GİZLİLİĞİ KORUNMUŞ ORTAK FİLTRELEME YÖNTEMLERİNİN DOĞRULUĞUNUN KABA KÜMELER TEORİSİ İLE İYİLEŞTİRİLMESİ

Yüksek Lisans Tezi

Adem Öztürk

Eskişehir, 2017

GİZLİLİĞİ KORUNMUŞ ORTAK FİLTRELEME YÖNTEMLERİNİN DOĞRULUĞUNUN KABA KÜMELER TEORİSİ İLE İYİLEŞTİRİLMESİ

Adem ÖZTÜRK

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Cihan KALELİ

Eskişehir
Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Ocak, 2017

Bu tez çalışması TÜBİTAK tarafından 114E571 no'lu proje kapsamında kısmen desteklenmiştir.

JÜRİ VE ENSTİTÜ ONAYI

Adem Öztürk'ün "Gizliliği Korunmuş Ortak Filtreleme Yöntemlerinin Doğruluğunun Kaba Kümeler Teorisi ile İyileştirilmesi" başlıklı tezi 19/01/2017 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Anadolu Üniversitesi Lisansüstü Eğitim - Öğretim ve Sınav Yönetmeliği"nin ilgili maddeleri uyarınca, Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi kabul edilmiştir.

Ünvanı-Adı Soyadı		**İmza**
Üye (Tez Danışmanı)	**: Doç. Dr. Cihan Kaleli**	.....
Üye	**: Yrd. Doç. Dr. Alper Bilge**	.....
Üye	**: Yrd. Doç. Dr. Efnan Şora Günal**	.....

.....

Enstitü Müdürü

ÖZET

GİZLİLİĞİ KORUNMUŞ ORTAK FİLTRELEME YÖNTEMLERİNİN DOĞRULUĞUNUN KABA KÜMELER TEORİSİ İLE İYİLEŞTİRİLMESİ

Adem ÖZTÜRK
Bilgisayar Mühendisliği Anabilim Dalı
Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Ocak, 2017
Danışman: Doç. Dr. Cihan KALELİ

İnternet kullanımının artmasıyla birlikte tavsiye sistemleri popüler hale gelmiştir. Tavsiye sistemlerinde kullanılan Ortak Filtreleme yöntemleri müşterilere çevrimiçi platformlar üzerinde ürün seçme konusunda yardımcı olmak için kullanılmaktadır. Bu yöntemlerin gizlilik, doğruluk, çevrimiçi performans, kapsama, çok seyrek veri seti ve ölçeklenebilirlik gibi bazı sorunları vardır. Bu sorunların üstesinden gelmek için Kaba Kümeler Teorisi kullanılabilir. Kaba Kümeler teorisi sistemin doğruluğunun iyileştirilmesi, kapsama performansının arttırılması amacıyla Ortak Filtreleme yöntemlerinde kullanılmaktadır. Ayrıca Ortak Filtreleme yöntemlerinin önemli sorunlarından birisi de gizliliktir. Bu sorunun üstesinden gelmek için Gizliliği-Korunmuş Ortak Filtreleme yöntemleri kullanılmaktadır. Ancak bu sistemlerde doğruluk ve gizlilik çakışan iki amaç olduğundan tavsiye sisteminin doğruluğunu düşürmektedir. Bu tezin amacı da Gizliliği-Korunmuş Ortak Filtreleme yöntemlerinin doğruluğunun iyileştirilmesi ve kapsama perfromansının arttırılmasıdır. Bu sorunların üstesinden gelmek için Kaba Kümeler Teorisinde ayırt edilemezlik ilişkisi kullanılarak geliştirilen ROUSTIDA algoritması kullanılmıştır. Bu yaklaşım gizliliği korunmuş bellek tabanlı, gizliliği korunmuş model tabanlı ve gizliliği korunmuş karma tabanlı üç farklı yöntemle test edilmiştir. Deneyler sonucunda Gizliliği Korunmuş Ortak Filtreleme yöntemlerinde doğruluğun ve seyrek veri sorunlarının iyileştiği görülmüştür.

Anahtar Kelimeler: Kaba Kümeler Teorisi, Ortak Filtreleme, Gizlilği Korunmuş Ortak Filtreleme, Tavsiye Sistemleri, ROUSTIDA.

ABSTRACT

IMPROVING ACCURACY OF PRIVACY-PRESERVING COLLABORATIVE FILTERING METHODS BY ROUGH SETS THEORY

Adem ÖZTÜRK

Computer Engineering Department

Anadolu University, Graduate School of Sciences, January, 2017

Supervisor: Assoc. Prof. Dr. Cihan KALELİ

Recommender systems have become popular with increasing use of Internet. Collaborative filtering methods which use recommender systems have been used in order to for selecting product over online platforms. There are some challenges such as privacy, accuracy, online performance, coverage, sparsity and scalability of these methods. Rough sets theory has been used in order to overcome these challenges. Rough sets theory has been used with the intent of improving accuracy, expansion of coverage and increasing online performance of the collaborative filtering methods. Privacy is one of the important issues of the collaborative filtering methods. After customers log in the system, this issue begins. Privacy-preserving collaborative filtering methods have been used to cope with this issue. However, these methods decrease accuracy of the recommender system. The purpose of this thesis is overcoming the issue of preserving of privacy. ROUSTIDA algorithm which improved using indiscernibility relation in the rough sets theory has been used to cope with this issue. This approach has been tested with three different algorithms, namely memory-based privacy-preserving, model-based privacy-preserving and hybrid privacy-preserving. In result of experiments, accuracy and sparsity issues have been relieved in the privacy-preserving collaborative filtering methods.

Keywords: Rough sets theory, Collaborative filtering, Privacy-preserving collaborative filtering, Recommender systems, ROUSTIDA.

19/01/2017

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmanın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilemeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Anadolu Üniversitesi tarafından kullanılan "bilimsel intihal tespit programı" ile tarandığını ve hiç-bir şekilde "intihal içermediğini" beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

Adem ÖZTÜRK

İÇİNDEKİLER

Sayfa

TABLOLAR DİZİNİ

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

KISALTMALAR DİZİNİ

GKOF : Gizliliği Korunmuş Ortak Filtreleme

HKOK : Hatalar Kareler Ortalamasının Karekökü

IS : Bilgi Sistemi

KKT : Kaba Kümeler Teorisi

OF : Ortak Filtreleme

OMH : Ortalama Mutlak Hata

RKT : Rastgele Karışıklık Tekniği

1. GİRİŞ

Günümüzde internet kullanımının artmasıyla birlikte mobil cihazların kullanımı da artmıştır. Bununla birlikte insanlar da daha fazla internette zaman harcamakta ve pek çok işini internet vasıtasıyla halletmektedir. Aynı zamanda sosyal medya kullanımı da artmakta ve mobil cihazlar için çok çeşitli uygulamalar geliştirilmektedir. Bu sayede pek çok insan birbirinden daha hızlı haber almakta ve gelişmeleri takip etmektedir. Bunun yanında internet üzerinden alışveriş imkanları da gelişmiştir. Bu alışveriş bir ürün satın alma olabileceği gibi bir hizmet alımı da olabilir. Bu hizmet bir film izleme, makale araştırma, haber takip etme vb. olabilir. İnternetin bu kadar yaygınlaşması ürün ve hizmet çeşitliliğinde artışa neden olmuş ve bu da insanların çok fazla sayıdaki ürün arasından beğenebileceği ve ihtiyacı olan ürünü bulmasını zorlaştırmaktadır. Tavsiye sistemleri de bu kadar fazla olan ürün çeşitliliği içerisinden müşteri için en uygun ve doğru ürünü seçmesini sağlamaktadır.

Tavsiye sistemleri müşterilere ürün seçmede kolaylık sağlamaktadır. Müşteriler bir web sitesi üzerinden alışveriş yapacağı zaman, satın alacağı ürün kategorisi içinde pek çok ürünle karşılaşır ve bu ürünler içinde kendisi için en uygun olanı bulması önemli bir sorunu oluşturmaktadır. Tavsiye sistemleri bu tür sorunların üstesinden gelmede kişilere ciddi kolaylıklar sağlamaktadır. Bu yüzden pek çok önemli firmalar bu sistemleri kullanmaktadır. Bunlardan bazıları Amazon.com, eBay, CDNOW, Netflix'dir [1]. Bu sistemlerin uygulama alanlarına ise, film önerme, otel önerme, restoran önerme, web sayfası önerme örnek olarak verilebilir. Tavsiye sistemleri üç alt sınıf altında toplanabilir. Bunlar, içerik tabanlı filtreleme sistemleri, ortak filtreleme sistemleri (*OF*) ve hibrid sistemlerdir [13]. İçerik tabanlı filtreleme sistemlerinde müşteri bir ürün hakkında tavsiye isterken, sistem ürünlerin profil bilgilerini (örneğin bir film için, yönetmen, tür, oyuncular) kullanarak tavsiye üretmektedir. Bu sistemlerde amaç müşterinin geçmişte beğendiği ürünlerle en iyi örtüşen ürünlerin bulunmasını sağlamaktır. Bunun içinde geçmişte beğendiği ürünlerin profilini çıkarırken kullanılan anahtar kelimelerden yararlanılır. *OF*'de müşteriye ve ürüne ait bir bilgiler yoktur. Sadece kullanıcının geçmişte değerlendirmede bulunduğu ürünlerin oy değerleri vardır. Hibrid sistemler ise *OF* ve içerik tabanlı filtreleme sistemlerinde kullanılan yöntemlerin karma olarak kullanılmasıyla ortaya çıkmış sistemlerdir.

OF sistemleri, tavsiye sistemleri içinde en yaygın olarak kullanılan tekniklerden biridir. Bu sistem ilk olarak 1992 yılında Goldberg ve ark. tarafından sunulmuştur [3]. *OF* sistemleri bazı yaklaşımlara göre farkı şekillerde gruplanabilir. Bu gruplamalardan

birisi, sistemin kullandığı oy değerlerine göre sayısal oy tabanlı ve ikili oy tabanlı sistemlerdir. Kullanılan varlıkların tipine göre kullanıcı tabanlı ve ürün tabanlı algoritmalardır. Sistemin yaptığı işe göre sayısal tahmin yaklaşımı ve üst-N tavsiye yaklaşımı olarak gruplanabilir. Sistemde kullanılan uzaklık ölçüm yöntemlerine göre ise benzerlik yöntemi kullanan ve güven tabanlı benzerlik yöntemi kullanan sistemler olarak farklı bir gruplama yapılabilir. Ayrıca kullanılan algoritmalara göre bellek tabanlı, model tabanlı ve hibrid yöntemler gibi yaklaşımlara bakılarak da gruplandırılabilir [5]. *OF*'in temel yaklaşımında şu vardır, bir kullanıcının gelecekte beğenebileceği ürünler, geçmişte beğenilen ürünlere göre kendisine en benzer kullanıcıların beğendiği ürünler olabilir, varsayımına dayanır. *OF*'in yaptığı işe göre iki temel amacı vardır. Bunlardan birincisi kullanıcının beğenebileceğini N adet üründen oluşan bir liste sunmaktır ve buna üst-N tavsiyesi denir. Üst-N tavsiye sisteminde sistem kullanıcının değerlendirmeye almadığı ürünler hakkında tahminler oluşturur ve bu tahminlerden en yüksek değerlerden başlayarak N adet ürünü kullanıcıya tavsiye eder. İkincisi de ilgili kullanıcı için bir ürün hakkında kendisinin ne kadar beğeneceği veya sadece beğenip-beğenmeme tavsiyesi ile ilgili olarak sayısal bir tahmin üretmektir. Bu sayısal tahmin belli bir sayı aralığında üretilebileceği gibi sevme veya sevmeme (1,0) olarak da üretilebilir. Bu tezde sayısal tahmin üzerinde çalışma yapılmıştır. Sayısal tahmin üretmede sistemin oy aralığı ne ise tahminde o aralıkta bir değer olur.

OF sistemlerinin üstesinden gelmek zorunda olduğu bazı zorlukları vardır. Bu zorluklardan biri kullanıcılar/müşteriler için en doğru tahminleri üretebilmektir. Tavsiye sistemi yanlış bir tahminde bulunduğunda veya tavsiye verdiğinde ürün pazarlanmış olsa bile kullanıcıda bir güven kaybı meydana gelebilir ve kullanıcı sistemden veya satın aldığı platformdan uzaklaşabilir. Ayrıca sistemden beklenen doğru tahmin üretmenin yanı sıra, bu tahminin de en hızlı şekilde kullanıcıya ulaştırılması gerekmektedir. Model tabanlı algoritmalar sistemin beklediği hızda tahmin üretmeyi amaçlamaktadır ve bu yöntemlerde sistemin hangi hızda çalışabileceğinin analizi yapılabilmektedir. Tahmin üretmeden önce sistem veri setini kullanarak bir model geliştirir ve kullanıcı sistemi ziyaret edip tahmin istediğinde hazır model üzerinden tahmin üretme işlemi yapılır. Ancak bu model belirli periyotlarda tekrar oluşturulmalıdır. Çünkü veri setine yeni dahil olmuş oy değerleri olabilir. Bu da modeli değiştirebilir. Bellek tabanlı algoritmalar da daha doğru tahminler üretebilmekte ancak ölçeklenebilirlik konusunda eksik kalmaktadır. Tavsiye sistemi olarak her iki yöntemin pozitif yönlerinden yararlanarak geliştirilmiş hibrid sistemler de kullanılmaktadır. *OF* karşılaştığı zorluklardan birisi de sisteme yeni dahil olmuş kulla-

nıcılardır. Bu kullanıcılar sisteme yeni dahil olduğu için benzerlik hesaplanabilecek bir verisi olmayacaktır. Sistemin bu tür bir zorluğun da üstesinden gelmesi gerekmektedir. *OF* algoritmalarının sonuçlarının daha iyi olması için seyrekliği düşük bir veri seti olması gerekmektedir. Ancak tavsiye sistemlerinde kullanılan veri setleri oldukça seyrektir. Bu sorununun üstesinden gelmek için veri setleri doldurulmaktadır. Bu tezde doğruluğun iyileştirilmesi için veri setini doldurma yöntemi kullanılmıştır. Ayrıca kullanıcıların dikkat ettiği bir diğer husus gizliliktir. Gizlilik, *OF* sistemlerinin üstesinden gelmesi gereken sorunlardan birisidir. İnsanlar çevrimiçi olarak satın aldıkları ürünlerin veya izledikleri filmlerin bilinmesinden çeşitli nedenlerden dolayı rahatsızlık duyabilir. Araştırmacılarda kullanıcıların bu hassasiyetini dikkate alarak gizliliği korunmuş ortak filtreleme (*GKOF*) yöntemlerini ortaya çıkarmışlardır. *OF*'de gizlilik sorunu ilk olarak 2002 yılında Canny tarafından ortaya atılmıştır [6]. Kullanıcılar etkileşimde bulundukları sisteme bilgilerinin gizliliği konusunda güven duymak istemektedir. Bu durum araştırmaların gizlilik yönüne doğru bir kapı açılmasına neden olmuştur.

