



Hücrel Otomata Tabanlı İkili Sınıflandırma

Araştırma Makalesi/Research Article

 Serkan PELDEK*¹,  Yaşar BECERİKLİ²

¹Bilgisayar Mühendisliği, Kastamonu Üniversitesi, Kastamonu, Türkiye

²Bilgisayar Mühendisliği, Kocaeli Üniversitesi, Kocaeli, Türkiye

*speldek@kastamonu.edu.tr, ybecerikli@kocaeli.edu.tr

(Geliş/Received:22.02.2018; Kabul/Accepted:21.12.2018)

DOI: 10.17671/gazibtd.397452

Özet— Bu çalışmada lineer olmayan hücrel otomatlar ikili örüntü sınıflandırmada kullanılmıştır. Hücrel otomatlar ilk olarak günümüz bilgisayar mimarisinin çalışma prensiplerini belirleyen Von Neumann tarafından biyolojik canlıların kendini yenileme yeteneklerini modellemek için önerilmiştir. Hücrel otomatlar, ızgara düzlemindeki hücrelerin etrafındaki hücrelerle etkileşime bağlı durum güncelleme mantığına dayanan hesaplama modeline sahiptir. Hücrel otomatlar üzerine yapılan çalışmalarda bazı durumların diğer durumlarla dinamik bir etkileşim içerisinde olduğu gözlemlendi. Bu durumlar diğer durumları etrafında toplayan ve çekim merkezi formunda hareket eden davranışlar sergiliyorlardı. Çekim merkezi formunda hareket eden durumlara cezbedici durum(veya cezbedici kap) adı verildi. Cezbedicilerin dinamik davranışları, bir örüntünün diğer örüntüleri çekmesi olarak ele alınması hücrel otomatların örüntü tanıma ve sınıflandırmada kullanma potansiyelini ortaya çıkarmıştır. Hücrel otomata tabanlı ilk örüntü tanıma yöntemleri XOR ve XNOR mantığına göre durum güncellemesi yapan kurallar kullanan lineer yöntemlerdir. Daha sonra lineer yöntemlerin kısıtlarını aşmak için lineer olmayan yöntemler geliştirilmiştir. Bu çalışmada cezbedici durumların nitelenmesi için erişilebilirlik ağacı tabanlı lineer olmayan yöntemler kullanılmıştır. Cezbedici durumlar farklı veri setleri üzerinde ikili sınıflandırma amacı ile kullanılmıştır. Elde edilen sonuçlar daha önce yapılan hücrel otomata tabanlı örüntü tanıma yöntemleri ve diğer yöntemlerle karşılaştırılmıştır.

Anahtar Kelimeler— hücrel otomata, çoklu cezbedici hücrel otomatlar, ikili sınıflandırma

Cellular Automata Based Binary Classification

Abstract— In this study, nonlinear cellular automata were used for binary pattern classification. Cellular automata were first proposed by Von Neumann, who determined the working principles of today's computer architecture, to model the self-renewal abilities of biological beings. Cellular automata have a computational model based on the state update logic due to the interaction with the cells around the cells in the grid plane. Studies on cellular automata have shown that some states are in a dynamic interaction with other states. These states gathered other states around itself and acted in the centre of attraction. States that behave in the form of attraction centre are called attractor state(or attractor basin). The dynamic behaviours of the attractor were considered as a pattern of attracting other patterns and revealed the potential of using cellular automata in pattern recognition and classification. The first pattern recognition methods based on cellular automata are linear methods that use rules that update the state according to XOR and XNOR logic. Later nonlinear methods have been developed to overcome the limitations of linear methods. In this study, reachability tree based nonlinear methods are used to characterize the attractors. Attractor states are used for binary classification purposes on different data sets. The results obtained were compared with previous cellular automata based pattern recognition methods and other known methods.

Keywords— cellular automata, multiple attractor cellular automata, binary classification

1. GİRİŞ (INTRODUCTION)

Bilgisayar bilimlerinin temel hedefi bilgisayarların hesaplama yeteneklerine uygun modeller tasarlamaktır [1].

Hücrel otomatlar ilk olarak günümüz bilgisayarlarının çalışma prensiplerini belirleyen John Von Neumann tarafından önerilmiştir [2]. Neumann bilgisayar mimarisinin hesaplama yeteneklerine uygun modeller tasarlayabilecek bilgiye ve tecrübeye sahiptir. Neumann

bilgisayar sistemlerinin ayrık ve ikilik sayı sistemi kullanarak çalışması gerektiğini belirtmiş ve önerdiği prensiplerini [3] bu doğrultuda yapmıştır.

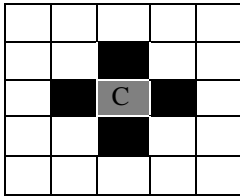
Neumann biyolojik organizmaların kendini üretebilme yeteneklerine sahip bir makine tasarlamak istiyordu. Bu amaçla biyolojik canlıların kendini yenileme yeteneklerini modellemek için hücresel otomata adını verdiği model geliştirmiştir. Neumann yaptığı çalışmaları yayınlama fırsatı bulamadı, bu çalışmalar ölümünden sonra Stan Ulam tarafından yayınlanmıştır [2].

Neumann biyolojik canlıların kendini yenileme yeteneklerinin mantıksal bir çerçeve ile modellenmesi ile ilgileniyordu. Langton'un belirttiği gibi kendini yenilme fenomenini genetik ve biyokimya bağlamında değil, mantıksal bir çerçeve ile modellemeye çalışıyordu [4].

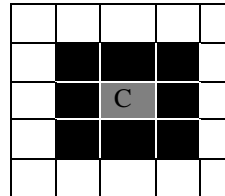
Hücresel otomatlar, günümüzde ilk önerildiğinden çok daha basit bir yapıya sahiptir. Hücresel otomatlar ilk önerildiğinde her bir hücre için 29 durum ve etkileşimde olduğu 5 komşu vardı. Durum sayısı daha sonraki yapılan çalışmalarda 8'e [5] ve 2'ye [6] düşürülmüştür. Wolfram ilk önerilen hücresel otomatanın karmaşık olmasını o zamanki bilimsel sezgilere dayandırmaktadır. Neumann'ın biyolojik canlılar gibi karmaşık bir yapının ancak karmaşık bir mantıksal yapı ile modellenebileceği fikrinde olduğunu belirtir [6].

Hücresel otomatlar 1970'lere kadar az sayıda bilim insanının dikkatini çekmiştir. Hücresel otomatların çalışma mantığının anlaşılması ve bilim insanlarının dikkatini çekmesi Conway'in tasarladığı "Hayat Oyunu" ile olduğu birçok araştırmacı [7] [6] [1] tarafından vurgulanmıştır.

Hayat Oyunu, Neumann tarafından önerilen her bir hücrenin 5 komşuya sahip olmasından farklı Moore [8] tarafından geliştirilen 9 komşulu bir yapıya sahipti. Moore komşuluğuna göre ızgara düzlemindeki her bir hücre kendi ve etrafındaki sekiz hücreye göre bir sonraki durumunu belirler. Aşağıda Neumann ve Moore komşulukları gösterilmektedir.



a) Neumann komşuluğu
(Neumann Neighborhood)



b) Moore komşuluğu
(Moore Neighborhood)

Şekil 1 Neumann ve Moore komşulukları.
(Neumann and Moore Neighborhood)

Hayat Oyunu, Conway tarafından geliştirilmesine rağmen bilim çevreleri tarafından bilinmesi Gardner tarafından olmuştur [9] [10]. Gardner Hayat Oyununu bir simülasyon oyunu olduğunu belirtir. Hayat Oyunu canlı organizma topluluklarının çoğalması, azalması ve dönüşüm geçirmesi

gibi benzerliklere sahip olduğu için hayatın bir tür benzetimi olarak düşünülmüştür [9].

Hayat Oyunu, hücresel otomatların kapasitesi ve mantığının anlaşılması açısından oldukça önemli bir etkiye sahiptir. O zamana kadar anlaşılması zor kavramlar olarak kalan hücre, durum geçişi, etkileşim gibi hücresel otomatların temel mantığına ait tanımlar daha anlaşılır hale gelmiştir.

Hücresel otomatların gelişimindeki en önemli çalışmaların Stephen Wolfram tarafından yapıldığı söylenebilir. Wolfram 1980'lerin başlarında hücresel otomata üzerine çalışmaya başlamıştır [11]. Wolfram Hayat Oyununun ortaya koyduğu gerçeklerin önemini bilim camiası tarafından yeteri kadar anlaşılmadığını belirtmiştir [6].

Wolfram'ın hücresel otomata üzerine yaptığı çalışmalar hücresel otomatlar için yeni bir evre olarak düşünülebilir ve modern [7] hücresel otomata çalışmalarının başlangıcı kabul edilir. Wolfram hücresel otomata ile basit kuralların karmaşık sistemleri modelleyebileceğini düşünmekteydi. Üzerinde çalıştığı hücresel otomata tek boyutlu ve iki durumdan oluşuyordu. Neumann ve Moore komşuluklarına sahip hücresel otomatlar iki boyutlu bir yapıya sahipti. Bu hücresel otomata türüne göre bir hücre kendi, sol ve sağ komşuları ile etkileşimine göre bir sonraki durumunu belirler. Aşağıda Wolfram'ın yoğunlukla üzerine çalıştığı bir boyutlu hücresel otomata türe gösterilmektedir.



Şekil 2 Üç komşulu hücresel otomata.
(Cellular Automata with Tree Neighborhood)

Wolfram'ın önerdiği hücresel otomata yapısı farklı disiplinlerdeki araştırmacıların büyük ilgisini çekti. Bu sayede hücresel otomata mühendislik ve sosyal bilimler gibi farklı alanlarda daha fazla kullanılmaya başlamıştır. Hücresel otomatların hesaplama yeteneği başta fizik [12] ve matematik [13] olmak üzere, coğrafya [14], biyoloji [15], sosyal bilimler [16], görüntü işleme [17] vb. alanlarda kullanılmıştır.

Durum uzayında bazı durumların diğer durumları etrafında toplaması ve çekim merkezi gibi davranmasının fark edilmesi, hücresel otomatların örüntü tanıma potansiyelini ortaya çıkarmıştır. Çekim merkezi gibi davranan bu durumlar 'cezbedici' olarak adlandırılmıştır.

