajlarından kurtulmak için alternatif bir yöntem kullanılmasını ileriye dönük çalışma olarak sunmuştur.

3.1.4. Tendermint

Tendermint [31] klasik bir BFT yöntemidir. PBFT'nin değişik bir varyasyonu uygulanır. PBFT'de hareketler bütün düğümlere gönderilirken Tendermint'te doğrulayıcı düğümlere gönderilir. Doğrulayıcı düğümler sisteme bir miktar depozit bırakmak zorundadırlar ve bıraktıkları depozit oranında oy gücüne sahip olurlar. Hatalı bir oy kullandıklarında veya uzlaşmaya katılacaklarını belirttikleri halde belirli bir süre boyunca çevrim dışı olurlarsa depozitlerinden bir miktar kaybederler. PBFT'yi yöneten lider klasik PBFT'den farklı olarak

sırayla dönüşümlü bir yapıda değişir. Sistemde üç temel bileşenden oluşur: Öneri, oylar (ön oy ve ön doğrulama oyu), kilitleme. Seçilen lider düğüm bir öneriyi bütün bağlantılı olduğu düğümlere yayar. Aynı anda farklı blok yayınlanmasını önlemek için kilitleme bilgisi kullanılır. Lider düğüm eğer bir düğüme kilitlenmişse bu bilgiyi de bloğa ekler. Liderin yaydığı bilgiyi alan düğümler kendi bağlı oldukları düğümlere yayarlar. Ön oy aşamasında doğrulayıcı düğümler eğer daha önce kilitledikleri düğüm varsa onu imzalar ve bu düğümle ilgili ön oy bilgisini yayar. Böyle bir durum yoksa gelen geçerli önerilen blok için ön oy bilgisini yayar. Hatalı bir durumda özel *nil* ön oy bilgisini yayar. Ön oy adımında herhangi bir kilitlenme olmaz. Ön yazma adımının başında bütün doğrulayıcılar bir karar verir. Eğer doğrulayıcılar bir blok için 2/3 oranında ön oy alırsa bu bloğu imzalar ve ön yazma bilgisini yayar. Aynı zamanda bu blok üzerine kilitlenir ve önceki kilitleri serbest bırakır. Ön yazma adımının sonunda her bir doğrulayıcı düğüm karar verir. Eğer düğüm belirli bir blok için 2/3 oranında ön yazma aldıysa yazma adımına geçer. Aksi takdirde bir sonraki tur için öneri adımına geçer.

3.1.5. Solidus

Solidus[27] açık Bizans uzlaşması tabanlı merkezi olmayan bir kripto para birimidir. Solidusda lider seçimi için PoW kullanılarak PBFT açık yapıya uygun hale getirilir. Bencil madenciliği önleme ve dürüst davranmaya teşvik edici bir yapısı vardır. Bitcoine göre blok geçerlilik kazanma süresini düşürürken hatalı kullanıcıların olduğu bir ortamda güvenlik ve canlılık sağlar. PBFT'nin kapalı yapısını açık hale çevirebilmek için belirli periyotlarla değişen komitenin oluşturulması gerekir. Komitenin süresi dolunca da daha fazla onay vermesinin engellenmesi ve yeni komitenin oluşturulması gerekir. Daha önceki çalışmalarda bu aşamaların nasıl yapıldığıyla ilgili detaylı bilgi yer almamaktadır. Diğer bir önemli eksiklik de diğer çalışmalar komitenin çoğunluğunun kurallara uyduğunu kabul eder. Fakat bunu teşvik edecek çalışmaları yoktur. Solidusta blok yayınlaması PoW ile yapılmaz. PoW fikir birliğini yönetecek lider seçimi için kullanılır. Solidusta zaman içinde değişen blok doğrulayan komite vardır. *Genesis* komite başlangıçta sistem üzerinde dahili olarak gelir. Sonrasında komiteye katılmak isteyen kullanıcı PoW çözerek mevcut komiteye sunar. Kabul edilen yeni

kullanıcı lider olur ve blok önerisini komiteye sunar. Yeni bir kullanıcı komiteye kabul edilince komiteden eski kullanıcı çıkarılır. Komiteye katılmak isteyen kullanıcılar aynı anda PoW çözümü sunarlarsa tur numarasına bakılır. Tur numarası kullanıcının derecesini belirler. Daha yüksek dereceli kullanıcı kabul edilir. Oyun teorisi kullanılarak düğümlerinin doğru davranmaya teşvik edilebileceği ispatlanmıştır.

3.1.6. Melez Fikir Birliği

Bilinen bütün açık blokzincir yapılarında ağın senkronize olduğu varsayımı vardır. Diğer bir ifadeyle protokol ağ gecikmesinin bir üst sınırı olduğunu bilir ve hareket doğrulanması bu süreden daha yavaştır. Melez fikir birliği çalışmasında [28] açık blokzincir yapısı üzerinde tepkisel işlemenin (responsiveness) uygulanabilir olup olmadığı incelenmiştir. Tepkisel protokolde hareket geçerliliği sadece gerçek ağ gecikmesine bağlı olmalıdır. Ön bir üst sınıra bağımlı olmamalıdır. Açık blokzincirin tepkisel olması için:

1. Protokolün ağ gecikmesinin üst sınırı olduğunu bilmesi gerekir.
2. Tepkisel olmayan bir ısınma periyodu oluşturarak sonrasında hareket doğrulamasının tepkisel olması sağlanabilir.
3. Dürüstlük yapışkandır: yani dürüst bir düğümün bozulması anlık olmaz. Bunun için belirli bir süre geçmesi gerekir.
4. Düğümlerin 1/3'ünden azı hatalı olmalıdır.

Açık blokzincir ağlarının kapalılardan farkı vardır. Açık ağlarda aşağıdaki özellikler varken kapalı ağlarda bu özellikler bulunmaz veya sınırlıdır.

1. İsteyen katılabilir.
2. Düğümler gelir gider.
3. Kaç tane düğümün olduğuyla ilgili ön bilgi yoktur.

4. Katılan düğüm sayısı zaman içinde değişebilir.

Melez Fikir Birliği çalışmasındaki amaç ağın gerçek performansından faydalanmaktır. Ağdaki işlem gerçekleştiği anda hareketlerin geçerli olması amaçlanır. Komitedeki bütün düğümler her periyotta yenilenmemelidir. Çünkü komiteye katılan düğümlerin farklı bir blokzincir üzerinde çalışması sağlanabilir. Bu çalışmada da ByzCoin [24] benzeri bir yapı vardır. Çevrimiçi madencilerden oluşan komiteler oluşturulur ve bu komiteler klasik PBFT yöntemiyle günlük fikir birliği sağlarlar. Blok sayısı belirli bir sayıya ulaşınca komite üyeleri değiştirilir. Değişim aşamasında bazı bloklar atılır. Blokların atılması tutarlılığın sağlanması için önemlidir. Çünkü bütün düğümlerin hemfikir olduğu bir blokzincir üzerinde çalışılır. Zincir büyümesine bağlı olarak yeni komitenin oluşturulması kısa sürecektir. Geçmişi bozma saldırısını önlemek için damgalama kullanılır. Bir komite süresini tamamladığında günlük kayıtlar için kriptografik özet değeri oluşturur ve komite üyelerine sunar. Komitedeki üye sayısının 1/3 ünden fazla üye imzalarsa blokzincire eklenir. Ana blokzincir üzerinde olmayan BFT diğer bir adıyla *DailyBFT*de komite üyeleri zincir dışında uzlaşma protokolüyle blokzincire eklenecek bloklar oluştururlar. *DailyBFT* sonucu oluşan kayıtlar gün sonunda blokzincire eklenir.

3.2 Parçalama Kullanan Yöntemler

Bu bölümde blokzincirin ölçeklenebilirliğini artırabilmek için parçalama yöntemi de kullanan çalışmalar anlatılmıştır.

3.2.1. OmniLedger

OmniLedger [25]'da ağdaki düğümler iki kategoriye ayrılmıştır: doğrulayıcılar ve izleyiciler. OmniLedger, blokzincir parçalama yapısı kullanarak ve düğümleri rastgele olarak shardlara atayarak ölçeklenebilir, güvenli bir dağıtık defter yapısı sunar. ByzCoin [24] üzerine kurulmuş bir sistemdir. ByzCoin'den farklı olarak parçalama kullanılır. Parça içi ve parçalar arası

tutarlılığı sağlayabilmek için paralel çalışan atomik işlem olarak adlandırdıkları bir yöntem önerilmiştir. Depolama problemine ve yeni başlamanın hızlı olabilmesi problemine çözüm olarak shard durumlarını özetleyen durum blok yapısı kullanılmıştır. Parça sayısıyla orantılı lineer ölçeklenebilme özelliğine sahiptir. Doğrulayıcıları parçalara atamak için PoW kullanılır.

OmniLedger güvenli parçalama sağlamak için tahminlere karşı dirençli dağıtık rasgelelik yöntemi uygular ve bu yöntem çoklu kimlik oluşturma(sybil) saldırılara karşı dayanıklı PoW [4], PoS [15], PoP (proof of personhood) [42] veya özel blokzincirlere uygulanabilirdir. Atomik işlem algoritması kullanılarak parçaların bloğu tamamen sonlandırdığını veya reddettiğini garantiler. Ağa yeni katılan bir düğümün hızlıca ve az kaynak kullanarak adapte olabilmesi için durum blokları önerilmiştir. Doğrulayıcı düğüm bir durum bloğunu yazdığı anda önceki blokları kendinden silebilir. RandHound [43] yöntemi yönelimlere karşı dağıtık rasgelelik sağlamada kullanılır. RandHound rastgele lider seçimiyle birleştirilir. İzleyiciler sayesinde parçaların tutarlılığı ve doğruluğu sağlanır. İşlemlerin paralel değerlendirilebilmesi için birbiri arasında bağlantı olmamalıdır. Bir hareket diğer hareketten etkileniyorsa paralel işlemede buna dikkat edilmelidir. Bu duruma çözüm olarak yönlü döngüsüz çizge yöntemi kullanılmıştır. Blok içindeki hareket başka bloktaki hareketi bekliyorsa bekleyen bloktan beklenilen bloğa doğru bir yönlü kenar çizilerek çizge oluşturulur. Bir hareket diğer bütün parçalara bağımlıysa sistemin performansı oldukça düşmektedir. Çoklu parçalı çalışmak yerine tek uzlaşma grubu olması daha etkin olmaktadır.

3.2.2. Elastico

Elastico [26] klasik Bizans protokolü ile Bitcoin protokolü arasında bir noktaya konuşlanır. Ana hatlarıyla Elastico ağı küçük komitelere (shard) böler ve her bir parçada farklı işlemlerle ele alınır. Her bir komite paralel olarak klasik Bizans fikir uzlaşmasını etkin çalıştıracak kadar az sayıda düğümden oluşur. Elasticoda bazı ön kabuller vardır:

- Dürüst düğümler doğrudan birbirine bağlıdır.

- Dürüst düğümler arasındaki iletişim kanalları senkronizedir.

Elastico geliştirilirken bazı zorluklarla karşılaşılmıştır:

1. İşlemcilerin dürüst olduğunu gösterecek herhangi bir kimlik veya açık anahtar alt yapısı (public key infrastructure-PKI) bulunmamaktadır. Bu nedenle işlemcilerin kendini tanıtabilmesi için bir yöntem sunulmalıdır.

2. f adet hatalı düğüm bulunduran bir ağda küçük grupların oluşturulması için yöntem belirlenmelidir. Her bir gruptaki dürüst düğümlerin çoğunlukta olduğu garantilenmelidir.

3. Düşman gözlemciler adaptif bir şekilde sistemi gözlemleyerek açık bir avantaj kazanmasını ve çoklu kimlikli düğümler oluşturması engellenmelidir. Fakat sistem yeni katılımcıları engellememelidir. Değişken oranda kimlik oluşturulmasına ve komite üyeleri arasındaki iletişime izin vermelidir.

Ağdaki düğümlerden farklı komiteler oluşturularak paralel hareket doğrulama yapılır. Bir adet son komite oluşturulur. Bu komite gruplardan gelen bilgileri birleştirmekten ve kriptografik özet oluşturup tüm ağa yaymaktan sorumludur. Komiteler belirli dönemlerder değişmektedir. Her bir düğümün diğerlerine doğrudan bilgisini göndermesi yerine farklı bir çözüm kullanılarak kimlik bilgileri ağa yayılır. Parçalardaki düğüm bilgilerini tutan dizin olarak görev yapan c sayıda düğümden oluşan özel bir komite vardır. Bütün düğümler dizin komitesiyle iletişime geçerek kendi komitesinde bulunan diğer düğümleri bulabilir. Elastico güvenli parçalama protokolüne sahip ilk açık blokzincir yapısıdır. Kısmi eşzamanlı ağda neredeyse lineer bir şekilde ölçeklenebilmektedir [26]. Ağdaki işlemci gücü arttıkça doğrulanan işlem sayısı lineer olarak artmaktadır. İşlemci gücünün 1/4 üne kadar hatalı davranışlar tolere edilebilir. Birbirine bağımlı hareketlerin paralel işlenmesiyle ilgili açık bir çözüm önerilmemiştir.

3.2.3. RsCoin

Diğer kripto paralardan farklı olarak RsCoin [30] para basma yetkisinin merkez bankasında olduğu bir sistemdir. Para basma yetkisi merkez bankasında olmasının yanında *mintette* olarak adlandırılan dağıtık yapıdaki bir sistem sayesinde sistemin tutarlılığı sağlanmış olunur. RsCoin kripto paraların klasik ölçeklenebilirlik problemine çözüm olarak düşünülmüştür. Bunun haricinde klasik kripto paralarda merkez bankaları paranın yönetimini tamamen kaybetmektedirler. RsCoin para oluşturma yetkisini merkez bankasına vererek bunun önüne geçmiştir. Kısacası para oluşturma ile defterin yönetimini birbirinden ayırmıştır. Hareketlerin doğruluğunun kontrolünü *mintette* olarak adlandırılan güvenilir düğümlere devretmiştir. Bu *mintette* bilindiği için herhangi bir yanlış davranış sonucunda cezalandırılabilir. Defterin yönetimi için iki aşamalı doğrulama (two phase commit-2PC) benzeri bir yöntem kullanılır. RsCoinin asıl amacı ölçeklenebilir bir blokzincir yapısı kurmanın yanında merkez bankasına para tedariği üzerinde kontrol yetkisi vermesidir. RsCoin de iki yapısal unsur yer almaktadır: merkez bankası ve *mintette*. Özetle *mintette* diğer kripto paralarda olduğu gibi işlemleri kullanıcılardan toplar. *Mintette*in işlemleri topladığı zaman dilimine *epoch* denir. *Mintette* oluşturduğu alt seviye blokları merkez bankasına göndereceği süreye de *period* denir. *Mintette*ler gruplanarak parçalar oluşturulur ve her *mintette* kendi parçasını bilir. Kullanıcı ilgili *mintette*'in kendi işlemini oylamasını ister. Giriş adresinin daha önce kullanılmadığına (paranın daha önce harcanıp harcanmadığı kontrolü) bakar. Kullanıcı giriş adresinin ait olduğu her *mintette*e hareket bilgisini gönderir. *Mintette*ler kontrolleri yaparak cevap olarak uygun dönerler ve kendi imzalarıyla imzalarlar. Kullanıcı gelen imzaların ilgili *mintette*e ait olup olmadığını kontrol eder. Çoğunluk sayıdaki *mintette* uygun cevabı dönerse kullanıcı çıkış adresini, ilgili hareketi ve kanıtları gönderir. Kanıtların içinde daha önceki uygun cevapları vardır. Bu bilgileri alan çıkış adresinin *mintette*i bilgilerin doğruluğunu kontrol ederler. *Mintette*lerin çoğunluğu işlemi onaylarsa işlem geçerli olmuş olur ve *mintette*lerin hareket kümelerine eklenir. Her periodun sonunda *mintette* hareket kümelerini merkez bankasına gönderirler. Merkez bankası bu bilgileri nihai blokta birleştirerek blokzincire ekler. *Mintette*ler kendi aralarında konuşmazlar, kullanıcıları kullanılarak iletişim sağlamış olurlar. Bu

sayede *Mintette*lerin birleşerek hatalı hareket üretmeleri engellenmiş olur. *Minttete*lerin güvenilir olduğu kabul edilir ve uzlaşma algoritması kullanılmaz. Bunun yerine 2PC benzeri bir yapıyla blokzincire blok eklenir. *Minttete*lerden gelen alt seviye bloklar merkez bankasında kontrol edilerek üst seviye blokta birleştirilir ve blokzincire eklenir.

3.2.4. Aspen ile Servis Odaklı Parçalama

Aspen çalışmada [44] blokzincir ağı farklı servislere bölünerek her servisin kendi içinde işlem yapması sağlanmıştır. Blok yayınlamada sadece servis parçalarındaki düğümlerin kendi içinde hemfikir olması beklenirken doğrulama noktalarında ağdaki bütün düğümlerin hem fikir olması beklenir. Bu çalışmayla blokzincir ağına aşağıdaki faydalar sağlanmıştır:

- Bütün blokzincirin güvenliğinin sağlanması için ağdaki bütün düğümlerin işlem gücünden faydalanılır.
- Bitcoindeki yapıyı kullanarak çift harcama problemini çözer.
- Madenci olmayan katılımcıları yani hizmet alan kullanıcıları ilgilendikleri hizmetlerin geçerliliğini doğrulamak için alakasız verileri saklama, işleme koyma ve yayma sorumluluğundan kurtararak ölçeklenebilirliği artırır.

Asıl problem blokzincirin bazı bölümlerinin bazı düğümlerden saklanmasına rağmen blokzincirin güven merkezi olmadan güvenli doğasını korumaktır. Bu çalışmada anahtar nokta hareketleri ait oldukları servislere göre bölmektir. Bir parçaya ait hareketin çıktısının başka bir parçada doğrudan kullanımına izin verilmez. Aynı başlangıç bloktan başlayan farklı blokzincirlerden ve ortak doğrulama noktasından oluşan bir yapı sunar. Hareketlerin ve yeni servis kaydı için iki farklı kanal tanımı vardır.

3.2.5. Chainspace: Parçalanmış Akıllı Sözleşmeler Platformu

Chainspace [45] akıllı sözleşmeleri de destekleyen parçalama özelliğine sahip bir blokzincir yapısıdır. *S-BAC* adını verdikleri dağıtık yazma protokolü sayesinde sistemin tutarlılığı sağlanmaktadır. Akıllı sözleşmelerin birlikte kullanımına da imkan vermektedir. Chainspace ağdaki düğüm sayısı arttıkça ölçeklenebilirliği artan, hatalara karşı dayanıklı ve tamamıyla açık ve denetlenebilir bir dağıtık defter yapısı sunar. *S-BAC* dağıtık atomik yazma protokolü sayesinde içlerinde hatalıların da olduğu düğümler arasındaki iletişimi başarılı bir şekilde koordine ederek sistemin güvenli, güvenilir ve sürekli olmasını sağlar. Chainspace'te nesneler durumları değişebilen temel yapılardır. Her nesneyi diğerlerinden ayırt edebilmek için biricik bir tanımlayıcı değeri ve akıllı sözleşme türünü de belirleyen bir türü vardır. Nesneler aktif ve aktif değil şeklinde iki temel durumda olabilirler. Aktif nesneler aktif hareketlerde kullanılabilirken aktif olmayanlar ise geçmişe dönük doğrulamalarda kullanılırlar. Ana tasarım amacı yüksek hareket çıktısı ve düşük ağ gecikmesine sahip ölçeklenebilirliğe erişmektir. Bunun için ağdaki düğümler nesnelerin durumlarını yönetmek için parçalara bölünürler. Her hareket kendi parçası içinde doğrulanır. Bir kullanıcı bir hareketi yazacağı zaman hareketin tutarlı ve doğruluğunun ispatlanabilir olması için yeterli bilgiyi sağlamak zorundadır. Hareket sadece ilgili düğümlere ve bu düğümlerin bulunduğu parçalara bildirilir. Bir hareket birden fazla parçayı ilgilendiriyorsa bütün parçalardan hareketin yazılması veya iptal edilmesi bilgisinin alınması gerekir. Parçalardan herhangi biri hareketi kabul etmezse bu hareket iptal edilir. *S-BAC* yöntemi iki aşamalı yazma yapısındadır. Ya hep ya hiç mantığına göre çalışır. Geleneksel optimistik eş zamanlılık kontrolüne benzemektedir. Bir hareket yazılacağı zaman sıralama ve doğrulama kuralları uygulanır. Bir hareketi ilgilendiren nesnelerin aktif olması ve diğer p da giriş nesnelerinin kullanılmamış olması gerekir. İki hareketin aynı aktif giriş nesnesini kullanması durumunda kurallar gereği bu hareketlerden sadece biri yazılır, diğeri ise işleme alınmaz. Şeffaflık ve denetlene bilirlik için her parçadaki her bir düğüm belirli periyotlarda imzalı özet zincirlerini kontrol noktası olarak yayınlar. Parçalar Merkle ağaçlarına bu turdaki bütün işlem bilgilerini ve diğer düğümlerden gelen imzaları içeren bir blok eklerler.

Bir parçadaki dürüst düğümlerin kontrol noktası için birbirinden bağımsız oluşturdukları

zincirler aynı olmak zorundadır. Parça içinde blok doğrulaması için BFT-Smart[22] kullanılırken parçalar arası işlemlerde iki aşamalı yazma [46] yapısı kullanılır. Denetlene bilirlik için bütün hareketler tekrar işletilerek sistemin doğruluğu kontrol edilebilir. Chainspace bazı işlemleri sonraki çalışma olarak sunmuştur. İlk sürümde parçaların oluşturulması sistem tanımında doğrudan tanımlanmıştır. Düğümlerin parçalara nasıl atanacağının belirlenmesi gerekmektedir. Parçalardan biri veya birkaçı bozulursa doğrulama noktaları kullanılarak geri dönüşün tam olarak nasıl yapılabileceği, çatallanmanın nasıl oluşacağı net değil. Aynı nesne üzerindeki işlem sayısı arttıkça iptal edilen hareket sayısı da artacaktır. Chainspacein parçalama mimarisi, hareketle ilgilenen düğümlerin hareketi işlemesini gerektirdiği için doğrusal olarak ölçeklenebilir olduğu söylenir.

