Journal of the Faculty of Engineering and Architecture of Gazi University 32:1 (2017) 53-64

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Bölünmüş veri-tabanlı gizliliği koruyan ortak filtreleme sistemlerinde gizli verinin elde edilmesi

Burcu Demirelli Okkalıoğlu[1], Mehmet Koç[2], Hüseyin Polat[3*]

[1]Yalova Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 77100, Yalova, Türkiye
[2]Bilecik Şeyh Edebali Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, 11210, Bilecik, Türkiye
[3]Anadolu Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 26470, Eskişehir, Türkiye

ÖNEÇIKANLAR

- YBV- ve DBV-tabanlı ve nümerik değerlemelere dayalı GTOF sistemlerinde veri imarı
- Aktif kullanıcı gibi davranarak gizli verinin elde edilmesi
- İlave bilgiler kullanılarak veri imarının iyileştirilmesi

Makale Bilgileri

Geliş: 13.10.2015
Kabul: 27.10.2016

DOI:
10.17341/gazimmfd.300594

Anahtar Kelimeler:
Gizlilik,
veri imarı,
bölünmüş veri,
ortak filtreleme,
atak

ÖZET

Ortak filtreleme algoritmalarının doğru ve güvenilir öneriler üretebilmesi için yeterli veriye ihtiyaç vardır. Bu nedenle yetersiz veriye sahip iki elektronik alışveriş sitesi gizliliklerini ihlal etmeden aralarındaki bölünmüş veriden öneriler sunmak isteyebilir. Bu amaçla gizliliği koruyan ortak filtreleme sistemleri geliştirilmiştir. Gizlilik-tabanlı ortak filtreleme sistemlerine karşı ataklar yapılarak gizli veri elde edilebilir. Bu çalışmada yatay ve dikey bölünmüş veri temelli gizliliği koruyan ortak filtreleme sistemlerine karşı atak senaryoları tasarlanıp ne kadar gizli veri elde edilebileceği gösterilmiştir. Ayrıca sistem hakkındaki ilave bilginin gizli veri elde etmeye katkısı çalışılmıştır. Gerçek verilerle yapılan deneyler bazı durumlarda gizli verinin önemli oranda elde edilebileceğini göstermiştir. Fakat ilave bilgi olmadan ve verinin yoğun olduğu durumlarda başarının çok düştüğü gözlenmiştir.

Deriving private data in partitioned data-based privacy-preserving collaborative filtering systems

HIGHLIGHTS

- Reconstruction of numeric data in HPD- and VPD-based PPCF systems
- Deriving private data by acting as an active user
- Improving data reconstruction by utilizing auxiliary information

Article Info

Received: 13.10.2015
Accepted: 27.10.2016

DOI:
10.17341/gazimmfd.300594

Keywords:
Privacy,
data reconstruction,
partitioned data,
collaborative filtering,
attack

ABSTRACT

Collaborative filtering algorithms need enough data to provide accurate and reliable predictions. Hence, two e-commerce sites holding insufficient data may want to provide predictions on their partitioned data with privacy. Different privacy-preserving collaborative filtering systems have been proposed for this purpose. Some attacks can be employed against such systems to derive confidential data. In this paper, attack scenarios are designed against horizontally and vertically partitioned data-based collaborative filtering with privacy schemes to show how much data can be derived. Also, how additional knowledge about the system helps data reconstruction is studied. Empirical outcomes on real data sets show that it is possible to derive high amount of private data in some cases. However, when there is no additional information and data is dense, data reconstruction success becomes very low.

* Sorumlu Yazar/Corresponding author: polath@anadolu.edu.tr / Tel: +90 537 873 1374

1. GİRİŞ (INTRODUCTION)

İnternet günlük yaşantımızın bir parçası haline gelmiştir. Günlük işlemlerin çoğu İnternet ortamına taşındığından, bu ortamlarda üretilen verinin hacmi her saniye artmaktadır ve dijital ortamlara aktarılan veri hızlı olarak büyümektedir [1]. Artan veri miktarı İnternet kullanıcılarının aşırı bilgi ile yüklenmesine sebep olur. Bilgi bombardımanı olarak adlandıran bu durum bir sistemin ele alabileceğinden çok daha fazla bilgiyle donanmış olması olarak ifade edilebilir. Bilgi bombardımanı karar süreçlerine doğrudan etki etmektedir. Şirketler karar verme süreçlerinde kendilerine yardımcı olabilecek ve öneriler sunabilecek değişik yöntemler uygulamaya başlamışlardır. Bu yöntemler arasında ortak filtreleme (OF) sistemleri de yer almaktadır. OF sistemleri kullanıcılara hızlı ve doğru öneriler sunabilen, en çok bilinen ve yaygın olarak kullanılan yöntemlerdir [2]. İnternet ile ortaya çıkan çoğu elektronik alışveriş sitesi OF sistemlerini kullanıcılarına öneri sunmak amacıyla kullanmaktadır. Klasik bir OF sisteminde $n \times m$ boyutunda bir kullanıcı-ürün matrisi vardır. Burada n ve m sırasıyla kullanıcı ve ürün sayısını temsil etmektedir. Bu matris genellikle çok boşluklu bir yapıdadır çünkü bir kullanıcıdan sistemin sahip olduğu ürünlerin çoğu hakkında bilgi sahibi olması beklenemez. OF sistemlerinin temel amacı kullanıcıların geçmiş tercihlerine göre harici bir bilgiye ihtiyaç duymadan onlara gelecekteki tercihleri hakkında öneriler sunmaktır [3]. OF sistemleri sıkça kullanılmasına rağmen gizlilik sağlayamayabilirler. Kişisel veriler değerlidir ve kişisel gizlilik açısından OF sistemleri bir tehdit oluşturmaktadır [4]. Kullanıcılar istenmeyen elektronik posta veya telefon alabilir ya da fiyat ayrımcılığına uğrayabilir [5]. Kullanıcı değerleme vektörleri incelenerek kullanıcı bilgileri çıkarılabilir. Bu sebeple hem kişisel gizliliği sağlayan hem de önerilerin doğruluğunu koruyabilen sistemlere ihtiyaç vardır. Gizlilik-tabanlı ortak filtreleme (GTOF) algoritmaları kullanıcıların gizliliğinden ödün vermeden öneri doğruluğunu en üst seviyede ortaya koymak için geliştirilmiş algoritmalardır. GTOF sistemlerinde genellikle kişisel gizliliği sağlamak için veri maskeleme yöntemleri [6] veya şifreleme teknikleri [7] tercih edilir. Temel olarak her kullanıcı kendi verisini gizleyerek öneri üretecek ana sunucuya gönderir ve ana sunucunun kişisel gizlilikten ödün vermeden doğru öneriler sunması beklenir. Ancak gizlilik ve doğruluk birbiri ile çakışan hedeflerdir [8]. GTOF sistemleri genel olarak merkezi sunucu tabanlıdır. Fakat bazı elektronik ticaret şirketleri pazara yeni girmiş ya da kendilerine yabancı yeni bir pazara girmek isteyebilirler. Bu gibi durumlarda bahsi geçen şirketlerin elinde yeterli veri olmayabilir. Bu zorluğu yenmek için iki şirket bir araya gelerek sahip oldukları veriyi paylaşabilirler. Böylece şirketlerin elinde bulunan veri zenginleşmiş ve öneri üretmeye daha yatkın hale gelmiş olur. Eldeki değerleme miktarı ne kadar çok olursa o veriden çıkarılabilecek ilişkiler o kadar sağlıklı olur ve böylelikle daha doğru öneriler üretilebilir. İki parti (şirket) arasında paylaşılan veri yatay ya da dikey olarak paylaşılmış olabilir. Yatay bölünmüş veri (YBV) durumunda iki parti arasında aynı ürün grubu için farklı kullanıcılar tarafından oylanmış veriler paylaşılır. Partilerin ürün sayısı aynı kalırken kullanıcı sayısı artacağından, OF algoritmalarının komşu seçimi ve öneri üretme aşamaları olumlu yönde etkilenecektir. Dikey bölünmüş veri (DBV) durumunda ise aynı kullanıcıların farklı ürünlere ait değerleme verileri paylaşılır. Bu durumda aynı kullanıcılar için ürün sayısı artacağından kullanıcılar arasında daha sağlıklı benzerlikler bulmak mümkün olacaktır. GTOF algoritmalarında doğru öneri üretmenin yanında kullanıcıların gerçek verilerini saklamak da önemlidir. Literatürde GTOF yöntemlerini hedef alarak gizli verilerden gerçek verileri elde edilmesini gösteren çalışmalar mevcuttur. Bu çalışmalardan ilham alarak çalışmanın özgün yönleri aşağıdaki gibi açıklanabilir: (1) Nümerik değerlemelere dayalı iki-partili GTOF sistemlerinde gizli verinin elde edilmesi için saldırı senaryoları tasarlanmıştır. Literatürde bilgimiz dâhilinde bu tür çalışma mevcut değildir. İkili değerlemelere dayalı iki-partili YBV-[9] ve DBV-tabanlı [10] GTOF sistemleri için üç farklı saldırı yöntemi tasarlanarak gizli veriler elde edilmeye çalışılmıştır. Fakat bu makalede saldırı yapılacak GTOF sistemleri nümerik veri temelli şemaları ve protokolleri içerdiğinden gerçek verilere ulaşmak için farklı ataklar tasarlanmıştır. (2) Bu çalışmada nümerik değerleme temelli iki-partili GTOF sistemlerinde gizli verinin ne kadar imar edileceği çalışılmıştır. Bu bağlamda benzer çalışma yoktur. Zhang vd. [11] tarafından yapılan çalışmada nümerik değerleme temelli ve merkezi sunucu-tabanlı GTOF sistemlerinde gizli verinin ne kadar imar edileceği çalışılmıştır (3) GTOF sistemleri hakkındaki ilave bilgilerin kullanılarak gizli verinin elde edilmesine katkıları irdelenmiştir. Okkalioglu vd. [12] atak yapılan merkezi sunucu tabanlı sistem hakkındaki ilave bilgilerden yararlanarak sahte ikili değerlemeleri tespit etmeye çalışmıştır. Başka bir çalışmada çeşitli yardımcı bilgiler kullanılarak gerçek nümerik verilerin yanında sahte oylanmış verilerden gerçekte hangilerinin oylanmış oldukları tespit edilmiştir [13]. Bu çalışmada ise YBV- ve DBV-tabanlı ve nümerik değerlemelere dayalı GTOF sistemlerinde veri imarı için ilave bilgilerin katkısı incelenmiştir. Bu çalışmanın bir sonraki bölümü ilgili çalışmaları listeler. Üçüncü bölümde ise bu çalışmanın hedef olarak belirlediği çalışmalar hakkında ön bilgiler verilmiştir. Bir sonraki bölümde ise saldırı senaryoları açıklanmıştır. Beşinci bölümde değerlendirme ölçütü, deneysel yöntem ve deneyler açıklanmıştır. Son bölümde sonuçlar sunulmuştur.

