Journal of the Faculty of Engineering and Architecture of Gazi University 31:4 (2016) 839-857

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

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

Kedi sürüsü optimizasyon algoritmasıyla doğru ve anlaşılabilir nümerik sınıflandırma kurallarının otomatik keşfi

Sinem Akyol[1], Bilal Alataş[2*]

[1]Munzur Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 62000, Tunceli, Türkiye
[2]Fırat Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, 23100, Elazığ, Türkiye

Ö N E Ç I K A N L A R

- Kedi sürüsü optimizasyonu ile veritabanlarında sınıflandırma kurallarının keşfi
- Kuraldaki niteliklerin aralıklarının ön işlem kullanılmadan otomatik bulunması
- Farklı kriterlerin kolaylıkla entegre edilebileceği esnek amaç fonksiyonu

Makale Bilgileri

Geliş: 22.01.2015
Kabul: 08.09.2016

DOI:

10.17341/gummfd.71909

Anahtar Kelimeler:

Kedi sürüsü optimizasyonu, sınıflandırma, veri madenciliği

ÖZET

Metasezgisel algoritmalar, büyük boyutlu optimizasyon problemleri için, kabul edilebilir sürede optimuma yakın çözümler verebilen algoritmalardır. Genel amaçlı metasezgisel optimizasyon algoritmaları; biyoloji tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı, spor tabanlı ve kimya tabanlı olmak üzere yedi farklı grupta değerlendirilmektedir. Sürü zekâsı tabanlı optimizasyon algoritmaları kuş, balık, kedi ve arı gibi canlı sürülerinin hareketlerinin incelenmesiyle geliştirilmiştir. Veri madenciliği, büyük ölçekli verilerden anlamlı ve faydalı bilginin keşfedilmesi işlemidir. Sınıflandırma kurallarının madenciliği üzerinde en çok çalışılan veri madenciliği problemlerinden biridir ve bu problemde veri kümelerinden kullanıcıların rahatça anlayabileceği kurallar çıkarılmaktadır. Bu çalışmada, en güncel sürü zekâsı optimizasyon algoritmalarından Kedi Sürüsü Optimizasyonu (KSO) bildiğimiz kadarıyla ilk kez nümerik ya da karışık tipte verilerden oluşan veritabanlarında sınıflandırma kurallarının otomatik keşfinde kullanılmıştır. Kuralda yer alabilecek ilgili niteliklerin doğru aralıkları için bir ön işlem kullanılmamış, kurallarla birlikte otomatik olarak bulunması da KSO tarafından sağlanmıştır. Ayrıca, kullanılan amaç fonksiyonu çok esnektir ve farklı amaçlar kolaylıkla fonksiyona entegre edilebilir. Bu amaçla, UCI veri ambarından alınan 4 adet veritabanı kullanılmış; doğru ve anlaşılabilir sınıflandırma kuralları keşfedilmiştir. Bulunan sonuçlarla Weka programından elde edilen sonuçlar karşılaştırılmıştır. KSO algoritması üzerinde herhangi bir iyileştirme yapılmadığı ve bu alanda ilk kez kullanıldığı halde, elde edilen sonuçların umut verici olduğu gözlenmiştir.

Automatic mining of accurate and comprehensible numerical classification rules with cat swarm optimization algorithm

H I G H L I G H T S

- Mining of classification rules within databases by cat swarm optimization
- Automatic discovery of attributes ranges in the rule without any preprocess
- Flexible objective function to which different criteria can be easily integrated

Article Info

Received: 22.01.2015
Accepted: 08.09.2016

DOI:

10.17341/gummfd.71909

Keywords:

Cat swarm optimization, classification, data mining

ABSTRACT

Optimization is the process of finding the best solution of a problem. There are many optimization algorithms proposed for optimization problems. The metaheuristic algorithm can give solutions close to optimum in an acceptable period of time for large-scaled optimization problems. The metaheuristic optimization algorithms are evaluated in seven different groups which are biology-based, physics-based, swarm-based, social-based, music-based, sport-based, and chemistry-based. The swarm-based optimization algorithms have been developed by observing the behaviors of creatures e.g. birds, fishes, cats, bees etc. Data mining is the process of discovery of meaningful and useful data within huge databases. Classification rules mining is one of the most commonly studied data mining problems and with methods for this problem, users can easily understand the rules extracted from databases. In this work, one of the most recent swarm based optimization algorithm, Cat Swarm Optimization (CSO), has been firstly used for classification rules mining within databases composed of numerical or mixed data types automatically. There is not any preprocess for finding true ranges for appropriate attributes of the rules, this has been automatically done by CSO. Furthermore, the used objective function is very flexible and many different objectives can easily be added to. For this purpose, four numerical databases obtained from UCI data warehouse have been used and accurate and comprehensible classification rules have been mined. The results have been compared with the results obtained from Weka program. Although CSO has not been embedded with any improvement and has firstly implemented in this research area, the obtained results seem promising.

* Sorumlu Yazar/Corresponding author: balatas@firat.edu.tr / Tel: +90 424 237 0000 - 5599

1. GİRİŞ (INTRODUCTION)

Sınıflandırma kural madenciliği en çok çalışılan veri madenciliği problemlerinden biridir. Sınıflandırma yöntemiyle veri kümelerinden kullanıcıların rahatça anlayabileceği kurallar çıkarılmaktadır. Daha sonra bu kurallar yardımıyla yeni gelen örnek, önceden belirlenmiş bir sınıfa atanmaktadır. Doğru tahmin oranı ve anlaşılabilirlik iki önemli değerlendirme ölçütüdür. Doğru tahmin oranı, oluşturulan modelin daha önceden görülmemiş örnekleri sınıflandırmadaki performansını, anlaşılabilirlik ise kullanıcılar tarafından anlaşılabilirliğini ölçmektedir. Sınıflandırma kural madenciliği için çeşitli yöntem ve algoritmalar bulunmaktadır [1]. Karar ağaçları; C4.5, ID3 ve CART gibi çeşitleri olan, akış şemalarına benzeyen ve "böl-ve-yönet" stratejisini kullanan bir veri sınıflandırma modelidir. Her bir özellik, düğüm tarafından temsil edilmektedir. En üst yapı "kök", en alt yapı "yaprak" ve bu ikisinin arasındaki yapılar da "dal" olarak adlandırılmaktadır. Karar verme işlemine açık bir şekilde hâkim olması ve çok fazla sayıda işlem yapmaya gerek duymadan sınıflandırma işlemini yapması, karar ağaçlarının avantajıdır. Ancak yanlış veya tutarsız örneklere aşırı uymaya neden olabileceğinden, yapılandırma öncesinde analize özen gösterilmelidir. Sürekli değişken yönetimi zorunludur ve değişken önceliğinin belirlenmesini gerektirmektedir [1, 2]. Karar listeleri, eğitim verisinden çıkarılan bilginin açık bir şekilde gösterilmesi yönünden karar ağaçlarına benzemektedir. Karar ağaçlarından farklı olarak "ayır-ve-yönet" stratejisini kullanmaktadırlar. Eğitim verisinin bir alt kümesini kapsaması için bir kural oluşturulmaktadır, sonra kalan örnekleri tekrarlı olarak kapsamak için daha fazla kural üretilmektedir. Karar listelerinin sonunda "Eğer - O Zaman" kuralları bulunmakta ve yeni bir örnek sınıflandırılırken sırayla bu kurallara uygulanmaktadır. Bir örnek, sınıflandırılıncaya kadar, ilk kuraldan başlanarak bütün kurallarda denenmektedir. Eğer örnek sınıflandırılmamışsa, listenin en altındaki varsayılan bir kural işletilmektedir. Dezavantajı, bireysel kuralların anlaşılmasının zor olabilmesidir. Listedeki bir kural, önceki tüm kurallar göz önünde bulunarak ele alınmalıdır. Ayrıca gürültülü eğitim verisi ve ezberleme probleminin giderilmesi amacıyla kural budama işlemi uygulanmaktadır. Evrimsel hesaplama, sınıflandırma kural madenciliğinde önemli yöntemlerden biridir. Genetik algoritma ve genetik programlama en bilinenleridir. Bu yöntemde genel bir arama yapılır ve diğer kaba seçim algoritmalarına göre nitelik etkileşimiyle daha iyi baş edilebilmektedir. Çok değişik tiplerdeki veriler işlenebilmektedir ve sonuçları açıklanabilirdir. Bu yaklaşımın dezavantajları, en iyi sonucun üretildiğinin garanti edilememesi ve işlem yükünün ağır olabilmesidir [1]. Yapay bağışıklık sistemlerinden klonal seçim algoritması, sürü zekâsı tekniklerinden karınca koloni optimizasyon algoritması ve parçacık sürü optimizasyonu algoritması gibi metasezgisel algoritmalar da sınıflandırma kural madenciliğinde kullanılmaktadır. Klasik algoritmaların çalışmadığı farklı veri tiplerinin olduğu veritabanlarında, bu tür metasezgisel algoritmalar iyi sonuçlar verebilmektedir. Bunlara ek olarak sınıflandırmada örnek-tabanlı öğrenme, yapay sinir ağları, lojistik gerileme ve Bayes ağları yaklaşımları da bulunmaktadır. Genel olarak dezavantajları, doğru tahmin oranlarının bazı durumlarda iyi olmasına karşın, açıklayıcı güçlerinin eksikliğidir. Bu yöntemlere bulanık mantık eklenerek, bulanık kurallar oluşturulmaktadır [1]. Kural budama, karar listelerinde, gürültülü eğitim verisinden kaçınmak için kullanılan bir işlemdir. Bu işlem için iki temel strateji vardır. İlk strateji, komple bir kural kümesinin oluşturulmasıdır, daha sonra ise nitelikleri kurallardan eleyerek ya da bireysel kurallar silinerek kural kümesinin basitleştirilmesidir. Bu, önceden tanımlı bazı budama kriterlerine bağlı olarak yapılmaktadır. Diğer strateji ise artımsal budamadır. Her kural, algoritmayla oluşturulduktan hemen sonra basitleştirilmektedir [1]. Son yıllarda sınıflandırma kural madenciliğinde yapılan bazı çalışmalar şunlardır: Gay ve Boulle, Bayes teoremi üzerine kurulmuş sayısal veri kümelerinde sınıflandırma kural madenciliği için yeni bir yapı önermişlerdir [3]. Olmo vd. sınıflandırma kurallarının keşfi için, özgünlüğü, anlaşılırlığı ve duyarlılığı optimize etmeye odaklanan, çok amaçlı bir karınca programlama algoritması önermişlerdir [4]. Changpetch ve Dennis, yeni bir simülasyon tasarımı aracılığıyla, doğru öngörücü modeli yakalamada, sınıflandırma kural madenciliği için ilginçlik ölçüsünün yeteneğini incelemişlerdir [5]. Al-diabat, veri madenciliğinde farklı kural tabanlı sınıflandırma yaklaşımlarını kullanarak, Arapça metin kategorizasyon problemini incelemiştir [6]. Jacques vd. tıbbi verilerde sınıflandırmanın, sınıf dengesizliği, eksik verinin çift anlamlılığı, volümetri veya yüksek yorumlanabilir sonuçlara ihtiyaç gibi çeşitli sorunları ortaya çıkardığını belirtmişlerdir ve bu sorunlar için, çok amaçlı sınıflandırma algoritmasını önermişlerdir [7]. Cano vd. yorumlanabilir ve anlaşılır IF-THEN sınıflandırma kurallarını etkili bir Evrimsel Programlama ile keşfetmişlerdir [8]. Mishra vd. ilişkisel sınıflandırma kural madenciliğini (ilişkisel ve sınıflandırma kural madenciliği tekniklerinin entegre edilmiş hali), çok amaçlı bir problem olarak ele almış ve bunu çözmek için paralel çok amaçlı Genetik Algoritma kullanmışlardır [9]. Salama vd. sürekli nitelik içeren problem alanlarında sınıflandırma kural keşfi için, Arı Kolonisi Optimizasyonu tabanlı cAnt-Miner algoritmasını önermişlerdir [10]. Haltaş vd. metin sınıflandırmada sezgisel arama algoritmalarının kullanılabilecek bir yöntem olduğunu benimsemişlerdir ve bunlardan GA, PSO, Evrimsel Arama ve Tabu Arama'yı kullanarak test etmişlerdir [11]. Köklü vd. sınıflandırma kurallarının keşfi için Yapay Bağışıklık Sistemi algoritmalarından CLONALG kullanmışlardır [12]. Sürü, birbirleriyle etkileşen dağınık yapılı bireyler yığını anlamında kullanılır. Sürülerde N adet temsilci bir amaca yönelik davranışı gerçekleştirmek ve hedefe ulaşmak için birlikte çalışmaktadır. Kolaylıkla gözlenebilen bu "kollektif zekâ" temsilciler arasında sık tekrarlanan davranışlardan doğmaktadır. Temsilciler faaliyetlerini idare etmek için basit bireysel kurallar kullanmakta ve grubun kalan

kısmıyla etkileşim yolu ile sürü amaçlarına ulaşmaktadır. Grup faaliyetlerinin toplamından bir çeşit kendini örgütleme doğmaktadır. Tek başına bir şey yapamayan tekil bireyler, sürüler halinde zeki davranışlar sergileyebilmektedir. Bu amaçla metasezgisel optimizasyon algoritmaları içerisinde sürü zekası tabanlı algoritmalar önerilmiştir. Kedi Sürüsü Optimizasyonu (KSO), kedigillerin hareketlerinin incelenmesi sonucu önerilmiş güncel bir sürü zekası tabanlı optimizasyon algoritmasıdır. Bu çalışmada KSO, ilk kez nümerik ya da karmaşık tipte verilerden oluşan veritabanlarında otomatik sınıflandırma kurallarının keşfinde kullanılmıştır. Bu amaçla, UCI veri ambarından alınan 4 adet veritabanı kullanılarak doğru ve anlaşılabilir sınıflandırma kuralları keşfedilmiştir. Bulunan sonuçlarla Weka programından elde edilen sonuçlar karşılaştırılmıştır. Diğer yöntemlerde, nümerik veriler üzerinde yapılması gereken ön işlemlere, bu çalışmada ihtiyaç duyulmamıştır. Bulunan kurallardaki niteliklerin değer aralıkları, kurallarla eşzamanlı olarak bulunmuştur. Böylece veri (ilgilenilen problem uzayı) üzerinde değişiklik yapılmamıştır. Ayrıca, KSO ile kural budama işlemine de gerek kalmamıştır. Nitelikler arası etkileşim problemi de KSO ile giderilmiştir. Değişik noktalarda farklı ilişkiler sergileyen; farklı tipte, büyük boyutlu ve çok sayıda verilerden oluşan, ayrıca hata ve eksik veri içeren veritabanlarında sınıflandırma kural madenciliği için KSO'nun etkili bir algoritma olduğu gösterilmiştir.

