

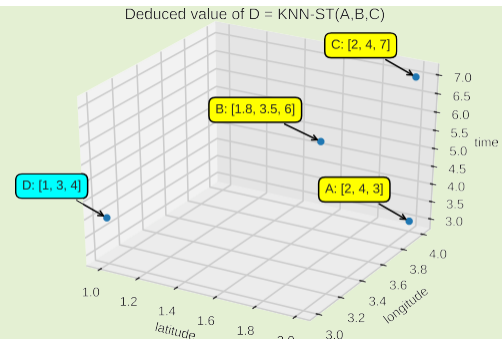


KNN-ST: Çevresel Kalabalık Algılamada Eksik Veri Çıkarsaması için Uzun-Zamansal Korelasyondan Yararlanma

Ningrinla Marchang, Kıdemli Üye, IEEE, ve Rakesh Tripathi, Üye, IEEE

Özet -Dar mobil kitle algılama, algılama görevlerinin sayısını önemli ölçüde azaltarak genel algılama maliyetini düşürmek için zaman içinde farklı konumlarda algılanan veriler arasındaki uzamsal ve zamansal korelasyondan yararlanan yeni bir kitle algılama paradigmasıdır. Sonuç olarak, yalnızca seyrek olarak seçilen uzamsal-zamansal hücreler algılanan verileri bildirirken, hücrelerin geri kalanı için verilerin algılanan verilerden çıkarılması gerekecektir. Büyük ölçüde kayıp veri çıkarımı olarak bilinen bu süreç bu çalışmanın odak noktasıdır. Nispeten daha hızlı ve basit olduğu bilinen KNN (K-En Yakın Komşu) yaklaşımını inceliyoruz. Ancak, algılanan veri seyrek olduğunda genellikle kötü performans gösterdiği kabul edilir. İçinde

Çevresel kalabalık algılama bağlamında, sadece uzamsal veya zamansal korelasyonu bağımsız olarak kullanmak yerine, verilerin uzamsal-zamansal korelasyonunu algoritmaya dahil ettiğimizde uygulanabilir bir kayıp veri çıkarım yaklaşımı olup olmadığını inceliyoruz. Bu nedenle, KNN'nin üç varyantını inceliyoruz: Seyrek veriler üzerinde KNN-ST (KNN-Uzamsal-Zamansal), KNN-S (KNN-Uzamsal) ve KNN-T (KNN-Zamansal). Ayrıca, voksellerin algılanan verilerin uzamsal-zamansal özelliklerinden ve dolayısıyla bunlar arasındaki uzamsal-zamansal korelasyondan yararlanmanın doğal bir yolu olduğunu bulduk. İlginç bir şekilde, KNN-ST'nin kayıp olasılığı 0,9 kadar yüksek olduğunda bile gerçekten iyi performans (yaklaşık 0,1'lik normalleştirilmiş mutlak hata) gösterdiğini bulduk. Ayrıca, aynı deneysel veri kümeleri üzerinde mevcut bir yöntemi uyguluyor ve ilgili karşılaştırmalı simülasyon sonuçlarını sunuyoruz.



I. GİRİŞ

Dijital Nesne Tanımlayıcı 10.1109/JSEN.2020.3024976

MOBİL kitlesel algılama (MCS), 'algılama ve bilgi işlem cihazlarına sahip bireylerin toplu olarak veri paylaşmasıdır ve ortak ilgi alanındaki olguları ölçmek ve haritalamak için bilgi çıkarır' [1]. Seyrek mobil kitle algılama, algılama görevlerinin sayısını önemli ölçüde azaltmak için zaman içinde farklı konumlarda algılanan veriler arasındaki uzamsal ve zamansal korelasyondan yararlanan yeni bir kitle algılama paradigmasıdır [2]. Algılama alanı alt alanlara veya hücrelere bölünmüştür.

Makale 24 Ağustos 2020'de alındı; 13 Eylül 2020'de kabul edildi. Yayın tarihi 21 Eylül 2020; güncel versiyon tarihi 6 Ocak 2021. Bu çalışma kısmen Bilim ve Mühendislik Araştırma Kurulu, Bilim ve Teknoloji Bakanlığı (SERB-DST), Hindistan Hükümeti, Grant EEQ/2017/000083 tarafından desteklenmiştir. Bu makalenin gözden geçirilmesini koordine eden ve yayınlanması için onay veren yardımcı editör Prof. (Sorumlu yazar: Ningrinla Marchang.) Ningrinla Marchang, Bilgisayar Bilimleri ve Mühendisliği Bölümü, Kuzey Doğu Bölgesel Bilim ve Teknoloji Enstitüsü, Itanagar 791109, Hindistan'da çalışmaktadır (e-posta: ningrinla@gmail.com; nm@nerist.ac.in).

Rakesh Tripathi, Raipur Ulusal Teknoloji Enstitüsü, Bilgi Teknolojileri Bölümü, Raipur 492010, Hindistan'da çalışmaktadır (e-posta: rtripathi.it@nitrr.ac.in).

Bu alt alanlardan algılanan veriler, özellikle sıcaklık gibi çevresel koşulların kitle algılamasında uzamsal ve zamansal korelasyon sergileme eğilimindedir. Bu nedenle, yalnızca seçilen bazı hücrelerin algılanması gerekir ve kalan hücreler için veriler, seçilen hücrelerin algılanan verilerinden çıkarılabilir. Açıkçası, seyrek MCS'deki önemli işlevlerden biri *Eksik Veri Çıkarsaması*dır. Kayıp verilerin (algılama için seçilmeyen hücrelerin verileri) algılanan verilerden ne kadar iyi çıkarılabileceği, algılama görevlerinin sayısındaki azalmanın miktarını ve dolayısıyla genel algılama maliyetindeki azalmayı belirleyecektir. Dahası, çıkarımın

doğruluğu, bu şekilde çıkarılan verilerin kalitesini ve dolayısıyla verilerin kullanılabilirliğini belirleyecektir. Bu nedenle, seyrek MCS için kayıp veri çıkarım yöntemlerinin iyi performans göstermesi çok önemlidir. Ayrıca, yöntemlerin çevrimiçi olarak uygulanabilmesi için çok zaman alıcı olmaması da aynı derecede önemlidir, yani MCS uygulama çalışıyor.

Bazı temsili kayıp veri çıkarım yaklaşımları K-En Yakın Komşular (KNN) [3], Delaunay Üçgenleme (DT) [4], Çok Kanallı Tekil Spektrum Analizi (MSSA) [5] ve Sıkıştırılmış Algılama (CS) [6]-[8]. Dışında

1558-1748 © 2020 IEEE. Kişisel kullanıma izin verilir, ancak yeniden yayınlama / yeniden dağıtım IEEE izni gerektirir.
Daha fazla bilgi için <https://www.ieee.org/publications/rights/index.html> adresine bakınız.

Bu yaklaşımlar arasında KNN'nin en az zaman karmaşıklığına sahip olduğu bilinmektedir [9]. Bu nedenle, MCS'de eksik veri çıkarımı için cazip bir adaydır. Bununla birlikte, dezavantajı, kayıp değerler daha fazla olduğunda nispeten düşük performans gösterdiğinin bilinmesidir [9], [10]. Bu çalışmanın motivasyonu da burada yatmaktadır. KNN'nin çıkarım yöntemlerinin karşılaştırmalı çalışmasında kullanılmasına rağmen, verilerin ya uzamsal korelasyonunun ya da zamansal korelasyonunun kullanıldığını görüyoruz [10], [11], [17]. Merak edilen soru şudur: "KNN'ye bu iki korelasyonu birlikte dahil edersek, çıkarım doğruluğu artar mı?". Bu nedenle, KNN'nin üç çeşidini formüle ediyoruz: a) KNN-S (K-En Yakın Komşular-mekansal), b) KNN-T (K-En Yakın Komşular-Zamansal) ve c) KNN-ST (K-En Yakın Komşular-Uzamsal-Zamansal). Ayrıca, vokseleştirme algılanan verilerin uzamsal-zamansal özelliklerinden ve dolayısıyla bunlar arasındaki uzamsal-zamansal korelasyondan yararlanmanın doğal bir yolu olduğunu görüyoruz. Vokseleştirme basitçe 'bir hacmin daha küçük sabit boyutlu hacimler koleksiyonuna bölünmesi' olarak tanımlanabilir. Böylece, ilerleyen bölümlerde görüleceği üzere, vokseleştirme ile KNN'ye uzamsal-zamansal korelasyonu kolayca dahil edebiliriz. Bu çalışmanın katkısının özeti şudur:

1. KNN'nin üç çeşidini sunuyoruz, yani KNN-S, KNN-T ve çevresel kitle algılamada eksik veri çıkarımı için KNN-ST.
2. Vokseleştirme kullanarak algılanan verilerin uzamsal-zamansal korelasyonundan yararlanıyoruz.
3. KNN-ST'deki ağırlıklandırma parametresini makul bir şekilde ayarlayarak algılanan verilerin uzamsal ve zamansal korelasyonunun kapsamından nasıl yararlanılabileceğini gösteriyoruz.
4. Önerilen yaklaşımı simülasyon sonuçları yardımıyla mevcut bir yöntemle [17] karşılaştırıyoruz.