2. İLGİLİ ÇALIŞMALAR (RELATED WORKS)

Gizlilik-tabanlı veri madenciliği (GTVM) yöntemlerinin gizliliği inanıldığı kadar koruyamadığı bazı çalışmalarda gösterilmiştir. Okkalioglu vd. [14] gizli veriden gerçek verinin elde edilmesi üzerine yaptıkları çalışmada genel olarak rasgeleleştirme veya çarpımsal karıştırma ile saklanmış GTVM metotlarını ve bu metotlarla saklanmış gizli verilerden gerçek verileri elde etme yöntemlerini

açıklarlar. Ayrıca literatürde bulunan çalışmaların istatiksel analizini değerlendirme ölçütleri ile birlikte ortaya koyarlar. Rasgeleleştirme metodunda gizli veriye (x_i) normal ya da tekdüze dağılım ile seçilmiş herhangi bir rasgele sayı (r) eklenir. Gizlenen gerçek veri, veri seti içerisinde x_i yerine artık $x_i + r$ olarak temsil edilir. Bu şekilde gizlenmiş veriden gerçek veriyi elde etmek için spektral filtreleme, temel bileşen analizi (TBA) ve tekil değer ayrışımı (TDA) gibi bazı temel yöntemler kullanılır. Kargupta vd. [15] rasgele matris teoreminin özelliklerini kullanarak, rasgeleleştirme ile eklenmiş verinin öz değerleri çıkarıldığında geri kalan öz değerlerin gerçek değerlerin elde edilmesi için kullanılabileceğini göstermişlerdir. Huang vd. [16] TBA kullanarak saklanmış verilerden gerçek verileri elde etmeyi göstermişlerdir. TBA'da belli sayıda temel bileşeni seçerek veri elde edilmeye çalışılır. Geri kalan temel bileşenler rasgeleleştirme ile eklenen veriyi temsil eder. Bu nedenle veriler arasında ilinti ne kadar büyükse TBA-tabanlı metotlar da o kadar iyi sonuçlar verir. Veri imarı için kullanılan diğer metot TDA'dır [17]. TDA sonucunda elde edilen tekil değerler gerçek verinin gizlenmiş veriden elde edilmesinde büyük rol oynar. Tekil değerler büyükten küçüğe sıralanarak ilk tekil değerler gizli veriyi elde etmekte kullanılır. Geri kalanlar ise gerçek verilere eklenmiş rasgele veriler olarak kabul edilirler. GTVM çalışmalarının yanı sıra GTOF yöntemlerinden gerçek veriyi elde etmeye yönelik çalışmalar yapılmıştır. Merkezi sunucu-tabanlı GTOF sistemlerini ilk olarak Zhang vd. [11] 2006 yılında iki farklı yöntem kullanarak hedef almıştır. Polat ve Du [18] tarafından geliştirilen rasgeleleştirme-tabanlı GTOF algoritmasında saklanan gizli verinin elde edilmesi çalışılmıştır. TDA ve *k*-ortalama teknikleri kullanılarak gerçek veriye ulaşmak hedeflenmiştir. *k*-ortalama tekniği ile saklanmış veriyi ölçeklendirme derecesine (1-5) göre kümeleyip, her bir kümeye bir derece atayarak gerçek veriyi elde etmeye çalışırlar. 2011 yılında Calandrino vd. [19] yaptıkları çalışmada çevrimiçi OF sistemlerini hedeflemiştir. Yazarlar çevrimiçi sistemlerin zaman içerisinde ürettiği ve herkese açık olan çıktılar arasındaki farkları gözeterek gerçek veri ile ilgili çıkarımlarda bulunmuşlardır. Çalışmalarında *k-nn* (*k*-en yakın komşu) tekniğine dayalı bir atak tasarlanmıştır. Bu teknik bir kullanıcının oylama verisinin bir kısmı bilindiği durumda etkilidir ve komşuluk-tabanlı OF yöntemlerini hedef alır. Sisteme, geçmiş oylama bilgisi bilinen kullanıcıya benzer *k* tane sahte kullanıcı eklenir ve bunların hedef kullanıcının en yakın *k* komşusu olması beklenir. Gerçek verileri elde etmenin yanında oylanmış ürünlerin tespiti ile ilgili de çalışmalar mevcuttur. 2015 yılında Okkalioglu vd. [12] merkezi-tabanlı GTOF'de ikili verilerin gizlilik analizini yapar ve gizlenmiş verilerin hangisinin oylandığı bilgisini herkese açık ve herkes tarafından erişilebilir veriler kullanarak belirli oranlarda elde eder. 2016 yılında merkezi sunucu tabanlı ve nümerik değerlemelere dayalı GTOF sistemlerini hedef alan başka bir çalışmada ise kullanıcıların farklı gizlilik gereksinimlerine ihtiyaç duymaları halinde gizlenmiş veriden ne kadar veri elde edilip edilemeyeceği gösterilmiştir [13]. Her kullanıcının farklı gizlilik endişesine göre sakladığı veriden gerçekte hangi ürünlerin oylanmış olduğunu elde etmeye çalışmışlardır. Çalışmada mevcut tekniklerin yanında yardımcı bilgiler kullanılarak veri imarının başarısı arttırılmıştır. Yukarıda bahsedilen GTOF sistemlerinden gerçek veriyi elde etmeyi amaçlayan çalışmalar merkezi sunucu temellidir. Okkalioglu vd. [9] ikili veriye dayalı iki-partili YBV temelli GTOF yöntemini hedef alırlar. Araştırmacılar bu çalışmada Polat ve Du [20], [21] tarafından önerilen iki-partili YBV için sunduğu GTOF yöntemini ele alır ve bu yönteme karşı üç farklı atak gerçekleştirirler. Mükemmel uyum atağı yazarlar tarafından sunulmuş olup gönderilen bir sorguyla benzerliği 1 ya da -1 olan kullanıcılardan (mükemmel uyum) faydalanarak verileri elde etmeye çalışır. Bir diğer çalışmada ise ikili veriye dayalı iki-partili DBV temelli GTOF [20], [21] yöntemine karşı bir önceki çalışmada yaptıkları saldırıları uygulayarak gizli veri elde etmeye çalışırlar [10]. İki parti arasında nümerik değerlemeler içeren bölünmüş veriye dayalı GTOF sistemlerinde partiler birbirlerinin gizli nümerik değerlemelerini ve oylanmış ürünlerini öneri üretme sırasında elde etmeye çalışabilir. Ancak bu gizli verilerin nasıl elde edileceğine dair literatürde herhangi bir çalışma bulunmamaktadır. Mevcut çalışmalarda ikili değerlemeler içeren YBV- [9] ve DBV-tabanlı [10] kullanıcı-ürün matrislerinden değişik saldırılar sonucunda ne kadar gizli verinin elde edilebileceği gösterilmiştir. Bizim çalışmamız onların gerçekleştirmiş olduğu çalışmalardan yukarıda bahsedilen sebeplerden dolayı farklıdır. Bizim çalışmamızda veri seti nümerik değerlemeler içerdiğinde ve veriler iki parti arasında yatay veya dikey olarak bölündüğünde yardımcı bilgiler kullanılarak ne kadar gizli verinin elde edileceği çalışılmıştır.

3. GİZLİLİĞİ KORUYARAK BÖLÜNMÜŞ VERİ TEMELLİ ÖNERİ ÜRETME (PROVIDING PRIVATE PREDICTIONS ON PARTITIONED DATA)

Bu bölümde, veriler iki parti arasında yatay [22] veya dikey [23] olarak bölündüğünde, gizliliğin korunarak önerilerin nasıl üretildiğini gösteren metotlar ve kullanılan gösterimler ön bilgi olarak açıklanacaktır.

3.1. Gizliliği Koruyarak YBV-Tabanlı Öneri Üretme (Providing Private Predictions On HPD)

OF sistemlerinde sınırlı sayıda kullanıcı olduğunda güvenilir ve doğru öneriler sunmak zordur. Ayrıca bu sistemler sınırlı kullanıcı ve değerlemeler nedeniyle bazı ürünler için öneri bile sunamayacak durumlarla karşılaşabilirler. Verilerin birleştirilmesi iki parti için yararlı olmasına rağmen, şirketler gizlilik, kanuni ve mali endişelerden dolayı verilerini birleştirmeye gönüllü olmayabilirler. Polat [22] bu endişeleri giderecek bir yöntem önermiştir. Önerilen yöntem çevrim-dışı ve çevrimiçi hesaplamaları kapsamaktadır. Bu yönteme göre önerilerin hesaplanması için öncelikle P değeri Eş. 1 ile hesaplanır:

$$P = \frac{\overbrace{\sum_k z_{ak}\left[\sum_{i=1}^{n_A} z_{ik} z_{iq}\right]}^{A_N} + \overbrace{\sum_k z_{ak}\left[\sum_{i=1}^{n_B} z_{ik} z_{iq}\right]}^{B_N}}{\underbrace{\sum_k z_{ak}\left[\sum_{i=1}^{n_A} z_{ik}\right]}_{A_D} + \underbrace{\sum_k z_{ak}\left[\sum_{i=1}^{n_B} z_{ik}\right]}_{B_D}} = \frac{A_N + B_N}{A_D + B_D} \quad (1)$$

Eş. 1'de n_A değeri *A* partisinin kullanıcı sayısını ifade ederken n_B ise *B* partisinin kullanıcı sayısını gösterir ve toplamda $n = n_A + n_B$ olacak şekilde veriler birleştirilir. *k* ise *a* ile *i* kullanıcısının birlikte oyladıkları ürünleri ifade eder. Eş. 1'den görüldüğü gibi önerileri üretmek için iki partinin (*A* ve *B* partisi) verisi gerekmektedir.