OF üzerindeki çalışmalardan bazıları gizlilik sorununa yönelik olmuştur. Bu çalışmalar *GKOF* konusunun ortaya çıkmasına neden olmuştur. Gizlilik algoritmalarının temelde iki amacı bulunmaktadır. Birincisi kişinin satın aldığı veya oy verdiği ürünlerin bilinmemesi, ikincisi de satın alınan ürünlerin veya oy verilen ürünlerin oy değerlerinin bilinmemesidir [4]. *GKOF* çalışmaları üç alt başlık altında gruplanabilir. Bunlar merkezi sunucu-tabanlı sistemler, bölümlenmiş sistemler ve eşten eşe sistemlerdir [4]. Merkezi sunucu-tabanlı sistemler kullanıcının oy değerlerini ve oy verilen ürünleri maskelenmiş olarak tutarlar. Bu sistemler hem ikili oy değerine sahip hem de sayısal oy değerine sahip tavsiye sistemlerinde kullanılabilirler. Merkezi sunucu-tabanlı sistemlerde kullanılan tekniklerin bazıları şunlardır, rasgele karışıklık tekniği (*RKT*), tekil değer ayrışımı-tabanlı, ayrık dalgacık dönüşümü-tabanlı, rastgelenmiş yanıt tekniği ve basit Bayesian sınıflandırıcı-tabanlı tekniklerdir. Rastgelenmiş yanıt tekniği ve basit Bayesian sınıflandırıcı-tabanlı teknikler ikili oya sahip sistemlerde kullanılan tekniklerdir. *RKT*, tekil değer ayrışımı-tabanlı, ayrık dalgacık dönüşümü-tabanlı teknikler ise sayısal oya sahip sistemlerde çalışan tekniklerdir. Bu tezde gizliliği sağlamak ve iyileştirme sağlanan teknik *RKT*'dir. Bunun yanında gizliliği sağlamak için bölümlenmiş sistemlerde kullanılan teknikler de dikey şekilde bölümlenmiş veri, yatay şekilde bölümlenmiş veri ve keyfi bölümlenmiş veri teknikleri olarak üç grupta toplanabilir. Bu tekniklerde daha iyi analiz yapabilmek için daha fazla veriye ihtiyaç duyan firmalar birbirleri arasında veri payla-

şımı yaparlar. Bu paylaşımda bilgi güvenliğini sağlamak için verileri yatay, dikey veya keyfi olarak bölümlendirerek paylaşım işlemini gerçekleştirirler. Eşten eşe sistemler de kullanıcı çiftleri arasında oy değişimine dayalı bir tekniktir. Ayrıca *GKOF* sistemleri kullandıkları tekniğe göre de gruplandırabilir. Böyle bir durumda gruplar şu şekilde olur; randomizasyon, şifreleme teknikleri, anonimleştirme ve toplama, yarı güvenilir üçüncü parti teknikleridir. Gizliliği sağlamak için sunulan tekniklerin ortak amacı gizliliği sağlamanın yukarıda bahsettiğimiz sorunların üstesinden gelmesidir. Maalesef tavsiye için kullanılacak veri setleri oldukça seyrektir. Bu da önerilerin doğruluğunu düşürmektedir. Doğru tavsiyenin yanı sıra hızlı bir tavsiyenin de olması gerekmektedir. Bu yüzden çalışmalar çoğunluğu bu sorunların üzerinde yoğunlaşmıştır [5]. Literatürdeki çalışmalarda kaba kümeler teorisi (*KKT*) yukarıda bahsedilen bazı sorunların çözümünde kullanılmıştır. Bu tezde *KKT*, yukarıda bahsedilen sorunların üstesinden gelmek için kullanılmıştır.

KKT matematiksel bir araç olarak ilk defa Pawlak tarafından 1982'de sunulmuştur [7]. Bu yöntem veri madenciliği, görüntü işleme, sağlık sistemleri, tavsiye sistemleri, bulanık mantık gibi pek çok alanda yaygın bir şekilde kullanılmıştır. Ayrıca bulanık mantık ve *KKT* birlikte kullanılarak bulanık kaba kümeler yöntemi geliştirilmiştir. Bu yöntemle bulanık mantıkla çıkarılan kuralların indirgenmesiyle daha iyi sonuçlar elde edilmiştir. *KKT* sınıflandırma, kümeleme analizi, öznitelik çıkarımı vb. gibi geniş bir kullanım amacı vardır. Gizlilik sorunu veri madenciliği alanında da bulunduğundan, gizliliği korunmuş veri madenciliği yöntemleri geliştirilmiş ve *KKT* bu alanda da kullanılmıştır. *KKT* veri madenciliği için kural çıkarımı, tahmin üretme, öznitelik çıkarma ve boyut indirgeme yapabilmektedir. *KKT*'ne dayalı olarak geliştirilen yöntemlerden birisi de ROUSTIDA algoritmasıdır. Bu algoritma, boşluklu veri setini yine veri setindeki verileri kullanarak doldurmaktadır. Bu sayede tavsiye sistemleri ve veri madenciliği yöntemleri seyrek veri setinden kaynaklanan sorunların üstesinden gelebilmektedir.

Bu tezin içeriği şu şekildedir. İkinci kısımda ilgili çalışmalar yer almaktadır. Bu çalışmalar *KKT* kullanarak seyreklik sorununun üstesinden gelmek ve doğruluğu iyileştirmek için yapılan çalışmalardır. Ayrıca bu bölüm gizliliği korunmuş veri madenciliğinde *KKT* kullanarak yapılan iyileştirme çalışmalarını içerir. Üçüncü kısımda tezde kullanılan algoritmalardan bahsedilecektir. Dördüncü kısımda önerilen yaklaşım açıklanacaktır. Beşinci kısımda yapılan deneylerden ve sonuçlarından bahsedilecektir. Son bölümde ise tezin özeti ve sonuçların değerlendirilmesi yapılacaktır.

2. İLGİLİ ÇALIŞMALAR

OF yöntemlerinin ortak amacı çok fazla ürün seçeneği arasından doğru ürünü tavsiye edebilmektir. Ancak her bir yöntemin belli başlı bazı eksik yönleri vardır. Gizlilik sorununun üstesinden gelen yöntemlerde doğruluk kaybıyla karşılaşılmaktadır. Karşılaşılan diğer sorunlar ise sisteme yeni dahil olmuş ve ürün geçmişi bulunmayan bir kullanıcının tavsiye istediğinde sistemin karşılaştığı zorluk, kendi beğenisini değil de genel oy dağılımına göre değerlendirme yapan kullanıcıların oluşturduğu bir veri setinde kullanıcının karakteristiğinin belirlenememesi, gizlilik, sistemin çevirim içi olarak tavsiye üretme hızındaki düşüklük gibi karşılaşılan zorluklardan bahsedebiliriz. Literatürdeki araştırmalarda *OF* ve veri madenciliğinde doğru sonuç üretirken bahsi geçen zorlukların üstesinden gelmek için *KKT* teorisini kullanan çalışmaları içermektedir. Bu çalışmalar iki alt başlık altında toplanmıştır. İlk olarak Bölüm 2.1'de tavsiye sistemlerinde *KKT* kullanılan çalışmalar bulunmaktadır. Daha sonra Bölüm 2.2'de gizliliği korunmuş veri madenciliğinde *KKT* kullanarak iyileştirme yapan çalışmalardan bahsedilmiştir. Aynı zamanda bu çalışmalar tezde önerilen çalışmanın ortaya çıkmasına katkıda bulunan çalışmalardır.

2.1 Tavsiye Sistemlerinde Kaba Kümeler Teorisi Kullanan Çalışmalar

Pattaraintakorn ve ark. [9] sağlık tavsiye sistemi geliştirmişler. Burada tavsiye üretirken kural-tabanlı uzman sistemlerden yararlanmışlar. Bu kurallar IF-THEN yapısındadır. Hastalık belirtileri alınarak durumlar belirlenir ve ardından karar verilir. Ancak kural-tabanlı sistemlerde değeri ve ağırlığı fazla olmayan bazı kurallarda çıkabilmektedir. Bu tür kurallar doğruluğu olumsuz etkilemektedir. Bu durumdan kurtulmak için *KKT* kullanılmış ve kurallar azaltılmıştır. Sonuç olarak da doğruluğu daha yüksek sonuçlar elde edilmiştir. Bu sonuçlar hastalık teşhisinde, hastaların yaptıracağı testlerin azalmasına, dolayısıyla daha az maliyetli bir süreç geçirmesine neden olmaktadır. Hastalığın tespiti doğrultusunda hangi testler gerekli ise o testlerin yapılması yeterli duruma gelmiştir. Bu çalışmadan anlaşıldığı gibi *KKT* gereksiz bir takım kuralların azaltılmasında kullanılabilmektedir.

KKT gereksiz kuralları azaltabileceği gibi karar vermede kullanılan özniteliklerin de gereksiz olanlarından kurtulmak için kullanılabilir. Haijun ve ark. [10] *KKT* ile özniteliklerin sayısını azaltmışlardır. Önerdikleri yöntemi internet sayfalarının tavsiyesinde kullanmışlardır. Bu tavsiye sisteminde öznitelik değerleri belirli kelimelerden oluşmakta-

dır. Karar tablosuna göre hangi özniteliklerin çıkarılacağına karar verilir. Tavsiye içinse kural çıkarım algoritması kullanılır.

OF'de en çok karşılaşılan sorunlardan birisi de çok seyrek veri setleridir. Veri setinin aşırı seyrek olması doğruluğu doğrudan etkileyen faktörlerden biridir. Nitelikli tavsiyede bulunmak için sistemlerin bu sorunun üstesinden gelmesi gerekmektedir. Bunun için Huang ve Gong [11] seyreklik oranını azaltmak için *KKT*'ne dayalı olarak geliştirilen ROUSTIDA algoritmasını kullanmışlardır. Bu algoritma kullanıcıların oylarını dikkate alarak veri setini doldurmayı amaçlayan bir algoritmadır. Bu sayede seyreklik sorununu çözmeyi başarmışlar ve doğruluk değerini iyileştirmişlerdir.

KKT hibrid sistemlerde de kullanılabilmektedir. Fan ve ark. [12] *KKT*'ni kullanarak tavsiye sistemlerinde kullanılan kümeleme yöntemini geliştirmişlerdir. Çalışmalarında öncelikle oy değerlerine göre ürünleri *KKT* kullanarak sınıflandırmışlar ve ardından kümeleme yöntemi ile tahmin üretmişlerdir. Böylece seyreklik sorununun üstesinden gelinmiş ve daha kaliteli sonuçlar elde edilmiştir.

Seyrek veri setleri nitelikli tahmin üretmenin önündeki engellerden biridir. Bu durum ürün-tabanlı tavsiye sistemleri için de geçerlidir. Bu yüzden Su ve Ye [13] ROUSTIDA algoritmasını kullanarak tahmin kalitesini arttırmaya çalışmışlardır.. Bu çalışmada seyrek olan veri seti ROUSTIDA algoritması ile yoğunlaştırılmış ve daha sonra tahmin üretilmiştir. Sonuç olarak daha kaliteli, doğruluğu yüksek sonuçlar elde etmişlerdir. Sonuçları kullanıcı tabanlı tavsiye sistemleri ile karşılaştırılmıştır.

KKT'nin kullanıldığı amaçlardan birisi de tahmin üretmektir. Su ve ark. [14] bu amaçla *KKT* kullanarak yeni bir tavsiye tekniği önermişlerdir. Bu teknikte *KKT*'nin yalın kümelerinden yararlanılmıştır. Bu yalın kümeler Bölüm 3'de açıklanmıştır.

KKT, ortak filtreleme yöntemlerine entegre edilmiş olarak çalışabilmektedir. Wang ve Tseng [15] *KKT*'ni ortak filtreleme yöntemleriyle birlikte kullanarak boşluklu verilerin tahmini için bir yöntem geliştirmişlerdir. Bu yöntemin adına da ortak filtreleme tabanlı kaba kümeler teorisi adını vermişlerdir. Bu çalışmayı DNA mikro dizi veri seti üzerinde test etmişlerdir. Sonuçlar k-en yakın komşu (*kNN*) algoritması ile karşılaştırıldığında doğruluğun daha iyi çıktığı görülmüştür. Ancak kullanılan veri seti sürekli değerlerden oluşmaktadır. *KKT* de ayrık değerlerde daha iyi sonuç vermektedir. Bunun için araştırmacılar öncelikle veri setini ayrıklaştırmışlardır. Bunun için Eşitlik 2.1'deki yöntemi kullanmışlardır.

$$(1)s(i) = deg_{mak(i)} - deg_{min(i)}/N$$
$$(2)deg_{yeni} = 1 + yuvarla((deg_{orj} - deg_{min(i)})/s(i)) \quad \textbf{(2.1)}$$

Yukarıdaki eşitlikte $deg_{mak(i)}$ i satırında ki en yüksek değeri temsil etmektedir ve $deg_{min(i)}$ i satırındaki en düşük değeri temsil etmektedir. Ayrıca deg_{orj} ve deg_{yeni} orjinal değeri ve ayrıklaştırılmış veriyi temsil etmektedir. Ayrıklaşmış olan değerin aralığı 1 ile N+1 arasındadır. Buradaki N ayrıklaştırma seviyesidir. Bu ayrıklaştırma yöntemi tezde gizlenmiş verilerin dönüşümünde kullanılmıştır. Gizlenen verilerde sürekli değerler olduğundan bu işlem yapılmıştır. Benzer bir çalışmada Zhang ve ark. [16] seyreklik sorunu için yeni bir algoritma önermişlerdir. Önerilen çalışmada *KKT* boşluklu verilerin tahmini için kullanılmıştır. Sonuçlar *kNN* ve *SVD-OF* ile karşılaştırılmıştır. Bu çalışmalara göre doğruluğun daha iyi olduğu görülmüştür.

2.2 Kaba Kümeler Teorisi Kullanarak Önerilen Gizlilik Tabanlı Çalışmalar

OF'de gizlilik sorunu çözüldüğünde bazı ilave sorunlarla karşılaşılmaktadır. Bunlardan bazıları doğruluğun düşmesi, tahmin süresinin artması, kapsamanın yeterince artmamasıdır. Bu yüzden araştırmacılar tavsiye sistemleri ve veri madenciliğinde gizliliği sağlarken bu sorunların da üstesinden gelmeye çalışmaktadır. Jensen ve Shan [17] yüksek boyutlu özniteliğe sahip veri setlerinde *KKT* ile boyut azaltma yöntemlerinden bahsetmişlerdir. Bu yöntemler *KKT* ve bulanık kaba kümelere dayalı yaklaşımlardır. Bu yaklaşımlar, kaba kümelerle öznitelik azaltma, ayırt edilebilirlik matrisi, değişken doğrulukla kaba kümeler, dinamik indirgeme ve bulanık kaba kümelerle öznitelik azaltmadır.

Zhang ve ark. [18] içerik-haberli yaklaşımda kişisel bilgilerin korunması için bir yaklaşım önermiştir. Bu tür bir tavsiye sisteminde tahmin üretmek için bazı kişisel bilgilerin bilinmesi gerekmektedir. Başlangıçta kişiler ile sistem arasında bir gizlilik anlaşması olur. Burada hangi bilgilerin kullanacağının tespiti *KKT* ile yapılır. Sonuçlar karşılaştırıldığında ise daha etkili bir yöntem olduğu anlaşılmıştır. Ancak hangi kişisel bilgilerin daha gerekli olduğunun tespit edilmesi gerekmektedir.

Gizlilik tabanlı veri madenciliği uygulamalarında bölümleme tekniği yaygın olarak kullanılmaktadır. Bu bölümleme işlemi yatay şekilde, dikey şekilde ve rastgele olarak yapılmaktadır. Bu tekniğin amacı birleştirilen verilerde paylaşılan bilgilerin güvenliğini sağlamaktır. Bu sayede şirketler daha rahat bilgi paylaşımında bulunabilmekte ve ana-

liz için yeterince veri setine sahip olabilmektedirler. *KKT*'nin öznitelik azaltma özelliği bu teknik üzerinde doğruluğun ve performansın arttırılmasında araştırmacılar tarafından kullanılmıştır. Ye ve ark. [19], [20] hem yatay bölümlemede hem de dikey olarak bölümlemede *KKT* kullanarak performansı arttırmayı başarmışlardır. Zhou ve ark. [21] benzer şekilde hem dikey bölümlenmiş veri seti hem de yatay bölümlenmiş veri setinde *KKT* kullanarak öznitelik sayısının indirgemişlerdir. Bu sayede hem doğrulukta hem de güvenlikte iyileşme sağlanmış. Hu ve ark. [22] yatay bölümlenmiş veri setinde öznitelik indirgeyerek karmaşıklığı azaltmışlardır. Raju ve ark. [23] indirgeme kümeleri oluşturmuşlardır. Daha sonra bu kümelere basit Bayesian algoritması kullanarak kural çıkarımları yapmışlardır. Bu sayede veri seti kirli, tutarsız ve tekrarlı verilerden kurtulmuştur.

3. KULLANILAN YÖNTEMLER

Bu bölümde tezde kullanılan yöntemler açıklanmıştır. Bu yöntemler *KKT* ve *KKT*'ye dayalı geliştirilen ROUSTIDA algoritması, *OF* sistemlerinde kullanılan bellek tabanlı, model tabanlı ve hibrid yaklaşımlardır. Ayrıca gizliliği sağlamak için kullanılan *RKT*'den bahsedilmiştir.

