



# YAPAY SİNİR AĞLARININ EŞZAMANLI EĞİTİMİ VE KURAL ÇIKARIMI: DIFACONN-MINER

**Sinem KULLUK**  
**Lale ÖZBAKIR**  
**Adil BAYKASOĞLU**

## ÖZET

Yapay sinir ağları (YSA) genelleme, tahminleme, sınıflandırma gibi kabiliyetlerinden dolayı veri madenciliği (VM) alanında yaygın olarak kullanılmaktadır. Yapay sinir ağları oldukça yüksek sınıflandırma doğrulukları elde etmelerine rağmen elde ettikleri bilgileri kullanıcıların anlayabileceği kurallar şeklinde sunamamaktadırlar ve tanımlama kabiliyetleri oldukça sınırlıdır. Bu nedenle günümüzde veri madenciliğinde yapay sinir ağlarının kullanımında eğilim, yapay sinir ağlarından kural çıkarımına yöneliktir. Bu çalışmada ele alınan DIFACONN-miner da bu eğilimden yola çıkarak, ileri beslemeli yapay sinir ağlarından sınıflandırma kuralları çıkarımına yönelik yazarlar tarafından 2010 yılında geliştirilmiş yeni bir sınıflandırma algoritmasıdır. Literatürde yapay sinir ağlarından kural çıkarımına yönelik pek çok çalışma mevcuttur ancak ele alınan DIFACONN-miner bu algoritmalarından oldukça farklıdır. Geçmiş çalışmalarda YSA eğitimi ve kural çıkarımı birbirinden bağımsız olarak gerçekleştirilmesine rağmen, sunulan algoritmada sınıflandırma kuralları çıkarımı için yapay sinir ağları eğitimi ve kural çıkarımı çoklu amaç değerlendirme yapısı ile eşzamanlı olarak gerçekleştirilmektedir. DIFACONN-miner yapay sinir ağlarının eğitimi için diferansiyel gelişim (DE) algoritmasını, kural çıkarımı için ise tur atan karınca koloni optimizasyonu (TACO) algoritmasını kullanmaktadır. Sınıflandırma referans veri kümeleri, bir gerçek hayat veri kümesi üzerinde deneysel çalışmalar ve diğer kural çıkarım algoritmaları ile karşılaştırmalar, DIFACONN-miner'in diğer algoritmalara göre daha doğru, kısa ve özlü kurallar çıkarmakta oldukça yetenekli olduğunu göstermiştir.

**Anahtar Kelimeler:** Veri madenciliği, Sınıflandırma, Kural çıkarımı, Yapay sinir ağları, Karınca koloni optimizasyonu, Diferansiyel gelişim.

## ABSTRACT

Artificial Neural Networks (ANNs) are widely used in data mining field due to their capabilities such as generalization, prediction and classification. Although ANNs can achieve high classification accuracies, they can't represent the discovered knowledge in the rule form for end user and their explanation capability is very limited. Therefore one of the main challenges in using ANNs in data mining applications is to extract explicit knowledge from them. Based on this motivation, DIFACONN-miner algorithm presented in this study is a novel approach for extracting classification rules from feed forward type ANNs which was developed by the authors in 2010. Although there are several approaches in the literature for classification rule extraction from ANNs, the present approach is fundamentally different from them. While ANN training and rule extraction is generally performed independently in previous studies, in the present study, training of ANNs and rule extraction are performed simultaneously with a multiple objective evaluation framework for classification rule extraction. DIFACONN-miner makes use of differential evolution (DE) algorithm for training and touring ant colony optimization (TACO) algorithm for rule extracting. Experimental study on the classification benchmark datasets, a real-world dataset and comparisons with other rule extraction



algorithms has shown that DIFACONN-miner is highly capable of extracting more accurate and concise classification rules with respect to other algorithms.

**Key Words:** Data mining, Classification, Rule extraction, Artificial neural networks, Ant colony optimization, Differential evolution.

## 1. GİRİŞ

Veri madenciliği veya başka bir ifadeyle “veriden gizli tahminleyici bilginin çıkarımı”, kullanıcılara büyük veri kümelerindeki önemli bilgiye odaklanmaya yardım etmekte büyük potansiyele sahip güçlü bir yeni teknolojidir. VM'nin genel amacı sadece doğru değil aynı zamanda kullanıcı için anlaşılır ve ilginç bilgiyi keşfetmektir. Verideki bilgiyi veya yapısal örüntüleri göstermenin karar ağaçları, karar tabloları, sınıflandırma kuralları, birliktelik kuralları gibi çok farklı yolları vardır. Literatürde girdiler ile çıktılar arasındaki öğrenmeyi göstermekte çeşitli metotlar sunulmuştur. Karar ağacı üretimine böl ve kazan yaklaşımı, istatistiksel modelleme, kural üretim kapsama, birliktelik kuralı madenciliği bunlara örnek olarak verilebilir. Bu çalışmada ele alınan DIFACONN-miner algoritmasında, böl ve kazan stratejisi kullanılarak kural kümesi üretme sürecine kural üretim kapsama yaklaşımı uyarlanmıştır.