Hücresel otomatların örüntü tanımadada kullanılması için gerekli olan kurallarının nitelenmesi için birçok çalışma yapılmıştır. Hücresel otomata kurallarının nitelenmesi üzerine çalışmalar lineer/toplamsal [18] ve lineer olmayan [19] [20] yöntemler olmak üzere ikiye ayrılır. Lineer yöntemler kural nitelemesi için cebirsel işlemler kullanırken, lineer olmayan yöntemler genetik algoritmalar ve erişilebilirlik ağacı temelli çözümler geliştirmişlerdir.

Bu çalışmada cezbedici durumlar ikili sınıflandırmada kullanılmıştır. Cezbedici durumların nitelenmesi için erişilebilirlik ağacı tabanlı algoritmaların sınıflandırma performansı, iki farklı veri seti üzerinde test edilmiştir.

İkinci bölümde hücrel otomata modeli kullanılarak yapılan benzer çalışmalar yer almaktadır. Üçüncü bölümde kullanılan hücrel otomata yönteminin teorik alt yapısı açıklanmıştır. Dördüncü bölümde hücrel otomata tabanlı ikili sınıflandırma iki farklı veri seti üzerinde kullanılmış ve elde edilen sonuçlar önde gelen makine öğrenmesi yöntemleri ile ve literatürdeki çalışmalarla karşılaştırılmıştır. Beşinci bölümde çalışmanın ve elde edilen sonuçların değerlendirilmesi yapılmıştır.

2. İLGİLİ ÇALIŞMALAR (RELATED WORKS)

Bu bölümde literatürde hücrel otomata ile yapılan bilimsel çalışmalara yer verilmiştir. Literatür değerlendirmesinin kapsamı bilgisayar bilimleri ve özde örüntü tanıma ile sınırlandırılmıştır. Bu amaçla literatür değerlendirmesi iki ana başlığa ayrılmıştır. Birinci bölümde hücrel otomata ile yapılan bilgisayar biliminin kapsamına giren çalışmalara yer verilmiştir. İkinci bölümde hücrel otomata durum uzayında çekim formunda davranan cezbedici durumların örüntü tanımada kullanılması üzerine yapılan çalışmalara yer verilmiştir.

2.1. Yöntemle İlgili Çalışmalar (Method Related Works)

Hücrel otomatların yerel etkileşime dayanan basit yapısının kompleks global davranışlar sergilemesi araştırmacıların dikkatini çekmesinin temel nedenidir [21]. Hücrel otomatlar ilk olarak biyolojik ve fiziksel sistemlerin modellenmesinde kullanıldı. 1980'lerden sonra gerçek hayat problemleri üzerinde kullanılmaya başlamıştır [22].

Hücrel otomata elektronik devre tasarımı ve VLSI gibi büyük devrelerin test ve hata bulmasında yoğunlukla çalışılan bir konudur. Hücrel otomatın eş zamanlı programlamaya imkan veren hesaplama modeli kompleks sistemlerin benzetimi [23], sıralama makineleri ve asal sayı üreteçleri [24] gibi hesaplama maliyeti yüksek işlemler için elektronik devre tasarımında kullanılmıştır.

VLSI devrelerinin test edilmesi için rasgele örüntü üreteçleri tasarlamak için bir çok çalışma yapılmıştır. Bu alanda öne çıkan çalışmalar Sukanta Das tarafından yapılmıştır [25] [26]. Das, UBIST adını verdiği rasgele örüntü üreticinin dört farklı örüntü üretme özelliği olduğunu ortaya koymuştur.

Son durumlu hücrel otomatlar biçimsel dillerin kabul edilmesinde kullanılmıştır. Buradaki temel fikir başlangıç ve son durum ayarları yapılan bir otomata bir dizge verilir ve dizgenin karakterleri için uygulanan iterasyonlar sonucunda son duruma varılırsa kabul edilir, varılmazsa ret edilir [7]. Çalıkoğlu, biçimsel dillerin gerçek zamandan daha kısa bir sürede kabul edilmesini bir boyutlu hücrel

otomata kullanarak ortaya koydu [27]. Sommerhalder ve Westrhenen deterministik olmayan bir boyutlu hücrel otomata ile düzenli dil kümesinin kabul veya ret edilmesinin test edilmesini gerçekleştirdiler. [28].

Hücrel otomatların en çok kullanıldığı uygulama alanlarından biride görüntü işlemedir. Hücrel otomatları ızgara düzlemi üzerinde çalışan hesaplama yapısı ve görüntü piksellerinin matris yapısı arasındaki benzerlik birçok araştırmacının dikkatini çekmiştir.

Hücrel otomatların eş zamanlı hesaplama imkan veren çalışma mantığı görüntü işleme gibi hesaplama maliyeti yüksek bir işlem için uygun bir fırsat sunuyordu.. Hücrel otomatlar kullanılarak yapılan ilk görüntü işleme çalışmaları ikili görüntüler üzerinde oluyordu [29] [30] [31] [32]. Yapılan çalışmalarda görüntü işleme amacına uygun hücrel otomata tabanlı işlemciler tasarlanırdır. Bu işlemcilerin komut setleri yapılan çalışmalarda rapor edilirdi.

Hücrel otomatların en çok kullandığı görüntü işleme uygulamaları, görüntüler üzerinde kenar bulma, gürültü giderme, inceltme ve bölümlenme gibi amaçlar içindir. Rosin ikili görüntüler [33] ve gri görüntüler [34] üzerinde işlem yapan iki ayrı çalışma yapmıştır. Görüntülerde kullanılacak kural seçimi için deterministik ve ardışık yüzen ileri arama(sequential floating forward search) mekanizmasını kullanmıştır.

Hücrel otomata temelli diğer bir görüntü işlem GrowCut adlı bölümlendirme çalışmasıdır [35]. Yöntem hücrel otomat fikrine dayalı interaktif bir görüntü bölümlendirme algoritmasıdır. Algoritma, rekabetçi bölge büyümesini andıran yineleyici bir etiketleme prosedürü kullanarak, orta ve zor nesnelerin 2D ve 3D'de güvenilir ve makul derecede hızlı bölünmesini sağlar. "GrowCut" çalışmasına benzer bir çalışma Ghosh ve ark. tarafından gerçekleştirilmiştir. Ghosh ve ark., denetimsiz büyüme kesme algoritması (unsupervised grow-cut algorithm) ile etkileşim halinde bir görüntü bölümlendirme yöntemi önermişlerdir [36].

2.2. Başlıkla İlgili Çalışmalar(Topic Related Works)

Örüntü tanıma uzun yıllardır üzerinde çalışılan bir konu ve örüntü tanıma uygulamaları [37] [38] yapay zekâ, robotik [39], bilgisayar mühendisliği, genetik, biyomedikal görüntü analizi, uzay yol yönlendirmeleri vb. bir çok alanda çözüm için geliştirilmektedir. Hücrel otomatların eş zamanlı hesaplama imkan sağlayan çalışma mantığının kullanıldığı alanlardan biride örüntü tanımadır.

Hücrel otomatlar üzerine yapılan ilk çalışmalar doğrudan örüntü tanıma üzerine olmamasına rağmen örüntü tanımda kullanılabileceğine dair kuvvetli bulgular içermektedir [40]. Hücrel otomata durum uzayında bazı durumlar diğer durumlar arasında dinamik bir ilişkiye sahipti. Bu durumlar diğer durumları etrafında toplayan çekim formunda davranış sergiliyordu. Bu durumlar cezbedici durum olarak adlandırıldı [19]. Bir durumun

diğer durumları çekmesi bir örüntünün diğer örüntüleri çekmesi olarak düşünülebileceği fikri cezbedici durumların örüntü tanıma ve sınıflandırmada kullanılmasını sağlamıştır [20].

Cezbedicilerin örüntü çekme yetenekleri yüz tanıma [41], hareket tanıma [42], boyut azaltma [43] gibi bir birinden farklı alanlarda kullanılmıştır. Peldek ve Becerikli cezbedici durumların örüntü çekme yeteneklerini boyut azaltmada kullanmışlardır [43]. Yapılan çalışmada cezbediciler benzer özellikleri temsil eden kuvvetli özellik olarak değerlendirilmiş ve özellik uzayı azaltılmıştır.

Cezbedici durumların örüntü tanımda kullanılması için yapılan çalışmaları temelde iki ana kategoriye ayırabilir; lineer [44] hücrel otomatlar ve lineer olmayan [20] [19] hücrel otomatlar. Lineer hücrel otomatlarda cezbedici durumların nitelenmesi için cebirsel işlemler kullanılabilir için lineer olarak adlandırılmıştır. Lineer olmayan hücrel otomatlarda cezbedici durumların nitelenmesi için cebirsel olmayan yöntemler geliştirilmiştir.

Lineer yöntemlerin cezbedici durumların nitelenmesinde kullanılabilmesi için kuralların XOR veya XNOR mantıksal operatörlere göre işlem yapması gerekir. Bu yaklaşım iki durumlu ve üç komşuya sahip hücrel otomatların kullanılabileceği 256 farklı kural sayısını 14'e indirgemektedir. Bu nedenle cezbedici durumların nitelenmesi için lineer olmayan yöntemler geliştirilmiştir.

Örüntü tanımda kullanılan lineer olmayan hücrel otomata yöntemlerini iki kategoriye ayırabilir; genetik algoritma tabanlı [20] ve erişilebilirlik ağacı tabanlı [19] yöntemler.

Bu çalışmada erişilebilirlik ağacı tabanlı [19] cezbedici durum nitelendirme yöntemleri örüntü tanımda kullanılmıştır. Erişilebilirlik ağacı tabanlı yöntemler ile geliştirilen algoritmalar zaman ve hafıza karmaşıklığı açısından verimli bir performansa sahiptir.

3. UYGULANAN YÖNTEM (APPLIED METHOD)

Bu bölümde 2 duruma ve 3 komşuya sahip 1 boyutlu hücrel otomatların durum güncelleme mantığı ve tek uzunluk döngülü cezbedicilerin nitelenmesi anlatılmıştır. Tek uzunluk döngülü cezbedicilerin nitelenmesi erişilebilirlik ağacı kullanılarak gerçekleştirilmiştir.