Makalenin geri kalanı aşağıdaki şekilde düzenlenmiştir. Bölüm II'de ilgili mevcut çalışmalar incelenmekte, Bölüm III'te önerilen şemaların geliştirilmesi açıklanmaktadır. Bölüm IV'te bir performans karşılaştırma çalışması verilmekte ve bunu Bölüm V'teki sonuçlar takip etmektedir.

II. İLGİLİ ÇALIŞMA

Kayıp veri ve çıkarım (veya enterpolasyon) üzerinde çok çalışılmış bir konudur. Uzamsal-zamansal enterpolasyon problemini çözmek için oldukça fazla sayıda yöntem önerilmiştir [18]-[23]. [18] ve [21]'de sinir ağı kullanılmıştır. K-En Yakın Komşu (KNN) [3], eksik değeri çıkarmak için en yakın K komşunun değerlerini kullanan basit bir yerel enterpolasyon yöntemidir. KNN, [24] ve [25]'te kısa vadeli trafik akışı tahmini için kullanılmıştır. Ancak, biz seyrek MCS ile eksik veri çıkarımı ile ilgileniyoruz. Delaunay Üçgenleme (DT) yöntemleri, toplanan verileri köşe olarak kabul ederek veri enterpolasyonu için sanal üçgenler oluşturur. Veriye uyarlanabilir ve parametrik olmayan bir yöntem olan Çok Kanallı Tekil Spektrum Analizi (MSSA) [5], özellikle coğrafi ve meteorolojik veriler için kullanılmaktadır. Sıkıştırılmış Algılama (CS) [6]-[8] iyi kullanılan bir yöntemdir. Kablosuz sensör ağlarında eksik verilerin yeniden yapılandırılması [9], sonda araçlarla kentsel trafik algılama [17] ve İnternet trafiği [10] için kullanılmıştır. Ayrıca Sıkıştırılmış Kalabalık

Algılama [11] yardımıyla çevrimiçi görev tahsisi için de kullanılır. Matris tamamlama, eksik veri çıkarımı için de kullanılır. Bu bir yöntemdir

Sunulan kısmi doldurulmuş veri matrisindeki eksik değerlerin doldurulması. Matris tamamlama için literatürde konveks optimizasyon [12], düşük rütbeli matris tamamlama [13], sezgisel açgözlü algoritma [14] ve Sinir Ağı [15], [16] gibi çeşitli yaklaşımlar tartışılmıştır. Tüm bu yöntemler arasında KNN'nin nispeten daha hızlı çalıştığı bilinmektedir. Dahası, algoritma çok basittir. Ancak, veri seyrek olduğunda kötü performans gösterme eğilimindedir. Çalışma süresinin bir sorun olduğu senaryolarda, kesinlikle çekici bir yöntemdir. Bu çalışma, verilerin uzamsal-zamansal özelliklerinden yararlanmak ve bunları KNN'ye dahil etmek için vokselleştirme yöntemini araştırmakta ve sonuçta KNN-S, KNN-T ve KNN-ST olmak üzere üç varyant ortaya çıkmaktadır. Vokselleştirme [26]'da havadan LIDAR (Light Detection and Ranging) nokta bulutlarındaki küme aykırı değerlerini tespit etmek için kullanılmıştır.

III. ÖNERİLEN PROGRAMLAR

Bu bölümde önerilen şemaların gelişimi sunulmaktadır. İlk olarak, verilerin uzamsal-zamansal özelliklerinden faydalanmak için vokselleştirmenin nasıl yapıldığını gösteriyoruz. Ardından, algılanan çevresel verilerin hem uzamsal hem de zamansal korelasyonunun neden dikkate alınması gerektiğini ve bunların KNN'ye nasıl dahil edildiğini açıklıyoruz.

A. Normalleştirme ve Ölçeklendirme Kullanarak

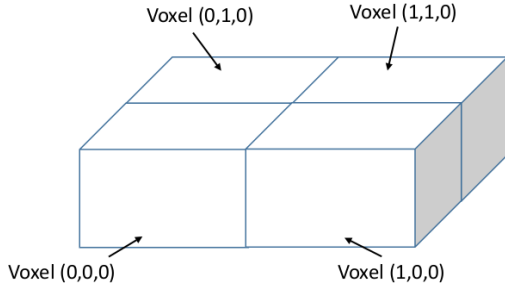
Vokselleştirme

Daha önce de belirtildiği gibi, vokselleştirmenin algılanan verilerin uzamsal-zamansal özelliklerinden ve dolayısıyla bunlar arasındaki uzamsal-zamansal korelasyondan yararlanmanın doğal bir yolu olduğunu görüyoruz. Vokselleştirme basitçe bir hacim nesnesini sabit boyutlu daha küçük hacimlere bölme işlemidir.

Uzamsal-zamansal özelliklere sahip algılanan verilerden oluşan bir koleksiyon (veri kümesi) düşünün. Örneğin, bir veri noktası belirli bir konumda ve belirli bir zaman anında algılanan sıcaklık olabilir. Dolayısıyla, her bir veri noktası en az üç özelliğe, yani enlem, boylam ve zaman özelliklerine sahiptir. Veri kümesinin bir bütün olarak 3 boyutlu uzayda bir hacim kapladığını söyleyebiliriz. 3 boyut bu üç özelliktir. Bu hacim sabit boyutlu daha küçük hacimlere (veya voksellere) bölünerek sabit hacimli kübiklerden oluşan bir 3B yapı elde edilir. Bir vokselin değeri, söz konusu voksele düşen algılanan değerdir. Eğer o vokselde algılanan bir değer yoksa, vokselin değerinin NULL olduğu söylenir.

Vokselleştirme, algılanan iki veri noktası arasındaki *mesafenin* kolayca hesaplanmasına yardımcı olur. Bu çalışmada, iki veri noktası arasındaki mesafe ile iki veri noktası arasındaki korelasyonun bir ölçüsünü kastediyoruz. Örneğin, çevresel algılama bağlamında, üç konumda (lo , lo_{12} , lo_3) aynı zaman anında algılanan sıcaklığı düşünün. Eğer lo , lo_{13} 'e lo_2 'den daha yakınsa, o zaman lo_1 'teki sıcaklık değerinin lo_3 'tekine lo_2 'tekinden daha yakın olması beklenir. Benzer şekilde, bir konumda üç zaman anında (t_1 , t_2 , t_3) algılanan sıcaklık göz önüne alındığında, t_1 'e t_{32} 'den daha yakınsa, t_1 'daki sıcaklık değerinin t_3 'ye t_2 'den daha yakın olması beklenir. Böylece, zamansal uzaklık ve uzamsal uzaklık olmak üzere iki uzaklık ölçütüne sahip oluruz. Ancak, enlem, boylam ve zamanın mutlak değerlerini

kullanamayız. Bunları benzer büyüklükte olacak şekilde ölçeklendirmemiz gerekir. Aksi takdirde, hesaplama sırasında biri diğerlerini gölgede bırakacaktır. Ölçekleme işlemi aşağıda ayrıntılı olarak açıklanmıştır.



Şekil 1. Dört komşu vokselle.