2. KEDİ SÜRÜSÜ OPTİMİZASYONU (CAT SWARM OPTIMIZATION)

KSO, kedigillerin hareketlerinin incelenmesiyle ortaya çıkarılmıştır. Kedilerin davranışının benzetimi yapılarak iki alt model oluşturulmuştur. Bu optimizasyondaki matematiksel modeller, kedilerin hareketlerinin analiziyle ortaya çıkarılmıştır [13]. KSO kullanılarak yapılan bazı çalışmalar şunlardır: Santosa vd. kümeleme problemi için KSO'yu kullanmışlardır [14]. Wang vd. en az önemli bitin yerine en iyisini getirmek için KSO stratejisi kullanmışlardır [15]. Hwang vd. müşterilere göre en uygun sözleşme kapasitesi problemini çözmek için KSO ve PSO'yu birlikte kullanmışlardır [16]. Destek vektör makinesi için özellik seçimi ve parametre optimizasyonu probleminde KSO, Lin ve Chen tarafından önerilmiştir [17]. Kalaiselvan vd. filigran performansının arttırılması amacıyla KSO uygulamışlardır [18]. Wang ve Wu e-öğrenmede duygu tanıma problemi için KSO'yu destek vektör makinesiyle birlikte kullanmışlardır [19]. KSO algoritmasının paralel versiyonu da, Tsai vd. tarafından önerilmiştir [20]. Pradhan ve Panda, çok amaçlı optimizasyon problemlerini çözmek için, mevcut KSO algoritmasını genişleterek yeni çok amaçlı bir evrimsel algoritma önermişlerdir [21]. Wang vd. KSO'yu, stego-görüntü kalitesi probleminin optimum veya optimuma yakın çözümünü elde etmek için uyarlamışlardır [22]. Naresh vd. çoklu makine güç sisteminde, çoklu güç sistem dengeleyici tasarımı için KSO'yu kullanmışlardır [23]. Kong vd. optimum yönlendirme yolunu önceden tespit edebilmek için, etkili ve paralel KSO tabanlı yeni bir şema önermişlerdir [24]. Saha vd. optimum lineer FIR filtre tasarımı için KSO algoritmasını uyarlamışlardır [25]. Kumar vd. KSO arama algoritmasını kullanarak, büyük ölçekli birincil dağıtım ağlarında, yüksek genel sistem güvenilirliğini sağlamak için, optimum yerleşimi sunmuşlardır ve çoklu dağıtılan jeneratörleri boyutlandırmışlardır [26]. Majumder ve Eldho, KSO ve Ters Parçacık İzleme ile Analitik Eleman Metodunu kombine ederek yeni bir yeraltı yönetim modeli önermişlerdir [27]. Meziane vd. hibrid güç sistemi tasarım optimizasyon problemini çözmek için KSO'yu kullanmışlardır [28]. Dong vd. üstel atalet katsayısı ve mutasyon özellikleri eklenerek geliştirilmiş KSO'yu kullanarak, sulama alanlarında düşük su kaybı ile kanal kesiti tasarlamayı amaçlamışlardır [29]. Lin vd. büyük veriler için bir metin sınıflandırma deneyinde özellikleri seçmek için geliştirilmiş KSO'yu uygulamışlardır [30]. Yang vd. sert olmayan çok modelli görüntü kaydı için KSO ile L-BFGS-B'nin birleşimi olan yeni bir optimizasyon metodunu önermişlerdir [31]. Guo vd. güneş pillerinin parametre tanımlama ve duyarlılık analizi için KSO algoritmasını kullanmışlardır [32]. Mohapatra vd. gen seçim sistemine dayalı geliştirilmiş KSO'yu, çekirdek ridge regresyonu ile mikrodizi tıbbi veri sınıflandırmada kullanmışlardır [33].

2.1. Kedigillerin Hareketleri (Behaviours of Cats)

Biyolojik sınıflandırmaya göre, yaklaşık 30 farklı kedi cinsi (örneğin aslan, leopar, kaplan, vb.) bulunmaktadır. Çoğunun farklı yaşam alanı olmasına rağmen, kedigiller benzer davranış modellerini sergilemektedir. Kedilerin avlanma becerisi kediler için kalıtsal değildir, alıştırmalar aracılığıyla kazanılmaktadır. Bu avlanma becerisi ile yaban kedileri yiyeceklerini temin etmeyi sağlamaktadır ve türlerinin hayatta kalması garanti altına alınmaktadır. Ayrıca evcil kediler de benzer doğal avlanma becerisi ve hareketli nesnelere güçlü bir merak sergilemektedir. Bütün kediler, bu güçlü merakı paylaşmasına rağmen, zamanlarının çoğunu hareketsiz (durağan) geçirmektedir. Kediler çok yüksek seviyede atikliğe sahiptir. Bu atiklik; dinlenme zamanlarında bile kendilerini bırakmamakta, büyük geniş gözler sürekli olarak etrafı gözetlemektedir. Kediler, çok zeki ve bilinçli (planlı) yaratıklar oldukları halde tembel gibi görünmektedir. Kedilerin dokuz canlı olduğu söylenerek, kedilerin güçlü canlılığına gönderme yapılmaktadır. Ev içinde olan kedi sık sık alçak frekansta ses çıkarmaktadır. Kediler hoşnut oldukları, tehlikede veya hasta oldukları zaman mırlar. Mırlamanın alçak frekansının, hücre onarımına yardım ettiğine ve bunun kedileri daha güçlü ve canlı yaptığına inanılmaktadır [34].

2.2. Algoritmalar (Algorithms)

KSO'da, kedilerin başlıca iki tane davranışsal özelliği modellenmiştir. Bunlar "arama modu" ve "izleme modu" olarak isimlendirilmiştir. Bu iki modun birleşimi, KSO'nun daha iyi bir performans göstermesine yardım etmektedir [34].

2.2.1. Çözüm kümesinin sunumu (Presentation of solution set)

KSO'da önerilen algoritmada, optimizasyon problemini çözmek için kediler ve kedilerin davranışlarının modeli kullanılmaktadır; örneğin çözüm kümesini tasvir etmek için kediler kullanılmaktadır. KSO'da ilk önce iterasyonda kaç kedinin kullanılacağına karar verilmektedir; daha sonra optimizasyon problemini çözmek için, aday çözümlere karşılık gelen kedilere, KSO sürecinin adımları uygulanmaktadır. Bütün kediler *M* boyuttan oluşan kendi pozisyonlarına, her bir boyut için hızlara, kedinin uyumunu uygunluk fonksiyonuna yansıtan bir uygunluk değerine ve kedinin izleme modunda mı yoksa arama modunda mı olduğunu belirlemek için bir bayrağa sahiptir. Final çözüm en iyi pozisyona sahip olan kedidir. KSO en iyi çözümü iterasyon sonuna ulaşıncaya kadar saklayacaktır [34].

2.2.2. Dinlenme ve tetikte olma - arama modu (Seeking mode)

Bu alt mod dinlenmede fakat tetikte olmanın bir periyodu boyunca kedinin modellenmesi için kullanılmaktadır (sonraki hareketi için çevresine bakınma). Arama modu aşağıda belirtildiği gibi dört gerekli faktöre sahiptir: arama hafızası havuzu (*AHH*), seçilen boyutun arama aralığı (*SBA*), değişen boyutların sayısı (*DBS*) ve kendi pozisyonunu değerlendirme (*KPD*). *AHH* her bir kedinin arama hafızasının boyutunu tanımlamak için kullanılır, bazı noktaları kediye göre sıralayarak belirtir. Daha sonra anlatılacak kurallara göre kedi, hafıza havuzundan bir nokta ayıracaktır. *SBA* seçilen boyutlar için mutasyon oranını temsil eder. Arama modu süresince, eğer bir boyut mutasyon için seçilmişse, yeni ve eski değerler arasındaki farklılık aralık dışında olmamalıdır; aralık *SBA* tarafından tanımlanır. *DBS* boyutlardan kaç tanesinin değişime uğrayacağını ifade eder. Bütün bu faktörler arama modunda önemli rol oynar. *KPD* bilinen bir ikili değerdir ve kedilerin bulunduğu noktanın hareket için aday noktalardan biri olup olmayacağını gösterir. *KPD*, *AHH* değerini etkilemez [34]. Arama modunun adımları Şekil 1'de verilmiştir.

$$P_i = \frac{|UD_i - UD_b|}{|UD_{maks} - UD_{min}|},\ 0 < i < j \text{ olduğunda} \quad (1)$$

Eğer uygunluk fonksiyonunun amacı minimum çözümü bulmaksa, $UD_b=UD_{maks}$, aksi takdirde $UD_b=UD_{min}$ olur [34, 14, 15].

2.2.3. Hareket- izleme modu (Tracing mode)

İzleme modu, kedinin hedefi izlemedeki durumunu modellemek için bir alt modeldir. Bir kere kedi izleme moduna girdiğinde, her bir boyutu için kendi hızlarına göre hareket etmektedir. İzleme modundaki çalışma, Şekil 2'de açıklanmıştır [34].

$$v_{k,d} = v_{k,d} + r_1 * c_1\left(x_{eniyi,d} - \mathrm{x_{k,d}}\right), d = 1,2,\dots,M \quad (2)$$

$X_{eniyi,d}$, en iyi uygunluk değerine sahip kedinin pozisyonu; $x_{k,d}$, $kedi_k$'nin pozisyonu, c_1 bir sabit ve r_1 rassal bir değerdir [34, 14, 15].

$$x_{k,d} = x_{k,d} + v_{k,d} \quad (3)$$

Adım 1: *j=AHH* olduğunda $kedi_k$'nin bulunduğu pozisyonda *j* tane kopya yap. Eğer *KPD* değeri doğruysa, *j=(AHH-1)* olur, sonra mevcut pozisyonu, adaylardan biri olarak tut.
Adım 2: *DBS*'ye göre, her bir kopya için mevcut değerin *SBA* yüzdesini gelişigüzel olarak arttır veya azalt ve eskisiyle yerini değiştir.
Adım 3: Bütün aday noktaların uygunluk değerini hesapla.
Adım 4: Eğer bütün uygunluk değerleri (*UD*) tam olarak aynı değilse, eşitliğe göre her bir aday noktanın seçilme olasılığını Eş. 1'e göre hesapla, aksi takdirde her bir aday noktanın seçilme olasılıklarının tümüne 1 ata.
Adım 5: Aday noktalardan farklı noktalara gidebilmek için, noktaları gelişigüzel çıkart (ayır), ve $kedi_k$'nin pozisyonuyla değiştir.

Şekil 1. Arama modu adımları (Steps of seeking mode)

Adım 1: Bütün boyutlar için hızları ($v_{k,d}$) Eş. 2'yi kullanarak güncelle.
Adım 2: Hızların, maksimum hız aralığında olduğunu kontrol et. Yeni hız aralığın dışındaysa, bu değeri ilgili limite eşitle (*limit=sınır*).
Adım 3: $kedi_k$'nin pozisyonunu Eş. 3'ü kullanarak güncelle.

Şekil 2. İzleme modu adımları (Steps of tracing mode)

Adım 1: Süreçte *N* tane kedi oluştur.
Adım 2: *M* boyutlu çözüm uzayına gelişigüzel kediler serpiştir ve her kedinin hızına maksimum hız aralığında olan gelişigüzel değerler ver. Sonra kedileri *KO*'ya göre izleme modu ya da arama moduna sok.
Adım 3: Amaç kriterini yansıtan uygunluk fonksiyonunu kedilerin pozisyonuna göre belirle ve en iyi kediyi hafızada sakla. Şimdiye kadarki en iyi çözümü yansıtmasından dolayı sadece en iyi kedinin pozisyonu saklanır.
Adım 4: Bayraklarına göre kedileri hareket ettir, eğer $kedi_k$ arama modundaysa, arama modu sürecine uygula, aksi takdirde izleme modu sürecine uygula.
Adım 5: Kedileri yeniden *KO*'ya göre izleme modu ya da arama moduna sok.
Adım 6: Sonlandırma (bitirme) koşulları sağlanmışsa programı sonlandır, aksi durumda Adım 3'ten Adım 5'e kadar tekrar et.