3.2.6. PolyShard: Kodlu Parçalama Aynı Anda Doğrusal Ölçeklendirme Verimliliği ve Güvenliğini Sağlar

PolyShard [47] (polinomik kodlanmış parçalanma) etkin depolama, sistem çıktısı ve güven için teorik olarak üst limitlere ulaşarak gerçekten ölçeklenebilir bir sistem sunar. Ölçeklenebilirliği 3 temel bileşen özelinde inceler; çıktı, birim zamanda doğrulanan hareket sayısı; depolama etkinliği düğümlerin ele alabileceği maksimum blokzincir büyüklüğü ve güvenlik sistemin tolere edebileceği maksimum hatalı düğüm sayısı. Bütün bilgilerin bütün düğümler tarafından bilindiği sistemlerde (bitcoin, ethereum vs.) yüksek güvenlik sağlarken depolama ve çıktı bakımından yetersiz kalmaktadırlar. PolySharddaki temel fikir tek bir kodlanmış parçayı kaydedip işlemek yerine her bir düğüm iyi bilinen Lagrange polinomu kullanılarak aynı büyüklükte kodlanmış bir parçayı depolar ve işler. Bu kodlama, kötü amaçlı düğümlerin hatalı sonuçlarına karşı güvenlik sağlamak için hesaplama karmaşıklığı oluşturur. Kodlama genel olarak dağıtık işleme uygundur fakat aşağıdaki iki özellik PolyShardın blokzincir için özellikle kullanılabilir olduğunu gösterir.

1. Açıklık: Aynı kodlanmış veri birden çok doğrulama nesneleri için kullanılabilir (imza doğrulaması, bakiye doğrulaması gibi).

2. Birikimsel: Her bir düğüme önceki bloklara erişmeden en son doğrulanan bloğu kodlayarak kendi lokal kodlanmış parçayı büyütme imkanı verir. Bu sayede zincirin büyümesine rağmen sabit bir kodlama yükü sağlar.

Asıl anlatılmak isteneni daha net ifade edebilmek için sadece parça içi hareketlerin olduğu kabul edilmiştir. Parçalar arası hareketler için *OmniLedger*dakine [25] benzer bir yapının kullanılabileceği ifade edilmiştir. Her parça kendi alt zincirini yönetir. Kullanıcı atama algoritmasıyla kullanıcılar parçalara atanarak K adet parça oluşturulur. Parça içinde düğümler standart blokzincir çalıştırarak blokların yayınlaması yapılır.

3.2.7. Doğal Parçalama ile Değer Aktarımı için Ölçeklendirilmiş Bir Blokzincir

Doğal parçalama ile değer aktarımı için ölçeklenebilir blokzincir çalışmasında [48] değer transfer defteri (Value Transfer Ledger-VTL) modeli oluşturulmuştur. Hareketin göndereni alıcıya hareketin doğruluğunu ispatlamak zorundadır. Alıcılarda hareketi alacağı zaman hareketin doğru ve güvenilirliğini kontrol etmek zorundadır. Çift harcama problemini önlemek için bütün hareket kümesinin bilinmesine gerek olmadığı gösterilmiştir. Bu tasarım faydalı sonuçlarından biri de düğümler hareketin çift harcama olmadığını ispatlayan hareketleri göstererek kendiliğinden iletişim maliyetlerini düşürmeye çalışırlar. Bunun sonucunda düğümler hareketlerin bir bölümlerini tuttuğu için ağ parçalara ayrılmış olur. Bu yapıya kendiliğinden parçalama denir. Her düğümün sadece kendini ilgilendiren hareketlerini tuttuğu kişisel zincirleri ve bu zincirlerin özetlerinden oluşan genel uzlaşmada kullanılmak üzere ana zincirden oluşan bir yapıya sahiptir. Bunun yanında yerel zincirlerde de çalışan bir doğrulama fonksiyonu da oluşturulmuştur. Bir işlemin geçerliliğini bilmek isteyen tüm düğümlerin, o işlemin geçerliliğini etkileyen tüm işlemler üzerinde tutarlı bir gözlemi olacağı garanti edilmiştir. Her düğümün başlangıçta bir miktar değeri olduğu kabul edilir, sonradan yeni değer oluşturma dikkate alınmaz. Bir hareketin bir göndericisi ve bir alıcısı olduğu kabul edilir. Blokzincirin izinli blokzincir olduğu kabul edilir. Bitcoindekine benzer bir harcanmamış işlem çıktısı(UTXO) mantığı kullanılır. Harcanmamış bir hareketin alıcısı o değerin sahibidir. Düğümler doğal olarak en az ispat maliyeti gerektiren harcanmamış değeri seçme eğiliminde

olacaklardır. Bunun sonucu olarak düğümler değeri bütün ağda dolaştırmak yerine daha küçük parçalarda dolaştırmayı tercih edecektir. Diğer bir ifadeyle güvenlikten ve dağıtıklıktan ödün vermeden ağ parçalara ayrılmış olur. Birim saniyede doğrulanan hareket sayısı O(N) üst limitindedir. Değerin sahibi istenildiği zaman değerin var olduğunu ve ona ait olduğunu ispat etmekle yükümlüdür. Diğer bir ifadeyle düğümler hareket kayıtları yerine sahip oldukları değerin varlığıyla ilgilenirler. Bir düğüm hareketleri sadece aşağıdaki durumlar için önemser:

1. Bir düğüm kendini ilgilendiren bir hareketi ispat için başka bir harekete ihtiyaç duyuyorsa o hareketi önemser.
2. Bir düğüm bir hareketin sadece ve sadece alıcıysa ve o hareketin doğruluğunu bilmiyorsa o hareketi önemser.

Sistem tasarımında sistem 3 parçaya bölünmüştür. Genel durum paylaşımı için ana zincir, her düğümün kendi için tuttuğu bireysel zincirler ve hareketlerin doğrulanması için doğrulama şeması. Bu yapıda diğer blokzincirlerden farklı olarak gönderici hareketin doğruluğunu ispatlayan bilgileri düğümlere vermeyi reddederse diğer düğümler hareketin doğruluğunu kanıtlayamaz.

3.2.8. Tüm Boyutların Parçalandığı Blokzincir(MultiVAC)

Yazarlarının ifadesine göre MultiVAC[49] ilk hızlı, etkin ve bütün boyutlarıyla shard edilmiş tamamen ölçeklenebilir blokzincirdir. Shardingi sadece çıktı olarak değil iletişim ve depolama için de gerçekleştirmiştir. Çoklu kimlik[3] saldırıları önlemek için PoS kullanılırken, doğrulanabilir rastgele fonksiyonlar ağı parça olarak adlandırılan bölümlere ayırmak için kullanılır. MultiVAC, depolama ve iletişimi parçalara arasında bölen sağlam bir mimari üreterek blokzincir için zarif bir dağıtılmış depolama çözümü sunar. Bu şekilde, blokzincirin merkeziyetçiliğe ihtiyaç duymaması, eşitlik ve güvenlik gibi temel değerlerini korurken, gerçek ekonomiye hizmet edebilecek bir çıktı üretimi sağlar. Bazı parçalama yöntemlerinde

sadece çıktı ölçeğinde iyileştirme sağlanırken, ağın depolama ve iletişim yükü üzerinde iyileştirme sağlamaz. Yine düğümler bütün blokzincir bilgisini kendinde tutmak zorundadır. Bu parçalama yöntemi diğer parçalama yöntemlerinden farklıdır ve diğer yöntemlerin eksik yanlarını geliştirmiştir. Sadece işlem için parçalama değil veri saklama ve iletişim için de parçalama olarak tasarlanan ilk blokzincirdir. MultiVAC temel teknolojik avantajları aşağıdaki gibi listenebilir.

1. Sadece hareketleri bölmek blokzincir ölçeklenebilirliği için yeterli değildir. Ağ iletişimi ve veri saklamanın da bölünmesi gerçekleştirilmiştir.

2. Eşitlik, güvenilirlik ve güvenlik ve blokzincir sistemleri için çok önemlidir. Her parçanın güvenilirliğini ve güvenliğini sağlayan doğrulanabilir rastgele işlevlere (VRF) dayalı adil bir yeniden parçalama oluşturma yöntemi kullanılmıştır.

3. Parçalar arası iletişim her parça kullanan sistem için zor bir problem olmuştur. UTXO ve Merkle kök veri yapıları kullanılarak bu problemin çözülmüştür.

3 tür düğüm vardır. Hafif düğümler: Sadece hareket oluşturan düğümler, normal sıradan kullanıcılar Madenci Düğümler: Uzlaşma algoritması çalıştıran ve değişik parçalara atanan düğümlerdir. Sistemin muhasebecisi gibi çalışırlar. Bütün hareket dökümünü bilmelerine gerek yoktur. Depocu Düğümler: Parçalara atanarak bilgilerin saklanmasından ve hareketlerin madenci düğümlere sunulmasından sorumludurlar. Madencilerin parçalara atanması için VRF kullanılır. VRF güvenilir olmayan bir ağda bir düğüm için rastgele bir sayı üreteci olarak işlev gören ve aynı zamanda diğer düğümlerin oluşturulan sayının meşru olarak rastgele olduğunu ve hiçbir şekilde manipüle edilmediğini doğrulamasına izin veren bir kanıt üreten fonksiyondur. Bütün düğümler yarı senkronize bir ağ üzerinden bulunurlar öyle ki bütün madenciler çok kısa bir süre içinde birbirleriyle konuşabilirler. Ağa madenci olarak katılmak isteyen düğümler bir miktar depozit bırakmak zorundadırlar. Depozit bırakan düğümler yeniden parçalama adımına kadar beklerler. Her parça ayrı çalıştığı için bir madenci birden fazla parçaya atanabilir. Madenci kendi özel anahtarı ve parçanın VRF değerini VRF fonksiyonuna sokarak parçaya atanıp atanmadığı bilgisine ulaşır. Parçadaki düğüm sayısı

çok artarsa veya belirli bir süre geçerse parça yeniden yapılandırılması devreye girer. Parça bölünmesinde depocu düğümler bir süre yeni ve eski parçalara aynı anda hizmet verir. Her kullanıcı kendi parçası içinde hareket oluşturabilir. Çıktılar ve durumları Merkle ağacında tutulur. Yeni bir hareket oluşturulacağı zaman depocu düğümlere gönderilir. Depocu düğümler parçalardaki bütün madenci düğümlere bildirir. Blok üzerinde uzlaşma sağlanırsa blok başlığı bütün parçalardaki madencilere, blok ise bütün depocu düğümlere iletilir. Madenciler, bütün blok başlıklarını, her parçanın Merkle ağaçlarının kök bilgisini ve Merkle yollarını tutmak zorundadır. Madenciler bütün bloğu tutmak yerine sadece Merkle kök tutması yeterlidir. Depocu düğümler için herhangi bir ödülden bahsedilmemiştir. Bir kazanç yoksa neden bu işi yapsınlar sorusu açıktır.

3.2.9. Blokzincir Sistemlerini Parçalama Yoluyla Ölçeklemeye Doğru

Güvenli izinli blokzincir ağını parçalara ayıran bu çalışmada [50] parçaların çalıştığı güvenli bir donanımsal ağ bulunur. Intel SGX(Security Guard Extensions) ortamı kullanılır. Her tur başında her bir düğüm *e* tur numarasıyla rastgele değer üreten özel fonksiyonunu (randomness beacon enclave) çağırır. Bu özel üreteç için *sgx-read-rand* fonksiyonunu iki kez çağırarak rastgele *q* ve *rnd* değeri üretir. *q=0* olan durumda *e* ve *rnd* değerlerini içeren imzalı sertifika oluşturarak bütün ağa yayar. Belirli bir zaman sonunda düğümler kendilerine iletilen en düşük *rnd* değerini kabul ederler ve komite dağılımında kullanırlar. *rnd* değeri elde edildikten sonra, düğümler komite atamaları için *rnd* ile başlatılmış rastgele permutasyon [1:N]'e kadar oluşan π kümesini oluştururlar. Daha sonra π kümesi parça sayısı kadar eşit parçalara bölünür. Bu sayede parçadaki düğümler bulunmuş olunur.

Bu donanımsal özel üreteçler sayesinde bir *rnd* değeri oluşturulduktan sonra aynı tur için farklı bir değer üretilmesine izin verilmez.

3.2.10. Harmony

Harmony[51] literatürde var olan blokzincir çalışmalarını birleştirerek yeni, tam ölçeklenebilir bir ürün ortaya çıkarmıştır. Aşağıdaki özellikleriyle diğer blokzincirlerden ayrılır.

- Blokzincir iletişimi ve blokzincirdeki işlemleri parçalamasının yanında blokzincir durumunu da parçalara ayırır.
- Dağıtık rastgelelik sayesinde parçalama işlemi güvenlidir.
- PBFT'den 100 kat daha hızlı ve etkin fikir birliği algoritmasına sahiptir. Fikir birliğine katılacak düğümlerin seçiminde PoW yerine PoS kullanır.
- Parçaların birbiriyle iletişimde olmasına izin vererek parçalar arası transfer yapılabilir.

Doğrulanabilir geciktirilmiş rasgelelik (Verifiable Delayed Function-VDF) kullanılarak düğümlerin parçalara ve komiteye atanması sağlanır. Lider düğüm son bloktan önceki bloğun kriptografik özet değerini içeren başlangıç mesajını bütün doğrulayıcılara gönderir. Bu mesajı alan her doğrulayıcı doğrulanabilir rasgele fonksiyon (Verifiable Random Function-VRF) ile $r_i, p_i = VRF(sk_i, H(B_{n-1}), v)$ oluşturur ve r_i, p_i değerlerini lidere döndürür. r_i rasgele değeri, p_i ise ispat bilgisini içerir. Lider en az $f + 1$ cevap gelene kadar bekler ve sonrasında rasgele değerleri ***XOR*** fonksiyonundan geçirerek ***pRnd*** değerini elde eder. Lider ***pRnd*** değerini son bloğa yerleştirerek BFT çalıştırır. ***pRnd*** değeri yazıldıktan sonra lider gerçek rasgele değeri elde etmek için $Rnd = VDF(pRnd, T)$ fonksiyonunu kullanır. ***T*** zorluk değişkenidir. ***Rnd*** elde edildikten sonra lider ***Rnd*** değerinin kabulü için BFT başlatır. VDF kullanılarak hatalı liderin kendi menfaatine göre rasgele değerler elde etmesi engellenir. Düğümleri parçalara ayırmak yerine oy güçlerini parçalara ayırır. Oy gücü düğümlere uzlaşmada oy verme hakkı tanıyan sanal bilet olarak düşünülebilir. Oy gücü için ihtiyaç duyulan hisse miktarı algoritmik olarak ayarlanır. ***Rnd*** ile başlatılmış rasgele permutasyon bütün oy güçleri için oluşturulur ve bu permutasyon *m* parçaya ayrılır. $i.$ parçadaki oy gücü $i.$ parçaya atanır. Dolayısıyla ilgili oy gücüne sahip düğümlerde o parçaya atanmış olur. Bu durumda bir doğrulayıcı birden fazla parçaya atanabilir.

3.2.11. RapidChain

RapidChain [52] her bir katılımcının doğrulanabilir gizli paylaşım (Verifiable Secret Sharing-VSS) değeri üretmesine izin verir. Bu üretilen değerleri birleştirerek rastgele başlangıç değerini elde eder. Fakat bu yöntem güvenilir değildir. Çünkü hatalı düğümler her düğüme farklı değer gönderebilir.

3.2.12. Algorand

Algorand [29] VRF tabanlı kriptografik *sortition* yöntemi kullanarak parçaları oluşturur. *Sortition* yönteminde hisse değerleri dikkate alınarak parçalara atama yapılır. Bir düğüm birden fazla parçaya atanabilir. Harmony Algoranda benzer şekilde atama yapar.

3.2.13. Zilliqa

Zilliqa çalışmasında [53] dizin servisi (Directory Service(DS)) olarak görev yapan düğümler bulunmaktadır. Parçaların oluşturulması için öncelikle DS komitesinin oluşturulması gerekmektedir. DS'e katılmak isteyen düğümler DS başlangıç değerini kullanarak PoW(Ethash-PoW) çözümü oluşturur. Çözümü DS komitesine gönderir. Eğer çözüm doğruysa komiteden en eski düğüm çıkarılır. Yeni düğüm komiteye dahil edilir ve lider olur. Fikir birliği için PBFT'nin Schnorr dijital imzalama [54] uygulanmış hali kullanılmaktadır. DS komite oluşturulduktan sonra ağa katılmak isteyen düğümler parça başlangıç değerini kullanarak PoW çözümü üretir ve DS komiteye gönderir. DS komite *nonce* değerine göre gelen çözümleri sıralar. Parçadaki düğüm sayısı n_0 ise ilk n_0 düğüm ilk parçaya sonraki n_0 düğüm sonraki parçaya atanır. Parça içinde en büyük *nonce* değerini ileten düğüm parça lideri olur. Herhangi bir rasgelelik kullanılmaz. Başlangıç tohum değeri 32 byte 0'ın SHA3-256 kriptografik özet değeridir. Sonraki turlarda ise tohum değeri bir önceki tohum değerinin SHA3-256 kriptografik özet değeridir.

3.3 GURU: Dağıtılmış Uzlaşma Protokolü için Evrensel İtibar Modeli

GURU çalışmasında [40] saygınlık modellerinin dağıtık uzlaşma yöntemiyle birleştirile bilirliği üzerine yapılmıştır. Güvenilirliği bilinmeyen doğrulayıcılardan oluşan büyük noktadan noktaya ağlarda ölçeklenebilir ve yüksek çıktı sayısı sağlayabilmek amaçlanmıştır. GURU modeli PBFT veya HoneyBadger [21] gibi uzlaşma yöntemlerinin üzerine kurulabilir. Uzlaşma turlarının sonucuna göre ağdaki düğümlere puan verir ve bu puanlara göre fikir birliği grubunu adaptif bir şekilde oluşturur. Ayrıca dışardan puan girilmesine de imkan verir. BFT yöntemlerinde ağdaki düğüm sayısının 1/3'üne kadar hatalı düğümlere dayanıklıyken GURU sayesinde bu oran 1/2'nin biraz üstüne kadar çıkabilmektedir. Oylama turu başarılı olursa GURU komite üyelerinin saygınlığını artırırken başarısız olursa saygınlığını azaltır. Komite üyelerini rastgele olarak seçer ve affedilebilir olmayı da sağlar. Yani hatalı davranıp sonra düzgün davranan veya tam tersi durumları da düşünerek düğümlerin komiteye katılımına imkan verir. Ayrıca hatalı düğümlerin çoğunluğu ele geçirene kadar düzgün davranıp sonrasında hatalı davranması sorununu da ele alır. Bu gibi durumlarda çok hızlı bir şekilde kurtarma yöntemini çalıştırır. Ağdaki düğümün cevabı belirli bir t süresinden uzunsa bu düğüm zararlı olarak değerlendirilir. En az ve en fazla zararlı olma ihtimali kullanılarak her bir düğüm için saygınlık hesaplanır. Hesaplanan saygınlık değerleri kullanılarak komite oluşturulur. Komite oluşturulurken saygınlık değerleri sıralanır. Daha sonra düğüm sayısı kadar rastgele sayılar oluşturulur. Bu rastgele sayılara karşılık gelen düğümler komiteye seçilir. İki farklı seçilme dağılım yöntemi kullanılır: üssel ve üçgensel. Üssel yöntem yüksek saygınlık değerlerine sahip düğümlerin seçilmesini sağlarken üçgensel dağılım daha adil davranır. Düşük saygınlığa sahip düğümlerin de komiteye seçilmesine imkan verir. Her bir döngüde ceza ve ödül hesaplanır. Başarısız olan bir döngüdeki yüksek saygınlığa sahip düğümlerin cezası yüksek olur. Başarılı olan bir döngüde düşük saygınlığa sahip düğümlerin ödülü yüksek olur. Başarı oranı değişkeni sisteme eklenerek sistemin izlenirliği sağlanır. Bu oranın hızlı düşmesi sisteme saldırı olduğu anlamına gelir.

4. FİKİR BİRLİĞİ KOMİTESİ OLUŞTURMA MODELİ

Bu bölümde önerilen modele neden ihtiyaç duyulduğu, modelin detaylı açıklamaları ve deneysel çalışmaların sonuçları anlatılmaktadır.

4.1 Arkaplan

Blokzincire ilginin artmasıyla birlikte kullanımı yaygınlaşmaya başlamıştır. Kullanımının artmasıyla birlikte bazı sorunlarının olduğu ortaya çıkmaya başlamıştır. Bu sorunların başında ölçeklenebilirlik problemi gelmektedir. Ölçeklenebilirlik, bir sistemin potansiyelini artan iş yükü ile baş edebilecek şekilde geliştirebilme kabiliyeti olarak tanımlanabilir. Blokzincir ölçeklenebilirlik probleminin arkasında yatan temel sebep uzlaşma algoritması ve blok büyüklüğüdür [6, 55]. Ölçeklenebilirlik de asıl amaç çıktı miktarını artırırken bloğun geçerlilik kazanma süresinin kısaltılmasıdır. Blokzincir özelinde çıktı bloklar dolayısıyla blok içinde yer alan hareketler olmaktadır. Bu problemin çözümü için yapılan ilk çalışmalarda blok büyüklüğü artırılmıştır [56–58]. Fakat blok büyüklüğünün artırılması belirli bir oranda fayda sağlamıştır. Kullanıcı sayısı arttıkça belirli oranda çıktının da oranlanabilmesi sağlanamamıştır. Bunun yanında blok büyüklüğünün artması bloğun ağda dolaşımını yavaşlatmıştır [59]. Sonraki çalışmalarda blokzincirle birlikte çalışan yan zincirler veya özel zincirlerin [60–63] kullanılması düşünülmüştür. Bu tür farklı zincirlerin kullanıldığı durumlarda güvenlik sorunu ortaya çıkmaktadır [60, 64].

Blokzincirin diğer ödeme kanallarıyla (Paypal, Visa, MasterCard) yarışacak hale gelebilmesi ve istenilen kullanım yaygınlığına kavuşabilmesi için birim zamanda gerçekleşen işlem sayısının çok daha fazla artırılmasına ihtiyaç duyulmaktadır. Ölçeklenebilirliğin sağlanabilmesi için genel kanı uzlaşma algoritmasının değiştirilmesi ve paralel hareket işleme kabiliyetinin eklenmesi gerektiği yönündedir.

Uzlaşma algoritması olarak en hızlı çözüm sunan ve çok fazla işlemci gücü gerektirmeyen

yöntem BFT tabanlı yöntemlerdir. BFT yönteminde blok üzerinde uzlaşma sağlandığı anda blok geçerlilik kazanır. Bitcoinde olduğu gibi belirli bir derinliğin oluşmasını beklemeye gerek yoktur. Yine genel olarak PBFT [20] yöntemi ve bu yöntemden türetilen yöntemler en çok tercih edilen yöntemlerdir. BFT fikir birliği protokolünü açık blokzincirlerde kullanabilmek için bazı sorunların çözülmesi gerekmektedir. BFT fikir birliği yapısı gereği ağdaki bütün düğümlerin bilinmesi gerekmektedir [33] ve ağdaki düğüm sayısı belirli bir sayıyı geçince performans problemleri yaşanmaktadır [65]. Açık blokzincirlerde BFT uygulayabilmek için bu sorunlara çözüm bulunması gerekir. Literatürdeki çalışmaların çoğu bütün üyelerin bir alt kümesini kullanarak bir komite oluştururlar. Komite oluşturulurken çoklu kimlik oluşturma saldırılarına karşı dayanıklı olması gerekmektedir. Bu nedenle [24–28, 53] çalışmalarında olduğu gibi komiteye katılmak isteyen kullanıcılardan PoW çözmesi beklenir. PoW çözen kullanıcılardan komite oluşturularak blok doğrulaması gerçekleştirilir. Bu komite içinden BFT'yi yönetecek lider sırasıyla dönüşümlü yapıda seçilir. Lider, blok doğrulama aşamalarını yönetir. BFT tabanlı uzlaşma yöntemi kullanan özel blokzincirlerde ağdaki kullanıcı sayısı sınırlı ve bilindiği için kullanımda sorun yaşanmaz [30–32]. Hatalı davranışlarda bulunan kullanıcı kimliği belli olduğu için bu kullanıcının cezalandırılması caydırıcı bir mekanizma olmaktadır.