Yapay sinir ağları tahminleme, deneyimlerden öğrenme ve geçmiş örneklerden genelleme konusunda oldukça yeteneklidir [1] ve bu yetenek YSA'ları sınıflandırma uygulamaları için oldukça uygun kılar. Yapay sinir ağları oldukça yüksek sınıflandırma doğrulukları elde edebilirler, ancak en önemli eksiklikleri sınırlı tanımla kabiliyetleridir [2]. Bunun en temel nedeni YSA'larda gizli olan bilginin aktivasyonlar ve nöron bağlantılarına dağılmış olmasıdır. Yapay sinir ağlarında gizli olan bilgiyi kullanıcıların anlayabilmesi için birçok araştırmacı, eğitilmiş YSA'lardan bilgi çıkarımı için zeki algoritmalar geliştirme çabasıdadır. Bu çalışmalar arasında Elalfi vd. [3], genetik algoritma (GA) kullanarak eğitilmiş yapay sinir ağları aracılığıyla veritabanlarından doğru ve anlaşılır kurallar çıkarmak için yeni bir algoritma sunmuşlardır. Optimal kromozomu kodlamış ve ilgili sınıflara ait kuralları elde etmekte kullanmışlardır. Markowska-Kaczmar ve Wnuk-Lipinski [4], sinir ağlarından pareto optimizasyonlu genetik yaklaşıma dayanan kural çıkarımını önermişlerdir. Yöntemlerini bilinen test problemleri üzerinde test etmiş ve sonuçlarını verip tartışmışlardır. Markowska-Kaczmar [5], çalışmada sınıflandırma problemleri için sinir ağlarından kural çıkarım metodunun parametrelerinin son sonuçlardaki etkilerini deneysel çalışmalarla incelemiştir. Metodu, her sınıfın ayrı popülasyonları içerdiği evrimsel yaklaşıma dayanmaktadır. Tokinaga vd. [6], zeki ve açıklayıcı bir değerlendirme sistemi kurmak için Genetik Programlanmaya (GP) dayanan sinir ağı kural çıkarma tekniklerinin kullanımını ele almışlardır. Uygulamaları neticesinde sundukları kural üretim yönteminin performansının karşılaştırılabilir olduğunu ve iyi sonuçlar verdiğini görmüşlerdir. Hruschka ve Ebecken [7], sınıflandırma problemlerinde eğitilen çok katmanlı algılayıcılarla kurallar çıkarmakta bir metot sunmuşlardır. Sundukları yaklaşımı, veri madenciliğinde referans olan dört veri kümesinde ve bir gerçek hayat meteorolojik veri kümesinde deneysel olarak değerlendirmişlerdir. Setiono vd. [8], eğitilmiş sinir ağlarından anlaşılır kural kümeleri çıkarmak üzerine bir çalışma yapmışlardır. Elde ettikleri sonuçlar, algoritmalarının anlaşılır, özlü ve doğru kural kümeleri çıkardığını göstermektedir. Özbakır vd. [9], sınıflandırma problemleri için eğitilmiş yapay sinir ağlarından bilgi çıkarımında karınca koloni optimizasyonunun kullanımı üzerine bir çalışma yapmışlardır. Deneysel çalışmalarını algoritmalarının doğru ve etkin kurallar üretebildiğini göstermektedir. Kahramanlı ve Allahverdi [10], eğitilmiş adaptif sinir ağlarından kurallar çıkarmak için yapay bağımsızlık sistemi algoritması kullanan yeni bir yöntem sunmuşlardır. Algoritmalarını iki referans veri kümesinde değerlendirmişlerdir. Özbakır vd. [11], yapay sinir ağlarının eğitimi ve kural çıkarımını bütünleştirmek için DIFACONN-miner algoritmasını geliştirmişlerdir. Diğer kural çıkarım algoritmalarından farklı olarak önerdikleri algoritmada, eğitim ve kural çıkarımı aşamaları eşzamanlı gerçekleştirilmektedir.

Bu çalışmada yazarlar tarafından 2010 yılında YSA'lardan sınıflandırma kuralları çıkarmak için geliştirilen DIFACONN-miner algoritması [11] ele alınmıştır. DIFACONN-miner algoritması, yapay sinir ağlarının eğitimi için diferansiyel gelişim algoritmasını, sınıflandırma kuralları üretmek için ise tur atan karınca koloni optimizasyonu algoritmasını kullanmaktadır. DE algoritmasının her eğitim adımında, YSA-ağırlıkları TACO algoritmasına, kural üretimi için gönderilmektedir. Daha sonra, YSA yapısının



uygunluğu “YSA hatası”, “kural sayısı” ve “eğitim doğruluğu” olmak üzere üç performans ölçütü içeren çoklu amaç fonksiyonuna göre değerlendirilmektedir. Bu çalışmada, ele alınan algoritmanın performansı altı referans veri kümesinde ve bir gerçek hayat veri kümesinde değerlendirilmiştir. Deneysel sonuçlar algoritmanın doğru ve etkili kurallar üretmekte başarılı olduğunu göstermektedir.

## 2. DIFACONN-MİNER ALGORİTMASI

DIFACONN-miner algoritmasındaki temel düşünce, DE algoritmasını YSA eğitiminde, TACO algoritmasını ise eğitimle eşzamanlı olarak sınıflandırma kuralları çıkarmak için kullanmaktır. Daha önceki yaklaşımlardan farklı olarak, ele alınan algoritma yapay sinir ağlarının eğitimini ve kural çıkarımı aşamalarını bir çatı altında birleştirmektedir. Dolayısıyla, kural çıkarımı için ekstra bir adım gerekmeden, sınıflandırma kuralları doğrudan YSA’lardan elde edilmektedir. Bu, her YSA eğitim adımında ilgili sınıflandırma kurallarının eşzamanlı olarak üretilmesi/değerlendirilmesi ve eğitimin daha doğru sınıflandırma kuralları üretmek için gerçekleştirmesi anlamına gelmektedir. DIFACONN-miner algoritması bir kural çıkarım algoritması olmaktan ziyade, aslında bir kural üretim algoritmasıdır ve birbirine bağlı üç bölümden oluşur. Bunlar;

- DE algoritması ile ileri beslemeli YSA’ların eğitimi
- TACO algoritması ile kural çıkarımı
- Uygunluk değerlendirme

DIFACONN-miner algoritmasının ana çatısını, DE tabanlı eğitim algoritması oluşturmaktadır ve TACO algoritması ilgili ağırlık grupları için kural kümeleri üretmek amacıyla DE algoritmasına dahil edilmiştir. Algoritmanın genel yapısı Şekil 1’de gösterilmiştir. İkili kodlanmış veri kullanıcı girdileri, ağırlık grupları ve kural kümeleri ise DIFACONN-miner algoritması tarafından hesaplanan değerlerdir. Algoritma ile çıkarılan kurallar EĞER-O HALDE şeklinde sınıflandırma kurallarıdır.

### 2.1. Veri Gösterimi

DIFACONN-miner algoritmasında veri kümesinde yer alan örnekler ikili diziler şeklinde kodlanır ve yapay sinir ağlarına girdi olarak gösterilir. Bu gösterim ilk olarak eğitilmiş YSA’lardan GA kullanarak kural çıkarımında Elalfi vd. [3] tarafından kullanılmıştır.