3.1 Kaba Kümeler Teorisi ve ROUSTIDA

KKT ilk olarak Pawlak tarafından 1982 yılında ortaya atılmıştır. Daha sonraları ise bu teknik makine öğrenmesinden veri madenciliğine pek çok alanda kullanılmıştır. *KKT* temel anlamda küme yaklaşımına dayanmaktadır. Klasik küme yaklaşımında küme elemanları hakkında herhangi bir bilgiye sahip olmadan işlemler yapılırken *KKT* yaklaşımında elemanlar hakkında bazı bilgilere ihtiyaç duyulmaktadır. Bu bilgiler öznitelik olarak adlandırılır. Bu bilgiler doğrultusunda nesneler veya varlıklar üzerinde çeşitli analizler yapılarak çıkarımlar yapılabilmektedir. Temel anlamda *KKT*'nde kullanılan veri setlerine bilgi sistemi adı verilir. Bu bilgi sistemi nesneler kümesinden ve nesnelere ait öznitelik kümesinden oluşur. Bilgi sistemi şu şekilde gösterilir.

$$IS = (U, A) \tag{3.1}$$

Eşitlik 3.1'deki , *U* sonlu nesneler kümesini ve *A* ise sonlu öznitelikler kümesini temsil eder. Her öznitelik kümesinin elemanları ile nesneler arasında $f_a : U \rightarrow V_a$ şeklinde bir bilgi fonksiyonu tanımlıdır. Buradaki V_a özniteliklerin sahip olduğu değerler kümesidir.

KKT'nin önemli özelliklerinden birisi de ayırt edilemezlik ilişkisi çıkarılarak elde edilen yalın kümelerdir. Aynı öznitelik değerine sahip olan nesneler birbirinden ayırt edilemezler. Bu şekildeki nesnelerin oluşturduğu kümelerde yalın kümelerdir. Örneğin $B \subset A$ olsun bu durumda ayırt edilemezlik ilişkisi Ind(*B*) ile gösterilir. İki nesnenin ayırt edilemez olması için, her $b \subset B$ için $b(x_i) = b(x_j)$ ise bu iki nesne *B* öznitelikleri içinde eşdeğer sınıfın elemanlarını yani yalın kümenin elemanlarını oluşturur. Burada x_i i'ninci nesne ve x_j de j'ninci nesneyi temsil etmektedir. *KKT* yaklaşımında nesnelerin bazıları kesinlikle ya bir kümeye aittir ya da o kümeye ait olduğuna kesinlik getirilemez. Buna karar vermek için alt küme yaklaşımına ve üst küme yaklaşımına bakılarak karar verilir. Alt küme yaklaşımında nesneler kesinlikle bulundukları kümeye aittir. Üst küme yaklaşımında nesneler bulundukları kümeye kesin olarak ait değillerdir. Örneğin X nesneler

kümesinin alt kümesi yani $X \subset U$ ve B de öznitelikler kümesinin alt kümesi yani $B \subset A$ olsun. Böyle bir durumda alt küme yaklaşımı formül olarak şu şekilde gösterilir;

$$\underline{BX} = x_i \in U[x_i]_{Ind(B)} \subset X \quad \textbf{(3.2)}$$

Eşitlik 3.2'e göre X kümesinin alt yaklaşımının elemanları x_i nesnelerinden meydana gelir. Nesnelerin bulundukları kümeye kesin olarak aittir diyemediğimiz üst küme yaklaşımı ise $\overline{BX}$ şeklinde gösterilir. Formül olarak üst küme yaklaşımı şu şekildedir;

$$\overline{BX} = x_i \in U[x_i]_{Ind(B)} \cap X \neq 0 \quad \textbf{(3.3)}$$

Üst yaklaşım kümesi ile alt yaklaşım kümesinin farkı da sınır kabul edilir.

$$BNX = \underline{BX} - \overline{BX} \quad \textbf{(3.4)}$$

Yukarıdaki eşitlik ile X'in sınırı bulunmuş olur. Sonuç olarak X'e ait alt yaklaşım kümesi, üst yaklaşım kümesi ve sınır bölgesi hesaplanmış olur.

ROUSTIDA algoritması ise *KKT*'ne dayalı olarak geliştirilmiştir [24]. Temel olarak ayırt edilebilirlik matrisini dikkate alarak nesneler arasındaki benzerliği bulmaya çalışır. Buradan yola çıkarak boş olan hücreleri doldurmayı amaçlar. Ancak tavsiye sistemlerinde kullanılan veri setleri çok seyrek olduğundan bütün hücreleri dolduramamaktadır. Oy değerlerinin dağılımı ne kadar homojen ise doldurma oranı da o kadar yükselmektedir. ROUSTIDA algoritması ayırt edilebilirlik matrisi üzerinden nesnelerin ortak oy verdiği özniteliklerine bakarak ilgili nesnenin özniteliğini doldurur. ROUSTIDA algoritması *OF* yöntemlerinde seyreklik ilk kullanıcı ve ilk ürün vb. sorunlara çözüm olması amacıyla kullanılmaktadır. Bu tezde de *GKOF* yöntemlerinde bu amaçla kullanılmıştır. Aynı zamanda araştırmacılar bu yöntemin doğruluk değerini de iyileştirdiğini göstermişlerdir.

Seyrek bir veri setinde ayırt edilebilirlik matrisi şu şekilde çıkarılır. Eşitlik 3.1'deki bilgi sistemi dikkate alınacak olursa U nesneler kümesi n elemanlı ($U = \{u_1, u_2, ...u_n\}$) ve A öznitelikler kümesi m elemanlı ($A = \{a_1, a_2, ...a_m\}$) olsun. Bu durumda bu nesneler arasındaki ayırt edilebilirlik matrisi nxn tipinde bir kare matrisi olur. Bu matris $M_E(A) = \{M(i,j)\}_{n*n}, 1 \leq i \leq n = |U|$ şekilde gösterilir. Bu ayırt edilebilirlik matrisi nesneler arasındaki öznitelik değerlerin farklı olan değerlerine göre hesaplanır. Bu matrisin hesaplanması şu şekilde olur.

$$M_E(i,j) = \{p : (a_p(u_i) \neq a_p(u_j)) \text{ve} (a_p(u_i \neq *)) \text{ve} (a_p(u_j \neq *)\}$$

$$p = 1, 2, ...m \quad \text{ve} \quad i, j = 1, 2, ...n$$

Yukarıdaki formülde i satırı, j sütunu ve * boş yani oylanmamış hücreyi temsil etmektedir. $a_p(u_i)$ ise i satırındaki p'ninci özniteliğin değerini temsil eder. Bu formüle göre herhangi iki nesne karşılaştırılırken öznitelik değerlerine bakılır. Öznitelik değerleri birbirinden farklı ve bu her iki nesnenin öznitelik değeri boş değilse o öznitelik ayırt edilebilir olarak karşılaştırılan nesnelerin matris hücresine yazılır. Bu şekilde bütün nesneler karşılaştırılarak M_E matrisi oluşturulur. Daha sonra her bir kullanıcı için boşluklu hücreler *MAS* isimli dizilerde tutulur. Bu dizinin elemanları şu şekilde hesaplanır.

$$MAS_i = \{p : a_p(u_i) = *, p = 1, 2, ..., m\}$$

Yukarıdaki eşitlikte i her bir kullanıcıyı p ise öznitelikleri temsil eder. Bütün kullanıcıların boş olan öznitelikleri *MAS* dizilerinde tutulur. Boş hücreler tutulduktan sonra ayırt edilebilirliği olmayan nesneler kümesi oluşturulur. Her bir nesne için ayırt edilebilirliği olmayan yani tamamen kendine benzeyen nesneler NS dizilerinde tutulur.

$$NS_i = \{j : M_E(i, j) = \oslash, i \neq j, j = 1, 2, ..., n\}$$

Daha sonra ROUSTIDA için gerekli olan *MOS* kümesine geçilir. *MOS* bilgi sistemindeki boş olan nesneler kümesidir ve *MAS* dizileri kullanılarak oluşturulur.

$$MOS = \{i : MAS_i \neq \oslash, i = 1, 2, ..., n\}$$

ROUSTIDA yaklaşımının akışı Algoritma 3.1'de gösterilmiştir. Bu algoritmada akış iki adımdan oluşmaktadır. Öncelikle doldurulması istenen veri seti algoritmanın girdisidir. Çıktı olarak da doldurulan veri seti elde edilmiş olur. Algoritmanın ilk adımında gerekli olan nesneler arasındaki ayırt edilebilirlik matrisleri hesaplanır. İkinci adımda ise hesaplanan matrisler kullanılarak doldurma işlemi yapılır. İkinci adımdaki *while* döngüsünün sonlanması için son oluşturulan veri seti ilk bir önceki veri setinin aynı olması gerekmektedir. Algoritma, doldurulacak olan herhangi bir nesne için kendisine benzer nesneler seçer. Bu nesnelerin ilgili öznitelikteki değerleri doldurulacak olan özniteliğin değerini belirler.

Algoritma 3.1 *ROUSTIDA algoritması*

Girdi ve Çıktılar
Girdi: Boşluklu veri seti
$IS^0 = (U^0, A)$
Çıktı: Doldurulmuş veri seti
$IS^s = (U^s, A)$
1. adım. $s = 0$ için başlangıç tüm ayırt edilebilirlik matrisleri hesaplanır.
$M_E^0, MAS_i^0 (i = 1, 2, ..., n) ve MOS^0$
2. adım
while $IS^{s+1} \neq IS^s$ **do**
  for $i \in MOS^s$ **do**
    NS_i hesapla
  end for
  for $i \notin MOS^s$ **do**
    for $(p = 1; p < m; p++)$ **do**
      $a_p(u_i^{s+1}) = a_p(u_i^s)$
    end for
  end for
  for $i \in MOS^s$ **do**
    for $(p = 1; p < m; p++)$ **do**
      if $p \in MAS$ **then**
        if $NS_i == 1$ **then**
          if $j \in NS_i$ **then**
            $a_p(u_i)^{s+1} = a_p(NS_i)$
          end if
        else
          if $(i, j \in NS_i) \wedge (a_p(u_i)) \neq (a_p(u_j) \wedge (a_p(u_i)) \neq * \wedge (a_p(u_j) \neq *$ **then**
            $a_p(u_i) = *$
          else
            $a_p(u_i)^{s+1} = a_p(u_i)^s$
          end if
        end if
      end if
    end for
  end for
end while

3.2 Ortak Filtreleme Yöntemleri

OF'nin çıkış noktası, geçmişte benzer davranışlar sergileyen kişiler gelecekte de benzer davranışlar sergiler fikridir. Bu tanıma göre *OF* sistemleri öncelikle aktif kullanıcı için ona en benzer olan kullanıcıların tespit edilmesi gerekir ve daha sonra hedef ürün (tahmin üretilecek ürün) için benzer kullanıcıların davranışına bakılır. Eğer benzer kullanıcıların ürün hakkındaki görüşleri ne ise aktif kullanıcıya da o doğrultuda tavsiye

üretmeyi amaçlar. Bu sistemlerde girdi olarak bir veri seti kullanılır ve çıktı olarak sayısal tahmin veya üst-N tavsiye listesi oluşturulur. Sistemin girdisi olan veri seti sayısal olarak farklı ölçeklemiş sayı aralıklarından oluşabilir. Örneğin oy değerleri 1 ile 10, 1 ile 5 veya -10 ile +10 arasında olabilir. Ayrıca sayısal olarak değil ikili sistemde (1,0) yani beğenip beğenmeme olarak da olabilir. Tablo 3.1'de oy değerleri 1 ile 5 arasında olan örnek bir veri seti gösterilmiştir. *OF* yöntemleri temel olarak 3 gruba ayrılır. Bunlar bellek-tabanlı, model tabanlı ve hibrid yaklaşımlardır. Bu tezde bellek tabanlı yaklaşımların içinden *kNN* yaklaşımı kullanılmıştır. *kNN* kullanıcı-tabanlı bir *OF* yöntemidir. *kNN*'de aktif kullanıcının diğer kullanıcılarla benzerliği hesaplanır. Benzerliği hesaplamak için farklı teknikler mevcuttur. Bunlardan bazıları Pearson korelasyon metriği, kosinüs benzerlik metriği, Spearman korelasyonu vb. yöntemlerdir [25]. Ancak bu tezde z-puan çarpımları kullanılmıştır. Çünkü kullanılan gizlilik tekniği z-puan değerleri üzerinden gizliliği sağlamaktadır. Bu z-puan hesaplama Eşitlik 3.5'te gösterilmiştir.

$$z_{k,i} = \frac{r_{k,i} - \overline{r}}{\sigma_k} \tag{3.5}$$

Eşitlik 3.5'de i ürünü için k kullanıcısının z-puanı hesaplanmaktadır. Burada k kullanıcıyı, r kullanıcının reytingini, $\overline{r}$ kullanıcının reyting ortalamasını ve σ_k kullanıcının standart sapmasını temsil etmektedir. Herhangi bir kullanıcının z-puan normalizasyonu yapılacak olursa, kullanıcının reyting değerinden ortalaması çıkarılıp standart sapmasına bölünmesi gerekmektedir. İki kullanıcı arasındaki benzerliği hesaplamak için de elde edilen z-puan değerleri çarpılır. Eşitlik 3.6'da kullanıcılar arasındaki benzerliğin ağırlıklandırılması gösterilmiştir.

$$ben_{k,l} = \sum_{i=1}^{m} z_{k,i} * z_{l,i} \tag{3.6}$$

Yukarıdaki formülde k ve l iki kullanıcıyı, m toplam oy verilen ürün sayısını temsil etmektedir. Bu formüle göre benzerlikleri hesaplanan kullanıcılar arasından en yüksek ağırlığa

Tablo 3.1. *Ortak Filtreleme için Örnek Veri Seti*

Kullanıcılar	Titanik	Esaretin Bedeli	Matrix	Testere	Babam ve Oğlum
Salih	4	2	-	3	4
Berk	-	2	5	-	4
Selin	1	2	5	3	-
Zeynep	5	-	-	3	4
Ali	3	2	-	3	-

sahip n tane seçilerek aktif kullanıcıya tahmin üretilir. Tahmin üretilirken Eşitlik 3.7'deki formül kullanılır.

$$t_{k,i} = \overline{r_k} + \sigma_k \frac{\sum_{u=1}^{n} \frac{r_{u,i} - \overline{u}}{\sigma_u} * ben_{k,u}}{\sum_{u=1}^{n} ben_{k,u}} \tag{3.7}$$

Yukarıdaki formülde n seçilen komşuların sayısı, u ise komşuluktaki kullanıcılardır. *OF* için kullanılan veri setlerinin çok seyrek olması tahmin üretmekte bazı sorunları beraberinde getirmektedir. Sisteme yeni katılan bir kullanıcının yeteri kadar reyting geçmişine sahip olamaması nedeniyle komşulukları hesaplanamayabilmektedir. Örneğin binlerce ürün çeşidi olan bir alışveriş sitesinden sadece 5 ürün alıp bunlara oy veren kullanıcının oy yoğunluğu çok düşük olacaktır. Eğer yeteri kadar hedef ürüne oy vermiş kullanıcı bulunamazsa aktif kullanıcı tavsiye sisteminden yararlanamayacaktır. *OF* yöntemlerinden karşılaşılan bu gibi sorunlara soğuk-başlama, seyreklik ve kapsama sorunları denmektedir. Ayrıca bellek-tabanlı *OF* sistemlerinde kullanıcılar sistemde oy verdikleri her ürün sonrasında komşulukları değişebilmektedir. Bu yöntemde komşuluklar çevrimiçi olarak hesaplandığından sistemin ölçeklenmesi sorunu ortaya çıkmaktadır. Aktif kullanıcı için hesaplanan komşuluk kümesi çevrimiçi olduğundan tahmin üretme süresi de model tabanlı sistemlere göre daha uzun sürmektedir. Bu tür sorunların çözümü için model tabanlı sistemler önerilmiştir. Ürün-tabanlı *OF* model-tabanlı *OF* yöntemlerinden birisidir. Bu tezde de model-tabanlı olarak ürün-tabanlı *OF* yöntemi kullanılmıştır. Buradaki fark kullanıcılar arasında komşuluk hesaplaması yerine ürünler arasında komşuluk hesaplanmasının yapılmasıdır [26]. Literatürde benzerlik hesaplamak için kosinüs benzerlik metriği, korelasyon-tabanlı metrik, ayarlı kosinüs benzerlik metriği kullanılmıştır. Ancak bu tezde gizlilik tabanlı sistemler üzerinde çalışıldığından benzerlik için yine z-puan değerleri kullanılmıştır. Ürün-tabanlı *OF* yöntemlerinde komşuluk hesaplama çevrimdışı olarak yapılır. Böylelikle bellek-tabanlı yaklaşımdaki ölçeklenebilirlik sorunu çözülmüş olur. Bunun yanında tahmin üretme hızı da bellek-tabanlı yaklaşımlara göre daha hızlı olmaktadır. Ürünlerin komşuluğun hesaplanmasında Eşitlik 3.8'teki formül kullanılmıştır. Farklı olarak kullanıcıların yerine ürünlerin oyları yer almaktadır.

$$ben_{i,j} = \sum_{k=1}^{n} z_{k,i} * z_{k,j} \tag{3.8}$$

Yukarıdaki formülde $ben_{i,j}$ i ve j ürünleri arasındaki benzerlik ağırlığını temsil etmektedir. Formülde n adet komşu üzerinden tahmin üretilmiştir. Z-puan hesaplaması ise aktif kullanıcının komşuluk kümesindeki ürünlerin z-puan değerleridir. Model-tabanlı sistemler seyrek veri setlerinden, bellek tabanlı sistemlere göre daha az etkilenmektedir.