3.1. Hücrel Otomata Temelleri (Cellular Automata Basic)

Hücrel otomatların çalışma mantığı ızgara(veya matris) düzleminde yer alan hücrelerin komşuları ile olan etkileşimine dayanır. Hücreler, sınırlı sayıda değere sahip olabildiği için ayrık bir yapıya vardır. Hücrelerin durumu kendi ve etkileşimde olduğu hücrelerin tuttuğu değerlere(duruma) göre belirlenir. Hücrelerin değerlerinin değişmesi zaman adımlarına göre yapılır.

Bir hücrenin t+1 zamanındaki değeri, kendisi, sol ve sağ komşularının t zamanındaki değerlerine bağlı olarak güncellenir. Bu üç hücrenin(kendisi, sol ve sağ komşusu) t zamandaki değerleri bir fonksiyona girdi olarak verilir ve hücrenin t+1 zamanındaki değeri elde edilir. Bir hücrenin t zamandaki değerini S_i^t sembolü ile gösterilsin, t+1 zamanındaki değerini aşağıdaki gibi formüle edebilir:

$$S_i^{t+1} = f_i(S_{i-1}^t, S_i^t, S_{i+1}^t) \quad (1)$$

Yukarıdaki formülde i'ninci hücrenin t zamandaki değeri S_i^t ; sol ve sağ komşularının değerleri S_{i-1}^t ve S_{i+1}^t ; f_i fonksiyonuna girdi olarak veriliyor ve hücrenin bir sonraki durumu yani t+1 zamanındaki durumu elde ediliyor.

Bütün bir hücrel düzlemin durum güncellemesini aynı yolla formüle edebilir. Bir hücrel düzlemin t zamandaki durumu S^t . N tane hücreden oluşan bir hücrel düzlemin t zamandaki durumu daha açık bir ifade ile aşağıdaki gibi formüle edilebilir.

$$S^t = (S_1^t, S_2^t, \dots, S_n^t) \quad (2)$$

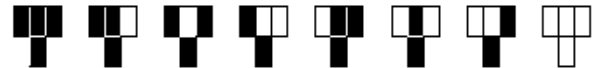
Yukarıdaki 1 ve 2 numaralı formülleri temel alarak bir hücrel düzlemin bir sonraki durumunu aşağıdaki gibi formüle edebilir:

$$S^{t+1} = (f_1(S_0^t, S_1^t, S_2^t), f_2(S_1^t, S_2^t, S_3^t), \dots, f_n(S_{n-1}^t, S_n^t, S_{n+1}^t)) \quad (3)$$

3.2. Kural (The Rule)

Hücrel otomatlarda bir sonraki durumun elde edilmesi fonksiyon yerine kural kelimesi ile ifade edilir. Kural fonksiyonun yapısı olarak düşünülebilir. Her bir hücrenin bir sonraki durumu belirlenen kurala göre güncellenir.

Hücrel otomatların görselleştirilebilir çalışma mantığı kural kavramının ne demek olduğunun daha iyi anlaşılmasına yardımcı olacaktır. Her bir hücre 0 ve 1 değerlerinden oluştuğuna göre bu iki değeri siyah ve beyaz iki renkle temsil edebiliriz. Siyah renk 1 değerine ve beyaz renk 0 değerine karşılık gelir [6]. Aşağıdaki şekildeki gibi görselleştirebilir:



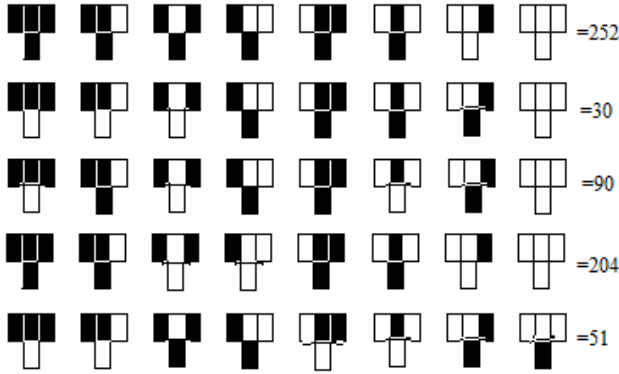
Şekil 1. Kural 254'e göre ortadaki hücrelerin bir sonraki durumu

(The next state of the middle cell according to Rule 254.)

Şekil 1'de iki durumlu ve üç komşuya(kendi dahil) sahip bir hücrenin sekiz farklı duruma göre bir sonraki durumları kural 254'e göre verilmiştir. Şekil 1'de yer alan 'T' biçimindeki şekillerdeki üstteki üç hücre; sol, kendi ve sağ hücrelerini temsil etmektedir. Altındaki tek hücre ortadaki hücrenin kural 254'e göre bir sonraki durumunu göstermektedir.

Şekil 1'deki bir sonraki durum güncellemesini kural 254'e göre yapıldığı nasıl belirlenir? Bunun açıklaması ortadaki hücrenin bir sonraki durumunun ikilik tabandaki karşılığında gizlidir. Siyah hücreler 1 ve beyaz hücreler 0 değerine karşılık geldiği için T şekillerinde üstteki üç hücrenin altındaki hücrelerin ikilik tabandaki karşılığı şöyle olacaktır; 11111110=254.

Aşağıda 252, 30, 90, 204 ve 51 kurallarına ait bir sonraki durum gösterimi yer almaktadır.



Şekil 2. 252, 30, 90, 204 ve 51 kurallarına göre ortadaki hücrelerin bir sonraki durumu.

(The next state of the middle cell according to Rules 252, 30, 90, 204 and 51)

İki duruma sahip üç tane hücrenin sekiz farklı kombinasyonu olabilir. Bu kural uzunluğunun sekiz bitten oluşacağı anlamına gelir. Bu $2^8=256$ farklı kuralı uygulama imkânı sağlar.

Hücresel düzlemde tüm hücrelere aynı kuralın uygulanması yaklaşımına tek düze hücresel otomatlar, birden fazla kuralın uygulanmasına melez hücresel otomata adı verilir.

Kuralların hücresel düzleme uygulanması kural vektörü kullanılarak gerçekleştirilir. Örneğin beş hücreden oluşan bir hücresel düzleme uygulanacak kural vektörü şöyle gösterilir: <252, 30, 90, 204, 51>. Kural vektöründeki tüm kurallar aynı olmadığı için bir melez hücresel otomata yaklaşımıdır.

3.3. Kural Min Terimi (Rule Min Term)

Bir boyutlu hücresel otomatlarda üç hücrenin durumu kuralda kullanacak basamak numarasını verir. Örneğin $\blacksquare\blacksquare\blacksquare$ durumu için ilgili kuralda altıncı(110=6) basamak kullanılır. $\blacksquare\blacksquare\blacksquare$ durumu için ortadaki hücrenin bir sonraki durumu; 252, 30, 90, 204 ve 51 kurallarına göre sırasıyla; siyah, beyaz, siyah, siyah ve beyaz renkleridir. Diğer bir ifade ile 110 durumu için ortadaki bitin bir sonraki durumu; 252, 30, 90, 204 ve 51 kurallarına göre sırasıyla; 1, 0, 1, 1 ve 0 değerleridir.

Bir sonraki durumun kurala ait hangi basamağa göre olacağı Kural Min Terimi(KMT) ile ifade edilir.

Yukarıdaki örnek için düşünülecek olursa, bir sonraki durumu ilgili kurala ait KMT6'ya göre güncellenir.

Her bir kural sekiz bitten oluştuğu için sekiz KMT'ye sahiptir; KMT0, KMT1, .., KMT7 biçiminde numaralandırılır. Aşağıdaki tabloda 252, 30, 90, 204 ve 51 kurallarına ait KMT ilişkisi yer almaktadır.

Tablo 1. <252, 30, 90,204, 51> kural vektöründeki kuralların KMT ilişkileri

(KMT relations of the rules in rule vector <252, 30, 90, 204, 51>)

	111	110	101	100	011	010	001	000
	KMT7	KMT6	KMT5	KMT4	KMT3	KMT2	KMT1	KMT0
252	1	1	1	1	1	1	0	0
30	0	0	0	1	1	1	1	0
90	0	1	0	1	1	0	1	0
204	1	1	0	0	1	1	0	0
51	0	0	1	1	0	0	1	1

Yukarıdaki tabloda kural ait KMT ilişkisi gösterilmektedir. En soldaki sütunda kurallar yer almaktadır. Kuralların sağındaki hücrelerde kuralların ikilik tabandaki değerleri yer alır. En üstteki satırda ikilik tabandaki kuralın bir sonraki durum güncellenmesinde kullanılacak basamağını ifade eden KMT'ler yer almaktadır. Örneğin 100=4 durumu için ortadaki hücrenin bir sonraki durumu 252, 30, 90, 204 ve 51 kurallarına göre KMT4'e dörde göre sırasıyla 1, 1, 1, 0 ve 1 olarak güncellenir.

Kural vektörüne göre bir sonraki durumun elde edilmesini örnek vererek açıklamak faydalı olacaktır. 5 hücreden oluşan bir hücresel düzlem için kullanılan kural vektörü Tablo 1'de verilen <252, 30, 90, 204, 51> kural vektörü olsun. Bu kural vektörüne göre 10101 örüntüsünün bir sonraki durumunu ilgili kuralda kullanılacak KMT değerleri belirleyecektir.

Bu çalışmada sıfır sınırlı hücresel otomata kullanıldığı için, bir sonraki durumu bulmadan önce örüntünün sol ve sağ uçlarına 0 eklenmesi gerekir: 0101010 . 0101010 örüntüsündeki her bir bitin bir sonraki durumunun bulunmasıyla örüntünün bir sonraki durumu bulunur.