İkili düzen şu şekilde çalışır: veri kümesi  $N$  adet nitelikten oluşsun. Her bir nitelik  $A_n$  ( $n=1, 2, \dots, N$ ),  $m_n$  bir  $A_n$  niteliğinin olası değerlerini göstermek üzere, sabit uzunlukta ikili alt-dizilere ( $a_1, a_2, \dots, a_{mn}$ ) kodlanabilir. İkili kodlama düzenine göre,  $m_n$  elemanlarından sadece biri eğer ilgili nitelik değerine sahipse 1 değerini alabilir ve diğer bütün elemanlar 0’dır. Dolayısıyla,  $LI$  girdi vektör uzunluğu şu şekilde hesaplanır:

$$LI = \sum_{n=1}^N m_n \quad (1)$$

Çıktı sınıf vektörü  $LO_k$  da ( $k=1, 2, \dots, K$ ),  $k$  sınıf sayısını göstermek üzere, sabit uzunlukta ikili dizilere kodlanabilir. Girdi nitelik alt-dizilerinde olduğu gibi, çıktı sınıf vektöründeki  $k$  elemandan sadece biri eğer çıktı vektörü sınıf  $k$ ’ya aitse 1 değerini alabilir ve vektördeki diğer bütün elemanlar 0’dır. Dolayısıyla, yapay sinir ağının çıktı katmanındaki çıktı nöron sayısı  $K$ ’dir.  $C_1, C_2, \dots, C_k$  sınıfları göstermek üzere, çıktı sınıf vektörü eşitlik (2)’deki gibidir.

$$LO_k = \{C_1, C_2, \dots, C_k\} \quad C_1, C_2, \dots, C_k \in \{0,1\} \quad (2)$$

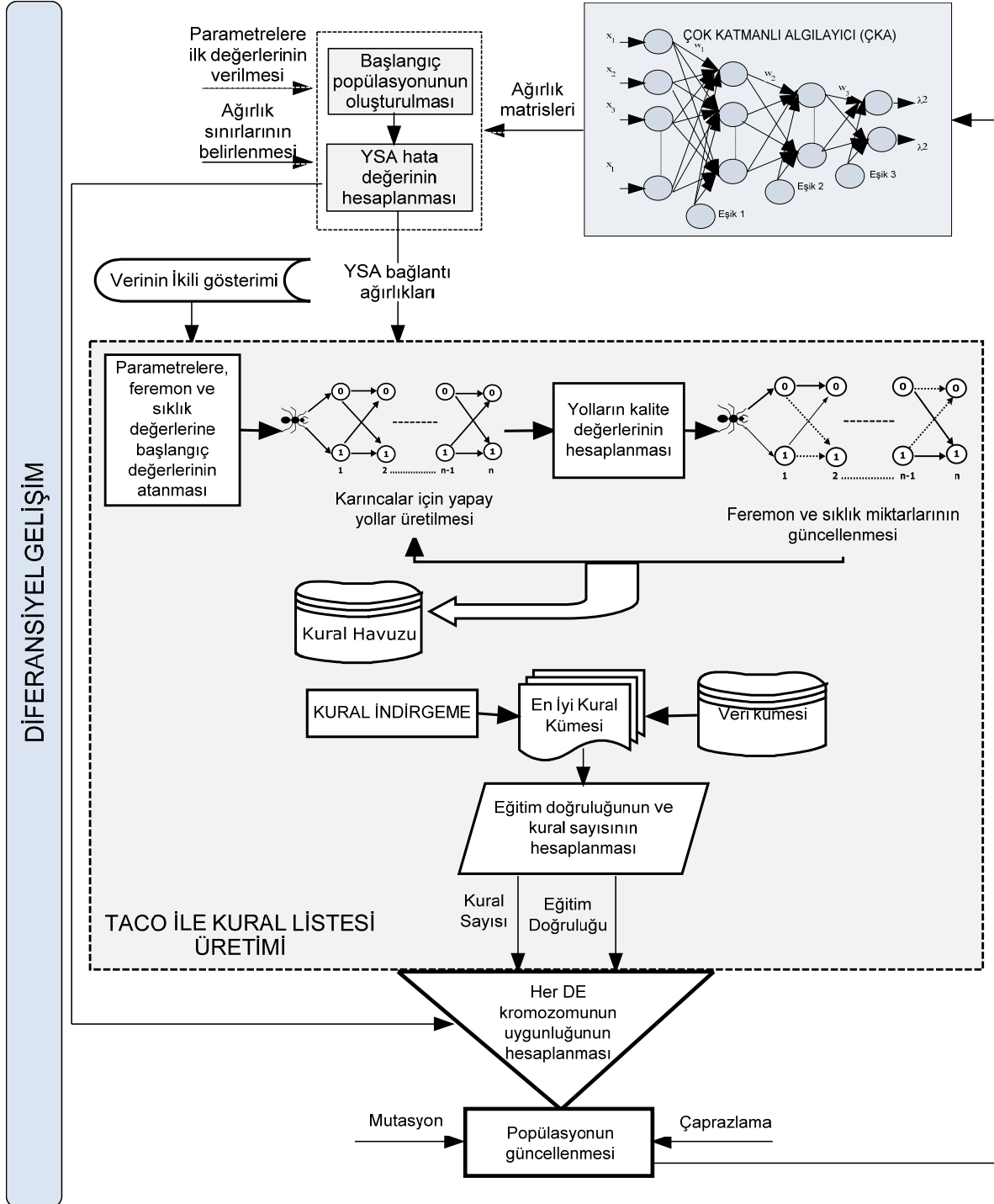
Bu ikili gösterim şekli ayrıca algoritmanın TACO bölümünde, kural gösteriminde kullanılmaktadır.  $LI$  ikili girdi vektörü sayısını,  $K$  ikili çıktı sınıf vektör sayısını göstermek üzere kural uzunluğu  $RL$  eşitlik (3)’deki gibi belirlenir.



$$RL = LI + K$$

(3)

Bu ikili gösterim şekli kesikli nitelikler içeren veri kümeleri için doğrudan kullanılabilir. Ancak sürekli girdiler içeren veri kümeleri için ikili forma dönüştürmeden önce sürekli değerlerin kesiklendirilmesi gerekir.