Hedef alan, algılamının yapıldığı alandır. Algılanan verilerin zamansal özelliğini de göz önünde bulunduruyoruz. Bu nedenle, hedef hacim daha uygun bir terim olacaktır. İlk olarak, voksellerin ayrıntı boyutuna karar veririz (yani, bir vokselin ne kadar alanı ve zaman süresini temsil ettiğine). Enlem, boylam ve zaman değerleri için hassasiyet (alınan ondalık basamak sayısı) tanecikliliği belirleyecektir. Daha sonra, değerleri tam sayıya dönüştürmek için gerekli 10'un kuvvetleri ile çarpıyoruz. Zaman için, bundan önce, değerler saniyelere ve ardından bir yuva boyutuna dayalı olarak zaman dilimlerine dönüştürülür. Slot boyutu ayarlanabilen bir parametredir (deneylerimizde 300s olarak aldık). Daha sonra normalize ediyoruz

bu değerlerin aralığı. Örneğin, bir boylam değeri olan l , $l - l_{min}$ değerine normalize edilir. Dolayısıyla, boylam değerleri aralığı $[0, l_{max} - l_{min}]$ olur; burada l_{max} ve l_{min} sırasıyla hedef hacimdeki tüm boylam değerlerinin maksimum ve minimumunu temsil eder. Benzer şekilde, enlem için de bu yapılır

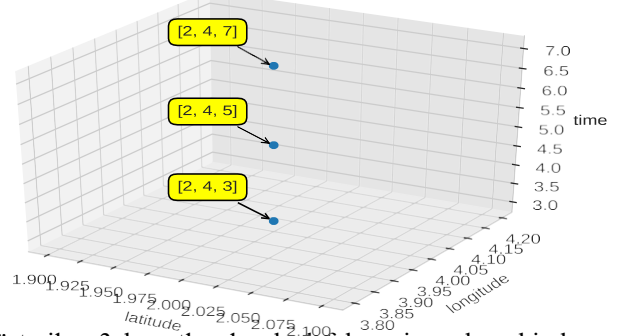
ve zaman. Normalleştirmeden sonra, üç aralıktan daha küçük olan iki aralık en büyük aralığa ölçeklendirilir. Bu, daha sonraki bir bölümde açıklanan mesafe hesaplaması sırasında birinin diğerini gölgede bırakmaması için yapılır. Bundan böyle, ne zaman boylam, enlem veya zaman değerlerinden bahsedilse, bunlar zaten normalleştirilmiş ve ölçeklendirilmiş değerlerdir ve mutlak değerler değildir. Önerilen yaklaşımda mesafe sadece göreceli olarak kullanıldığından mutlak değerlerin kullanılmasının gerekli olmadığı belirtilebilir. Bundan böyle, mesafe ile iki nokta arasındaki göreceli mesafeyi kastediyoruz. Şekil 1'de dört komşu vokselle ve bunların nasıl adlandırıldığı gösterilmektedir.

B. Uzaklık Metriğinde Mekansal-Zamansal Korelasyonun Yakalanması

KNN'de iki veri noktası (algılanan değerler) arasındaki mesafe algoritmasının temelini oluşturur. Bu alt bölümde, mesafe metriğinin geliştirilmesini açıklıyoruz. İlk olarak, sadece uzamsal mesafeyi dikkate almanın iki veri noktası arasındaki uzamsal-zamansal korelasyonu yakalayamadığını gösteriyoruz. Her veri noktasına atanan bir tuple vardır: (enlem, boylam, zaman) ve hedef hacimdeki bir vokseli temsil eder. Bu üç değer son alt bölümde açıklandığı gibi normalize edildiği ve ölçeklendirildiği burada tekrar belirtilebilir (bkz. Bölüm III-A). Veri noktasının değeri, söz konusu vokselde algılanan değeri temsil eder.

Lütfen Şekil 2'ye bakınız. Bu şekil sadece tasvir edilen veri noktalarının 3 boyutlu uzayda birbirlerinden ne kadar uzakta oldukları hakkında bir fikir vermek için kullanılmıştır.

3 datapoints from same location but different times



Gösterilen 3 boyutlu alan hedef hacmin sadece bir kısmıdır. Veri noktası p_1 2 enleminde, 2 boylamındadır.

Şekil 2. Zamana göre belirlenen veri noktaları arasındaki mesafe.

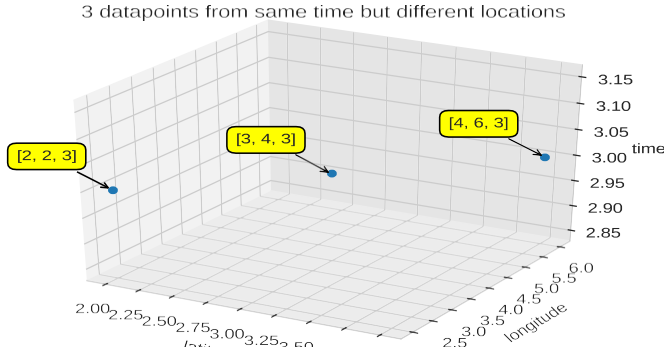
4 ve zaman 7. Benzer şekilde, p_2 ve p_3 veri noktaları sırasıyla (2,4,5) ve (2,4,3) noktalarındadır. $d_S(p_i, p_j)$ iki veri noktası, p_i ve p_j arasındaki uzamsal uzaklığı gösterebilir. Burada, p_i ve p_j 'nin uzamsal 'koordinatlarını' kullanarak Öklid mesafesini dikkate alıyoruz. Burada sadece göreceli mesafenin gerekli olduğu tekrar belirtilmektedir. Dolayısıyla, koordinat değerleri zaten normalleştirilmiş olduğundan Öklid mesafesi yeterince iyi bir ölçüdür. Dünya üzerindeki herhangi iki konum arasındaki gerçek mesafenin hesaplanması Haversine fonksiyonu gibi yöntemlerin kullanılmasını gerektirecektir. O halde, $d_S(p_1, p_2) = (2 - 2)^2 + (4 - 4)^2 = 0$. Benzer şekilde, $d_S(p_1, p_3) = d_S(p_2, p_3) = 0$. Ancak, ile başladığımız için algılanan verilerin uzamsal-zamansal özellikler sergilediği varsayımı

korelasyon, p_1 veri noktasının p 'e p_{23} 'den daha yakın olmasını bekleriz. Başka bir deyişle, p_1 ile ilişkili algılanan değerin p_2 'e p_3 'ten daha yakın olmasını bekleriz. Bu, d_S 'nın veri noktaları arasındaki uzamsal-zamansal korelasyonu etkili bir şekilde yakalamadığını göstermektedir. Benzer şekilde, $d_T(p_i, p_j)$ iki veri noktası, p_i ve p_j arasındaki zamansal mesafeyi gösterebilir. Burada, p_i ve p_j 'nin zamansal 'koordinatları' arasındaki farkın mutlak değerini alıyoruz. Böylece, $d_T(p_1, p_2) = |7 - 5| = 2$; $d_T(p_1, p_3) = |7 - 3| = 4$ ve $d_T(p_2, p_3) = |5 - 3| = 2$. Başka bir deyişle, şekilde görüldüğü gibi, p_1 ile p_2 arasındaki uzamsal-zamansal mesafe (veya

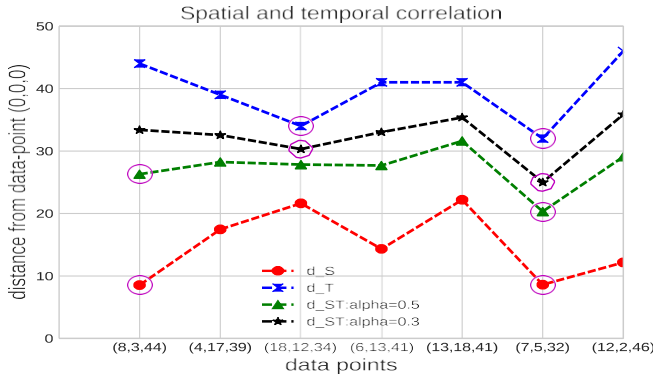
p_2 ve p_3) gerçekten de p_1 ve p_3 veri noktaları arasındakinden daha küçüktür. Bu nedenle, d_T 'nin uzamsal-zamansal korelasyonu yakalayabildiği sonucuna varıyoruz.