Çevrim-dışı Hesaplama: Her partinin pay ve payda olmak üzere iki hesaplama yapması gerekmektedir. Veriler yatay olarak dağıtıldığından bu hesaplamalar için partiler birbirinin verisine ihtiyaç duymazlar. Bu nedenle pay ve paydadaki değerler partilerin sahip oldukları veriler kullanılarak bağımsız olarak hesaplanır. Polat [22] hesaplamaların çevrim-dışı yapılmasını önermiştir. İki partinin yapmış olduğu işlemler aynı olduğundan, *A* partisinin gerçekleştirdiği adımlar aşağıdaki gibi açıklanabilir [22]:

- *A* partisi elindeki değerlemeleri *z-skor* değerlerine dönüştürür.
- *A* partisi j = 1, 2, …, m için $\sum_{i=1}^{n_A} z_{ij}$ değerlerini hesaplar. $\sum$AD adlı 1× m boyutunda bir matriste, $\sum AD = [\sum AD_1, \sum AD_2, \ldots, \sum AD_m]$ şeklinde payda için gerekli tüm verileri saklar.
- *A* partisi pay kısmındaki verileri hesaplar. Bu durumda her bir q ürünü için, q = 1, 2,...,m, *A* partisi $\sum_{j=1}^{m}\sum_{i=1}^{n_A} z_{ij}z_{iq}$ hesaplar ve $\sum$AN adlı *1*× m boyutunda bir matriste, $\sum AN = [\sum AN_1, \sum AN_2, \ldots, \sum AN_m]$ şeklinde saklar.

B partisi de *A* partisinin gerçekleştirmiş olduğu adımları yaparak $\sum$BD ve $\sum$BN matrislerinde gerekli veriyi saklar.

Çevrimiçi Hesaplama: Bir parti yönetici olarak seçilir. *A* partisinin yönetici parti olduğunu kabul edelim. Çevrimiçi hesaplama şu şekilde özetlenebilir [22]:

- Aktif kullanıcı (*a*) kendi verisini ve bir sorguyu (öneri isteyeceği ürünü genellikle q olarak adlandırılır) her iki partiye gönderir.
- *B* partisi gizli skaler çarpım hesaplama (GSÇH) protokolünü kullanarak B'_N ve B'_D değerlerini hesaplayarak *A* partisine gönderir.
- *A* partisi *B* partisinden değerleri aldıktan sonra, kendi A_N ve A_D değerlerini hesaplar ve daha sonra P' değerini elde eder.
- Son olarak, *A* partisi aktif kullanıcının değerlemelerinin standart sapması ile P' değerini çarpar ve bu çarpımın sonucuna değerlemelerin ortalamasını ekleyerek P'_{ag} değerini bulur. Bu değeri daha önce tanımlanmış eşik değeri ile karşılaştırarak aktif kullanıcının hedef üründen hoşlanıp hoşlanmayacağını aktif kullanıcıya bildirir.

Gizli Skaler Çarpım Hesaplama Protokolü: Yönetici olmayan *B* partisi, B_N ve B_D değerlerini *A* partisinden saklamak için aşağıda önerilen GSÇH protokolünü kullanır [22]:

- Aktif kullanıcının oyladığı toplam ürün sayısını (C_B) hesaplar.
- Eğer $C_B < \lfloor m_B/2 \rfloor$ ise (m_B değeri *B* partisinin toplam ürün sayısı), *B* partisi aktif kullanıcının oylamadığı toplam ürün sayısını ($m_B - C_B$) hesaplar. *B* partisi (1, $m_B - C_B$) aralığından bir rasgele sayı (S_{Ba}) belirler ve rasgele olarak aktif kullanıcıdan S_{Ba} kadar oylanmamış ürünleri ilgili ürünlerin ortalamaları ile doldurur.
- Eğer $C_B > \lfloor m_B/2 \rfloor$ ise, *B* partisi (1, C_B) aralığından bir rasgele sayı (S_{Br}) seçer. Daha sonra rasgele olarak aktif kullanıcının oylamış olduğu ürünlerden S_{Br} kadar ürünü seçerek bu ürünlerin değerlerini kaldırır.

A partisi S_{Ba} ve S_{Br} değerlerini ve hangi ürünlerin eklendiğini/çıkarıldığını bilmediğinden, aktif kullanıcı gibi davranıp aynı ürün için birden çok istekte bulunsa bile B'_N ve B'_D değerlerinden B_D ve B_N değerlerini elde edemez.

3.2. Gizliliği Koruyarak DBV-Tabanlı Öneri Üretme
(Providing Private Predictions On VPD)

Elektronik alışveriş sitelerinin sayısı gün geçtikçe artmakta ve müşteriler alışveriş için farklı siteleri tercih etmektedirler. Bu durum OF amacıyla toplanan dikey bölünmüş veriye neden olmaktadır. İki parti gizliliklerini ifşa etmeden dikey bölünmüş verilerini birleştirdiğinde müşterilerine daha iyi öneriler sunabilir. Polat ve Du [23] *P* değerini Eş. 2 ile hesaplamayı önermiştir:

$$P = \frac{\overbrace{\sum_{k_A} z_{ak_A}\left[\sum_{i=1}^{n} z_{ik_A} z_{iq}\right]}^{A_N} + \overbrace{\sum_{k_B} z_{ak_B}\left[\sum_{i=1}^{n} z_{ik_B} z_{iq}\right]}^{B_N}}{\underbrace{\sum_{k_A} z_{ak_A}\left[\sum_{i=1}^{n} z_{ik_A}\right]}_{A_D} + \underbrace{\sum_{k_B} z_{ak_B}\left[\sum_{i=1}^{n} z_{ik_B}\right]}_{B_D}} = \frac{A_N + B_N}{A_D + B_D} \quad (2)$$

Eş. 2'de k_A değeri *A* partisinin sahip olduğu ürünler içinden *a* ile kullanıcı *i*'nin oylamış oldukları ürünleri tanımlarken, k_B değeri ise *B* partisinde bulunan ürünler içinden *a* ile kullanıcı *i*'nin oylamış olduğu ürünleri gösterir. $k = k_A + k_B$ ve *k* iki partinin ürünlerinin toplam sayısını ifade eder. DBV durumu için önerilen protokol çevrim-dışı ve çevrimiçi hesaplamadan oluşur [23].

Çevrim-dışı Hesaplama: Eş. 2'de görüldüğü gibi A_N ve B_N değerleri hesaplanırken q ürününe sahip olmayan parti $\sum_{i=1}^{n} z_{ik_j} z_{iq}$ değerini hesaplayabilmesi için i = 1, 2, …, n için z_{iq} değerlerine ihtiyacı vardır. Polat ve Du [23] partilerin A_N ve B_N değerlerini çevrim-dışı olarak hesaplayabilmeleri için bir şema sunmuştur. *A* partisi ilk olarak $n \times m_A$ matrisini yatay olarak c_A alt matrise böler. Burada m_A değeri *A*

partisinin ürün sayısını ifade eder. Daha sonra her bir alt matristeki verileri birbirinden bağımsız olarak değiştirir. Her bir alt matris için şu işlemler gerçekleştirilir:

- $\prod Ai$ permütasyon fonksiyonu kullanarak A tüm m_A sütun vektörlerinin yerlerini değiştirir.
- j = 1, 2,…, m_A için yerleri değiştirilmiş sütun vektörü $\prod Ai(I_{ij})$ toplamları yine bu vektör olacak şekilde rasgele vektörlere $\prod Ai(I_{ij}) = \sum_{z=1}^{d_{ij}} X_{ijz}$, d_{ij} rasgele vektörlerin sayısı, bölünür.
- Bir önceki adımda $X_{i11},\ldots,X_{i1di1},X_{i21},\ldots,X_{i2di2},\ldots X_{imA1},\ldots X_{imAdimA}$ rasgele vektörleri tekrar sadece A tarafından bilinen bir permütasyon fonksiyonu ile yerleri değiştirilerek B partisine gönderilir.
- $D_{Ai} = d_{i1} + d_{i2} + \ldots\ldots + d_{imA}$ değiştirilmiş rasgele vektörler B partisine gönderilir. B bu değiştirilmiş rasgele vektörler ile kendi m_B sütun vektörlerinin ilgili parçaları arasındaki sayısal çarpımı hesaplar.
- Sayısal çarpım sonuçları bulunduktan sonra B partisi homomorfik şifreleme kullanarak kendi genel anahtarı ile sonuçları şifreler ve A partisine gönderir.
- A partisi homomorfik şifreleme özelliklerini kullanarak kendi m_A ve B'nin m_B sütun vektörlerinden şifrelenmiş sayısal sonuçlarını hesaplar. İlk başta yatay olarak matrisini böldüğü için elde ettiği sonuçlardan yine homomorfik şifreleme özelliğini kullanarak şifrelenmiş olarak son sayısal çarpım sonuçlarını bulur. ∑A matrisini oluşturarak sonuçları saklar.
- A partisi kendi verisini B partisinin öğrenmesini istemediğinden tüm şifrelenmiş son çarpım sonuçları kadar rasgele sayılar üretir. B partisinin genel anahtarını bildiğinden bu rasgele sayıları şifreler. Şifrelemiş olduğu rasgele sayıları son bulmuş olduğu şifrelenmiş sayısal sonuçlara homomorfik şifreleme özelliğini kullanarak ekler ve değiştirilmiş son sayısal çarpım sonuçlarını ($\sum'$A) B partisine gönderir. Diğer taraftan A oluşturmuş olduğu tüm rasgele sayıları bir matriste saklar. B partisi $\sum'$A matrisini aldıktan sonra kendi gizli anahtarı ile matrisin şifresini çözerek değerleri $\sum''$A matrisinde saklar.

