T.C. İSTANBUL TİCARET ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SIRADAKİ HÜCRE TAHMİNİ İÇİN DÖNGÜ TABANLI MARKOV ZİNCİRİ

Bahadır YALIN

Danışman
Prof. Dr. Serhan YARKAN

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ ANABİLİM DALI
İSTANBUL - 2022

KABUL VE ONAY SAYFASI

Bahadır YALIN tarafından hazırlanan **"Sıradaki Hücre Tahmini için Döngü Tabanlı Markov Zinciri"** adlı tez çalışması 05/09/2022 tarihinde aşağıdaki jüri üyeleri önünde başarı ile savunularak, İstanbul Ticaret Üniversitesi Fen Bilimleri Enstitüsü **Elektronik ve Haberleşme Mühendisliği Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman	**Prof. Dr. Serhan YARKAN** İstanbul Ticaret Üniversitesi	.....
Jüri Üyesi	**Prof. Dr. Fatih ÜSTÜNER** İstanbul Ticaret Üniversitesi	.....
Jüri Üyesi	**Dr. Öğr. Üyesi Fatih KELEŞ** İstanbul Üniversitesi-Cerrahpaşa	.....

Onay Tarihi : 28.09.2022

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsünün 28.09.2022 tarih ve 2022/359 numaralı Yönetim Kurulu Kararının 16. maddesi gereğince, ders yüklerini ve tez yükümlülüğünü yerine getirdiği belirlenen "Bahadır YALIN" adlı öğrencinin mezun olmasına oy birliği ile karar verilmiştir.

Prof. Dr. Doğan KAYA
Enstitü Müdürü

AKADEMİK VE ETİK KURALLARA UYGUNLUK BEYANI

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

28.09.2022

Bahadır YALIN

İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER	i
ÖZET	iii
ABSTRACT	iv
TEŞEKKÜR	v
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ	viii
1. GİRİŞ	1
2. LİTERATÜR ÖZETİ	6
3. AĞ VERİ ANALİTİĞİ FONKSİYONU	9
3.1. Ağ Veri Analitiği Fonksiyonunun Sağladığı Analitikler	10
4. YÖNTEM	13
4.1. Sistem Tasarımı	13
4.2. Markov Zinciri	14
4.3. Markov Zinciri ile Bir Sonraki Hücre Tahmini	16
4.4. Döngü Tabanlı Markov Zinciri	18
4.5. Sistem Algoritması ve Kaynak Kodları	21
5. HESAPLAMA SONUÇLARI	27
6. SONUÇ VE ÖNERİLER	31
KAYNAKLAR	32
ÖZGEÇMİŞ	34

ÖZET

Yüksek Lisans Tezi

SIRADAKİ HÜCRE TAHMİNİ İÇİN DÖNGÜ TABANLI MARKOV ZİNCİRİ

Bahadır YALIN

İstanbul Ticaret Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Serhan YARKAN

2022, 34 sayfa

Bu çalışmada Markov Zinciri yeni bir döngü tabanlı yaklaşımla Ağ Veri Analitiği Fonksiyonu (NWDAF) için kullanılarak bir Kullanıcı Ekipmanı (UE) için bir sonraki olası hücre ya da hücre sekansı tahmin edilmektedir. Başlangıçta, bu yeni yaklaşım ile kullanıcının bir sonraki durumu için tahmininin daha anlamlı olacağı varsayımı yapılmıştır. Çalışmanın sonucunda ise, bir UE'nin bulunduğu son hücreden sonra en çok ziyaret ettiği hücre arasındaki geçiş durumlarının çıkarımı yapılmış ve bu iki hücre arasında bir ilişki kurulduğunda yapılan tahminin güvenilirliğinin, döngü tabanlı Markov Zinciri yaklaşımıyla (geleneksel Markov Zinciri'ne kıyasla) arttığı gözlemlenmiştir. Ek olarak çalışma sonunda ise, üzerinde durulan yöntemin geliştirilebilmesi için kullanılan yönteme eklenebilecek özelliklerden de olarak bahsedilmiştir.

Anahtar Kelimeler: Ağ veri analitiği fonksiyonu, kullanıcı ekipmanı hareketliliği, markov zinciri, 5G ağ fonksiyonları.

ABSTRACT

M.Sc. Thesis

LOOP-BASED MARKOV CHAIN FOR NEXT CELL PREDICTION

Bahadır YALIN

İstanbul Commerce University
Graduate School of Applied and Natural Sciences
Department of Electronics and Communication Engineering

Supervisor: Prof. Dr. Serhan YARKAN

2022, 34 pages

In this study, next possible cell for a User Equipment (UE) is predicted for Network Data Analytics Function (NWDAF) by using Markov Chain with novel loop-based approach. At the beginning, it is assumed that the prediction made for next state of a UE will be more meaningful with our novel approach. As a result of the study, the transition states between the most visited cell after the last cell where a UE is located were extracted and it was observed that the reliability of the prediction made is improved (compared to traditional Markov Chain), when a semantic was established between two cells with the loop-based Markov Chain approach. In addition, at the end of the study, the development of the method emphasized and the features that can be added to the method used are mentioned.

Keywords: Markov chain, network data analytics function, user equipment mobility, 5G network functions.

TEŞEKKÜR

Bu araştırma için beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan değerli Danışman Hocam Prof. Dr. Serhan YARKAN teşekkürlerimi sunarım. Literatür araştırmalarımda yardımcı olan değerli hocam Prof. Dr. Serhan YARKAN'a, teşekkür ederim.

Tezimin her aşamasında beni yalnız bırakmayan aileme sonsuz sevgi ve saygılarımı sunarım.

Bahadır YALIN
İSTANBUL, 2022

ŞEKİLLER

Sayfa

Şekil 1.1. Herhangi bir NF'nin NWDAF'a gönderdiği analitik talep..... 2
Şekil 1.2. SBA içerisindeki SBI Mesaj Yolu 2
Şekil 1.3. NWDAF'a gelen analitik talep üzerine NWDAF'ın ilgili NF'den veri toplaması..... 3
Şekil 1.4. UE'nin farklı hücreler arasındaki hareketliliği..... 4
Şekil 4.1. Sistemin akış diyagramı..... 21
Şekil 5.1. 3 farklı veri kümesinin sıçrama döngüsü dağılımları..... 29
Şekil 5.2. 1 ve 2 sıçramalı tahminin doğrulukları..... 30

ÇİZELGELER

Sayfa

Çizelge 3.1. NWDAF tarafından sağlanan analitik örnek listesi ve tanımları 11
Çizelge 4.1. Son hücre için durum geçiş matrisi 17
Çizelge 4.2. Örnek UE hareketlilik geçmişi verisi 19

SİMGELER VE KISALTMALAR

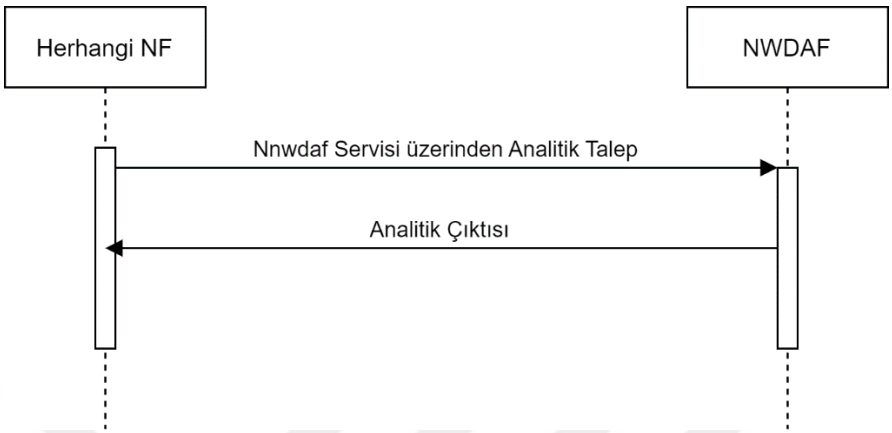
ADRF	Analitik Veri Depolama Fonksiyonu (Analytics Data Repository Function)
AI	Yapay Zeka (Artificial Intelligence)
AMF	Erişim ve Hareketlilik Fonksiyonu (Access and Mobility Function)
ANN	Yapay Sinir Ağları (Artificial Neural Network)
API	Uygulama Programlama Arayüzü (Application Programming Interface)
Bi-LSTM-Attention	Dikkate Dayalı Çift Yönlü Uzun Kısa Süreli-Bellek (Attention-Based Bidirectional Long Short-Term Memory)
HMM	Gizli Markov Modeli (Hidden Markov Model)
LSTM	Uzun Kısa-Süreli Bellek (Long Short-Term Memory)
LTE	Uzun Süreli Gelişim (Long Term Evolution)
ML	Makine Öğrenmesi (Machine Learning)
NF	Ağ Fonksiyonu (Network Function)
NEF	Ağ Açığa Çıkarma Fonksiyonu (Network Exposure Function)
NWDAF	Ağ Veri Analitiği Fonksiyonu (Network Data Analytics Function)
NWDAF (AnLF)	Mantık Fonksiyonu içeren NWDAF (NWDAF containing Analytics Logical Function)
NWDAF(MTLF)	Model Eğiten Mantık Fonksiyonu içeren NWDAF (NWDAF containing Model Training Logical Function)
PDU	Protokol Veri Birimi (Protocol Data Unit)
PLMN	Kamusal Mobil Ağ (Public Land Mobile Network)
RoI	İlgi Bölgesi (Region of Interest)
QoS	Hizmet Kalitesi
SBA	Servis-Temelli Mimari (Service-Based Architecture)
SBI	Servis-Temelli Arayüz (Service-Based Interface)
SMF	Oturum Yönetimi Fonksiyonu (Session Management Function)
UE	Kullanıcı Ekipmanı (User Equipment)
VNF	Sanallaştırılmış Ağ Fonksiyonu (Virtualized Network Function)
5G	5. Jenerasyon
3GPP	3. Nesil Ortaklık Projesi (3rd Generation Partnership Project)