Ancak model-tabanlı sistemlerde bellek ihtiyacı daha fazla olmaktadır. Doğruluk değeri ise kullanıcı-tabanlı sistemle karşılaştırıldığında birbirlerine yakın olduğu görülmektedir [26]. Ancak burada da sisteme yeni bir ürün dahil olduğunda komşuluk hesaplama sorunu (soğuk başlama) ortaya çıkacaktır. Ayrıca sisteme dahil olan yeni oy değerlerini komşuluk hesaplamasına dahil etmek için komşuluk hesaplamalarının belli periyotlarda güncellenmesi gerekmektedir. Böylelikle yeni oy değerleri sisteme katılmış olur.

Bellek-tabanlı sistemler ile model tabanlı sistemlerin bazı eksikliklerinden dolayı hibrid sistemler geliştirilmiştir. Bu sistemler hem bellek-tabanlı yöntemlerin hem de model-tabanlı yöntemlerin bazı özelliklerini kullanmaktadır. Bu tezde k-ortalama kümeleme tabanlı *kNN* yaklaşımı kullanılmıştır. Bu yaklaşımda öncelikle veri seti kümeleme analizi ile k adet kümeye ayrılmıştır. Bu kümeler birbirine benzeyen nesnelerden oluşmaktadır. Bu aynı zamanda benzerlik hesaplamasını oluşturmaktadır. Bellek-tabanlı yaklaşımda kullanılan komşuluk hesaplama yönteminin yerine kümeleme yaklaşımı yapılmıştır. Ancak bu hesaplama kümeleme yaklaşımında bellek-tabanlı yaklaşımındakinin aksine çevirim dışı olarak gerçekleşmektedir. K-ortalama kümeleme yaklaşımın veri setinin ilk k tane elemanını alıp bunları küme merkezi yapmaktadır. Daha sonraki nesnelerin bu küme merkezlerine olan uzaklığı hesaplanır. Ardından yeni elemanların ortalaması alınarak küme merkezleri değiştirilir. Sonraki adımda tekrar tüm elemanların küme merkezlerine olan uzaklığı hesaplanır ve en yakın oldukları yere atanır. Eğer bir önceki küme ile sonraki kümeler aynı kalıyorsa algoritma sonlanır. Elemanların küme merkezlerine olan uzaklıklarını hesaplamak için farklı metrikler mevcuttur. Bunlardan bazıları Jaccard, Kosinüs, Overlap, Dice vb. metriklerdir. Bu metrikler sayısal değerlere sahip vektörler için geçerlidir. Bu tezde kullanılan metrik ise z-puan tabanlı bir metriktir. Çünkü gizlilik tabanlı tavsiye sistemi z-puan tabanlı çalışmaktadır. Bu metrik Eşitlik 3.9'da gösterilmiştir.

$$d_{k,c} = \sum_{i=1}^{n} z_{k,i} * c_i \tag{3.9}$$

Yukarıdaki eşitlikte k kullanıcıyı, i ilgili ürünü, c küme merkezini ve n ise toplam ürünü temsil etmektedir. Eğer değerler kategorik değerler ise farklı metrikler kullanılır. Kullanıcılar çevrimdışı olarak k tane kümeye ayrıldıktan sonra çevrimiçi olarak aktif kullanıcı için seçilecek olan komşular ait olduğu küme içerisinden belirlenir. Böylece komşuluk hesaplama maliyeti azaltılmış olur. Bu yaklaşımdaki süreç hem bellek tabanlı hem de model tabanlı olduğundan hibrid bir yaklaşım ortaya çıkmış olur. Bu yaklaşım da veri seti seyrekliğinden olumsuz etkilenmektedir. K-ortalama kümeleme yaklaşımının doğru-

luk değerleri hem kullanıcı-tabanlı *OF*'den hem de ürün-tabanlı *OF*'den daha iyi olduğu araştırmacılar tarafından belirtilmiştir [27].

3.3 Gizlilik Tabanlı Ortak Filtreleme Yöntemi

OF sistemleri için gizlilik üstesinden gelinmesi gereken bir sorundur. Bunu ilk olarak 2002 yılında Canny ortaya atmıştır [6]. Kullanıcılar hizmet aldıkları sistemin kişisel bilgilerini veya oy verdikleri ürünlerin bilgilerinin sistem tarafından bilinmesini istemeyebilir. Eğer kullanıcılar kendilerini sisteme karşı daha güvende hissederlerse daha cesur ve gerçekçi oy verebilirler. Bu sorunun üstesinden gelmek için araştırmacılar çeşitli gizlilik yaklaşımlarını *OF* sistemlerine uygulamışlar ve yeni yaklaşımlar ortaya atmışlardır. Bu yaklaşımlar içinde bu tez için Polat ve Du'nun ortaya attığı *GKOF*'de önerilen *RKT* kullanılmıştır [28]. Bu yaklaşım sunucu merkezli olarak çalışmaktadır. Yaklaşımın amacı kullanıcıların oy değerlerinin güvenliğinin sağlanmaktır. Sistem kullanıcıların gerçek oy değerlerini bilmeden kullanıcılar için hedef ürün üzerine tahmin üretebilmektedir. Gizlilik tabanlı çalışmaların temel olarak iki amacı vardır. Birincisi kullanıcıya ait gerçek reyting değerlerinin bilinmemesi, ikincisi de oy verilen ürünlerin ne olduğunun bilinmemesini sağlamaktır. Kullanıcının oy değeri bilinsin veya bilinmesin hangi ürüne oy vermişse o ürünü kullandığı anlamına gelmektedir. Bu da kullanıcılar için gizlilik ihlalidir. *RKT* yaklaşımı her iki amacı da gerçekleştirmektedir. Yaklaşımın temel amacı oy değerlerini maskelemek ve oy verilen ürünlerin dışında rastgele olarak başka ürünlere de değer atamaktır.

RKT'nde maskelenmiş oy değerleri sisteme gönderilir ve bu şekilde tutulur. Kullanıcıların oylarının saklanması temel olarak sayıl çarpım tekniğine dayanır. Örneğin bir $A = (a_1, a_2, ..., a_n)$ ve $B = (b_1, b_2, ..., b_n)$ şeklinde iki vektör olsun. Bu vektörlerden A vektörü $R = r_1, r_2, ..., r_n$ şeklinde ve B vektörü $P = p_1, p_2, ..., p_n$ şeklinde iki vektörle gizlenmiş olsun. R ve P vektörleri $[-\alpha, \alpha]$ arasından tekdüze şekilde dağılmış rastgele değerlerden oluşur. Bu durumda $A' = A + R$ ve $B' = B + P$ olur. Bu iki vektör arasında sayıl çarpım şu şekilde olur.

$$A'.B' = (A + R).(B + P) = \sum_{i=1}^{n}(a_i b_i + a_i p_i + b_i r_i + r_i p_i) \qquad \textbf{(3.10)}$$

Yukarıdaki R ve P vektörleri birbirinden bağımsız olarak $[-\alpha, \alpha]$ arasında üretilmiş değerlerdir. Eşitlik 3.10'a dikkat edilecek olunursa toplamın içindeki son üç değerin rastgele

dağılmış değerlerle ilgili olduğu görülür. Bu rastgele değerlerin bağımsızlığından dolayı $\sum_{i=1}^{n} r_i p_i \approx 0$, $\sum_{i=1}^{n} b_i r_i \approx 0$ ve $\sum_{i=1}^{n} a_i p_i \approx 0$ olur. Sonuç olarak eklenmiş olan rastgele değerlerin sayıl toplam sonucunda yaklaşık olarak eklenmemiş haldeki vektörün sayıl çarpımının sonucuna denktir.

$$\sum_{i=1}^{n}(a_i + r_i)(b_i + p_i) \approx \sum_{i=1}^{n} a_i b_i \tag{3.11}$$

Her iki vektör içinde sayıl çarpımlarının sonucu vektör elemanlarının toplamına yaklaşık olarak eşittir (eşitlik 3.12).

$$\begin{aligned} \sum_{i=1}^{n}(a_i + r_i) &= \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} r_i \approx \sum_{i=1}^{n} a_i, \\ \sum_{i=1}^{n}(b_i + p_i) &= \sum_{i=1}^{n} b_i + \sum_{i=1}^{n} p_i \approx \sum_{i=1}^{n} b_i \end{aligned} \tag{3.12}$$

Bu teknik *OF* yöntemlerinde uygulandığında kullanıcı verileri saklanmış olur. Ancak bu tekniğin uygulaması z-puan normalizasyonu ile yapılmalıdır. Çünkü normalize edilmemiş verilerde gizlenerek tahmin üretildiğinde rastgele eklenen değerlerin aralığının daha da büyümesi doğruluğun daha fazla kötüleşmesine yol açabilir. Eğer rastgele değer aralığı küçük tutulursa bu kez de tam anlamıyla gizlilik sağlanmamış olur. Bu aralık için optimum değer belirlenmelidir. Dağılım için bu tezde tekdüze dağılım yöntemi kullanılmıştır. Dağılım aralığı için de çeşitli yollar mevcuttur [28]. Bunlardan birisi sabit bir aralık belirleyip, bu aralıktaki değerlerin tekdüze bir şekilde rastgele üretilmesidir. Bu yöntemde kullanıcı $[-\beta, \beta]$ arasından rastgele sayılar seçer ve gizliliğini sağlar. İkincisi de belirlenecek olan aralığın da rastgele olarak seçilmesi ve ardından bu aralık içinden tekdüze rastgele sayıların üretilmesidir. Bu seçimde kullanıcı $[0, \beta]$ arasından rastgele bir α sayısı seçer. Ardından kullanıcı $[-\alpha, \alpha]$ aralığından rastgele sayılar seçer. Bu aralıktaki sayıların dağılımı da iki farklı şekilde olabilir [29]. Bu dağılımların birincisi tekdüze dağılım ikincisi de normal dağılımdır. Tezde kullanılan dağılım yöntemi yukarıda da açıklanan tekdüze dağılım yöntemidir. Kullanıcının hangi ürünleri değerlendirdiğinin bilinmemesi için de kullanıcının oy yoğunluğu (d) dikkate alınarak o yoğunluğunun tamamı, yarısı veya çeyreği kadar seçilen boş ürünleri belirlenen aralıktaki rastgele sayılar ile doldurulur [30]. Bütün bu işlemler kullanıcı tarafında gerçekleşir. Sistem sadece gizlenmiş olan kullanıcı-ürün matrisine sahiptir. Rastgele sayıların yerleştirilmesi ile kullanıcının vektörü sistem tarafından veya üçüncü kişiler tarafından çözülemez hale gelmiştir. Böylece kullanıcı sisteme güvenebilecek ve daha rahat hizmet alabilecektir. Ancak bu yöntemde eklenen rastgele sayılar ile tahmin üretildiğinde, bu sayıların eklenmemiş halindeki tah-

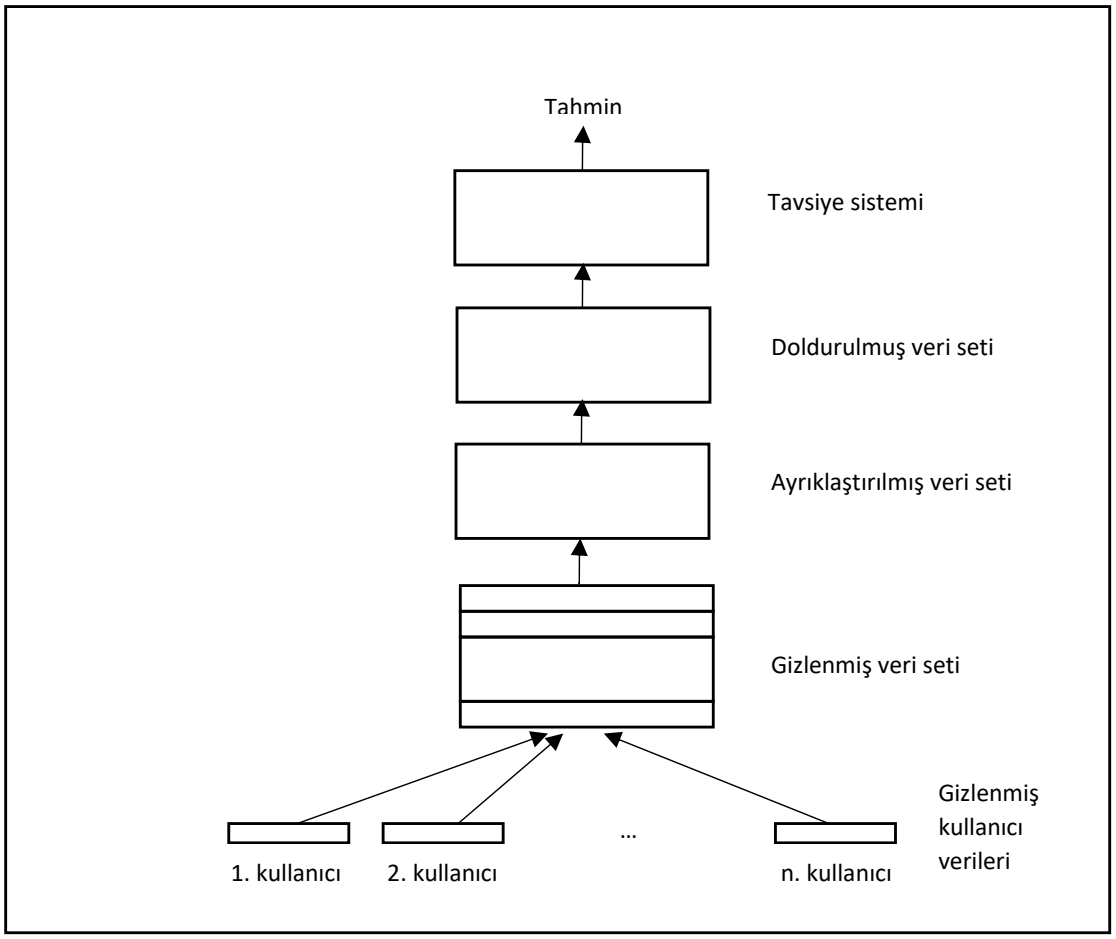
minlerin doğruluk değerleri karşılaştırıldığında doğrulukta kötüleşme olduğu yaptığımız deneylerde görülmüştür. Bunun için bu tezde kullanılan ROUSTIDA yöntemi ile doğruluk değeri iyileştirilmeye çalışılmıştır. *RKT* hem kullanıcı-tabanlı ortak filtreleme yöntemlerinde, hem ürün-tabanlı ortak filtreleme yöntemlerinde, hem de hibrid yaklaşımlarda kullanılabilmektedir. Bu tezde önerilen yöntem üç yaklaşımda da test edilmiştir.