BFT temelli fikir birliği yöntemlerinde, komitedeki düğümlerin 1/3'ünden fazlası hatalı ise fikir birliğine varmak mümkün değildir. Blok yayma ve doğrulama gibi blokzincir işlemlerinin engellenmemesi için açık blokzincir ağlarında 1/3 oranının arttırılması çok önemlidir. Bu nedenle komiteye katılmak isteyen düğümlerin güven değerinin yüksek olması önem arz etmektedir. Hesaplanan güvenilirlik değerine göre düğümlerin uzlaşma grubuna katılımı sağlanacaktır. Bu sayede hatalı düğümlerin uzlaşma komitesine katılımı azaltılmaya çalışılacaktır ve sistemin güvenilirliği artırılacaktır. P2P ağlarda düğüm güvenilirlik bulma çalışmaları [34–39] yaygın bir şekilde çalışılmasına rağmen blokzincir dünyasında bu tür çalışmalara az rastlanılmaktadır [40].

Fikir birliği komitesi oluşturulmasında daha akıllı öğrenmeye dayalı adaptif yöntemlerin kullanılması düğümlerin davranışlarının takip edilmesi anlamına gelir. Davranışların takip

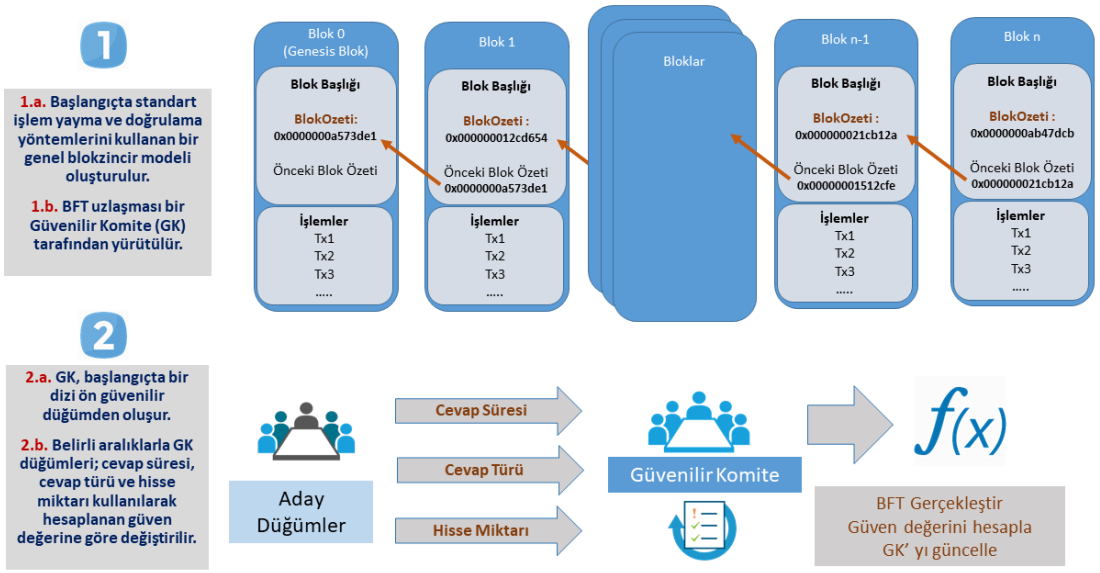
edilmesiyle düğümler doğru davranmaya teşvik edilmiş olunur. Bu çalışmada fikir birliği komitesi oluşturulmasında adaptif çevrimiçi karar tabanlı bir öğrenme yöntemi [1] kullanılacaktır. Bu algoritmanın çıkış gerekçesine örnek olarak şu durum verilebilir: At yarışlarında sürekli kaybeden ve arkadaşlarının kazançlarını kıskanan kumarbaz, bir grup kumarbazın kendi adına bahis oynamasına karar verir. Her yarışa yatırdığı para miktarı sabit olacaktır; fakat parasını arkadaşlarına paylaştırırken arkadaşlarının başarı durumuna göre dağıtır. Eğer belirli bir süre içinde arkadaşlarından hangisinin en çok kazanacağını bilseydi bütün bahislerini o arkadaşının yönetmesine izin verecekti. Bu geleceği görme durumu mümkün olmadığı için, en çok kazanan arkadaşının bahislerini yönettiğinde kazanacağı miktara yakın para kazanmaya çalışır. Hedge [66] yöntemi bu tür dinamik karar verme aşamaları için önerilmiş bir yöntemdir. Bu yöntemin amacı en iyi stratejiye göre birikimsel kaybı minimize etmektir. Her stratejinin belirli bir ağırlığı vardır ve bu ağırlık her turda bir miktar kayba (loss) uğrar. Hedge yöntemini kullanabilmek için parametrelerin ayarlanması gerekmektedir. Bu parametrelerin ayarlanması her problem için kolay olmamaktadır. Bu nedenle [67] bu çalışmada parametresiz yeni bir hedge yöntemi olan NormalHedge önerilmiştir. NormalHedge uygulaması çok basit ve uygulanması kolay bir yöntemdir. Her turda tek bir satır araması ve bunu takiben tüm işlemler için ağırlıkların güncellenmesini içermektedir. Öğrenici her harekete belirli bir ağırlık değeri atar. Atamalar sonrasındaki turlarda her hareketin ağırlığı önceki turda atanan ağırlıklar kullanılarak hesaplanan beklenen değer kadar kayba uğrar. Bir hareket için pişmanlık (regret) öğrenicinin birikimsel kaybı ile ilgili hareketin birikimsel kaybı arasındaki farktır. Öğrenicinin amacı düşük pişmanlığa ulaşmaktır. Ağırlıkların en başından beri tutularak hesaplanması öğreniciyi yanıltabilir. Değişikliklerin ağırlığa daha hızlı yansıyabilmesi için adaptif hedge yöntemi [1] önerilmiştir. Bu yöntemde ağırlıkların hesaplanmasında kullanılan birikimsel pişmanlık hesaplanırken bütün geçmişe değil de belirli sayıdaki son turdaki değerlere bakılır.

Önerilen modelde adaptif hedge algoritmasıyla komite seçimi yapılacaktır. Bu model PoW yöntemine kıyasla çok daha az işlemci gücü gerekmektedir ve ağdaki kullanıcıların değişen davranışlarına kolayca ve kısa sürede uyum sağlayabilmektedir.

4.2 Model Genel Yapısı ve Adaptif Hedge Algoritması Temel İlkeleri

Önerilen modelde blok doğrulama ve yayınlama gibi klasik blokzincir işlemleri diğer blokzincir sistemlerinde olduğu gibi yapılır. Şekil 4.1'de anlatıldığı şekliyle bu işlemlere ek olarak komite seçimi için önerdiğimiz modelin işlemleri gerçekleştirilir. Önerdiğimiz bu model BFT fikir birliği yöntemi kullanan herhangi bir fikir birliği algoritmasına uygulanabilirdir. Sistemin başlangıcında ön güvenilir düğümlerden oluşan güvenilir komite (GK) oluşturulmuştur. Sistemin başlangıcında bu düğümler GK'ya katılmak isteyen düğümler için güven değeri hesaplayarak GK'nın yeniden şekillenmesini sağlayacaktır. Düğüm değiştirme tur sayısına ulaşıldığında düğümler değiştirilir. Sistemin merkezileşmemesi için GK'da ön güvenilir düğüm varsa ilk önce bu düğümler çıkarılır. GK tamamıyla yeni seçilen düğümlerden oluştuğunda ise seçim yöntemlerinden biri veya bir kaçı uygulanarak çıkarılacak düğümler seçilerek çıkarılır. GK lideri güven değeri en yüksek üyedir (lider sırayla da olunabilir). Son p tur için güven değeri hesaplaması yapılır. Bu sayede değişen özelliklere daha kısa sürede uyum sağlanır. Güven değeri hesaplanırken blokzincirde fikir birliğinin doğru oluşması için önemli olduğunu düşündüğümüz cevap süresi, verilen cevap türü ve sistemde sahip olunan hisse miktarı gibi özelliklerden oluşan özellik vektörü kullanılır. Bu özellikler dışında yeni özellikler eklenebilir veya özellik vektöründen özellikler çıkarılabilir. Bir düğümün GK'da sürekli kalmasını önlemek için düğümlerin GK'da bulunma sayısı özellik vektörüne güven değerini negatif etkileyecek şekilde eklenebilir.

Süreç şu şekilde ilerler: Başlangıçta GK bu komiteye girmek isteyen kullanıcılar için katılım mesajını yayar. Komiteye katılmak isteyen düğümler açık anahtarlarını içeren cevap bilgisini GK'ya döner. GK üyesi her düğüm adayları izlemeye alır. Her blok yayınlamanın sonunda adaylar ve GK'daki düğümler için *hedge* yöntemiyle güven hesaplaması yapar. Belirli bir blok sayısı sonucunda GK'daki düğümler kendi arasında aday bilgilerini paylaşırlar. Bir düğüm için hesaplanan güven değeri farklıysa küçük olan değer kullanılır. Eşik değerinden büyük güven değerine sahip adaylardan belirli bir sayıda rastgele seçilir. Seçilen sayı kadar da GK'dan çıkacak düğümler seçilir ve GK güncellenir. Ağa katılacak ve ağdan çıkacak düğümlerin seçimini GK lideri yapar. GK'daki düğümler güven bilgilerini paylaştıkları



Şekil 4.1: Sistemin çalışma adımları

için kimlerin seçileceğini bilirler. Eğer lider yanlış düğümleri seçerse lider değişikliği mesajı yayılır. GK'daki düğümlerin çoğundan lider değişikliği mesajını alan düğümler GK'daki güven değeri liderden sonra en yüksek olan düğümü lider olarak kabul eder.

Sisteme katılımı, doğru davranışı teşvik etmek ve hatalı davranışların önüne geçmek için sisteme ödül ve ceza eklenebilir. Bunun için sisteme katılmak isteyen düğümlerden bir miktar hissesini depozito olarak özel hesaba aktarması istenebilir. Düğüm dürüst davrandıkça GK'ya girme olasılığı artar. GK'daki düğümlerde her blok yayınlandıkça bir miktar ödül kazanır. Aday düğümler veya GK'daki düğümler hatalı davranmaya başladığında güven değeri belirli bir seviyenin altına indiğinde ceza olarak depozitosuna el konulabilir. Bu sayede dürüst davranışlar ödüllendirilir ve hatalı davranışlar cezalandırılır. Düğümler dürüst davranmaya teşvik edilir. Bu çalışmada asıl amacımız güven değeri kullanarak GK'ya hatalı düğüm seçme olasılığını azaltmak olduğu için ceza ve ödül sistemini uygulamadık.

4.3 Özellik Seçimi

Öğrenme modelinin doğru çalışabilmesi için kullanılacak özellikler doğru seçilmelidir. BFT tabanlı sistemleri incelediğimizde bloğun geçerlilik kazanmasının ağdaki düğümler arasında iletişimin sağlanması ve oyların toplanması sonucunda olduğunu görebiliriz. Bu nedenle seçilecek özelliklerin BFT tabanlı sistemlerin işleyişini doğrudan etkilemesi büyük önem arz etmektedir. Blok yayınlanmasının yanı sıra sistemin güvenliğini koruyacak özelliklerinden de kullanılması gerekir. Saldırganların işini zorlaştıran ve sistemi ele geçirmesini engelleyen özellikler eklenerek sistemin devamlılığı sağlanır.

Fikir birliği için BFT tabanlı bir yöntem kullanılacaktır. Bu yöntemlerde, bloğun geçerlilik kazanıp yayınlanması için teklifin onaylanması, diğer bir ifadeyle $2f + 1$ düğüm tarafından kabul edilmesini gerektirir (f = hatalı düğüm sayısı). Verilen cevap türü blokların geçerlilik kazanmasını sağladığı için sistemin başarısı üzerinde büyük etkiye sahiptir.

Blok yayınlama süresinin artması sistemin performansını doğrudan etkileyeceği için düğümün cevap süresi önemli bir özelliktir. Modelimiz, diğer özellikler için yüksek değerlerle birlikte düşük yanıt süresine sahip düğümlere yüksek güven değeri atayacak şekilde tasarlanmıştır. Konsensüs komitesi aktif cevap veren düğümlerden oluştuğundan, mesaj değişiminin de hızlı bir şekilde gerçekleşmesi beklenmektedir.

Düğümün sistemdeki payı ne kadar çoksa o düğüm sistemin o kadar sağlam kalmasını ister. Bu nedenle düğümlerin sistemde sahip olduğu pay da önemli bir özelliktir. Sistemdeki paylara dayalı çalışan uzlaşma yöntemi (Proof of Stake) PeerCoin[15] gibi alternatif kripto paralarda blok doğrulamada kullanılan bir yöntemdir. Ayrıca çoklu kimlik saldırılarına karşı sistemi güvenli hale getirmek için hisse miktarı kullanımı bilinen bir savunma yöntemidir.

Bu sebeplerden cevap süresi, hisse değeri ve cevap türü bilgilerini öğrenmede kullanılacak özellikler olarak seçtim. Öğrenme modelimde kullandığım özellikler ölçeklenebilirliği ve sistemin sürekliliğini doğrudan etkileyen özelliklerdir. Model tasarımı farklı özelliklerin

eklenmesi kolay olacak şekilde yapılmıştır. Yeni bir özellik eklenmek istendiğinde özellik değerinin vektöre eklenmesi yeterli olacaktır. Yeni eklenen özellikler güven değerinin hesaplanmasında kullanılacaktır. Farklı özellikler seçilirken sistemi doğrudan etkilemesi önem arz etmektedir. Sistem başarısı üzerinde az etkisi olan özelliklerin eklenmesi güven değerlerinin bulunmasında yanıltıcı etki yapabilir.

Sistemi ele geçirmeyi daha zor hale getirmek için kriptografik özet gücü de yeni bir özellik olarak eklenebilir. Bu sayede Bitcoin ve Ethereum gibi büyük kripto paraların güvenlikleri birleştirilmiş olunur. Ayrıca dinamik özellik çıkarımı kullanımının sistemin başarısı üzerinde etkisinin incelenmesi de ileriki çalışma olarak not edildi.

4.4 Adaptif Hedge Algoritmasının Uyarlanması

Adaptif hedge yöntemini (AHY) uzlaşma grubuna katılacak düğümlerin belirlenmesinde kullanabiliriz. AHY [1]'de kullanılan formülleri(2, 3, 4, ...) blokzincire uyarlayarak her düğüm için bir güven değeri hesaplayacağız ve bu güven değerini uzlaşma grubuna seçim yaparken kullanacağız. Bu sayede uzlaşma grubundaki düğümlerin zararlı olma ihtimalini minimize etmiş oluruz. AHY uygulayabilmek için öncelikle öğrenmede kullanacağımız özellik değerlerini içeren bir özellik vektörü tanımlamamız gerekir. Bu özellik vektörü cevap süresi, cevap türü, düğümün sistemdeki hisse miktarı gibi özellikleri içerecektir. Güvenilir komiteye katılmak isteyen ve komitedeki her düğüm için bir güven değeri hesaplanır. Güven değerini hesaplayabilmek için bir kayıp fonksiyonu oluşturmamız gerekir. Bu kayıp değeri l_t^k (2) formülü ile ifade edilir.

$$l_t^k = \max\left(S_t^k\right) - S_t^k \tag{2}$$

Kayıp için her bir özelliğin güvenilir komitedeki maksimum değeri ilgili düğüme ait özellik değeri arasındaki fark diyebiliriz. S özellik vektörünü, t blok indeksini gösterirken k özellik indeksini gösterir. Pişmanlık değeri r_t^k (3) formülü ile hesaplanır.

$$r_t^k = \overline{l}_t^k - l_t^k \tag{3}$$

$\overline{l}_t^k$ bütün özellikler üzerinde ağırlıklı ortalama kayıptır ve $\overline{l}_t^k = \sum_{k=1}^{K} w_t^k l_t^k$ şeklide hesaplanır. w ağırlık değerini, K özellik sayısını gösterir. NormalHedge algoritmasında birikimsel pişmanlık R_t^k her bir özellik k ve her bir tur t için optimize edilerek yeni ağırlıklar $w_{t+1}^k, \ldots, w_{t+1}^K$ hesaplanır. R_t^k (4) formülü ile ifade edilir.

$$R_t^k = \sum_{\tau=1}^{t} r_\tau^k \tag{4}$$

Yakın geçmiş birikimsel pişmanlığın mevcut duruma etkisinin daha fazla olması gerekir. Çünkü düğümün davranışı, sonucu doğrudan etkilemektedir. Düzgün davranan bir düğüm yanlış davranmaya karar verirse ya da ele geçirilirse veya yanlış davranan bir düğüm doğru davranmaya karar verirse yakın geçmişe bakmak düğümlerin güven değeri hakkında daha doğru sonuçlar verecektir. Bunun haricinde güven hesaplamada kullanılan özellikler güven değerini etkileme oranı farklı olabilir. AHY'de bu gibi davranış ve etki değişikliklerini birlikte dikkate alarak birikimsel pişmanlık hesaplanır. Bu değişiklikler ve etki geçici bir durum olabileceği için her bir özelliğin kaybı l^k tur periyodu Δt için Gauss dağılımı ortalaması μ_t^k ve standart sapması σ_t^k'ya uygun olarak hesaplanır.

$$\mu_t^k = \frac{1}{\Delta t} \sum_{\tau=t-\Delta t+1}^{t} l_\tau^k, \tag{5}$$

$$\sigma_t^k = \sqrt{\frac{1}{\Delta t} \sum_{\tau=t-\Delta t+1}^{t} (l_\tau^k - \mu_\tau^k)}. \tag{6}$$

Daha sonra *t* turunda *k* özelliğinin kararlılığı (7) formülü ile hesaplanır.

$$s_t^k = \frac{|l_t^k - \mu_t^k|}{\sigma_t^k}. \tag{7}$$

s_t^k ne kadar düşük olursa ilgili özellik o kadar kararlıdır ve bu nedenle mevcut pişmanlığın birikimsel pişmanlık üzerindeki etkisinin yüksek olması sağlanır. Tersi durumlarda da ilgili özellik çok değişken demektir. Bu nedenle birikimsel pişmanlık hesaplanırken geçmiş birikimsel pişmanlığın etkisinin yüksek olması sağlanır. Bu ilkeye dayanarak (8) kullanılarak adaptif birikimsel pişmanlık hesaplanır.

$$R_t^k = (1 - \alpha_t^k)R_{t-1}^k + \alpha_t^k r_t^k, \tag{8}$$

$$\alpha_t^k = \min(g, \exp(-\gamma s_t^k)), \tag{9}$$

γ ölçekleme faktörü iken g mevcut pişmanlık üzerindeki maksimum oranı ifade etmektedir. NormalHedge algoritmasında olduğu gibi adaptif hedge algoritmasında da birkimsel pişmanlığı minimize etmek için aşağıdaki yöntem (10) kullanılır,

$$w_{t+1}^k \propto \frac{[R_t^k]_+}{c_t} \exp(\frac{{[R_t^k]_+}^2}{2c_t}), \tag{10}$$

$[R_t^k]_+, \max(0, R_t^k)$ gösterirken c_t ölçekleme parametresidir ve aşağıdaki denklem (11) çözülerek bulunur.

$$\frac{1}{K}\sum_{k=1}^{K} \exp(\frac{{[R_t^k]_+}^2}{2c_t}) = e. \tag{11}$$

N düğümüne ait güven değeri 12 formülünde de gösterildiği üzere, t döngüsündeki kayıp değerlerinin ağırlıklı ortalamasının 1'den çıkarılmasıyla hesaplanır. Kayıp ne kadar büyükse düğüm diğer düğümlerden o kadar uzaklaşmıştır. Diğer bir ifadeyle zararlı olmaya yaklaşmıştır.

$$\theta = 1 - \bar{l}_t^k. \tag{12}$$

Modelimizde Algoritma 1 kullanılır.

Algoritma 1 Model algoritması

1: Özellik değerleri 0-1 arasında normalize et. $\frac{N_k - min(N)}{max(N) - min(N)}$
2: Kayıp fonksiyonu için maksimum değerleri hesapla
3: Her düğüm için kayıp değerlerini bul.
4: Her düğüm için ağırlıklı ortalama kayıp hesapla
5: Anlık ve ortalama güven değerlerini hesapla
6: Yeni ağırlık değerlerini hesapla
7: **if** tur sayısı **mod** düğüm değiştirme tur sayısı == 0 **then**
8: Aday düğümleri güven değerine göre sırala
9: Seçim yöntemlerine göre değiştirilecek düğüm sayısı kadar rastgele sayı seç.
10: Seçilen sayılara karşılık gelen düğümleri değiştir.
11: **else**
12: Başlangıçtan devam et
13: **end if**

4.5 Komiteye Seçim

Güven değeri hesaplanan düğümlerin komiteye seçimi için [40] olduğu gibi üssel ve üçgensel seçim dağılımları kullanılabilir. Üssel dağılım seçimi yüksek güven değerine sahip düğümlerin seçimine öncelik verip düşük güven değerli düğümlerin seçimini önlemektedir. Üçgensel dağılım ise bütün düğümlere daha adil yaklaşmaktadır. Düşük güven değerine sahip düğümlerin de komiteye girmesine imkan tanır. Normal dağılıma uygun seçim ve en yüksek güven değerine sahip düğümlerin seçimi için de geliştirmeler yapılmıştır. Bu seçim sonuçları deneysel çalışmalar bölümünde anlatılacaktır.

4.6 Saldırı Analizi

BFT tabanlı sistemlerde sistemin bozulması için f+1 düğümün hatalı davranması gerekir. f hatalı düğüm sayısını gösterirken N toplam düğüm sayısını göstermektedir ve f=N/3 olarak kabul edilir. Hatalı düğüm sistemin çalışmasını engellemek istiyorsa GK'daki en az f+1 düğümü ele geçirmesi gerekir. Hatalı düğüm sistemi tamamıyla ele geçirmek ve istediği blokları oluşturmak istiyorsa ise GK'daki en az 2f düğümü ele geçirmesi gerekir. Sisteme saldırıyı iki farklı şekilde gruplayabiliriz:

- **Basit Saldırı:** Hatalı düğümlerin sürekli olarak hatalı davranması olarak tanımlayabiliriz. Hatalı düğüm hatalı davranmaya devam ettiği sürece güven değeri düşük çıkacaktır ve GK'ya giremeyecektir.