1. GİRİŞ

Devreye alınan ve geliştirilmeye başlanan 5. Jenerasyon (5G) ile eski sistemlere göre birçok şey değişmektedir. Bu değişimlerden biri olan Sanallaştırılmış Ağ Fonksiyonu (VNF) yeniliği sayesinde, şebekede alınacak geliştirme ve değişiklik gibi aksiyonlar artık çok fazla plan ve maliyet gerektiren donanımsal değişiklikler yerine yazılımsal değişikliklerle sağlanacak, çünkü 5G'deki Ağ Fonksiyonları (NFler) birbirlerine bulut-temelli haberleşme üzerinden bağlanıyor olacak. Bir merkezden veri alınan önceki teknolojilere kıyasla, 5G içerisindeki operasyon-temelli olarak görevlendirilmiş fonksiyonlar birbiri arasında veri paylaşabilecek. Özellikle verilerin kullanım amacı doğrultusunda toplanması, toplanan verinin daha anlamlı ve daha az maliyetli olmasını sağlayacak. Veriden bahsedilirken akla gelmesi gereken hususlardan bir tanesi ise disk kullanım problemidir. Bunun için, bilginin veya verinin bir birimde (bir NF'de) ne kadar süre ikamet etmesi ve ne zaman kullanım dışı kalması gerektiği operasyon öncesinde planlanmış olmalıdır. Aksi durumda, mevcut operasyon için kullanılmayacak verinin barınıyor olması optimize edilmemiş senaryoyu ortaya çıkartır. 5G, doğası gereği birçok alanı kapsar ve önceki nesillerde benzeri görülmeyen bir dönüşüm gerektirir. Bu bağlamda, toplanan verinin analitik haline getirilmesi, operasyonla ilgili alakasız ve düzensiz verinin elimine edilmesi için mihenk taşı görevi görmektedir. Veri Analitiği, 5G özelliklerini gerçeğe dönüştürürken öngörülen zorlu durumlarla başa çıkmak için 5G haberleşmesi için güçlü bir araç olarak görülmektedir (Pateromichelakis vd., 2019). Bu veri analitiğini sağlamak için 3. Jenerasyon Ortaklık Projesi (3GPP) tarafından 5G için Ağ Veri Analitiği Fonksiyonu (NWDAF) tanımlanmıştır (3GPP, 2022). NWDAF, merkezi olarak verileri diğer Servis-Temelli Mimari'de (SBA) tanımlı NFlerden (Erişim ve Hareketlilik Fonksiyonu (AMF), Oturum Yönetimi Fonksiyonu (SMF) gibi) toplayarak, bu verileri istatistik veya tahmin temelli analitikler haline getirir ve bu analitikleri kendisinden analitik talep eden NF ile, senkron veya asenkron, şekilde paylaşır.

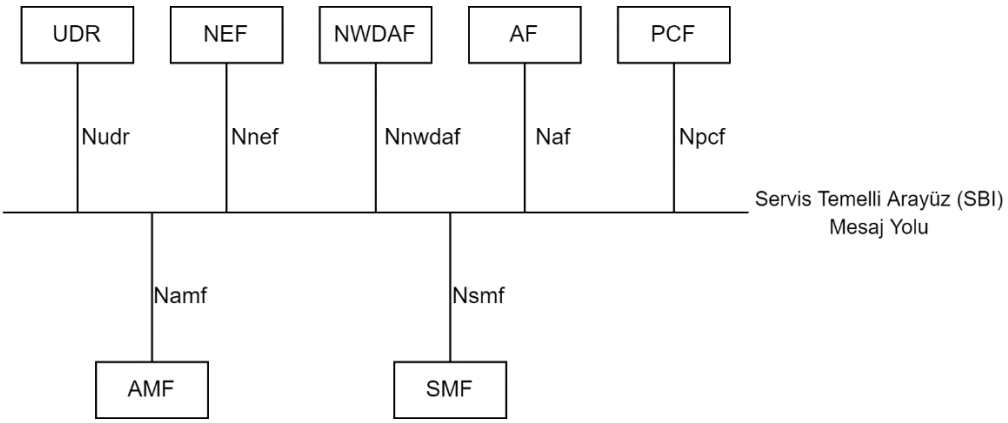
Burada asenkron NF'nin NWDAF'a abonelik talebi göndererek, belirttiği parametrelere göre analitikleri istemesi durumuyken, senkron direkt olarak talep/yanıt şeklinde tanımlanır. Bu talep (veya diğer NFlerden farklı talepler) için

3GPP'nin tanımladığı ve her fonksiyonun sahip olduğu bir Servis-Temelli Arayüz (SBI) bulunmaktadır. Şekil 1.1'de NWDAF'a bir NF'nin gönderdiği analitik talebin temsili görülebilir.



Şekil 1.1. Herhangi bir NF'nin NWDAF'a gönderdiği analitik talep

Nnwdaf, SBA'da NWDAF'a diğer NFlerin ulaşabilmesi için tanımlanan, NWDAF'a ait SBI'dır. Bunun gibi 5G içerisindeki her fonksiyonun kendine ait SBIları bulunmaktadır ve basitçe hepsi Nnf olarak isimlendirilir (Namf, Nsmf, Npcf vb.). Bir NF başka bir NF ile iletişime geçebilmek (veri talep edebilmek) için, ilgili NF'nin servisini çağırmalıdır. İletişime girmek isteyen NF servis çağırma işlemi için, o servisin Uygulama Programlama Arayüzü'nü (API) kullanılmalıdır. Daha önce de belirtildiği gibi her bir NF'nin kendine ait SBI'sı mevcuttur ve bu SBIların bir arada bulunduğu mimari olan SBA içerisinde NFlerin birbiriyle etkileşime girebildiği yapıya ise, Şekil 1.2'de görüldüğü gibi, SBI Mesaj Yolu ismi verilir.



Şekil 1.2. SBA içerisindeki SBI Mesaj Yolu

Bu Mesaj Yolu ile herhangi bir NF, diğeriyle etkileşime girebilir. Yani herhangi bir NF (NWDAF'dan başka) NWDAF'a analitik talebi gönderebilir veya bu analitikler için abonelik talep edebilir. NWDAF için 3GPP tarafından tanımlanmış birden fazla kullanım senaryosu mevcuttur ve bunlardan bazıları Ağ Dilimi Yük Seviyesi, Gözlemlenen Servis Deneyimi, NF Yükü, Ağ Performansı, Kullanıcı Veri Tıkanıklığı, Hizmet Kalitesinin (QoS) Sürdürülebilirliği ve Kullanıcı Ekipmanı (UE) Hareketlilik Analitikleri olarak bilinir (3GPP, 2022). NWDAF her bir analitik için, analitiği oluşturmakta gerekli olan veriyi toplamak adına, verinin ilgili olduğu olayı gözlemleyen NF ile o NF'nin servisi üzerinden, etkileşime geçmelidir. NWDAF'tan analitik talep ederken senkron durum söz konusuyken, diğer NFlere gönderilecek taleplerin tamamı abonelik talebi olmak durumundadır. Bir NWDAF Hizmet Tüketicisi'nin NWDAF'a gönderdiği analitik talebin ve NWDAF'ın bu talep üzerine ilgili NF'den veri toplamasının akışı, Şekil 1.3'teki gibidir.



Şekil 1.3. NWDAF'a gelen analitik talep üzerine NWDAF'ın veri toplaması

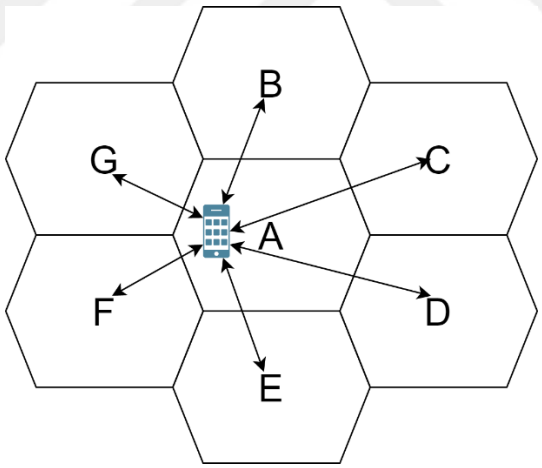
Burada 1. Adım NWDAF Hizmet Tüketicisi'nin (ya da NF Tüketici'nin) NWDAF'a gönderdiği analitik talebi temsil etmektedir. Bu adım üzerine, eğer NWDAF içerisinde hali hazırda (analitiği oluşturmak için) gerekli veriler bulunmuyorsa, ilgili NF ile (o NF'nin servis operasyonu üzerinden) etkileşime geçerek verileri abonelik yoluyla talep edecektir. Bu abonelik üzerine Olay Sağlayıcı (Veri Toplayıcı) NF, 3. Adımda görüldüğü gibi abonelik ile talep edilen ve NWDAF'tan istenen analitiği sağlamak için gerekli olan verileri NWDAF ile paylaşacaktır. Son, yani 4., adımda ise NWDAF adım 3'te topladığı verileri ve kullanım senaryosuna uygun olarak geliştirilmiş çeşitli algoritmaları kullanarak, oluşturduğu analitiği NWDAF Hizmet Tüketicisi ile paylaşacaktır.

Öncesinde de bahsedildiği gibi NWDAF, istatistik ve tahmin temelli olarak iki tipte analitik oluşturabilir. Bu analitiklerden tahmin-temelli olanı için NWDAF içerisinde Makine Öğrenmesi (ML) ve Yapay Zeka (AI) modelleri ve/veya

algoritmaları kullanılmaktadır. Bu modeller kullanılarak oluşturulacak tahmin-temelli analitik, analitiği talep eden NF'nin kendi operasyonundaki sürdürülebilirliği sağlaması için öncül olmaktadır.

Bu çalışmada NWDAF'ın AMF tarafından gözlemlenen olaylardaki verileri abonelik üzerine toplamasıyla ve bu verileri Markov Zinciri ve Döngü-Tabanlı Markov Zinciri tahmin algoritmaları kullanılmasıyla oluşturacağı UE Hareketlilik Analitiği oluşturma senaryosu üzerine bir kıyaslama yapılacaktır.

UE Hareketliliğinin tanımı, bir UE'nin giriş yapmış olduğu bölgelerin (tanecik yapısına göre hücre veya İzleme Alanı (TA) gibi) sıralı verileri anlamına gelir. Şekil 1.4'te görülen altıgensel topoloji içerisinde herhangi bir UE'nin (akıllı araç veya şekildeki mobil cihaz gibi) birbirine komşu olan farklı hücreler arasında geçiş yapmasına UE hareketliliği adı verilmektedir.



Şekil 1.4 UE'nin farklı hücreler arasındaki hareketliliği

Bu anlamda çalışma içerisinde yapılacak uygulamada, herhangi bir UE'nin sıralı hücre listesinin baz alınarak, sonraki durumlarda yapacağı hareket şemaları tahmin edilmeye çalışılacaktır.