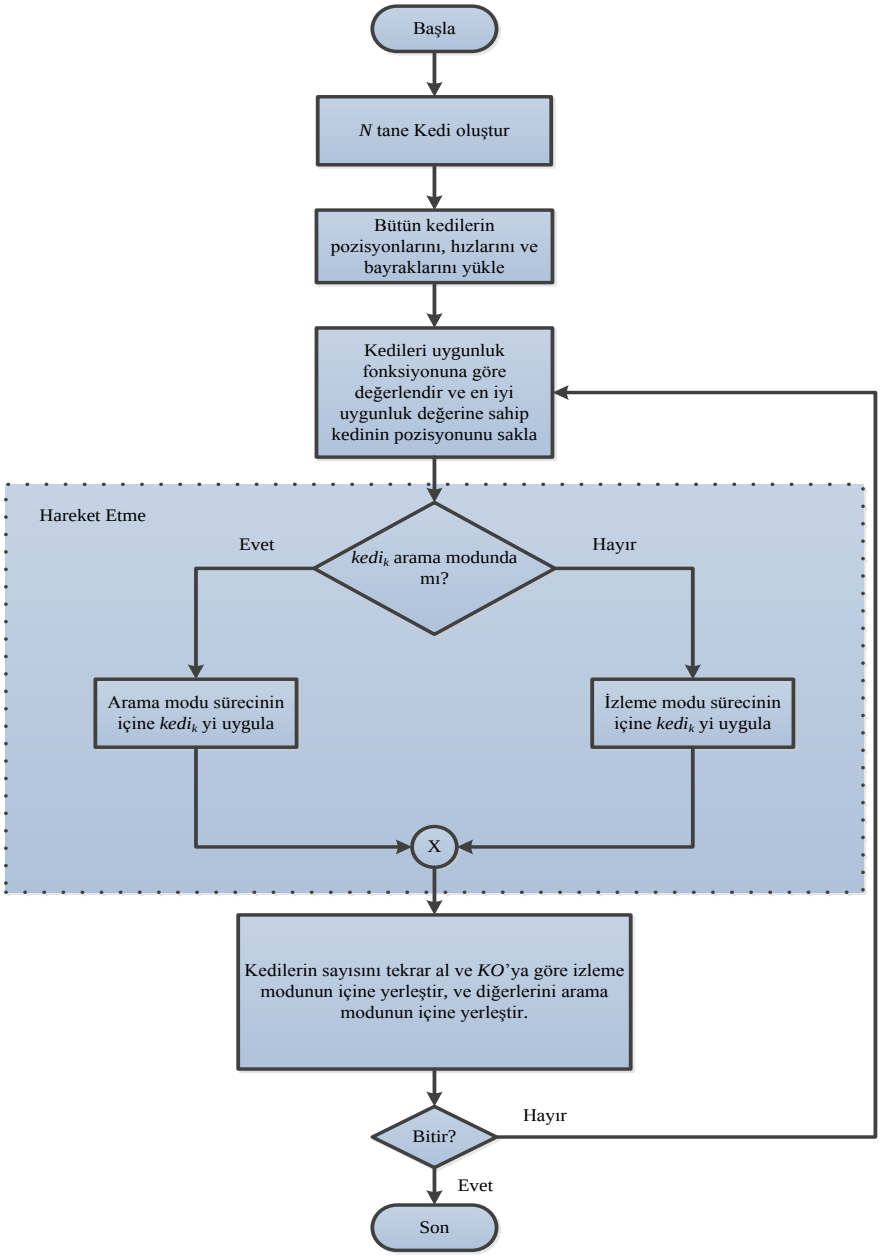
Şekil 3. KSO süreci (Process of CSO)

2.3. Kedi Sürüsü Optimizasyon Algoritması
(Cat Swarm Optimization)

KSO'nun, arama modu ve izleme modu adında iki alt modu vardır. Bu iki modu algoritma şeklinde birleştirmek için, arama moduyla izleme modunu birleştirmeyi sağlayan bir karışım oranı (*KO*) tanımlanmaktadır. Kediler dinlenme zamanında hareket etmeye karar verdiklerinde, hareket çok dikkatli ve yavaşça yapılmaktadır. Bu hareket, arama moduna yansıtılmaktadır. İzleme modu kedi tarafından bir hedefin takip edilmesini modellemektedir. Kediler, enerji kaynaklarını fazla kullanmalarına yol açan objeleri takip etmeye çok az zaman harcamaktadır. Kedilerin zamanlarının çoğunu dinlenmeye ve gözetlemeye (mesela zamanlarının çoğu arama modunda geçmektedir) harcadığını garantilemek için *KO*'ya çok küçük bir değer atanmaktadır. KSO'nun süreci Şekil 3'teki sözde kodda açıklanmıştır [34, 14, 15]. KSO sürecinin diyagramı Şekil 4'te sunulmaktadır.

3. KEDİ SÜRÜSÜ OPTİMİZASYON ALGORİTMASIYLA SINIFLANDIRMA KURALLARININ KEŞFİ
(MINING OF CLASSIFICATION RULES WITH CAT SWARM OPTIMIZATION ALGORITHM)

Bu çalışmada, KSO algoritmasının çalışma mantığına göre Visual C# dili kullanılarak bir program kodlanmıştır. Kodlanan bu programda UCI veri ambarından alınan Ecoli [35], Diabetes [36], Liver Disorders (Bupa) [37] ve New



Şekil 4. KSO'nun süreç diyagramı (Process diagram of CSO)

Thyroid [38] veritabanları kullanılarak kurallar otomatik olarak keşfedilmiştir. Bu veritabanlarının seçiminde; halka açık olması, sınıflandırma için kullanılabilmesi, niteliklerinin nümerik verilerden oluşması ve eksik veri içermemesi gibi özellikler dikkate alınmıştır. Keşfedilen kurallar ise Weka programındaki algoritmaların bulduğu kurallar ile karşılaştırılmıştır. Bu nedenle, öncelikle Weka programındaki bu algoritmalar ile UCI veri ambarından alınan veri tabanları hakkında kısa bilgiler verilecek; daha sonra ise doğru ve anlaşılabilir nümerik sınıflandırma kurallarının otomatik olarak bulunabilmesi amacıyla, KSO algoritması için yapılan düzenlemeler ve uyarlamalar açıklanacaktır.

3.1. Weka (Weka)

Weka (Waikato Environment for Knowledge Analysis), Waikato Üniversitesi tarafından Java programlama dili ile gerçekleştirilmiş, açık kaynak kodlu bir programdır. Weka veri madenciliği görevleri için makine öğrenme algoritmalarının bir koleksiyonudur. Algoritmalar doğrudan bir veritabanına uygulanabilmektedir ya da Java kodundan da çağrılabilmektedir. Veri ön işleme, sınıflandırma, regresyon, kümeleme, birliktelik kuralları ve görselleştirme için araçlar içermektedir. Bu çalışmada elde edilen sonuçlar Weka 3.6 sürümünde bulunan sınıflandırma algoritmalarından; hem nümerik verilerde çalışan, hem de çıktı olarak sınıflandırma kuralları kümesini de verebilen 4 algoritma olan One-R, Ridor, JRip ve Part'tan elde edilen sonuçlarla karşılaştırılmıştır [39].

3.1.1. One-R algoritması (One-R algorithm)

One-R veya "One Rule (Bir Kural)" R. C. Holt tarafından önerilen basit bir algoritmadır [40]. Bu algoritma eğitim verilerinde her bir özellik için bir kural üretmektedir ve sonra One-R'sine göre en küçük hata oranına sahip kural seçilmektedir [41].

3.1.2. Part algoritması (Part algorithm)

I. H. Witten ve E. Frank tarafından önerilen, böl ve yönet mantığına göre çalışan bir algoritmadır [42]. Part Algoritması, kurallar kümesini sıralayan ve karar listeleri olarak adlandırılan kümeler üretmektedir. Bu algoritma her bir çalıştırmada kısmi bir C4.5 karar ağacı oluşturmaktadır ve en iyi kural bir yaprağın içinde olmaktadır [41].

3.1.3. Ridor algoritması (Ridor algorithm)

Ridor, ilk olarak varsayılan kuralı, sonra ise varsayılan kural için (ağırlıklı) hata oranı en az olan istisnayı oluşturmaktadır. Daha sonra her bir istisna için en iyi istisnaları üretmektedir ve saf (kusursuz) olana kadar bu işlemi tekrarlamaktadır. Böylece istisnaların genişlemesi gibi bir ağaç sunmaktadır ve yaprakta sadece varsayılan kural olmaktadır, istisnalar olmamaktadır. İstisnalar varsayılan kuraldaki sınıftan başka bir sınıfı öngören kurallar kümesidir [41].

3.1.4. Jrip algoritması (Jrip algorithm)

W. W. Cohen tarafından önerilen ve iteratif şekilde artan budama ile hata azaltımı yapan bir algoritmadır [43]. Bu algoritma bütün pozitif örnekler kapsanıncaya kadar boş bir kural kümesine kuralları art arda ekleyerek bir kural kümesi oluşturmaktadır. Kurallar, hiçbir negatif örnek kapsanmadığı sürece, bir kuralın öncülüne (boş bir öncülle başlamaktadır), koşulların açgözlülükle eklenmesiyle oluşturulmaktadır. Bir kural kümesi oluşturulduktan sonra uygulanacak optimizasyonla, kural kümesinin eğitim verileriyle olan uyumu arttırılmaktadır ve boyut azaltılarak iyileştirme sağlanmaktadır [41].

3.2. Kullanılan Veritabanları (Used Databases)

Bu çalışmada, KSO algoritmasının çalışma mantığına göre Visual C# dili kullanılarak bir program kodlanmıştır. Kodlanan bu programda, UCI veri ambarından alınan ve nümerik verilerden oluşan Ecoli [35], Diabetes [36], Liver Disorders (Bupa) [37] ve New Thyroid [38] veritabanları kullanılarak, otomatik şekilde doğru ve anlaşılabilir kurallar üretilmiştir.

3.2.1. Ecoli veritabanı (Ecoli database)

Ecoli veritabanında 336 tane örnek bulunmaktadır ve bu örnekler 8 adet sınıfa ayrılmıştır. Bu 8 adet sınıf ve sınıflardaki örnek sayısı Tablo 1'de gösterilmektedir. Bu veritabanında 7 tane nitelik bulunmaktadır. Bu nitelikler ile aldıkları değer aralıkları ise Tablo 2'de gösterilmektedir. Bu tablodan görüldüğü gibi, bütün nitelikler [0, 1] arasında reel değerler almaktadır. Ayrıca bu veritabanının temel istatistiksel göstergeleri Tablo 3'te sunulmuştur.

Tablo 1. Ecoli veritabanındaki sınıflar
(Classes in Ecoli database)

Sınıflar	Bu Sınıftaki Örnek Sayısı
cp	143
im	77
pp	52
imU	35
om	20
omL	5
imL	2
imS	2

Tablo 2. Ecoli veritabanındaki nitelikler ve aldıkları değer aralıkları
(Attributes in Ecoli database and their value intervals)

Nitelikler	En Düşük Değer	En Yüksek Değer
mcg	0	0,89
gvh	0,16	1
lip	0,48	1
chg	0,5	1
aac	0	0,88
alm1	0,03	1
alm2	0	0,99

Tablo 3. Ecoli veritabanının temel istatistiksel göstergeleri
(Basic statistical indicators of Ecoli database)

Nitelik karakteristiği	Nümerik
Örnek Sayısı	336
Nitelik Sayısı	8
Kayıp Değer Sayısı	0
Alan	Biyoenformatik

Ecoli veritabanından bir kesit, Şekil 5'te gösterilmektedir.

No	mcg	gvh	lip	chg	aac	alm1	alm2	Sınıf
1	0,49	0,29	0,48	0,5	0,56	0,24	0,35	cp
2	0,07	0,4	0,48	0,5	0,54	0,35	0,44	cp
3	0,56	0,4	0,48	0,5	0,49	0,37	0,46	cp
4	0,59	0,49	0,48	0,5	0,52	0,45	0,36	cp
5	0,23	0,32	0,48	0,5	0,55	0,25	0,35	cp
6	0,67	0,39	0,48	0,5	0,36	0,38	0,46	cp
7	0,29	0,28	0,48	0,5	0,44	0,23	0,34	cp
8	0,21	0,34	0,48	0,5	0,51	0,28	0,39	cp
9	0,2	0,44	0,48	0,5	0,46	0,51	0,57	cp
10	0,42	0,4	0,48	0,5	0,56	0,18	0,3	cp
11	0,42	0,24	0,48	0,5	0,57	0,27	0,37	cp
12	0,25	0,48	0,48	0,5	0,44	0,17	0,29	cp
13	0,39	0,32	0,48	0,5	0,46	0,24	0,35	cp
14	0,51	0,5	0,48	0,5	0,46	0,32	0,35	cp
15	0,22	0,43	0,48	0,5	0,48	0,16	0,28	cp
16	0,25	0,4	0,48	0,5	0,46	0,44	0,52	cp
17	0,34	0,45	0,48	0,5	0,38	0,24	0,35	cp
18	0,44	0,27	0,48	0,5	0,55	0,52	0,58	cp
19	0,23	0,4	0,48	0,5	0,39	0,28	0,38	cp
20	0,41	0,57	0,48	0,5	0,39	0,21	0,32	cp
21	0,4	0,45	0,48	0,5	0,38	0,22	0,0	cp
22	0,31	0,23	0,48	0,5	0,73	0,05	0,14	cp
23	0,51	0,54	0,48	0,5	0,41	0,34	0,43	cp
24	0,3	0,16	0,48	0,5	0,56	0,11	0,23	cp

Şekil 5. Ecoli veritabanından bir görüntü
(A section from Ecoli database)

Ecoli veritabanında cp, im, pp, imU, om ve omL sınıfları için kurallar bulunmuştur. imL ve imS sınıflarında örnek sayısı az olduğu için kural bulunmamıştır.