İkinci olarak, sadece zamansal farklılığın dikkate alınmasının iki veri noktası arasındaki uzamsal-zamansal korelasyonu yakalayamadığını gösteriyoruz. Şekil 3'te 3 veri noktasının bulunduğu başka bir senaryo gösterilmektedir: p_1 , p_2 ve p_3 sırasıyla (2,2,3), (3,4,3) ve (4,6,3) ile gösterilmiş olup farklı konumlarda ancak aynı zamanda yer almaktadır. Bir kez daha, şekil sadece veri noktalarının 3 boyutlu uzayda birbirlerinden ne kadar uzak olduklarını göstermek için kullanılmıştır. Burada, $d_S(p_1, p_2) = 2.24$; $d_S(p_1, p_3) = 4.47$ ve $d_S(p_2, p_3) = 2.24$. Ancak, $d_T(p_1, p_2) = d_T(p_1, p_3) = d_T(p_2, p_3) = 0$. Bu durumda senaryosunda, d_T uzamsal-zamansal korelasyonu yakalayamaz veri noktaları arasındaki mesafeyi etkin bir şekilde belirlerken d_S belirler. Bu nedenle, mesafe fonksiyonunun tasarımında hem zamansal hem de mekansal mesafelerin dikkate alınması gerektiği sonucuna varıyoruz. Bu nedenle, ikisinin ağırlıklı ortalaması olan yeni bir mesafe metriğini aşağıdaki gibi formüle ediyoruz:

$$d_{ST}(x, y) = \alpha.d_S(x, y) + (1 - \alpha).d_T(x, y) \quad (1)$$



Şekil 3. Konuma göre belirlenen veri noktaları arasındaki mesafe.



Şekil 4. Veri noktasından (0,0,0) göreceli uzaklık.

Burada α , mekansal korelasyon derecesinin bir ölçüsü olan d_S 'ye verilen ağırlıktır. Burada, $\alpha = [0, 1]$. İlk bakışta, d_S ve d_T farklı varlıklar gibi görünse de, Bölüm III-A'da gerçekleştirilen normleştirme ve ölçeklendirme nedeniyle, bir ağırlık faktörü (α) kullanılarak birleştirilebilirler. Uygulamaya bağlı olarak α ayarlanabilir. Örneğin, uygulama zamansal korelasyondan çok mekansal korelasyon içeriyorsa

Algılanan veriler arasında korelasyon varsa, α değeri 0,5'ten büyük bir değere ayarlanabilir. Böyle bir örnek uygulama ortamdaki Oksijen seviyesinin algılanması olabilir. Yakındaki iki konumdaki oksijen seviyesi, uzaktaki iki konumdakinden daha benzer olacaktır (örneğin, biri dağın tepesinde diğeri ovada). Ancak, zamanın bir yerdeki Oksijen seviyesi ile fazla bir ilişkisi yoktur. α değerinin belirlenmesine ilişkin kılavuz bilgiler Bölüm IV-A.1'de verilmiştir. d_S 'nin d_{ST} 'nin özel bir durumu olduğunu unutmayın (yani, $\alpha = 1$ olduğunda, d_{ST} d_S olur). Benzer şekilde,

d_T , $\alpha = 0$ olduğunda d_{ST} 'den başka bir şey değildir.

Eşitlik (1)'in geçerliliği simülasyon sonuçları ile gösterilmiştir

Bölüm IV-A.1'de sunulmuştur. Ek olarak, Eşitlik (1)'in doğruluğunu göstermek için bir grafik sunuyoruz. Lütfen Şekil 4'e bakınız. Eşitlik (1) kullanılarak bir referans veri noktasından (burada (0,0,0)) 7 veri noktasına (x eksen etiketlerine bakınız) olan mesafenin nasıl hesaplandığını göstermektedir. (0,0,0)'dan en yakın iki veri noktası, karşılık gelen grafiklerde her mesafe metriği için daireler kullanılarak gösterilmiştir. 6. veri noktasının, (7,5,32) hem d_S hem de d_T

Bu da Eşitlik (1)'in iki veri noktası arasındaki uzamsal-zamansal korelasyonu doğru bir şekilde ölçtüğünü göstermektedir.

d_{ST} mesafe metriğine dayalı olarak KNN-ST algoritması geliştirilmiştir. Algoritma için k , α ve tip olmak üzere üç giriş parametresi vardır. Burada α , denklem (1)'deki ağırlıktır. Böylece, $\alpha = 1$ ve $\alpha = 0$ olduğunda (bkz. denklem (1)), KNN-ST sırasıyla KNN-S ve KNN-T olur. Parametre $türü$, KNN-ST'nin türünü, yani naif veya ağırlıklı olduğunu belirtir. Naif KNN-ST'de, çıkarılan değer basit bir şekilde

k en yakın komşunun değerleri. Ancak ağırlıklı KNN-ST'de komşuların değerlerinin ağırlıklı ortalaması çıkarılan değerdir. Burada, bir komşu ne kadar yakınsa, ona daha fazla ağırlık verilir. Bir komşunun ağırlığı, çıkarılacak veri noktasına olan uzaklığı ile ters orantılıdır.

mesafe metriği için (0,0,0)'dan en yakın nokta olduğuna dikkat edin. Bu, (0,0,0)'a hem uzamsal hem de zamansal olarak diğer tüm noktalara kıyasla en yakın veri noktası olduğu anlamına gelmektedir. Dolayısıyla, d_{ST} metriğinin hesaplanmasında α 'nın değeri ne olursa olsun, yine de (0,0,0)'dan en yakın nokta olacaktır,

KNN-ST Algoritması

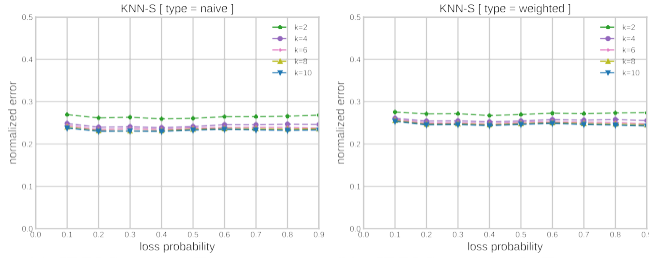
```

//k,  $\alpha$  ve tip giriş parametreleridir.
her eksik veri noktası x için şunları yapın
/ x'in çıkarılan değerini bul,  $x_d$ 
En yakın k noktanın değerlerini S[1..k] dizisine ve ilişkili
 $d_{ST}$  uzaklıklarını D[1..k] dizisine alın.
 $x_d \leftarrow 0$ 
if type = "naive" then
    toplams  $\leftarrow 0$ 
    for i = 1 ila k do
        toplams  $\leftarrow$  toplams + S[i]
    için son
     $x_d \leftarrow$  toplams / k
başka
if type = "weighted" then
    sums  $\leftarrow 0$ ; sumd  $\leftarrow 0$ 
    for i = 1 to k do
        toplamd  $\leftarrow$  toplamd + 1/D[i]
    için son
    for i = 1 ila k do
         $x_d \leftarrow x_d + ((1/D[i])/toplam_D).S[i]$ 
    için son
end if
end if
için son

```

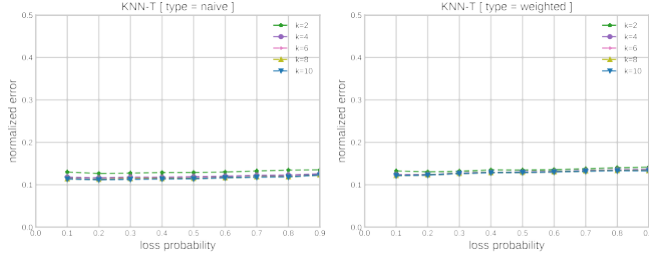
IV. PERFORMANS DEĞERLENDİRMESİ

Bu bölüm, önerilen şemaların ve [17]'de uygulanan mevcut bir teknik olan Sıkıştırılmalı algılamanın sayısal simülasyon sonuçlarını vermektedir. Kullanılan programlama dili ubuntu18 işletim sistemi üzerinde Python'dur (sürüm 3.7.4, anaconda). Algoritmalar, *Sıcaklık* [27] ve *Hava kirliliği* [28] olmak üzere iki veri kümesi üzerinde uygulanmıştır. *Sıcaklık* [27], 4 gün boyunca Roma'daki 289 taksiye sabitlenmiş sensörler tarafından kaydedilen dış ortam sıcaklığının bir veri kümesidir. Biz bir günün verilerini aldık. *Hava kirliliği* [28] veri kümesi, Seul'deki 25 istasyon tarafından üretilen pm2.5, pm10, co ve benzeri hava kirliliği parametrelerinin ölçüm verilerinden oluşur. Veriler bir gün için alınmıştır. Bu bölüm üç kısma ayrılmıştır: i) *Sıcaklık veri kümesi* kullanılarak elde edilen sonuçlar, ii) *Hava kirliliği veri kümesi* kullanılarak elde edilen sonuçlar ve iii) *Sıcaklık dağılımı* ve *hava kirliliği* verileri ve sonuçlar üzerindeki etkisi.