Çalışma için kullanılacak bir AMF geliştirilmiş olmadığından veriler sentetik olarak (NWDAF tarafından AMF üzerinden UE Hareketlilik Analitiği için

topladığına denk şekilde) oluşturulacaktır. Bu analitiği sağlarken temel amaç, UE'nin geçmiş verilerini (ziyaret ettiği hücreleri) kullanarak, gelecek anda bulunacağı durumun (yani hücrenin) tahmini için istatistiksel bir yaklaşım kurarken, Markov Zinciri'nin sağlayabildiği bir sonraki durum (hücre) tahmininin iki hücre arasında bir anlambilim kurularak geliştirilmesi şeklindedir.

2. LİTERATÜR ÖZETİ

UE Hareketliliği ve sonraki konum tahmini/ön görüsü, 5G'nin kullanım senaryolarından biri olmadan önce de (önceki jenerasyonlarda) araştırmacıların farklı yöntemler sunduğu bir konudur.

Örneğin bu çalışmalardan bir tanesinde, Uzun Kısa-Süreli Bellek (LSTM), Dikkate Dayalı Çift Yönlü Uzun Kısa Süreli-Bellek (bi-LSTM-attention) ve Yapay Sinir Ağları (ANN) kullanılarak AI temelli hareketlilik tahmini Microsoft Research Asia'nın Geolife projesindeki veriler ile sunulmuştur (Wang vd., 2021). Kendi çalışmalarında ortaya koydukları deneysel yaklaşım sonucunda, başarı endeksi olarak tahmin sonucunun güvenilirliği hesaba katıldığında Bi-LSTM-Attention, eğitim süresi hesaba katıldığında ANN daha iyi sonuç vermiştir.

Bir başka makalede (Duong ve Tran, 2015) ise yazarlar çalışmalarında, ML modellerin kullanıcıların sıralı şablonlarını (bulundukları bölgeleri) kullanarak onların hareketlilik profilini oluşturmasının, operasyona yeni bir kullanıcı geldiğinde problemli olduğunu ve benzer modellerin farklı kullanıcılar için kullanılmasının sonucunda olağan kullanıcı hareketlerinin bile rastgele şekilde algılanabileceğinden ötürü veri madenciliği yaklaşımının daha iyi olduğunu sunmuşlardır. Çalışmalarında kullanıcıların bireysel hareket geçmişlerini kullanmak yerine, benzer hareket eden kullanıcıların hareket geçmişini bir çatı altında toplayarak bir sonraki adımı ön görmeyi amaçlamışlardır.

Bir başka çalışmada ise, Uzun Süreli Gelişim (LTE) hücresel ağında insan hareketliliği esas alınarak, Gizli Markov Modeli (HMM) yönetimi yardımıyla, bireysel kullanıcının Gelişmiş B Düğümü (eNodeB) sekansı ile hareketlilik tahmini yapılmıştır (Lv vd., 2014). Güney Çin'de 2800 farklı kullanıcının hareket verisi kullanılarak HMM ile %53 tahmin doğruluğuna ulaşılmıştır.

Bir başka çalışmada (Anagnostopoulos vd., 2011) ise yazarlar lokasyon tahminine denetimli öğrenme üzerinden bir sınıflandırma yaklaşımını sentetik veriler ve gerçek-dünya verileriyle denemişlerdir. Hücresel model kullanılan bu çalışmada temel amaç, bir kullanıcı için sıralı olarak kullanılan geçmiş hücre verisi ile kullanıcının bir sonraki zamanda hangi hücre içerisinde olabileceğinin tahmin edilmesidir. Çalışmalarında 10, 20, 30 ve 40 günlük verilerle modellerini eğitmişlerdir ve deneysel olarak alınan sonuca göre 40 günlük eğitim verisinin diğerlerinden daha iyi sonuçlar verdiğini, 40'tan fazla gün hesaba katıldığında da sonuç her ne kadar daha iyi de olsa artacak karmaşıklığın verimliliği düşüreceğini sunmuşlardır.

Farklı bir çalışmada ise yazarlar insanların hareketlilik davranışlarının rastgele olmaktan çok uzak (bir başka deyişle Gaussal-olmayan) olduğu varsayımıyla oluşturulan hareketlilik verilerini Markov-temelli yaklaşım kullanarak modellemişlerdir ve bu model ile insanların sonraki hareketlerini tahmin etmeyi amaçlamışlardır (Qiao vd., 2018). Çalışmalarında kullandıkları 21 günlük 3474 farklı kullanıcının, Gaussal-olmayan verilerini, oluşturdukları model ile test ettiklerinde %56 doğruluk sonucuna ulaşmışlardır.

Bir çalışmada ise yazarlar, çalışmaları için en çok ziyaret edilen yerlerin çıkarımını yaparak, bir sonraki konumu tahmin etmek için belirli bir süre boyunca kullanıcının hareketlilik davranışını izlemişlerdir (Killijian vd., 2012). Çalışmalarında, tahmin için yalnızca mevcut durumu kullanan Markov Özelliği'nin ötesine geçmeyi amaçlamışlardır. Buna göre, kullandıkları model yalnızca, Markov'un Özelliği'nde olduğu gibi, bilinen son konumu değil, N tarihsel bilgiyi de kapsar. Deneysel yaklaşımları sonucunda kullandıkları algoritmanın doğruluk oranının %75-95 aralığında olduğunu bildirmişlerdir. Ayrıca, $N > 2$ durumu için algoritmanın hesaplama maliyetinin, verdiği doğrulukla karşılaştırıldığında optimize sonuç vermediğini saptamışlardır.

Bazı çalışmalarda, kullanıcının hareketliliği tahmin edilirken İlgi Bölgesi (RoI) de dikkate alınmıştır (Li vd., 2008), (Zhang vd., 2009). Kısaca, bir kullanıcının geçmişte bulunduğu konumlar arasında daha sık ortaya çıkanı (ev gibi) ve daha

nadir ortaya çıkanı (alışveriş merkezi gibi) konumlar RoI olarak adlandırılır. Bu yöntemi kullanan çalışmalardan birinde, Markov Zinciri ile kullanıcı hareketliliğini tahmin etmek için RoI puanları hiyerarşik bir sırayla kullanılmıştır (Pan vd., 2016). Yazarlar yaptıkları çalışmada, tarihsel verileri olan bölgelerin birbirine yakınlıklarına göre aynı RoI'ye veya farklı RoIlere ait olduğu sonucuna varmışlardır. Bu çıkarımdan sonra Markov Zinciri ile bir sonraki konumu tahmin etmeye çalışmışlardır. Aynı zamanda geçmişe ait K durumlarını kullanarak bir sonraki konumu tahmin ederek Markov Özelliği kavramını genişletmeyi de amaçlamışlardır. Her iki Markov Zinciri yöntemi için sonuç değerlendirmelerini entropi ile test ettiklerinde, sistemleri için K ön verilerini kullanan Markov zincir yönteminin hareketlilik tahminindeki belirsizliği azalttığı sonucuna varmışlardır.

3. AĞ VERİ ANALİTİĞİ FONKSİYONU

NWDAF, önceki bölümlerde de bahsedildiği gibi SBA'nın bir parçasıdır ve farklı NFler ile farklı amaçlar uğruna etkileşime girebilir (3GPP, 2022). Bu amaçlar şu şekildedir:

- Farklı olaylar için o olayı izlemekten ve veri toplamaktan sorumlu olan NF ile,
- Veri Depolarının (Analitik Veri Depolama Fonksiyonu (ADRF) gibi) devreye alındığı durumda, analitik oluşturmakta gerekli veriyi toplamak için bu depolarla;
- NF ile ilgili bilgileri toplamak için Ağ Açığa Çıkarma Fonksiyonu (NEF) ile,
- Kendisinden analitik talebinde bulunan NWDAF Tüketicileri ile, oluşturduğu analitikleri paylaşmak üzere etkileşime girebilir.

3GPP'nin standartlarını oluşturduğu NWDAF'ın temel olarak iki farklı tipi bulunmaktadır (3GPP, 2022). Bunlardan ilki, bir tüketici NF'nin analitik talepte bulunmasını mümkün kılan Analitik Mantık Fonksiyonu içeren NWDAF (NWDAF(AnLF)), bir diğeri ise operasyon temelli olarak ML modellerini eğiten ve analitikler üzerinde uygulayan ve yalnızca NWDAF(AnLF) ile etkileşime geçebilen bir Model Eğiten Mantık Fonksiyonu içeren NWDAF (NWDAF(MTLF)) olmaktadır.

NWDAF(AnLF), kendisine gelen bir analitik talep üzerine NWDAF(MTLF) ile Nnwdaf servisi üzerinden etkileşime geçerek oluşturulacak analitik için uygun modeli talep edebilir. Burada belirtilen talep içerisinde istenen analitiğin bilgisi (yani Analitik Tanımlayıcısı ya da Analitik ID), analitiğe karşılık gelen modelin gönderilmesi için gereklidir.

Devreye alınmış bir NF ile NWDAF'ın etkileşime girebilmesi için, aynı Kamusal Mobil Ağ (PLMN) içerisinde bulunması gerekmektedir (3GPP, 2022). Bir NF, NWDAF'ın sağlayabildiği bir analitik bilgisini, analitik/analitikler için gerekli parametreleri belirterek, NWDAF'a HTTP POST ya da HTTP GET talebi ile gönderir. HTTP POST yöntemi ile gönderilen talepler 'abonelik/bildirim' mekanizması üzerine olurken, HTTP GET ile gönderilen talepler tek seferlik

‘istek/yanıt’ mekanizması üzerinedir. Burada abonelik talebi için gönderilen HTTP POST isteği üzerine NWDAF, talebi gönderen NF Tüketici için bir abonelik oluşturur ve bu aboneliğe benzersiz bir değer (Abonelik Korelasyon Tanımlayıcısı) atayarak, sonraki operasyonlar için (güncelleme, silme) bu değeri NF Tüketici ile paylaşır (3GPP, 2022).

Her iki HTTP yöntemlerinden biriyle gönderilen analitik talep üzerine NWDAF, analitik oluşturmak için gerekli verileri hali hazırda bulundurmuyorsa, öncelikle bu verilerin olaylarını izlemekle sorumlu olan NF’ye (NF Sağlayıcı) verinin olay bilgisini içeren HTTP POST talebi gönderir. Bu talep karşısında NF verileri NWDAF’a yine HTTP POST yöntemi ile gönderir. NWDAF, elde ettiği bu verilerle analitiği oluşturduktan sonra, kendisine abone olan NF’ye HTTP POST yöntemi ile en başta talep edilen analitiği/analitikleri gönderir. Burada, NWDAF tarafından NF Sağlayıcı’ya gönderilen, ilk HTTP POST talebi NF Sağlayıcı içerisinde NWDAF için bir abonelik oluşturma hedefiyle gönderilirken, NF Sağlayıcı’nın gönderdiği HTTP POST talebi, toplanan verilerin NWDAF ile paylaşılması adına kullanılmaktadır.