3.2.2. Pima indians diabetes veritabanı (Pima indians database)

Diabetes veritabanında 768 tane örnek bulunmaktadır ve sınıf etiketi 2 adet değer almaktadır. Bu 2 adet sınıf ve sınıflardaki örnek sayısı Tablo 4'te gösterilmektedir. Bu veritabanında 8 tane nitelik bulunmaktadır. Bu nitelikler ile aldıkları değer aralıkları ise Tablo 5'te gösterilmektedir. Bütün nitelikler tamsayı veya reel değerlerden oluşmaktadır. Bu veritabanının temel istatistiksel göstergeleri Tablo 6'da sunulmuştur.

Tablo 4. Diabetes veritabanındaki sınıflar
(Classes in Diabetes database)

Sınıflar	Bu Sınıftaki Örnek Sayısı
tested_negative	500
tested_positive	268

Tablo 5. Diabetes veritabanındaki nitelikler ve aldıkları değer aralıkları
(Attributes in Diabetes database and their value intervals)

Nitelikler	En Düşük Değer	En Yüksek Değer
preg	0	17
plas	0	199
pres	0	122
skin	0	99
insu	0	846
mass	0	67,1
pedi	0,078	2,42
age	21	81

Tablo 6. Diabetes veritabanının temel istatistiksel göstergeleri
(Basic statistical indicators of Diabetes database)

Nitelik karakteristiği	Nümerik
Örnek Sayısı	768
Nitelik Sayısı	8
Kayıp Değer Sayısı	0
Alan	Sağlık

Pima Indians Diabetes veritabanının örnek bir görüntüsü Şekil 6'da gösterilmektedir. Pima Indians Diabetes veritabanında sadece "tested-negative" sınıfı için kural bulunmuştur. Bulunan kuralların dışında kalan örnekler "tested-positive" sınıfı olarak düşünülmüştür.

3.2.3. Liver disorders (bupa) veritabanı
(Liver disorders (bupa) database)

Liver Disorders (Bupa) veritabanında 345 tane örnek bulunmaktadır ve bu örnekler 2 sınıfa ayrılmıştır. Bu 2 adet sınıf ve sınıflardaki örnek sayısı Tablo 7'de gösterilmektedir.

Tablo 7. Bupa veritabanındaki sınıflar
(Classes in Bupa database)

Sınıflar	Bu Sınıftaki Örnek Sayısı
1	145
2	200

Bu veritabanında 6 tane nitelik bulunmaktadır. Bu nitelikler ile aldıkları değer aralıkları ise Tablo 8'de gösterilmektedir. Bütün nitelikler tamsayı veya reel değerler almaktadır. Liver Disorders (Bupa) veritabanının örnek bir kesiti Şekil 7'de gösterilmektedir. Tablo 9'da bu veritabanının temel istatistiksel göstergeleri sunulmuştur.

No	preg	plas	pres	skin	insu	mass	pedi	age	Sınıf
1	6,0	148,0	72,0	35,0	0,0	33,6	0,627	50,0	tested_positive
2	1,0	85,0	66,0	29,0	0,0	26,6	0,351	31,0	tested_negative
3	8,0	183,0	64,0	0,0	0,0	23,3	0,672	32,0	tested_positive
4	1,0	89,0	66,0	23,0	94,0	28,1	0,167	21,0	tested_negative
5	0,0	137,0	40,0	35,0	168,0	43,1	2,288	33,0	tested_positive
6	5,0	116,0	74,0	0,0	0,0	25,6	0,201	30,0	tested_negative
7	3,0	78,0	50,0	32,0	88,0	31,0	0,248	26,0	tested_positive
8	10,0	115,0	0,0	0,0	0,0	35,3	0,134	29,0	tested_negative
9	2,0	197,0	70,0	45,0	543,0	30,5	0,158	53,0	tested_positive
10	8,0	125,0	96,0	0,0	0,0	0,0	0,232	54,0	tested_positive
11	4,0	110,0	92,0	0,0	0,0	37,6	0,191	30,0	tested_negative
12	10,0	168,0	74,0	0,0	0,0	38,0	0,537	34,0	tested_positive
13	10,0	139,0	80,0	0,0	0,0	27,1	1,441	57,0	tested_negative
14	1,0	189,0	60,0	23,0	846,0	30,1	0,398	59,0	tested_positive
15	5,0	166,0	72,0	19,0	175,0	25,8	0,587	51,0	tested_positive
16	7,0	100,0	0,0	0,0	0,0	30,0	0,484	32,0	tested_positive
17	0,0	118,0	84,0	47,0	230,0	45,8	0,551	31,0	tested_positive
18	7,0	107,0	74,0	0,0	0,0	29,6	0,254	31,0	tested_positive
19	1,0	103,0	30,0	38,0	83,0	43,3	0,183	33,0	tested_negative
20	1,0	115,0	70,0	30,0	96,0	34,6	0,529	32,0	tested_positive
21	3,0	126,0	88,0	41,0	235,0	39,3	0,704	27,0	tested_negative
22	8,0	99,0	84,0	0,0	0,0	35,4	0,388	50,0	tested_negative
23	7,0	196,0	90,0	0,0	0,0	39,8	0,451	41,0	tested_positive

Şekil 6. Pima Indians Diabetes veritabanından bir görüntü (A section from Diabetes database)

No	mcv	alkphos	sgpt	sgot	gammagt	drinks	Sınıf
1	85,0	92,0	45,0	27,0	31,0	0,0	1
2	85,0	64,0	59,0	32,0	23,0	0,0	2
3	86,0	54,0	33,0	16,0	54,0	0,0	2
4	91,0	78,0	34,0	24,0	36,0	0,0	2
5	87,0	70,0	12,0	28,0	10,0	0,0	2
6	98,0	55,0	13,0	17,0	17,0	0,0	2
7	88,0	62,0	20,0	17,0	9,0	0,5	1
8	88,0	67,0	21,0	11,0	11,0	0,5	1
9	92,0	54,0	22,0	20,0	7,0	0,5	1
10	90,0	60,0	25,0	19,0	5,0	0,5	1
11	89,0	52,0	13,0	24,0	15,0	0,5	1
12	82,0	62,0	17,0	17,0	15,0	0,5	1
13	90,0	64,0	61,0	32,0	13,0	0,5	1
14	86,0	77,0	25,0	19,0	18,0	0,5	1
15	96,0	67,0	29,0	20,0	11,0	0,5	1
16	91,0	78,0	20,0	31,0	18,0	0,5	1
17	89,0	67,0	23,0	16,0	10,0	0,5	1
18	89,0	79,0	17,0	17,0	16,0	0,5	1
19	91,0	107,0	20,0	20,0	56,0	0,5	1
20	94,0	116,0	11,0	33,0	11,0	0,5	1
21	92,0	59,0	35,0	13,0	19,0	0,5	1
22	93,0	23,0	35,0	20,0	20,0	0,5	1
23	90,0	60,0	23,0	27,0	5,0	0,5	1
24	96,0	68,0	18,0	19,0	19,0	0,5	1

Şekil 7. Liver Disorders (Bupa) veritabanından bir görüntü (A section from Bupa database)

Tablo 8. Bupa veritabanındaki nitelikler ve aldıkları değer aralıkları
(Attributes in Bupa database and their value intervals)

Nitelikler	En Düşük Değer	En Yüksek Değer
mcv	65	103
alkphos	23	138
sgpt	4	155
sgot	5	82
gammagt	5	297
drinks	0	20

Tablo 9. Bupa veritabanının temel istatistiksel göstergeleri
(Basic statistical indicators of Bupa database)

Nitelik karakteristiği	Nümerik
Örnek Sayısı	345
Nitelik Sayısı	7
Kayıp Değer Sayısı	0
Alan	Sağlık

3.2.4. Thyroid disease (new thyroid) veritabanı
(Thyroid disease (new thyroid) database)

New Thyroid veritabanında 215 tane örnek bulunmaktadır ve bu örnekler 3 adet sınıfa ayrılmıştır. Bu 3 adet sınıf ve sınıflardaki örnek sayısı Tablo 10'da gösterilmektedir. Bu veritabanında 5 tane nitelik bulunmaktadır. Bu nitelikler ile aldıkları değer aralıkları ise Tablo 11'de gösterilmektedir. Bütün nitelikler tamsayı veya reel değerler almaktadır.

No	T3resin	Thyroxin	Triiodothyronine	Thyroidstimulating	TSH_value	Sınıf
1	107,0	10,1	2,2	0,9	2,7	1
2	113,0	9,9	3,1	2,0	5,9	1
3	127,0	12,9	2,4	1,4	0,6	1
4	109,0	5,3	1,6	1,4	1,5	1
5	105,0	7,3	1,5	1,5	-0,1	1
6	105,0	6,1	2,1	1,4	7,0	1
7	110,0	10,4	1,6	1,6	2,7	1
8	114,0	9,9	2,4	1,5	5,7	1
9	106,0	9,4	2,2	1,5	0,0	1
10	107,0	13,0	1,1	0,9	3,1	1
11	106,0	4,2	1,2	1,6	1,4	1
12	110,0	11,3	2,3	0,9	3,3	1
13	116,0	9,2	2,7	1,0	4,2	1
14	112,0	8,1	1,9	3,7	2,0	1
15	122,0	9,7	1,6	0,9	2,2	1
16	109,0	8,4	2,1	1,1	3,6	1
17	111,0	8,4	1,5	0,8	1,2	1
18	114,0	6,7	1,5	1,0	3,5	1
19	119,0	10,6	2,1	1,3	1,1	1
20	115,0	7,1	1,3	1,3	2,0	1
21	101,0	7,8	1,2	1,0	1,7	1
22	103,0	10,1	1,3	0,7	0,1	1
23	109,0	10,4	1,9	0,4	-0,1	1
24	102,0	7,6	1,8	2,0	2,5	1

Şekil 8. New Thyroid veritabanından bir görüntü (A section from New Thyroid database)

New Thyroid veritabanından örnek bir kesit Şekil 8'de gösterilmektedir. Tablo 12'de bu veritabanına ait temel istatistiksel göstergeler sunulmuştur.

Tablo 10. New Thyroid veritabanındaki sınıflar
(Classes in New Thyroid database)

Sınıflar	Bu Sınıftaki Örnek Sayısı
1	150
2	35
3	30

Tablo 11. New Thyroid veritabanındaki nitelikler ve aldıkları değer aralıkları
(Attributes in New Thyroid database and their value intervals)

Nitelikler	En Düşük Değer	En Yüksek Değer
T3resin	65	144
Thyroxin	0,5	25,3
Triiodothyronine	0,2	10
Thyroidstimulating	0,1	56,4
Tsh_value	-0,7	56,3

Tablo 12. New Thyroid veritabanının temel istatistiksel göstergeleri (Basic statistical indicators of New Thyroid database)

Nitelik karakteristiği	Nümerik
Örnek Sayısı	215
Nitelik Sayısı	5
Kayıp Değer Sayısı	0
Alan	Sağlık

3.3. Geliştirilen Uygulama (Developed Application)

Bu çalışmada, Visual C# programlama dili kullanılarak KSO algoritmasının çalışma mantığına göre, nümerik veritabanlarında otomatik sınıflandırma kural keşfi yapan bir program kodlanmıştır.

3.3.1. Kedilerin oluşturulması ve ilk değerlerin atanması (Generating of cats and initialization)

Kullanılan dört veritabanı için de, 200 adet kedi oluşturulmuştur. Her kedi için pozisyon ve hız değerleri gelişigüzel atanmıştır. Her kedinin ayrıca bayrakları bulunmaktadır ve bu bayrak değerlerine göre arama ya da izleme modu süreçleri uygulanmaktadır. Programda her bir kedi, bir aday kuralı temsil etmektedir ve program çalıştırıldıktan sonra en iyi uygunluk değerine sahip kediler,

elde edilen sınıflandırma kurallarıdır. Kedinin her bir pozisyonu ise uygulanan veritabanındaki nitelikleri belirtmek için kullanılmaktadır. Her bir nitelik için Şekil 9'da da gösterildiği gibi üç pozisyon tutulmaktadır. İlki olan *b*, o niteliğin aday kuralda olup olmadığını göstermektedir. Bu değer [0, 1] aralığında değerler almaktadır; eğer 0,5'in altındaysa o nitelik kuralda yer almamaktadır, 0,5'in üstündeyse de nitelik kuraldadır. İkincisi *a*, nitelik kuralda varsa o niteliğin alabileceği alt sınır, üçüncüsü olan *ü ise*, alabileceği üst sınır olmaktadır. *a* ve *ü* değerleri kedilere, ilgili niteliğin gerçek alt ve üst aralıklarına bağlı olarak programda gelişigüzel bir şekilde atanmaktadır.