Şekil 1. DIFACONN-Miner Algoritmasının Genel Yapısı [11]



## 2.2. DE Algoritması ile İleri Beslemeli YSA'ların Eğitimi

Yapay sinir ağları, nöron olarak da isimlendirilen, birbirine bağlı işlem elemanlarından oluşur.  $y_i$  nöron çıktısını,  $x_j$  nörona  $j$ . girdiyi,  $w_{ij}$  nöronla  $x_j$  girdisi arasındaki bağlantı ağırlığını,  $\theta_i$  nöron eşik değerini ve  $f_i$  nöron aktivasyon fonksiyonunu göstermek üzere,  $i$ . nöronun çıktısı eşitlik (4) ile hesaplanmaktadır.

$$y_i = f_i \left( \sum_{j=1}^n x_j w_{ij} + \theta_i \right) \quad (4)$$

YSA'ların uygunluğu için hata fonksiyonu  $E$  en küçüklenerek sağlanır.  $E(t)$   $t$ . iterasyondaki hatayı,  $d_k$  istenen çıktıyı,  $o_k$   $k$ . çıktının mevcut değerini,  $k$  çıktı sayısını ve  $n$  örnek sayısını göstermek üzere, hata fonksiyonu eşitlik (5) ile ifade edilir.

$$E(t) = \frac{1}{n} \sum_{j=1}^n \sum_{k=1}^K (d_{jk} - o_{jk})^2 \quad (5)$$

DIFACONN-miner algoritmasının ana yapısını DE algoritması oluşturmaktadır ve DE, tek gizli katmanlı ileri beslemeli YSA'ların eğitimi için kullanılmaktadır. DE, sürekli uzayda, küresel optimizasyon için kayan-noktalı kodlanmış evrimsel bir algoritmadır. Bu nedenle, tipik ileri beslemeli YSA'ların ağırlık uzayında küresel aramaya uygulanabilir [12].

Birçok eğitim algoritmasında yapay sinir ağlarının hata fonksiyonu, YSA ağırlıklarını optimize ederek en küçüklenmesi gereken amaç fonksiyonudur. DE algoritmasında YSA'ların ağırlık matrisleri birey olarak kullanılmış ve bu ağırlıklar bazı DE işlemleriyle değiştirilmiştir. YSA eğitiminde kullanılan daha önceki çalışmalardan farklı olarak DIFACONN-miner algoritmasında YSA hatasının yanı sıra ilave iki performans ölçütü uygunluk fonksiyonuna eklenmiştir ve ağırlıkları sınıflandırma kuralları şeklinde ifade edebilmek için bur kural üretim algoritması DE eğitim algoritmasına dahil edilmiştir.

**Parametreler:** Yapay sinir ağlarının eğitiminde kullanılan diferansiyel gelişim algoritması parametreleri şöyledir:

- $NP$ : popülasyon büyüklüğü (kromozom sayısı)  $NP \geq 4$  (1, 2, ...,  $p$ )
- $D$ : değişken sayısı (kromozom uzunluğu) (1, 2, ...,  $r$ )
- $CR$ : çaprazlama faktörü [0, 1]
- $G$ : nesil sayısı (1, 2, ...,  $G_{max}$ )
- $F$ : mutasyon oranı (ölçekleme faktörü) (0, 1+)
- $x_{r,p,g}$ :  $g$ . nesil,  $p$ . kromozomun  $r$ . parametresi
- $n_{r,p,g+1}$ : mutasyon ve kromozoma giren ara kromozom
- $u_{r,p,g+1}$ :  $x_{r,p,g}$ 'den sonra gelecek nesil için üretilen kromozom
- $r_{1,2,3}$ : yeni kromozom üretmekte kullanılacak rastgele seçilmiş kromozomlar
- $x_r^{(l)}, x_r^{(u)}$ : parametrenin (ağırlık) alt ve üst sınırları

**Problem Formülasyonu:** Uygunluğu en küçükleyecek problem eşitlik (6) ile ifade edilir.

$$\begin{aligned} & \text{En küçükle Uygunluk} \\ & \text{s.t. } x_r^{(l)} \leq x_r \leq x_r^{(u)} \quad r = 1, 2, 3, \dots, D \end{aligned} \quad (6)$$

**Kodlama ve Başlangıç Popülasyonu:** DE algoritmasında bir kromozomdaki her bir gen, YSA ağırlık matrisindeki bir ağırlığı gösterir. DE algoritmasının mevcut kromozomdan ve birbirlerinden farklı üç kromozoma ihtiyaç duymasından dolayı kromozom sayısı üçten büyük olmalıdır. Ağırlık matrislerini içeren başlangıç popülasyonu eşitlik (7) ile elde edilir.

$$\begin{aligned} & \forall p \leq NP \wedge \forall r \leq D : x_{r,p,g=0} = x_r^{(l)} + rand_r(x_r^{(u)} - x_r^{(l)}) \\ & p = (1, 2, 3, \dots, NP) \quad r = (1, 2, 3, \dots, D) \quad g = 0 \quad rand_r \in [0, 1] \end{aligned} \quad (7)$$



**Kural listesi üretimi:** Başlangıç popülasyonu oluşturulduktan sonra, TACO algoritmasına dayanan bir kural üretme algoritması her nesilde her kromozom için gerçekleştirir. Bu kural üretme algoritmasının çıktısı bir sınıflandırma kural kümesidir. TACO-tabanlı kural üretme algoritması ilerleyen bölümlerde detaylı olarak açıklanmıştır.

**Mutasyon:** Mutasyon eşitlik (8)'e göre kromozom genleri üzerinde rastgele değişiklikler gerçekleştirir. Mutasyonda, mevcut kromozomdan ve birbirlerinden farklı üç kromozom seçilir ve mutasyon bu kromozomlara göre yapılır.

$$\begin{aligned} r_1, r_2, r_3 &\in \{1, 2, \dots, NP\} \quad r_1 \neq r_2 \neq r_3 \neq p \\ \forall r \leq D : n_{r,p,g+1} &= x_{r,r_3,g} + F(x_{r,r_1,g} - x_{r,r_2,g}) \end{aligned} \quad (8)$$

**Çaprazlama:** Çaprazlama ile,  $x_{r,p,g}$  ve elde edilen ara kromozom kullanılarak yeni aday kromozom ( $u_{r,p,g+1}$ ) üretilir. Aday kromozomun genleri CR olasılığı ile ara kromozomdan ve  $1-CR$  olasılığı ile mevcut kromozomdan seçilir. Buradaki  $r=r_{rand}$  en az bir genin yeni üretilen kromozomdan alınmasını garanti etmek için kullanılır. Çaprazlama eşitlik (9)'a göre gerçekleştirilir.

$$\forall r \leq D : x_{r,u,g+1} = \begin{cases} n_{r,p,g+1} & \text{eger } rand[0,1] \leq CR \vee r = r_{Rand} \\ x_{r,p,g} & \text{aksi durumda.} \end{cases} \quad (9)$$

**Seçim:** Gelecek nesle aktarılacak olan en küçük uygunluk değerine sahip kromozom,  $u_{r,p,g+1}$  ve  $x_{r,p,g}$  kromozomlarından eşitlik (10) kullanılarak seçilir.

$$x_{r,p,g+1} = \begin{cases} u_{r,p,g+1} & \text{eger } uygunluk(u_{r,p,g+1}) \leq uygunluk(x_{r,p,g}) \\ x_{r,p,g} & \text{aksi durumda.} \end{cases} \quad (10)$$

**Durdurma Kriteri:** DIFACONN-miner algoritmasının durdurma kriteri en büyük nesil sayısıdır ( $G_{max}$ ). Kural listesi üretimi, mutasyon, çaprazlama ve seçim adımları belirlenen  $G_{max}$  değerine ulaşıncaya kadar tekrarlanır. Problemin en iyi çözümü, son iterasyondaki en küçük uygunluk değerine sahip çözümdür.