3.1. Ağ Veri Analitiği Fonksiyonunun Sağladığı Analitikler

Önceki bölümlerde de belirtildiği gibi, NWDAF, farklı operasyon gereklilikleri için farklı analitikler sağlayabilir. Bu analitiklerin, örnek olarak, birkaç tanesi ID karşılıkları ve tanımlarıyla birlikte Çizelge 3.1’deki gibidir.

Çizelge 3.1. NWDAF tarafından sağlanan örnek analitik listesi ve tanımları

Analitik Bilgi	**Analitik Tanımlayıcı**	**Tanım**
Dilim yük seviyesi bilgisi	dilim yük bilgisi	Bu bilgi, bir Ağ Dilimi için, bir Ağ Dilimi'ne yönelik kayıtlı UE ve Protokol Veri Birimi (PDU) Oturumları sayısı için yük seviyesini ve bir Ağ Dilimi Örneği için kaynak kullanımını gösterir.
NF yük bilgisi	NF yük bilgisi	Bu analitik, belirli bir NF ya da NF grubu için yük istatistiklerini ya da tahminlerini tanımlar.
Ağ Performansı Bilgisi	Ağ Performansı	Bir İlgi Alanında konumlanmış UE ya da UE grupları için performansın istatistiklerini ya da tahminlerini tanımlar.
UE Hareketliliği Bilgisi	UE Hareketliliği	UE hareketliliğinin istatistiksel ya da tahmini analitiklerini tanımlar.

Önceki bölümlerde de bahsedildiği gibi, bu çalışmada NWDAF tarafından sağlanan analitiklerden biri olan UE Hareketliliği Analitiği üzerine deneysel bir araştırma sunulacaktır. Spesifik olarak, bir UE'nin, geçmişte bulunduğu hücreler baz alınarak, gelecekte hangi hücrede olabileceği, seçilen ML modeli ile tahmin edilecektir.

NWDAF'a UE Hareketlilik Analitiği için talep gönderen bir NF, hangi UE'nin hareketlilik bilgisi için bu analitiğin gerekli olduğunu belirtmek adına bir tanımlayıcı (Abonelik Kalıcı Tanımlayıcı (SUPI)) göndermelidir. Gönderilen talep üzerine NWDAF (analitiği oluşturmak için gerekli veriye sahip olmadığı varsayıldığında), AMF ile UE hareketlilik bilgisini toplamak için etkileşime geçer. Burada NWDAF, SUPI ile belirteceği UE için geçmiş hücre giriş bilgilerini ve bu hücrelere giriş zamanı bilgisini toplar (3GPP, 2022). NWDAF, topladığı bu verileri kullanarak, istatistik veya tahmin temelli analitikleri oluşturabilir. Çalışmada ele

alınacak tahmin analitiği için, bir UE'nin geçmişte ve son anda bulunduğu hücre listesi (UE'nin bu hücrelere girişindeki zaman damgaları olmadan) kullanılarak, bir sonraki adımda bulunma ihtimali olan hücre (veya hücreler) ortaya çıkartılmaya çalışılacaktır.

Daha öncesinde bahsedildiği gibi UE Hareketliliği için AMF'den toplanacak veriler (çalışma için devreye alınmış bir AMF olmadığından) sentetik olarak oluşturulmuştur. Bu sebeple standartlara göre, AMF tarafından raporlanan UE tanımlayıcısı olarak belirtilen "supi" ve hücre tanımlayıcısı olarak belirtilen "nrCellid" değişkenleri (3GPP, 2022), çalışma için sırasıyla bir değişken ve o değişkene ait olan bir listeyle temsil edilmiştir.

4. YÖNTEM

Bu bölümde ilk olarak sistem modeli, sonrasında ise sentetik olarak oluşturulan UE hareketlilik verileri kullanılarak UE'nin bir sonraki durumunu tahmin etmek için kullanılacak yöntemler detaylandırılacaktır. Bir sonraki kısımda Markov Zinciri'nin temelleri detaylandırılmıştır. Sonrasında ise Markov Zinciri'nin çalışmada karşılık geldiği kısımlar ve çalışmanın katkısı olan Döngü-Tabanlı Markov Zinciri yönteminin anlatımı yapılmıştır. Son olarak uygulamanın yapılacağı algoritmadan bahsedilmiştir ve üç farklı veri kümesi için UE hareketliliğini tahmin etmek adına, bahsi geçen iki yöntem uygulandığında alınan hesaplama sonuçları Bölüm 5'te gösterilmiştir.

4.1 Sistem Tasarımı

Çalışmada kullanılan sentetik veri kümeleri, daha önceki hareketlilik tahmini çalışmalarında kullanılmış sentetik ve gerçek veri modellerine uygun olarak oluşturulmuştur. Örneğin, hareketlilik tahmini yapılırken kullanılacak veri kümesi için bir öğrencinin hareketliliği düşünüldüğünde, okul ve ev ziyareti herhangi bir kafe ziyaretinden daha sık ortaya çıkacaktır. Bu mantıkla, herhangi bir UE'nin geçmişe dayalı veri kümesinde, bazı konumlar (yani hücreler) diğerlerinden daha sık ortaya çıkmaktadır. Yani, çalışmada bulunan sırasıyla 10bin, 20bin ve 40bin sentetik UE hareketlilik verisi içeren kümelerde mevcut/son hücre ev olarak nitelendirildiğinde, favori bir sonraki hücre okul/iş yeri olarak nitelendirilebilir. Bahsi geçen üç veri kümesi modellemesinde UE'nin düzenli bir harekete sahip olduğu (başka bir deyişle rastgele hareket etmediği) varsayımı kullanılmıştır. Üç veri kümesinin birbirinden farklılıkları, UE'nin favori bir sonraki hücreyle arasındaki farklı sıçramalardan kaynaklanmaktadır. Örneğin bir veri kümesinde favori bir sonraki hücrenin tek sıçramada ortaya çıkışı, diğer veri kümesinde değişiklik göstermektedir.

Bir UE'nin sıradaki durumunu/hücresini tahmin etmek için, geçmişte (bir sonraki adımda) ziyaret ettiği hücrelerin sıklıklarına bakılması bir yaklaşım olabilir. Çünkü, hareketlilik kalıpları düşünüldüğünde, gelecek konumun durumu işaretin

gücünden çok konumların geçmişteki meydana gelme sıklıklarına bağlıdır (Lee vd., 2010). Hücrelerin geçmişte ortaya çıkma sıklıklarını (frekanslarını) kullanarak bir sonraki hücreyi tahmin etmek için, hareketlilik geçmişinin son hücresi kullanılabilir. Bu anlamda, Markov Zinciri, literatür taramasında da belirtildiği gibi, olası bir sonraki hücreyi tahmin etmede kullanılabilir. Ancak, veri kümesindeki son hücre ilk defa meydana geliyorsa, geçmişte bu hücre bulunmadığından tahminde yanılma meydana gelecektir. Bu sebeple, veri kümesindeki son hücrenin ilk defa meydana gelme durumu operasyonun dışarısında tutulmaktadır.

4.2 Markov Zinciri

Gelecekteki (bir sonraki durumdaki) olasılıkların tahmin edilebilmesi için, yalnızca mevcut (bilinen son) durumun kullanıldığı hesaplama yapısına Markov Zinciri ya da Markov Analizi denir. Bir başka deyişle, ileride oluşabilecek olaylar (ya da gelecekteki durumlar) hesaplanırken, yalnızca son duruma bakma operasyonu Markov Zinciri olarak adlandırılabilir ve bu özellik Markovyen Özellik ya da Markov Özelliği olarak bilinir (Alp ve Öz, 2009).

Bir başka deyişle k kadar geçmiş veriye sahip bir süreç için bir sonraki durumu gösteren indeksin (t+1) hesaplanması için yalnızca t indeksine karşılık gelen k durumun kullanılmasına 'Markov Özelliği' adı verilir. Bahsi geçen şekilde k+1 durumuna karşılık gelen t+1 indeksinin hesaplanması 4.1'deki gibidir.

$$Pr(X_{t+1} = x_{k+1} | X_t = x_k, x_{k-2} \dots) = Pr(X_{t+1} = x_{k+1} | X_t = x_k) \quad (4.1)$$

Yukarıdaki denklemin de gösterdiği gibi, k+1'e ait durumun hesaplanması için, yanlızca k indeksine ait durum kullanılmaktadır. Tahminin yapılırken daha önceki durumların ne olduğu, Markov Özelliği gereği, hesaplamanın dışında bırakmaktadır. Denklemi sağlayan (yani Markov Özelliği'ne uyan) ayrıca durum uzayı kesikli ve parametre uzayı kesikli veya sürekli olabilen süreçlere ise Markov Zinciri adı verilir (Alp ve Öz, 2009). Bir Markov Zinciri'nde tek adım geçiş olasılığı 4.2'deki gibi gösterilir.

$$Pr(X_{t+1} = j \mid X_t = i) = p_{ij} \tag{4.2}$$

Daha önce de ifade edildiği gibi, Markov Özelliği yalnızca tek adımdaki değişime baktığından, bir başka deyişle bir sonraki durum tahminin yalnızca bilinen son durum kullanılarak yapıldığından, 4.2'deki denklem tek adımda geçiş olasılığını vermektedir. Bu anlamda bir sonraki durumu belirtmek için kurulacak bir Durum Geçiş Matrisinin herhangi p_{ij} elemanı, 4.3'ü sağlamalıdır.

$$0 \leq p_{ij} \leq 1 \quad , \quad i, j \geq 0$$
$$\sum_{j=0}^{n} p_{ij} = 1, \quad i = 0,1,2,3 \ldots \tag{4.3}$$

Her geçişin içinde bulunacağı Durum Geçiş Matrisi, verilen durum ile gelecek durumun koşullu olasılığını içerir (Chung ve Walsh, 2005). Markov Zinciri, bir stokastik sistemin zaman içerisindeki değişimini ve bulunabileceği farklı durumları incelemek için kullanılır. Bunun yanında, Markov Zinciri'nin, sistemin belli bir zaman sonra ulaşabileceği denge (stabilite) durumunu tahmin edebilirliği de mevcuttur (Aytemiz ve Şengönül, 2004). Bahsi geçen denge durumu, Markov Zinciri'nin 4.4'teki gibi Durum Geçiş Matrisi ile tanımlanabilir.