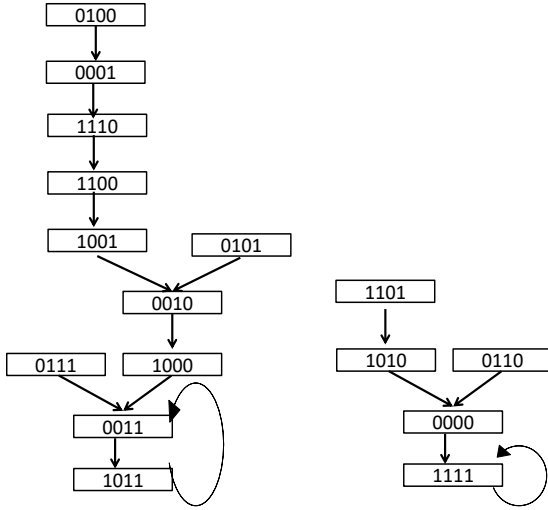
Örüntüyü soldan sağa doğru tarayarak bir sonraki durumunu bulalım. Birinci bitin(010) bir sonraki durumu 252 kuralının KMT2'sine göre 1'dir. İkinci bitin(010) bir sonraki durumu 30 kuralının KMT5'ne göre 0'dır. Üçüncü bitin(010) bir sonraki durumu 90 kuralının KMT2'sine göre 0'dır. Dördüncü bitin(010) bir sonraki durumu 204 kuralının KMT5'ne göre 0'dır. Beşinci bitin(010) bir sonraki durumu 51 kuralının KMT2'sine göre 0'dır. Bu şartlarda 10101 örüntüsünün bir sonraki durumu 10000 örüntüsüdür.

3.4. Cezbediciler (Attractors)

Hücresel otomata üzerine yapılan çalışmalarda bazı durumların çekim formunda davrandığını ve diğer durumları etrafına topladığını gözlemlerler. Bu bir

örüntünün diğer örüntüleri çekmesi anlamında değerlendirildi ve hücrel otomatların örüntü tanıma ve sınıflandırma potansiyelini ortaya çıkardı.

Durum geçiş diyagramı n hücreli bir otomata düzleminde 2^n tane durumun önceki ve sonraki durumlarını gösteren graf yapısıdır. Durum geçiş diyagramlarını gözlemleyerek cezbedici durumları daha iyi anlaşılabilir. Aşağıda $\langle 105, 129, 171, 65 \rangle$ kural vektörüne ait durum geçiş diyagramları yer almaktadır.



Şekil 3. $\langle 105, 129, 171, 65 \rangle$ kural vektörüne ait durum geçiş diyagramları.

(State transition diagrams belong to the rule vektör $\langle 105, 129, 171, 65 \rangle$)

Kural uzunluğu dört olan bir kural vektörü $2^4 = 16$ farklı durumun bir sonraki durumunu belirler. Yukarıdaki durum geçiş diyagramlarında 16 farklı durum yer almaktadır.

Bir veya birkaç zaman adımı sonraki durumu kendi olan durumlara cezbedici durumlar adı verilmiştir. Yukarıdaki durum geçiş diyagramlarında iki tane cezbedici durum vardır. Cezbedici durumlar geçiş diyagramlarının en altında ve kavisli oklar ile gösterilmiştir. Cezbedici durumlardan biri $1111 \rightarrow 1111$ döngüsüne sahip diğeri ise $1011 \rightarrow 0011 \rightarrow 1011$ döngüsüne sahiptir.

Yukarıdaki geçiş diyagramında yer alan cezbedicilerden biri tek uzunluk döngülü cezbedici ($1111 \rightarrow 1111$), diğeri ise çok uzunluk döngülü cezbedici ($1011 \rightarrow 0011 \rightarrow 1011$) olarak adlandırılır. Tek uzunluk döngülü cezbedici durumun bir sonraki durumu kendisi iken, çok uzunluk döngülü cezbedici birkaç durum sonra kendine döner.

3.5. Özellik 1 (Property 1)

Cezbedici durumlar diğer durumların bir sonraki durumu kendisidir. Bu noktada kural vektörünün yapısı önemli bir rol oynamaktadır. Kurallar cezbedici durum oluşturma potansiyeline göre farklılık gösterirler. Kuralların cezbedici durum oluşturmaya katkısı "özellik 1" adlandırılması kullanılarak tanımlanmıştır [19]. Das ve arkadaşlarının "Özellik 1" tanımı aşağıda yer almaktadır.

Özellik 1: Bir kuralın cezbedici durum oluşturmaya katkısı kuralın KMT'lerine karşılık gelen bitlerin değerine göre belirlenir. KMT0, KMT1, KMT4 ve KMT5 bitlerinin 0 olması ve KMT2, KMT3, KMT6 ve KMT7 bitlerinin 1 olması cezbedici oluşumuna katkı sağlar.

Tablo 1'de yer alan kuralların özellik 1'i karşılayan KMT'leri şöyledir:

252: KMT0, KMT1, KMT2, KMT3, KMT6, KMT7

30: KMT0, KMT2, KMT3, KMT5

90: KMT0, KMT3, KMT5, KMT6

204: Tüm KMT'leri özellik 1'i karşılar: KMT0, KMT1, KMT2, KMT3, KMT4, KMT5, KMT6, KMT7

51: Hiçbir KMT'si özellik 1'i karşılamaz

Bu çalışmada tek uzunluk döngülü cezbediciler örüntü sınıflandırmada kullanılmıştır. Cezbedici durumların nitelenmesi için erişilebilirlik ağacı kullanılmıştır. Bir sonraki bölümde erişilebilirlik ağacının inşa edilmesi anlatılmıştır.

3.6. Erişilebilirlik Ağacı (Reachability Tree)

Das ve arkadaşları hücrel otomata durum uzayında ulaşılabilir ve ulaşılamaz durumların nitelenmesi için erişilebilirlik ağacı önermişlerdir [45]. Yukarıdaki Şekil 3'te yer alan durum geçiş diyagramlarında bazı durumların atası yoktur. Atası olmayan durumlar ulaşılamaz durum olarak adlandırılmıştır [19]. Şekil 3'te yer alan 0111, 0100, 0101, 1101 ve 0110 durumları atası olmadığı için ulaşılamaz durumlardır, geri kalan durumlar ulaşılabilir durumlardır.

Erişilebilirlik ağacındaki her bir düğüm ilgili kuralın KMT'leri kullanılarak oluşturulur. Düğümlerin solundaki kenar 0-kenarı ve sağdaki kenar 1-kenar olarak adlandırılır. N hücreli(veya kuralı) bir hücrel otomatın erişilebilirlik ağacı (n+1) seviyeden oluşur. Kök düğüm seviye 0'da ve yaprak düğümler seviye n'de yer alır. Seviye i'de yer alan düğümler kural vektöründeki (i+1)'inci kuralın KMT'lerine göre oluşturulur [45] [19].

Erişilebilirlik ağacındaki yaprak düğümler erişilebilir durumların değerini tutar. Kök düğümünden ilgili yaprak düğüme gitmek için kullanılan kenar dizisi yaprak düğümün tuttuğu değeri verir.

Erişilebilirlik ağacı, kural vektörü soldan sağa doğru taranarak oluşturulur. Kök düğüm soldan birinci kurala göre oluşturulur. Kural vektörü taranırken yavaşık iki kuralın KMT'leri dikkate alarak oluşturulur. i'ninci kural ile (i+1)'inci kuralın KMT'leri bir biriyle ilişkilidir. Aşağıdaki tabloda bu ilişki gösterilmektedir.

Tablo 2. i ve $i+1$ kurallarının KMT'leri arasındaki ilişki [19].(The relationship between the KMTs of i and $i+1$ rules.)

i 'nci kuralda kullanılan KMT	$(i+1)$ 'inci kuralda kullanılabilecek KMT'ler
KMT0(000)	KMT0(000),KMT1(001)
KMT1(001)	KMT2(010),KMT3(011)
KMT2(010)	KMT4(100),KMT5(101)
KMT3(011)	KMT6(110),KMT7(111)
KMT4(100)	KMT0(000),KMT1(001)
KMT5(101)	KMT2(010),KMT3(011)
KMT6(110)	KMT4(100),KMT5(101)
KMT7(111)	KMT6(110),KMT7(111)

Tablo 2'de i 'nci ve $(i+1)$ 'inci kurallara ait KMT'ler arasındaki ilişki verilmektedir. Örneğin i 'nci kuralda kullanılan KMT 0 ise bir sonraki kuralda kullanılabilecek KMT'ler KMT 0 ve KMT 1'dir. i 'nci kuralda kullanılacak KMT numarasına A olsun, $(i+1)$ 'inci kuralda

kullanılabilecek KMT numaraları B ve C olsun. B ve C hesaplamasını aşağıdaki gibi formüle edebilir [19];

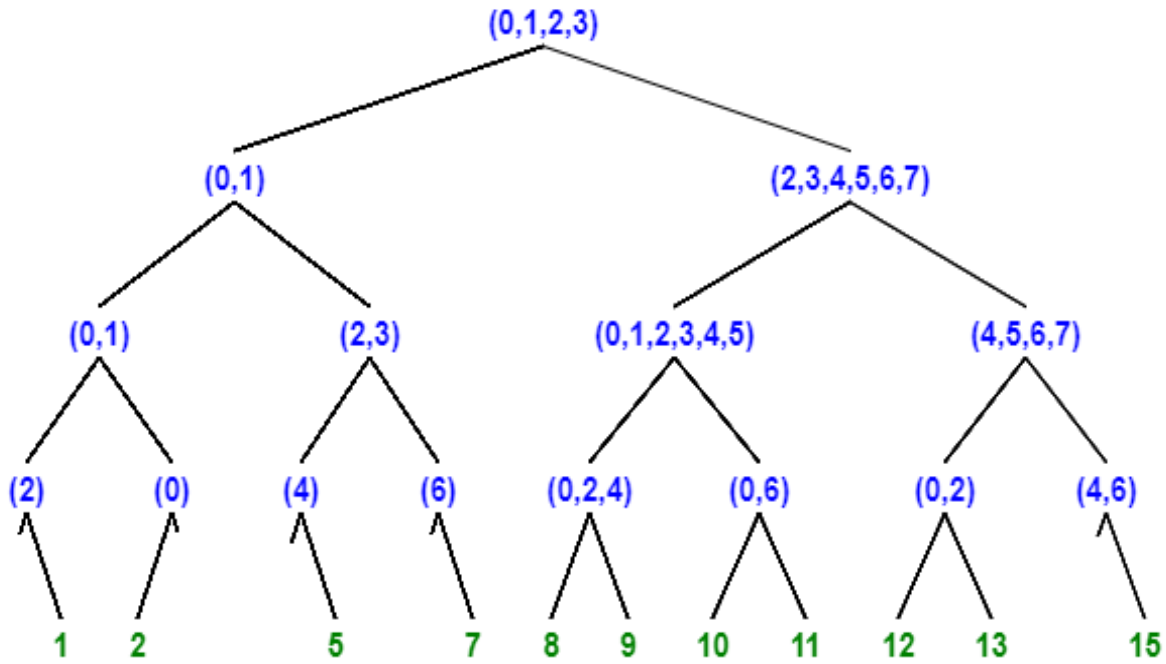
$$B = (2 * A) \bmod 8 \quad (4)$$

$$C = (2 * A) \bmod 8 + 1 \quad (5)$$

Erişilebilirlik ağcını oluştururken Tablo 2'de yer alan iki yanaşık kural arasındaki ilişkiden faydalanılır. Aşağıda Tablo 3'te $\langle 78, 142, 201, 92 \rangle$ kural vektörüne ait KMT ilişkisi gösterilmekte ve Şekil 4'te bu kural vektörüne göre inşa edilen erişilebilirlik ağacı yer almaktadır.