### 2.3. TACO Algoritması ile Kural Listesi Üretimi

DIFACONN-miner algoritması, elde edilen her ağırlık matrisi (kromozom) için yazarlar tarafından [9] YSA'daki gizli bilgiyi doğru ve anlaşılır sınıflandırma kurallarına dönüştürmek için geliştirilmiş TACO-tabanlı kural çıkarım algoritmasını kullanarak bir kural listesi üretir. Yazarların önceki çalışmasından [9] farklı olarak DIFACONN-miner algoritmasında duyarlık ve belirlilik ölçütlerine dayanan farklı bir kalite fonksiyonu sınıflandırma kuralları üretmek için kullanılmıştır.

**Parametreler:** Sınıflandırma kuralları üretmek için kullanılan TACO-tabanlı kural çıkarım algoritmasının parametreleri şöyledir:

$M$	: karınca sayısı
$T$	: en büyük iterasyon sayısı
$\rho$	: buharlaşma oranı ( $0 \leq \rho \leq 1$ )
$f$	: sıklık faktörü
$Q$	: pozitif sabit
$P_{ij}$	: alt-yol ( $i \rightarrow j$ ) olasılığı
$f_{ij}$	: alt-yol ( $i \rightarrow j$ )'nin sıklığı
$\tau_{ij}$	: alt-yol ( $i \rightarrow j$ )'nin yapay feromonu
$\Delta \tau_{ij}^k$	: ( $t$ ) ve ( $t+1$ ) zaman aralığında $k$ . yapay karınca tarafından alt-yol ( $i \rightarrow j$ )'ye bırakılan feromon miktarı



$\Delta\tau_{ij}(t, t+1)$  : (t) ve (t+1) zaman aralığında alt-yol (i → j)'ye bırakılacak feromon miktarı

$\tau_{ij}(t+1)$  : (t+1) zamanında (i → j) alt-yolunun feromon miktarı

**Problem Formülasyonu:** DIFACONN-miner algoritmasında sınıflandırma kuralları üretmek için duyarlık ve belirlilik ölçütlerine dayanan bir kalite fonksiyonu kullanılır. Duyarlık ( $S_e$ ), doğru sınıflandırılan pozitif örneklerin oranını, belirlilik ( $S_p$ ) ise doğru sınıflandırılan negatif örneklerin oranını ölçer. Kalite değerini hesaplamakta kullanılan fonksiyon,  $tp$  (doğru pozitif) ve  $tn$  (doğru negatif) doğru sınıflandırmaları;  $fp$  (yanlış pozitif) ve  $fn$  (yanlış negatif) yanlış sınıflandırmaları göstermek üzere eşitlik (11)'de verilmiştir.

$$Kalite = S_e \times S_p = \frac{tp}{tp + fn} \times \frac{tn}{tn + fp} \quad (11)$$

Pozitif kalite değerlerine sahip çözüm dizileri kural indirgeme prosedürü için aday kurallar olarak saklanır. Kalite fonksiyonunu en büyüklemeyle ilgili problem eşitlik (12) ile verilebilir.

$$\begin{aligned} & \text{En büyükle} \quad \text{kalite} \\ & \text{s. t.} \quad x_i = 0 \text{ veya } 1 \end{aligned} \quad (12)$$

**Başlangıç Popülasyonu ve Kodlama:** Kural gösteriminde de daha önce anlatılan ikili gösterim kullanılır. TACO-tabanlı kural üretme algoritmasında başlangıç popülasyonu rastgele üretilir.

**Yapay Karıncalar için Yapay Yolların Üretimi:** TACO algoritmasında yapay yollar olasılık fonksiyonuna göre üretilir. Temel TACO algoritmasının erken yakınsamaya neden olabilecek saf feromon-tabanlı yön seçim stratejisi kullanmasından dolayı DIFACONN-miner algoritmasında farklı bir olasılık fonksiyonu kullanılmıştır. Kullanılan fonksiyon tabu arama algoritmasının hafıza özelliğine dayanmaktadır. Sıklık tabanlı hafıza, bir alt-yolun karıncalar tarafından ne sıklıkla kullanıldığını tutar. Eğer karıncalar bazen yollarını bu prensibe göre seçerlerse, farklı yollar bulabilirler ve küresel en iyiyi yakalayabilirler. Alt-yolun 0 ve 1 (0 → 1) arasında tercih edilmesinin olasılığı hesaplanırken,  $p_{01}$  alt-yol (0 → 1)'in olasılığını,  $\tau_{00}$  ve  $\tau_{01}$  sırasıyla alt-yol (0 → 0) ve alt-yol (0 → 1)'in yapay feromonlarını ve  $f$  sıklık faktörünü ( $f > 1$ ) göstermek üzere eşitlik (13) kullanılır. Eşitlik (13)'e göre, eğer ( $f * f_{01} < f_{00}$ ) durumu sağlanırsa (0 → 1) yolu doğrudan seçilir; aksi takdirde feromon-tabanlı yön seçim stratejisi kullanılır.

$$p_{01}(t) = \begin{cases} 1 & \text{eger } (f * f_{01} < f_{00}) \\ \frac{\tau_{01}}{\tau_{01} + \tau_{00}} & \text{aksi durumda} \end{cases} \quad (13)$$

**Feromon ve Sıklık Miktarlarının Güncellenmesi:** Yapay feromon ve sıklık sırasıyla eşitlik (14) ve (15) kullanılarak hesaplanır. Eşitlik (14)'de  $\Delta\tau_{01}^k$ , k karıncası tarafından (0 → 1) alt-yoluna bırakılan feromon miktarını,  $F_k$ , k. karınca tarafından bulunan çözüm kullanılarak hesaplanan amaç fonksiyon değerini ve Q pozitif sabiti ifade etmektedir. DIFACONN-miner algoritmasında  $F_k$ , eşitlik (11) kullanılarak hesaplanan kalite olarak ele alınmıştır.