$Nitelik_1$			$Nitelik_2$			...			$Nitelik_n$		
b_1	a_1	$ü_1$	b_2	a_2	$ü_2$	...	...	...	b_n	a_n	$ü_n$

Şekil 9. Bir kedinin yapısı (Structure of a cat)

Bir veritabanında *n* tane nitelik varsa kedilerin her birinde *n*×3 tane pozisyon bulunmaktadır. Örnek olarak Liver Disorders (Bupa) veritabanında 6 tane nitelik bulunmaktadır. Bu veritabanı için oluşturulan her kedinin, Şekil 10'da da gösterildiği gibi 18 tane pozisyonu olmaktadır. Şekil 10'da, *Alkphos, Sgpt* ve *Drinks* nitelikleri *b* değerleri 0,5'ten büyük olduğu için kuralda yer almaktadır, yani aynı zamanda kuralın sol tarafında yer almaktadır. Kuralın sağ tarafı ise sınıfını temsil etmektedir. Bu aday kuralın "Sınıf 2" için çalıştırıldığı düşünülürse, bu kuralın ifade edilişi Şekil 11'deki gibi olmaktadır.

3.3.2. Uygunluk fonksiyonunun hesaplanması
(Computing of fitness function)

Kedilerin uygunluk değerleri Eş. 4'teki uygunluk fonksiyonu ile hesaplanmaktadır ve sadece uygunluk değeri en iyi olan kedinin pozisyonu hafızada saklanmaktadır.

$$uygunluk = \omega_1 * \frac{DP}{DP+YN} * \frac{DN}{DN+YP} - \omega_2 * anlaşılırlık$$

$$-\omega_3 * aralıkoranı + \omega_4 * \frac{DP}{DP+YP} \quad (4)$$

ω_1, ω_2, ω_3 ve ω_4 değerleri kullanıcı tarafından tanımlanan ağırlıklardır ve bunlara amacımıza bağlı olarak keşfedilecek kuralların sahip olması istenilen özelliğine göre değer verilebilmektedir. Yani amaçlarımız arasından kuralın doğru tahmin oranının en önemli olması isteniyorsa ω_1, kuralın anlaşılabilir olması çok daha önemli ise ω_2 daha yüksek seçilir. Bu dört değerin toplamı 1'i vermektedir. *DP*, doğru pozitiflerdir; yani kuralla aynı sınıf etiketine sahip olup, kural tarafından kapsanan örneklerin sayısıdır. *YP*, yanlış pozitiflerdir; bunlar kuraldan farklı sınıf etiketine sahip olup, kural tarafından kapsanan örneklerin sayısıdır. *YN*, yanlış negatiflerdir; kural tarafından kapsanmayan ama kuralla aynı sınıf etiketine sahip örneklerin sayısıdır. *DN*, doğru negatiflerdir; bunlar ise kural tarafından kapsanmayıp, kuralla da aynı sınıf etiketine sahip olmayan örneklerin sayısıdır. *anlaşılırlık* kriteri daha basit kuralların keşfi için kullanılmaktadır. Bu kriterin hesaplanması Eş. 5'te gösterilmektedir. Burada *n* veri kümesindeki karar niteliklerinin sayısını vermektedir.

$$anlaşılırlık = 1 - \frac{kuralın\ solundaki\ nitelik\ sayısı - 1}{n} \quad (5)$$

Aralıkoranı, ise kuraldaki niteliklerin alt sınır ve üst sınır aralığının dar tutulması için konulan bir kriterdir. Bu değer Eş. 6'da gösterildiği gibi hesaplanır. Burada *n*, veri kümesindeki karar niteliklerinin sayısını, $ü_i$, *i.* niteliğin üst sınırı, a_i, *i.* niteliğin alt sınırı, $Nitelik_i maks$, *i.* niteliğin veri kümesinde aldığı maksimum değeri, $Nitelik_i min$, *i.* niteliğin veri kümesinde aldığı minimum değeri ifade etmektedir.

$$aralıkoranı = \frac{\sum_{i=1}^{i=n} \frac{ü_i - a_i}{Nitelik_i maks - Nitelik_i min}}{n} \quad (6)$$

Daha farklı kurallar elde edilmek istenirse, yukarıdaki kriterlere ek olarak ilginçlik, sürprizlik gibi objektif ya da sübjektif başka kriterler de uygunluk fonksiyonuna kolaylıkla entegre edilebilir. Yani, fonksiyon farklı amaçlara cevap verecek şekilde esnektir.

3.3.3. Sınırlayıcıların idaresi (Constraints handling)

Bir kedinin pozisyonu güncellenirken, eğer *a* (alt sınır) değeri, *ü* (üst sınır) değerinden daha büyükse veya alt sınır değeri ilgili niteliğin alabileceği minimum değerden daha küçükse; alt sınır değeri Eş. 7'deki gibi güncellenmiştir:

Mcv			Alkphos			Sgpt			Spot			Gammagt			Drinks		
0,36	65	93	0,8	78	122	0,75	29	63	0,15	70	81	0,04	201	214	0,8	2	8

Şekil 10. Liver Disorders (Bupa) veritabanı için oluşturulan bir kedinin temsili
(A representation of a cat generated for Bupa database)

78 < Alkphos < 122 ve 29 < Sgpt < 63 ve 2 < Drinks < 8 ⟶ Sınıf 2

Kuralın sol tarafı — Kuralın sağ tarafı

Şekil 11. Bir aday kuralın ifade edilişi (Meaning of a candidate rule)

$$a_i = Nitelik_i min + r(\ddot{u}_i - Nitelik_i min) \quad (7)$$

Bu eşitlikte; a_i , *i.* niteliğin yeni alt sınır değerini; $Nitelik_i min$, *i.* niteliğin veri kümesinde alabileceği minimum değerini; $\ddot{u}_i$ *i.* niteliğin mevcut üst sınır değerini; r ise [0, 1] aralığında rassal bir sayıyı ifade etmektedir. Benzer şekilde üst sınır değeri ilgili niteliğin alabileceği maksimum değerden daha büyükse; üst sınır değeri Eş. 8'deki gibi güncellenmiştir:

$$\ddot{u}_i = a_i + r(Nitelik_i maks - a_i) \quad (8)$$

Bu eşitlikte; $\ddot{u}_i$, *i.* niteliğin yeni üst sınır değerini; $Nitelik_i maks$, *i.* niteliğin veri kümesinde alabileceği maksimum değerini ifade etmektedir.

3.3.4. Bitim şartı (Terminating criterion)

Uygulamada bitim şartı her bir kural için 100 iterasyon olarak alınmıştır. Bu 100 iterasyon sonunda en iyi uygunluk değerine sahip kedi, keşfedilen kuralı temsil eder. Her bir sınıf için de, o sınıftaki örneklerin %80-%100'ü keşfedilen kurallar tarafından kapsanıncaya kadar kural keşfi yapılmaktadır.

3.3.5. Karmaşıklık analizi (Complexity analysis)

Algoritma kompleks bir problem olan nümerik verilerden sınıflandırma kurallarının keşfi için uyarlanmıştır. Herhangi bir önişlem yapmadan, tüm sınıflar için hangi niteliklerin ilgili kuralda yer alınacağının ve bu niteliklerin uygun aralıkların bulunması; aynı zamanda bulunacak kuralların doğru ve anlaşılabilir olmasının sağlanması zor bir mühendislik problemidir. Bu problem için algoritmanın karmaşıklığının en önemli kısmını, kedi pozisyonlarının uygunluk hesabı için veritabanlarındaki kayıtlarla karşılaştırma oluşturmaktadır. Karşılaştırma sayısını da belirleyen dört temel unsur vardır. Bunlar; veritabanındaki Örnek Sayısı (*ÖS*), veritabanındaki Nitelik Sayısı (*NS*), Kedi Sayısı (*KS*) ve İterasyon Sayısı (*İS*). Buna bağlı olarak karşılaştırma sayısı *ÖS*×*NS*×*KS*×*İS* şeklinde hesaplanabilir.

3.3.6. Uygulama sonuçları (Application results)

Her veritabanı için 3 ayrı çalıştırmada elde edilen sonuçlar, tablo şeklinde aşağıda verilmiştir. Şekillerde her satırda ilk başta kuralın ait olduğu sınıf, daha sonra kuralın alabileceği nitelik değerleri bulunmaktadır. Her kuralın karşısında ise o kural için DP ve YP değerleri verilmektedir. Bir çalıştırmadaki başarı oranı ise "Toplam DP/Veritabanındaki toplam veri sayısı" olarak hesaplanmaktadır. Bunlara ek olarak her veritabanı için Weka programında One-R, Ridor, JRip ve Part algoritmalarından elde edilen sonuçlar da tablolar halinde gösterilmiştir. Her sınıf için kurallar bulundukça, bulunan kuralın kapsadığı veriler, veritabanından silinmektedir. Sınıflandırma kural keşfi yapılırken, veritabanlarındaki verilerin hepsi eğitim verisi olarak kullanılmıştır ve bulunan kurallar, verilerin tümü üzerinde test edilmiştir. Test verisi geldiğinde KSO tarafından oluşturulan kurallar kümesine sırayla uygulanır ve ilk uyan kuralın sınıf etiketini alır.

Ecoli Veritabanı Sonuçları (Results for Ecoli Database)

Ecoli veritabanı için elde edilen sonuçlar Tablo 13, Tablo 14 ve Tablo 15'te gösterilmiştir. Tablo 13'te 4 tane "cp", 5 tane "im", 2 tane "imU", 3 tane "om", 3 tane "pp" ve 1 tanede "omL" sınıfından olmak üzere toplam 18 tane kural bulunmuştur. Bu çalıştırmadaki verilerin doğru sınıflandırılma oranı 295/336=0,878'dir. Yani %87,8 oranla doğru çalışmaktadır. Tablo 14'te Ecoli veritabanı için program ikinci kez çalıştırıldığında elde edilen sonuçlar yer almaktadır. Buna göre "cp" sınıfından 3 tane, "im" sınıfından 3 tane, "imU" sınıfından 2 tane, "om" sınıfından 1 tane, "pp" sınıfından 2 tane ve "omL" sınıfından 1 tane olmak üzere, toplam 12 tane kural bulunmuştur. Bu çalıştırmada verilerin doğru sınıflandırılma oranı 292/336= 0,869'dur. Tablo 15'te Ecoli veritabanı için program üçüncü kez çalıştırıldığında elde edilen sonuçlar yer almaktadır. Bu çalıştırmada "cp" sınıfı için 3 tane, "im" sınıfı için 3 tane, "imU" sınıfı için 2 tane, "om" sınıfı 1 tane, "pp" sınıfı için 2 tane ve "omL" sınıfı için 1 tane olmak üzere, toplam 12 tane kural bulunmuştur. Bu çalıştırmada verilerin doğru sınıflandırılma oranı 288/336=0,857'dir. Ecoli veritabanı için yukarıda verilen 3 ayrı sonuca göre, veriler ortalama %86,8 oranında doğru sınıflandırılmıştır. Kullanılan tüm algoritmalardan elde edilen sonuçlar ise Tablo 16'da gösterilmektedir. Bu tabloda, Ecoli veritabanı için algoritmaların performansını karşılaştırmak amacıyla literatürde kullanılan doğruluk yüzdesi, kesinlik (precision), hassasiyet (recall) ve F-ölçütü (F-measure) değerleri verilmiştir. Tablodan, bu çalışmada önerilen KSO algoritmasının kesinlik, hassasiyet ve F-ölçütü için diğer algoritmalara göre en yüksek performansa sahip olduğu görülmektedir. Veritabanındaki sınıflar için algoritmaların bulduğu Tip-1 ve Tip-2 hataları ise Tablo 17'de gösterilmiştir.

Pima indians diabetes veritabanı sonuçları
(Results for pima indians diabetes database)

Pima Indians Diabetes veritabanı için elde edilen sonuçlar Tablo 18, Tablo 19 ve Tablo 20'de gösterildiği gibidir. Bu veritabanında sadece 2 tane sınıf olduğundan yalnızca "tested-negative" sınıfı için kurallar bulunmuştur. Ayrıca, son olarak "bu kurallara uymayan veriler "tested-positive" sınıfına aittir" kuralı, kurallar kümesine eklenmiştir. Son kural için doğru pozitif sayısı ise veritabanında "tested-positive" sınıf etiketine sahip olan verilerin sayısıdır.

Tablo 18'de Pima Indians Diabetes veritabanı için program çalıştırıldığında elde edilen sonuçlar yer almaktadır. Buna göre "tested-negative" sınıfı için toplam 6 tane kural bulunmuştur. Bu çalıştırmada verilerin doğru sınıflandırılma oranı 616/768=0,802'dir. Tablo 19'da Pima Indians Diabetes veritabanı için program ikinci defa çalıştırıldığında elde edilen sonuçlar gösterilmektedir. Buna göre "tested-negative" sınıfı için 8 tane kural bulunmuştur.