(a) KNN-S: naive; Temperature (b) KNN-S: weighted; Temperature

Şekil 5. Hataya karşı kayıp olasılığı [KNN-S; Sıcaklık]. (a) Naif. (b) Ağırlıklı.



(a) KNN-T: naive; Temperature (b) KNN-T: weighted; Temperature

Şekil 6. Hataya karşı kayıp olasılığı [KNN-T; Sıcaklık]. (a) Naif. (b) Ağırlıklı.

Kullanılan kayıp modeli, her veri noktasının aynı kayıp olasılığı ile silindiği (ve dolayısıyla daha sonra çıkarıldığı) rastgele kayıp modelidir. Kullanılan performans ölçütü, şu şekilde verilen normalize edilmiş hatadır (NE):

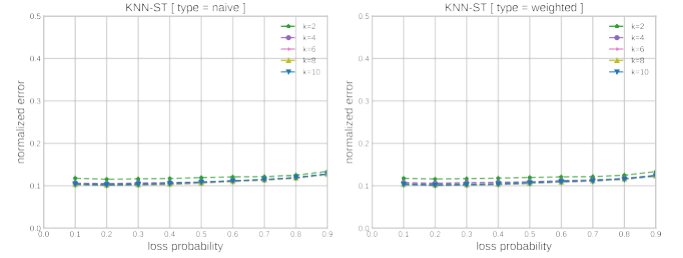
$$NE = \frac{\sum_{i,j,t: S(i,j,t)=0} |S(i,j,t) - \hat{S}(i,j,t)|}{\sum_{i,j,t: S(i,j,t)=0} |S(i,j,t)|} \quad (2)$$

burada, $S(i,j,t)$ vokseldeki (i,j,t) gerçek algılanan değerdir ve $\hat{S}(i,j,t)$, (i,j,t) vokselindeki çıkarılan değerdir. Ne zaman $S(i,j,t) = 0$, (i,j,t) vokselindeki değerin eksik olduğu (ve çıkarılması gerektiği) anlamına gelir.

A. Sıcaklık Veri Seti

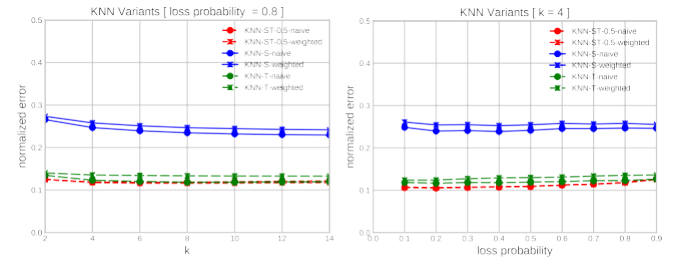
Bu alt bölümde, *Sıcaklık* veri kümesi [27] için sonuçları gösteriyoruz. Şekil 5a ve 5b sırasıyla naif ve ağırlıklı KNN-S için değişen k değerleri için normalize edilmiş hataya karşı kayıp olasılığını göstermektedir. Hata 0,2 ile k 'nın tüm değerleri için 0,3'tür, bu da k değerini artırmanın performansı etkilemediğini göstermektedir. Ayrıca, naif ve ağırlıklılandırılmış KNN-S arasında gözlemlenebilir bir fark yoktur. KNN-T için karşılık gelen grafikler Şekil 6a ve 6b'de bulunmaktadır. KNN-T, hata 0,1 ile 0,15 aralığına düşürüldüğü için KNN-S'den çok daha iyi performans göstermektedir. Ardından, $\alpha = 0,5$ olduğunda KNN-ST için karşılık gelen grafikler Şekil 7a ve 7b'de gösterilmiştir. Görüyoruz ki KNN-ST KNN-T'den marjinal olarak daha iyi performans gösterir ve böylece hem KNN-S hem de KNN-T'den daha iyi performans gösterir.

Şekil 8a şemalar arasında daha iyi bir karşılaştırma sağlar. Kayıp olasılığı 0,8 olduğunda (yani verilerin %80'i kayıp olduğunda) değişen k değerleri için hatayı göstermektedir. KNN-ST'nin ($\alpha = 0,5$ ile) en iyi performansı verdiğini açıkça gözlemleyebiliriz. Şekil 8b, $k = 4$ olduğunda değişen kayıp olasılığı için hatayı göstermektedir.

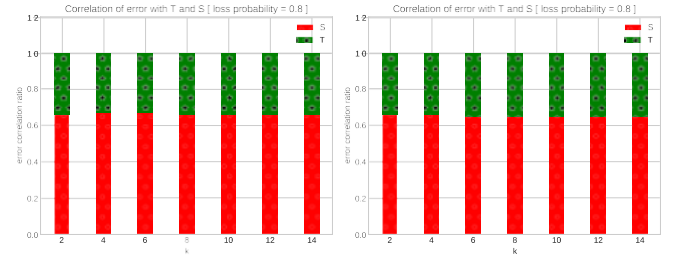


(a) KNN-ST: naive; Temperature (b) KNN-ST: weighted; Temperature

Şekil 7. Hataya karşı kayıp olasılığı [KNN-ST; Sıcaklık]. (a) Naif. (b) Ağırlıklı.



(a) Error vs. k (loss prob.=0.8; Temperature) (b) Error vs. loss probability (k=4; Temperature)

Şekil 8. KNN varyantları [Sıcaklık]. (a) Hata vs. k (kayıp olasılığı = 0,8). (b) Hataya karşı kayıp olasılığı ($k = 4$).

(a) naive; Temperature

(b) weighted; Temperature

Bu senaryoda da KNN-ST daha iyi performans göstermektedir

Şekil 9. Hata korelasyon oranları vs. k.

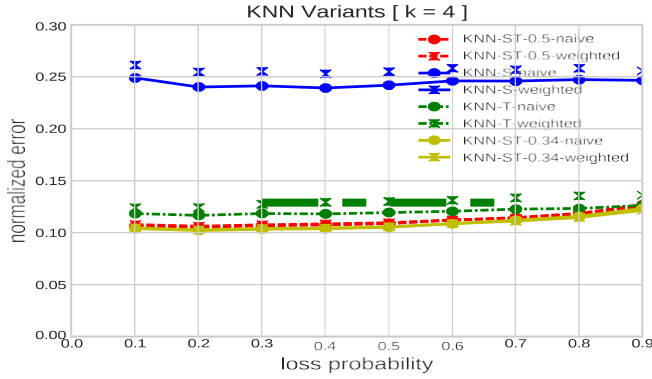
KNN-T ve KNN-S. Kayıp olasılığı 0,9 olduğunda bile elde edilen hatanın KNN-ST için 0,14 içinde olması dikkat çekicidir.