Tablo 3. $\langle 78, 142, 201, 92 \rangle$ kural vektöründeki kuralların KMT ilişkileri(KMT relations of the rules in rule vector $\langle 78, 142, 201, 92 \rangle$)

	111	110	101	100	011	010	001	000
	KMT7	KMT6	KMT5	KMT4	KMT3	KMT2	KMT1	KMT0
78	d	d	d	d	1	1	1	0
142	1	0	0	0	1	1	1	0
201	1	1	0	0	1	0	0	1
92	d	1	d	1	d	1	1	d

Şekil 4. $\langle 78, 142, 201, 92 \rangle$ kural vektörüne ait erişilebilirlik ağacı.(Reachability tree belongs to rule vektör $\langle 78, 142, 201, 92 \rangle$)

Ağaçtaki düğümlerde ilgili kuralın KMT'leri yer almaktadır. Kök düğümü oluşturmak için kural 78'i kullanılacaktır. Kök düğümün kullanılabilecek KMT'leri KMT0, KMT1, KMT2 ve KMT3'tür. Sıfır sınırı hücreyel otomata kullanıldığı için KMT3'ten sonraki KMT'ler kök düğüm için kullanılmamaktadır ve tabloda d harfi atanmıştır. Aynı durumu son kuralın tek numaralı KMT'leri için de geçerlidir.

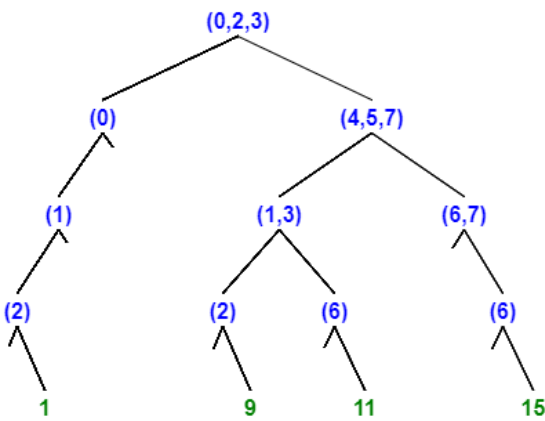
Kök düğümünden sonraki her seviyedeki düğümün oluşturulmasını ebeveyn düğümündeki KMT'leri kullanarak gerçekleştirilir. Seviye1'de yer alan düğümlerdeki KMT numaraları kök düğümde yer alan KMT numaraları kullanılarak elde edilmiştir. Daha genel bir tanımla; $(i+1)$ 'nci seviyedeki düğümlere ait KMT numaralarının elde edilmesini i 'nci seviyede yer alan ebeveyn düğüm kullanılarak gerçekleştirilir.

Erişilebilirlik ağacındaki düğümlerin sol kenarının 0 ve sağ 1 olduğunu ve kök düğümde kullanılan KMT numaralarının (0, 1, 2, 3) olduğunu yukarıda belirtilmişti. Bu KMT'lerin kural 78'in ikilik tabandaki değerine göre sahip olacakları bit değeri sırasıyla 1, 1, 1, 0'dır. Bu değerler bir sonraki seviyedeki düğümlerin oluşturulmasında kullanılacak KMT'leri belirlenmesine yarayacaktır. 0 değerine sahip KMT'ler sol kenara bağlı çocuk düğümün oluşturulmasında, 1 değerine sahip KMT'ler sağ kenara bağlı çocuk düğümün oluşturulmasında kullanılacaktır.

Kural 78'in KMT0 sol kenarda ve KMT1, KMT2, KMT3 sağ kenarda yer alacaktır. Sol kenarda yer alan çocuk düğüm için KMT0'ı ve sağ kenarda yer alan çocuk düğüm için KMT1, KMT2 ve KMT3'ü kullanılacaktır. Seviye 1'de yer alan (0,1) KMT numaraları Tablo 3'te sol sütunda KMT0'ın karşısında yer alan KMT'lerdir. Seviye 1'de yer alan (2, 3, 4, 5, 6, 7) KMT numaralarına sahip düğümün KMT numaraları kök düğümün sağında yer alan KMT1, KMT2 ve KMT3'ten türetilmiştir. Erişilebilirlik ağacındaki tüm düğümler aynı yaklaşımla oluşturulur.

Yaprak düğümdeki değer kök düğümünden ilgili yaprak düğüme gitmek için kullanılacak kenar dizisine göre belirlenir. Örneğin kök düğümünden yaprak düğüme gitmek için kullanılan kenarlar sol, sağ, sağ ve sağ ise, yaprak düğüm değeri 0111=7 olacaktır.

Erişilebilirlik ağacını sadece cezbedici durumları temsil edecek yapıya dönüştürülebilir. Bunun için düğümlerde yer alan KMT'lerin özellik 1'i karşılaması yeterli olacaktır. Cezbediciler için oluşturulan erişilebilirlik ağacındaki yaprak düğümlerin hepsi tek uzunluk döngülü cezbedici durumlardır. Şekil 5'te <78, 142, 201, 92> kural vektörüne ait cezbedici durumlar için erişilebilirlik ağacı yer almaktadır.



Şekil 5. <78, 142, 201, 92> kural vektörüne ait cezbedici durumlar için erişilebilirlik ağacı.

(Reachability tree efor attractor state belongs to the rule vector <78, 142, 201, 92>)

Kapsayıcı bitler cezbedici durumlar daha az sayıda bitle temsil etmeye yarayan özel konumdaki bitlerdir. Kapsayıcı bitler erişilebilirlik ağacına bakılarak kolaylıkla elde edilebilir.

Cezbediciler için erişilebilirlik ağacında sola ve sağa kenarları olan düğümlerin yer aldığı kurallara ait bitler kapsayıcı bitlerdir [19]. Şekil 5'te yer alan cezbediciler için erişilebilirlik ağacında kök düğüm, seviye 1'deki (4, 5, 7) KMT değerlerini tutan düğüm ve seviye 2'deki (1, 3) KMT değerlerini tutan düğümler sol ve sağ kenarlara sahiptir. Bunun için 1, 2 ve 3'üncü bitler kapsayıcı bitlerdir. Bu kapsayıcı bitler kullanılarak cezbedici durumlar tanımlanabilir.

Kural vektörünün ürettiği dört cezbedici durum şöyledir; **0001**(1), **1001**(9), **1011**(11) ve **1111**(15). Cezbedicilerin ilk üç biti dört cezbedici durum için benzersizdir. M tane kapsayıcı bit 2^m tane cezbediciyi benzersiz temsil etmek kapasitesine sahiptir [19].

3.7 Tek Uzunluk Döngülü Cezbedicilerin Nitelenmesi (Characterization of Single Length Cycle Attractor)

Bu bölümde araştırmacıların kural vektörünün sadece tek uzunluk döngülü cezbedici üretecek yapıda oluşturulması için geliştirdikleri teoremlere yer verilmiştir.

Sadece tek uzunluk döngülü cezbedici üreten ve çok uzunluk döngülü cezbedici oluşumunu kısıtlayan kural vektörü elde etmek için Mukherjee "özellik 3" adını verdiği teorem geliştirmiştir [40].

Özellik 3: S_0 ve S_1 kümeleri şöyle tanımlansın: $KMT0, KMT1, KMT4, KMT5 \in S_0$ ve $KMT2, KMT3, KMT6, KMT7 \in S_1$. Eğer i 'nci hücreye ait kural KMT'lerinden özellik 1'i karşılamayan r_n ve r_m KMT'leri varsa; bu KMT'ler $r_n \in S_0$ (veya S_1) ve $r_m \in S_1$ (veya S_0) farklı kümelere ait ise $(i+1)$ 'inci hücreye ait kuralın ardışık KMT'lerinin seçimi **çok uzunluk döngülü cezbedici oluşmaması** için aşağıdaki gibi olmalıdır [40]:

- $2 * r_n \bmod 8$, $(2 * r_n + 1) \bmod 8$ KMT'leri özellik 1'i **karşılmalı**; $2 * r_m \bmod 8$, $(2 * r_m + 1) \bmod 8$ KMT'leri özellik 1'i **karşılamamalı**
- Veya; $2 * r_m \bmod 8$, $(2 * r_m + 1) \bmod 8$ KMT'leri özellik 1'i **karşılmalı**; $2 * r_n \bmod 8$, $(2 * r_n + 1) \bmod 8$ KMT'leri özellik 1'i **karşılamamalı**
- Veya: $2 * r_n \bmod 8$, $(2 * r_n + 1) \bmod 8$, $2 * r_m \bmod 8$, $(2 * r_m + 1) \bmod 8$ KMT'leri **aynı** kümeden ise hiç biri özellik 1'i **karşılamamalı**
- Veya: $2 * r_n \bmod 8$, $(2 * r_n + 1) \bmod 8$, $2 * r_m \bmod 8$, $(2 * r_m + 1) \bmod 8$ KMT'leri **farklı** kümeden ise hepsi özellik 1'i **karşılmalı**

Özellik 3 kullanılarak sadece tek uzunluk döngülü cezbediciler üreten kural vektörleri oluşturulabilir. Ancak oluşacak cezbedici sayısı kontrol altına alınmamıştır. Das ve arkadaşları cezbedici sayısını kontrol altına alan ve

sadece tek uzunluk döngülü cezbediciler üreten kural vektörü elde etmek için iki ayrı teorem geliştirmişlerdir:

Teorem 1: N hücreli bir hücresele otomata için, eğer i 'nci bit kapsayıcı bit ise KMT 0, 1, 2, 3 veya KMT 4, 5, 6, 7 veya tüm sekiz KMT'sinde $(i-1)$ 'inci kuralın özellik 1'i sağlayan KMT'lerine bağlı olarak sağlar [19].