$$\Delta\tau_{01}^k(t, t+1) = \begin{cases} Q * F_k & \text{eger karınca alt - yol (0 → 1)'den geçerse} \\ 0 & \text{aksi durumda.} \end{cases} \quad (14)$$

$$\Delta F_{01}^k(t, t+1) = \begin{cases} 1 & \text{eger } i - j \text{ yolu kullanılırsa} \\ 0 & \text{aksi durumda} \end{cases} \quad (15)$$



Bütün karıncalar arama süreçlerini tamamladıktan ve çözümlerini ürettikten sonra,  $(t)$  ve  $(t+1)$  zaman aralığında alt-yol  $(0 \rightarrow 1)$ 'e yapıştırılacak feromon ve sıklık miktarları eşitlik (16) ve (17) kullanılarak hesaplanır.

$$\Delta \tau_{01}(t, t+1) = \sum_{k=1}^N \Delta \tau_{01}^k(t, t+1) \quad (16)$$

$$\Delta F_{01}(t, t+1) = \sum_{k=1}^M \Delta F_{01}^k(t, t+1) \quad (17)$$

$(t+1)$  zamanında aynı alt-yolun feromon ve sıklık miktarları eşitlik (18) ve (19) ile güncellenir. Eşitlik (18)'de  $\rho$ , buharlaşma parametresi olarak isimlendirilen bir katsayıdır  $(0 \leq \rho < 1)$ .

$$\tau_{01}(t+1) = \rho \tau_{01}(t) + \Delta \tau_{01}(t, t+1) \quad (18)$$

$$F_{ij}(t+1) = F_{ij}(t) + \Delta F_{ij}(t, t+1) \quad (19)$$

**Durdurma Kriteri ve Kural İndirgeme:** TACO-tabanlı kural üretme algoritması belirlenen iterasyon sayısı  $(T)$  kadar çalıştırılır. Daha sonra mevcut ağırlık matrislerinin kural listelerini elde etmek için kural indirgeme gerçekleştirilir. Bu adımda kapsama algoritması [13], aday kuralların kalite değerlerine göre kural indirgemek için uygulanır. En iyi aday kuraldan başlayarak, eğitim kümesindeki tüm örnekler sınıflandırılıncaya veya değerlendirilecek kural kalmayıncaya kadar bir kural listesi üretilir. İkili gösterim düzeninden dolayı, bir kuraldaki anlamsız bölümler kural yapısından çıkarılabilir. Dolayısıyla alt-dizilerinde aynı ikili değeri içeren nitelikler kural dizisinden çıkarılır. Böylece daha kısa ve anlaşılır kurallar elde edilebilir. Eğitim doğruluğu ve kural sayısı da bu aşamada hesaplanmaktadır. Bu iki ölçüt, YSA hatasının yanı sıra uygunluk hesaplamada performans ölçütleri olarak kullanılmaktadır.

## 2.4. Uygunluk Değerlendirme

Algoritmanın tamamının uygunluğu, "eğitim doğruluğu", "kural sayısı" ve "YSA hatası" olmak üzere üç performans kriteri içeren çoklu amaç yaklaşımı kullanılarak belirlenir. Bu performans ölçütleri içinde "eğitim doğruluğu" en büyüklenmeli, "kural sayısı" ve "YSA hatası" ise en küçüklenmelidir. Bu nedenle bu üç kriter, minimum sapma metodu (MDM) kullanılarak tek bir amaç fonksiyonunda birleştirilmiştir. MDM metodu, tekil amaçların kısmi sapmaları toplamını minimize eden en iyi uzlaşma çözümü bulmayı amaçlar [14].

DIFACONN-miner algoritmasının uygunluğu,  $TA$ ,  $E$  ve  $RN$  sırasıyla "eğitim doğruluğu", "YSA hatası" ve "kural sayısı"ni göstermek üzere eşitlik 20 ile hesaplanır. Bu eşitlikte  $f_1^0(TA)$ ,  $f_2^0(E)$  ve  $f_3^0(RN)$   $TA$ ,  $E$  ve  $RN$ 'nin kabul edilebilir en alt düzeylerini ifade etmektedir.  $f_1^{\max}(TA)$ , eğitim doğruluğunun hedef değerini (% 100 olarak alınmıştır),  $f_2^{\min}(E)$ , YSA hatasının hedef değerini (0.01 olarak alınmıştır) ve  $f_3^{\min}(RN)$ , ele alınan problemin sınıf sayısı olarak ayarlanan kural sayısının hedef değerini göstermektedir.

$$Uygunluk = \frac{f_1^{\max}(TA) - f_1(TA)}{f_1^{\max}(TA) - f_1^0(TA)} + \frac{f_2^{\min}(E) - f_2(E)}{f_2^{\min}(E) - f_2^0(E)} + \frac{f_3^{\min}(RN) - f_3(RN)}{f_3^{\min}(RN) - f_3^0(RN)} \quad (20)$$

Eşitlik (20) ile verilen uygunluk fonksiyonunu en küçükleyerek, en iyi sınıflandırma kural kümesini bulmak için kullanılan DIFACONN-miner algoritması Şekil 2'de özetlenmiştir.



**BEGIN**

Parametrelere,  $x^{(l)}$  ve  $x^{(h)}$  parametre sınırlarına değerlerini ver.

Eşitlik (7)'ye göre başlangıç popülasyonunu oluştur.

**REPEAT**

Popülasyondaki her kromozom için gerçekleştir

**BEGIN**

Birbirinden ve mevcut kromozomdan farklı üç kromozom ( $r_1, r_2, r_3$ ) seç.

Eşitlik (8)'e göre mutasyon gerçekleştir

Eşitlik (9)'a göre çaprazlama yap

Eşitlik (5)'i kullanarak ağın hatasını hesapla

Kural listesi, eğitim doğruluğu ve kural sayısını elde etmek için TACO-tabanlı kural üretme algoritmasını yeni kromozoma uygula.

Eşitlik (20)'ye göre kromozomun uygunluğunu hesapla.

Eşitlik (10)'a göre seçim yap.

**END**

UNTIL en büyük nesil sayısı.