1) α Değerinin Ayarlanması: Yukarıdaki sonuçlardan, α değerinin akıllıca ayarlanması sonucuna varabiliriz (bkz. denklem

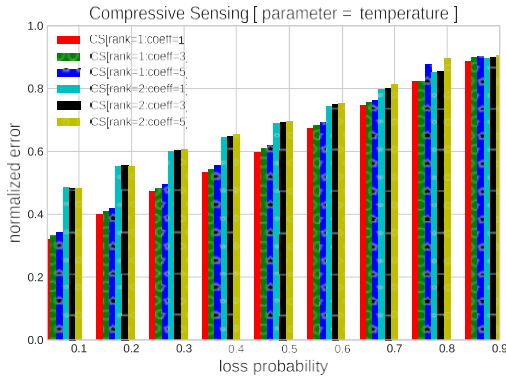
(1)) optimum performans elde etmek için kritik öneme sahiptir. Bir veri kümesi verildiğinde, bunu α 'nın optimum değerini tahmin etmek için kullanabiliriz. Bunun için zamansal (T) ve uzamsal (S) bileşenlerin *hata korelasyon oranı* adı verilen bir metrik hesaplarız. *OranS* ve *oranT* ilgili hataları temsil etsin. Ardından, $oranS = \frac{hata(KNN-S)}{hata(KNN-S) + hata(KNN-T)}$ ve $oranT = \frac{hata(KNN-T)}{hata(KNN-S) + hata(KNN-T)}$ gelen hatayı temsil eder

sırasıyla KNN-S ve KNN-T uygulandığında. Benzer şekilde, $oranT = \frac{hata(KNN-T)}{hata(KNN-S) + hata(KNN-T)}$. Kayıp olasılığı 0,8 iken k 'nın değişen değerleri için hata oranları *naif* ve *ağırlıklı* için sırasıyla Şekil 9a ve 9b'de gösterilmiştir. Hata oranı ne kadar yüksekse, korelasyonun o kadar az olduğu sonucuna varıyoruz. Böylece, zamansal korelasyonun uzamsal korelasyondan daha yüksek olduğunu gözlemliyoruz. Çünkü, hata oranı

korelasyon ölçüsü ile dolaylı olarak orantılı olduğundan, bunu α değerini ayarlamak için kullanabiliriz. α mekansal bileşene atanan ağırlık olduğu için $\alpha = 1 - oranS$ olarak ayarlanmıştır. Yani, temel olarak



Şekil 10. Hataya karşı kayıp olasılığı [KNN varyantları; $k=4$; Sıcaklık].



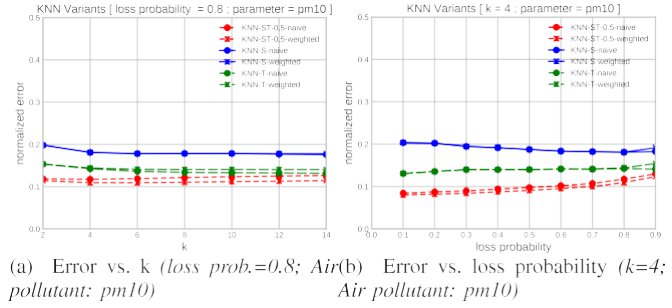
Şekil 11. Hataya karşı kayıp olasılığı [Sıkıştırılmalı algılama (Sıcaklık)].

Şekil 9a ve 9b'de, $k = 4$ olduğunda $\alpha = 0,34$ olarak ayarlanmıştır (k 'nın diğer değerleri için de kayda değer bir değişiklik yoktur). Konuyu açıklamak için $\alpha = 0,34$ olduğunda elde edilen sonuçları diğerleriyle karşılaştırıyoruz (bkz. Şekil 10). Şekilde, kayıp olasılığı 0.8 olduğunda $\alpha = 0,34$ 'ün en iyi performansı verdiğini görüyoruz.

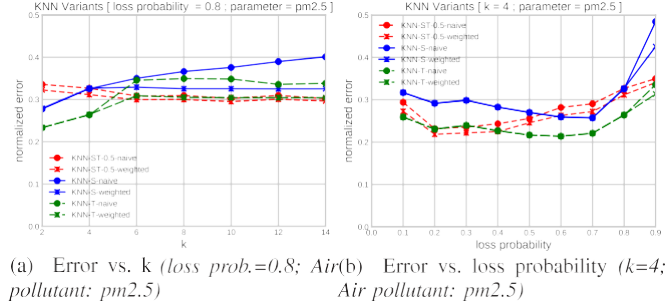
Daha sonra, [17]'de uygulandığı şekliyle CS yönteminin performansına bakacağız. Şekil 11, değişen kayıp olasılığı için normalleştirilmiş hatayı göstermektedir. CS için değişen en küçük kareler algoritmasında kullanılan iterasyon sayısı 100'dür. Değiş tokuş katsayısı grafikte katsayı ile gösterilmiştir. Farklı rank ve katsayı değerleri ile yapılan denemeler sonucunda sadece en iyi sonuçlar gösterilmiştir. CS yönteminin sonuçlarını KNN-ST ile karşılaştırdığımızda (bkz. Şekil 10), KNN-ST'nin CS yönteminden daha iyi performans gösterdiğini görüyoruz. Bunun bir nedeni, Şekil 18a'da görüldüğü gibi Sıcaklık veri kümesindeki veri dağılımının oldukça seyrek olmasıdır. Bu da Sıkıştırarak algılama yöntemini uygulamak için oluşturulan ölçüm matrisinin seyrek olmasıyla sonuçlanır. Dolayısıyla, KNN-ST'ye kıyasla daha zayıf bir sonuç elde edilmiştir. KNN-ST'nin (bkz. Şekil 10) CS yönteminden (bkz. Şekil 11) daha sağlam olması dikkat çekicidir, çünkü kayıp olasılığı arttıkça CS yönteminin performansı düşerken KNN-ST için böyle bir durum söz konusu değildir.

B. Hava Kirliliği Veri Seti

Bu alt bölümde, hava kirliliği veri kümesi [28] için sonuçlar gösterilmektedir. Yer kısıtlamaları nedeniyle



Şekil 12. KNN varyantları KNN varyantları [Hava kirliliği: pm10]. (a) Hava vs. k (kayıp olasılığı = 0,8). (b) Kayıp olasılığına karşı hata ($k = 4$).



sadece birkaç sonuç gösterilmiştir. Şekil 12a, pm10 ölçümü için kayıp olasılığı 0,8 olduğunda (yani, verilerin %80'i eksik olduğunda) k 'nın değişen değerleri için hatayı göstermektedir. KNN-ST'nin ($\alpha = 0,5$ ile) en iyi performansı (en az hata) verdiğini gözlemliyoruz. Şekil 12b şunları göstermektedir

Şekil 13. KNN varyantları KNN varyantları [Hava kirliliği: $pm2.5$]. (a) Hata vs. k (kayıp olasılığı = 0,8). (b) Kayıp olasılığına karşı hata ($k = 4$).

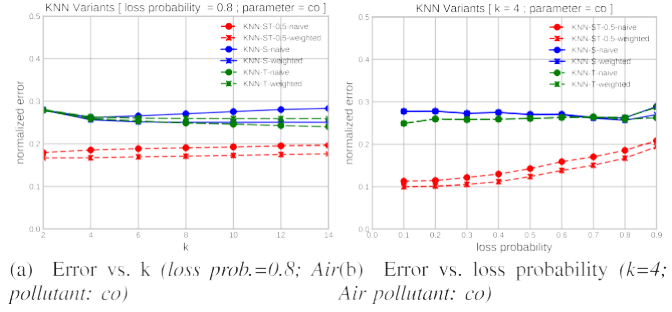
Şekil 14. KNN varyantları KNN varyantları [Hava kirliliği: co]. (a) Hata vs. k (kayıp olasılığı = 0,8). (b) Hataya karşı kayıp olasılığı ($k = 4$).

$k = 4$ olduğunda değişen kayıp olasılığı için hata. Bu durumda da KNN-ST, KNN-T ve KNN-S'den daha iyi performans göstermektedir.