Tablo 13. Ecoli veritabanı için birinci çalıştırmada elde edilen kurallar
(Obtained rules from the first run for Ecoli database)

Sınıf	Kural	DP	YP
cp	0,175<mcg<0,529 ve 0,24<aim2<0,537	105	4
cp	0,227<gvh<0,475 ve lip=0,48 ve 0,03<aim1<0,555	27	1
cp	0,167<mcg<0,302	4	4
cp	0,552<gvh<0,663 ve 0,523<aim2<0,553	2	0
im	0<mcg<0,551 ve lip=0,48	42	2
im	0,64<aim1<0,811 ve 0,316<aim2<0,737	11	4
im	0,572<mcg<0,774 ve lip=0,48 ve 0,801<aim2<0,99	10	3
im	0,701<aim1<0,721	3	3
im	lip=0,48 ve 0,698<aim1<0,716 ve 0,735<aim2<0,965	2	2
imU	0,705<mcg<0,89 ve 0,725<aim1<0,921	15	0
imU	0,618<aim2<0,771	7	0
om	0,644<aac<0,88 ve 0,203<aim2<0,533	15	0
om	0,898<gvh<0,904	1	0
om	0,742<aac<0,751	1	0
pp	0,588<gvh<0,897 ve 0,157<aim2<0,436	34	2
pp	0,608<mcg<0,784 ve lip=0,48 ve chg=0,5 ve 0,294<aim2<0,55	11	0
pp	chg=0,5 ve 0,687<aim1<1 ve 0,184<aim2<0,202	1	0
omL	0,636<mcg<0,735 ve 0,543<aim1<0,603	4	0

Tablo 14. Ecoli veritabanı için ikinci çalıştırmada elde edilen kurallar
(Obtained rules from the second run for Ecoli database)

Sınıf	Kural	DP	YP
cp	0<mcg<0,511 ve 0,03<aim1<0,463	116	3
cp	0,16<gvh<0,467 ve lip=0,48 ve 0,353<aim2<0,615	17	0
cp	0,16<gvh<0,587 ve lip=0,48 ve 0,389<aim2<0,616	5	5
im	0,053<mcg<0,551 ve chg=0,5 ve 0,565<aim1<1	45	2
im	0,563<mcg<0,76 ve chg=0,5 ve 0,771<aim1<1	13	4
im	0,557<aac<0,703ve 0,643<aim1<1 ve 0,576<aim2<0,755	6	1
imU	0,729<mcg<0,89 ve 0,614<aim2<0,99	19	1
imU	0,423<mcg<0,695 ve lip=0,48 ve 0,732<aim2<0,773	6	0
om	0,551<gvh<1 ve 0,634<aac<0,88	15	0
pp	0,576<gvh<0,884 ve 0,355<aim1<0,668	40	0
pp	0,414<aac<0,505 ve 0,305<aim1<0,688	5	1
omL	lip=1 ve chg=0,5	5	1

Tablo 15. Ecoli veritabanı için üçüncü çalıştırmada elde edilen kurallar
(Obtained rules from the third run for Ecoli database)

Sınıf	Kural	DP	YP
cp	0,166<mcg<0,551 ve 0,173<aim2<0,479	104	4
cp	0,215<gvh<0,487 ve 0,03<aim1<0,553	27	2
cp	0,215<gvh<0,487 ve chg=0,5 ve 0,391<aim1<0,463	5	1
im	0<mcg<0,612 ve lip=0,48 ve 0,567<aim1<1	51	3
im	0,584<mcg<0,747 ve 0,636<aim2<0,75	8	3
im	0,625<mcg<0,749 ve 0,801<aim2<0,932	6	3
imU	lip=0,48 ve 0,768<aim2<0,922	13	0
imU	0,41<aac<0,593 ve 0,59<aim2<0,781	10	2
om	0,569<gvh<0,9 ve 0,647<aac<0,88	14	0
pp	0,589<gvh<0,898 ve chg=0,5 ve 0,357<aim1<0,659	40	0
pp	0,362<gvh<0,591 ve 0,446<aac<0,519 ve 0,347<aim1<1	5	0
omL	lip=1 ve chg=0,5	5	1

Tablo 16. Ecoli veritabanı için elde edilen sonuçlar (Obtained results for Ecoli database)

Kullanılan Algoritma	Bulunan Kural Sayısı	Toplam DP/Toplam Veri Sayısı	Doğruluk Yüzdesi	Kesinliğin Ağırlıklı Ortalaması	Hassasiyetin Ağırlıklı Ortalaması	F-Ölçütünün Ağırlıklı Ortalaması
One-R	7	235/336	%69,94	0,652	0,699	0,648
Part	13	308/336	%91,66	0,917	0,917	0,909
Ridor	46	297/336	%88,39	0,872	0,884	0,877
Jrip	10	305/336	%90,77	0,887	0,911	0,898
KSO	19	295/336	%87,8	0,920	0,939	0,928

Tablo 17. Ecoli veritabanı için elde edilen Tip-1 ve Tip-2 hatalar
(Obtained Type-1 and Type-2 errors for Ecoli database)

		One-R		Part		Ridor		Jrip		KSO	
		Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2
Sınıf	cp	15	25	1	5	3	9	4	13	7	7
	im	4	45	3	15	11	13	9	5	9	9
	imU	29	2	15	1	10	5	4	6	2	2
	om	20	0	2	1	2	4	2	2	0	0
	pp	24	29	3	5	8	7	2	4	0	0
	omL	5	0	2	0	1	1	5	0	1	1
	imL	2	0	0	1	2	0	2	0	0	2
	imS	2	0	2	0	2	0	2	0	0	2
Toplam		101	101	28	28	39	39	30	30	19	19

Tablo 18. Diabetes veritabanı için birinci çalıştırmada elde edilen kurallar
(Obtained rules from the first run for Diabetes database)

Sınıf	Kural	DP	YP
tested_negative	0<plas<111,915 ve 0<mass<66,942	278	45
tested_negative	0<plas<127,998	110	47
tested_negative	48,049<plas<166,752 ve 1,926<pres<122 ve 0<mass<30,155	48	15
tested_negative	0<preg<7,991 ve 75,012<plas<165,303 ve 61,191<pres<91,751 ve 6,36<skin<56,241 ve 118<insu<727,65 ve 5,44<mass<50,286	20	13
tested_negative	0<preg<16,848 ve 31,705<pres<95,138 ve 0<insu<86,08	3	0
tested_negative	22,614<plas<144,59 ve 74,382<pres<85,686	14	5
tested_positive	Üstteki kurallara uymayan veriler	143	27

Tablo 19. Diabetes veritabanı için ikinci çalıştırmada elde edilen kurallar
(Obtained rules from the second run for Diabetes database)

Sınıf	Kural	DP	YP
tested_negative	0<plas<111,998	284	45
tested_negative	0<plas<129,94 ve 0<pres<109,715 ve 0,078<pedi<2,413	110	54
tested_negative	0,104<pres<122 ve 0<mass<29,997	48	21
tested_negative	0<pres<118,805 ve 40,028<mass<40,877	8	0
tested_negative	91,671<plas<154,832 ve 28,132<pres<88,481 ve 2,376<skin<31,705	13	7
tested_negative	0<preg<16,88 ve 0<skin<98,967 ve 506,991<insu<798,541 ve 6,976<mass<64,632 ve 29,343<age<49,433	2	0
tested_negative	0<plas<119,811 ve 0<mass<67,07 ve 21<age<80,976	5	1
tested_negative	4,577<preg<17 ve 0<plas<179,985 ve 18,172<pres<122 ve 59,119<age<81	3	0
tested_positive	Üstteki kurallara uymayan veriler	140	27

Tablo 20. Diabetes veritabanı için üçüncü çalıştırmada elde edilen kurallar
(Obtained rules from the third run for Diabetes database)

Sınıf	Kural	DP	YP
tested_negative	0<plas<111,925 ve 0<mass<54,959	278	44
tested_negative	37,569<plas<143,354 ve 30,177<pres<89,893 ve 0<skin<96,567 ve 0,078<pedi<2,283	108	44
tested_negative	43,175<plas<151,165 ve 3,629<mass<27,272	27	2
tested_negative	4,315<preg<17 ve 39,12<plas<180,549 ve 0<pres<118,05 ve 0<mass<64,018 ve 58,026<age<81	5	0
tested_negative	0,048<pres<122 ve 0<skin<98,98 ve 579,2<insu<846	2	0
tested_negative	136,02<insu<426,951 ve 1,106<mass<67,1 ve 22<age<28	7	0
tested_negative	0<preg<15,768 ve 66,852<plas<171,43 ve 59,814<pres<90,481 ve 34,224<skin<66,755 ve 0,002<mass<67,1	5	0
tested_negative	1,101<preg<17 ve 0<plas<114,784 ve 0,078<pedi<2,396	6	0
tested_negative	116,521<plas<148,919ve 5,222<pres<122 ve 0<mass<30,048 ve 23<age<81	5	0
tested_positive	Üstteki kurallara uymayan veriler	178	57

Tablo 21. Diabetes veritabanı için elde edilen sonuçlar
(Obtained results for Diabetes database)

Kullanılan Algoritma	Bulunan Kural Sayısı	Toplam DP/ Toplam Veri Sayısı	Doğruluk Yüzdesi	Kesinliğin Ağırlıklı Ortalaması	Hassasiyetin Ağırlıklı Ortalaması	F-Ölçütünün Ağırlıklı Ortalaması
One-R	10	586/768	%76,30	0,759	0,764	0,755
Part	13	624/768	%81,25	0,845	0,813	0,817
Ridor	4	605/768	%78,77	0,799	0,788	0,769
Jrip	4	609/768	%79,29	0,789	0,793	0,788
KSO	10	621/768	%80,90	0,805	0,926	0,859

Tablo 22. Diabetes veritabanı için elde edilen Tip-1 ve Tip-2 hatalar
(Obtained Type-1 and Type-2 errors for Diabetes database)

		One-R		Part		Ridor		Jrip		KSO	
		Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2
Sınıf	Tested-negative	56	125	119	25	23	140	57	102	90	57
	Tested-positive	125	56	25	119	140	23	102	57	57	90
Toplam		181	181	144	144	163	163	159	159	147	147

Tablo 23. BUPA veritabanı için birinci çalıştırmada elde edilen kurallar
(Obtained rules from the first run for BUPA database)

Sınıf	Kural	DP	YP
2	35<gammagt<297	84	33
2	4<sgpt<19,495 ve 13,312<sgot<82 ve 0<drinks<19,98	51	16
2	65<mcv<88,966 ve 21,066<sgot<82	25	9
2	82,757<alkphos<97,955 ve 16,976<gammagt<231,981	3	0
2	65<mcv<97,889 ve 31,798<alkphos<126,708 ve 23,617<sgot<28,621	8	0
2	90,354<mcv<97 ve 53<alkphos<123 ve 1<drinks<3	6	0
2	65<mcv<87,877 ve 3,077<drinks<7,976	3	0
2	65<mcv<90,99 ve 4,049<drinks<7,913	1	0
2	23<alkphos<76,942 ve 32,05<sgot<82	1	0
2	23<alkphos<53,985 ve 27,148< gammagt <297	2	0
2	55,678<alkphos<57,852 ve 16,593< gammagt <297	3	0
1	Üstteki kurallara uymayan veriler	87	13

Tablo 24. BUPA veritabanı için ikinci çalıştırmada elde edilen kurallar
(Obtained rules from the second run for BUPA database)

Sınıf	Kural	DP	YP
2	22,027< sgot <56,671 ve 20,295< gammagt <279,023	89	33
2	23<alkphos<78,977 ve 4<sgpt<19,794	48	10
2	28< gammagt <297	27	17
2	65<mcv<103 ve 4<sgpt<155 ve 8,588<drinks<15,975	1	0
2	23<alkphos<137,267 ve 30,054<sgot<82 ve 0,516<drinks<20	2	0
2	65<mcv<102,497 ve 35,362<alkphos<56,845 ve 21,602<sgot<82	5	0
2	65<mcv<86,84 ve 54,39<alkphos<76,461	6	0
2	65<mcv<89,988 ve 65,452<alkphos<94,924 ve 3,06<drinks<20	2	0
2	65<mcv<90,881 ve 99,659<alkphos<138 ve 0,523<drinks<20	2	0
2	65<mcv<86,906 ve 23<alkphos<98,706 ve 3,009<drinks<20	1	0
1	Üstteki kurallara uymayan veriler	85	17

Tablo 25. BUPA veritabanı için üçüncü çalıştırmada elde edilen kurallar
(Obtained rules from the third run for BUPA database)