- **Gizli Saldırı:** Bu saldırı türünü, hatalı düğümlerin GK'daki çoğunluğu ele geçirinceye kadar dürüst davranması ve istediği sayıya ulaştığında hatalı davranmaya başlaması olarak tanımlayabiliriz. Hatalı düğümlerin GK'da çoğunluğa ulaşabilmesi için öncelikle çoklu kimlik oluşturarak aday komiteye dahil olması ve GK'ya seçilmesi gerekir. Hatalı düğümün GK'ya seçilebilmesi için güven değeri yüksek düğümler arasına girmesi gerekir. Bir düğümün güven değerinin yüksek kalması için hisse değerini yüksek tutması, hızlı ve doğru cevap vermesi şarttır. Hızlı ve doğru cevap koşulları kolaylıkla sağlanabilirken hisse değerinin yüksek olması için yatırım yapması gerekir. GK'ya seçim işlemi rasgele yapıldığı için güven değeri yüksek düğümlerin GK'ya seçilme garantisi yoktur. Hatalı düğümlerin GK'ya katılma olasılığını artırmak için çoklu kimlik oluşturmaları gerekir ki bu da yatırım yapmaları anlamına gelir. Bunlara ek olarak GK'dan çıkarılacak düğümler de rasgele seçilir. Hatalı düğümler aday komiteden GK'ya seçilirken, GK'daki hatalı düğümlerin GK'dan çıkma olasılıkları da vardır.

Ayrıca sisteme katılmak isteyen düğümlerden bir miktar depozito alınarak hatalı davrandığının tespit edilmesi durumunda ceza kesilmesi caydırıcı olacaktır.

4.7 Deneysel Çalışmalar

Bu bölümde önerilen yaklaşım deneyler açısından analiz edilmiş ve tartışılmıştır. İlk olarak, simülasyon kurulumu ayrıntılı olarak açıklanmaktadır. Daha sonra dokuz farklı test senaryosu anlatılmıştır. Bu testlerin her biri, önerilen yöntemin farklı koşullara nasıl tepki verdiğini değerlendirmek için yapılmıştır. Her test alt bölümünün sonunda sonuçlarla ilgili görüşler ve nedenler anlatılmıştır.

4.7.1. Simülasyon Kurulumu

Önerilen modelin doğruluğunu test etmek için, blok zincir ağını simüle eden bir yöntem tanımlanmıştır. Bu yönteme göre, önce bazı sayısal değerler belirlenmelidir. Bu sayısal değerler aşağıdaki gibidir:

- Güvenilir komitedeki düğüm sayısı **N**
- Güvenilir komitedeki zararlı düğüm sayısı **k**
- Aday düğüm sayısı **M**
- Aday düğümler içindeki zararlı düğüm sayısı **q**
- Düğüm değiştirme tur sayısı **r**
- Değiştirilecek düğüm sayısı **y**
- Dikkate alınacak geçmiş güven değeri sayısı **p**
- Örnekleme katsayısı **c**

Önerilen modelin davranışlarını test etmek için farklı deneyler yapmak üzere genel bir simülasyon kurulumu oluşturulmuştur. Önerilen modelin uzlaşma komitesini güvende tutma yeteneğini ölçmek için farklı değişkenler ve testler kullanılmıştır. Bu düzeneğin temel akışı şu şekilde verilir:

1. Başlangıç güven komitesi oluşturulur. **(N)**
2. Aday düğümler oluşturulur. **(M)**
    - Zararlı düğümler **(q)**
    - Dürüst düğümler **(M-q)**
3. Fikir birliği komite lideri bir blok önerir.

- Hem güvenilir komitedeki düğümler hem de aday düğümler blok doğrulaması yapar.
- Doğrulama süresince özellikler kullanılarak adaptif çit yöntemiyle ağırlıklar güncellenir. (Detaylı anlatım için 4.4 bölüm incelenebilir)
- Eğer düğüm değiştirme sayısına **(r)** ulaşıldıysa
  - Aday düğümler güven değerine göre sıralanır.
  - Güven komitesindeki belirli sayıdaki düğüm **(y)** adaylarla değiştirilir.
- 3. adıma dön.

İlk adımda, dürüst olduğu bilinen bir dizi N düğümü ile bir başlangıç güvenilir komite oluşturulur. Daha sonra (M) sayısı kadar aday düğümler oluşturulur. Adaylar arasında, bir dizi q düğümü kötü niyetli ve geri kalanı (M-q) dürüst olmak üzere seçilir. Kötü amaçlı düğümlerin blok zinciri bozacak davranışlar sergilediği kabul edilir. Bu nedenle özellik değerleri, isteklere cevap vermeyecek veya geç, hatalı yanıtlar verecek şekilde rastgele oluşturulur. Bu düğümlere rastgele olarak 0 ile 0.5 arasında yüksek gecikme değerleri ve 0 ile 0.5 arasında seçilen düşük hisse değerleri verilir. Dürüst düğümlere gelince, sistemdeki doğru davranışları simüle etmek için tasarlanmışlardır. Bir doğrulama talebi gelince beklemeden cevap verirler. Bu nedenle 0.5 ile 1 arasında gecikme değerlerine ve 0.5 ile 1 arasında hisse değerlerine sahip olacak şekilde özelliklere sahiptirler. Hem güvenilir komite hem de aday düğüm yanıtları davranış türlerine göre rastgele oluşturulur. Tüm özellik değerleri her turda özellik değeri üretildikten sonra [0,1] aralığına normalleştirilir. Gecikme özelliği için, 0 gecikme değeri en yüksek gecikmeyi gösterirken, 1 değeri en düşük gecikmeyi gösterir.

Deneysel akışın üçüncü adımı, yeni bir bloğun tam işlenmesini içerir. Bir blok önerildiğinde hem uzlaşma komitesindeki düğümler hem de aday düğümler gelen blok için doğrulama sonuçları üretir. Bundan sonra, her bir düğümün güven değeri Bölüm 4.4'de açıklanan adaptif çit yöntemine göre hesaplanır. Her r tur sonunda güvenilir komitedeki y adet düğüm aday düğümlerle değiştirilir. Güvenilir komitedeki en düşük güven değerine sahip y adet düğüm

komiteden çıkarılır ve yerlerine yeni düğümler geçer. Öte yandan, aday düğümler itibarlarına göre sıralanır, daha sonra yüksek itibarlı adaylardan y düğümün örnekleme faktörü c katı kadarı seçilir. Bu $c \times y$ düğümlerden her düğümün güvenilir komiteye girebilmesine imkan veren adil bir seçim yapılır ve aday düğümlerin bir alt kümesi elde edilir. Bu alt küme güvenilir komiteye dahil olacak düğümlerdir. Bu 3 adım her yeni blok geldiğinde tekrarlanır.

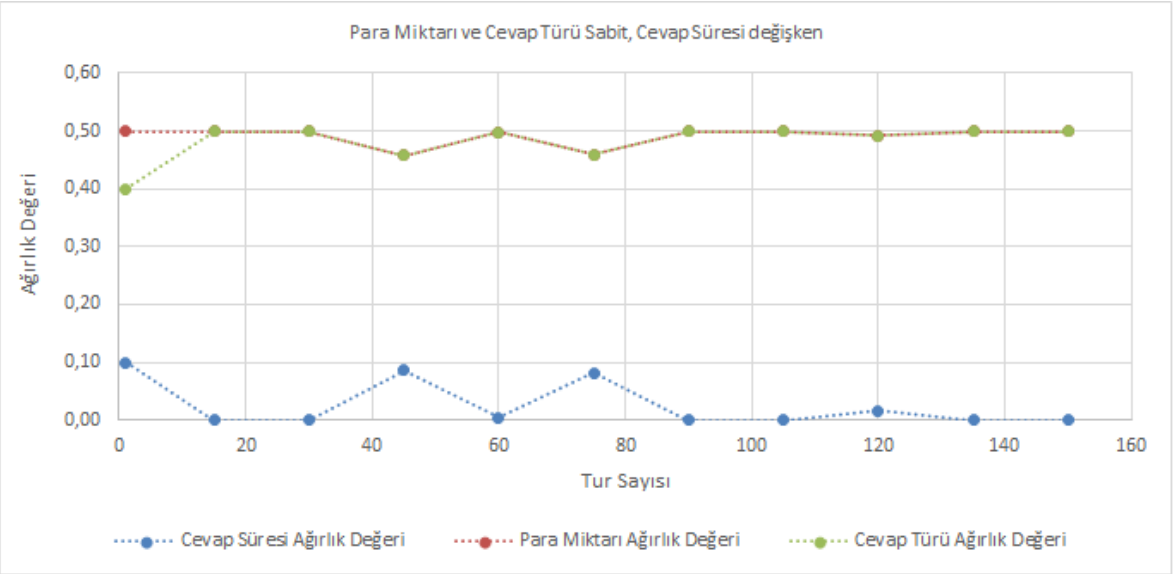
Bu genel test akışı, farklı koşullar altında önerilen model tepkisinin incelenmesinde kullanılır. Aşağıdaki alt bölümlerde sonuçlar ve tartışmalarıyla birlikte dokuz farklı deney ayrıntılı olarak verilmektedir. Verilen test sonuçlarının her biri 100 çalışmanın ortalaması alınarak elde edilmiştir.

4.7.2. Ağırlıkların Güncellenmesi Testi

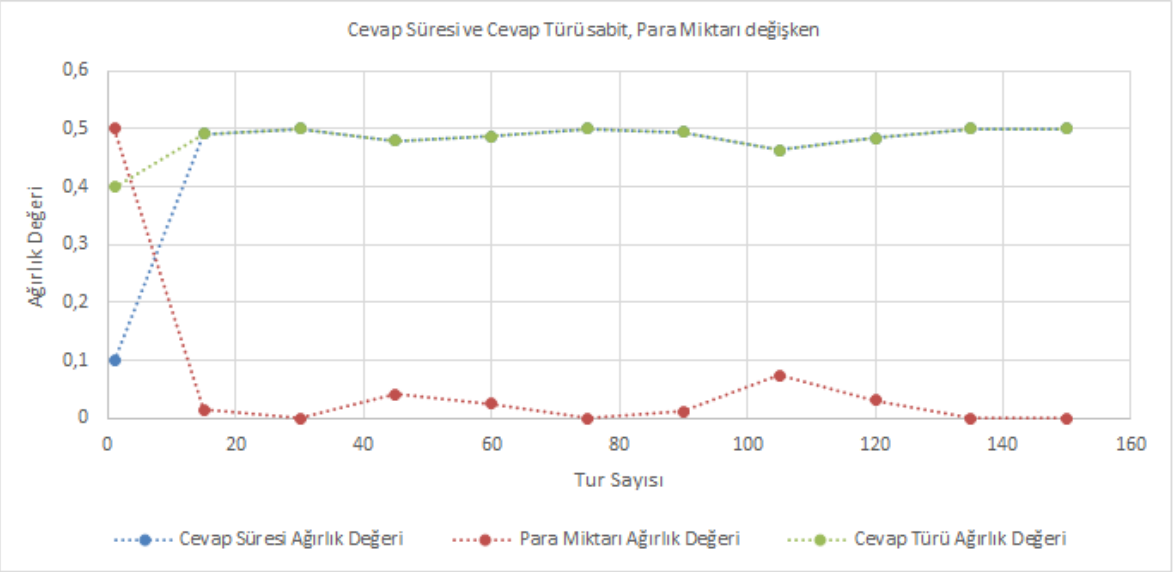
Modelimizde ağırlık değerlerini belirlemek için özellik değerlerini kullanıyoruz. Bir özellik ne kadar kararlı olursa, ağırlığı o kadar yüksek olur. Bu teoriyi test etmek için bu deneyi kurduk. Özellik vektöründeki ilk değer cevap süresi rastgele seçildi. Diğer iki özellik, hisse miktarı ve cevap türü sabit tutuldu. Bu testte değişken olan özelliğin ağırlığının diğer özelliklerin ağırlığından daha düşük olması ve kararlı, değişmeyen özelliklerin de aynı ağırlığa sahip olması beklenmektedir. Şekil 4.2'de görüldüğü gibi beklenen durum meydana gelmiştir. Ağırlıklar her turda değiştirilmiştir ve cevap süresinin ağırlığı diğerlerinden daha düşüktür. Cevap süresindeki kayıp güven değeri üzerinde diğer özelliklere göre daha az etkili olacaktır.

Şekil 4.3'te, gecikme ve yanıt türü sabit tutuldu ve para miktarı değişken olarak seçildi. Sabit tutulan özelliklerin ağırlıklarının eşitlendiği ve bu şekilde devam ettiği görülmüştür. Diğer özelliklerden daha değişken olan özelliğin ağırlığının diğer özelliklerden daha düşük olduğu görülmektedir.

Şekil 4.4'te, tüm özelliklerin değişmediği bir özellik vektörü kullanılır. Başlangıçta farklı

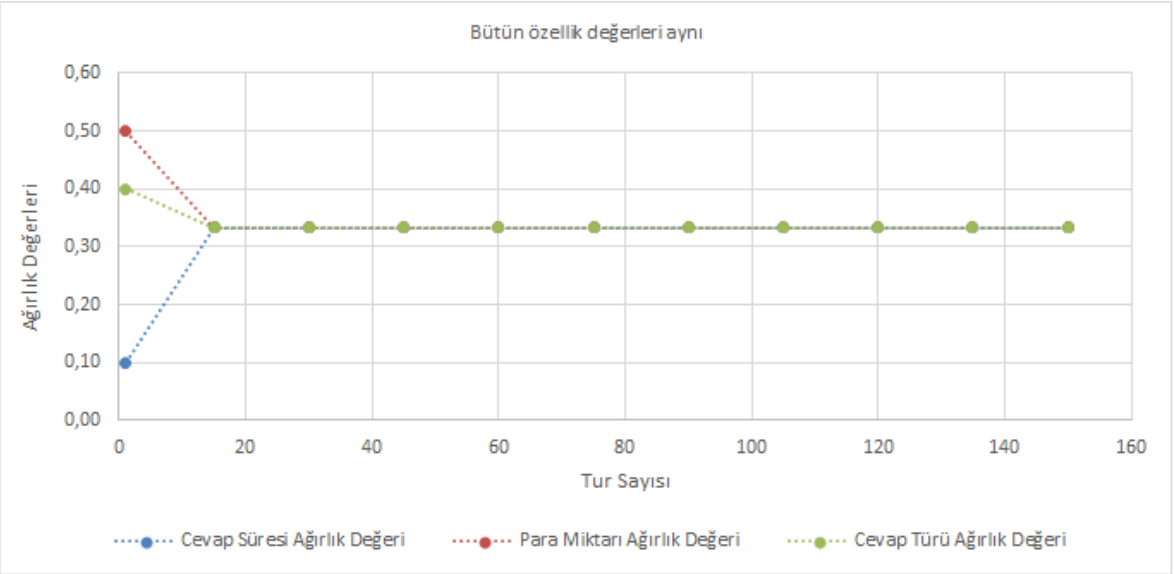


Şekil 4.2: Para Miktarı ve Cevap Türü Sabit, Cevap Süresi değişken



Şekil 4.3: Cevap Süresi ve Cevap Türü sabit, Para Miktarı değişken

olan ağırlık değerlerinin aynı değere ulaştığı ve bu şekilde devam ettiği görülebilir. Bu sonuçlara göre, önerilen modeldeki ağırlık değerlerinin güncellenmesi özelliklere göre başarıyla yapılmaktadır.

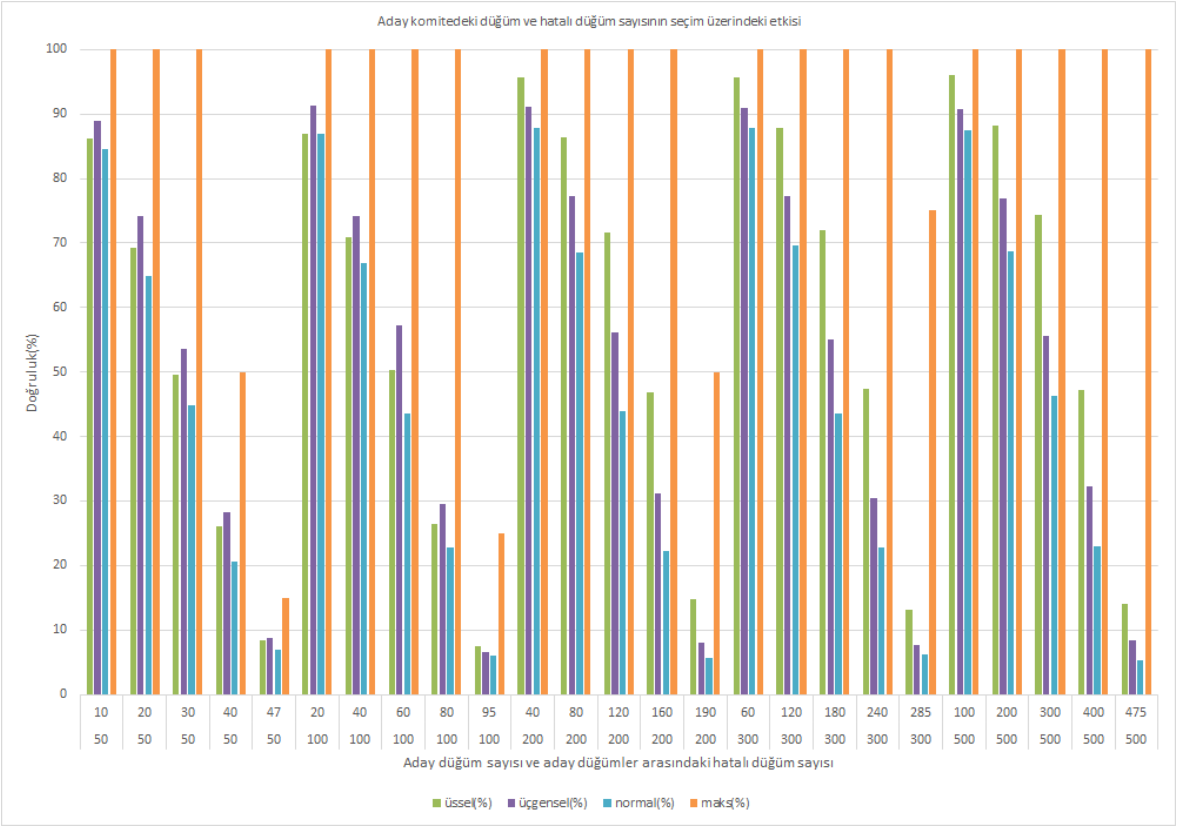


Şekil 4.4: Bütün özellik değerleri sabit

4.7.3. Aday Komitedeki Hatalı ve Dürüst Düğüm Sayısının Başarı Üzerindeki Etkisi Testi

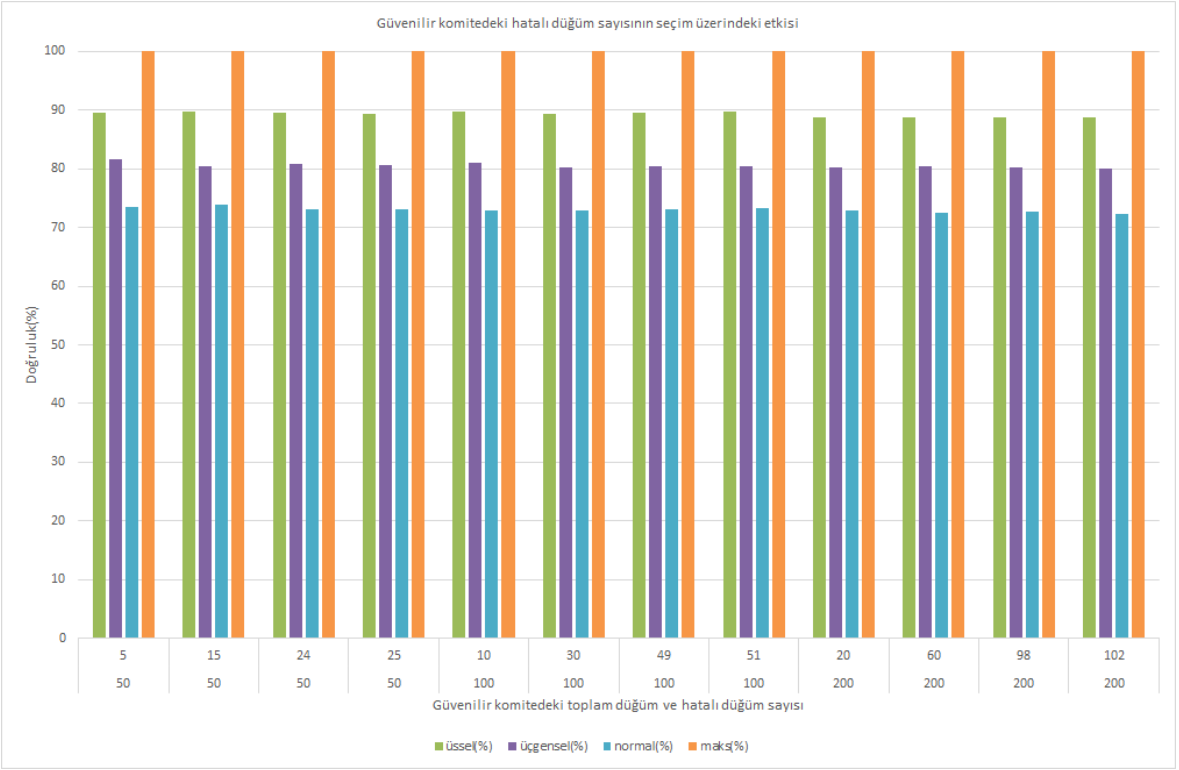
Aday düğüm sayısının seçim üzerindeki etkisini ölçmek için yapılan testte güvenilir komitedeki düğüm sayısı sabit tutularak aday düğüm sayısı ve aday komitedeki hatalı düğüm sayısı değiştirilerek seçilen düğümlerin hatalı olma yüzdesi hesaplanmıştır. Farklı seçim algoritmaları kullanılarak seçimin sonuçları Şekil 4.5'te gösterilmiştir. Üssel dağılıma göre seçim, üçgensel dağılıma göre seçim, normal dağılıma göre seçim ve en yüksek güven değerine sahip düğümlerin seçimi yöntemlerine göre seçim uygulanmıştır. Üssel dağılım güven değeri yüksek düğümlerin seçimine öncelik verirken üçgensel seçim güven değeri düşük düğümlerin seçimine de imkan vermektedir. Aday düğüm sayısı düşük olduğunda üçgensel ve üssel seçimin beklendiği gibi çalışmadığı görülmüştür. Düğüm sayısı arttıkça daha iyi sonuçların alındığı görülmüştür.