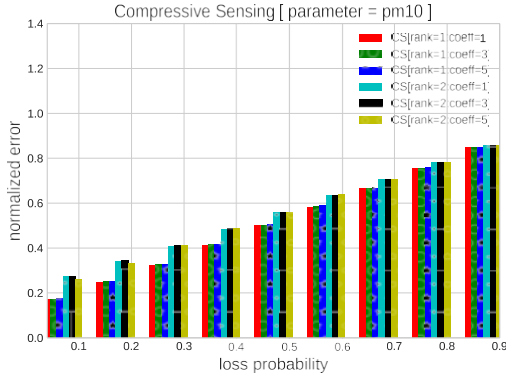
Şekil 13a ve 13b $pm2.5$ ölçümü için ilgili grafikleri vermektedir. Grafiklerden, $pm2.5$ ölçümü için uzamsal-zamansal korelasyonun zayıf olduğu sonucuna varabiliriz. Bunun nedeni elde edilen hataların çok yüksek olması ve grafiklerin gelişigüzel olmasıdır. co kirleticisi için ilgili grafikler Şekil 14a ve 14b'de gösterilmektedir. Her iki grafikten de KNN-ST'nin ($\alpha = 0,5$ ile) KNN-S ve KNN-T'den daha iyi performans gösterdiğini gözlemliyoruz. α değerinin makul bir şekilde ayarlanması için aşağıdaki yönergeleri takip edebiliriz

Bölüm IV-A.1'de

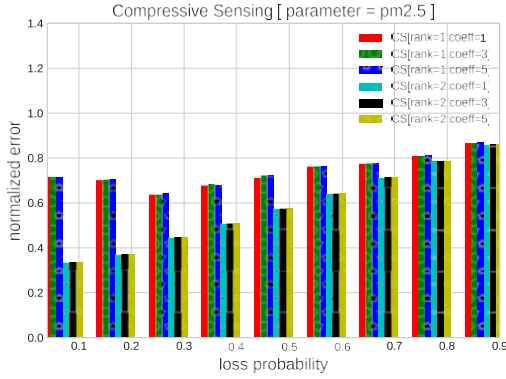
Hava kirliliği veri kümesi kullanılarak elde edilen CS yöntemi sonuçları Şekil 15 ile 17'de gösterilmektedir. KNN-ST'nin CS yönteminden daha iyi performans gösterdiğini gözlemliyoruz (Şekil 12b ile Şekil 15'i, Şekil 13b ile Şekil 16'yı ve Şekil 14b ile Şekil 17'yi karşılaştırın). KNN-ST'de, $pm10$ parametresi durumunda performans diğer ikisinden daha iyidir, yani $pm2.5$ ve co , $pm10$ durumunda nispeten daha iyi bir uzamsal-zamansal korelasyon göstermektedir. Aynı davranış CS yönteminin sonuçlarında da görülmektedir. Bu da hem KNN-ST hem de CS yönteminin algılanan veriler arasındaki korelasyonu yakaladığını doğrulamaktadır.



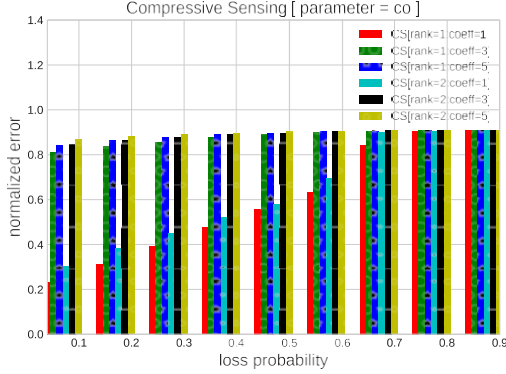
(a) Error vs. k (loss prob.=0.8; Air pollutant: co) (b) Error vs. loss probability ($k=4$; Air pollutant: co)



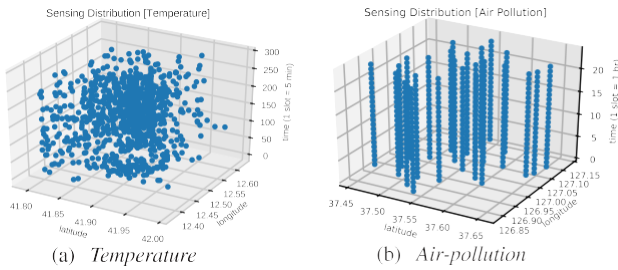
Şekil 15. Hataya karşı kayıp olasılığı [Sıkıştırılmış algılama (Hava Kirliliği: pm10)].



Şekil 16. Hataya karşı kayıp olasılığı [Sıkıştırılmış algılama (Hava Kirliliği: pm2.5)].



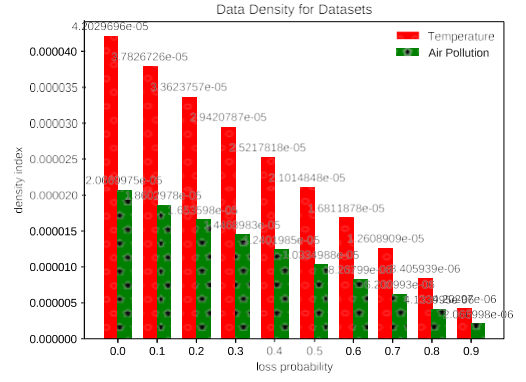
Şekil 17. Hataya karşı kayıp olasılığı [Sıkıştırılmış algılama (Hava Kirliliği: co)].



Şekil 18. Algılanan veri dağılımı. Algılanan veri dağılımı.

C. Algılanan Veri Dağıtımı

Sıcaklık ve Hava kirliliği veri kümeleri için algılanan değerlerin uzamsal-zamansal dağılımı sırasıyla Şekil 18a ve 18b'de gösterilmektedir. Açıkça gözlemleyebiliriz ki



Şekil 19. Yoğunluk indeksine karşı kayıp olasılığı.

dağılımı Şekil 18a'da Şekil 18b'ye göre daha eşittir. Buna göre, Sıcaklık için Hava kirliliğinden daha iyi sonuçlar gözlemlenmektedir.

Son olarak, algılanan verilerin seyrekliğini tahmin etmek için bir metrik olan yoğunluk endeksini kullanıyoruz. Yoğunluk endeksi, algılanan değerlere sahip voksel sayısının 3B hedef hacmindeki toplam voksel sayısına oranıdır. Şekil 19, kayıp olasılığının değişen değerleri için yoğunluk endeksini göstermektedir. Yoğunluk endeksinin çok düşük olduğunu açıkça görüyoruz. Yine de KNN-ST'nin buna rağmen iyi çalıştığını görüyoruz. Ayrıca, Sıcaklık veri kümesinin yoğunluk dağılımı Hava kirliliği veri kümesinden daha yoğun olduğundan, veri çıkarım sonuçları Sıcaklık veri kümesi için nispeten daha iyidir.

V. SONUÇ

Bu çalışma, uzamsal-zamansal bir K En Yakın Komşu şemasının geliştirilmesini sunmaktadır. İlk olarak, mekansal (KNN-S) ve zamansal (KNN-T) korelasyonun KNN'ye bağımsız olarak dahil edilmesinin nasıl iyi çalışmadığını gösteriyoruz. Bu gözlemden yola çıkarak, KNN-S ve KNN-T'nin ağırlıklı bir kombinasyonu olan KNN-ST ile sonuçlanan hem mekansal hem de zamansal korelasyonu dahil ediyoruz. Böylece, KNN-ST'deki zamansal ve mekansal korelasyon bileşenlerinin derecesi, uygulamanın gereksinimine göre ağırlıklandırma parametresi (α) ayarlanarak değiştirilebilir. Ayrıca aynı parametrenin (α) nasıl ayarlanacağına dair yönergeler de sunmaktayız. İki gerçek hayat veri kümesi üzerinde uygulandıktan sonra elde edilen sayısal simülasyon sonuçları, önerilen şemanın geçerliliğini göstermektedir. Ayrıca, başka bir tümdengelim yöntemi olan Sıkıştırma algılama ile karşılaştırma da sunulmuştur. İlginç bir şekilde, KNN-ST'nin kayıp olasılığı 0,9 kadar yüksek olduğunda bile gerçekten iyi performans (yaklaşık 0,1'lik normleştirilmiş mutlak hata) gösterdiğini görüyoruz. Böylece, K-En Yakın Komşular yönteminin veri seyrek olduğunda genellikle kötü performans gösterdiği bilinse de, önerilen KNN-ST şemasının böyle bir senaryo altında bile uzamsal-zamansal özelliklerden yararlanıldığında yeterince iyi performans gösterdiğini doğruluyoruz.