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{bmatrix} \tag{4.4}$$

Başlangıç olasılıkları verilen *P* matrisini kendisiyle çarptığımız takdirde, belirli bir ıraksamadan sonra (elemanları fazla sayıda kendisiyle çarpmadan sonra) sistem stabil hale gelecektir. Öyle ki tanımlanacak bir $\pi = [\pi_1, \pi_2, \pi_3 \ldots, \pi_s]$ vektörü için:

$$\lim_{n\to\infty} P^n = \begin{bmatrix} \pi_1 & \pi_2 & \ldots & \pi_S \\ \pi_1 & \pi_2 & \ldots & \pi_S \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 & \pi_2 & \ldots & \pi_S \end{bmatrix}$$
$$\sum_{i \in S} \pi_i = 1 \tag{4.5}$$

Durumu sağlanıyorsa ve

$$\pi_j = Pr(X_1 = j) = \sum_{i \in S} Pr(X_1 = j | X_0 = i) \Pr(X_0 = i) = \sum_{i \in S} \pi_i P_{i,j}, \ j \in S$$

(4.6)

Denklemi sağlanıyorsa π dağılımı, ancak ve ancak π vektörü P matrisi tarafından değiştirilemez ise, durağanlığın (stabilitenin) sağlandığı anlamına gelir (Privault, 2018).

4.3 Markov Zinciri ile Bir Sonraki Hücre Tahmini

Markov modelleri verimliliklerinden, kolay uygulanabilirliklerinden ve düşük hesaplama gücü gereksinimlerinden dolayı birçok tahmin algoritmasında kullanılmıştır (Bui vd., 2016). Markov-temelli yöntem, etki alanından bağımsız tahmin edici sınıfı ailesinin üyelerinden biridir ve k-sıralı (ya da O(k)) bir Markov-temelli tahmin edici, mevcut içerik/durum ile sonraki konumun tahmin edilebileceğini varsaymaktadır (Song vd., 2006). Bu çalışmada uygulanacak Markov-temelli model UE'nin geçmiş verilerini hücre (yani UE'nin daha önce giriş yapmış olduğu hücreler) bazında ele almaktadır ve sonucunda ise tahmin için en yüksek olasılığa sahip hücreyi ya da hücre sekansını sunmaktadır. UE tarafından geçmişte ziyaret edilmiş hücre listesi $C = [x_i, x_{i+1}, x_{i+2} \ldots . x_n]$ ele alındığında, x_n UE tarafından ziyaret edilen son hücre olmaktadır. Yani, Markov Zinciri kullanarak bir sonraki hücreyi olasılıksal olarak tahmin etmek mümkün olacaktır. Burada her x_i, $0 \leq \boldsymbol{i} \leq n$ olup, geçmişte ziyaret edilen sınırlı sayıda hücreyi temsil etmektedir.

Bir sonraki olası hücreyi hesaplamak için başlangıç noktası, son hücrenin geçmişte bir sonraki hücre geçişlerini kontrol etmektir. Markov Zinciri, 4.7 ve 4.8'de görüldüğü gibi, gelecek durumu tahmin etmek için yalnızca mevcut durumu kullanır. Bu hafızasızlık durumu Markov Özelliği olarak bilinir (Rosário vd., 2020).

$$P(x_{n+1} \mid x_n, x_{n-1}, x_{n-2} \ldots) = P(x_{n+1} \mid x_n) \quad (4.7)$$

Üzerinde durulan problem için Markov Zinciri, 4.2'de ifade edildiği gibidir.

$$P(x_{n+1} = A \mid x_n = B) \quad (4.8)$$

Burada $A, B \in \{x_i, x_{i+1}, x_{i+2} \ldots . x_n\}$ şeklindedir. Her B son (mevcut) durumda ziyaret edilen hücreyi temsil ederken, A ise B hücresinde bulunan kullanıcının bir sonraki hücre olması durumunu temsil eder.

Örneğin bir UE, 5 farklı hücreye sahip hücre kümesi içinde $S = \{S_1, S_2, S_3, S_4, S_5\}$ bir hareketlilik geçmişine sahip olabilir. Bununla birlikte, bir UE'nin hareketlilik geçmişi, S kümesinin herhangi bir çeşidine sahip olabilir. UE hareketlilik geçmişinin son elemanı (hücresi) kullanılarak yapılan hareketlilik tahmini, olası tüm sonraki hücreler arasında en yüksek olasılığa sahip olan hücreyi tahmin olarak döndürmelidir. Bu Çizelge 4.1'deki durum geçiş matrisinde görülebilir.

Çizelge 4.1. Son hücre için durum geçiş matrisi

	x_i	x_{i+1}	x_{i+2}	…	x_n
x_i	0	$P(x_{n+1} = x_{i+1} \mid x_n = x_i)$	$P(x_{n+1} = x_{i+2} \mid x_n = x_i)$	…	$P(x_{n+1} = x_n \mid x_n = x_i)$

Burada mevcut ve sonraki adım arasındaki olasılıklar, geçmişte bu olasılıkların meydana gelmiş her durumunu kapsamaktadır. Çizelge 4.1'de görülebileceği gibi $Pr(x_{n+1} = x_i \mid x_n = x_i) = 0$ olmaktadır, çünkü yapılan çalışmanın amacı bir UE'nin hareketlilik tahminini çıkartmak olduğundan, durağanlık durumu operasyon dışında bırakılmaktadır.

Markov Zinciri, hareketlilik tahmini için iyi bir yaklaşım sunmasına rağmen, bazı eksikleri mevcuttur. Örneğin, hücreler arası anlambilim düşünüldüğünde Markov Zinciri herhangi bir ilişki oluşturmaz. Çünkü, Markov Zinciri uygulanırken tüm olası sonuçlar hesaplanır ve her adımda yalnızca bir sonraki durumun tahmini çıktı olarak verilir. Başlangıçta varsayımda bulunmanın sorunu çözebileceği

söylenebilir. Örneğin, son adımda hesaplanan olasılıklara bir eşik değeri verilmesi, böylece hücreler arasındaki herhangi bir etkileşimin bu eşikten daha az olması durumunda çıktıda verilmemesi düşünülebilir. Ya da elde edilen verilere göre yalnızca belirli bir eşik üzerindeki hücrenin tahmin olarak kabul edilmesi bir başlangıç varsayımı olarak düşünülebilir. Öte yandan, eşiğin önceden verilmesi, düzgün dağılıma sahip hareketlilik geçmişleri için iyi bir yaklaşım olmayabilir. Bu nedenle hem hücrelerin hem de hücreler arası geçişlerin frekanslarını kullanmak daha iyi bir yaklaşım olabilir.

Hücrelerin ve hücre geçişlerinin frekanslarını hesaplarken, çıktı, bazı hücrelerin ve geçişlerin anlambilimine sahip olduğu sonucunu vermektedir. Örneğin, en yüksek frekansa sahip hücre (yani en çok ziyaret edilen hücre), bir kullanıcı için "ev" olarak ve en az frekansa sahip hücre (geçmişte en az ziyaret edilmiş hücre) bir “alışveriş merkezi” olarak tanımlanabilir. Aynı zamanda, en çok ziyaret edilmiş hücreden sonra gelen hücre “ofis” veya “okul” olarak tanımlanabilir. Bu da demek oluyor ki, bir kullanıcının hareketlilik bilgisi belirli anlambilimlere sahiptir (Chen vd., 2014).

Hücrelerin anlambilimini kullanmak, problemi son hücre ile bir sonraki en çok ziyaret edilen hücre (yani favori bir sonraki hücre) arasındaki bir hareketliliği tespit eden ve bunlar arasında bir döngü oluşturan ve temel Markov Zinciri uygulamasına katkıda bulunmayı amaçlayan çalışmadaki yaklaşıma götürür.

4.4 Döngü Tabanlı Markov Zinciri

Bir sonraki hücre tahmininde, Markov Zincirinden daha iyi sonuç alabilmek için hücrelere anlam vermek yaklaşımın temelidir. Bu sebeple çalışmada yapılacak katkının temel amacı, iki hücre arasındaki geçişlere bir anlam yükleyerek sonuç kısmında verilecek tahminin güvenilirliğini (veya doğruluğunu) güçlendirmektir. Yapılan katkı için aralarında anlam kurulması amaçlanan iki hücreden biri, UE hücre geçmişi listesinin son elemanı (bir başka deyişle ziyaret edilen son hücre) olacaktır. Diğeri ise daha önceden de bahsedildiği gibi son hücreden sonra

geçmişte en çok ziyaret edilen sıradaki hücre (bir başka deyişle favori bir sonraki hücre) olmaktadır.

Ziyaret edilen son hücrenin S_1 olduğu ve hesaplama sonrası bu hücrenin geçmişte bir sonraki en çok ziyaret ettiği hücrenin S_3 olarak bulunduğunu varsayalım. Bu durumda S_1 için sonraki favori hücre S_3 olacaktır. Aynı zamanda bu, hücre geçmişi listesine sahip olunan UE'nin S_1 hücresine girdiğinde S_3'e geçme eğilimde olduğu anlamına gelir. Yani, bu UE için S_1 ve S_3 hücreleri arasında bir anlambilim olduğu anlamına gelir.

Yukarıda da bahsedildiği gibi, bir UE, düzenli harekete sahip olduğu düşünüldüğünde, iki farklı hücre arasında daha sık geçişe sahip olma eğilimdedir. Detaylı olarak, son hücrenin gelecekte geçiş yapacağı potansiyel bir sonraki hücrenin kontrolü yapıldığında, favori bir sonraki hücre ile son hücre (mevcut hücre) arasında bir anlambilim ortaya çıkacaktır. Bu da, mevcut hücreden, bu hücre ile arasında anlambilim kurulan favori bir sonraki hücreye geçişin bir adımda olmayıp birden çok sıçrama sonrasında da olabileceği anlamına gelmektedir. Örneğin bu UE, S_1 hücresine girdikten iki geçiş sonra S_3 hücresine giriş yapıyorsa, bu UE S_3'e giriş yapmak için S_1'den sonra iki sıçrama yapmış anlamına gelmektedir. Bu nedenle amaç, son hücre ile favori bir sonraki hücre arasındaki sıçrama dizilerini (sekanslarını) kontrol etmek ve hesaplamaktır. Örnek bir UE hareketlilik geçmiş listesi Çizelge 4.2'de görülebilir.