**END****Şekil 2.** DIFACONN-Miner Algoritmasının Temel Adımları**3. DIFACONN-Miner ALGORİTMASI'NIN DEĞERLENDİRİLMESİ****3.1. Sınıflandırma Referans Veri Kümeleri**

DIFACONN-miner algoritmasının performansı ilk olarak UCI (University of California at Irvine) makine öğrenme deposundan (<http://mllearn.ics.edu/MLRepository.html>) alınan altı veri kümesi üzerinde değerlendirilmiştir. Tablo 1 veri kümelerinin temel özelliklerini göstermektedir. Bazı veri kümeleri eksik nitelikler içermektedir, fakat bu nitelikler veri kümesinden tamamen çıkarılmamış, sadece eksik değer içeren örneklerdeki ilgili nitelikler çıkarılmıştır. Sürekli nitelikler içeren veri kümeleri için kesiklendirme Weka (Waikato Environment for Knowledge Analysis) 3.5 veri madenciliği paket programı kullanılarak basit gruplama ile gerçekleştirilmiştir. Tablo 2, veri kümelerinin niteliklerini ve ilgili ikili değişkenlerini göstermektedir.

**Tablo 1.** Veri Kümelerinin Temel Özellikleri

Veri kümesi	Örnek sayısı	Kategorik nitelik sayısı	Sürekli nitelik sayısı	Eksik nitelik değeri	Sınıf sayısı
ECG	132	1	6	Var	2
Heart-C	303	8	5	Var	5
LBC	286	9	-	Var	2
Nursery	12.960	8	-	Yok	5
Pima	768	-	8	Yok	2
Vote	435	16	-	Var	2

**Tablo 2.** Veri Kümelerinin Nitelikleri ve İlgili İkili Değişkenler

İkili nitelikler (26)	ECG nitelikleri	İkili nitelikler (27)	Nursery nitelikleri
X <sub>1</sub> , X <sub>2</sub> , X <sub>3</sub> , X <sub>4</sub>	Kalp krizi esnasındaki yaş (-; 50], (50; 60], (60; 70], (70; -)	X <sub>1</sub> , X <sub>2</sub> , X <sub>3</sub>	Ebeveynler {sıradan},{iddialı},{çok-iddialı}
X <sub>5</sub> , X <sub>6</sub>	Perikardiyal efüzyon {0}, {1}	X <sub>4</sub> , X <sub>5</sub> , X <sub>6</sub> , X <sub>7</sub> , X <sub>8</sub>	Hemşirelik {uygun},{az-uygun},{uygunsuz}, {kritik},{çok-kritik}
X <sub>7</sub> , X <sub>8</sub> , X <sub>9</sub> , X <sub>10</sub>	Fraksiyonel kısalma (-, 0, 15], (0, 15; 0, 3], (0, 3; 0, 45], (0, 45; -)	X <sub>9</sub> , X <sub>10</sub> , X <sub>11</sub> , X <sub>12</sub>	form {eksizsiz},{tamamladı},{eksik}, {teşvik}
X <sub>11</sub> , X <sub>12</sub> , X <sub>13</sub> , X <sub>14</sub>	Epps (-; 9], (9; 13], (13; 17], (17; -)	X <sub>13</sub> , X <sub>14</sub> , X <sub>15</sub> , X <sub>16</sub>	çocuk {1},{2},{3},{daha fazla}
X <sub>15</sub> , X <sub>16</sub> , X <sub>17</sub> , X <sub>18</sub>	Lvdd (-; 4], (4; 5], (5; 6], (6; -)	X <sub>17</sub> , X <sub>18</sub> , X <sub>19</sub>	ev {uygun},{az-uygun},{kritik}
X <sub>19</sub> , X <sub>20</sub> , X <sub>21</sub> , X <sub>22</sub>	Duvar hareket skoru (-; 10], (10; 13], (13; 17], (17; -)	X <sub>20</sub> , X <sub>21</sub>	finans {uygun},{uygunsuz}
X <sub>23</sub> , X <sub>24</sub> , X <sub>25</sub> , X <sub>26</sub>	Duvar hareket indeksi (-; 1], (1; 1, 5], (1, 5; 2], (2; -)	X <sub>22</sub> , X <sub>23</sub> , X <sub>24</sub>	Sosyal {uygun değil},{az-uygun}, {problemlili}
İkili nitelikler (41)	LBC nitelikleri	X <sub>25</sub> , X <sub>26</sub> , X <sub>27</sub>	Sağlık {önerilir},{öncelik},{önerilmez}
X <sub>1</sub> , X <sub>2</sub> , X <sub>3</sub> , X <sub>4</sub> , X <sub>5</sub> , X <sub>6</sub>	Yaş (10; 19], (20; 29], (30; 39], (40; 49], (50; 59], (60; 69], (70; 79], (80; 89], (90; 99]	İkili nitelikler (32)	Pima nitelikleri
X <sub>7</sub> , X <sub>8</sub> , X <sub>9</sub>	Menapoz {t40},{ge40},{premeno}	X <sub>1</sub> , X <sub>2</sub> , X <sub>3</sub> , X <sub>4</sub>	Pregnant sayısı (-; 4, 25], (4, 25; 8, 5], (8, 5; 12, 75], (12, 75; -)
X <sub>10</sub> , X <sub>11</sub> , X <sub>12</sub> , X <sub>13</sub> , X <sub>14</sub> , X <sub>15</sub> , X <sub>16</sub> , X <sub>17</sub> , X <sub>18</sub> , X <sub>19</sub> , X <sub>20</sub>	Tümör boyutu (0; 4], (5; 9], (10; 14], (15; 19], (20; 24], (25; 29], (30; 34], (35; 39], (40; 44], (45; 49], (50; 59]	X <sub>5</sub> , X <sub>6</sub> , X <sub>7</sub> , X <sub>8</sub>	Glukoz plazma konsantrasyonu ... (-; 49, 75], (49, 75; 99, 5], (99, 5; 149, 25], (149, 25; -)
X <sub>21</sub> , X <sub>22</sub> , X <sub>23</sub> , X <sub>24</sub> , X <sub>25</sub> , X <sub>26</sub> , X <sub>27</sub>	Inv-noktalar (0; 2], (3; 5], (6; 8], (9; 11], (12; 14], (15; 23], (24; 26]	X <sub>9</sub> , X <sub>10</sub> , X <sub>11</sub> , X <sub>12</sub>	Diyastolik kan basıncı (-; 30, 5], (30, 5; 61], (61; 91, 5], (91, 5; -)
X <sub>28</sub> , X <sub>29</sub>	Nokta-başları {yes}, {no}	X <sub>13</sub> , X <sub>14</sub> , X <sub>15</sub> , X <sub>16</sub>	Triseps deri katman inceliği (-; 24, 75], (24, 75; 49, 5], (49, 5; 74, 25], (74, 25; -)
X <sub>30</sub> , X <sub>31</sub> , X <sub>32</sub>	Deg-malig {1}, {2}, {3}	X <sub>17</sub> , X <sub>18</sub> , X <sub>19</sub> , X <sub>20</sub>	2-saat serum insulin (-; 211, 5], (211, 5; 423], (423; 634, 5], (634, 5; -)
X <sub>33</sub> , X <sub>34</sub>	Göğüs {sol}, {sağ}	X <sub>21</sub> , X <sub>22</sub> , X <sub>23</sub> , X <sub>24</sub>	Vücut kitle indeksi (-; 16, 775], (16, 775; 33, 55], (33, 55; 50, 325], (50, 325; -)
X <sub>35</sub> , X <sub>36</sub> , X <sub>37</sub> , X <sub>38</sub> , X <sub>39</sub>	Göğüs-bölümü {sol-üst},{sol-alt},{sağ-üst},{sağ-alt},{merkez}	X <sub>25</sub> , X <sub>26</sub> , X <sub>27</sub> , X <sub>28</sub>	Diyabet soy seceresi (-; 0, 6635], (0, 6635; 1, 249], (1, 249; 1, 8345], (1, 8345; -)
X <sub>40</sub> , X <sub>41</sub>	Parlak {evet}, {hayır}	X <sub>29</sub> , X <sub>30</sub> , X <sub>31</sub> , X <sub>32</sub>	Yaş (-; 36], (36; 51], (51; 66], (66; -)
İkili nitelikler (32)	Vote nitelikleri	İkili nitelikler (38)	Heart-Cnitelikleri
X <sub>1</sub> , X <sub>2</sub>	Özür-lü-çocuk {evet}, {hayır}	X <sub>1</sub> , X <sub>2</sub> , X <sub>3</sub>	Yaş (-; 45], (45; 61], (61; -)
X <sub>3</sub> , X <sub>4</sub>	Su-projesi-maliyet-paylaşımı {evet}, {hayır}	X <sub>4</sub> , X <sub>5</sub>	Cinsiyet {erkek},{kadın}
X <sub>5</sub> , X <sub>6</sub>	Bütçe-kararını-kabul-etme {evet}, {hayır}	X <sub>6</sub> , X <sub>7</sub> , X <sub>8</sub> , X <sub>9</sub>	Cp {1},{2},{3},{4}
X <sub>7</sub> , X <sub>8</sub>	Doktor-ücret-dondurma {evet}, {hayır}	X <sub>10</sub> , X <sub>11</sub> , X <sub>12</sub>	Trestbps (-; 129, 333], (129, 333; 164, 667], (164, 667; -)
X <sub>9</sub> , X <sub>10</sub>	El-salvador-yardımları {evet}, {hayır}	X <sub>13</sub> , X <sub>14</sub> , X <sub>15</sub>	Chol (-; 272], (272; 418], (418; -)
X <sub>11</sub> , X <sub>12</sub>	Okuldaki-dini-gruplar {evet}, {hayır}	X <sub>16</sub> , X <sub>17</sub>	Fbs {doğru},{yanlış}
X <sub>13</sub> , X <sub>14</sub>	Anti-uydu-test-yasağı {evet}, {hayır}	X <sub>18</sub> , X <sub>19</sub> , X <sub>20</sub>	Rectecg {0},{1},{2}
X <sub>15</sub> , X <sub>16</sub>	Nikaragua-kontrasına-destek {evet}, {hayır}	X <sub>21</sub> , X <sub>22</sub> , X <sub>23</sub>	Thalach (-; 114, 667], (114, 667; 158, 33], (158, 33; -)