4. ÖNERİLEN YAKLAŞIM

Veri setinin seyrekliği kullanıcı-tabanlı yaklaşımlarda, ürün-tabanlı yaklaşımlarda ve hibrid yaklaşımlarda genel bir sorundur. *OF* yöntemlerinin kullanıldığı veri setleri genellikle çok fazla ürün içerdiğinden (Netflix, Amazon vb.) kullanıcılar tüm ürünler hakkından görüş bildirememektir. Bu da veri setinin seyrek olmasına neden olmaktadır. Ayrıca bir kullanıcının oy verdiği ürünler daha seyrek olarak oylanmışsa başka kullanıcılar ile komşuluk oluşturamamaktadır. Bu durumda sistemin tahmin üretmesi zorlaşmaktadır. Bu sorun genel olarak *OF* yöntemleri için kapsama sorununu ortaya çıkarmaktadır. Yani sistem tahmin isteyen her kullanıcı için hizmet verememektedir. Bunun yanında kullanıcı-tabanlı yaklaşımlarda sisteme dahil edilen her bir reytingin aktif kullanıcının komşularını değiştirebilir. Bu da sistemin ölçeklenmesinde ve performansının test edilmesinde sorun oluşturmaktadır. *OF* sistemlerinin en önemli amaçlarından bir diğeri de kullanıcılara tahmin üretirken en doğru sonucu sunmasıdır. Bunun için komşulukların en iyi şekilde hesaplanması gerekmektedir. Ancak veri setinin seyrek olması bu doğruluğun kötüleşmesine neden olmaktadır. *OF* sistemleri için bir başka sorunda sahte hesaplardır. Kötü niyetli kullanıcılar veya şirketler tavsiye sistemi içindeki ürünlerin komşuluklarını manipüle ederek pazarlamak istediği ürünü popüler hale getirmeye çalışabilir. Başka bir amaç da rakip bir firmanın ürününü popülerliği düşürmeye çalışabilir. Tavsiye sistemleri karşılaştığı bu tür kötücül hesaplara/kullanıcılara karşı sistemi dirençli yapmalıdır. Ayrıca tavsiye sistemlerinin karşılaştığı başka bir sorun da sisteme yeni dahil olan bir ürün veya kullanıcının yeterince oyunun olmamasıdır. Bu sorun da soğuk başlangıç olarak adlandırılır. Böyle bir sorunda sistem kullanıcıya veya ürüne tavsiye üretememektedir. Bu tezin amacı da *GKOF* yöntemlerinin karşılaştığı doğruluk, seyreklik, kapsama performansı ve soğuk başlangıç sorunlarını çözmeye çalışmaktır.

Bu tezde önerilen çalışma dört aşamadan oluşmaktadır. Bu aşamalar *RKT* ile kullanıcı vektörlerinin gizlenmesi, gizlenmiş veri setinin ayrıklaştırılması, ROUSTIDA yöntemi ile veri setinin doldurulması ve *OF* yöntemleri ile tahmin üretilmesidir. Bu aşamalar Şekil 4.1'de gösterilmiştir. İlk olarak kullanıcının verileri *RKT* ile gizlenir. *RKT* ile gizleme işlemi şu aşamalardan oluşur.

1. Kullanıcılar için σ ve β değerleri seçilir.
2. $[0, \sigma]$ arasında rastgele üretilmiş sürekli bir σ_u değeri seçilir.
3. $\alpha = \sigma_u * \sqrt{3}$ ile α değeri hesaplanır.
4. β değerine göre göre doldurulacak hücre sayısı bulunur.



Şekil 4.1. *Önerilen tavsiye sistemi.*

5. $[-\alpha, \alpha]$ arasında doldurulacak hücre sayısı kadar tekdüze değerler üretilir. Bu değerler ilgili hücrelere eklenir.

Gizlenmiş kullanıcı veriler sistemde toplanır. Sistemin elinde kullanıcının z-puan değerlerine ve rastgele boş ürünlere eklenmiş rastgele değerler mevcuttur. Bu değerler sürekli değerlerdir. Boş hücrelerin ROUSTIDA ile doldurulabilmesi için ayrık değerlere sahip olması gerekmektedir. Bu yüzden gizlenmiş değerler ayrıklaştırılır. Eğer veriler ayrık değerlere dönüştürülmezse gerekli olan ayırt edilebilirlik matrisleri oluşturulamayacaktır. Sürekli değerlerde verilerin çoğunlukla farklı olması tüm nesnelerin birbirinden farklı olarak algılanmasına yani benzerliklerinin bulunamamasına neden olur. Bunun için veriler Eşitlik 2.1'deki formül kullanılarak ayrık hale dönüştürülür. Bu aşamadan sonra veriler ROUSTIDA algoritmasının girdisi olarak kullanılır. Algoritma 3.1'e göre veri seti işlenerek doldurma işlemi yapılır. Algoritmanın doldurma performansı veri setindeki değerlerin dağılımından doğrudan etkilenmektedir. Eğer veriler belirli ürünler ve kullanıcılar

üzerinden yoğunlaşmışsa algoritma o bölgedeki eksik verilerin doldurulmasında başarılı olmaktadır. Eğer veriler tüm veri seti üzerinde eşit oranda dağılmış ise doldurma işleminden tüm veri seti yararlanmaktadır. Çünkü nesneler arasında ayırt edilebilirlik matrisi oluşturulurken daha çok nesne veya kullanıcı karşılaştırılabilir. Algoritmanın çıktısı olarak doldurulmuş veri seti oluşturulur. Bu veri seti tavsiye sisteminin kullanacağı veri setidir. Tavsiye sistemi bu veri setindeki değerleri kullanarak kullanıcılara tavsiye üretir. Bu tavsiye sistemi kullanıcı-tabanlı, ürün-tabanlı veya hibrid bir sistem olabilir.

4.1 ROUSTIDA Kullanarak Gizliliği Sağlanmış Kullanıcı-Tabanlı Ortak Filtreleme Yöntemi

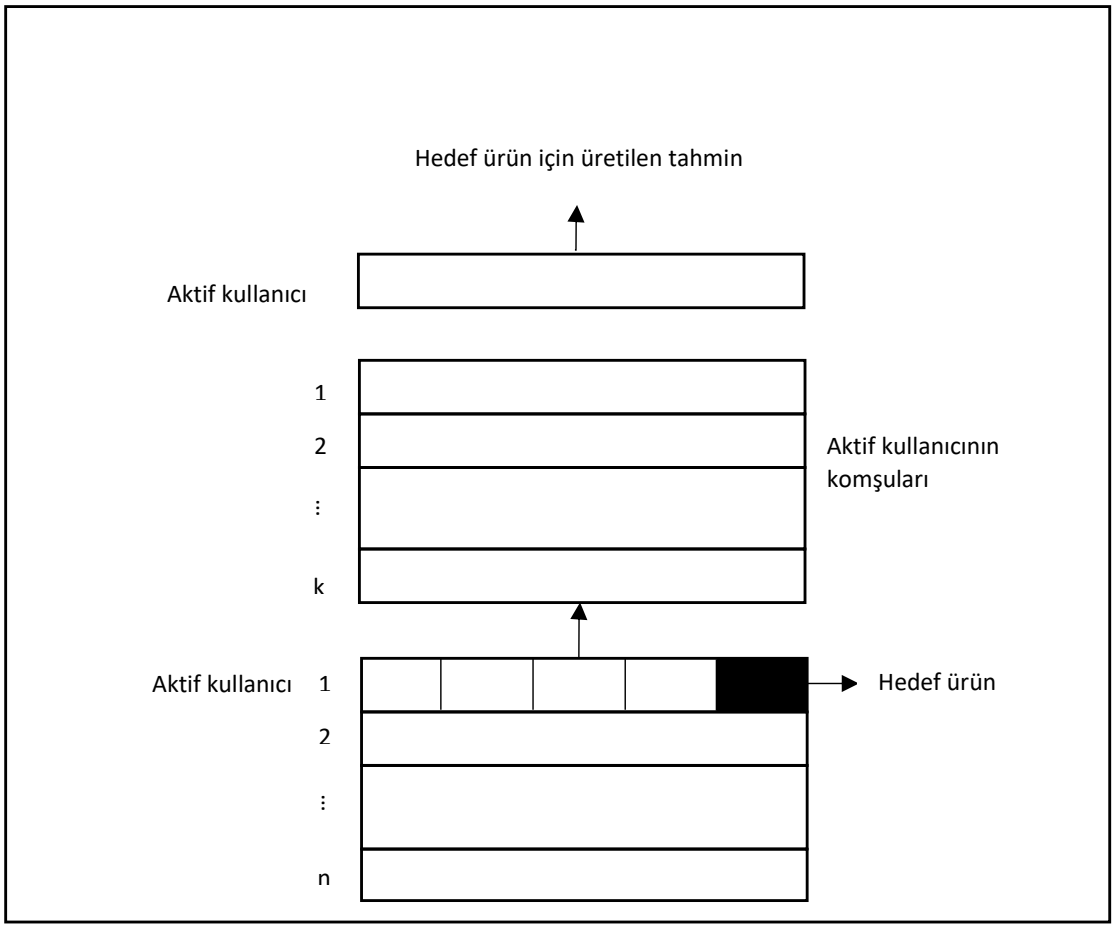
Gizliliği sağlanmış kullanıcı-tabanlı sistemlerde veri setinin seyrekliğinden kaynaklanan doğruluk kaybıyla karşılaşılır. Bunun için ROUSTIDA algoritması ile seyreklik azaltılıp doğruluğun iyileşmesi ve tahmin kapsamının arttırılması sağlanmıştır. Şekil 4.2'de basit bir gizliliği sağlanmış kullanıcı-tabanlı *OF* yöntemi gösterilmiştir. Bu yöntemde kullanılan veri seti ROUSTIDA ile doldurulmuş bir veri setidir. Burada önerilen yöntemin aşamaları şu şekildedir;

1. Kullanıcıların verileri *RKT* ile gizlenir.
2. Gizlenmiş veriler Eşitlik 2.1 ile ayrıklaştırılır.
3. Ayrıklaştırılmış olan veri seti ROUSTIDA ile doldurulur.
4. Gizlenmiş ve doldurulmuş veri seti üzerinden aktif kullanıcının k tane komşusu çıkarılır.
5. Belirlenen komşular üzerinden hedef ürün için Eşitlik 4.2 kullanılarak tahmin üretilir.

Bu yöntemde gizlenmiş veriler ayrıklaştırıldıktan sonra doldurulmuş ve ardından tekrar Eşitlik 3.5 yardımıyla z-puan değerleri hesaplanmıştır. Komşuluklar hesaplanırken bu z-puan değerleri üzerinden hesaplanır. Buradaki aktif kullanıcı ile diğer kullanıcılar arasında ağırlık hesaplanırken Eşitlik 4.1 kullanılır.

$$ben_{a,k} = \sum_{i=1}^{n} z'_{a,i} * z'_{k,i} \quad \textbf{(4.1)}$$

Yukarıdaki eşitlikte $z'_{a,i}$ ve $z'_{k,i}$ aktif kullanıcının doldurulduktan sonraki z-puan değeri ve k kullanıcısının doldurulduktan sonraki z-puan değeridir. Toplam ürün sayısı da n ile



Şekil 4.2. *Kullanıcı-Tabanlı Ortak Filtreleme*

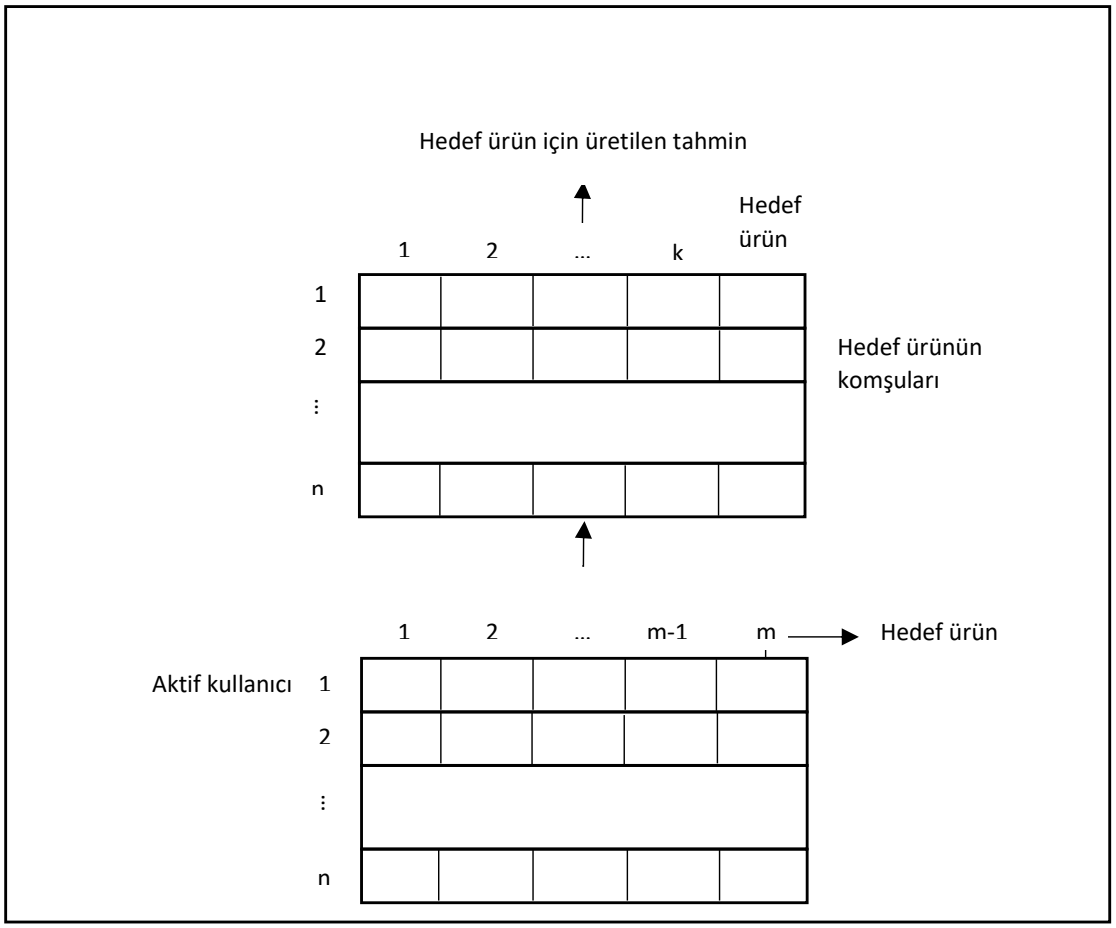
ifade edilmektedir.

$$t_{a,k} = \overline{r_a} + \sigma_a \frac{\sum_{u=1}^{n} ben_{a,u} z'_{u,k}}{\sum_{u=1}^{n} ben_{a,u}} \quad \textbf{(4.2)}$$

Yukarıdaki formülde a kullanıcısının k ürünü için tahmin üretilir. Buradaki z-puan değerleri gizlenmiş verilerdir.

4.2 ROUSTIDA Kullanarak Gizliliği Sağlanmış Ürün-Tabanlı Ortak Filtreleme Yöntemi

Gizliliği sağlanmış ürün-tabanlı tavsiye sisteminde benzerlikler ürün-ürün çiftleri arasında yapılır. Buradaki hesaplamada Eşitlik 3.8 kullanılır. Benzerliklerin veya komşulukların çıkarılması çevrimdışı olarak yapılır. Sisteme dahil olan yeni oyların hesaba katılması için belirli periyotlarda güncellemenin yapılması gerekmektedir. Bu yöntemin işleyiş aşamaları şu şekildedir.