Çizelge 4.2. Örnek UE hareketlilik geçmişi verisi

S_1	S_2	S_3	S_1	S_4	S_2	S_1	S_3	S_4	S_2	S_4	S_5	S_3	S_1	S_2	S_4	S_1	S_2	S_3	S_1

Çizelge 4.2'de görüldüğü gibi son hücre S_1'dir ve Çizelge 4.1'deki hesaplama kullanıldığında, bu hücrenin geçmişte en çok ziyaret ettiği hücre S_2 olarak görülmektedir. Bu sebeple, sıradaki hesaplanması gereken, S_1'den başlayıp S_2 ile biten birden çok sıçrama durumlarıdır (Çizelge 4.2). Örneğin bu UE dördüncü adımda giriş yaptığı S_1 hücresinden sonra S_2 hücresine girebilmek için iki sıçrama yapmıştır (S_2 hücresinden önce S_4 hücresine de uğramıştır). Farklı sıçramaların tahminleri olasılıksal olarak karşılaştırılacağından, S_1 hücresinden

S_4 hücresine geçiş ve S_4 hücresinden S_2 hücresine geçiş olasılıkları da, yine Çizelge 4.1'deki durum geçiş matrisi kullanılarak, sırasıyla hesaplanmalıdır. Bu iki durum geçişi için yapılması gereken hesaplama 4.9'da görüldüğü gibidir.

$$P(x_{n+2} = S_2 | x_n = S_1) = P(x_{n+2} = S_2 | x_{n+1} = S_4) * P(x_{n+1} = S_4 | x_n = S_1) \quad (4.9)$$

Diğer tüm sıçramalar için (bir sıçramadan daha fazla olan) 4.9 denklemi uygun formata getirilerek Çizelge 4.2'deki hücre listesine ($S_n \in S$ ve n = [1,2,3,4]) göre 4.10'daki gibi çözümlenebilir.

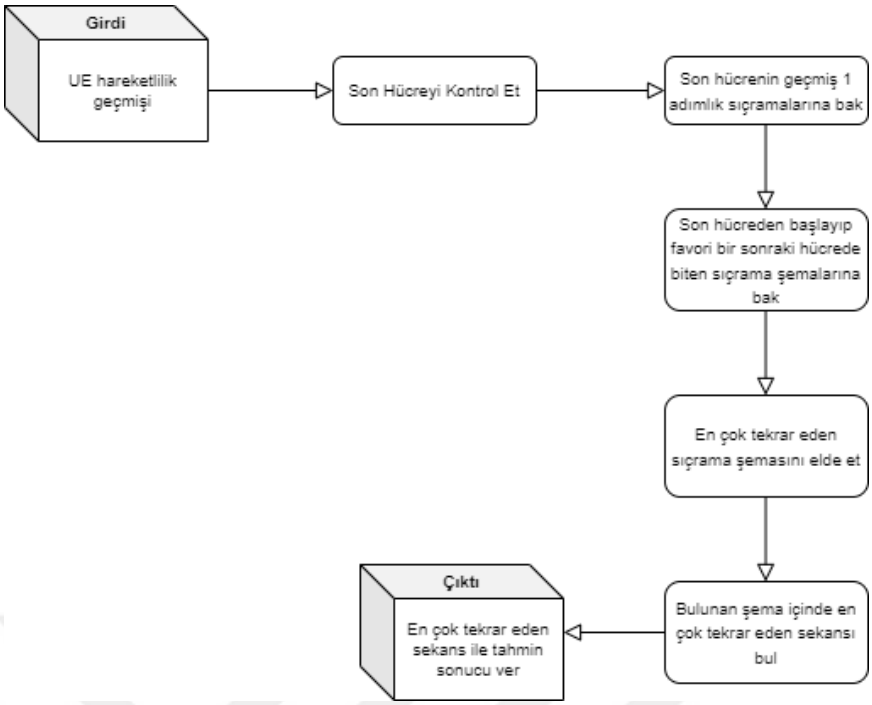
$$P(x_{n+k} = S_n \mid x_n = S_n) = \prod_{l=1}^{k} P(x_{n+l} = S_n | x_{n+l-1} = S_n) \quad (4.10)$$

(4.4)'deki denkleme göre, üst limit k sıçrama sekansına karar vermektedir (k=1 çalışmada Markov Zinciri olarak tanımlanmaktadır). Çizelge 4.2'de görüldüğü gibi iki hücre arasında birden çok sıçrama sekansı olacağından, bütün olası sıçramalar için bir döngü kurmak S_1 ve S_2 arasındaki bütün olası sıçramalar sonucunu verecektir. Aynı zamanda, herhangi bir sıçrama senaryosu (son hücre ve favori bir sonraki hücreden başka), bu şekilde uygulanabilir. Diğer bir taraftan, yukarıdaki UE hücre geçmişi örneği için S_1 ve S_2 arasında bir anlambilim olduğundan, tahmin için bu iki hücre arasındaki sıçramaları kullanmak tahmin için daha anlamlı bir sonuç verecektir**.**

Sonuç olarak, UE hücre geçmişi listesindeki son hücre ve bu hücrenin favori bir sonraki hücresi arasında, anlambilimleri bakımından, döngü-tabanlı bir yaklaşımda bulunmak Markov Zincirinin temel özelliğini hücreler arasındaki sıçramaya göre tahmin yapmak adına geliştirebilir.

4.5 Sistem Algoritması ve Kaynak Kodları

Girdi olarak alınacak UE hareketlilik geçmişi verisinin (listesinin) sırasıyla gireceği operasyonlar Şekil 4.1'deki gibidir. Bahsi geçen adımların python kodu karşılıkları ise Şekil 4.1'in alınta adımların açıklaması olarak verilmiştir.



Şekil 4.1 Sistemin akış diyagramı

Birinci adımda elde edilecek girdi, bu çalışmada oluşturulmuş sentetik UE hareketlilik verisidir. Bahsi geçen veri içerisinde ilk olarak son eleman (UE'nin bulunduğu bilinen son/mevcut hücre) elde edilir. Çünkü problem bu UE'nin sonraki durumlarda nerede olabileceğinin tahminidir. Aynı zamanda, eğer, son hücre listede ilk defa bulunmuşsa, bu hücrenin geçmişte herhangi bir tekrarı olmadığı için bir sonraki adımı da olmayacağından, son hücre listeden silinir.

```
def checklast(uehistory):
    for i in range(len(uehistory)):
        if uelist.count(uehistory[len(uehistory)-1]) < 2:
            uehistory.remove(uehistory[len(uehistory])
    last_known_cell = uehistory[len(uehistory)-1]
    return last_known_cell
```

Son hücre yukarıdaki gibi elde edildikten sonra, geçmişte son hücreden bir adım sonra (yani bir sıçramada) bu UE'nin hangi hücrelere gittiği oranları ortaya çıkartılır.

```
def next_possible_ratio(history_list,current_cell):
```

```
    possible_next = []

    for i in range(len(history_list)-1):
        if history_list[i] == current_cell:
            possible_next.append(history_list[i+1])

    _uniq_possible_next = np.unique(possible_next)

    ratio_possible_next = []
    for i in range(len(_uniq_possible_next)):
        ratio_possible_next.append(_uniq_possible_next[i])
        ratio_possible_next.append(possible_next.count(_uniq_possible_next[i])/(le
        n(possible_next)))

    return ratio_possible_next
```

Yukarıdaki foksiyon çalıştırılıp oranlar elde edildikten sonra, bir sıçramada en yüksek oran bulunur. Bu oranı içeren bir sonraki hücre, favori bir sonraki hücredir.

```
def _1_hop_output(next_ratios, lastcell):
    highest = []

    for t in range(0,len(next_ratios)-1,2):
        highest.append(next_ratios[t+1])
    highest = max(highest)

    highest_with_cell = []
    for k in range(len(next_ratios)):
        if highest == next_ratios[k]:
            highest_with_cell.append(next_ratios[k-1])
    highest_with_cell.append(highest)

    favorite_next_cell = highest_with_cell[0]
    favorite_1hop_scheme = [lastcell, highest_with_cell[0], highest_with_cell[1]]
```

```
    return favorite_1hop_scheme
```

Favori bir sıçramalık şema ile favori bir sonraki hücre elde edildikten sonra, bu iki hücre arasında (son hücre ile başlayıp favori bir sonraki hücre ile biten) farklı sıçramalar elde edilebilecektir. Bunun öncesinde ise bu iki hücre arasında yapılmış farklı sıçramaların sayısına bakılır.

```
def most_hopping_seq(history_list, nextcell, cellname):
    count_1_hop = 0
    count_2_hop = 0
    count_3_hop = 0
    count_4_hop = 0
    hop_list = []

    for i in range(len(history_list)-1):
        if history_list[i] == cellname and history_list[i+1] == nextcell:
            count_1_hop = count_1_hop + 1
    for i in range(len(history_list)-2):
        if history_list[i] == cellname and history_list[i+2] == nextcell:
            count_2_hop = count_2_hop + 1
    for i in range(len(history_list)-3):
            if history_list[i] == cellname and history_list[i+1] != nextcell and
            history_list[i+2] != cellname and history_list[i+3] == nextcell:
                    count_3_hop = count_3_hop + 1
    for i in range(len(history_list)-4):
        if history_list[i] == cellname and history_list[i+1] != nextcell and
history_list[i+2] != cellname and history_list[i+2] != nextcell and history_list[i+3]
!= cellname and history_list[i+3] != nextcell and history_list[i+4] == nextcell:
            count_4_hop = count_4_hop + 1
    hop_list = [count_1_hop, "one hop occurrence", count_2_hop, "two hop
occurrence", count_3_hop, "three hop occurrence", count_4_hop, "four hop
occurrence"]

    print(hop_list)
```

```
    _hop_scheme = []
    for j in range(0,len(hop_list),2):
        _hop_scheme.append(hop_list[j])

    most_occured_hop = []

    most_occured_hop.append(max(_hop_scheme))
    if max(_hop_scheme) == _hop_scheme[0]:
        most_occured_hop.append("most occured is 1 hop")
    if max(_hop_scheme) == _hop_scheme[1]:
        most_occured_hop.append("most occured is 2 hop")
    if max(_hop_scheme) == _hop_scheme[2]:
        most_occured_hop.append("most occured is 3 hop")
    if max(_hop_scheme) == _hop_scheme[3]:
        most_occured_hop.append("most occured is 4 hop")

    print(most_occured_hop)

    most_occured_hop_num = most_occured_hop[0]

    return most_occured_hop_num
```

Bu fonksiyonun sonucu ise hangi sıçrama şemasının ne kadar yapıldığını söyler ve en çok yapılan sıçrama şemasının sayısını döndürür. Bu fonksiyonun veri kümelerine uygulanmış çıktısı Bölüm 5 Hesaplama Sonuçları'nda bulunmaktadır.