Çevrimiçi Hesaplama: Her iki parti aktif kullanıcı gibi davranıp diğer partinin verilerini elde etmeye çalışacağından çevrimiçi hesaplama aşağıdaki gibi yapılır [23]:

- a kendi verisini ve bir sorguyu (öneri istediği ürün q) q ürününe sahip olan partiye gönderir. A partisinin q ürününe sahip olduğunu farz edelim. A partisi $B'_N + A_N$ ve A_D hesaplamalarını yapabilir. Fakat B partisi aktif kullanıcı gibi davranıp A partisinin A_D ve A_N verilerinden gerçek verileri elde etmeye çalışabilir. Bu nedenle GSÇH protokolü kullanılır.
- A partisi $\sum''$B matrisinin q. satırındaki veri ile aktif kullanıcının verisini kullanarak B'_N değerini hesaplayabilir. B'_N değeri şu şekilde gösterilebilir: $B'_N = B_N + R_q$. A partisinin B_N değerini elde etmemesi için B partisi $\sum''$B matrisi oluşturulurken rasgele sayılar (R_q) ekler. A partisi $B_N + R_q + A'_N$ ve A'_D değerleri ile birlikte aktif kullanıcının yeni ortalamasını, standart sapmasını ve B partisinin sahip olduğu ürünleri oylayan aktif kullanıcının *z-skor* değerlerini hesaplayarak aktif kullanıcıya gönderir.
- Aktif kullanıcı A partisinden aldığı veriyi B partisine gönderir. B partisi ilk önce eklemiş olduğu rasgele R_q değerini $B_N + R_q + A'_N$ toplamından çıkarır ve A_D değerini hesaplar. Sonuç olarak, Eş. 2'de gösterilen dört değer hesaplanır ve A partisi P' değerini hesaplayarak aktif kullanıcıya q ürününü sevip sevmeyeceğini iletir.

4. SALDIRI SENARYOSU (ATTACK SCENARIO)

4.1. Yatay Bölünmüş Veride Saldırı Senaryoları
(Attack Scenarios on Horizontally Partitioned Data)

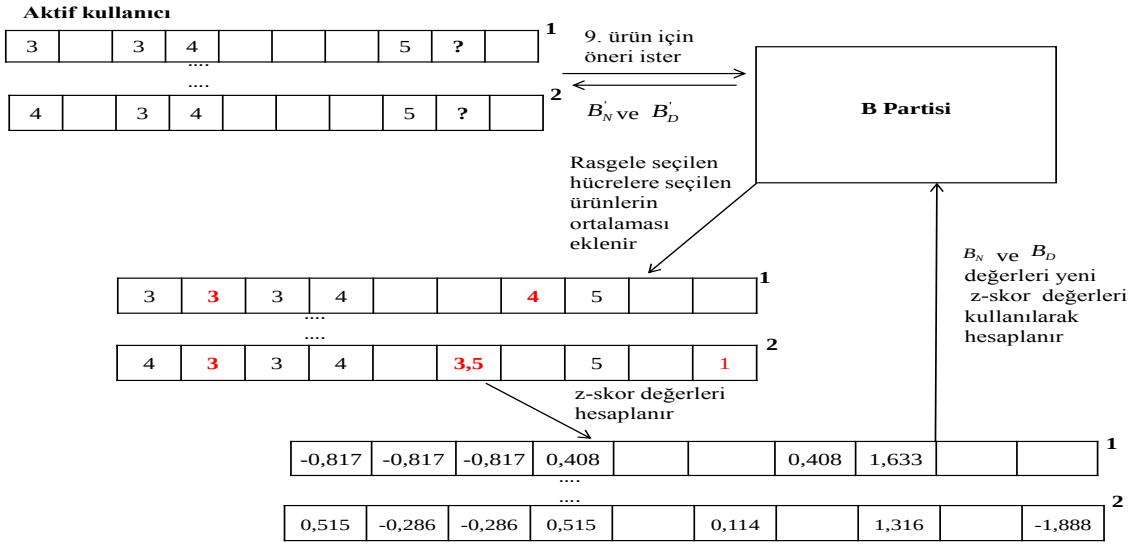
YBV durumunda, partiler birbirlerinin gizli verisini elde etmek için aktif kullanıcı gibi davranabilirler. Aktif kullanıcı gibi davranma saldırı senaryosu şöyle gerçekleşir: Saldırı yapacak parti aktif kullanıcı gibi kendini göstererek bir değerleme vektörü oluşturur. Sonraki sorgulamalarda bu vektörden tek bir değerleme değiştirmesi sonucu elde etmiş olduğu ara toplamlardan bir çıkarım elde ederek karşı partinin verilerini anlamaya çalışır. Her seferinde tek bir ürünün değerini değiştirerek karşı taraftan ara toplam sonuçlarını elde eder ve bu işlemi tüm ürünler hakkında bilgi elde edene kadar tekrarlar. Böylelikle ilk gelen sonuç ile aktif kullanıcının oluşturmuş olduğu vektördeki bir ürünün değerini değiştirerek elde ettiği sonuç arasındaki fark o ürünün değerinin ortaya çıkmasına neden olur. Polat [22] bu tarz saldırılara karşı önermiş olduğu şemanın güvenli olması açısından GSÇH protokolünü uygulamıştır. Bu protokole göre aktif kullanıcının göndermiş olduğu değerleme vektörü alındıktan sonra uygulanan protokol sonucunda ya rasgele seçilmiş bazı oylanmamış ürünlere ilgili ürünlerin değerleme ortalaması eklenir ya da rasgele seçilmiş bazı oylanmış ürünlerin değerlemeleri ilgili değerleme vektöründen çıkarılır. Bu protokolün daha iyi anlaşılması için protokolü bir örnekle açıklayalım. Her iki parti kullanıcılarından kendi ürünleri hakkındaki değerlemeleri toplar ve kullanıcı-ürün matrislerini oluşturur. Daha sonra bu değerlemeler *z-skor* değerlerine dönüştürülür. Şekil 1'de B partisinin 5 kullanıcısı ve 10 ürünü olduğunu varsayalım. B partisi 5 kullanıcısından değerlemeleri aldıktan sonra tüm değerlemeleri *z-skor* değerlerine dönüştürür. Polat [22] tarafından önerilen protokol ise Şekil 2'de bir örnek üzerinde gösterilmiştir. A partisinin B partisinin verisini elde etmek istediği varsayımında bulunulmuştur. Bu örnekte aktif kullanıcının iki kez aynı ürün için öneri istediği gösterilmiştir. İlk öneri isteğinde aktif kullanıcı değerleme vektörünü ve öneri istediği ürünü (örnekte dokuzuncu ürün) B partisine gönderir. B partisi rasgele olarak oylanmamış ürünleri seçer. Bu örneğimizde ikinci ve yedinci hücrelerin seçildiğini varsayalım. Daha sonra aktif kullanıcı değerleme vektörünün ikinci ve yedinci hücreleri bu ürünlerin ortalamaları ile doldurulur.

	Ürün 1	Ürün 2	Ürün 3	Ürün 4	Ürün 5	Ürün 6	Ürün 7	Ürün 8	Ürün 9	Ürün 10
Kullanıcı 1		4	3	3		3		5		
Kullanıcı 2	3		2		3	4			2	1
Kullanıcı 3		4	5			3		5		
Kullanıcı 4		2	3		4	4		4		
Kullanıcı 5	2	2	3				4		4	

Gerçek değerlerin z-skor değerlerine dönüştürülmesi

	Ürün 1	Ürün 2	Ürün 3	Ürün 4	Ürün 5	Ürün 6	Ürün 7	Ürün 8	Ürün 9	Ürün 10
Kullanıcı 1		0,447	-0,671	-0,671		0,671		1,565		
Kullanıcı 2	0,477		-0,477		0,477	1,430			-0,477	-1,430
Kullanıcı 3		-0,261	0,783			-1,306		0,783		
Kullanıcı 4		-1,565	-0,447		0,671	0,671		0,671		
Kullanıcı 5	-1,000	-1,000	0,000				1,000		1,000	

Şekil 1. Değerlemeler ve z-skor değerleri (Ratings and their z-scores)



Şekil 2. Protokolün örnek üzerinde gösterilmesi (An example case for the protocol)

	Ürün 1	Ürün 2	Ürün 3	Ürün 4	Ürün 5	Ürün 6	Ürün 7	Ürün 8	Ürün 9	Ürün 10
Kullanıcı 1		4	3	3		3		5		
Kullanıcı 2	3		2		3	4			2	1
Kullanıcı 3		4	5			3		5		
Kullanıcı 4		2	3		4	4		4		
Kullanıcı 5	2	2	3				4		4	

1 kez oylanmış ürünlerin yerleri

	Ürün 1	Ürün 2	Ürün 3	Ürün 4	Ürün 5	Ürün 6	Ürün 7	Ürün 8	Ürün 9	Ürün 10
Kullanıcı 1				X						
Kullanıcı 2										X
Kullanıcı 3										
Kullanıcı 4										
Kullanıcı 5							X			

Şekil 3. Yardımcı bilgi: 1 kez oylanmış ürünlerin yerleri (Auxiliary information: locations of the one time-rated items)