Sınıf	Kural	DP	YP
2	67,77<mcv<96,958 ve 21,084< gammagt<72,36	99	40
2	4<sgpt<19,414	47	19
2	78,65<mcv<89,136 ve 98< gammagt<242,606	6	0
2	76,672<mcv<88,163 ve 35<alkphos<91,271 ve 3,305<drinks<11,153	7	0
2	66,887<alkphos<104,774 ve 47<sgot<82	5	0
2	38,2<alkphos<74,752 ve 5<sgot<65 ve 145< gammagt<247,606	3	0
2	26,938<sgot<50,58 ve 9,292<drinks<14,875	3	0
2	35,214<alkphos<62,628 ve 3,254<drinks<5,706	6	0
2	89,902<mcv<93,819 ve 40,625<sgpt<45,992 ve 22,567<sgot<37,061	2	0
2	80,304<mcv<85,95 ve 57,113<alkphos<74,587	4	0
1	Üstteki kurallara uymayan veriler	86	18

Tablo 26. BUPA veritabanı için elde edilen sonuçlar (Obtained results for BUPA database)

Kullanılan Algoritma	Bulunan Kural Sayısı	Toplam DP/Toplam Veri Sayısı	Doğruluk Yüzdesi	Kesinliğin Ağırlıklı Ortalaması	Hassasiyetin Ağırlıklı Ortalaması	F-Ölçütünün Ağırlıklı Ortalaması
One-R	14	235/345	%68,11	0,695	0,693	0,694
Part	15	297/345	%86,08	0,866	0,861	0,858
Ridor	3	246/345	%71,30	0,761	0,751	0,739
Jrip	5	270/345	%78,26	0,729	0,730	0,726
KSO	12	274/345	%79,40	0,808	0,962	0,878

Tablo 27. BUPA veritabanı için elde edilen Tip-1 ve Tip-2 hatalar
(Obtained Type-1 and Type-2 errors for BUPA database)

		One-R		Part		Ridor		Jrip		KSO	
		Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2
Sınıf	2	57	49	37	11	67	19	59	34	58	13
	1	49	57	11	37	19	67	34	59	13	58
Toplam		106	106	48	48	86	86	93	93	71	71

Tablo 28. New Thyroid veritabanı için birinci çalıştırmada elde edilen kurallar
(Obtained rules from the first run for New Thyroid database)

Sınıf	Kural	DP	YP
3	4,239<Thyroidstimulating<41,033	23	0
2	3,287<Triiodothyronine<7,822	20	0
2	17,161<Thyroxin<23,019	7	0
2	87,643<T3resin<99,24 ve 10,56<Thyroxin<15,188	5	0
1	Üstteki kurallara uymayan veriler	150	10

Tablo 29. New Thyroid veritabanı için ikinci çalıştırmada elde edilen kurallar
(Obtained rules from the second run for New Thyroid database)

Sınıf	Kural	DP	YP
1	6,595<Thyroxin<11,035	108	1
1	99,809<T3resin<117,545 ve 5,636<Thyroxin<13,196	27	2
1	112,674<T3resin<129,425 ve 11,328<Thyroxin<16,324	10	0
2	17,183<Thyroxin<24,119	26	0
2	2,695<Triiodothyronine<5,808	7	0
3	Üstteki kurallara uymayan veriler	28	6

Tablo 30. New Thyroid veritabanı için üçüncü çalıştırmada elde edilen kurallar
(Obtained rules from the third run for New Thyroid database)

Sınıf	Kural	DP	YP
1	6,588<Thyroxin<11,036	108	1
1	11,079<Thyroxin<12,916	22	4
1	5,689<Thyroxin<6,515	12	2
1	106,924<T3resin<121,659 ve 4,54<Thyroxin<16,502 ve 0,28<TSH_value<3,743	7	0
2	2,697<Triiodothyronine<7,822	22	0
2	15,084<Thyroxin<20,541	8	0
3	Üstteki kurallara uymayan veriler	27	2

Tablo 31. New Thyroid veritabanı için elde edilen sonuçlar
(Obtained results for New Thyroid database)

Kullanılan Algoritma	Bulunan Kural Sayısı	Toplam DP/Toplam Veri Sayısı	Doğruluk Yüzdesi	Kesinliğin Ağırlıklı Ortalaması	Hassasiyetin Ağırlıklı Ortalaması	F-Ölçütünün Ağırlıklı Ortalaması
One-R	3	198/215	%92,09	0,921	0,921	0,920
Part	4	213/215	%99,06	0,991	0,991	0,991
Ridor	7	206/215	%95,81	0,969	0,967	0,967
Jrip	4	209/215	%97,20	0,972	0,972	0,972
KSO	7	206/215	%95,81	0,959	0,990	0,974

Tablo 32. New Thyroid veritabanı için elde edilen Tip-1 ve Tip-2 hatalar
(Obtained Type-1 and Type-2 errors for New Thyroid database)

		One-R		Part		Ridor		Jrip		KSO	
		Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2	Tip-1	Tip-2
Sınıf	1	5	12	0	2	0	7	3	3	7	1
	2	6	4	0	0	3	0	2	2	0	2
	3	6	1	2	0	4	0	1	1	2	0
Toplam		17	17	2	2	7	7	6	6	9	3

İkinci çalıştırmada verilerin doğru sınıflandırılma oranı 613/768=0,798'dir. Tablo 20'de Pima Indians Diabetes veritabanı için program üçüncü kez çalıştırıldığında elde edilen sonuçlar yer almaktadır. Buna göre "tested-negative" sınıfı için 9 tane kural bulunmuştur. Bu çalıştırmada verilerin doğru sınıflandırılma oranı 621/768=0,809'dur. Yani %80,9 oranla doğru sonuçlar vermektedir. Pima Indians Diabetes veritabanı için yukarıda verilen 3 ayrı sonuca göre, veriler ortalama %80,3 oranında doğru sınıflandırılmıştır. Bu veritabanı için kullanılan algoritmalardan elde edilen sonuçlar ise Tablo 21'de gösterilmektedir. Bu tabloya göre; KSO algoritması sadece doğruluk yüzdesi ve kesinlik ölçütü için Part algoritmasından kötü, diğer ölçütler için tüm algoritmalardan daha iyi sonuç vermiştir. Veritabanındaki sınıflar için algoritmaların bulduğu Tip-1 ve Tip-2 hataları ise Tablo 22'de gösterilmiştir.

Liver disorders (BUPA) veritabanı sonuçları
(Results for liver disorders (BUPA) database)

Liver Disorders (BUPA) veritabanı için elde edilen sonuçlar Tablo 23, Tablo 24 ve Tablo 25'te gösterildiği gibidir. BUPA veritabanında da sadece 2 tane sınıf olduğundan "2" sınıfı için kurallar bulunmuştur. Bu kuralların dışında kalanlar ise "1" sınıfına ait olmaktadır. "1" sınıfına ait olan doğru pozitif sayısı ise veritabanında "1" sınıf etiketine sahip olan verilerin sayısıdır. Tablo 23'te

BUPA veritabanı için programın ilk çalıştırılmasında elde edilen kurallar yer almaktadır. "2" sınıfına ait toplam 11 tane kural bulunmuştur. Bu çalıştırmada verilerin doğru sınıflandırılma oranı 274/345=0,794'tür. Tablo 24'te BUPA veritabanı için ikinci çalıştırmada elde edilen kurallar yer almaktadır. "2" sınıfına ait toplam 10 tane kural bulunmuştur. Buna göre verilerin doğru sınıflandırılma oranı 268/345=0,777'dir. Tablo 25'te BUPA veritabanı için üçüncü çalıştırmada elde edilen kurallar yer almaktadır. "2" sınıfına ait toplam 10 tane kural bulunmuştur. Buna göre verilerin doğru sınıflandırılma oranı 268/345=0,777'dir. Liver Disorders (BUPA) veritabanı için yukarıda verilen 3 ayrı sonuca göre, veriler ortalama %78,3 oranında doğru sınıflandırılmıştır. Bu veritabanı için kullanılan algoritmalardan elde edilen sonuçlar ise Tablo 26'da gösterilmektedir. Bu tabloya göre; KSO algoritması sadece doğruluk yüzdesi ve kesinlik ölçütü için Part algoritmasından kötü, diğer ölçütler için tüm algoritmalardan daha iyi sonuç vermiştir. Veritabanındaki sınıflar için algoritmaların bulduğu Tip-1 ve Tip-2 hataları ise Tablo 27'de gösterilmiştir.

Thyroid disease (new thyroid) veritabanı sonuçları
(Results for thyroid disease (new thyroid) database)

Thyroid Disease (New Thyroid) veritabanı için elde edilen sonuçlar Tablo 28, Tablo 29 ve Tablo 30'da gösterilmiştir. Bu veritabanında 3 tane sınıf bulunmaktadır. Sınıflandırma kuralları bulunurken sadece iki sınıf için kural bulunmaktadır. Bu kuralların dışında kalanlar ise üçüncü sınıfta olmaktadır. Son sınıf için DP sayısı, veritabanında son sınıf ile aynı sınıf etiketine sahip olan verilerin sayısıdır. Tablo 28'de Thyroid Disease (New Thyroid) veritabanı bir kere çalıştırıldığında elde edilen sonuçlar yer almaktadır. Program "3" ve "2" sınıfları için çalıştırılmıştır ve "3" sınıfından 1 tane, "2" sınıfından da 3 tane olmak üzere, toplamda 4 tane sınıflandırma kuralı üretilmiştir. Buna göre verilerin doğru sınıflandırılma oranı 205/215=0,953'tür. Tablo 29'da Thyroid Disease (New Thyroid) veritabanı ikinci kez çalıştırıldığında elde edilen sonuçlar yer almaktadır. Bu çalıştırmada "1" ve "2" sınıfları için kurallar bulunmuştur. "1" sınıfından 3 tane ve "2" sınıfından 2 tane olmak üzere, toplamda 5 tane kural bulunmuştur. Buna göre verilerin doğru sınıflandırılma oranı 206/215=0,958'dir. Tablo 30'da Thyroid Disease (New Thyroid) veritabanı üçüncü kez çalıştırıldığında elde edilen sonuçlar yer almaktadır. Bu çalıştırmada da "1" ve "2" sınıfları için kurallar bulunmuştur. "1" sınıfı için 4 tane ve "2" sınıfı için 2 tane olmak üzere, toplamda 6 tane kural bulunmuştur. Buna göre verilerin doğru sınıflandırılma oranı 206/215=0,958'dir. Yani program %95,8 doğru kurallar üretmektedir. Thyroid Disease (New Thyroid) veritabanı için yukarıda verilen 3 ayrı sonuca göre, veriler ortalama %95,6 oranında doğru sınıflandırılmıştır. Bu veritabanı için kullanılan algoritmalardan elde edilen diğer sonuçlar Tablo 31'de gösterilmektedir. Bu tabloya göre; KSO algoritması doğruluk yüzdesi için One-R'den daha iyi Ridor ile aynı; kesinlik ölçütü için One-R'den iyi, hassasiyet ve F-ölçütleri için One-R, Ridor ve Jrip'ten daha iyi sonuç vermiştir. Veritabanındaki sınıflar için algoritmaların bulduğu Tip-1 ve Tip-2 hataları ise Tablo 32'de gösterilmiştir.

4. SONUÇLAR (CONCLUSIONS)

Optimizasyon problemleri için birçok algoritma önerilmiştir. Metasezgisel algoritmalar, büyük boyutlu optimizasyon problemleri için, kabul edilebilir sürede optimuma yakın çözümler verebilen genel çözüm yöntemleridir. Bu çalışmada, sürü zekası tabanlı güncel metasezgisel optimizasyon algoritmalarından KSO ile Visual C# dilinde, ilk kez nümerik veritabanlarında otomatik sınıflandırma kural keşfi yapacak bir program geliştirilmiştir. UCI veri ambarından alınan 4 adet veritabanı bu programda uygulanmış ve elde edilen sonuçlar Weka programından elde edilen sonuçlar ile karşılaştırılmıştır. Buna göre;