Açılama: n-hücreli bir hücresele otomata düşünelim; eğer birinci bit kapsayıcı bit ise erişilebilirlik ağacının 0'ıncı seviyesindeki düğümünün 0(sol) ve 1(sağ) kenarlarında yer alan KMT'lerin özellik 1'i karşılaması gerekir. Bu nedenle eğer birinci ve ikinci bitler kapsayıcı bit ise K_1 kuralına ait 0, 1, 2 ve 3 KMT'lerinin özellik 1'i karşılaması gerekir. Eğer ikinci bit kapsayıcı bit değil ise K_1 kuralına ait 0, 1 KMT'lerinin en az birinin özellik 1'i karşılaması gerekir ve K_1 kuralına ait 2, 3 KMT'lerinin en az birinin özellik 1'i karşılaması gerekir.

Teorem 2: N hücreli bir hücresele otomata için: Eğer i 'nci bit kapsayıcı bit değil ise K_i kuralının oluşturulması $(i+1)$ 'inci bitin kapsayıcı bit olup olmamasına göre oluşturulması aşağıdaki gibidir [19].

- $(i+1)$ 'inci bit kapsayıcı bit olduğunda: K_{i-1} kuralının özellik 1'i karşılayan KMT'lerine bağlı olarak K_i kuralının 0, 1, 4, 5 KMT'leri özellik 1'i karşılar ve 2, 3, 6, 7 KMT'leri özellik 1'i karşılamaz; ya da 2, 3, 6, 7 KMT'leri özellik 1'i karşılar ve 0, 1, 4, 5 KMT'leri özellik 1'i karşılamaz
- $(i+1)$ 'inci bit kapsayıcı bit olmadığında: K_{i-1} kuralının özellik 1'i karşılayan KMT'lerine bağlı olarak K_i kuralının iki denk KMT'si (Örnek: $KMT_0=KMT_4$ veya $KMT_1=KMT_5$) özellik 1'i karşılar diğer altı KMT'si özellik 1'i karşılamaz.

Açıklama: N hücreli bir hücresele otomata düşünülün. Eğer birinci bit kapsayıcı bit değilse ve ikinci bit kapsayıcı bit ise birinci bitin 0, 1 KMT'leri veya 2, 3 KMT'leri özellik 1'i karşılamalıdır. İkinci bit kapsayıcı bit olduğundan 0, 1, 2, 3 KMT'leri veya 4, 5, 6, 7 KMT'leri özellik 1'i karşılamalı ki çok uzunluk döngülü cezbedici oluşumunu engellesin. Bu nedenle ikinci bit kapsayıcı bit olduğunda birinci kural özellik 1'i ya 0, 1 KMT'lerinde veya 2, 3 KMT'lerinde biri karşılamalıdır. Eğer ikinci bit kapsayıcı bit değil ise birinci kuralın 0, 1, 2, 3 KMT'lerinden yalnızca bir KMT özellik 1'i karşılamalıdır.

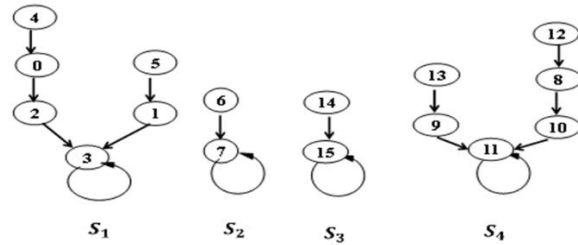
Bu bölümde anlatılan hücresele otomatların cezbedici nitelenmesi ve sentezi örüntü tanımaya kolaylıkla uygulanabilir. Bir sonraki bölümde hücresele otomatların iki ayrı veri seti üzerinde ikili sınıflandırmada kullanılması anlatılmıştır.

4. HÜCRESEL OTOMATA TEMELLİ İKİLİ SINIFLANDIRMA (CELLULAR AUTOMATA BASED BINARY CLASSIFICATION)

Cezbediciler şimdiye kadar örüntü tanımının kapsamında olan veri madenciliği, görüntü sıkıştırma ve elektrik devrelerin arıza teşhisi çalışmalarında kullanılmıştır. Birbirinden farklı bu uygulamalarda kullanılması, cezbedicilerin ne kadar kullanışlı olduğunu göstermektedir.

4.1. Yaklaşımın Temel Mantığı (Basic Logic of Approach)

N tane kuraldan oluşan bir kural vektörünün ürettiği cezbediciler doğal bir sınıflandırıcı olarak düşünülebilir [19] [20]. Kural vektörünü oluşturan kuralların meydana gelecek sınıflandırıcıları nasıl etkilediğini yukarıda bahsedildi. Şekil 6'da görüldüğü gibi her bir cezbedici bir sınıfı temsil edebilir.



Şekil 6. Cezbedicilerin sınıflandırıcı olarak değerlendirilmesi [19]
(Evaluation of attractors as classifier)

Şekil 6'da dört cezbedici dört farklı sınıflandırıcı olarak numaralandırılmıştır. Eğitim aşamasında, bu sınıflandırıcıların temsil ettikleri sınıflar belirlenir.

Bu çalışmada cezbediciler ikili sınıflandırma için kullanılmıştır. Eğitimi aşamasında S_1, S_2 ve S_3 sınıflandırıcıları bir sınıfı ve S_4 diğer sınıfı temsil edebilir. Bu durum eğitim aşamasında sınıflandırıcıların en çok temsil ettikleri sınıfa göre belirlenir.

4.2. Kullanılan Veri Setleri (Used Datasets)

Bu çalışmada kullanılan veri setleri UCI makine öğrenmesi veri setleri kaynağında alınmıştır. Kullanılan veri setleri ve özellikleri aşağıdaki Tablo 4'te yer almaktadır. Kullanılan veri setlerinin hepsi ikili sınıflandırma yapısına sahiptir.

Tablo 4'de yer alan veri setlerinden Monk1, Monk2 ve Monk3 veri setlerindeki özellikler kategorik değerlerden oluşmaktadır. Özelliklerin hücresele otomata sınıflandırıcı modeline uygun hale getirmek için ikilik tabana dönüştürülmüştür. Veri setinin ilk hali ve dönüşümden sonraki hali Şekil 7'de gösterilmektedir.

Bir sonraki bölümde ön işlemleri yapılan iki veri seti üzerindeki deneysel sonuçlara yer verilmiştir.

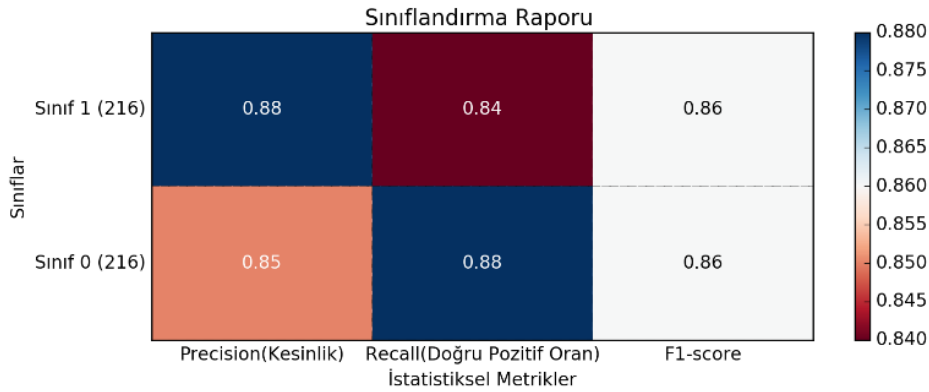
4.3. Deneysel Sonuçlar (Experimental Results)

Makine öğrenmesi modeli, veri setindeki verilerin bir kısmını modeli eğitmek için bir kısmını modeli test etmek için kullanılır. Küçük sayıda örneğe sahip veriler için önerilen eğitim ve test oranlar sırasıyla yüzde 80 ve yüzde

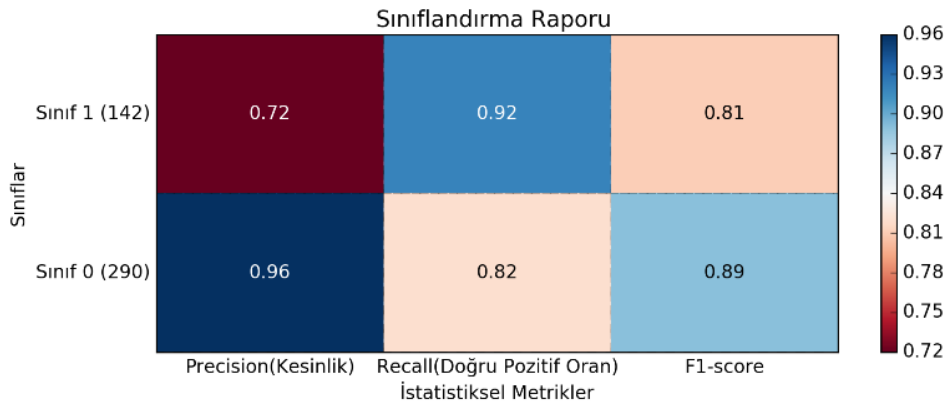
20'dir. Pima Indian veri setleri için bu oranlar kullanılmıştır.

Monk veri setleri setini oluşturulan kişiler tarafından eğitim ve test olarak ayrıldığı için bu veri setlerindeki eğitim ve test örnekleri buna göre alınmıştır.