Maksimum güven değerine sahip düğümlerin seçilmesi yönteminde düzgün düğümlerin her zaman seçildiğini görebiliriz. Buna göre güven değeri hesaplama yönteminin başarılı bir şekilde çalıştığı görülmektedir. 500 aday düğümden 490'ının hatalı davranması durumunda



Şekil 4.5: Aday komitedeki düğüm ve hatalı düğüm sayısının seçim üzerindeki etkisi

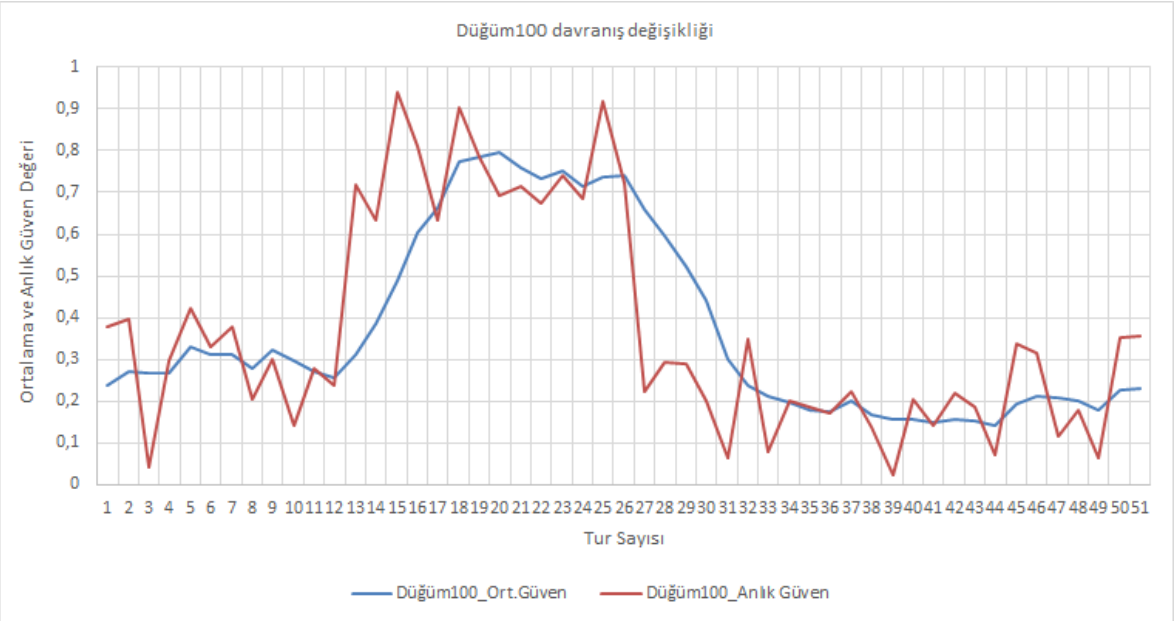
bile seçilecek düğüm sayısı az olduğu durumlarda maksimum seçme yönteminde hatalı düğümün seçilme yüzdesi 0 olmaktadır. Aday düğümlerin yarısının hatalı olduğu durumda bile üssel seçim yönteminde hatalı düğümün seçilme yüzdesinin ortalama yüzde 20 civarında olduğu görülmektedir. Şekil 4.5'teki sonuçları incelediğimizde aday düğüm sayısından çok aday düğüm içindeki hatalı düğümlerin sayısının önemli olduğunu söyleyebiliriz. Hatalı düğüm sayısı arttıkça güvenilir komiteye hatalı düğüm seçme olasılığı artmaktadır.



Şekil 4.6: Güvenilir Komitedeki Hatalı ve Dürüst Düğüm Sayısının Başarı Üzerindeki Etkisi Testi

4.7.4. Güvenilir Komitedeki Hatalı ve Dürüst Düğüm Sayısının Başarı Üzerindeki Etkisi Testi

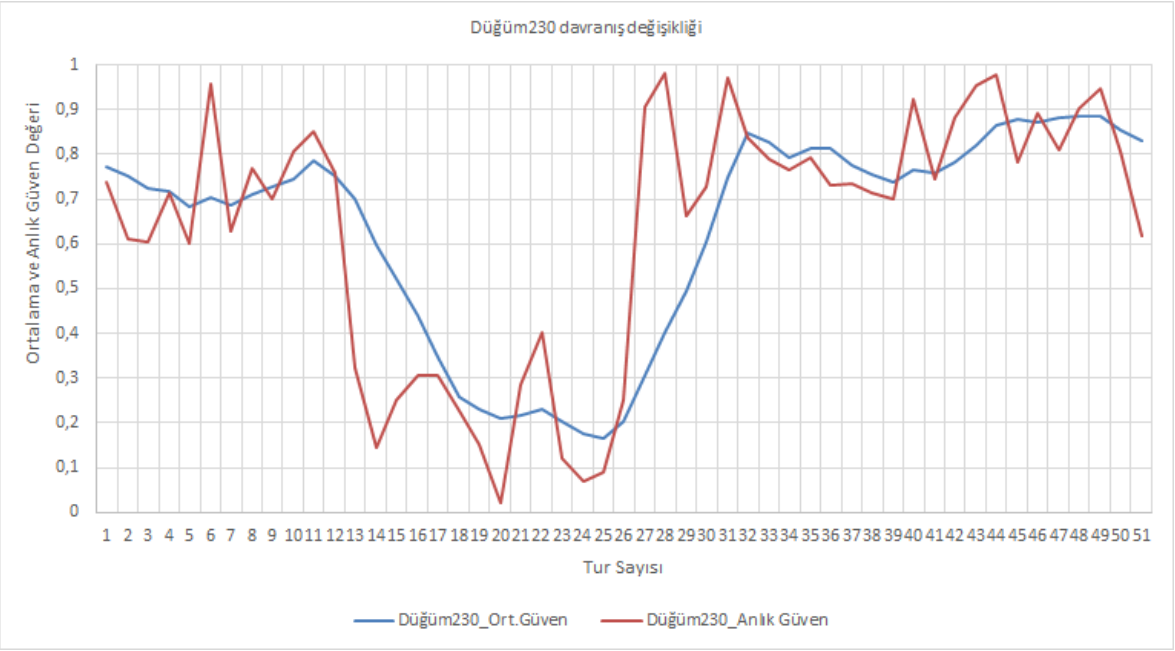
Test sonucunda Şekil 4.6 incelendiğinde, güvenilen komitedeki hatalı düğüm sayısının hatalı düğüm seçimi yüzdesini etkilemediğini söyleyebiliriz. Ancak ağırlık değişimleri incelendiğinde ağırlıkların beklendiği gibi güncellenmediği ve sadece cevap tipi özelliğinin kullanıldığı görülmektedir. Diğer özelliklerin itibar değeri üzerindeki etkisinin sıfırlandığı gözlenmiştir. Tüm özelliklerin yararlı olması için, güvenilir komitedeki hatalı düğüm sayısının toplam düğüm sayısının en fazla yarısı kadar olmalıdır. Veri dönüşümü yapılırken çoğunluğun yanıt türü 1 olarak kabul edildiği için yalnızca yanıt türü özelliği kullanılmıştır. Dürüst düğümler azınlıktayken diğer özelliklerin ağırlıkları en aza indirilir. Önerilen yöntemde ağırlıkların sıfırın altına düşmesine izin verilmez. Sıfırdan küçük ağırlıklar 0 olarak kabul edilir. Bu nedenle diğer özelliklerin ağırlığı sıfır olmuştur.



Şekil 4.7: Davranış Değişikliklerine Karşı Modelin Başarısı

4.7.5. Davranış Değişikliklerine Karşı Modelin Başarısının Testi

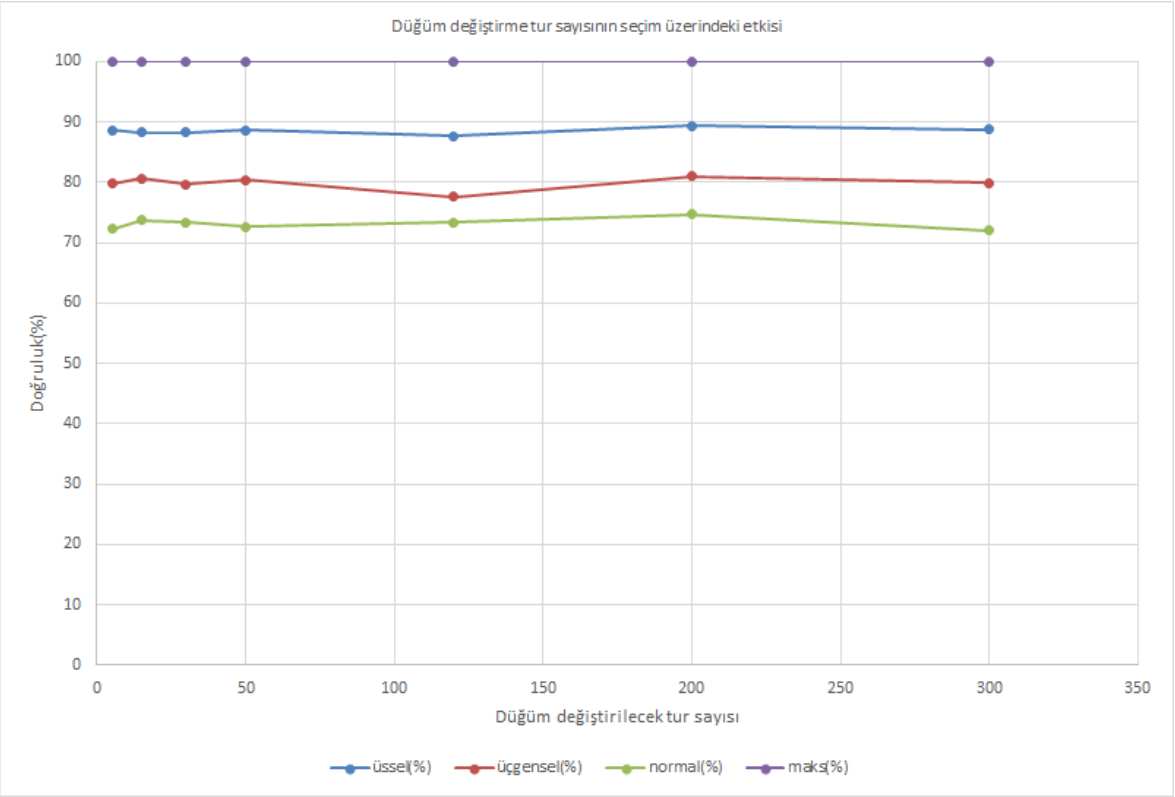
Başlangıçta hatalı davranan bir düğümün sonrasında dürüst davranmaya karar verdiğinde veya tersi durumda başlangıçta dürüst davranan bir düğümün sonrasında hatalı davranmaya karar verdiğinde güven değerinin nasıl değiştiğini görmek için yeni bir test düzenlendi. Rastgele iki düğüm 100. düğüm (Düğüm100 olarak adlandırılacaktır.) ve 230. düğüm (Düğüm230 olarak adlandırılacaktır) seçilerek test gerçekleştirilmiştir. Düğüm100 belirli sayıda tur sırasında yanlış davranır ve daha sonra dürüst davranır ve daha sonra tekrar yanlış davranmaya döner. Düğüm230 için ise, Düğüm100'ün ters durumu oluşturulur. Değişimi daha net görebilmek için Düğüm100'ün dürüst davranmaya başladığı zamandan sonra oluşturulan özellikler güvenilir komite ortalamasına sabitlenmiştir. Şekil 4.7 ve 4.8'de görüldüğü üzere Düğüm100 dürüst davranmaya başladığı andan itibaren güven değeri artmıştır ve belirli tur sonunda güven değerinin tam olduğu görülmektedir. Düğüm230 için ise hatalı davranmaya karar verdiği andan itibaren güven değeri düşmeye başlamıştır. Buna göre önerilen yöntem değişim gösteren davranışlara adapte olabilmektedir. Şekil 4.7 ve 4.8'de Düğüm100 ve Düğüm230 11 ile 25. tur arasında davranış değişikliği göstermişlerdir.



Şekil 4.8: Davranış Değişikliklerine Karşı Modelin Başarısı

4.7.6. Düğüm Değiştirilecek Tur Sayısının Model Başarısı Üzerindeki Etkisi Testi

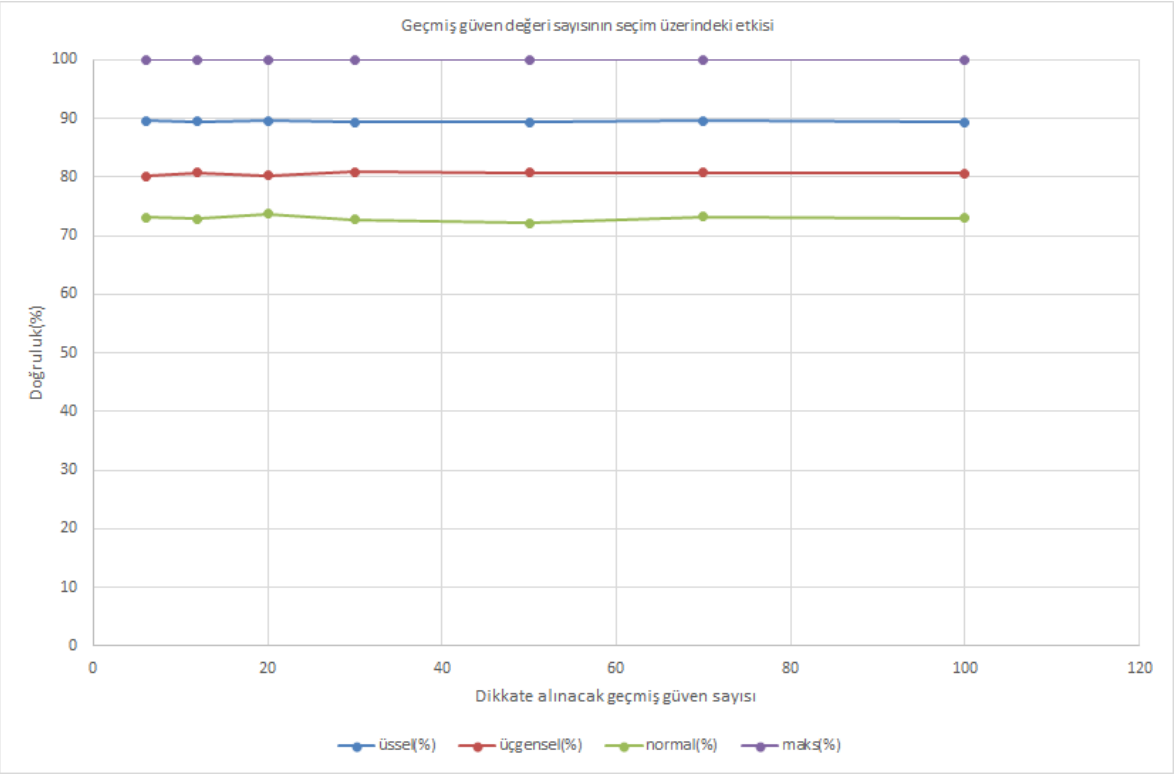
Düğüm değiştirme tur sayısının başarı üzerindeki etkisini ölçebilmek için bir test yapılmıştır. Yapılan bu testte düğüm değiştirilecek tur sayısı haricindeki değişkenler sabit tutulmuştur ve tur sayısı değiştirilmiştir. Farklı her tur sayısı için GK'ya seçilen düğümlerin hatalı olma oranı bulunmuştur. Şekil 4.9 de elde edilen sonuçlara göre düğüm değiştirilecek tur sayısının başarı üzerinde belirgin bir etkisi bulunmamaktadır. Düğüm değiştirilecek tur sayıları aralığındaki özellik değerlerine göre dalgalanmalar oluşmaktadır. Güven değeri tam oluşmadan düğümlerin değiştirilmesi GK'ya hatalı düğüm girme olasılığını artıracaktır. Tur sayısının fazla seçilmesi de aday düğümlerin istekli davranmasına negatif etki yapabilir. Uzun süre seçilmeyi beklemek zorunda kalan düğümlerde isteksizlik oluşmasına sebep olabilir.



Şekil 4.9: Düğüm Değiştirilecek Tur Sayısının Model Başarısı Üzerindeki Etkisi

4.7.7. Güven Değeri Hesaplanırken Geçmiş Güven Değeri Sayısının Başarı Üzerindeki Etkisi

Düğümün güven değeri hesaplanırken belirli sayıdaki güven değerinin ortalaması alınır. Bunun nedeni anlık hatalı davranma durumunun düğümü güvensiz olarak belirlemesinin engellenmesidir. Aynı şekilde anlık güvenilir davranma durumunun düğümü güvenilir olarak belirlemesinin engellenmesidir. Geçmiş güven sayısı hatalı düğümler hatalı davranmaya ve dürüst düğümler dürüst davranmaya devam ettiği sürece başarı oranını etkilememektedir. Şekil 4.10'da bu durum görülebilir. Davranış değişikliklerinin güven değeri üzerindeki etki süresinin kısa veya uzun olması geçmiş güven değeri sayısına bağlıdır. Şekil 4.8'de görüleceği üzere güven değerinin artma oranı geçmiş güven değeri sayısıyla doğru orantılıdır.

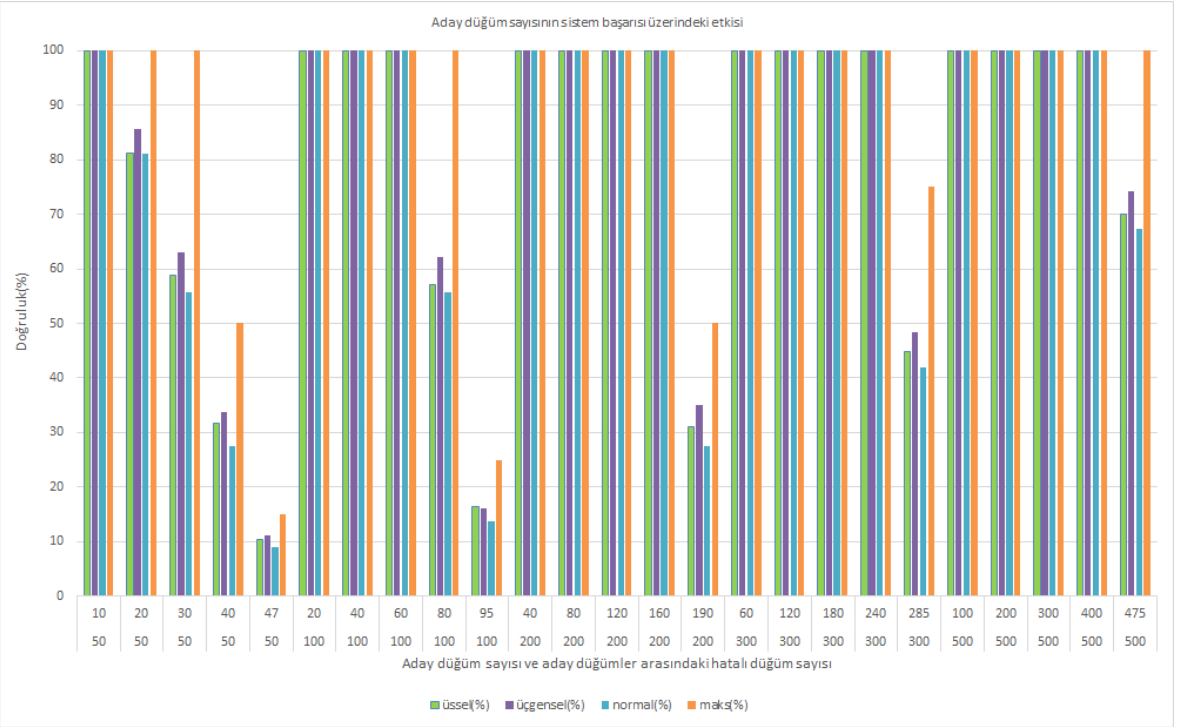


Şekil 4.10: Geçmiş güven değeri sayısının seçim üzerindeki etkisi

4.7.8. Örneklemden Seçim

Önerilen modelin güven değerlerini başarılı bir şekilde hesaplayabildiği görülmektedir. Güvenilir komiteye seçim yapılırken bütün düğümlere bakıldığı için güven değeri hesaplamadaki başarı seçim üzerine tam yansımamaktadır. Bu nedenle başarı oranını artırabilmek için adaylar arasından örneklem oluşturularak seçimin bu örneklem üzerinden yapılması yöntemi denenmiştir. Bu yönteme göre aday düğümler güven değerine göre sıralanır. Değiştirilecek düğüm sayısının örneklem katsayısı kadar katı kadar aday sıralı şekilde alınarak yeni bir seçim kümesi oluşturulur. Seçilecek düğümler bu yeni kümeden seçilir.

Örneklemden seçim yönteminin sonuçları olan Şekil 4.11 ve 4.12'de de görüleceği üzere hatalı düğümlerin güvenilir komiteye seçilme yüzdesi düşmüştür. Yani sistemin başarı oranı artmıştır. Aday komitedeki hatalı düğüm sayısının yarıdan fazla olduğu durumlarda bile hatalı düğüm seçilme yüzdesinin çok düşük veya sıfır olduğu görülmektedir. Şekil 4.11

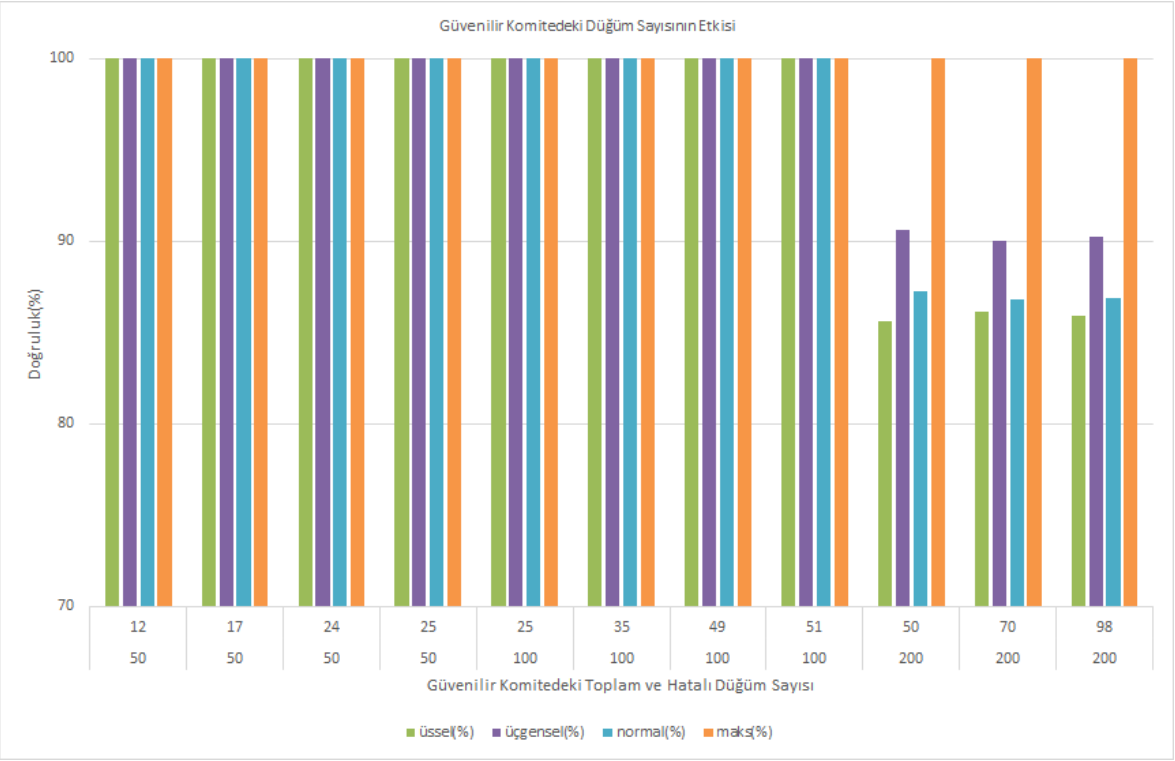


Şekil 4.11: Örneklemden Seçim-Aday düğüm sayısının sistem başarısı üzerindeki etkisi

da ise örneklemden seçim kullanıldığında aday komitedeki hatalı düğüm sayısının başarı üzerindeki etkisini görebiliriz. Bu test için güvenilir komitedeki düğüm sayısı 100, aday düğümlerin yarısı hatalı, değiştirilecek düğüm sayısı güvenilir komitenin yüzde 20'si ve örneklem katsayısı 2 olarak belirlenmiştir.