Şekil 1'de gösterildiği gibi ikinci ürün ortalaması (4+4+2+2)/4 = 3 ve yedinci ürün ortalaması 4/1 = 4 olarak hesaplanır. *B* partisi daha sonra aktif kullanıcının gerçek değerlerini *z-skor* değerlerine dönüştürerek B'_N ve B'_D değerlerini hesaplar ve aktif kullanıcıya geri gönderir. Aktif kullanıcı ikinci öneri isteğinde sadece ilk değerlemenin değerini değiştirerek aynı ürün için tekrar öneri istemektedir. Fakat *B* partisi her seferinde farklı boş hücrelerin yerini seçerek (ikinci, altıncı ve onuncu ürünler) bunları ürünlerin ortalaması ile doldurduğundan aktif kullanıcı birçok kez aynı ürün için istekte bulunsa bile bu ara toplamlardan (B'_N ve B'_D) bir sonuç elde edemez. Örnek üzerinden devam ederek matematiksel olarak verinin elde edilip edilemeyeceği incelenmiştir. *A* partisi aktif kullanıcı gibi davrandığından *B* partisinden elde ettiği B_N ve B_D değerlerinden gerçek verilere ulaşmak istemektedir. Bu durumu kısaca B_D değerleri göz önünde bulundurarak açıklayalım. Eş. 3'te B_D değerleri şu şekilde hesaplanır:

$$B_D = z_{a1}\sum_{i=1}^{nB} z_{i1} + z_{a2}\sum_{i=1}^{nB} z_{i2} + \ldots + z_{am}\sum_{i=1}^{nB} z_{im} \tag{3}$$

Örnek üzerinden hesaplamalar yaparsak aktif kullanıcı normalde protokol uygulanmasa Eş. 4'teki sonucu elde eder.

$$B_{D_1} = z_{a1}\sum_{i=1}^{nB} z_{i1} + z_{a3}\sum_{i=1}^{nB} z_{i3} + z_{a4}\sum_{i=1}^{nB} z_{i4} + z_{a8}\sum_{i=1}^{nB} z_{i8} \tag{4}$$

Fakat protokol uygulandığından Şekil 2'de de görüldüğü gibi *B* partisi rasgele hücrelere veri eklemiştir. Bu nedenle aktif kullanıcının elde etmiş olduğu değer Eş. 5'te gösterilmiştir.

$$B'_{D_1} = z_{a1}\sum_{i=1}^{nB} z_{i1} + z_{a2}\sum_{i=1}^{nB} z_{i2} + z_{a3}\sum_{i=1}^{nB} z_{i3} + z_{a4}\sum_{i=1}^{nB} z_{i4} + z_{a7}\sum_{i=1}^{nB} z_{i7} + z_{a8}\sum_{i=1}^{nB} z_{i8} \tag{5}$$

Aktif kullanıcı aynı değerleme vektörünü kullanarak ardı ardına her seferinde tek bir ürününün değerini değiştirse bile Polat [22] tarafından uygulanan protokol sayesinde *B* partisi her seferinde farklı boş hücreleri dolduracağı için toplam sonuçlardan bir çıkarım elde edemez. Örneğin Şekil 2'de gösterilen aktif kullanıcının değerleme vektörü gönderildiğinde elde ettiği sonuç Eş. 6 ile hesaplanır.

$$B'_{D_2} = z_{a1}\sum_{i=1}^{nB} z_{i1} + z_{a2}\sum_{i=1}^{nB} z_{i2} + z_{a3}\sum_{i=1}^{nB} z_{i3} + z_{a4}\sum_{i=1}^{nB} z_{i4} + z_{a6}\sum_{i=1}^{nB} z_{i6} + z_{a8}\sum_{i=1}^{nB} z_{i8} + z_{a10}\sum_{i=1}^{nB} z_{i10} \tag{6}$$

B partisi tarafından aktif kullanıcının oylanmamış ürünleri rasgele seçilerek bu ürünlerin yerlerinin doldurulması ara sonuçlardan bilgi elde etmeyi engeller. Çünkü aktif kullanıcı aynı değerleme vektörünü bile gönderse her seferinde *B* partisi rasgele boş hücreleri seçerek dolduracağından bir çıkarım yapılamaz. Polat [22] tarafından önerilen protokol gizliliği arttırırken rasgele ürünleri doldurmak doğruluğu düşürebilir. Gizlilik ve doğruluk birbiri ile çakışan hedefler olduğundan, gizlilik artarken doğruluğun hangi oranda düştüğü önemlidir. Diğer bir önemli nokta ise boşlukları doldurunca aktif kullanıcının değerleme değerlerinin orijinalliği bozulur ve sonuç olarak doğruluğu etkiler. Doğruluk oranının önemli olduğu varsayılırsa, aktif kullanıcının göndermiş olduğu değerleme vektörlerine rasgeleleştirme eklenmediği kabul edilir. Bu durumda aktif kullanıcının ara toplamlardan veri elde edip edemeyeceğini inceleyelim. Partiler gerçek değerler yerine *z-skor* değerlerini kullandığından aktif kullanıcının da kendi gerçek değerleri yerine *z-skor* değerlerini gönderdiğini varsayalım. Aktif kullanıcı *z-skor* değerlerine dönüştürülmüş değerleme vektörünü *B* partisine gönderdiğinde, *B* partisi sadece aktif kullanıcının oylamış olduğu ürünleri kullanarak Eş. 4'teki sonucu aktif kullanıcıya gönderir. Aktif kullanıcı kendi gerçek değerlerini *z-skor* değerlerine dönüştürdüğü için sadece ilgili ürünün *z-skor* değerini arttırıp/azaltarak diğer ürünlerin *z-skor* değerlerini değiştirmeden *B* partisinden bir sonuç elde etmeye çalışır. Aktif kullanıcı sadece ilk ürünün *z-skor* değerini değiştirip bir öneri istediğinde Eş. 7'deki sonucu elde eder.

$$B_{D_2} = z_{a1'}\sum_{i=1}^{nB} z_{i1} + z_{a3}\sum_{i=1}^{nB} z_{i3} + z_{a4}\sum_{i=1}^{nB} z_{i4} + z_{a8}\sum_{i=1}^{nB} z_{i8} \tag{7}$$

Eş. 7'nin içeriğine baktığımızda aslında aktif kullanıcının değiştirmemiş olduğu diğer ürünlerin çarpımı Eş. 5 ile aynı olduğu görülmektedir. Eğer iki eşitliği birbirinden çıkarırsak aradaki fark bize birinci ürün hakkında bir sonuç elde etmemize yardım edebilir. Eş. 4 ve Eş. 7 sonuçlarını kullanarak şu sonuç elde edilir.

$$\begin{aligned} B_{D_2} - B_{D_1} &= z_{a1'}\sum_{i=1}^{nB} z_{i1} + z_{a3}\sum_{i=1}^{nB} z_{i3} + z_{a4}\sum_{i=1}^{nB} z_{i4} + z_{a8}\sum_{i=1}^{nB} z_{i8} - z_{a1}\sum_{i=1}^{nB} z_{i1} \\ &\quad - z_{a3}\sum_{i=1}^{nB} z_{i3} - z_{a4}\sum_{i=1}^{nB} z_{i4} - z_{a8}\sum_{i=1}^{nB} z_{i8} \\ B_{D_2} - B_{D_1} &= z_{a1'}\sum_{i=1}^{nB} z_{i1} - z_{a1}\sum_{i=1}^{nB} z_{i1} \\ B_{D_2} - B_{D_1} &= \sum_{i=1}^{nB} z_{i1}\left(z_{a1'} - z_{a1}\right) \end{aligned} \tag{8}$$

Eş. 8'in sonucuna baktığımızda aktif kullanıcı B_{D1} ve B_{D2} değerlerine sahiptir. Ayrıca kendi *z-skor* değerlerini de bildiğinden buradan $\sum_{i=1}^{n_B} z_{i1}$ değerlerini kolaylıkla elde edebilir. Başka bir ifadeyle, aktif kullanıcı birinci ürünü oylayan kullanıcıların toplam değerlerini elde edebilir. Burada kaç kullanıcının bu ürünü oyladığını veya hangi kullanıcıların bu ürünü oyladığı bilgisine erişemez.

$$\sum_{i=1}^{k} z_{i1} = z_{11} + z_{21} + \ldots + z_{k1} \quad (9)$$

Eş. 9'da aktif kullanıcı birinci ürünü oylayanların sayısını bilmediğinden *k* sayısını tahmin edemez. *k* sayısını tahmin etse bile hangi kullanıcılar olduğunu hiçbir şekilde bilemez. Yalnızca elde ettiği sonuçlardan o ürünü kullanıcıların oylayıp oylamadığı sonucuna ulaşır. Çünkü eğer sonuç sıfır gelirse o ürünü hiçbir kullanıcının oylamadığı çıkarımı yapılabilir. Polat ve Du [24] iki tane gizlilik koşulu tanımlamışlardır. Bunlardan ilki ürünlerin gerçek değerlerini saklamaktır. Diğer gizlilik koşulu ise hangi ürünlerin oylandığının saklanmasıdır. Önceki örnekte görüldüğü gibi uygulanan gizlilik koşulları nedeniyle kullanıcıların verilerini ve hangi ürünleri oyladıklarını elde edemiyoruz. Fakat ikinci gizlilik koşulunu ihlal edersek, yani hangi kullanıcının hangi ürünleri oyladığı bilgisini yardımcı bilgi olarak dışardan alırsak, bu durumda kullanıcıların verilerine ulaşıp ulaşamayacağımızı inceledik. Bu durumda bir kez oylanmış ürünlerden başlayarak bazı bilgileri elde edebiliriz. Şekil 3'te dördüncü, yedinci ve onuncu ürünlerin bir kez oylandığını ve yerlerini bildiğimizi varsayalım. Aktif kullanıcı sadece bu bilgi sayesinde B_D değerlerinden bu ürünlerin değerlerini elde eder. Daha sonra bu kullanıcıların (birinci, ikinci ve beşinci) hangi ürünleri oyladığını ve değerlemeleri kendi oluşturmuş olduğu değerleme vektörünü kullanarak tahmin eder. Bu saldırıda önemli olan her seferinde bir ürünün değerini değiştirerek kullanıcıdan gelen B_N değerlerine göre o ürünün oylanıp oylanmadığını veya oylandı ise değerinin ne olduğunu elde etmektir. Benzer şekilde iki kez oylanmış ürünlerin yerleri yine yardımcı bilgi olarak elde edilirse, gelen ara toplam değerlerden daha önce elde etmiş olduğumuz veriler kullanılarak bu ürünlerin değerleri bulunur.