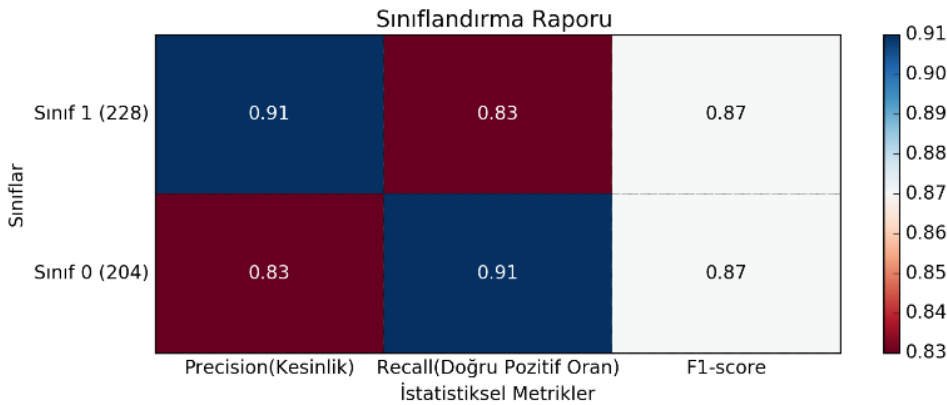
Aşağıdaki 9, 10, 12 ve 13 numaralı şekillerde tüm veri setleri için elde edilen sınıflandırma raporları yer almaktadır.



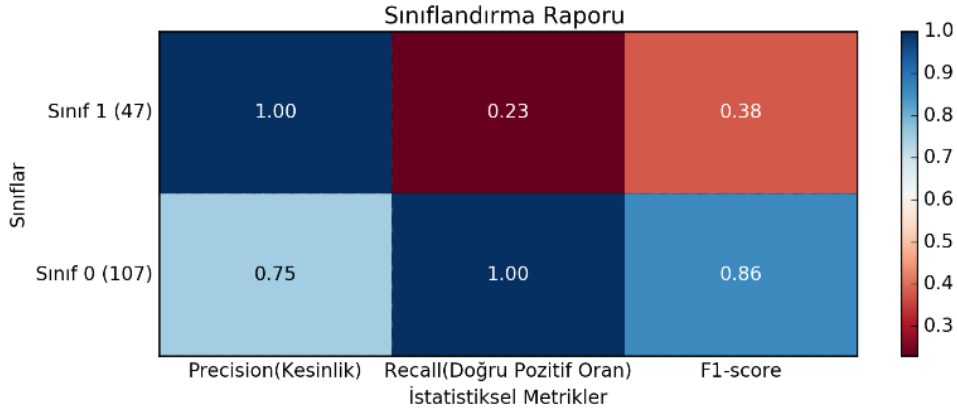
Şekil 9. Monks1 İçin Sınıflandırma Raporu
(Classification Report for Monks1)



Şekil 10. Monks2 İçin Sınıflandırma Raporu
(Classification Report for Monks2)



Şekil 11. Monks3 İçin Sınıflandırma Raporu
(Classification Report for Monks3)



Şekil 12. Pima Indian İçin Sınıflandırma Raporu
(Classification Report for Pima Indian)

4.5. Diğer Çalışmalar İle Karşılaştırma (Comparison with Other Works)

Bu bölümde elde edilen başarı oranlarının karşılaştırması iki farklı tablo içerisinde verilmiştir. Tablo 5'te önde gelen makine öğrenmesi algoritmaları ile bu çalışma kapsamında

elde edilen sonuçlara yer verilmiştir. Tablo 6'da literatürde yapılan çalışmalar ile bu çalışma kapsamında elde edilen sonuçlar yer almaktadır.

Tablo 5. Önde gelen makine öğrenmesi algoritmaları ile elde edilen sonuçlar
(Our results with state of the art machine learning algorithms)

Veri Seti	Yöntem	Performans(%)
Monks 1	Karar Ağaçları	93.52
	Rasgele Orman	91.20
	Elde Edilen Başarı Oranı	86.11
	Destek Vektör Makineleri: kernel=rbf	81.71
	En Yakın Komşu Algoritması	81.48
	Gauss Naive Bayes	67.59
	Destek Vektör Makineleri kernel=lineer	66.67
	Logistik Regresyon	66.67
	Lineer Ayrıştırma Analizi	66.20
	Perceptron	50.69
Monks 2	Elde Edilen Başarı Oranı	84.86
	Karar Ağaçlar	82.87
	Rasgele Orman	78.94
	En Yakın Komşu Algoritması	68.75
	Destek Vektör Makineleri: kernel=rbf	67.59
	Destek Vektör Makineleri kernel=lineer	67.13
	Logistik Regresyon	65.51
	Gauss Naive Bayes	64.35
	Lineer Ayrıştırma Analizi	62.96
	Perceptron	57.41
Monks 3	Destek Vektör Makineleri: kernel=rbf	96.3
	Rasgele Orman	93.52
	Karar Ağaçlar	90.05
	Gauss Naive Bayes	89.58

	En Yakın Komşu Algoritması	89.35
	Elde Edilen Başarı Oranı	87.03
	Lineer Ayrıştırma Analizi	81.48
	Destek Vektör Makineleri kernel=lineer	81.25
	Logistik Regresyon	80.09
	Perceptron	75.23
Pima Indian	Elde Edilen Başarı Oranı	76.62
	Destek Vektör Makineleri kernel=lineer	76.21
	Lineer Ayrıştırma Analizi	76.05
	Logistik Regresyon	75.08
	Gauss Naive Bayes	74.74
	Rasgele Orman	74.41
	Karar Ağaçlar	69.72
	En Yakın Komşu Algoritması	69.68
	Destek Vektör Makineleri: kernel=rbf	64.33
	Perceptron	63.06

Tablo 6. Monks veri setleri ve Pima Indian veri seti üzerinde elde edilen sonuçların literatürde yer alan diğer çalışmalar ile karşılaştırılması

(The results obtained on the Monks and Pima Indian datasets are compared with other studies in the literature)

Veri Seti	Yöntem	Performans(%)
Monks 1	Weighted KNN [46]	99.7
	CA Tabanlı [19]	91.93
	KNN [46]	89.5
	Elde Edilen Başarı Oranı	86.11
	CA Tabanlı [47]	72.87
	CA Tabanlı [40]	61.11
Monks 2	Elde Edilen Başarı Oranı	84.86
	CA Tabanlı	75.73
	CA Tabanlı [40]	67.13
	CA Tabanlı [47]	67.13
Monks 3	KNN [46]	98.6
	Weighted KNN [46]	98.6
	CA Tabanlı [19]	95.05
	Elde Edilen Başarı Oranı	87.03
	CA Tabanlı [47]	84.33
	CA Tabanlı [40]	80.65
Pima Indian	CA Tabanlı [19]	81.54
	Elde Edilen Başarı Oranı	76.62
	DTDN [48]	76.00
	CA Tabanlı [40]	73.79
	CA Tabanlı [47]	67.59

Tablo 5'te bilinen makine öğrenmesi modelleri kullanılarak elde edilen başarı oranları, bu çalışma kapsamında hüresel otomata kullanılarak elde edilen başarı oranıyla beraber verilmiştir. Elde edilen başarı oranı; Monks1, Monks2, Monks3 ve Pima Indian veri setleri üzerinde diğer yöntemlerle elde edilen başarı

oranlarıyla karşılaştırıldığında sırasıyla üçüncü, birinci, altıncı ve birinci sıradadır.

Tablo 6'da aynı veri setleri üzerinde yapılan bilimsel çalışmalarda elde edilen başarı oranları, bu çalışma kapsamında elde edilen başarı oranlarıyla beraber

verilmiştir. . Elde edilen başarı oranı; Monks1, Monks2, Monks3 ve Pima Indian veri setleri üzerinde diğer yöntemlerle elde edilen başarı oranlarıyla karşılaştırıldığında sırasıyla dördüncü, birinci, dördüncü ve ikinci sıradadır.

Elde edilen başarı oranları hücrel otomatların örüntü tanıma potansiyelinin önemli bir onayıdır. Hücrel otomatların örüntü tanımadaki kullanılabilirliği için veri setindeki değerlerin ikilik tabana dönüştürülmesi gerekir. Veri setindeki değerlerin ikilik tabana dönüşümünde veri kayıpları yaşanmaktadır. Hücrel otomata temelli sınıflandırma başarı oranının diğer yöntemlerden bazı veri setleri üzerinde düşük çıkmasının önemli bir nedeni, veri setinin ikilik tabana dönüştürülmesinde yaşanan veri kayıplarıdır. Hücrel otomata sınıflandırıcıları eğitim aşamasında sınıflar arasında en iyi ayrıştırmaya yapacak en iyi kural vektörünü bulmayı hedefler. Uygun kural vektörünün bulunması için kural vektörlerinin rasgele elde edilmesi yukarıda verilen Teorem [19] 1 ve Teorem [19] 2'ye göre yapılmıştır. Bu çalışmada kural vektörünün seçimi Das ve ark yaptığı çalışmadan [19] farklı olarak cezbedici oluşumuna katkı yapan kurallara öncelik verilmesiyle gerçekleştirilmiştir. Böylece, eğitim aşamasının azaltılmasına ve uygun kural vektörünün bulunması sürecindeki arama olasılıkları azaltılmıştır.

5. SONUÇLAR (RESULTS)

Bu çalışmada ayrık ve dinamik bir hesaplama modeli olan hücrel otomatlar ikili sınıflandırmada kullanılmıştır. İkili sınıflandırma işlemi çekim merkezi formunda hareket eden cezbediciler kullanılarak gerçekleştirilmiştir. Kullanılan sınıflandırma yöntemi için her hangi bir programla kütüphanesi mevcut değildir. Sınıflandırma kütüphanesinin kodlanması bu makalenin yazarları tarafından geliştirilmiştir. Cezbedici durumlar döngü uzunluğuna göre tek uzunluk döngülü cezbediciler ve çok uzunluk döngülü cezbediciler olmak üzere iki ayrılır. Bu çalışmada tek uzunluk döngülü cezbedicilerin örüntü çekme yetenekleri ikili sınıflandırmada kullanılmıştır. Cezbedicilerin örüntü sınıflandırma için kullanılmasında en önemli adım cezbedici üreten kural vektörünün elde edilmesidir. Kural vektörünün nitelenmesi ve elde edilmesi için erişilebilirlik ağacı temelli yöntemler kullanılmıştır.

Hücrel otomata tabanlı sınıflandırma yöntemi UCI veri seti kaynağında yer alan Monks1, Monks2, Monks3 ve Pima Indian veri setleri üzerinde kullanılmıştır. Elde edilen sonuçlar daha önceki hücrel otomata tabanlı yöntemler ve diğer bilinen makine öğrenmesi yöntemleri ile karşılaştırılmıştır. Elde edilen sonuçlar tek uzunluk döngülü cezbediciler örüntü sınıflandırma potansiyelini ortaya koymaktadır. Diğer bilinen, teorik alt yapısı yıllar önceye dayanan ve üzerinde çalışan araştırmacı sayısı çok fazla olan yöntemlerle benzer performans sonuçları vermiştir. Bazı veri setlerinde üzerinde daha iyi sonuçlar vermiştir.