Şekil 4.11 da güvenilir komite düğüm sayısı 50, güvenilir komitedeki hatalı düğüm sayısı düğüm sayısının yüzde 35'i, değiştirilecek düğüm sayısı güvenilir komitenin yüzde 20'si ve örneklem katsayısı olarak da 2 belirlenmiştir.

Şekil 4.11'i Şekil 4.5 ile Şekil 4.12'yi Şekil 4.6 ile karşılaştırdığımızda başarı oranının arttığını görebiliriz.



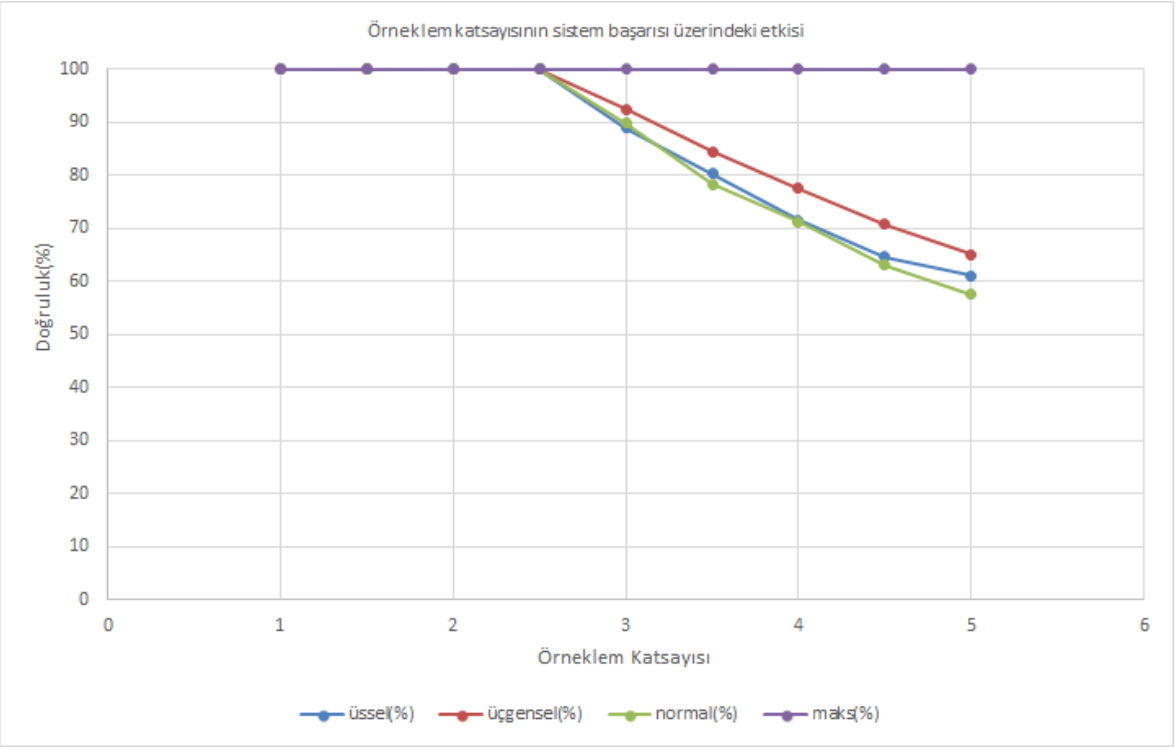
Şekil 4.12: Örneklemden Seçim-Güvenilir komitedeki düğüm sayısının sistemin başarısı üzerindeki etkisi

4.7.9. Örneklem Katsayısının Başarı Üzerindeki Etkisi

Örneklem katsayısının sistemin başarısı üzerindeki etkisini ölçmek için yeni bir test yapılmıştır. Bu testte aday düğüm sayısı 100, aday düğümler içinde hatalı düğüm sayısı aday düğümlerin yarısı, güvenilir komitedeki düğüm sayısı 100, güvenilir komitedeki hatalı düğüm sayısı güvenilir komitenin yüzde 35'i değiştirilecek düğüm sayısı da güvenilir komitenin yüzde 20'si olarak belirlenmiştir. Buna göre örneklem katsayısı arttıkça hatalı düğümler örneklem içine girmektedirler. Bu nedenle örneklem katsayısı arttıkça sistemin başarısı düşmektedir. Şekil 4.13'de bu durum görülmektedir.

4.7.10. Değiştirilecek Düğüm Sayısının Sistem Başarısı Üzerindeki Etkisi

Bu testte, aday düğüm sayısı 100, aday düğümlerdeki hatalı düğüm sayısı; aday düğümlerin yarısı olarak ve güvenilir komitedeki düğüm sayısı 100, güvenilir komitedeki hatalı

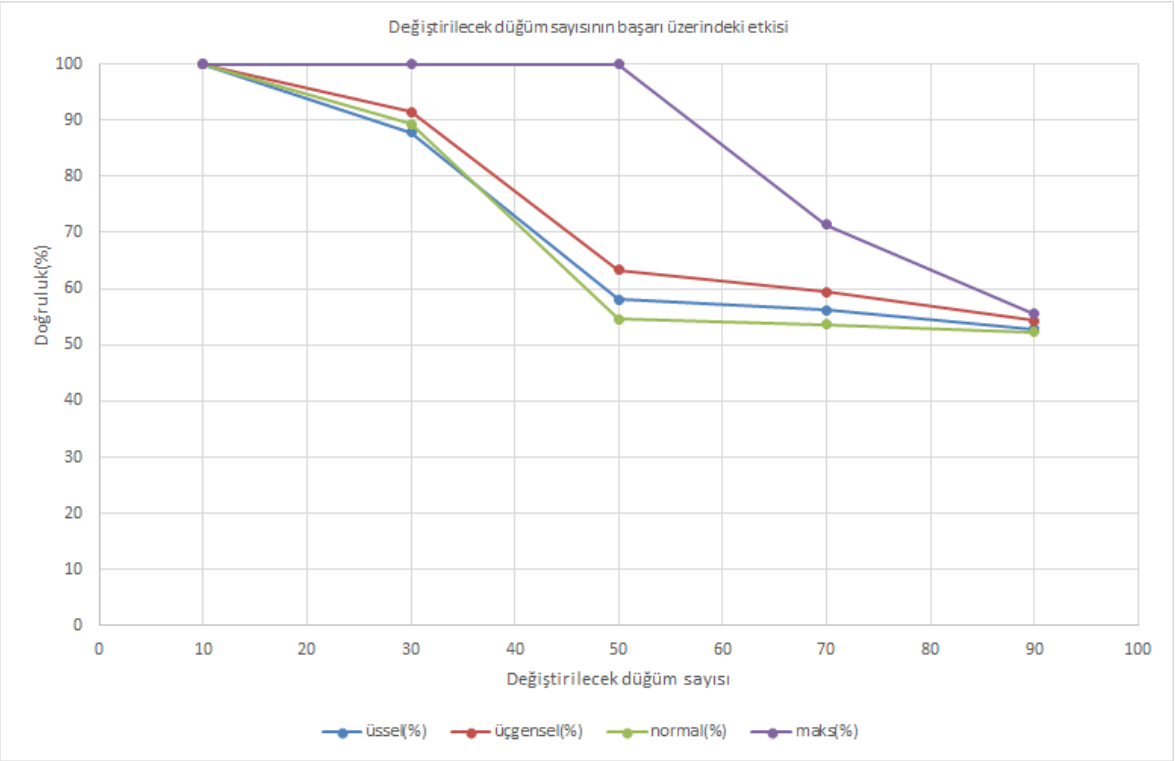


Şekil 4.13: Örneklem Katsayısının Başarı Üzerindeki Etkisi

düğüm sayısı güvenilir komitenin yüzde 35'i olarak belirlenmiştir. Değiştirilecek düğüm sayısı arttıkça yanlış düğümlerin seçilme olasılığı arttığı için sistemin başarısı azalmıştır. Değiştirilecek düğüm sayısı hatalı düğüm sayısına yaklaştıkça veya değiştirilecek düğüm sayısı hatalı düğüm sayısından fazla olduğunda başarı oranı azalır. Test sonuçları Şekil 4.14'te gösterilmektedir.

4.7.11. Performans Kıyaslaması

5000 düğümlü bir ağla, iki çekirdekli i7 3.60 GHz işlemcili bir bilgisayar kullandığımızda, güven değeri hesaplaması bir turda yaklaşık 74 ms sürmektedir. Modelimizi kodlamak için Python programlama dili kullanılmıştır. Güven değeri hesaplanması aday düğümler için Python'da kolayca paralelleştirilebilir. Böylece blok yayınlamada kullanılacak fikir birliği oluşturma aşamasına 74 ms'den daha az zaman eklenecektir. Yalnızca uzlaşma komitesinin güncellenmesi daha uzun sürebilir ve ek zaman gerektirebilir. Bu adım kurulum aşaması gibi



Şekil 4.14: Değiştirilecek Düğüm Sayısının Başarı Üzerindeki Etkisi

düşünülebilir ve günlük olarak gerçekleştirilebilir. Bu çalışmada zaman ve hız açısından bir kısıtlama bulunmamaktadır. Zaman ve hız kısıtı da dikkate alınarak daha kapsamlı bir çalışma yapılabilir.

Diğer çalışmaların çoğu, fikir birliği komitesi seçimi için PoW kullanır ve fikir birliği komitesindeki kötü amaçlı düğümlerin oranı hakkında yeterli bilgi vermez. Guru [40] olarak adlandırılan bir yaklaşım önerdiğimiz modele benzer bir çalışma yapmıştır. Bu çalışma ile aynı koşulları modelimize uygulayarak sonuçlar elde ettik. (harici itibar yok; ağdaki 5000 düğüm; 100 komite boyutu; kötü niyetli oran 0.1, 0.25, 0.33 ve 0.45; 10000 tur ve 100 tekrar.) Tablo 4.1'de farklı zararlı düğüm oranları için elde edilen sonuçlar verilmiştir.

Çizelge 4.1: Başarı Oranı Karşılaştırması

Hatalı Düğüm Oranı	Guru (Üssel)	Önerilen Model (Üssel)	Guru (Üçgensel)	Önerilen Model (Üçgensel)
0.1	100%	100%	100%	100%
0.25	99.7%	**100%**	98.8%	**100%**
0.33	99.6%	**100%**	96.3%	**100%**
0.45	96.5%	**100%**	60%	**100%**

5. PARÇALAMA MODELİ

Bu bölümde dikey ölçekleme olarak da adlandırılan parçalama yönteminde düğümlerin parçalara atanmasında güven değerinin kullanılması için önerilen model anlatılacaktır. Modeli anlatmadan önce parçalama çalışmaları düğümlerin parçalara atanması özelinde özetlenecektir. Yaptığımız deneysel çalışmalarda önerdiğimiz model ile [26, 53] çalışmalarının sonuçları karşılaştırmalı olarak verilmiştir.

5.1 Giriş

Blokzincirin güvenilir bir otoriteye ihtiyaç duymadan işlemleri yapabilme yeteneği sayesinde farklı alanlarda kullanılabileceği düşünülmüştür [68–72]. Blokzincirin kullanımı ve işlem hacmi arttıkça ölçeklenebilirlik problemi olduğu ortaya çıkmıştır. Bu problemin çözümü için değişik çalışmalar yapılmıştır. [61, 62, 64, 73] çalışmaları sadece yan zincir, özel kanal çözümlerine odaklanırken, [15, 24, 27, 28, 31, 74–80] çalışmaları uzlaşma algoritmasını değiştirerek ölçeklenebilirlik problemine çözüm önerisi sunmuşlardır. [25, 29, 30, 45, 47–53, 81] çalışmaları ise uzlaşma algoritması değişikliğine ek olarak sharding yöntemi de uygulamışlardır.

5.2 Arkaplan

Bir bütünü farklı parçalara bölerek parçaları daha kolay yönetme ve daha hızlı işlem yapabilme özelliği kazandırma işlemine parçalama denir. Blokzincir ölçekleme çalışmalarında son zamanlarda sıkça başvurulan yöntemlerin başında gelir. Farklı bölümlerin aynı anda hareketlerin işlenmesini sağladığı için birim zamanda doğrulanan hareket sayısını artırmaktadır. Her parça kendi blokzincirini çalıştırır. Farklı parçaları ilgilendiren hareketler için sistemin tutarlılığının sağlanması adına iki aşamalı yazma[46, 82] ve iki aşamalı kilitleme[83] gibi yöntemler kullanılmaktadır. Parçalama yapabilmek için blokzincir ağındaki düğümleri

farklı bölümlere ayırmak gerekmektedir. Düğümlerin parçalara atanmasında dört temel yaklaşım kullanılmaktadır.

1. **PoW çözümü:**
   Bu yöntemde düğümlerin kimlik bilgilerini de ekleyerek bir kriptografik özet değeri elde etmesi beklenir. Elde edilen kriptografik değeri belirli bir zorluk derecesinden küçükse bu PoW çözümü kabul edilir. PoW çözümünde kullanılan *nonce* değeri veya kriptografik özet değeri kullanılarak düğümlerin parçalara ataması yapılır.

2. **Doğrulanabilir Rasgele Fonksiyonların[9] Kullanımı (Verifiable Random Functions-VRF):**
   Bu yöntemde VRFte kullanılmak üzere bir tohum değerinin oluşturulması gerekir. Tohum değeri, blokzincirdeki son bloğun kriptografik özeti alınarak veya ağdaki düğümlerin ürettikleri rasgele değişkenlerin birleştirilmesi sonucunda elde edilir. Tohum değeri elde edildikten sonra VRF fonksiyonunda kullanılır. VRF ile oluşturulan rasgele değere göre düğümler parçalara atanır.

3. **Doğrulanabilir Gecikmeli Fonksiyonların[84] Kullanımı (Verifiable Delayed Functions-VDF):**
   Rasgele değerin önceden oluşmasını engellemek için veya lider düğümün rasgele değeri kendi çıkarları için kullanmasını engellemek için VRF' e ek olarak ikinci bir tohum değeri oluşturularak kullanılır. İlk tohum değeri VRF ile oluşturularak bloğa yazılır. Bu rasgele oluşturulan ilk değer kullanılarak ikinci bir tohum değeri oluşturularak düğümlerin parçalara atanmasında kullanılır.

4. **Herkesçe Doğrulanabilir Gizli Paylaşım[85] Kullanımı (Publicly Verifiable Secret Sharing-PVSS):**
   Her düğüm kendine özel bir Lagrange polinomu oluşturur. Bu polinomun derecesi en az kaç katsayı ile baştaki değerin oluşturulabileceğine göre değiştirilir. Polinom derecesine göre katsayılar farklı düğümlere dağıtılır. Daha sonra t tane rasgele düğümden gönderilen katsayılar alınarak rasgele değerler elde edilir. Elde edilen rasgele değerlere göre düğümlerin parçalara ataması yapılır.

Her bir parça kendi içinde blok doğrulamasını ve yayınlamasını yapar. Her düğüme her hareketi işlemeyi bıraktırdığınızda, ağın geri kalanı bu işlemleri gerçekleştirmediğinden geçersiz işlemlerin bir kötü amaçlı düğüm kümesi tarafından kaydedilebileceği ve denetlenmeden kalma riskini teorik olarak yükseltirsiniz. Ağdaki düğüm sayısı arttıkça, ağı hareketlerin işlenmesi yükünü paylaşabilecek parçalara ayırma kabiliyeti artar ve aynı zamanda her bir parça güvenilir uzlaşma için yeterince büyük kalır.

Bu yöntemler düğümlerin güvenilirliği hakkında fikir sahibi değildir. Parçalar blokzincir ağına göre daha az sayıda düğümden oluştuğu için bozulma olasılıkları daha yüksektir. Bu olasılığı azaltmak için düğümlerin parçalara atanmasında güven değerinin kullanılması önerilmiştir.

5.3 Genel Yapı ve Temel İlkeler

Parçalara oluşturulurken parçalardaki aktif düğüm sayısının takip edilmesi ve belirli bir sayının altına düştüğü zaman parçaların birleştirilmesi ve yeniden düzenlenmesi gerekecektir. Bu nedenle güvenilir komite sürekli parçalardaki düğümlerle iletişim halinde olmak zorundadır. Güvenilir komite her parçadaki bütün düğümlerden hesapladıkları güven değeri bilgileri alabilmek için sürekli iletişim halindedir. Belirli bir süre içinde cevap alamazsa düğümün çevrim dışı olduğuna karar verilir. Herhangi bir parçadaki aktif düğüm sayısı belirli bir sayının altına düşerse o parçaya yeni düğüm ataması yapılarak aktif düğüm sayısı artırılır. Aktif düğüm sayısı belirli bir sayının altına düşen parça sayısı fazla olursa parçaların birleştirilmesi işlemi gerçekleştirilir.

Parçalara atama işlemini iki farklı zaman için düşünebiliriz: 1) Gün içinde parçalardaki aktif düğüm sayısının belirli bir seviyenin altına düşmesi 2) Gün içinde birden fazla parçadaki aktif düğüm sayısının belirli bir seviyenin altına düşmesi 3) Gün başında parçaların oluşturulması İlk senaryo için sadece ilgili parçaya yeni düğüm ataması yapılır. 2 ve 3. Senaryo için ise blok doğrulama işlemi durdurularak yeniden parçaların oluşturulması yapılır. Oluşturulan parçalardan biri güvenilir komite olarak da görev yapar. Parçalar oluşturulurken aşağıdaki

kurallara dikkat edilir: 1. Parçalardaki ortalama güven değerlerinin birbirine yakın olması 2. Parçalardaki düğüm sayılarının birbirine yakın olması

5.4 İdeal Parça Sayısının ve Parçadaki Düğüm Sayısının Bulunması

Blokzincir parçalara ayrılırken parçaların güvenliğini tehlikeye atmayacak ve BFT gerçekleştirebilecek kadar sayıda düğümden oluşmasına dikkat edilmelidir. Düğüm sayısı arttıkça dürüst düğümlerin sisteme dahil olma olasılığı artar. Fakat BFT performansı düğüm sayısı arttıkça azalır. Diğer bir ifadeyle düğüm sayısı arttıkça güvenlik artarken sistem performansı düşer [86, 87]. Bu nedenle kullanılacak BFT algoritmasına göre her bir parçadaki düğüm sayısı belirlenebilir. Bu nedenle NEO [88] 7 düğüm, EOS[89] 21 düğüm, Ripple[32] 32 düğüm ve HyperLedger[81] 16 düğüm ile sınırlanmışlardır. Daha gelişmiş BFT kullanan yöntemler daha fazla düğüm sayıları ile başarılı bir şekilde çalışabilmektedirler. Örneğin HoneyBadger[21] 64 düğüm ile, ByzCoin[24] 100'den fazla düğüm ile ve LinBFT[90] 200 den fazla düğüm ile başarılı bir şekilde çalışabilmektedir. Parça sayısı arttıkça aynı anda gerçekleşen işlem sayısı artacaktır. Bu nedenle düğüm sayısı göz önünde bulundurularak parça sayısı belirlenebilir.

5.5 Parçalara Atama Yöntemleri

Adaptif Hedge yöntemiyle düğümlerin güven değeri hesaplandıktan sonra hangi düğümün hangi parçaya atanacağının belirlenmesi gerekir. Güven değerleri bulunup parça sayısı belirlendikten sonra ortaya çıkan durum çoklu sayı bölümleme[91] problemine denk gelir. Çoklu sayı bölümleme probleminde amaç verilen bir sayı kümesini toplamları eşit olacak şekilde alt kümelere bölmektir. Sayı kümelerini ikiye bölmek için bazı algoritmalar olsa da, bu algoritmalar çok yönlü bölümlemede yüksek performans ile çalışmamaktadır. Sayı seti ne kadar fazla alt kümeye bölünmek isteniyorsa kullanılan algoritmanın da performansı o kadar fazla olmalıdır. Çünkü alt küme sayısı ne kadar fazla ise çalışma zamanı da o kadar artmaktadır.

Ayrıca sayı bölümleme algoritmalarında neredeyse hep aynı dağılım elde edilir. Sistemi izleyen zararlı düğümler aynı parçaya atanacak şekilde özellikler oluşturarak bir parçayı ele geçirebilir. Bu nedenle tahmin edilmesi zor rasgele bir yöntem kullanılması sistemin daha güvenli olmasını sağlayacaktır.

Dağıtık rasgelelik [43] gibi yöntemler eklenerek tamamen rasgele seçim yapılarak parçaların daha güvenli olması sağlanabilir. Bu çalışmamızda güven değeri kullanımının parçalardaki hatalı düğüm sayısını azalttığını göstermeye yönelik olduğu için daha basit rasgele seçim yöntemi kullanılmıştır. Ayrıca rasgelelik kullanılmadan sezgisel bir yaklaşım kullanılarak güven değeri kullanımının hatalı düğümlerin parçalar arasında dengeli dağıtılmasını sağladığı görülmüştür.

Bilindik kümeleme yöntemleri benzer özelliklere sahip elemanları aynı kümeye toplamaya çalışır. Bizim burada istediğimiz ise farklı özellikteki düğümleri aynı kümeye toplayabilmektir. Güven değerinin parçalara dengeli dağıtılması için güven değeri yüksek düğüm ile güven değeri düşük düğümün aynı kümeye atanması gerekir. Kümeleme yöntemlerindeki benzerlik fonksiyonu değiştirilerek istenilen şekilde kümeleme yapılabilir.