Örneğin Şekil 4'te beşinci ürünün iki kullanıcı tarafından oylandığını bildiğimizi varsayalım. B_D değerinden normalde biz sadece bu iki kullanıcının *z-skor* değerlerinin toplamını elde ediyoruz. Toplam değerden bu iki kullanıcının *z-skor* değerlerini tahmin etmemiz imkânsızdır. Çünkü elimizde bir eşitlik ve iki bilinmeyen olduğundan sonsuz çözüm olur. Fakat varsayımlarımızı kullanarak ürünlerin yerleri önceden bilinirse (Şekil 3'deki gibi), ikinci kullanıcının ürünlerinin değerlerini tahmin ettiğimizden ve beşinci ürünün değerini bildiğimizden, gelen toplam değerden kolaylıkla dördüncü kullanıcının beşinci ürüne verdiği değerleme bulunur. Aynı saldırı tekniği uygulanarak dördüncü kullanıcının sadece beşinci ürüne verdiği değerleme değeri bulunursa, dördüncü kullanıcının bütün değerleme değerleri başka bir yardımcı bilgi olmadan kolaylıkla bulunabilir. Sahip olduğumuz yardımcı bilgiler arttıkça elde ettiğimiz bilgiler de o oranda artmaktadır.

4.2. Dikey Bölünmüş Veride Saldırı Senaryoları
(Attack Scenarios on Vertically Partitioned Data)

Veriler dikey olarak bölündüğünde aktif kullanıcı öneri istediği *q* ürünü hangi partide ise o partiye değerleme vektörünü gönderir. Partilerden biri aktif kullanıcı gibi davranarak karşı partinin verisini elde etmeye çalışır. Örneğin aktif kullanıcının öneri istediği ürün *B* partisinde olsun. *B* partisi $A'_N + B_N$ ve B_D değerlerini hesaplayarak *A* partisine gönderir ve *A* partisi kendi A_D değerini hesaplayarak aktif kullanıcıya bir sonuç döndürür. *B* partisi kendi verileri hakkında *A* partisi bir çıkarım yapamasın diye yatay bölmede olduğu gibi aynı protokolü uygular [23]. Bu protokol sayesinde aktif kullanıcının değerleme vektörüne ekleme/çıkarım yapıldıktan sonra $A'_N + B'_N$ ve B'_D değerleri hesaplanır. Bir önceki bölümde açıklandığı gibi her seferinde farklı ürünlerin eklenmesi/çıkarılması bu ara toplamlardan bir sonuç çıkarılmamasını garantiler. DBV durumunda, çevrim-dışı hesaplama kısmında iki parti veri alışverişinde bulunurlar [23]. Çünkü partiler ürünlerin yarısına sahip olduklarından diğer yarısındaki ürünlerin değerleri çevrimiçi hesaplamalarda gereklidir. Örneğin *B* partisi çevrimiçi hesaplamada $A'_N + B'_N$ değerindeki A'_N değerini hesaplamak için önceden $\sum^{''}$A matrisini hesaplar. Fakat hem permütasyon fonksiyonlarının kullanılması hem de homomorfik şifrelemenin kullanılması bu alışveriş sırasında veri elde etmeyi engeller. Ayrıca tüm şifrelenmiş son çarpım sonuçlarından (∑A veya ∑B) karşı partinin verileri elde etmemesi için rasgele sayılar üreterek bu sayılar şifrelenip bu çarpım sonuçlarına eklenerek karşı partiye gönderilir. Karşı parti şifrelenmiş son çarpım sonuçlarının şifresini çözdüğünde gerçek değerleri elde edemez. Bu önlemler verinin karşı parti tarafından elde edilmesini engellerler.

5. DENEYLER (EXPERIMENTS)

Dördüncü bölümde önerilen protokol kullanıldığında aktif kullanıcı atağı ile neden veri elde edilemeyeceği açıklanmıştır. Bu bölümde eğer partiler önerilen protokolü kullanmadan veri alışverişinde bulunursa, aktif kullanıcı atağı ile partilerden ne kadar veri elde edilip edilemeyeceği her iki durumda da incelenmiştir.

	Ürün 1	Ürün 2	Ürün 3	Ürün 4	Ürün 5	Ürün 6	Ürün 7	Ürün 8	Ürün 9	Ürün 10
Kullanıcı 1										
Kullanıcı 2	X				X				X	
Kullanıcı 3										
Kullanıcı 4					X					
Kullanıcı 5	X								X	

Şekil 4. Yardımcı bilgi: 2 kez oylanmış ürünlerin yerleri (Auxiliary information: locations of the two times-rated items)

5.1. Veri Seti Ve Değerlendirme Metriği
(Data Set and Evaluation Metric)

Deneylerde yaygın olarak kullanılan gerçek veri seti MovieLens kullanılmıştır. MovieLens 943 kullanıcı ve 1682 ürüne sahiptir. Değerleme değerleri için 1 (en düşük) ile 5 (en yüksek) arasında değişen 5 yıldızlı ölçek kullanılmıştır. Değerlendirme metriği olarak doğruluk oranı seçilmiştir. Doğruluk oranı doğru tahmin edilen değerleme sayısının toplam değerleme sayısına bölünmesi ile hesaplanır.

5.2. Metodoloji (Methodology)

Deneylerde kullanılan veri seti yatay ve dikey olarak iki parti arasında bölünmüştür. Yatay bölmede, *A* partisi 471 kullanıcı ve 1682 ürüne sahipken, *B* partisi 472 kullanıcı ve 1682 ürünü içerir. Dikey bölmede ise partilerin her ikisi de 943 kullanıcıya ve 841 ürüne sahiptir. Deneylerde saldırı yapacak parti aktif kullanıcı değerleme vektörünü anlık olarak oluşturur. Aktif kullanıcı değerleme vektörü rasgele olarak 100 ürün seçilerek yine rasgele olarak değerleme değerleri atanarak oluşturulur. Bir ürünün değerini elde etmek için iki kez öneri istemek yeterlidir. Sadece aktif kullanıcı değerleme vektöründe o ürünün değeri değiştirilerek ara toplamlardan sonuç elde edilir. Her ürünün değerleme değerlerini öğrenmek için aynı aktif kullanıcı değerleme vektörü kullanılabileceği gibi, her ürün için farklı değerleme vektörleri oluşturulabilir. Deneylerde her kullanıcının her ürünü için farklı aktif kullanıcı değerleme vektörleri kullanılmıştır. Her seferinde sadece tek bir değer değiştirerek aynı sorguyu göndermek yerine her ürün için farklı aktif kullanıcı değerleme vektörleri gönderilmiştir.

5.3. Deney Sonuçları (Experimental Results)

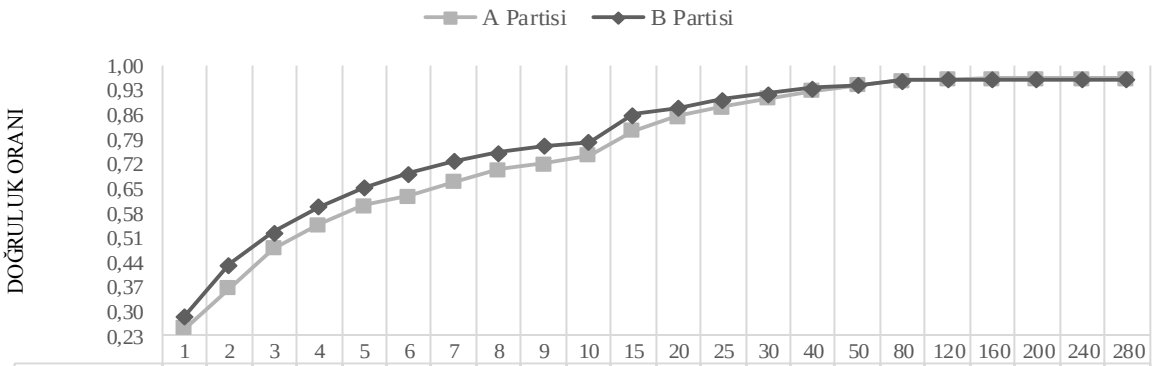
Deneyler iki bölüme ayrılmıştır. İlk önce veriler yatay olarak bölündüğünde partilerin veri elde etmesi incelenmiştir. Daha sonra veriler dikey olarak bölündüğünde partilerin ne kadar veri elde ettikleri incelenmiştir.

Deney 1: İlk deneyde veriler yatay olarak bölündüğünde *A* partisinin *B* partisinin verisinin ne kadarını tahmin ettiği incelenmiştir. Benzer şekilde *B* partisi aktif kullanıcı gibi davranarak *A* partisinin verisini elde etmeyi amaçlamıştır. Bu deneyde MovieLens veri seti direk olarak ortadan yatay olarak ikiye bölünmüştür. Bu durumda *A* partisi toplamda 53219 değerlemeye sahipken *B* partisi 46781 değerlemeye sahiptir. Şekil 5'deki sonuçlara bakıldığında, yardımcı bilgi olarak bilinen ürün sayısının küçük değerlerinde *B* partisinin verilerini elde etmedeki doğruluk oranının *A* partisine göre daha yüksek olduğu görülmektedir. Ürün sayısı arttıkça değerler birbirine çok yaklaşmakta ve sonuç olarak yaklaşık her iki partinin de %96 verisi doğru olarak tahmin edilmektedir. Örneğin *A* partisinin 165 tane ürününün bir kez oyladığını ve bu değerlemelerin yerlerini bildiğimizde *A* partisinin tüm verisinin yaklaşık %25'ini, yani 13251 değerlemeyi, doğru olarak tahmin ediyoruz. Aslında *A* partisinin 165 tane ürünü yerine 54 tane ürününü bildiğimizde de aynı sonucu elde edebiliyoruz çünkü yalnızca bir kez oylanmış ürünleri sadece belli kullanıcılar oyladıkları için bu kullanıcıların bir tek değerleme yerini bilmek kullanıcının bütün değerleme vektörünü elde etmek için yeterlidir. Bu durumda *A* partisinde 54 farklı ürünün bir kez oylandığını ve yerlerini bilindiğinde 13251 değerlemeyi yani yaklaşık verinin %25'ini elde edebiliyoruz.