Ecoli veritabanı için, bu çalışmada elde edilen kuralların doğruluk yüzdesi ortalama olarak %86,8 olmaktadır. Weka programında ise One-R algoritması için %69,04; Part algoritması için %91,66; Ridor algoritması için %88,39 ve Jrip algoritması içinde %90,77 olmaktadır. Karşılaştırma yapıldığında One-R algoritmasına göre daha iyi sonuçlar elde edilirken, diğer 3 algoritmaya göre daha kötü sonuçlar elde edilmiştir. KSO algoritmasının kesinlik, hassasiyet ve F-ölçütü için ise diğer algoritmalara göre en yüksek performansa sahip olduğu görülmüştür. Pima Indians Diabetes veritabanı için elde edilen kuralların doğruluk yüzdesi ortalama olarak %80,3 olmaktadır. Weka programında ise One-R algoritması için %76,30; Part algoritması için %81,25; Ridor algoritması için %78,77 ve Jrip algoritması için%79,29 olmaktadır. Bulunan değerler karşılaştırıldığında Part algoritmasına göre daha kötü sonuçlar elde edilirken diğer 3 algoritmadan ise daha iyi sonuçlar bulunmuştur. KSO algoritması kesinlik ölçütü için Part algoritmasından kötü, hassasiyet ve F-ölçütleri için diğer tüm algoritmalardan daha iyi sonuçlar vermiştir. Liver Disorders (BUPA) veritabanı için elde edilen kuralların doğruluk yüzdesi %78,3 olmaktadır. Weka programında ise One-R algoritması için %68,11; Part algoritması için %86,08; Ridor algoritması için %71,30 ve Jrip algoritması için %78,26 olmaktadır. Değerler karşılaştırıldığında, Part algoritmasına göre daha kötü sonuçlar elde edilirken, diğer üç algoritmaya göre ise daha iyi sonuçlar elde edilmiştir. KSO algoritması kesinlik ölçütü için Part algoritmasından kötü, hassasiyet ve F-ölçütleri için diğer algoritmalardan daha iyi sonuçlar vermiştir. Thyroid Disease (New Thyroid) veritabanı için elde edilen kuralların doğruluk yüzdesi ise ortalama olarak %95,6 olmaktadır. Weka programında ise One-R algoritması için %92,09; Part algoritması için %99,06; Ridor algoritması için %95,81 ve Jrip algoritması için %97,20 olmaktadır. Karşılaştırma yapıldığında Part ve Jrip algoritmalarının göre daha kötü sonuçlar elde edilirken, One-R algoritmasından daha iyi sonuçlar bulunmuştur ve Ridor algoritmasına ise yakın sonuçlar bulunmuştur. KSO algoritması kesinlik ölçütü için One-R'den iyi, hassasiyet ve F-ölçütleri için One-R, Ridor ve Jrip'ten daha iyi

sonuçlar vermiştir. Elde edilen ve karşılaştırılması yapılan bu sonuçlara göre, KSO algoritmasının sınıflandırma kural keşfinde etkili bir yöntem olduğu görülmektedir. KSO algoritması üzerinde herhangi bir iyileştirme yapılmadığı ve bu alanda ilk kez kullanıldığı halde elde edilen sonuçların umut verici olduğu gözlenmiştir. Ayrıca geliştirilen programda, bulunan kurallardaki niteliklerin değer aralıkları, kurallarla eş zamanlı olarak bulunmaktadır. Yani niteliklerin değer aralıkları için ayrıca bir ön işlem gerekmemekte ve kuralla birlikte otomatik olarak ayarlanmaktadır. Kullanılan amaç fonksiyonunun çok esnek olması ve farklı objektif ya da sübjektif amaçların kolaylıkla fonksiyona entegre edilebilir olması, nitelikler arası etkileşimle baş edebilmesi, kural budama gibi ek işlemlere ihtiyaç duymaması; yöntemin diğer üstünlükleri olarak sıralanabilir. KSO çözüm uzayı üzerinde yani bu çalışmada veritabanlarında ön bir bilgiye ihtiyaç duymaz ve varsayımlar yapmaz. Bu uzaydaki süreksizlikler, eksik ve hatalar; arama süresince çok az etkiye sahiptir. KSO farklı tipte verilerden oluşan veritabanlarında etkili bir kural keşfetme algoritması olarak görülmektedir. Bu makalede kural keşfi için uyarlanan KSO ilk önerilen haliyle kullanılmıştır. Yani algoritma üzerinde herhangi bir iyileştirme yapılmamıştır. Daha başarılı sonuçların elde edilmesi amacıyla, algoritmanın ilk aşaması olan başlangıç popülasyonu üretme aşamasında iyileştirmeler yapılabilir. Yani başlangıç kedi pozisyonlarının gelişigüzel üretilmesi yerine, sistematik bir yolla düzenli olarak arama uzayına dağıtımı yapılabilir. Bu amaçla kaos, düzenli popülasyon, ortogonal diziler gibi birçok başlangıç popülasyonu üretim tekniği kullanılabilir. Ayrıca, KSO'nun pozisyon güncelleme eşitlikleri; kaos, bulanık mantık ya da uyarlamalı katsayılarla iyileştirilebilir. Ayrıca algoritmanın sonlanma şartı, literatürde bulunan sonlandırma yöntemlerinden biri seçilerek değiştirilebilir. Bunların haricinde, sınıflar için bulunan kuralların kapsadığı kayıtların veritabanından silinmesi yerine, bu kayıtlara birer ceza verilip bulunacak diğer kuralların doğru pozitif hesabında bu cezaların hesaba katılması yoluyla, doğruluk oranı arttırılabilir. Yöntemin iyi ayarlanmış parametrelerle adalı, paralel, dağıtık ve hibrit versiyonları düzenlenebilir ve nümerik veritabanlarında veri madenciliğinin kümeleme kurallarının keşfi, birliktelik kurallarının keşfi, ardışık örüntü keşfi vb. alanlarında da etkili şekilde kullanılabilir.

5. TEŞEKKÜR (ACKNOWLEDGEMENT)

Bu çalışma Tunceli Üniversitesi Bilimsel Araştırma Projeleri Birimi (TÜNİBAP) tarafından YLTUB011-15 No'lu proje kapsamında desteklenmiştir. Desteklerinden ötürü yazarlar TÜNİBAP'a teşekkür eder.

KAYNAKLAR (REFERENCES)

1. Alataş B., Akın E., Sınıflandırma Kurallarının Karınca Koloni Algoritmasıyla Keşfi, ELECO'2004, Bursa, 357-361, Aralık, 2004.
2. Özkan Y., Veri Madenciliği Yöntemleri, Papatya Yayıncılık Eğitim, 2008.
3. Ga D., Boullé M., A Bayesian Approach for Classification Rule Mining in Quantitative Databases, Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, 243-259, 2012.
4. Olmo J. L., Romero J. R., Ventura S., Classification Rule Mining Using Ant Programming Guided by Grammar with Multiple Pareto Fronts, Soft Computing, 16 (12), 2143-2163, 2012.
5. Changpetch P., Lin D. K., Interestingness Measures for Classification Rule Mining: Model Selection Ability, International Journal of Computer Science Issues (IJCSI), 12 (4), 76, 2015.
6. Al-diabat M., Arabic Text Categorization Using Classification Rule Mining, Applied Mathematical Sciences, 6 (81), 4033-4046, 2012.
7. Jacques J., Taillard J., Delerue D., Dhaenens C., Jourdan L., Conception of a Dominance-based Multi-Objective Local Search in the Context of Classification Rule Mining in Large and Imbalanced Data Sets, Applied Soft Computing, 34, 705-720, 2015.
8. Cano A., Zafra A., Ventura S., An Interpretable Classification Rule Mining Algorithm, Information Sciences, 240, 1-20, 2013.
9. Mishra B. S. P., Addy A. K., Roy R., Dehuri S., Parallel Multi-Objective Genetic Algorithms for Associative Classification Rule Mining, Proceedings of the 2011 International Conference on Communication, Computing & Security, ACM, 409-414, 2011.
10. Salama K. M., Abdelbar A. M., Otero F. E., Freitas A. A., Utilizing Multiple Pheromones in an Ant-based Algorithm for Continuous-Attribute Classification Rule Discovery, Applied Soft Computing, 13 (1), 667-675, 2013.
11. Haltaş A., Alkan A., Karabulut M., Performance Analysis of Heuristic Search Algorithms in Text Classification, Journal of the Faculty of Engineering and Architecture of Gazi University, 30 (3), 417-427, 2015.
12. Köklü M., Kahramanlı H., Allahverdi N., A New Accurate and Efficient Approach to Extract Classification Rules, Journal of the Faculty of Engineering and Architecture of Gazi University, 29 (3), 477-486, 2014.
13. Chu S. C., Tsai P. W., Pan J. S., Cat Swarm Optimization, 9th Pacific Rim International Conference on Artificial Intelligence, LNAI, 4099, 854-858, 2006.
14. Santosa B., Ningrum M.K., Cat Swarm Optimization for Clustering, International Conference of Soft Computing and Pattern Recognition, 54-59, 2009.
15. Wang Z. H., Chang C. C., Li M. C., Optimizing Least-Significant-Bit Substitution Using Cat Swarm Optimization Strategy, Information Sciences, 10.1016/j.ins.2010.07.011, 2010.
16. Hwang J. C., Chen J. C., Pan J. S., Huang Y.C., CSO and PSO to Solve Optimal Contract Capacity for High Tension Customers, International Conference on Power Electronics and Drive Systems, 246-251, 2009.
17. Lin K. C., Chien H. Y., CSO-Based Feature Selection and Parameter Optimization for Support Vector

Machine, Joint Conferences on Pervasive Computing, 783-788, 2009.

18. Kalaiselvan G., Lavanya A., Natrajan V., Enhancing the Performance of Watermarking Based on Cat Swarm Optimization Method, International Conference on Recent Trends in Information Technology, 1081-1086, 2011.

19. Wang W., Wu J., Emotion Recognition Based on CSO&SVM in E-Learning, Seventh International Conference on Natural Computation, 1, 566-570, 2011.

20. Tsai P. W., Pan J. S., Chen S.M., Liao B.Y., Hao S.P., Parallel Cat Swarm Optimization, International Conference on Machine Learning and Cybernetics, 6, 3328-3333, 2008.

21. Pradhan P. M., Panda G., Solving Multiobjective Problems Using Cat Swarm Optimization, Expert Systems with Applications, 39 (3), 2956-2964, 2012.

22. Wang Z. H., Chang C. C., Li M. C., Optimizing Least-Significant-Bit Substitution Using Cat Swarm Optimization Strategy, Information Sciences, 192, 98-108, 2012.

23. Naresh G., Ramalinga Raju M., Narasimham S. V. L., Enhancement of Power System Stability Employing Cat Swarm Optimization based PSS, 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), IEEE, 1-6, 2015.

24. Kong L., Pan J. S., Tsai P. W., Vaclav S., Ho J. H., A Balanced Power Consumption Algorithm Based on Enhanced Parallel Cat Swarm Optimization for Wireless Sensor Network, International Journal of Distributed Sensor Networks, 2015.

25. Saha S. K., Ghoshal S. P., Kar R., Mandal D., Cat Swarm Optimization Algorithm for Optimal Linear Phase FIR Filter Design, ISA Transactions, 52 (6), 781-794, 2013.

26. Kumar D., Samantaray S. R., Kamwa I., Sahoo N. C., Reliability-Constrained based Optimal Placement and Sizing of Multiple Distributed Generators in Power Distribution Network Using Cat Swarm Optimization, Electric Power Components and Systems, 42 (2), 149-164, 2014.

27. Majumder P., Eldho T. I., A New Groundwater Management Model by Coupling Analytic Element Method and Reverse Particle Tracking with Cat Swarm Optimization, Water Resources Management, 30 (6), 1953-1972,2016.

28. Meziane R., Amara M., Hamzi A., Boufala S., Hybrid Solar Gas Reliability Optimization Using Cat Swarm Optimization Under Performance and Cost Constraints, 2015 International Conference on Renewable Energy Research and Applications (ICRERA), Palermo, IEEE, 252-257, November, 2015.

29. Dong L., Yuxiang H., Qiang F., Imran K. M., Song C., Yinmao Z., Optimizing Channel Cross Section in Irrigation Area using Improved Cat Swarm Optimization Algorithm, International Journal of Agricultural and Biological Engineering, 9 (5), 76-82, 2016.

30. Lin K. C., Zhang K. Y., Huang Y. H., Hung J. C., Yen N., Feature Selection based on an Improved Cat Swarm Optimization Algorithm for Big Data Classification, The Journal of Supercomputing, 72 (8), 1-12, 2016.

31. Yang F., Ding M., Zhang X., Hou W., Zhong C., Non-Rigid Multi-Modal Medical Image Registration by Combining L-BFGS-B with Cat Swarm Optimization, Information Sciences, 316, 440-456, 2015.

32. Guo L., Meng Z., Sun Y., Wang L., Parameter Identification and Sensitivity Analysis of Solar Cell Models with Cat Swarm Optimization Algorithm, Energy Conversion and Management, 108, 520-528, 2016.

33. Mohapatra P., Chakravarty S., Dash P. K., Microarray Medical Data Classification using Kernel Ridge Regression and Modified Cat Swarm Optimization based Gene Selection System, Swarm and Evolutionary Computation, 28, 144-160, 2016.

34. Chu S. C., Tsai P. W., Computational Intelligence Based on The Behavior of Cats, International Journal of Innovative Computing, Information and Control, 3, 163-173, 2007.

35. UCI Machine Learning Repository. Ecoli veritabanı. http://archive.ics.uci.edu/ml/datasets/Ecoli. Erişim tarihi 20 Haziran 2014.

36. UCI Machine Learning Repository. Diabet veritabanı. http://archive.ics.uci.edu/ml/datasets/Diabetes. Erişim tarihi 1 Eylül 2014.

37. UCI Machine Learning Repository. BUPA Liver Disorders veritabanı. http://archive.ics.uci.edu/ml/datasets/Liver+Disorders. Erişim tarihi 1 Kasım 2014.

38. KEEL. New Thyroid veritabanı. http://sci2s.ugr.es/keel/dataset.php?cod=66. Erişim tarihi 20 Kasım 2014.

39. Weka veri madenciliği programı. http://www.cs.waikato.ac.nz/~ml/weka/. Erişim tarihi 20 Haziran 2014.

40. Holt R. C., Very Simple classification rules perform well on most commonly used datasets, Machine Learning, 11, 69-90, 1993.

41. Novakovic J., Minic M., Veljovic A., Genetic Search for Feature Selection in Rule Induction Algorithms, 18th Telecommunications forum TELFOR, Serbia, Belgrade, November 23-25, 2010.

42. Witten I.H., Frank E., Weka machine learning algorithms in java, in Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann , 265-320, 2000.

43. Cohen W.W., Fast Effective Rule Induction, Twelfth International Conference on Machine Learning, Morgan Kaufmann, 115-123, 1995.