Bundan sonra ise farklı sıçrama şemaları için, farklı sıçrama sekansları (son hücre ve favori bir sonraki hücre arasında) kontrol edilir. Örneğin son hücre 'A' ve favori bir sonraki hücre 'B' ise, iki sıçramalı sekanslardan biri 'A','C','B' olabilir. İkili sıçrama sekanslarının tamamını kontrol eden ve olasılığı en yüksek olanı ortaya çıkartan iki fonksiyon sırasıyla aşağıdaki gibidir.

```
def _2_hop(cell_list,start_cell,end_cell):
    possible_2_hops = []
    starting_cell = []
```

```
    pr_1 = 0
    middle_cell_next = []
    pr_2 = 0

    for i in range(len(cell_list)-2):
        if cell_list[i] == start_cell and cell_list[i+2] == end_cell:
            starting_cell = next_possible_ratio(UE_history_1, start_cell)
            pr_1 = starting_cell[possible_next.index(cell_list[i+1])+1]
            middle_cell_next = next_possible_ratio(UE_history_1, cell_list[i+1])
            pr_2 = middle_cell_next[middle_cell_next.index(end_cell)+1]

            possible_2_hops.append(start_cell)
            possible_2_hops.append(cell_list[i+1])
            possible_2_hops.append(end_cell)
            possible_2_hops.append(pr_1 * pr_2)

    return possible_2_hops
```

Son hücre ve favori bir sonraki hücre arasında olan bütün iki sıçramalı sekanslar yukarıdaki fonksiyon ile elde edildikten sonra, bunlar arasında en çok meydana gelmiş olan, bir başka deyişle olasılığı en yüksek olan, tahmin çıktısı olarak verilir.

```
def _2_hop_output(_2_hop_pr_list):

    highest_2_loop = []

    for i in range(3,len(_2_hop_pr_list)-3,4):
        highest_2_loop.append(_2_hop_pr_list[i])
    highest = max(highest_2_loop)

    highest_with_2loop_pattern = []
    for j in range(len(_2_hop_pr_list)-3):
        if _2_hop_pr_list[j] == highest:
            if _2_hop_pr_list[j-3] not in highest_with_2loop_pattern:
                highest_with_2loop_pattern.append(_2_hop_pr_list[j-3])
            if _2_hop_pr_list[j-2] not in highest_with_2loop_pattern:
                highest_with_2loop_pattern.append(_2_hop_pr_list[j-2])
```

```
            if _2_hop_pr_list[j-1] not in highest_with_2loop_pattern:
                highest_with_2loop_pattern.append(_2_hop_pr_list[j-1])
    highest_with_2loop_pattern.append(highest)

    return highest_with_2loop_pattern
```

Yukarıdaki fonksiyonlar ile elde edilen son hücre ve favori bir sonraki hücre (bir sıçrama) ile son hücre ile favori bir sonraki hücre arasında en çok ziyaret edilen hücre (iki sıçrama) sekanslarının tahminleri verilmiştir. Bu iki sıçrama sekansı arasındaki karşılaştırma Bölüm 5'te verilmiştir.

5. HESAPLAMA SONUÇLARI

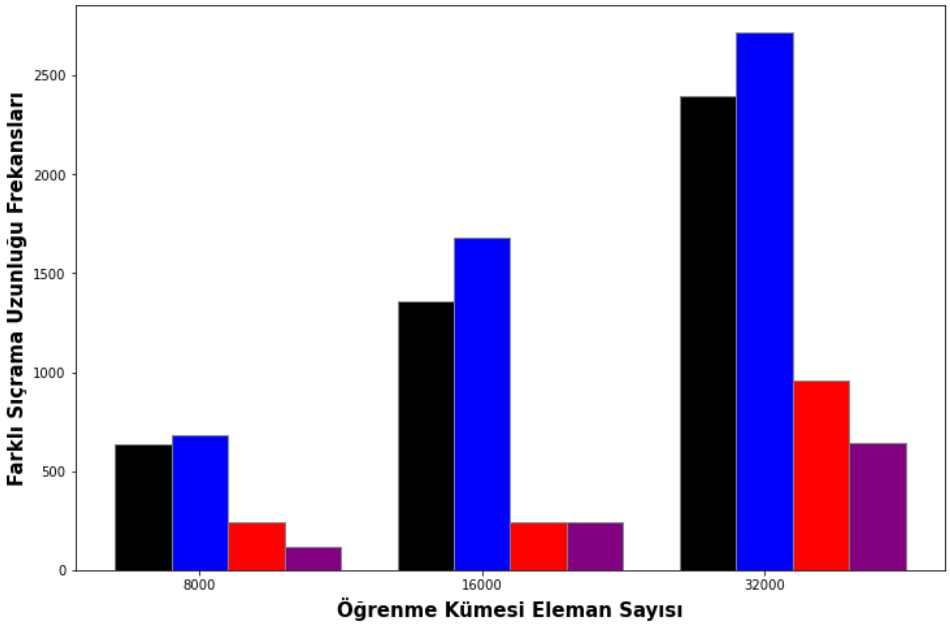
Bu çalışmada, UE Mobilite tahmini, ilk olarak bir Markov Zinciri ile uygulanmıştır. Daha sonra bu çalışmada teorik olarak anlatılan ve standart Markov Zinciri ile yapılan tahminin üzerine çıkması amaçlanan Döngü-tabanlı Markov Zinciri aynı veri kümesine uygulanmıştır. İki yöntemin uygulanabilmesi için de veri kümesi (bir başka deyişle UE hareketlilik geçmişi) AMF raporuna uygun olarak hazırlanmıştır. Çalışmada iki hücre arasında döngü kurmak ve farklı sıçramalarını kontrol etmenin arkasında yatan fikir, iki hücre arasında bir anlambilim kurulabileceğidir. Bu perspektiften bakıldığında, Markov Zinciri'nin verdiği sonuç, Döngü-tabanlı Markov Zinciri'nin verdiği sonuca göre daha az bilgi barındırmaktadır. Çünkü, Markov Zinciri son hücreye göre bir sonraki hücrenin ne olacağını kontrol ederken iki hücre arasında bir ilişki kurmaz ve her adımda bir sonraki hücrenin ne olacağını hesaplar. Bunun negatif bir sonucu olarak ise, UE'nin yapabileceği herhangi bir rastgele hareketi Markov Zinciri operasyon dışında bırakmaz.

Öte yandan, bu araştırmada sunulan yeni yöntem iki hücre arasında bir anlambilim oluşturacağından, gereksiz bilgi elimine edilebilir. Örneğin, son hücreden sonraki herhangi bir hücre geçmişte yalnıca bir kere ortaya çıktıysa, bu iki hücre arasındaki farklı sıçrama geçişleri hesaplanmayacaktır. Bu nedenle, sentetik olarak oluşturulan UE hareketlilik modeli için, Döngü tabanlı Markov Zinciri daha anlamlı sonuçlar vermeyi başarmıştır. Burada anlamlı sonuç, iki hücrenin atlama şemaları tarafından oluşturulan tahminin doğruluğunu ifade eder.

Yaklaşımımızı test etmek için öncelikle 3 farklı sentetik veri kümesi kullanılarak son ve favori bir sonraki hücre arasındaki farklı sıçramalar çıkarılmıştır. Bu 3 veri kümesi sırasıyla 10bin, 20bin ve 40bin UE hareketlilik geçmişi içermektedir. Veri kümelerinin birbirinden farklılaşan kısmı, aynı UE'nin tek sıçramada ve birden çok sıçramada ziyaret ettiği farklı sekanslardır. Oluşturulan 3 veri kümesi için varsayım, bu verilerin aynı UE'den alınmış olmasıdır. Aynı UE'den alınan verilerin farklılıkları ise, bir UE'nin farklı anlarda giriş yapacağı hücrelerin ve/veya hücre

sekanslarının değişmesidir. Nitekim bir UE'nin hareketlilik profili düşünüldüğünde, ev veya iş yeri gibi sürekli olarak bulunduğu bölgelere ait olan hücreler daha çok ortaya çıkacaktır ve farklı veriler alınmış da olsa bahsi geçen UE'nin hareketlilik şeması yüksek oranda değişmeyecektir (Chen vd., 2014). Çalışma için oluşturulan veriler, daha önce de bahsedildiği gibi, sentetik olarak oluşturulduğu için, bu çalışma kapsamında kontrol edilecek ilk şey, bir UE'nin yalnızca giriş yaptığı hücre listesinin bulunduğu senaryoda (zaman damgası gibi özellikler olmaksızın) geliştirilen modelin kullanımının kontrolüdür. Nitekim Bölüm 6'da zaman damgası özelliğinin de dahil olduğu geliştirmelerden bahsedilecektir.

Veri kümesi içerisindeki farklı sıçrama sekansları Bölüm 4.5'teki algoritma ve kaynak kodları kullanılarak hesaplanırken, bir sıçrama sekansının uzunluğu diğerlerinden büyük olabilir (örneğin, 2 sıçramalı döngü daha çok meydana gelmiş olabilir). Bu nedenle, en sık döngü önceden hesaplandığından (eğitim kümesinde) hesaplanacak ilk şey, yeni veri (test) kümesindeki döngülerin oluşumlarını karşılaştırmaktır. Bu nedenle, en sık döngü (örn., 2-döngü) farklı atlama şemalarına sahipse tahmin sonucu olarak en yüksek olasılığa sahip olan, yani kontrol sonucunda en çok tekrar eden, çıktı olarak verilecektir. 3 farklı veri kümesi arasındaki döngü dağılımı Şekil 5.1'de görülebilir. Siyah, mavi, kırmızı ve mor çubuklar sırasıyla 1 sıçrama (1 döngü ya da Temel Markov Zinciri), 2 döngü, 3 döngü ve 4 döngüyü temsil eder.