Hücrel otomatların eş zamanlı programlamaya olanak sağlayan ayrık ve dinamik yapısı hücrel otomata temelli sınıflandırma yöntemlerini önemini artırmaktadır. Hücrel otomatlar sadece ikili örüntüler üzerinde işlem yapan hesaplama modeline sahip olması kullanılan yöntemin zayıf yanı olarak değerlendirilebilir. Veri setindeki verilerin hücrel otomata için uygun hale getirilmesi için yapılan ön işlemler veri kayıplarına neden olabilir. Ancak buna rağmen elde edilen sonuçlar tatmin edici seviyededir.

Bu çalışmada hücrel otomatlar sadece iki ayrı sınıfa ait veri setleri üzerinde kullanılmıştır. Daha sonraki çalışmalarda, kullanılan sınıflandırma yöntemi, çoklu sınıflara sahip veri setleri üzerinde kullanılmasını araştırılacaktır.

KAYNAKLAR (REFERENCES)

- [1] S. Das, **Theory and Applications of Nonlinear Cellular Automata In VLSI Design**, Ph. D Thesis, Bengal: The Bengal Engineering and Science University, 2006.
- [2] J. v. Neumann, **The theory of self-reproducing Automata**, A. W. Burks, Dü., Urbana and London: Univ. of Illinois Press, 1966.
- [3] V. V. Nabiyev, **Algoritmalar Teoriden Uygulamalara**, Seçkin Yayıncılık, Ankara, Türkiye, 2013.
- [4] C. G. Langton, "Self-reproduction in cellular automata", *Physica D: Nonlinear Phenomena*, 10(1), 135-144, 1984.
- [5] E. F. Codd, **Cellular Automata**, Academic Press, New York, A.B.D., 1968.
- [6] S. Wolfram, **A new kind of science**, Champaign: IL: Wolfram Media, 2002.
- [7] P. Sarkar, "A Brief History of Cellular Automata", *ACM Computing Surveys (CSUR)*, 1, 80-107, 2000.
- [8] E. F. Moore, "Machine models of self-reproduction", *Proc. Symp. Appl. Math.*, 14, 17-33, 1962.
- [9] M. Gardner, "Mathematical games: The fantastic combinations of John Conway's new solitaire game life", *Scientific American*, 223(4), 120-123., 1970.
- [10] M. Gardner, "Cellular automata, self-reproduction, Garden of Eden and Game life", *Scientific American*, 224(2), 112, 1971.
- [11] S. Wolfram, "Universality and complexity in cellular automata", *Physica D: Nonlinear Phenomena*, 10(1), 1-35, 1984.
- [12] J. Strotmann, M. Chopra, R. T. Bonnacaze, "A cellular automata simulation of atomic layer etching", *Advanced Etch Technology for Nanopatterning VII*, 10589, 105890H, 2018.
- [13] E. R. Speer, C. Maes, J. L. Lebowitz, "Probabilistic Cellular Automata: Some Statistical Mechanical Considerations", **Pattern Formation In The Physical And Biological Sciences**, CRC Press, 175-188, 2018.
- [14] K. Clarke, "Cellular Automata and Agent-Based Models", **Handbook of Regional Science**, Berlin, Heidelberg, Springer, 2018.
- [15] M. J. Phipps, "From local to global: the lesson of cellular automata", **Individual-based models and approaches in ecology**, Chapman and Hall/CRC, 165-187, 2018.

- [16] X. Liu, L. Ma, X. Li, B. Ai, S. Li, Z. He, "Simulating urban growth by integrating landscape expansion index (LEI) and cellular automata", *International Journal of Geographical Information Science*, 28(1), 148-163, 2014.
- [17] P. Rosin, A. Adamatzky, **Cellular automata in image processing and geometry**, X. Sun, Dü., Springer International Publishing, 2014.
- [18] P. P. Chaudhuri, D. R. Chowdhury, S. Nandi, S. Chattopadhyay, **Additive Cellular Automata: Theory and Applications**, Washington: Wiley-IEEE Computer Society, 1997.
- [19] S. Das, S. Mukherjee, N. Naskar, B. K. Sikdar, "Characterization of Single Cycle CA and its Application in Pattern Classification", *Electronic Notes in Theoretical Computer Science*, 181-203, 2009.
- [20] N. Ganguly, P. Maji, B. K. Sikdar, P. P. Chaudhuri, "Generalized Multiple Attractor Cellular Automata (Gmaca) Model For Associative Memory", *International Journal of Pattern Recognition and Artificial Intelligence*, 16(7), 781-793, 2002.
- [21] D. Kagaris, S. Tragoudas, "Von Neumann hybrid cellular automata for generating deterministic test sequences", *ACM Transactions on Design Automation of Electronic Systems*, 6(3), 308-321, 2001.
- [22] K. Bhattacharje, N. Naska, S. Roy, S. Das, "A Survey of Cellular Automata: Types, Dynamics, Non-uniformity and Applications", *ACM Computing Surveys*, 2016.
- [23] T. Toffoli, N. Margolus, **Cellular Automata Machines: A New Environment for Modeling**, Cambridge, MA.: MIT Press, 1987.
- [24] P. C. Fischer, "Generation of primes by a one-dimensional real-time iterative array", *Journal ACM*, 12, 388-394, 1965.
- [25] S. Das, H. Rahaman, B. K. Sikdar., "Cost Optimal Design of Nonlinear CA Based PRPG for Test Applications", **Proc. of Asian Test Symposium**, 2005.
- [26] S. Das, B. Sikdar, "A Scalable Test Structure for Multicore Chip", *IEEE Trans. on CAD of Integrated Circuits and Systems*, 29(1), 127-137, 2010.
- [27] D. Çahkoğlu, **On the computational properties of a class of cellular automata**, The University of Iowa, 1973.
- [28] R. Sommerhalder, S. Westrhenen, "Parallel language recognition in constant time by cellular automata", *Acta Inf.*, 19, 397-407, 1983.
- [29] E. S. Deutsch, "Thinning algorithms on rectangular, hexagonal, and triangular arrays", *Communications of the ACM*, 15, 9, 827-837, 1970.
- [30] E. E. Triendl, "Skeletonization of noisy handdrawn symbols using parallel operations", *Pattern Recognition*, 2(3), 215-226, 1970.
- [31] M. Duff, D. Watson, T. Fountain, G. Shaw, "A cellular logic array for image processing", *Pattern Recognition*, 5(3), 241-247, 1973.
- [32] A. R. Smith, "Two-dimensional formal languages and pattern recognition by cellular automata", **Switching and Automata Theory**, East Lansing, 1971.
- [33] P. Rosin, "Training cellular automata for image processing", *IEEE transactions on image processing*, 15(7), 2076-2087, 2006.
- [34] P. Rosin, "Image processing using 3-state cellular automata", *Computer Vision And Image Understanding*, 114(7), 790-802, 2010.
- [35] V. Vezhnevets, V. Konouchine, "“GrowCut” - Interactive Multi-Label N-D Image Segmentation By Cellular Automata", *proc. of Graphicon*, 1, 150-156, 2005.
- [36] P. Ghosh, S. K. Antani, L. R. Long, G. R. Thoma, "Unsupervised Grow-Cut: Cellular Automata-based Medical Image Segmentation", **Healthcare Informatics, Imaging and Systems Biology (HISB), 2011 First IEEE International Conference on**, San Jose, 2011.
- [37] B. Karasulu, "Görüntülerde İnsan Kulağı Tespit Ve Bölütlemesini Temel Alan Biyometrik Yetkilendirme Üzerine Bir İnceleme", *Bilişim Teknolojileri Dergisi*, 9(2), 97, 2016.
- [38] H. Göker, H. Tekedere, "FATİH Projesine Yönelik Görüşlerin Metin Madenciliği Yöntemleri İle Otomatik Değerlendirilmesi", *Bilişim Teknolojileri Dergisi*, 10(3), 291-299, 2017.
- [39] A. Karacı, M. Erdemir, "Arduino ve Wifi Temelli Çok Sensörlü Robot Tasarımı Ve Denetimi", *Bilişim Teknolojileri Dergisi*, 10(4), 435-449, 2017.
- [40] S. Mukherjee, **Theory and Applications of Nonlinear Cellular Automata In Pattern Recognition**, Shibpur: Bengal Engineering And Science University, 2009.
- [41] S. Peldek, **Face Recognition Using Cellular Automata Technique**, Karabük: Karabuk University, 2012.
- [42] S. Peldek, Y. Becerikli, "GMACA ile Hareket Tespiti Yapılan Video Görüntülerde İnsan Hareketlerinin Tanınması", *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 2018(18-2), 2018.
- [43] S. Peldek, Y. Becerikli, "Use of multiple attractor cellular automata for dimension reduction", **2017 International Conference on Computer Science and Engineering (UBMK)**, Antalya, 2017.
- [44] P. P. Chaudhuri, **Additive cellular automata**, Los Alamitos, Calif [u.a.]: IEEE Computer Society Press, 1997.
- [45] S. Das, B. K. Sikdar, P. P. Chaudhuri, "Characterization of Reachable/Nonreachable Cellular Automata States", **Proceedings of Sixth International Conference on Cellular Automata for Research and Industry**, Netherlands, 2004.
- [46] W. Duch, K. Grudziński, "Ensembles of Similarity-Based Models", **Advances in Intelligent and Soft Computing**, cilt 10, M. Kłopotek , M. Michalewicz ve S. Wierzczoń , Dü., Heidelberg, Physica, Heidelberg, 75-85, 2001.
- [47] N. Naskar, **Theory and applications of nonlinear cellular automata in pattern classification**, Shibpur: Bengal Engineering And Science University, 2009.
- [48] M. R. Bozkurt, N. Yurtay, Z. Yılmaz, C. Sertkaya, "Comparison of different methods for determining diabetes", *Turkish Journal of Electrical Engineering & Computer Sciences*, 22, 1044-1055, 2014.