Deney 2: Verileri direk olarak ikiye bölmek yerine rasgele kullanıcıları iki parti arasında böldükten sonra saldırı senaryosu sonucunda ne kadar veri elde ettikleri hesaplanmıştır. Şekil 6'daki sonuçların Şekil 5'deki sonuçlara benzer olduğu görülmektedir. Benzer şekilde *B* partisinin ilk değerler için daha iyi sonuç verdiği görülmektedir. Şekil 5 ve Şekil 6'nın benzer sonuçları vermesindeki en büyük neden her ikisinde partilerin sahip oldukları değerlemeler birbirine yakındır. Başka bir ifadeyle, veri yatay olarak ikiye bölündüğünde partilerin sahip olduğu ürün sayısı kullanıcı sayısına göre fazla olduğundan, kullanıcıların farklı ürünlere değerleme vermesini bekleriz. Bu da uyguladığımız yardımcı bilginin saldırı sırasında bize beklediğimiz gibi yardım etmesine neden olur. Bir kullanıcının oylamış olduğu bir değerlemenin yerini bilmek ve özellikle bu ürünün bir kez oylanmış olması o kullanıcının örneğin 500 ürün oylamış ise hangileri olduğunu bilmesek bile hepsinin değerini ve hangileri olduğunu bilmemize yardım eder.

Deney 3: MovieLens veri seti ortadan dikey olarak ikiye bölündüğünde partilerin gerçekleştirmiş olduğu saldırı sonucu ne kadar veri elde edebileceği bu deneyde incelenmiştir. Direk veriyi ikiye böldüğümüzde MovieLens veri setinin yapısından dolayı *A* partisi 86993 değerlemeye sahipken *B* partisi 13007 değerlemeye sahiptir. MovieLens veri setinde son ürünlerde çok fazla değerleme değeri bulunmamaktadır. Bu nedenle direk veri seti bölündüğünde *A* partisinin değerleme sayısı fazladır. Veriler yatay bölündüğünde olduğu gibi partiler aktif kullanıcı gibi davranarak ara toplamlardan karşı partinin verisini elde etmeye çalışırlar. Şekil 7'de görüldüğü gibi yardımcı bilgi bile kullanılsa, *A* partisinin verisini elde etmedeki başarısı çok düşüktür. Çünkü *A* partisinin verisi çok yoğun olduğundan bir kez oylanmış ürünlerin sayısı çok azdır. Bu nedenle veriler sürekli toplam değerlerden oluştuğundan sadece o ürünler için toplam değerler bulunurken kullanıcıların değerleme değerlerine ulaşılamaz. Diğer yandan *B* partisinin verisi seyrek olduğundan toplamda verisinin %40'ı kadarı tahmin edilebilir.

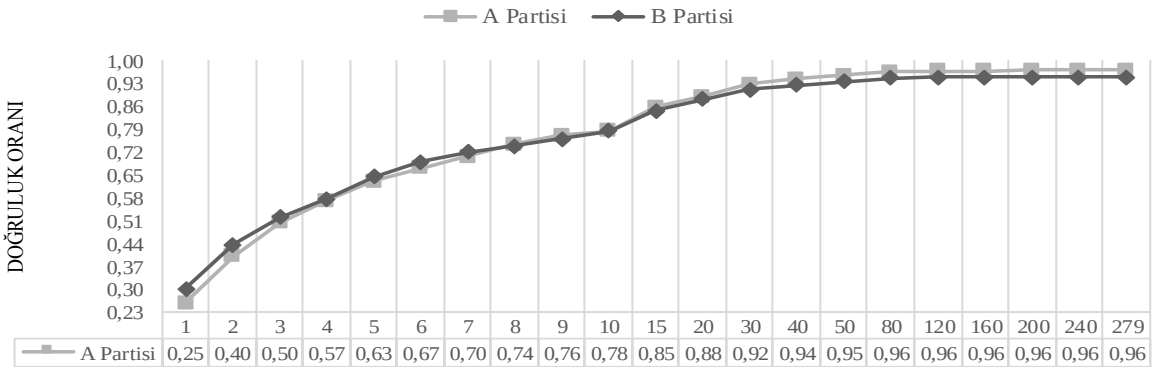
Deney 4: Veriler dikey olarak rasgele iki parti arasında bölündüğünde saldırı sonucunda partilerin ne kadar veri elde ettikleri araştırılmıştır. Veriler rasgele olarak ikiye bölündüğünde her iki parti de ortalama veri setinin yarısı kadar (50000) değerlemeye sahiptir. Önceki deneyde *A* partisinin veri seti çok yoğun olduğunda yardımcı bilgi kullanılsa bile ara toplamlardan çok az bir çıkarım yaptığı gösterilmiştir.



	1	2	3	4	5	6	7	8	9	10	15	20	25	30	40	50	80	120	160	200	240	280
A Partisi	0,24	0,36	0,47	0,54	0,60	0,62	0,66	0,70	0,72	0,74	0,81	0,85	0,88	0,90	0,92	0,94	0,95	0,96	0,96	0,96	0,96	0,96
B Partisi	0,28	0,43	0,52	0,59	0,65	0,69	0,72	0,75	0,77	0,78	0,85	0,87	0,90	0,92	0,93	0,94	0,95	0,95	0,96	0,96	0,96	0,96

YARDIMCI BİLGİ OLARAK BİLİNEN ÜRÜN SAYISI

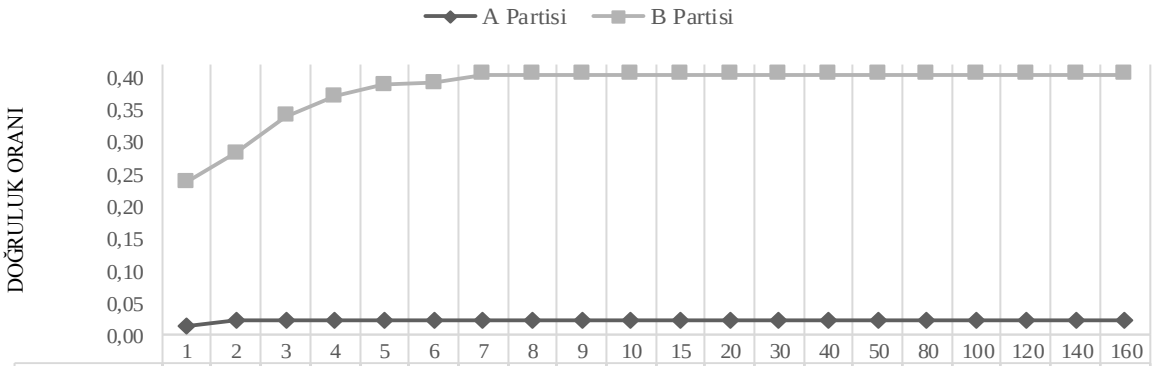
Şekil 5. YBV'de doğruluk oranları (Accuracy rates on HPD)



	1	2	3	4	5	6	7	8	9	10	15	20	30	40	50	80	120	160	200	240	279
A Partisi	0,25	0,40	0,50	0,57	0,63	0,67	0,70	0,74	0,76	0,78	0,85	0,88	0,92	0,94	0,95	0,96	0,96	0,96	0,96	0,96	0,96
B Partisi	0,29	0,43	0,51	0,57	0,64	0,68	0,71	0,73	0,76	0,78	0,84	0,88	0,91	0,92	0,93	0,94	0,94	0,94	0,94	0,94	0,94

YARDIMCI BİLGİ OLARAK BİLİNEN ÜRÜN SAYISI

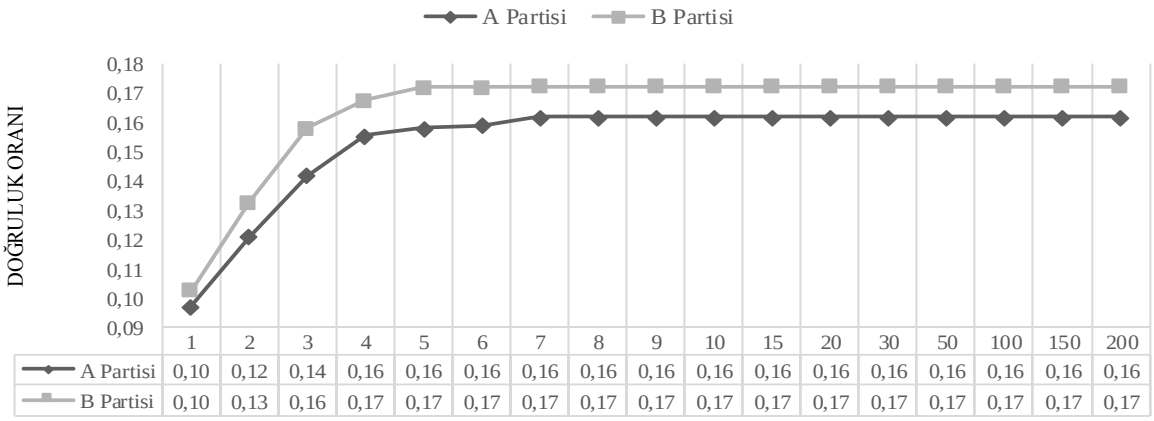
Şekil 6. Rasgele YBV'de doğruluk oranları (Accuracy rates on random HPD)



	1	2	3	4	5	6	7	8	9	10	15	20	30	40	50	80	100	120	140	160
A Partisi	0,01	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02
B Partisi	0,23	0,28	0,33	0,37	0,38	0,39	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40	0,40

YARDIMCI BİLGİ OLARAK BİLİNEN ÜRÜN SAYISI

Şekil 7. DBV'de doğruluk oranları (Accuracy rates on VPD)



	1	2	3	4	5	6	7	8	9	10	15	20	30	50	100	150	200
A Partisi	0,10	0,12	0,14	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,16	0,16
B Partisi	0,10	0,13	0,16	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17

Şekil 8. Rasgele DBV'de doğruluk oranları (Accuracy rates on random VPD)

Bu deney partiler yaklaşık olarak eşit değerleme sayısına sahip olursa elde ettikleri veri miktarı ne kadar olur sorusuna cevap aramak için gerçekleştirilmiştir. Şekil 8'deki sonuçlar incelendiğinde partilerin birbirinden elde ettikleri değerlerin yaklaşık olarak birbirine eşit olduğu görülmektedir. *A* partisi *B* partisinden biraz daha fazla değerlemeye sahip olduğundan, *A* partisinin değerleri *B* partisine göre biraz düşüktür. Şekil 7 ve Şekil 8 sonuçları karşılaştırıldığında verilerin seyrek olmasının veri elde etmekte önemli bir unsur olduğu görülür. OF sistemleri düşünüldüğünde, verilerin genelde çok seyrek olduğu görülmektedir. Örneğin *B* partisi deney 3'te çok seyrek bir veri setine sahipken deney 4'te daha yoğun bir veri seti içerir. Buradaki bir diğer önemli nokta ise ürün sayısı nispeten daha azken kullanıcı sayısının fazla olmasıdır. Seyrek veya az olarak oylanan ürün bulmak zordur. Bu nedenle kullanılan yardımcı bilgi deney 1 ve deney 2'deki kadar sonuçları elde etmede başarılı olamaz.