5.6 Toplam Güven Değeri En Yakın Yöntemi

Bu yöntem ile düğümlerin parçalara atanması için aşağıdaki adımlar uygulanır:

1. İstenilen bölüm sayısı kadar alt dizi oluştur. Her alt dizi için toplam değerini tut (başlangıç için 0 olarak atanır).
2. Ana dizideki elemanlar azalan şekilde sıralanır.
3. Alt diziler içinde toplam değeri en küçük olan diziyi bul.
4. Ana dizideki ilk sıradaki elemanı bulunan alt diziye ekle.
5. Alt dizinin toplam değerini eklenen sayı kadar artır ve ana diziden çıkar.

6. Ana dizide eleman kalmayıncaya kadar bu işlemi tekrarla.

Örneğin Ana dizi: 50, 36, 33, 29, 28, 15, 13, 11, 9, 6, 5, 1 şeklinde güven değerlerinden oluşsun ve 3 parçaya bölünmek istendiğinde aşağıdaki gibi parçalara ayrılır. S1: 50, 15, 9, 6 toplam: 80 S2: 36, 28, 11, 5 toplam:80 S3: 33, 29, 13,1 toplam:76

5.7 Parçalardan Seçim Yöntemi

Bu yöntem ile düğümlerin parçalara atanması için aşağıdaki adımlar uygulanır:

1. Ana dizideki elemanlar güven değeri azalan şekilde sıralanır.
2. Ana dizi parça sayısı kadar alt listeye bölünür.
3. Alt listelerden sırasıyla birer eleman seçilerek her bir parçaya atama yapılır.
4. Alt listelerde atanmayan eleman kalmayıncaya kadar bu işlemi tekrarla.

Örneğin Ana dizi: 50, 36, 33, 29, 28, 15, 13, 11, 9, 6, 5, 1 şeklinde güven değerlerinden oluşsun ve 3 parçaya bölünmek istendiğinde aşağıdaki gibi parçalara ayrılır. Alt Liste 1: 50, 36, 33, 29 Alt Liste 2: 28, 15, 13, 11 Alt Liste 3: 9, 6, 5, 1 Alt listelerden rasgele birer eleman al ve parçalara ata S1: 36, 11,1 S2: 50,28,9 S3:33,13,6

5.8 Deneysel Çalışmalar

Önerilen modelin doğruluğunu test etmek için uzlaşma komitesi seçimindeki blokzincir simülasyonu ve güven değeri bulma yöntemi kullanılmıştır. Her düğüm için rasgele özellik değerleri oluşturularak adaptif hedge algoritmasıyla güven değerleri hesaplanmıştır. Farklı parça sayısı, düğüm sayısı ve hatalı düğüm sayısı kombinasyonlarına göre parçalara atama yapılmıştır. Başlangıçta bazı düğümler hatalı olarak işaretlenmiştir. Parçalara atama yapılırken güven değeri kullanılmıştır. Atama sonucunda hatalı düğüm sayısının toplam düğüm

sayısına oranı bulunmuştur. Parça dağılımının varyansı başlangıçtaki hatalı düğüm oranı ortalamasına göre hesaplanarak grafikler oluşturulmuştur. Elastico ve Ziliqadaki PoW zorluk derecesi sonuçları alabilmek için küçük bir değer seçildi (1000). Düğümler çözümü bulduğu anda göndermektedir. Küçük *nonce* veya büyük *nonce* değeri oluşturmak için yeniden denememektedir. Elasticoda tohum değerinin doğru olduğu kabul edilmiştir.

5.8.1. Dağıtım sonrası parça bilgileri

Tablo 1 de toplam düğüm sayısı 100, hatalı düğüm sayısı 33 ve parça sayısı 3 iken düğümlerin parçalara dağılımı gösterilmektedir. Tablo 2 de ise toplam düğüm sayısı 500, hatalı düğüm sayısı 165 ve shard sayısı 5 iken düğümlerin parçalara dağılımı gösterilmektedir. Parçalardaki düğüm sayısı, hatalı düğüm sayısı, toplam güven değeri ve hatalı düğümlerin toplam düğümlere oranı verilmiştir. BFT tabanlı uzlaşma algoritması kullanan blokzincirlerde hatalı düğüm oranının 1/3 ün altında olması gerekir. Koyu renkle işaretlenen satırlar hatalı düğüm sayısı 1/3ten fazla olan parçaları göstermektedir. Önerilen yöntemlerden Toplam güven değeri en yakın yöntemi hiç bir parça bu oranı geçmemektedir.

Çizelge 5.1: Parça sayısı 3 iken düğüm dağılımları

Yöntem	shardId	Düğüm Sayısı	Zararlı Düğüm Sayısı	Toplam Güven Değeri	Zararlı Düğüm Oranı
Ziliqa	shard2	33	10	21	0,3
Ziliqa	**shard3**	**33**	**14**	**17,67**	**0,42**
Ziliqa	shard1	34	9	22,26	0,26
Elastico	shard2	25	7	16	0,28
Elastico	**shard3**	**30**	**15**	**15,28**	**0,5**
Elastico	shard1	45	11	29,65	0,24
ToplGüvYakın	shard2	33	11	20,1	0,33
ToplGüvYakın	shard3	33	11	20,01	0,33
ToplGüvYakın	shard1	34	11	20,82	0,32
ParçalardanSeçim	shard2	33	11	20,16	0,33
ParçalardanSeçim	shard3	33	11	19,98	0,33
ParçalardanSeçim	shard1	34	11	20,79	0,32

5.8.2. Düğüm sayısına göre dağılım varyansı

Şekil 5.1 ve 5.2 de toplam düğüm sayıları ve hatalı düğüm sayıları değiştirilerek parça dağılımların varyansı bulunmuştur. Varyans hesaplaması yapılırken ortalama değeri olarak

Çizelge 5.2: Shard sayısı 5 iken düğüm dağılımları

Yöntem	shardId	Düğüm Sayısı	Zararlı Düğüm Sayısı	Toplam Güven Değeri	Zararlı Düğüm Oranı
Ziliqa	**shard2**	**100**	**36**	**58,89**	**0,36**
Ziliqa	shard3	100	32	61	0,32
Ziliqa	shard4	100	29	63,3	0,29
Ziliqa	**shard5**	**100**	**35**	**60,04**	**0,35**
Ziliqa	shard1	100	33	61,2	0,33
Elastico	**shard2**	**125**	**46**	**73,33**	**0,37**
Elastico	shard3	74	24	45,12	0,32
Elastico	shard4	51	16	31,68	0,31
Elastico	**shard5**	**74**	**25**	**44,62**	**0,34**
Elastico	shard1	176	54	109,7	0,31
ToplGüvYakın	shard2	100	33	61,04	0,33
ToplGüvYakın	shard3	100	33	60,95	0,33
ToplGüvYakın	shard4	100	33	60,89	0,33
ToplGüvYakın	shard5	100	33	60,82	0,33
ToplGüvYakın	shard1	100	33	60,74	0,33
ParçalardanSeçim	shard2	100	31	62	0,31
ParçalardanSeçim	shard3	100	32	61,53	0,32
ParçalardanSeçim	**shard4**	**100**	**36**	**59,46**	**0,36**
ParçalardanSeçim	**shard5**	**100**	**34**	**60,1**	**0,34**
ParçalardanSeçim	shard1	100	32	61,34	0,32

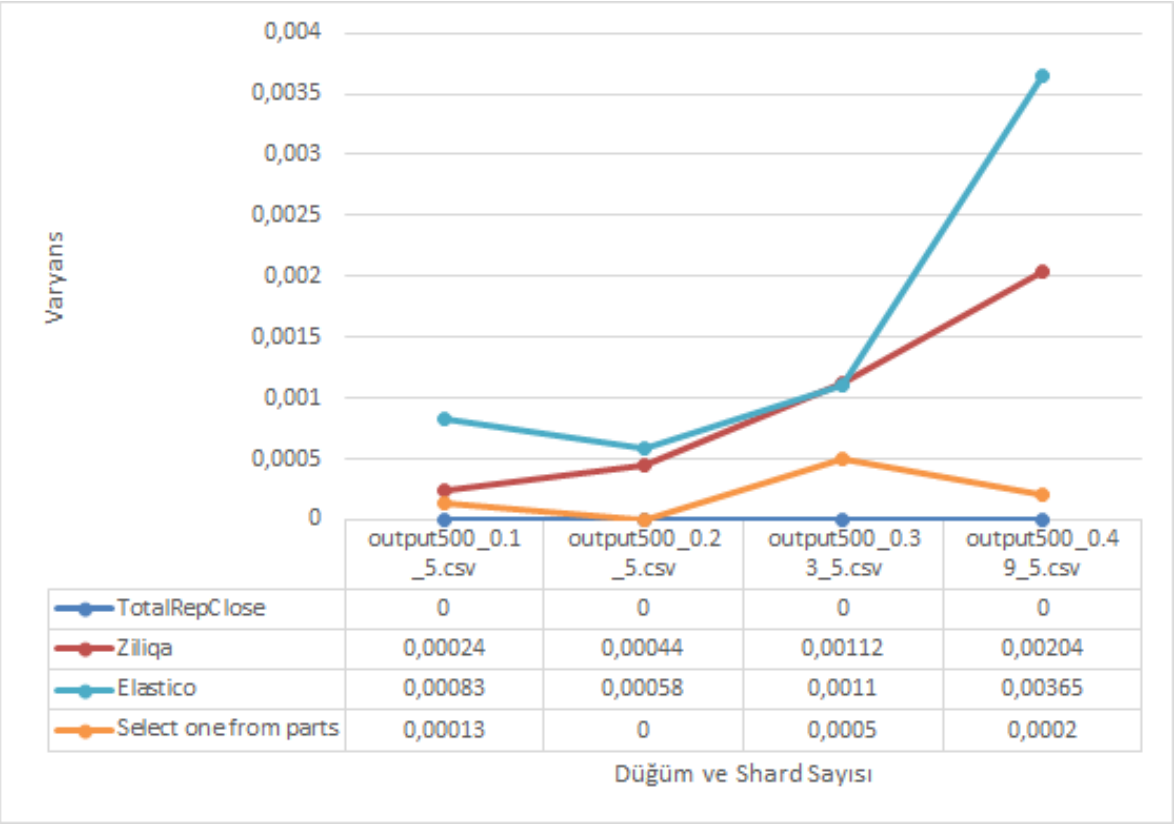
başlangıçtaki hatalı düğümün toplam düğüm sayısına oranı kullanılmıştır. Önerilen yöntemlerden parçalardan seçim yönteminde rasgele seçim yöntemi kullanıldığı için dağılım trendi izlenememektedir. Diğer yöntemlerle kıyaslandığında önerilen modellerin hatalı düğümleri daha eşit oranda dağıttığını söyleyebiliriz.

5.8.3. Hatalı düğüm oranı sabitken toplam düğüm sayısının dağılım üzerindeki etkisi

Şekil 5.3 ve 5.4 te hatalı düğüm sayısının toplam düğüm sayısına oranı sabit tutularak düğüm sayıları artırılarak varyans değerleri izlenmiştir. Düğüm sayısı arttığında bütün yöntemlerin başarı oranının arttığını söyleyebiliriz. Önerilen yöntemler düğüm sayısının az olduğu durumlarda bile diğer yöntemlerden daha başarılıdır.

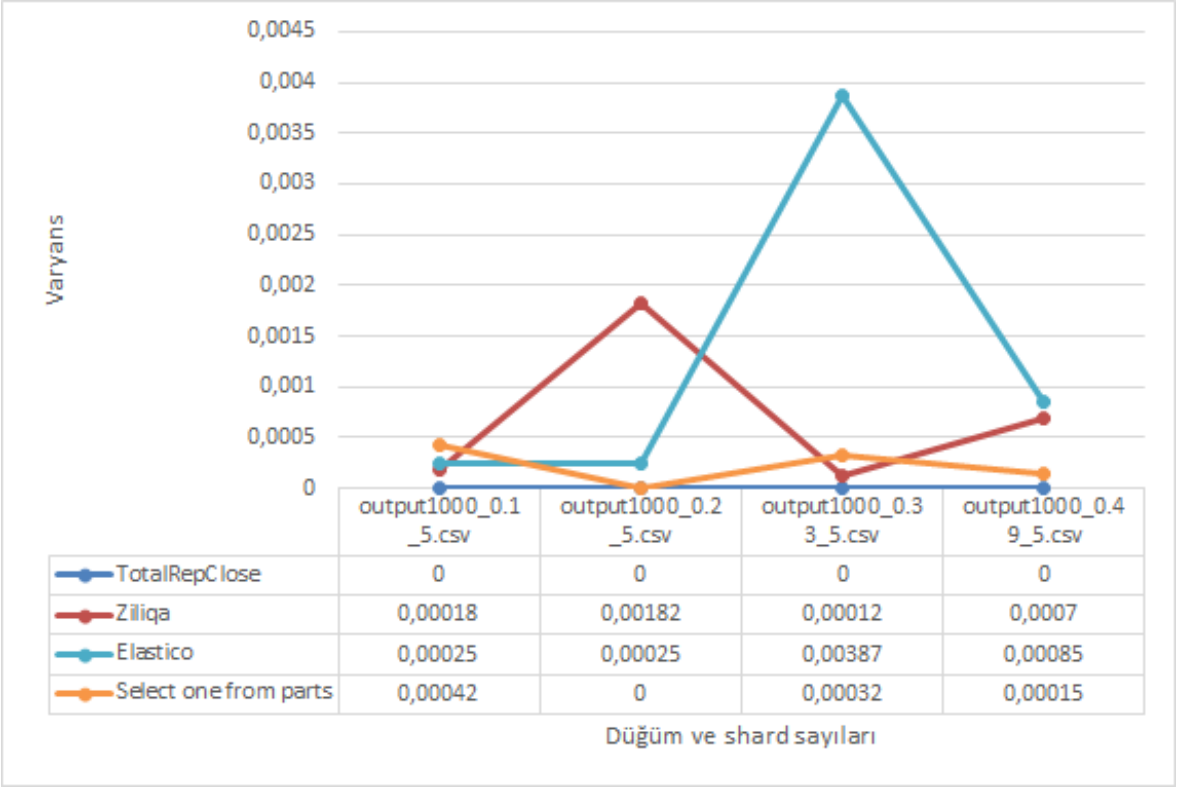
5.8.4. Parça sayısının dağılım üzerindeki etkisi

Şekil 5.5 ve 5.6 da düğüm sayısı ve hatalı düğüm sayısı sabit tutularak parça sayısının dağılımı nasıl etkilediği incelenmiştir. Önerilen yöntemlerin çok etkilenmediği diğer yöntemlerin



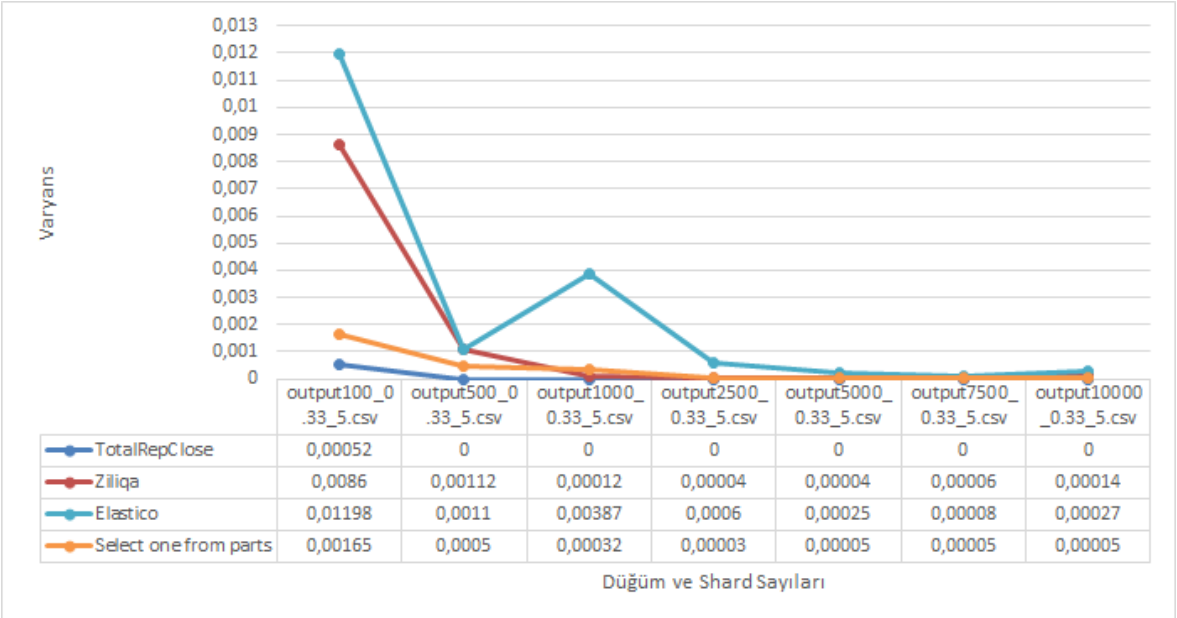
Şekil 5.1: Düğüm sayısına göre dağılım varyansı

ise başarısının düştüğü görülmektedir.



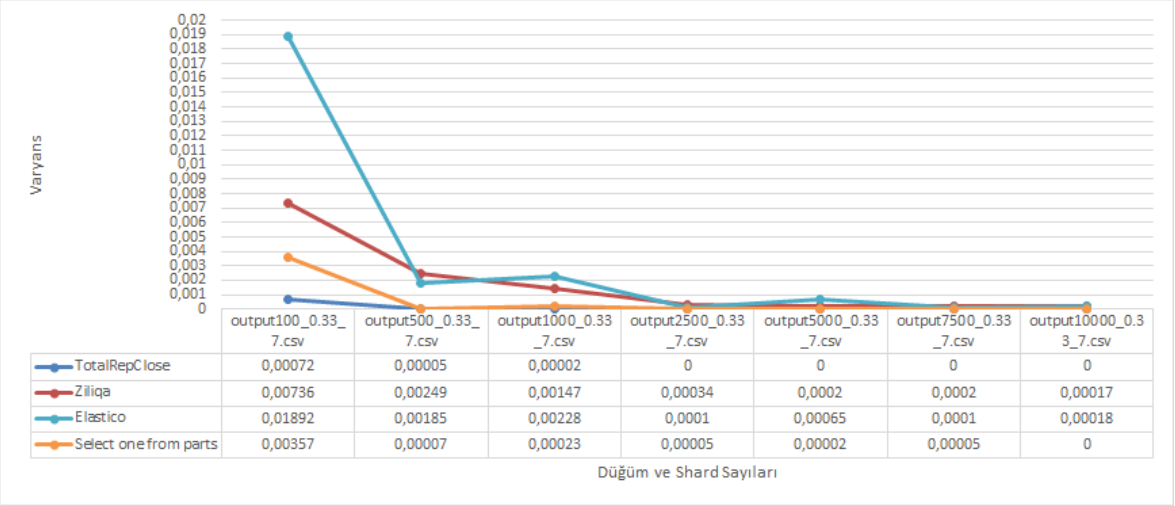
	output1000_0.1_5.csv	output1000_0.2_5.csv	output1000_0.33_5.csv	output1000_0.49_5.csv
TotalRepClose	0	0	0	0
Ziliqa	0,00018	0,00182	0,00012	0,0007
Elastico	0,00025	0,00025	0,00387	0,00085
Select one from parts	0,00042	0	0,00032	0,00015

Şekil 5.2: Düğüm sayısına göre dağılım varyansı



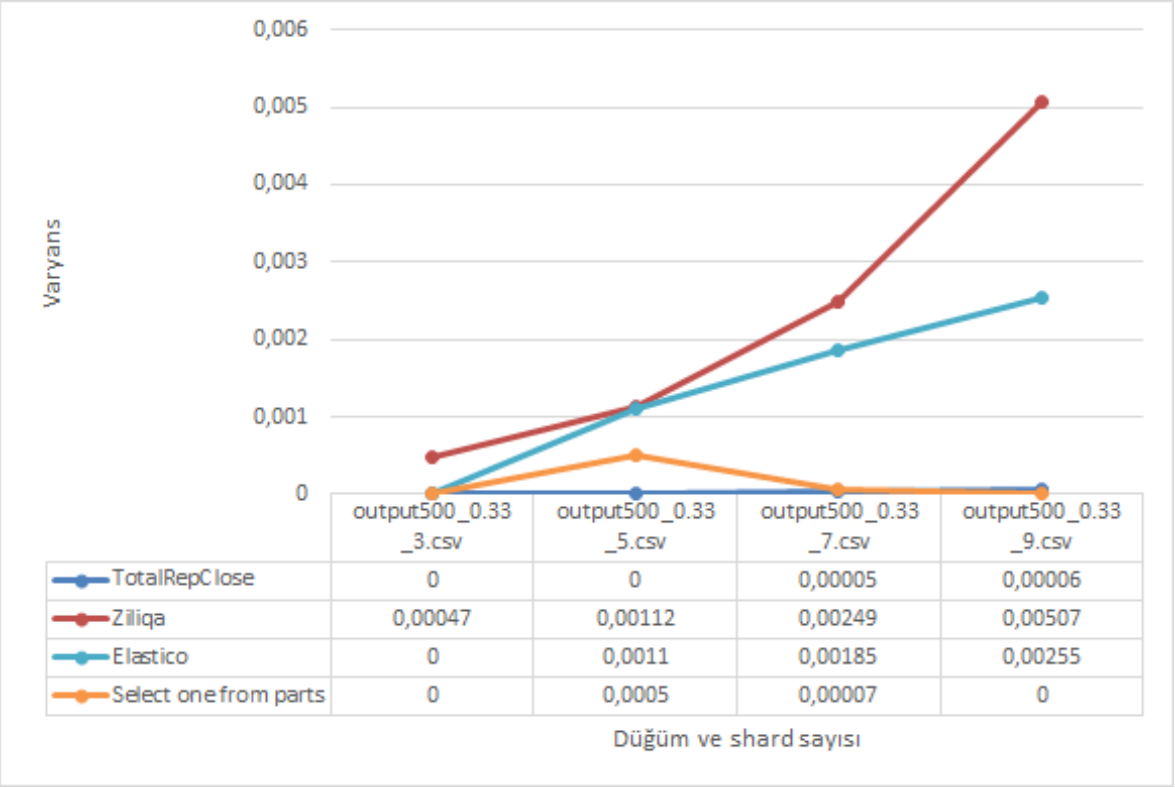
	output100_0.33_5.csv	output500_0.33_5.csv	output1000_0.33_5.csv	output2500_0.33_5.csv	output5000_0.33_5.csv	output7500_0.33_5.csv	output10000_0.33_5.csv
TotalRepClose	0,00052	0	0	0	0	0	0
Ziliqa	0,0086	0,00112	0,00012	0,00004	0,00004	0,00006	0,00014
Elastico	0,01198	0,0011	0,00387	0,0006	0,00025	0,00008	0,00027
Select one from parts	0,00165	0,0005	0,00032	0,00003	0,00005	0,00005	0,00005