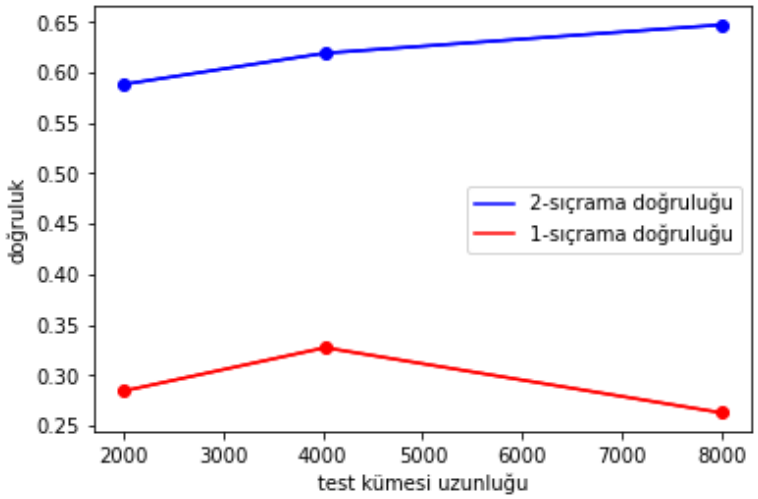


Şekil 5.1. 3 farklı veri kümesinin sıçrama döngüsü dağılımları

Şekil 5.1'den de görülebileceği gibi, bu UE sıradaki favori hücresine bir ya da iki adımda geçiş yapma eğilimindedir. Sistem çıktı adımında tahmin olarak bir hücre yerine bir sekans vereceğinden, bu UE'nin hareket özelliği daha çok yaptığı sıçrama olan 2-sıçramalı döngü olmalıdır. En yüksek tekrara sahip 2-sıçramalı döngü çıkartıldıktan sonra doğruluk hesaplaması yapılacaktır. Bu doğruluk hesaplaması her sıçrama için 5.1'deki gibi yapılmaktadır.

$$\frac{iki\ sıçramada\ S_1\ ile\ başlanıp\ S_2\ ile\ biten\ sekanslar\ arasında\ en\ çok\ meydana\ gelen}{iki\ sıçramada\ S_1\ ile\ başlanıp\ S_2\text{'de biten tüm sekanslar}} \quad (5.1)$$

Sonrasında ise 3 veri kümesinin 0.8 kadarı eğitim için ve 0.2 kadarı ise test için kullanıldığında, 1-sıçrama ve 2-sıçrama için doğruluk sonuçları Şekil 5.2'deki gibi olmaktadır.



Şekil 5.2. 1 ve 2 sıçramalı tahminlerin doğrulukları

Sonucun da gösterdiği gibi, iki hücre arasında anlambilim kurmak doğruluğu daha yüksek tahmin sonucu ortaya çıkartmıştır. 3 farklı veri kümesi 0.8'i eğitim ve 0.2'si test olarak ayrıldığında ortaya çıkan sonuç Şekil 5.2'e göre, bu UE'nin farklı zamanlarda 1 sıçramada favori bir sonraki hücreye geçme eğilimi değişirken, 2 sıçramada öngörülen sekansı tekrarlama eğilimindedir ve bunun bir karşılığı olarak artan veri kümesinde doğruluk oranı da artmıştır. Başka bir deyişle verilerimiz ve yaklaşımımız için doğruluk sonucu, bu UE'nin 2 atlamada son hücrenin favori bir sonraki hücresine geçtiğini göstermektedir. Ancak, diğer herhangi bir UE hareket özelliği için sonuç değiştirilebilir. Öte yandan, normal UE hareketi çalışmanın varsayımı olduğundan, bir UE sıklıkla 5 veya daha fazla döngü oluşturmaz. Çünkü son hücre ile favori bir sonraki hücre arasındaki hücre sayısının çok fazla olması beklenmez. Ayrıca, bu UE'nin hareketliliği gelecekte farklı özelliklere (dağılımlara) sahip olacaksa, tahmin başarısız olabilir. Örneğin, yeni veri kümesi 2 atlamadan daha fazla 4 atlama içeriyorsa, eğitim kümesi bunun tersini içerdiğinden tahmin daha az doğruluk sağlayacaktır. Bu nedenle UE hareketlilik listesine bir özellik daha eklemek (örneğin hücreye girişin zaman damgaları) bir çözüm olabilir.

6. SONUÇ VE ÖNERİLER

Bu çalışmada UE hücre geçmişi listesindeki son hücre ile bu hücrenin sonraki favori hücresi arasında bir anlambilim kurularak bir sonraki durum tahmini yapılmaya çalışılmıştır. Çalışmadaki fikri geliştirebilmek için eklenebilecek özelliklerden biri, başlangıçta yalnızca son durumu (hücreye) kontrol etmek yerine, k kadar önceki durumun kontrolü tahmini güçlendirmek için yapılabilir. Bir başka deyişle, Markov Özelliği ilerletilebilir. Örneğin, UE geçmiş hücre listesindeki son hücreden bir önceki hücre 'favori bir sonraki hücre' olarak adlandırılan hücre ise, 'tekrar bu hücreye geçiş için birden fazla sıçrama yapmak gereklidir' gibi bir yaklaşım sağlanabilir. Aynı zamanda, çalışmada kapsanan tek şey UE hücre listesindeki hücre sekansları (yani hareketlilik izleri) olduğundan, bu hücreye girişlerin zaman damgası da bir özellik olarak eklenerek, tahmin güçlendirilebilir. Bu nedenle, bir sonraki t periyodunda bir UE'nin gireceği hücreleri tahmin etmek için, daha önce girilmiş olan hücrelerin zaman damgalarını dikkate almak gerekecektir.

İki geliştirme ile, tahmin daha kesin (veya güvenilir) hale gelebilir. Aynı zamanda, çalışmada kullanılan veri kümesine göre UE'nin düzenli hareketlilik geçmişine sahip olduğu varsayımı düşünüldüğünde, herhangi bir UE rastgele harekete sahipse, kullanılan model tutarlı sonuç veremeyecektir.

KAYNAKLAR

3rd Generation Partnership Project [3GPP], 2022. TS 29.518. 5G System; Access and Mobility Management Services, 16.11.0, Erişim Tarihi: 01.06.2022 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3339

3rd Generation Partnership Project [3GPP], 2022. TS 29.520. 5G System; Network Data Analytics Services; Stage 3, , 16.11.0, Erişim Tarihi: 01.06.2022 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3355

3rd Generation Partnership Project [3GPP], 2022. TS 23.288. Architecture enhancements for 5G System (5GS) to support network data analytics services, 16.10.0, Erişim Tarihi: 01.06.2022 https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579

Ahmet Ş., Tevfik A., 2004. Markov Zincirlerinin Ekonomik Bir Probleme Uygulanması : Perakende Alışverişlerinde Bireysel Olarak Kullanılan Madeni Para Stratejilerinin Karşılaştırılmalı Analizi. Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 6(4), 29-43.

Ben L., Chansu Y., Myungchul K., Seung-Min P., Weetit W., Won-Tae K., 2011. Behavior-based mobility prediction for seamless handoffs in mobile wireless networks. Wireless Networks, 17, 645–658.

Changchun P., Genke Y., Haichun L., Junjie J., 2016. Predicting Human Mobility Based on Location Data Modeled by Markov Chains. 2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS), 2016, 145-151, Shangai.

Denis R., Eduardo C., Fábio A., Felipe A., Kássio M., Leonardo A. V., 2020. Ensemble mobility predictor based on random forest and Markovian property using LSBN data. Journal of Internet Services and Applications, 11(7), 1-12.

Emmanouil P., Fabrizio M., Christian M., Paul A., Mehrdad S., Michael E., Qing W., Ömer B., Antonio de D., 2019. End-to-end Data Analytics Framework for 5G Architecture, IEEE Access, 40295-40312.

Jun P., Ran X., Xihui C., 2014. Constructing and Comparing User Mobility Profiles. ACM Transactions on the Web, 8(4), 1-25.

Kai Lai C., John B. W., 2005. Markov Processes, Brownian Motion, and Time Symmetry, 2(249). Springer Science.

Libo S., David K., Ravi J., Xiaoning H., 2006. Evaluating Next-Cell Predictors with Extensive Wi-Fi Mobility Data. IEEE TRANSACTIONS ON MOBILE COMPUTING, 5(12), 1633-1649.

Lizhu Z., Wei-Ying M., Xing X., Yu Z., 2009. Mining interesting locations and travel sequences from GPS trajectories. Proceedings of the 18th international conference on World wide web, 2009, 791-800, New York.

Marc-Oliver K., Miguel Núñez del Prado C., Sébastien G., 2012. Next place prediction using mobility Markov chains. Proceedings of EuroSys 2012 Workshop on Measurement, Privacy, and Mobility (MPM)(3), 2012, 1-6, Bern..

Nicola B., Matteo C., S. Amir H., Qi L., Ilaria M., Joerg W., 2016. Anticipatory Networking in Future Generation Mobile Networks. IEEE Communications Surveys & Tutorials, 2017, 19(3), 1-31.

Nicolas P., 2018. Understanding Markov Chains, Springer, Singapore, 166.

Qiujian L., Zongshan M., Yuanyuan Q., Yufei Z., Zhenming L., 2014. Hidden Markov Model Based User Mobility Analysis in LTE Network. 17th International Symposium on Wireless Personal Multimedia Communications (WPMC), 2014, 379-384, Sydney.

Quannan L., Wei-Ying M., Wenyu L., Xing X., Yu Z., Yukun C., 2008. Mining user similarity based on location history. 16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems(34), 2008, 1-10, California.

Selçuk A., Ersoy Ö., 2009. Markov Zinciri Yöntemi ile Taşınabilir Bilgisayar Tercihlerinin Analizi. Akademik İncelemeler, 4(2), 2009.

Theodoros A., Christos A., Stathes H., 2011. Mobility Prediction based on Machine Learning. 2011 12th IEEE International Conference on Mobile Data Management, 2011, 27-30 Lulea.

Thuy Van T. D., Dinh Que T., 2015. A Fusion of Data Mining Techniques for Predicting Movement of Mobile Users. Journal of Communications and Networks, 17(6), 568-581.

Wenhui W., Xiaoyan D., Wanfei S., Ming A., 2021. Research on Mobility Prediction in 5G and beyond for vertical industries. IEEE/CIC International Conference on Communications in China, 2021, 379-383, China.

Yuanyuan Q., Zhongwei S., Yanting Z., Fehmi Ben A., Xinyu Z. Jie Y., 2018. A hybrid Markov-based model for human mobility prediction. Neurocomputing, 278, 99-109.

ÖZGEÇMİŞ

Adı Soyadı : Bahadır YALIN

Eğitim Durumu

Lise : Özel Toros Okulları, 2014

Lisans : İstanbul Ticaret Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği

Yüksek Lisans : İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yayınları

Yalın, B., Yarkan, S. 2022. Loop-Based Markov Chain For Next Cell Prediction, Journal of Technology and Applied Sciences, Basımda.