REFERANSLA

R

- [1] R. Ganti, F. Ye, ve H. Lei, "Mobile crowdsensing: Mevcut durum ve gelecekteki zorluklar," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32-39,

Nov. 2011.

- [2] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han ve A. M'hamed, "Sparse mobile crowdsensing: Zorluklar ve fırsatlar," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 161-167, Jul. 2016.

- [3] T. Cover ve P. Hart, "En yakın komşu örüntü sınıflandırması," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21-27, Jan. 1967.
- [4] L. Kong, D. Jiang, ve M.-Y. Wu, "Ortam soyutlaması için siber-fiziksel sistemlerin uzamsal-zamansal dağılımını optimize etme," *Proc. IEEE 30th Int. Konf. Dağıtım. Comput. Syst. içinde*, Haziran 2010, s. 179-188.
- [5] H. Zhu, Y. Zhu, M. Li ve L. M. Ni, "SEER: Kayıplı duyuşal verilere dayalı büyükşehir ölçeğinde trafik algısı," *Proc. IEEE INFOCOM-28th Conf. Comput. Commun.*, Nisan 2009, s. 217-225.
- [6] R. Baraniuk, "Compressing sensing," *IEEE Signal Process. Mag.*, vol. 24, no. 4, s. 118-121, Temmuz 2007.
- [7] E. J. Candes, J. Romberg, ve T. Tao, "Sağlam belirsizlik ilkeleri: Yüksek oranda eksik frekans bilgisinden kesin sinyal yeniden yapılındırması," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489-509, Feb. 2006.
- [8] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289-1306, Nisan 2006.
- [9] L. Kong ve diğerleri, "Data loss and reconstruction in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, cilt 25, no. 11, pp. 2818-2828, Kasım 2014, doi: [10.1109/TPDS.2013.269](https://doi.org/10.1109/TPDS.2013.269).
- [10] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and Internet traffic matrices," in *Proc. ACM SIGCOMM Konf. Data Commun. (SIGCOMM)*, 2009, s. 267-278.
- [11] L. Wang ve diğerleri, "CCS-TA: Sıkıştırılmış kalabalık algılamada kalite garantili çevrimiçi görev tahsisi," içinde *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Osaka, Japonya, 2015, s. 683-694.
- [12] E. J. Candès ve B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717-772, Dec. 2009.
- [13] X. Liu, X. Wang, L. Zou, J. Xia ve W. Pang, "Spatial imputation for air pollutants data sets via low rank matrix completion algorithm," *Environ. Int.*, vol. 139, Jun. 2020, Art. no. 105713.
- [14] Z. Wang, M.-J. Lai, Z. Lu, W. Fan, H. Davulcu, and J. Ye, "Rank-one matrix pursuit for matrix completion," in *Proc. Int. Konf. Mach. Learn.*, Pekin, Çin, 2014, s. 91-99.
- [15] J. Y. Gotoh, A. Takeda, and K. Tono, "DC formulations and algorithms for sparse optimization problems," *Math. Program.*, cilt 169, pp. 141-176, Mayıs 2018.
- [16] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda ve M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Temmuz 2017, s. 5115-5124.
- [17] Y. Zhu, Z. Li, H. Zhu, M. Li, ve Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2289-2302, Nov. 2013.
- [18] A. M. Durán-Rosal, C. Hervás-Martínez, and A. J. Tallón-Ballesteros, "Massive missing data reconstruction in ocean buoys with evolutionary product unit neural networks," *Ocean Eng.*, vol. 117, pp. 292-301, May 2016.
- [19] S. Tak, S. Woo ve H. Yeo, "Yol bağlantılarının kesit birimlerindeki trafik verileri için veri güdümlü imputasyon yöntemi," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1762-1771, Jun. 2016.
- [20] F. Tonini, W. W. Dillon, and E. S. Money, "Spatio-temporal reconstruction of missing forest microclimate measurements," *Agric. Forest Meteorol.*, vols. 218-219, pp. 1-10, Mar. 2016.
- [21] S. Londhe, P. Dixit, S. Shah, and S. Narkhede, "Yapay sinir ağı kullanarak eksik günlük yağış kayıtlarının doldurulması," *ISH J. Hydraulic Eng.*, vol. 21, no. 3, pp. 255-264, Eylül 2015.
- [22] J. Tipton, M. Hooten, and S. Goring, "Reconstruction of spatio-temporal temperature from sparse historical records using robust probabilistic principal component regression," *Adv. Stat. Climatol., Meteorol. Oceanogr.*, vol. 3, no. 1, pp. 1-16, 2017.
- [23] W. Ruan, P. Xu ve Q. Z. Sheng, "Recovering missing values from corrupted spatio-temporal sensory data via robust low-rank tensor completion," in *Proc. Conf. Veritabanı Sist. Adv. Appl. içinde*, Cham, İsviçre: Springer, 2017, s. 607-622.
- [24] S. Cheng, F. Lu, P. Peng, and S. Wu, "A spatiotemporal multi-view-based learning method for short-term traffic forecasting," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 6, p. 218, 2018, doi: [10.3390/ijgi7060218](https://doi.org/10.3390/ijgi7060218).
- [25] X. Luo, D. Li, Y. Yang ve S. Zhang, "KNN ve LSTM ile spatiotemporal trafik akışı tahmini," *J. Adv. Transp.*, vol. 2019, pp. 1-10, Feb. 2019, doi: [10.1155/2019/4145353](https://doi.org/10.1155/2019/4145353).
- [26] S. Griffioen, "A voxel-based methodology to detect (clustered) outliers in aerial LIDAR point clouds," M.S. thesis, Dept. Urbanism, Dept. Archit. Built Environ, Delft Univ. Technol., Delft, Hollanda, Kasım 2018.
- [27] M. A. Alswailim, H. S. Hassanein ve M. Zulkernine. (Kasım 2015). *CROWDAD Veri Seti Queensu/Kalabalık Sıcaklığı (V. 20151120): Roma/Taksi'den Türetilmiştir (V. 20140717)*. [Çevrimiçi]. Mevcut: https://crowdad.org/queensu/crowd_temperature/20151120, doi: [10.15783/C7CG65](https://doi.org/10.15783/C7CG65).
- [28] (27 Nisan 2020). *Air Pollution in Seoul*. Erişim tarihi: Nisan 3, 2020. [Çevrimiçi]. Mevcut: <https://www.kaggle.com/bappekim/air-pollution-in-seoul>



Ningrinla Marchang (Kıdemli Üye, IEEE) 1993 yılında Itanagar, Hindistan'daki Kuzey Doğu Bölgesel Bilim ve Teknoloji Enstitüsü'nden bilgisayar bilimi ve mühendisliği alanında lisans derecesi, 1995 yılında Delhi'deki Hindistan Teknoloji Enstitüsü'nden (IIT Delhi) bilgisayar bilimi ve mühendisliği alanında yüksek lisans derecesi ve 2010 yılında NERIST'ten bilgisayar bilimi ve mühendisliği alanında doktora derecesi almıştır. 1995-1996 yılları arasında Bilgisayar Bilimleri Bölümü'nde Araştırma Mühendisi olarak görev yapmıştır.

ve Mühendislik, IIT Delhi. 1996-2001 yılları arasında Hindistan'ın Chennai kentindeki Sathyabama Engineering Col- lege'de Bilgisayar Uygulamaları Bölümü'nde ders vermiştir. 2001 yılından bu yana NERIST'te öğretim üyesidir ve Bilgisayar Bilimi ve Mühendisliği Bölümü'nde Doçent olarak görev yapmaktadır. Araştırma alanları arasında mobil ad hoc ağlar, bilişsel radyo ağları ve mobil crowdsensing yer almaktadır.



Rakesh Tripathi (Üye, IEEE) 2018 yılında Hindistan Teknoloji Enstitüsü Guwahati'den doktora derecesini almıştır. NIT Raipur Bilgi Teknolojileri Bölümü'nde Yardımcı Doçent olarak görev yapmaktadır. Araştırma alanları arasında blok zinciri, veri merkezi ağları ve izinsiz giriş tespiti yer almaktadır.