Şekil 4.3. *Ürün-Tabanlı Ortak Filtreleme*

1. Ürünlerin verileri *RKT* ile gizlenir.
2. Gizlenmiş veriler Eşitlik 2.1 ile ayrıklaştırılır.
3. Ayrıklaştırılmış olan veri seti ROUSTIDA ile doldurulur.
4. Ayrıklaştırılmış olan veri setinin tekrar Eşitlik 3.5 ile z-puan değerleri hesaplanır.
5. Hedef ürünün komşuları Eşitlik 3.8 ile hesaplanır.
6. Belirlenen komşular üzerinden n tanesi seçilir ve Eşitlik 4.3 kullanılarak tahmin üretilir.

$$t_{a,k} = \overline{r_a} + \sigma_a \frac{\sum_{i=1}^{n} ben_{i,k} z'_{a,i}}{\sum_{i=1}^{n} ben_{i,k}} \quad \textbf{(4.3)}$$

Yukarıdaki eşitlikte $ben_{i,k}$ hedef ürün olan k ürünü ile komşuları arasındaki benzerliktir. n ise hedef ürünün komşularının sayısıdır. Eşitlikteki $z'_{a,i}$ değeri ise hedef ürünün komşularının üzerindeki z-puan değerleridir. *OF*'de kullanılan *RKT* yönteminde bütün işlem gizlenmiş veriler üzerinden yapılır. Önerilen yöntemde gizlenmiş veriler ayrıklaştırılır, ardından tekrar z-puan değerleri hesaplanır. Bu şekilde tahmin üretilir. Ayrıca *RKT*'ne

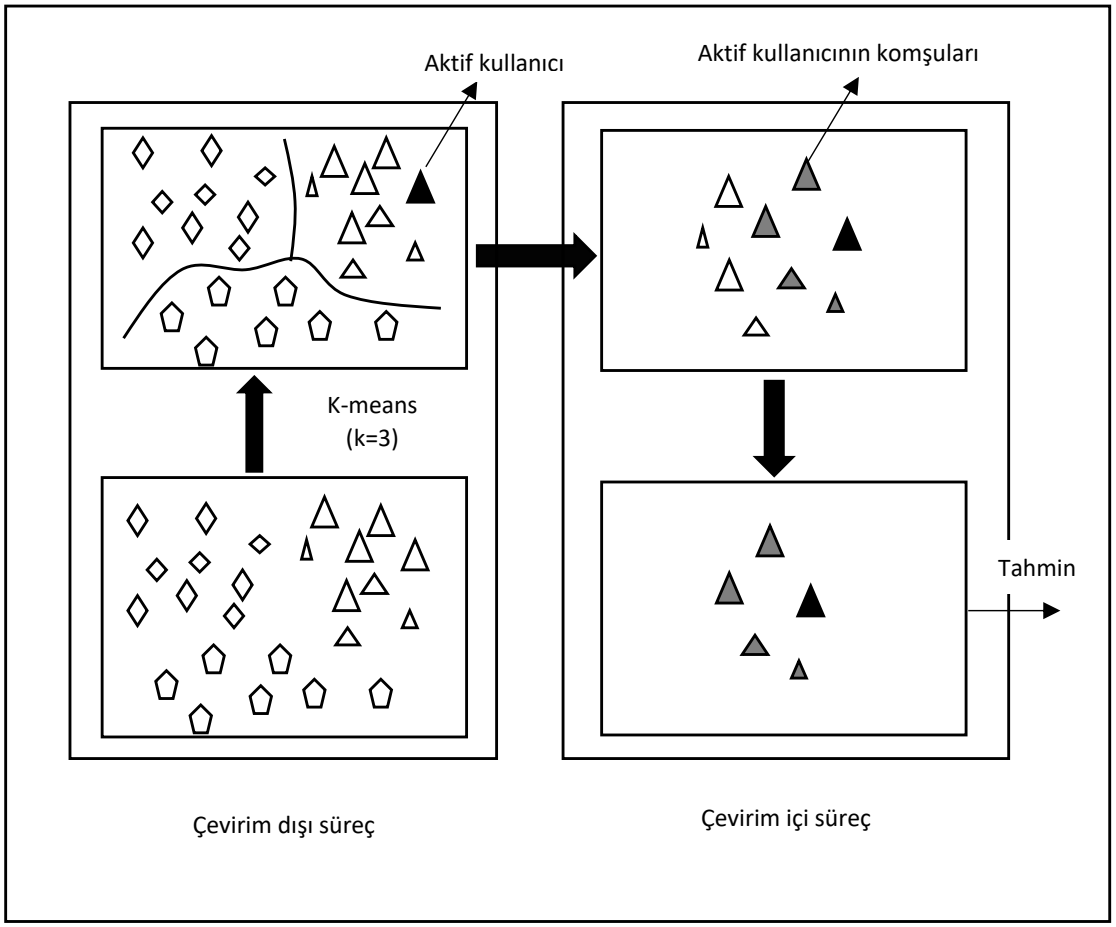
dayalı *GKOF* yönteminde boş olan veriler kullanıcının ortalaması ile doldurulur. Önerilen sistemde ise bu yöntem kullanılmaz.

4.3 ROUSTIDA Kullanarak Gizliliği Sağlanmış Kümeleme-Tabanlı Ortak Filtreleme Yöntemi

Kümeleme-tabanlı yaklaşımlarda süreç hem çevrimiçi hem de çevrimdışı olarak gerçekleşir. Çevrimdışı süreçte veri seti k tane kümeye ayrılır. Daha sonra çevrimiçi olarak seçilen aktif kullanıcının ait olduğu kümeler içinden komşular seçilir. Daha sonra tahmin üretme aşamasına geçilir. Gizliliğin işleminin yapılması en baştaki işlemdir. Genel olarak sürecin aşamaları şu şekildedir.

1. Kullanıcıların verileri *RKT* ile gizlenir.
2. Gizlenmiş veriler Eşitlik 2.1 ile ayrıklaştırılır.
3. Ayrıklaştırılmış olan veri seti ROUSTIDA ile doldurulur.
4. Ayrıklaştırılmış olan veri setinin Eşitlik 3.5 kullanılarak z-puan değerleri hesaplanır.
5. Kullanıcılar k-ortalama yöntemi ile k adet kümeye ayrılır. Burada küme merkezi ile kullanıcıların mesafeleri hesaplanırken Eşitlik 3.9 kullanılır.
6. Aktif kullanıcının n adet komşusu ait olduğu küme içerisinden çıkarılır.
7. Belirlenen komşular üzerinden Eşitlik 4.2 kullanılarak tahmin üretilir.

Bu yöntemde çevrimiçi olarak gerçekleşen süreç kullanıcı-tabanlı yönteme göre daha hızlı sonuç üretmektedir. Çünkü komşuluk hesaplarken aktif kullanıcının diğer tüm kullanıcılarla arasındaki benzerliğe değil sadece kendi kümesindeki kullanıcılarla arasındaki benzerliğe bakılır.



Şekil 4.4. *Kümeleme-Tabanlı Ortak Filtreleme*

5. DENEYSEL SONUÇLAR

Önerilen yaklaşım üç farklı yöntem üzerinde test edilmiştir. Bunlar kullanıcı-tabanlı tavsiye sistemi, ürün-tabanlı tavsiye sistemi ve hibrid bir yaklaşım olan kümeleme-tabanlı tavsiye sistemidir. Bunun nedeni önerilen yaklaşımın hangi tür yaklaşımlarda daha iyi sonuç verdiğinin tespit edilebilmesidir. Deney sonuçlarında hata değeri ve yaklaşımların kapsama performansı karşılaştırılmıştır. Hata değerini ölçmek için Ortalama Mutlak Hata (*OMH*) yöntemi ve Hatalar Kareler Ortalamasının Karekökü (*HKOK*) yöntemi kullanılmıştır. *OMH* değeri şu şekilde hesaplanır;

$$OMH = \frac{1}{n}\sum_{i=1}^{n}|g_i - t_i| \tag{5.1}$$

Yukarıdaki formülde n toplam üretilen tahmin sayısı, g_i gerçek değerleri ve t_i ise tahmin edilen değeri ifade eder. *HKOK* yöntemi ise şu şekildedir;

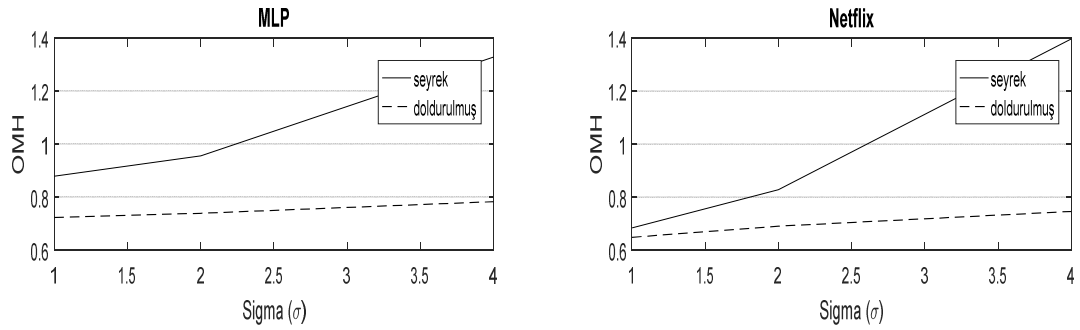
$$HKOK = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(g_i - t_i)^2} \tag{5.2}$$

OMH yönteminde bazı uç değerler ortalamada kaybolurken, *HKOK* yönteminde ise üretilen tahmin ile gerçek değer arasındaki fark daha fazla cezalandırılmaktadır. Bu nedenle deney sonuçları iki farklı yöntemle ölçülmüştür. Deney için iki farklı veri seti kullanılmıştır. Bunlar MLP ve Netflix veri setleridir. MLP veri seti 943 kullanıcı ve 1682 üründen oluşur. Burada ürünler kümesi filmlerden oluşmaktadır. Her bir kullanıcının en az oylamış olduğu 20 ürün vardır. Buradaki oy değerleri 1 ile 5 arasında ayrık değerlerdir. 5 değeri en yüksek beğeniyi, 1 değeri de en düşük beğenmeme değerini belirtmektedir. MLP veri setindeki toplam oy sayısı 100.000'dir. Bu yüzden bu veri setinin seyrekliği $1 - (100000/943 * 1682) = 0,937$'dir. Netflix veri seti 480.189 kullanıcı ve 17.770 üründen oluşur. Bu veriler Ekim 1998 ve Aralık 2005 arasında toplanmıştır. Burada da ürünler kümesi filmlerden oluşmaktadır. Toplam oy sayısı da 100.480.507'dir. Bu veri setindeki oy değerleri de 1 ile 5 arasında ayrık değerlerdir. 5 en yüksek beğeniyi, 1 de en düşük beğenmemeyi gösterir. Buna göre Netflix için veri seyrekliği 0,988'dir. Bu durumda MLP veri setinin yoğunluğu Netflix veri setinden daha fazladır. Bu veri setinden veri seti yoğunluğu ve her bir reyting değerinin yoğunluğu değişmeden bir alt küme seçilmiştir. Bu veri setinden 10000 kullanıcı ve 4000 ürün bulunmaktadır. Daha sonra bu alt küme veri setinden rastgele seçilmiş olan 3000 kullanıcı üzerinden deney gerçekleştirilmiştir. İki farklı veri setiyle deney yapılmasındaki amaç, yoğunluklar değiştiğinde sonuçların nasıl değiştiğinin görülebilmesidir.

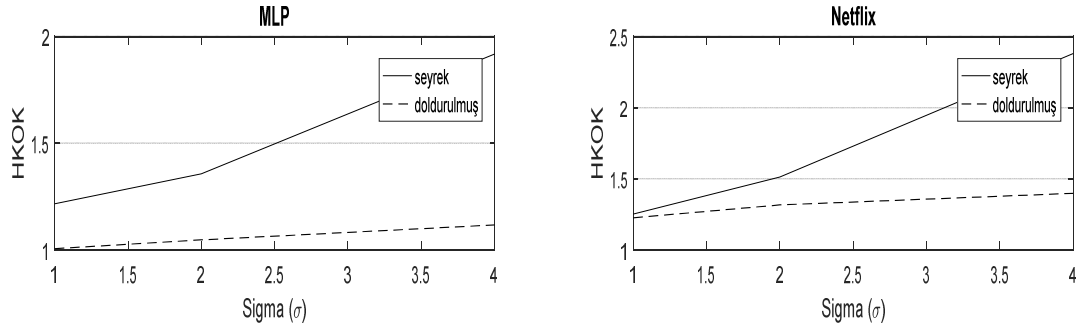
Deney yöntemi olarak MLP veri seti için 243 adet kullanıcı test veri seti ve kalan 700 kullanıcı da eğitim veri seti olarak rastgele seçilmiştir. Diğer önerilen yaklaşımlarla da aynı test ve eğitim veri setleri kullanılmıştır. Test veri setindeki tüm oylar için tahmin üretilmiştir. Üretilen sonuçlar *OMH* ve *HKOK* hata metrikleri ile ölçülmüştür. Netflix veri seti için de 1000 adet kullanıcı test verisi ve 2000 adet kullanıcı da eğitim veri seti olarak rastgele seçilmiştir. Bu veri setleri tüm yöntemler üzerinde kullanılmıştır. Deney grafiklerinde karşılaştırmada kullanılan veri setleri seyrek ve doldurulmuş olarak açıklanmıştır. Buradaki seyrek veri seti ROUSTIDA ile doldurmanın yapılmadığı veri setini ifade ederken, doldurulmuş veri seti de ROUSTIDA ile doldurmanın yapılmış olduğu veri setini ifade eder. Deneyler hem gizlilik sağlanarak hem de gizlilik olmadan gerçekleştirilmiştir. Gizliliği sağlanmış kullanıcı-tabanlı *OF* yöntemleriyle yapılmış deneyler ve sonuçları Bölüm 5.1'de, gizliliği sağlanmış ürün-tabanlı *OF* yöntemleriyle yapılmış deneyler ve sonuçları Bölüm 5.2'de ve gizliliği sağlanmış kümeleme-tabanlı *OF* yöntemleriyle yapılmış deneyler ve sonuçları Bölüm 5.3'de açıklanmıştır.

5.1 Gizliliği Sağlanmış Kullanıcı-Tabanlı Ortak Filtreleme Yönteminin Deney Sonuçları

Kullanıcı-tabanlı ortak filtreleme yöntemlerinde en yakın komşu sayısını belirlerken optimum değer olarak 40 komşu seçilmiştir. Test kullanıcısı için eğitim veri seti içinden en yakın 40 adet komşu alınmıştır. Gizlilik için 7 farklı deney yapılmıştır. Öncelikle α değerini belirlerken kullandığımız σ değerleri seçilmiştir. Bu değerler 1,2 ve 4 olarak belirlenmiştir. Daha sonra $[0, \sigma]$ arasında rastgele bir σ_2 sayısı seçilir. Kullanıcıların oylarına eklencek olan sayıların aralığı olan α sayısı rastgele seçilen σ_2 sayısının $\sqrt{3}$ katıdır. Yani $\alpha = \sigma_2 * \sqrt{3}$'tür. Daha sonra σ değeri 2'de sabit tutularak β değeri değiştirilmiştir. Bu β değeri kullanıcıların boş olan ürünlerinin ne kadarının doldurulacağına karar veren değerdir. β değeri sırasıyla 50,100,200 ve 400 olarak belirlenmiştir. β'nın 50 seçilmesi kullanıcının oy yoğunluğunun $1/4$'ü, 100 değeri $1/2$'si, 200 değeri yoğunluğu kadar ve 400 değeri de yoğunluğun iki katı kadar oy vermediği ürüne rastgele değer eklendiği anlamına gelir. Yani kullanıcının oy yoğunluğu d ise tekdüze olarak rastgele eklenecek sayı 50'de $d/4$, 100'de $d/2$, 200'de d ve 400'de $2d$'dir. Seçilecek σ değerinin büyümesi α değerini büyüteceğinden gizliliğin sağlanmasında doğrudan etkilidir. Bu değer büyüdükçe hata oranı da artmaktadır. Şekil 5.1'de ise ROUSTIDA ile doldurulmuş veri setlerinin hatalarının



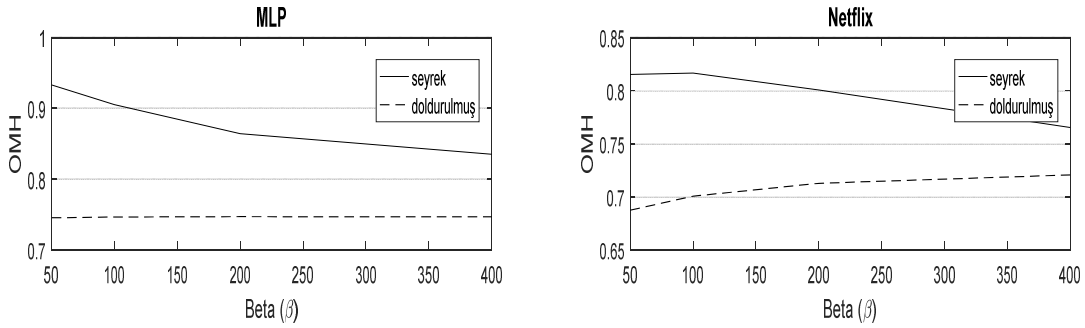
Şekil 5.1. *Gizliliği Sağlanmış Kullanıcı-Tabanlı Ortak Filtrelemede* $\beta = 0$ *ve Değişen* σ *Değerleri için Hata*



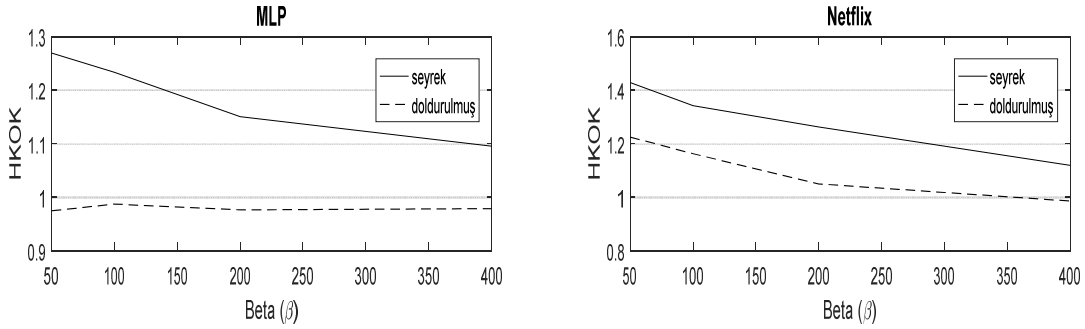
Şekil 5.2. *Gizliliği Sağlanmış Kullanıcı-Tabanlı Ortak Filtrelemede* $\beta = 0$ *ve Değişen* σ *Değerleri için Hata*

karşılaştırılması yapılmıştır. Buna göre her iki veri seti içinde önerilen yaklaşımın daha başarılı olduğu görülmektedir. Burada gizlilik sağlanırken σ değeri kullanılmıştır. Şekilde σ değeri arttıkça hata değerinin de arttığı görülmektedir. Hata metriği olarak bu şekilde *OMH* ile karşılaştırılmıştır. Şekil 5.2'e göre ise karşılaştırmada *HKOK* metriği kullanılmıştır. Burada da doldurulmuş olan her iki veri setinde de önerilen yaklaşımın daha başarılı olduğu görülmektedir. Yine aynı şekilde σ değeri arttıkça hata değerinin de arttığı görülmüştür. Bu sonuçlara göre optimum σ değeri olarak 2 seçilmiş ve bundan sonraki β değerlerinin etkisinin ölçülmesinde σ değeri 2'de sabit tutulmuştur. Eğer daha büyük σ değeri seçilirse hata oranı çok fazla artmaktadır. Daha küçük seçildiğinde ise gizlilik seviyesi azalmaktadır. Şekil 5.3'de gizlilik için β değerinin değişiminde hata değerinin nasıl etkilendiğine bakılmıştır. Burada karşılaştırma *OMH* ile yapılmıştır. Sonuçlara göre önerilen doldurma yöntemi sonucunda hatanın azaldığı görülmektedir.