6. SONUÇLAR VE TARTIŞMALAR (RESULTS AND DISCUSSIONS)

İki parti arasında yatay veya dikey olarak bölünmüş veriler, daha doğru öneriler sunmak için birleştirilebilir. Partiler gizliliklerini koruyarak bölünmüş veriden öneriler sunacak sistemleri kullanabilirler. Bu sistemlerde kullanılan protokoller sayesinde aktif kullanıcı gibi hareket etme saldırısında partilerin gizli verileri elde etmesi zordur. Yaptığımız analizler aktif kullanıcının değerleme vektörünün bazı değerlemelerinin rasgele doldurulması/çıkarılması sonucunda her seferinde farklı ara toplamlar elde edildiğinden buradan bir çıkarım yapılamayacağını göstermiştir. Diğer önemli nokta ise aktif kullanıcının değerleme vektörünün doğruluğunu etkilememek için protokol kullanılmadığını düşündüğümüzde ise sadece ürünlere verilen değerlemelerin toplamına erişebiliyoruz. Sadece toplam değerler elde edildiği için herhangi bir ürünü kaç kullanıcının oyladığı veya hangi kullanıcıların oyladığı bilgisine ulaşmak zordur. Tüm bunların yanında bir takım varsayımlardan yararlanarak bazı bilgileri elde edip edemeyeceğimizi test ettik. Daha önce belirtildiği gibi gizlilik tanımı olarak sadece kullanıcıların gerçek değerlerinin saklanması olarak kabul edersek ve kullanıcıların bazı ürünlerinin yerlerini bildiğimizi varsayarak bir takım bilgileri elde ettiğimizi deneylerle gösterdik. Deney sonuçları karşılaştırıldığında yatay olarak bölünme durumunda veriyi tahmin etme oranı dikey olarak bölünme durumundaki veriyi tahmin etme oranından daha yüksek olduğunu gördük. Buradaki en önemli neden ise veri yatay olarak bölündüğünde ürün sayısının kullanıcı sayısından daha çok olmasıdır. Kullanıcı sayısı az iken ürün sayısının fazla olması farklı kullanıcıların farklı ürünlere oy verme olasılığını arttırmaktadır. Eğer yeteri kadar kullanıcının farklı ürünlere oy verdiği bilgisi bilinirse verinin büyük bir kısmı kolaylıkla elde edilebilir. Böylelikle kullanıcıların farklı farklı ürünleri oyladıkları bilgisinden yararlanarak gerçek *z-skor* verilerine erişebildik. Diğer yandan veriler dikey olarak bölündüğünde, veri setinin yoğunluğu saldırı sırasında önemli bir etkidir. Çünkü ürün sayısı az ve bu ürünleri oylayan kullanıcı sayısı fazla olduğunda farklı kullanıcıların aynı ürünlere oy verme olasılığı artmaktadır. Bu nedenle bu kullanıcıların oyladıkları ürünlerin yerlerini bilsek bile tek bir toplam değerden kullanıcıların gerçek değerlerine ulaşılamaz. Diğer yandan rasgele dikey bölünmede veri setlerinin yoğunlukları eşitlenmekte ve elde edilen başarı oranı %17 civarındadır. Fakat veriler rasgele yatay olarak bölündüğünde veya direk bölündüğünde, başarı oranları değişmemektedir. Yaklaşık verilerin %96'sı tahmin edilmektedir. Kullanıcı sayısının az olması ve ürün sayısının fazla olması daha seyrek bir dağılıma sebep olur. Bu çalışmamızda iki partinin ortak filtreleme algoritmalarını uygulamak istediklerinde veriler yatay veya dikey olarak birleştirildiğinde birbirlerinden veri elde edip edemeyeceğinin analizi yapılmıştır. Fakat veriler iki parti arasında değil de daha çok parti arasında birleştirilmek istendiğinde partilerin birbirinin verilerini elde etmek için nasıl saldırı senaryoları gerçekleştirileceği gelecek çalışmalarımızda araştırma konusu olarak incelenecektir.

TEŞEKKÜR (ACKNOWLEDGEMENT)

Bu çalışma TUBİTAK tarafından 113E262 numaralı proje ile desteklenmiştir.

KAYNAKLAR (REFERENCES)

1. Avcı E., Tuncer T., Avcı D., İkili imgeler için mayın tarlası oyunu tabanlı yeni bir veri gizleme algoritması, Journal of the Faculty of Engineering and Architecture of Gazi, 31 (4), 951-959, 2016

2. Shi Y., Larson M., Hanjalic A., Collaborative Filtering Beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges. ACM Comput. Surv. 47 (1), 1-45, 2014.

3. Koren Y., Bell R., Advances in Collaborative Filtering, Recommender Systems Handbook, Editör: Ricci, F., Rokach, L., Shapira, B., Springer US, Boston, MA, 77–118, 2015.

4. Canny J., Collaborative filltering with privacy via factor analysis, Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 238-245, 11-15 Ağustos, 2002.

5. Cranor L.F., I didn't buy it for myself': Privacy and ecommerce personalization, Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society, Washington, DC, USA, 111-117, 27-30 Ekim, 2003.

6. Bilge A., Polat H., A Comparison of Clustering-based Privacy-preserving Collaborative Filtering Schemes. Appl. Soft Comput. 13 (5), 2478–2489, 2013.

7. Li D., Chen C., Lv Q., Shang L., Zhao Y., Lu T., Gu N., An algorithm for efficient privacy-preserving item-based collaborative filtering. Futur. Gener. Comput. Syst. 55, 311–320, 2016.

8. Bilge A., Kaleli C., Yakut I., Gunes I., Polat H., A survey of privacy-preserving collaborative filltering schemes, Int. J. Software Eng. Knowl. Eng., 23 (8), 1085-1108, 2013.

9. Okkalioglu M., Koc M., Polat H., On the privacy of horizontally partitioned binary data-based privacy-preserving collaborative filtering, Lect. Notes Comput. Sci., 9481, 199-214, 2015.

10. Okkalioglu M., Koc M., Polat H., A Privacy Review of Vertically Partitioned Data-based PPCF Schemes, International Journal of Information Security Science, 5 (3), 51-68, 2016.

11. Zhang S., Ford J., Makedon, F., Deriving private information from randomly perturbed ratings, Proceedings of the 6th SIAM International Conference on Data Mining, Bethesda, MD, USA, 59-69, 20-26 Nisan, 2006.

12. Okkalioglu M., Koc M., Polat H., On the discovery of fake binary ratings, Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, 901-907, 13-17 Nisan, 2015.

13. Demirelli Okkalioglu B., Koc M., Polat H., Reconstructing rated items from perturbed data, Neurocomputing, 207, 374-386, 2016.

14. Okkalioglu B.D., Okkalioglu M., Koc M., Polat H., A survey: Deriving private information from perturbed data, Artificial Intelligence Review, 44 (4), 547-569, 2015.

15. Kargupta H., Datta S., Wang Q., Sivakumar K., Random-data perturbation techniques and privacy-preserving data mining, Knowledge and Information Systems, 7 (4), 387-414, 2005.

16. Huang Z., Du W., Chen B., Deriving private information from randomized data, Proceedings of the 24th ACM SIGMOD International Conference on Management of Data, Baltimore, MD, USA, 37-48, 14-16 Haziran, 2005.

17. Guo S., Wu X., Li, Y., Determining error bounds for spectral filtering based reconstruction methods in privacy preserving data mining, Knowledge and Information Systems, 17 (2), 217-240, 2008.

18. Polat H., Du W., Privacy-preserving collaborative filtering using randomized perturbation techniques, Proceedings of the 3rd IEEE International Conference on Data Mining, Melbourne, FL, USA, 625-628, 19-22 Kasım, 2003.

19. Calandrino J.A., Kilzer A., Narayanan A., Felten E.W., Shmatikov V., You might also like: Privacy risks of collaborative filtering, Proceedings of the 2011 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 231-246, 22-25 Mayıs, 2011.

20. Polat H., Du W., Privacy-preserving top-n recommendation on horizontally partitioned data, Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, Compiègne, France, 725-731, 19-22 Eylül, 2005.

21. Polat H., Du W., Privacy-preserving top-n recommendation on distributed data, Journal of the American Society for Information Science and Technology, 59 (7), 1093-1108, 2008.

22. Polat H., Privacy-preserving collaborative filtering, Doktora Tezi, Syracuse University, Computer and Information Science, Syracuse, NY, 2006.

23. Polat H., Du W., Privacy-preserving collaborative filtering on vertically partitioned data, Lect. Notes Comput. Sci., 3721, 651-658, 2005.

24. Polat H., Du W., Achieving private recommendations using randomized response techniques, Lect. Notes Comput. Sci., 3918, 637-646, 2006.