Şekil 5.3: Hatalı düğüm oranı sabitken 5 parça için dağılım varyansı



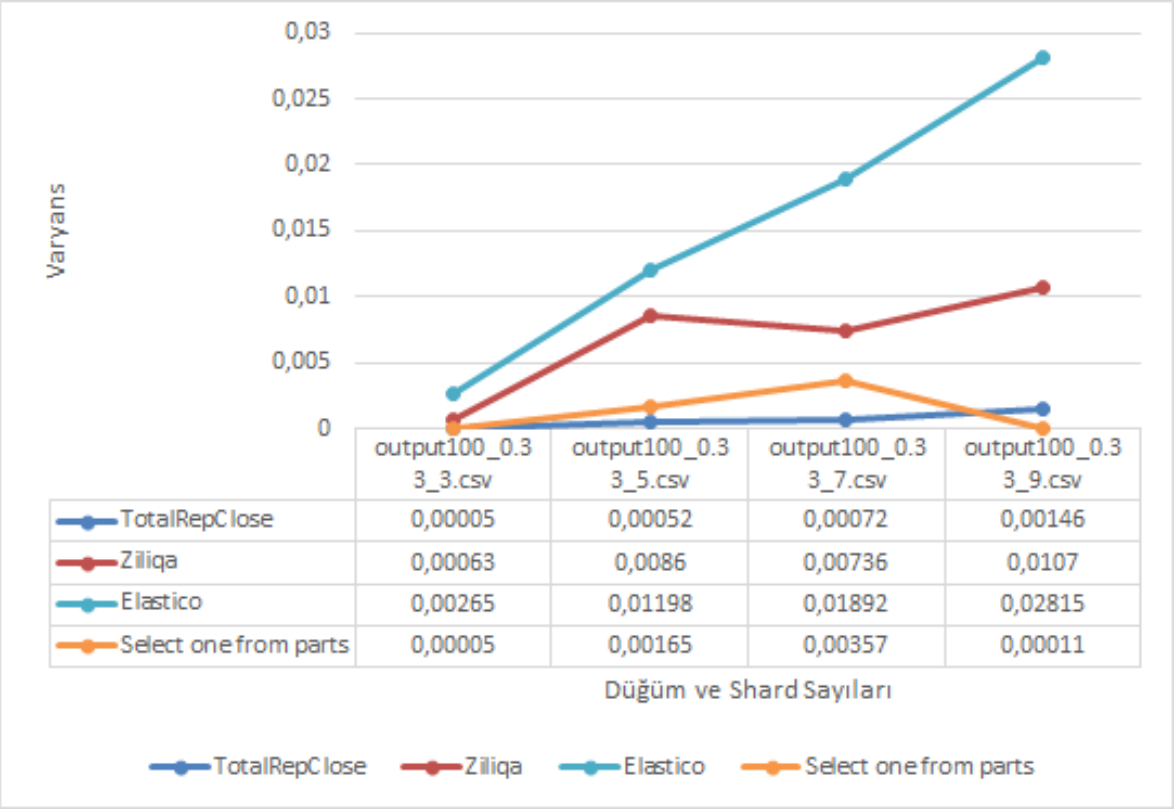
	output100_0.33_7.csv	output500_0.33_7.csv	output1000_0.33_7.csv	output2500_0.33_7.csv	output5000_0.33_7.csv	output7500_0.33_7.csv	output10000_0.33_7.csv
TotalRepClose	0,00072	0,00005	0,00002	0	0	0	0
Ziliqa	0,00736	0,00249	0,00147	0,00034	0,0002	0,0002	0,00017
Elastico	0,01892	0,00185	0,00228	0,0001	0,00065	0,0001	0,00018
Select one from parts	0,00357	0,00007	0,00023	0,00005	0,00002	0,00005	0

Şekil 5.4: Hatalı düğüm oranı sabitken 7 parça için dağılım varyansı



	output500_0.33_3.csv	output500_0.33_5.csv	output500_0.33_7.csv	output500_0.33_9.csv
TotalRepClose	0	0	0,00005	0,00006
Ziliqa	0,00047	0,00112	0,00249	0,00507
Elastico	0	0,0011	0,00185	0,00255
Select one from parts	0	0,0005	0,00007	0

Şekil 5.5: Hatalı ve toplam düğüm sayısı sabitken (sırasıyla 165, 500) parça sayısının dağılım üzerindeki etkisi



	output100_0.33_3.csv	output100_0.33_5.csv	output100_0.33_7.csv	output100_0.33_9.csv
TotalRepClose	0,00005	0,00052	0,00072	0,00146
Ziliqa	0,00063	0,0086	0,00736	0,0107
Elastico	0,00265	0,01198	0,01892	0,02815
Select one from parts	0,00005	0,00165	0,00357	0,00011

Şekil 5.6: Hatalı ve toplam düğüm sayısı sabitken (sırasıyla 33, 100) parça sayısının dağılım üzerindeki etkisi

6. SONUÇ VE SONRAKİ ÇALIŞMALAR

Blokzincir teknolojisi güvenilir bir taraf olmadan birbirine güvenmeyen kişiler arasında işlemlerin gerçekleşmesine izin veren bir teknolojidir. Bu özelliğinden dolayı farklı alanlarda çalışan kişilerin ilgisini çekmiştir. Farklı alanlara hakim diğer ürünlerle yarışabilir hale gelmesi için ölçeklenebilirlik probleminin çözülmesi gerektiği ortaya çıkmıştır. Ölçeklenebilirlik probleminin çözümü için fikir birliği yönteminin değiştirilmesi önerilmiştir. Problemin çözümü için en çok tercih edilen yöntem BFT tabanlı yöntemler olmuştur. Fikir birliği yöntemi olarak BFT tabanlı yöntemlerin kullanılabilmesi için ağdaki düğümlerin birbiri hakkında bilgi sahibi olması ve fikir birliği komitesinin oluşturulması zorlu bir problem olarak ortaya çıkar. Yaptığımız bu çalışmada bu zorlu probleme çözüm olarak çevrimiçi öğrenme tabanlı bir model önerilmiştir. Bu modelde PoW gibi işlem maliyeti yüksek bir yöntem yerine daha az işlem maliyeti gerektiren, değişen durumlara hızlı uyum sağlayan bir öğrenme yöntemi kullanılmıştır. Bu model sayesinde ağdaki düğümlerin özellikleri diğer düğümlerle kıyaslanarak bir güven değeri hesaplanır. Güven değerine göre komiteye seçim yapılır. Bu sayede komitenin güvenirliliği artırılır. Ağdaki düğümler hakkında bilgi sahibi olunur. Önerilen modelin testlerinde güvenilir komitedeki hatalı düğüm sayısı oranı toplam düğüm sayısının %50'sini aşmadığı durumlarda güven değerinin doğru hesaplandığı görülmüştür. Aday komitedeki hatalı düğüm sayısının toplam düğüm sayısının %50'sini geçtiği durumlarda bile (değiştirilecek düğüm sayısına bağlı olarak) komiteye seçilen hatalı düğüm sayısının 0 olduğu görülmüştür.

Ölçeklenebilirlik problemine çözüm olarak önerilen diğer bir yöntem parçalama yöntemidir. Bu yöntemin kullanılabilmesi için bütün blokzincir ağının nasıl parçalara ayrılacağı problemi ortaya çıkar. Yaptığımız bu tezde blokzincir ağı parçalara ayrılırken güven değerinin kullanımı için bir yöntem önerilmiştir. Önerdiğimiz yöntem düğümlerle ilgili bir fikir oluşturduğu için daha akıllı seçimler yapılabilmektedir. Parçalardaki düğüm sayısı bütün ağa göre daha az olduğu için bozulma olasılığı daha yüksek olmaktadır. Bu nedenle parçalardaki hatalı düğüm oranı daha önemli hale gelmektedir. Bu nedenle güven değeri kullanılarak parçalardaki hatalı düğüm dağılımı dengeli bir şekilde yapılabilmektedir.

Bu çalışmada öğrenme modelinde kullanılan özellikler (hisse miktarı, cevap süresi ve cevap türü) sistemin başında belirlenir ve bu özelliklerin değerleri kullanılarak güven değeri hesaplanır. Değişen koşullara göre özelliklerin de dinamik olarak değiştirilmesi ileriye dönük çalışmaların başında gelecektir. Farklı özellikler kullanılarak sistemin değişen koşullara daha iyi uyum sağlaması sağlanabilir. Değişen özelliklerin kullanımı yönelimle sistemsel açıkların bulunmasını da zorlaştıracaktır.

Hesaplanan güven değeri sayesinde düğümler hakkında fikir sahibi olunur. Akıllı sözleşmeler yazılarak güven değerine göre düğümler için farklı işlemler yapılması sağlanabilir. Örneğin güven değeri düşük olan düğümlerin komiteye katılmasına izin verilmeyebilir veya komitedeki düğümler için verdiği cevap daha az oranla etki etmesi sağlanabilir. Ayrıca güven değeri düşük düğümlerle yapılan işlemlerde daha fazla kural işletilmesi sağlanabilir.

Bu çalışmada karar tabanlı çevrimiçi öğrenme modeli kullanılmıştır. Bu model blokzincir için uygun bir model olmakla birlikte daha farklı öğrenme yöntemleri denenebilir. Bu sayede daha güvenli bir blokzincir ağı oluşturulabilir.

Blokzincir parçalara ayrılırken parça sayısı rasgele seçilirek blokzincir parçalara ayrılmıştır. Parça sayısının belirlenmesi için optimizasyon yöntemleri kullanılarak daha anlamlı parçalar oluşturulabilir.

KAYNAKLAR

[1] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang. Hedged deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4303–4311. **2016**.

[2] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. **2008**.

[3] John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, **2002**.

[4] Markus Jakobsson and Ari Juels. Proofs of work and bread pudding protocols. In *Secure information networks*, pages 258–272. Springer, **1999**.

[5] Adam Back et al. Hashcash-a denial of service counter-measure. **2002**.

[6] Vincent Gramoli. From blockchain consensus back to byzantine consensus. *Future Generation Computer Systems*, **2017**.

[7] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, **2014**.

[8] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, **1982**.

[9] Silvio Micali, Michael Rabin, and Salil Vadhan. Verifiable random functions. In *40th annual symposium on foundations of computer science (cat. No. 99CB37039)*, pages 120–130. IEEE, **1999**.

[10] Ralph C Merkle. Protocols for public key cryptosystems. In *1980 IEEE Symposium on Security and Privacy*, pages 122–122. IEEE, **1980**.

[11] Ralph C Merkle. Method of providing digital signatures, **1982**. US Patent 4,309,569.

[12] Henri Massias, X Serret Avila, and J-J Quisquater. Design of a secure timestamping service with minimal trust requirement. In *the 20th Symposium on Information Theory in the Benelux*. Citeseer, **1999**.

[13] Stuart Haber and W Scott Stornetta. Secure names for bit-strings. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28–35. **1997**.

[14] Georg Becker. Merkle signature schemes, merkle trees and their cryptanalysis. *Ruhr-University Bochum, Tech. Rep*, **2008**.

[15] Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19, **2012**.

[16] Thomas Kerber, Aggelos Kiayias, Markulf Kohlweiss, and Vassilis Zikas. Ouroboros crypsinous: Privacy-preserving proof-of-stake. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 157–174. IEEE, **2019**.

[17] LM Goodman. Tezos: A self-amending crypto-ledger position paper. *Aug*, 3:2014, **2014**.

[18] Pavel Vasin. Blackcoin's proof-of-stake protocol v2. *URL: https://blackcoin. co/blackcoin-pos-protocol-v2-whitepaper. pdf*, 71, **2014**.

[19] Jonah Brown-Cohen, Arvind Narayanan, Alexandros Psomas, and S Matthew Weinberg. Formal barriers to longest-chain proof-of-stake protocols. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 459–473. **2019**.

[20] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186. **1999**.

[21] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The honey badger of bft protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 31–42. **2016**.

[22] Alysson Neves Bessani and Marcel Santos. Bft-smart-high-performance byzantine-faulttolerant state machine replication, **2011**.

[23] Gang Wang, Zhijie Jerry Shi, Mark Nixon, and Song Han. Sok: Sharding on blockchain. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 41–61. **2019**.

[24] Eleftherios Kokoris Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing bitcoin security and performance with strong consistency via collective signing. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 279–296. **2016**.

[25] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta, and Bryan Ford. Omniledger: A secure, scale-out, decentralized ledger via sharding. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 583–598. IEEE, **2018**.

[26] Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 17–30. ACM, **2016**.

[27] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus. *CoRR, abs/1612.02916*, **2016**.

[28] Rafael Pass and Elaine Shi. Hybrid consensus: Efficient consensus in the permissionless model. In *31st International Symposium on Distributed Computing (DISC 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, **2017**.

[29] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 51–68. ACM, **2017**.

[30] George Danezis and Sarah Meiklejohn. Centrally banked cryptocurrencies. *arXiv preprint arXiv:1505.06895*, **2015**.

[31] Ethan Buchman. *Tendermint: Byzantine fault tolerance in the age of blockchains*. Ph.D. thesis, **2016**.

[32] Marcel T Rosner and Andrew Kang. Understanding and regulating twenty-first century payment systems: The ripple case study. *Mich. L. Rev.*, 114:649, **2015**.

[33] David Mazières. The stellar consensus protocol. *A Federated Model for Internet-level Consensus. Version July*, 14, **2015**.

[34] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, **1999**.

[35] Zaiqing Nie, Yuanzhi Zhang, Ji-Rong Wen, and Wei-Ying Ma. Object-level ranking: bringing order to web objects. In *Proceedings of the 14th international conference on World Wide Web*, pages 567–574. ACM, **2005**.

[36] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651. ACM, **2003**.

[37] Li Xiong and Ling Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE transactions on Knowledge and Data Engineering*, 16(7):843–857, **2004**.

[38] Runfang Zhou and Kai Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on parallel and distributed systems*, 18(4):460–473, **2007**.

[39] Ernesto Damiani, De Capitani di Vimercati, Stefano Paraboschi, Pierangela Samarati, and Fabio Violante. A reputation-based approach for choosing reliable

resources in peer-to-peer networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 207–216. ACM, **2002**.

[40] Alex Biryukov, Daniel Feher, and Dmitry Khovratovich. Guru: Universal reputation module for distributed consensus protocols. Technical report, University of Luxembourg, **2017**.

[41] Ewa Syta, Iulia Tamas, Dylan Visher, David Isaac Wolinsky, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, and Bryan Ford. Keeping authorities" honest or bust" with decentralized witness cosigning. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 526–545. Ieee, **2016**.

[42] Maria Borge, Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, and Bryan Ford. Proof-of-personhood: Redemocratizing permissionless cryptocurrencies. In *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 23–26. IEEE, **2017**.

[43] Ewa Syta, Philipp Jovanovic, Eleftherios Kokoris Kogias, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Michael J Fischer, and Bryan Ford. Scalable bias-resistant distributed randomness. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 444–460. Ieee, **2017**.

[44] Adem Efe Gencer, Robbert van Renesse, and Emin Gün Sirer. Service-oriented sharding with aspen. *arXiv preprint arXiv:1611.06816*, **2016**.

[45] Mustafa Al-Bassam, Alberto Sonnino, Shehar Bano, Dave Hrycyszyn, and George Danezis. Chainspace: A sharded smart contracts platform. *arXiv preprint arXiv:1708.03778*, **2017**.

[46] Yousef Al-houmaily and George Samaras. *Two-Phase Commit*, pages 3204–3209. **2009**. ISBN 978-0-387-35544-3. doi:10.1007/978-1-4899-7993-3_713-2.

[47] Songze Li, Mingchao Yu, Salman Avestimehr, Sreeram Kannan, and Pramod Viswanath. Polyshard: Coded sharding achieves linearly scaling efficiency and security simultaneously. *arXiv preprint arXiv:1809.10361*, **2018**.

[48] Zhijie Ren, Kelong Cong, Taico Aerts, Bart de Jonge, Alejandro Morais, and Zekeriya Erkin. A scale-out blockchain for value transfer with spontaneous sharding. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 1–10. IEEE, **2018**.

[49] MultiVAC. Multivac sharding yellowpaper: The all-dimensional sharded blockchain, **2018**.

[50] Hung Dang, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin, and Beng Chin Ooi. Towards scaling blockchain systems via sharding. In *Proceedings of the 2019 International Conference on Management of Data*, pages 123–140. ACM, **2019**.

[51] Harmony-One. Harmony technical white paper, **2018**.

[52] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. Rapidchain: Scaling blockchain via full sharding. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 931–948. ACM, **2018**.

[53] ZILLIQA Team et al. The zilliqa technical whitepaper. *Retrieved September*, 16:2019, **2017**.

[54] Douglas R Stinson and Reto Strobl. Provably secure distributed schnorr signatures and a (t, n) threshold scheme for implicit certificates. In *Australasian Conference on Information Security and Privacy*, pages 417–434. Springer, **2001**.

[55] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. On scaling decentralized blockchains. In *International conference on financial cryptography and data security*, pages 106–125. Springer, **2016**.

[56] Kurt M Alonso. Zero to monero privacy in the blockchain: 05/02/2018 draft 0.11. 2.

[57] Daniel Palmer. Scalability debate continues as bitcoin xt proposal stalls. *CoinDesk LLC*, 11, **2016**.

[58] Marco Alberto Javarone and Craig Steven Wright. From bitcoin to bitcoin cash: a network analysis. In *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pages 77–81. ACM, **2018**.

[59] Johannes Göbel and Anthony E Krzesinski. Increased block size and bitcoin blockchain dynamics. In *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, pages 1–6. IEEE, **2017**.

[60] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. Enabling blockchain innovations with pegged sidechains. *URL: http://www. opensciencereview. com/papers/123/enablingblockchain-innovations-with-pegged-sidechains*, 72, **2014**.

[61] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, **2016**.

[62] Christian Decker and Roger Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. In *Symposium on Self-Stabilizing Systems*, pages 3–18. Springer, **2015**.

[63] Matthew Green and Ian Miers. Bolt: Anonymous payment channels for decentralized currencies. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 473–489. ACM, **2017**.

[64] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 45–59. **2016**.

[65] Yair Amir, Claudiu Danilov, Jonathan Kirsch, John Lane, Danny Dolev, Cristina Nita-Rotaru, Josh Olsen, and David Zage. Scaling byzantine fault-tolerant replication towide area networks. In *International Conference on Dependable Systems and Networks (DSN'06)*, pages 105–114. IEEE, **2006**.

[66] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, **1997**.

[67] Kamalika Chaudhuri, Yoav Freund, and Daniel J Hsu. A parameter-free hedging algorithm. In *Advances in neural information processing systems*, pages 297–305. **2009**.

[68] Gunnlaugur K Hjalmarsson, Frigrik Hreigarsson, Mohammad Hamdaqa, and Gysli Hjalmtysson. Blockchain-based e-voting system. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 983–986. IEEE, **2018**.

[69] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *2016 2nd International Conference on Open and Big Data (OBD)*, pages 25–30. IEEE, **2016**.

[70] Antorweep Chakravorty and Chunming Rong. Ushare: user controlled social media based on blockchain. In *Proceedings of the 11th international conference on ubiquitous information management and communication*, pages 1–6. **2017**.

[71] Steve Cheng, Matthias Daub, Axel Domeyer, and Martin Lundqvist. Using blockchain to improve data management in the public sector. *Retrieved May*, 18:2018, **2017**.

[72] Avi Spielman. *Blockchain: digitally rebuilding the real estate industry*. Ph.D. thesis, Massachusetts Institute of Technology, **2016**.

[73] Joshua Lind, Ittay Eyal, Peter Pietzuch, and Emin Gün Sirer. Teechan: Payment channels using trusted execution environments. *arXiv preprint arXiv:1612.07766*, **2016**.

[74] Iddo Bentov, Rafael Pass, and Elaine Shi. Snow white: Provably secure proofs of stake. *IACR Cryptology ePrint Archive*, 2016(919), **2016**.

[75] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, **2017**.

[76] Bernardo David, Peter Gaži, Aggelos Kiayias, and Alexander Russell. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 66–98. Springer, **2018**.

[77] Rafael Pass and Elaine Shi. Fruitchains: A fair blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, pages 315–324. **2017**.

[78] Sunoo Park, Albert Kwon, Georg Fuchsbauer, Peter Gaži, Joël Alwen, and Krzysztof Pietrzak. Spacemint: A cryptocurrency based on proofs of space. In *International Conference on Financial Cryptography and Data Security*, pages 480–499. Springer, **2018**.

[79] Iddo Bentov, Pavel Hubácek, Tal Moran, and Asaf Nadler. Tortoise and hares consensus: the meshcash framework for incentive-compatible, scalable cryptocurrencies. *IACR Cryptology ePrint Archive*, 2017:300, **2017**.

[80] Jiangshan Yu, David Kozhaya, Jeremie Decouchant, and Paulo Esteves-Verissimo. Repucoin: Your reputation is your power. *IEEE Transactions on Computers*, 68(8):1225–1237, **2019**.

[81] Christian Cachin et al. Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers*, volume 310, page 4. **2016**.

[82] George Samaras, Kathryn Britton, Andrew Citron, and C Mohan. Two-phase commit optimizations in a commercial distributed environment. *Distributed and Parallel Databases*, 3(4):325–360, **1995**.

[83] Sang H Son and Rasikan David. Design and analysis of a secure two-phase locking protocol. In *Proceedings Eighteenth Annual International Computer Software and Applications Conference (COMPSAC 94)*, pages 374–379. IEEE, **1994**.

[84] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In *Annual international cryptology conference*, pages 757–788. Springer, **2018**.

[85] Markus Stadler. Publicly verifiable secret sharing. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 190–199. Springer, **1996**.

[86] Yong Yuan, X Ni, Shuai Zeng, and F Wang. Blockchain consensus algorithms: the state of the art and future trends. *Acta Automatica Sinica*, 44(11):2011–2022, **2008**.

[87] Christian Cachin, Simon Schubert, and Marko Vukolić. Non-determinism in byzantine fault-tolerant replication. *arXiv preprint arXiv:1603.07351*, **2016**.

[88] Elad Elrom. Neo blockchain and smart contracts. In *The Blockchain Developer*, pages 257–298. Springer, **2019**.

[89] Ian Grigg. Eos-an introduction. *White paper. https://whitepaperdatabase. com/eos-whitepaper*, **2017**.

[90] Yin Yang. Linbft: Linear-communication byzantine fault tolerance for public blockchains. *arXiv preprint arXiv:1807.01829*, **2018**.

[91] Richard Earl Korf. Multi-way number partitioning. In *Twenty-First International Joint Conference on Artificial Intelligence*. **2009**.