Şekil 5.3'de ise karşılaştırmalar *HKOK* ile yapılmıştır. Burada da yine önerilen yaklaşım başarılı olmuştur. Ayrıca önerilen yaklaşımda çıkan tüm sonuçlar Şekil **??** ile kar-



Şekil 5.3. *Gizliliği Sağlanmış Kullanıcı-Tabanlı Ortak Filtrelemede σ = 2 ve Değişen β Değerleri için Hata*



Şekil 5.4. *Gizliliği Sağlanmış Kullanıcı-Tabanlı Ortak Filtrelemede σ = 2 ve Değişen β Değerleri için Hata*

şılaştırıldığında önerilen yaklaşımın gizlilikten dolayı düşmüş olan doğruluk değerini iyileştirdiği görülmektedir. Gizlilik olmadan sadece kullanıcı-tabanlı olarak gerçekleştirilen deneyde MLP veri seti için hata değerleri $OMH = 0,7606, HKOK = 1,0871$ olarak bulunmuştur. Bu durumda hatalar karşılaştırıldığında bazı gizlilik parametrelerinde (σ, β) çıkan hata sonuçlarının gizlilik olmadan çıkan hata değerlerinden daha iyi düşük olduğu görülmüştür. Eğer bu parametreler kullanılarak bir kullanıcı-tabanlı *OF* sistemi geliştirilirse hem doğruluk iyileşmiş hem de gizlilik sağlanmış olur. Hataların karşılaştırılmasının yanı sıra önerilen yaklaşımın kapsama performansı da ölçülmüştür. Bunun için $1 - \frac{t}{T}$ formülü kullanılmıştır. Burada t üretilemeyen tahmin sayısını ve T ise üretilmesi gereken tahmin sayısını ifade etmektedir. Tablo 5.1 ve Tablo 5.2'e bakıldığında bazı parametreler için kapsama performansının önerilen yaklaşımda iyileştiği görülmektedir. Ancak bazı parametreler üzerinde ise daha düşük performans göstermiştir. Ayrıca sistem σ = 2 ve β = 200, σ = 2 ve β = 400 parametrelerinde bütün tahminleri ürettiği görülmektedir.

Tablo 5.1. *MLP veri seti üzerinde Gizliliği Sağlanmış Kullanıcı-Tabanlı Ortak Filtreleme ile Gizliliği Sağlanmış Kullanıcı-Tabanlı Önerilen Ortak Filtrelemenin kapsama performansının karşılaştırılması.*

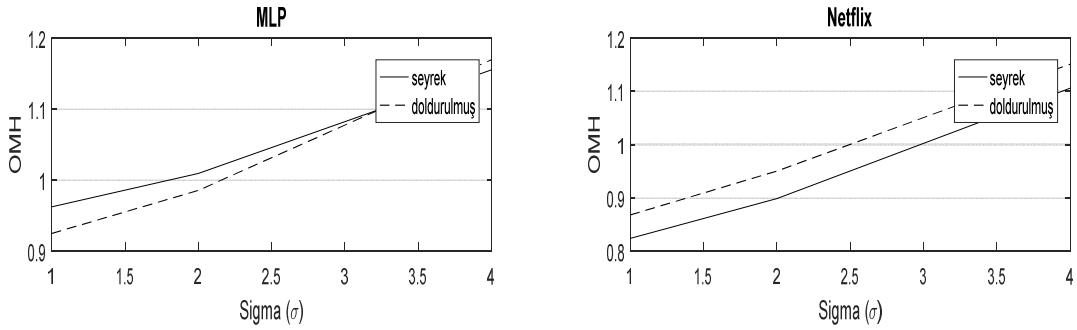
Parametreler	Gizliliği Sağlanmış Kullanıcı-Tabanlı OF	Önerilen Yaklaşım
σ = 1 ve β = 0	0,984	0,989
σ = 2 ve β = 0	0,981	0,983
σ = 4 ve β = 0	0,978	0,978
σ = 2 ve β = 50	0,994	0,999
σ = 2 ve β = 100	0,998	0,999
σ = 2 ve β = 200	1	0,998
σ = 2 ve β = 400	1	0,999

Tablo 5.2. *Netflix veri seti üzerinde Gizliliği Sağlanmış Kullanıcı-Tabanlı Ortak Filtreleme ile Gizliliği Sağlanmış Kullanıcı-Tabanlı Önerilen Ortak Filtrelemenin kapsama performansının karşılaştırılması.*

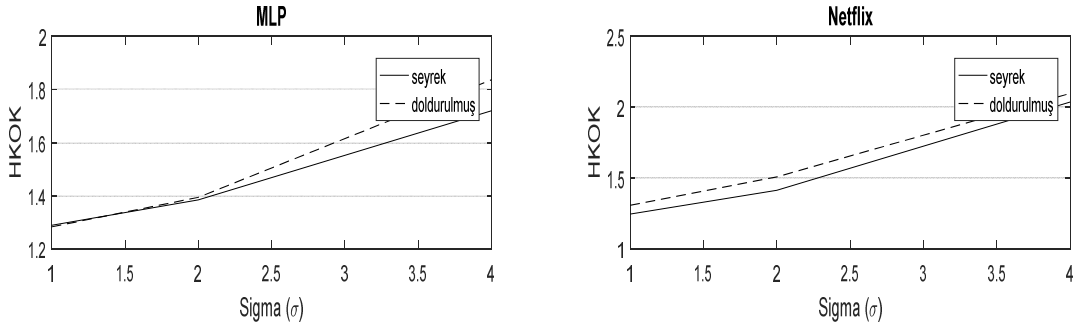
Parametreler	Gizliliği Sağlanmış Kullanıcı-Tabanlı OF	Önerilen Yaklaşım
σ = 1 ve β = 0	0,939	0,935
σ = 2 ve β = 0	0,929	0,923
σ = 4 ve β = 0	0,915	0,917
σ = 2 ve β = 50	0,951	0,944
σ = 2 ve β = 100	0,966	0,957
σ = 2 ve β = 200	0,982	0,980
σ = 2 ve β = 400	0,996	0,995

5.2 Gizliliği Sağlanmış Ürün-Tabanlı Ortak Filtreleme Yönteminin Deney Sonuçları

Ürün-tabanlı *OF* yönteminde seçilen optimum komşuluk sayısı 30'dur. Aktif ürüne eğitim veri setinden 30 adet komşu çıkarılarak tahmin üretilmiştir. Hata karşılaştırmaları için *OMH* ve *HKOK* yöntemleri kullanılmıştır. Şekil 5.5'de *OMH* metriği ile karşılaştırılmıştır. Buradaki sonuçlarda, MLP veri seti için iyileşme görülürken Netflix veri setinde sonuçların başarısız olduğu görülmüştür. Benzer sonuçlar Şekil 5.6'deki değerlerde de söz konusudur. Burada da yine Netflix veri setinde iyileşmeden söz edilememektedir. Yine burada da σ değeri arttıkça hatanın arttığı gözlenmektedir. MLP veri setinden gizlilik olmadan elde edilen hata değerleri $OMH = 0,7634$ ve $HKOK = 1,0638$'dir. Bu değerler gizlilik sonucunda elde edilen hata değerleri ile karşılaştırıldığında hatanın art-



Şekil 5.5. *Gizliliği Sağlanmış Ürün-Tabanlı Ortak Filtrelemede $\beta = 0$ ve Değişen σ Değerleri için Hata*



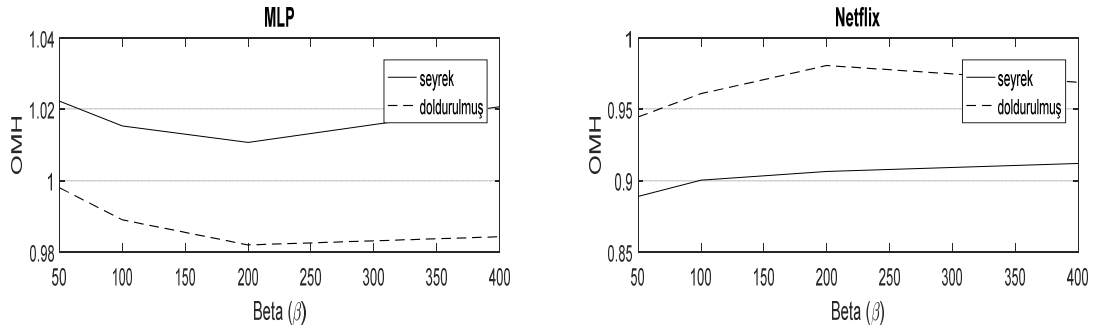
Şekil 5.6. *Gizliliği Sağlanmış Ürün-Tabanlı Ortak Filtrelemede $\beta = 0$ ve Değişen σ Değerleri için Hata*

tığı görülmektedir. Önerilen yaklaşım MLP veri seti için hata değerlerini azaltmaktadır. Ancak gizlilikten kaynaklanan farkı tamamen ortadan kaldıramamaktadır. Netflix veri setinde ise iyileşme olmamaktadır. Netflix veri setinden gizlilik olmadan elde edilen hata değerleri $OMH = 0,7630$ ve $HKOK = 1,2430$'dur.

Şekil 5.7 ve Şekil 5.8'da β değerine göre hata performanslarına bakılacak olursa burada da MLP veri seti üzerinde önerilen yaklaşımın başarılı olduğu görülmektedir. Ancak Netflix veri setinde hem *OMH* hem de *HKOK* metriğine göre sonuçlar başarısız olmuştur. Kapsama performansına gelince yine önerilen yaklaşım doldurulmamış haldeki performansın gerisinde kalmıştır. Buradaki sonuçlar Tablo 5.3 ve Tablo 5.4'te gösterilmiştir. Buna göre önerilen yaklaşım sadece MLP veri seti üzerinde bazı parametreler için daha fazla tahmin üretebilmiştir. Ancak aradaki farklar incelendiğinde bu performansta büyük oranda bir kayıp olmadığı görülecektir. Kapsama performansının sonuçları değerlendirildiğinde MLP veri seti için β parametresinin artması kapsama performansını arttırdığı görülmüştür. Bu durum Netflix veri setinde geçerli değildir.

Tablo 5.3. *MLP veri seti üzerinde Gizliliği Sağlanmış Ürün-Tabanlı Ortak Filtreleme ile Gizlilik-Tabanlı Önerilen Ortak Filtrelemenin kapsama performansının karşılaştırılması.*

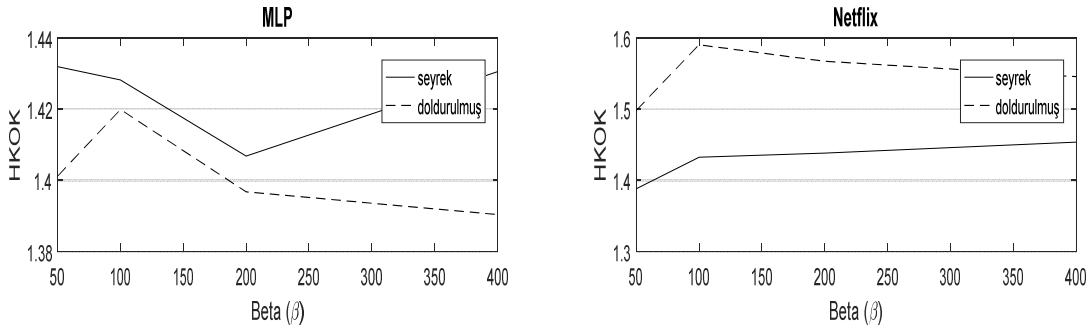
Parametreler	Gizliliği Sağlanmış Ürün-Tabanlı OF	Önerilen Yaklaşım
σ = 1 ve β = 0	0,995	0,991
σ = 2 ve β = 0	0,995	0,991
σ = 4 ve β = 0	0,993	0,987
σ = 2 ve β = 50	0,995	0,992
σ = 2 ve β = 100	0,996	0,992
σ = 2 ve β = 200	0,994	0,992
σ = 2 ve β = 400	0,990	0,992



Şekil 5.7. *Gizliliği Sağlanmış Ürün-Tabanlı Ortak Filtrelemede σ = 2 ve Değişen β Değerleri için Hata*

Tablo 5.4. *Netflix veri seti üzerinde Gizliliği Sağlanmış Ürün-Tabanlı Ortak Filtreleme ile Gizlilik-Tabanlı Önerilen Ortak Filtrelemenin kapsama performansının karşılaştırılması.*

Parametreler	Gizliliği Sağlanmış Ürün-Tabanlı OF	Önerilen Yaklaşım
σ = 1 ve β = 0	0,975	0,964
σ = 2 ve β = 0	0,972	0,962
σ = 4 ve β = 0	0,966	0,961
σ = 2 ve β = 50	0,976	0,965
σ = 2 ve β = 100	0,976	0,958
σ = 2 ve β = 200	0,976	0,950
σ = 2 ve β = 400	0,974	0,949



Şekil 5.8. *Gizliliği Sağlanmış Ürün-Tabanlı Ortak Filtrelemede $\sigma = 2$ ve Değişen β Değerleri için Hata*

5.3 Gizliliği Sağlanmış Kümeleme-Tabanlı Ortak Filtreleme Yönteminin Deney Sonuçları

Gizliliği sağlanmış kümeleme-tabanlı *OF* yönteminde veri setleri 3 adet kümeye ayrılmıştır. Test veri setindeki kullanıcılar için ait olduğu küme içerisinden 40 adet komşu çıkarılmıştır. Bu komşular kullanılarak test veri setindeki aktif kullanıcı için tahmin üretilmiştir. Burada deneyler MLP ve Netflix veri setleri üzerinden gerçekleştirilmiştir. Gizlilik parametreleri dikkate alınarak *OMH* metriğine göre sonuçlar karşılaştırıldığında, Şekil 5.9 ve Şekil 5.11'ye göre hata değerinin düştüğü görülmektedir. Ayrıca hata değerinin gizlilik parametrelerinden daha az etkilendiği görülmüştür. Bu da istenilen parametre değerine göre istenilen düzeyde gizlilik sağlandığında daha doğru tahmin üretilebildiğini göstermektedir. Yine aynı şekilde sonuçlar Şekil 5.10 ve Şekil 5.12'te *HKOK* metriğine göre karşılaştırıldığında benzer sonuçlar görülmektedir. Burada da değişen σ ve β değerlerine göre hatanın büyük oranda değişmediği ve boş veri setlerine göre daha iyi sonuç verdiği görülmektedir. Bu durum her iki veri seti içinde geçerlidir. İstenilen gizlilik düzeyinde oluşturulan veri setindeki hata değerlerinin seyrek veri setlerindeki hata değerlerine göre çok fazla değişmenin olmadığı görülmektedir. MLP veri seti üzerinde gizlilik olmadan elde edilen hata değerleri $OMH = 0,7701$ ve $HKOK = 1,0410$'tür. Önerilen yaklaşımda gizlilikten kaynaklanan artan hataların bazı parametrelerde ortadan kaldırdığı görülmektedir. Netflix veri setinde ise hata değerleri $OMH = 0,7444$ ve $HKOK = 1,0192$'dir. Bu sonuçlarda da önerilen yaklaşımın değerleri gizlilik etkisinden kaynaklanan hata farkını giderdiği görülmektedir.

Kapsama performansları karşılaştırıldığında Tablo 5.5 ve Tablo 5.6'ya göre perfor-

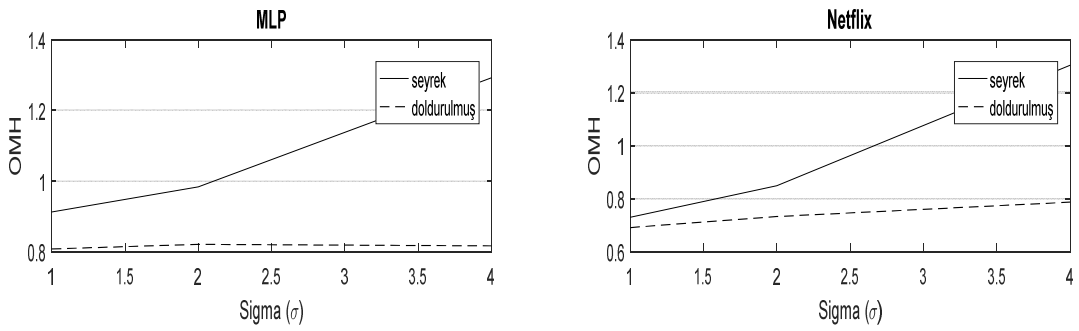
Tablo 5.5. *MLP veri seti üzerinde Gizliliği Sağlanmış Ürün-Tabanlı Ortak Filtreleme ile Gizlilik-Tabanlı Önerilen Ortak Filtrelemenin kapsama performansının karşılaştırılması.*

Parametreler	Gizliliği Sağlanmış Kümeleme-Tabanlı OF	Önerilen Yaklaşım
σ = 1 ve β = 0	0,984	0,971
σ = 2 ve β = 0	0,972	0,973
σ = 4 ve β = 0	0,976	0,972
σ = 2 ve β = 50	0,994	0,995
σ = 2 ve β = 100	0,995	0,994
σ = 2 ve β = 200	0,999	0,994
σ = 2 ve β = 400	1	0,994

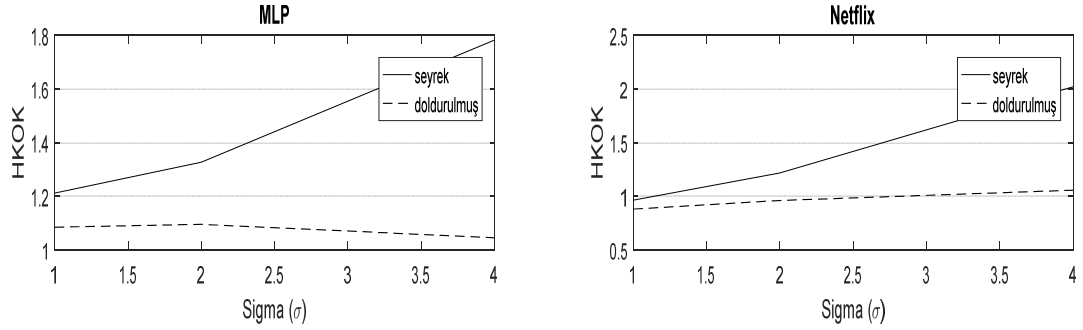
Tablo 5.6. *Netflix veri seti üzerinde Gizliliği Sağlanmış Kümeleme-Tabanlı Ortak Filtreleme ile Gizlilik-Tabanlı Önerilen Ortak Filtrelemenin kapsama performansının karşılaştırılması.*

Parametreler	Gizliliği Sağlanmış Kümeleme-Tabanlı OF	Önerilen Yaklaşım
σ = 1 ve β = 0	0,936	0,917
σ = 2 ve β = 0	0,925	0,905
σ = 4 ve β = 0	0,915	0,902
σ = 2 ve β = 50	0,948	0,933
σ = 2 ve β = 100	0,964	0,939
σ = 2 ve β = 200	0,983	0,969
σ = 2 ve β = 400	0,996	0,989

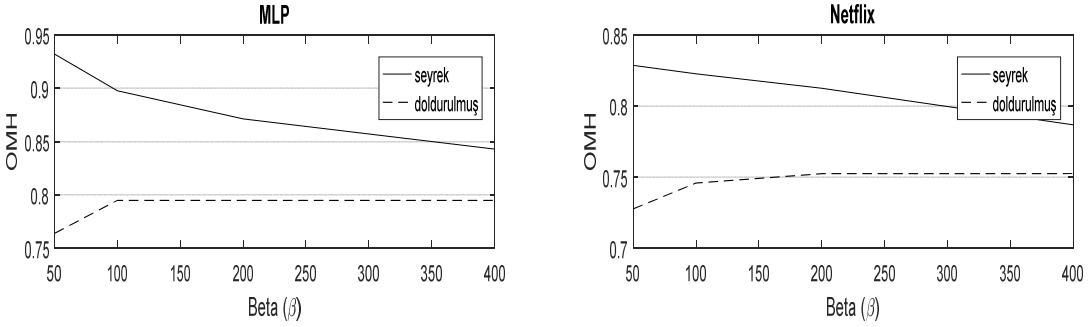
mansın beklenen düzeyde olmadığı görülmektedir. MLP veri seti için bazı parametrelerde iyileşmenin olduğu görülür. Ancak Netflix veri setinde, doldurulmamış veri setine göre daha fazla ürüne tahmin üretilememiştir.



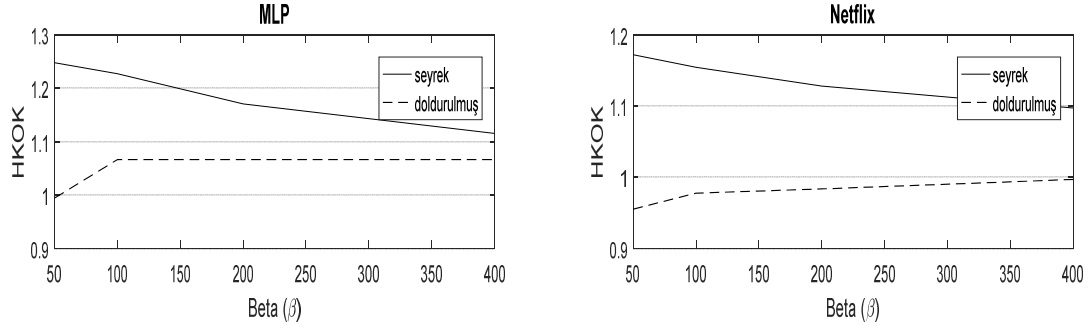
Şekil 5.9. *Gizliliği Sağlanmış Kümeleme-Tabanlı Ortak Filtrelemede β = 0 ve Değişen σ Değerleri için Hata*



Şekil 5.10. *Gizliliği Sağlanmış Kümeleme-Tabanlı Ortak Filtrelemede $\beta = 0$ ve Değişen σ Değerleri için Hata*



Şekil 5.11. *Gizliliği Sağlanmış Kümeleme-Tabanlı Ortak Filtrelemede $\sigma = 2$ ve Değişen β Değerleri için Hata*



Şekil 5.12. *Gizliliği Sağlanmış Kümeleme-Tabanlı Ortak Filtrelemede $\sigma = 2$ ve Değişen β Değerleri için Hata*

6. DEĞERLENDİRME

Önerilen yaklaşım hem doğruluk performanslarına hem de kapsama performanslarına göre karşılaştırılmıştır. Bütün bu değerlendirmeler açısından bakıldığında kullanıcı-tabanlı olarak önerilen yaklaşımda doğruluk ve performansın iyileştiği görülmüştür. Bu sonuçlar istenilen düzeyde gizlilik sağlandığında hem gizlilik olarak hem de doğruluk olarak tavsiye sisteminin iyileştirildiğini göstermektedir. Kullanıcılar kendi gizlilik seviyelerini belirleyerek istedikleri ürün hakkında daha iyi tahmin alabilmektedir. Hatta burada bazı gizlilik parametrelerinde önerilen yaklaşımın sonuçları gizleme olmadan üretilen sonuçların hata değerlerinden daha iyi çıkmıştır. Ancak ürün-tabanlı sistemlerde beklenen düzeyde iyileşme olmamıştır. Sadece yoğunluğu daha yüksek olan veri setindeki hata değerlerinde iyileşme olmuştur. Bu sistem ürünler arasındaki benzerliğe göre komşuluk hesapladığından doldurmanın olumlu etkise görülmemiştir. Kümeleme-tabanlı sistemlerde her iki yönlü hem doğruluk performansı hem de kapsama performansında bazı parametreler üzerinde olumlu etki ve iyileşme görülmüştür. Sonuçlar iki farklı metriğe göre değerlendirildiğinde iyileşmenin değişmediği görülmüştür. Eğer doldurma performansı daha da iyileştirilirse performanslar bellek-tabanlı, model-tabanlı ve hibrid yaklaşımlarda arttırılabilir.

Bu tezde kullanılan gizleme tekniği *RKT*'dir. Farklı gizleme teknikleri üzerinde bu doldurma yöntemi uygulanarak doğruluk performansının iyileştirmesi sağlanabilir. Çünkü tavsiye sistemlerinde seyreklik sorunu sık karşılaşılan bir sorundur. Bu soruna çözüm olarak önerilen yaklaşım başarılı olmuştur. Sisteme yeni dahil olmuş kullanıcılarda yeterli sayıda oy geçmişine sahip olamamaktadır. Bu yüzden bu kullanıcılar için sistem tavsiye üretmekte zorlanmaktadır. Bu tezde önerilen doldurma tekniği ile bu tür kullanıcıların ürünleri doldurulup diğer kullanıcılar ile komşuluğu hesaplanabilir. Bu durum yeni eklenen ürünler için de geçerlidir. Sisteme yeni dahil olmuş bir ürün yeteri kadar oy değerine sahip olmadığı için üst-N olarak oluşturulacak tavsiye listesinde kullanıcının beğenebileceği bir ürün olmasına rağmen listeye giremeyebilir. Bu tür sorunlarda ürünler arasında bu tezde kullanılan doldurma tekniği ile doldurma işlemi yapılırsa tavsiye listesine girebilme olasılığı artabilir. Aynı zamanda kullanıcılar için oluşturulacak tavsiye listesinin doğruluğu arttırılabilir.

KAYNAKÇA

[1] Schafer, J. B., Konstan, J., & Riedl, J. (1999). Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce,* pp. 158-166.

[2] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*. 4.

[3] Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM 35(12)*, pp. 61-70.

[4] Bilge, A., Kaleli, C., Yakut, I., Gunes, I., & Polat, H. (2013). A survey of privacy-preserving collaborative filtering schemes. *International Journal of Software Engineering and Knowledge Engineering, 23(08)*, pp. 1085-1108.

[5] Ozturk, A., & Polat, H. (2015). From existing trends to future trends in privacy-preserving collaborative filtering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5(6)*, pp. 276-291.

[6] Canny, J. (2002). Collaborative filtering with privacy. In *Security and Privacy, Proceedings. 2002 IEEE Symposium*, pp. 45-57.

[7] Pawlak, Z. (1982). Rough sets. *International Journal of Computer & Information Sciences, 11(5)*, pp. 341-356.

[8] Dubois, D., & Prade, H. (1990). Rough fuzzy sets and fuzzy rough sets.*International Journal of General System, 17(2-3)*, pp. 191-209.

[9] Pattaraintakorn, P., Zaverucha, G. M., & Cercone, N. (2007). Web based health recommender system using rough sets, survival analysis and rule-based expert systems. *Lecture Notes in Computer Science 4482*, pp. 491-499.

[10] Haijun, X., Qi, Z., & Baoyi, W. (2008). Rough set page recommendation algorithm based on information entropy. In *Computer Science and Software Engineering, 2008 International Conference on,* (Vol. 4), pp. 735-738.

[11] Huang, C. B., & Gong, S. J. (2008). Employing rough set theory to alleviate the sparsity issue in recommender system. In *2008 International Conference on Machine Learning and Cybernetics,* (Vol. 3), pp. 1610-1614.

[12] Fan, Y., Mai, J., & Ren, X. (2009). Rough Set-Based Clustering Collaborative Filtering Algorithm in E-commerce Recommendation System. In *2009 International Conference on Information Management, Innovation Management and Industrial Engineering,* (Vol. 4), pp. 401-404.

[13] Su, P., & Ye, H. (2009). An item based collaborative filtering recommendation algorithm using rough set prediction. In *Artificial Intelligence, 2009. JCAI'09. International Joint Conference*, pp. 308-311.

[14] Su, J. H., Wang, B. W., Hsiao, C. Y., & Tseng, V. S. (2010). Personalized rough-set-based recommendation by integrating multiple contents and collaborative information. In *Information Sciences 180(1),* pp. 113-131.

[15] Wang, B. W., & Tseng, V. S. (2012). Improving missing-value estimation in microarray data with collaborative filtering based on rough-set theory. *International Journal of Innovative Computing, Information and Control, 8,* pp. 2157-2172.

[16] Zhang, S., Li, C., Ma, L., & Li, Q. (2013). Alleviating the sparsity problem of collaborative filtering using rough set. *COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, 32(2),* pp. 516-530.

[17] Jensen, R., & Shen, Q. (2004). Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. *IEEE Transactions on Knowledge and Data Engineering, 16(12),* pp. 1457-1471.

[18] Zhang, Q., Qi, Y., Zhao, J., Hou, D., Zhao, T., & Liu, L. (2007). A study on context-aware privacy protection for personal information. In *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference,* pp. 203-224.

[19] Ye, M., Hu, X., & Wu, C. (2010). Privacy Preserving Attribute Reduction for Vertically Partitioned Data. In *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on* (Vol. 1), pp. 320-324.

[20] Ye, M., Hu, X., & Wu, C. (2010). Privacy preserving attribute reduction for horizontally partitioned data. In *Intelligent Systems and Knowledge Engineering (ISKE), 2010 International Conference,* pp. 315-319.

[21] Zhou, Z., Huang, L., & Yun, Y. (2009). Privacy preserving attribute reduction based on rough set. In *Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on*, pp. 202-206.

[22] Hu, D., Yu, X., & Feng, Y. (2008). Distributed Mining Core of Attributes on Horizontally Partitioned Data. In *Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on,* Vol. 2, pp. 11-15.

[23] Raju, N. L., Seetaramanath, M. N., Rao, P. S., & Nandini, G. (2013). Rough set based Privacy Preserving Attribute Reduction on Horizontally partitioned data and generation of Rules. *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 2, pp. 4343-4348.

[24] Zhu, W., Zhang, W., & Fu, Y., (2003) An Incomplete Data Analysis Approach Using Rough Sets Theory. *Pattern Recognition and Artificial Intelligence,* vol. 16, no. 2, pp. 158-163.

[25] Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999, August). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 230-237.

[26] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pp. 285-295.

[27] Sarwar, B. M., Karypis, G., Konstan, J., & Riedl, J. (2002, December). Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, vol. 1.

[28] Polat, H., & Du, W. (2003). Privacy-preserving collaborative filtering using randomized perturbation techniques. *Electrical Engineering and Computer Science*, p. 18

[29] Polat, H., & Du, W. (2005). Privacy-preserving collaborative filtering. *International journal of electronic commerce*, 9(4), 9-35.

[30] Batmaz Z. & Polat H., (2016). Randomization-based Privacy-preserving Frameworks for Collaborative Filtering, *Procedia Computer Science 96*, pages 33-42.