İkili nitelikler (32)	Vote nitelikleri	İkili nitelikler (38)	Heart-Cnitelikleri
X <sub>17</sub> , X <sub>18</sub>	Mx-füze {evet}, {hayır}	X <sub>24</sub> ,X <sub>25</sub>	Exang {evet},{hayır}
X <sub>19</sub> , X <sub>20</sub>	Göç {evet}, {hayır}	X <sub>26</sub> , X <sub>27</sub> , X <sub>28</sub>	Oldpeak (-;2, 067],[2,067;4,133],[4,133;-)
X <sub>21</sub> , X <sub>22</sub>	Synfuel-kurum-kesintisi {evet}, {hayır}	X <sub>29</sub> ,X <sub>30</sub> ,X <sub>31</sub>	Slope {1},{2},{3}
X <sub>23</sub> , X <sub>24</sub>	Eğitim-harcaması {evet}, {hayır}	X <sub>32</sub> ,X <sub>33</sub> , X <sub>34</sub> ,X <sub>35</sub>	Ca {0},{1},{2},{3}
X <sub>25</sub> , X <sub>26</sub>	Superfund-davan-hakkı {evet}, {hayır}	X <sub>36</sub> ,X <sub>37</sub> , X <sub>38</sub>	Thal {3},{6},{7}
X <sub>27</sub> , X <sub>28</sub>	Suç {evet}, {hayır}		
X <sub>29</sub> , X <sub>30</sub>	Gümrüksüz-ihracat {evet}, {hayır}		
X <sub>31</sub> , X <sub>32</sub>	Güney-Afrika-ihracat- yönetim-yasası {evet}, {hayır}		

**Veri kümelerinin ayrımı:** ECG veri kümesindeki 132 örneğin 58'i eksik sınıf değeri içermektedir ve bu örnekler veri kümesinden çıkarılarak kalan 74 örnek sınıflandırma için kullanılmıştır. Bu veri kümesinde örnek sayısının az olmasından dolayı, verinin % 66'sı eğitim ve kalanı test olmak üzere tek eğitim ve test veri kümesine ayrılmıştır. Diğer beş veri kümesine on-katlı çapraz-doğrulama prosedürü uygulanmıştır. Her veri kümesi, on alt veri kümesine ayrılmış ve DIFACONN-miner algoritması her ayırım için bir kez çalıştırılmıştır. Her çalıştırmada farklı bir parça test kümesi olarak, kalan 9 parça ise gruplandırılarak eğitim kümesi olarak kullanılmıştır. Eğitim kümesi algoritmaya iyi öğrenme kabiliyeti sağlamak için algoritmanın eğitiminde, test kümesi ise ele alınan algoritma tarafından elde edilen son kural kümesinin genelleme yeteneğini değerlendirmek için kullanılmıştır.

**Performans ölçütleri:** Bu çalışmada, ortalama kural sayısı ve test doğruluğu performans ölçütü olarak ele alınmıştır. Doğruluk, sınıflandırıcının doğru sonuçlar elde etmekteki yeteneğini gösterir ve eşitlik (22) ile hesaplanır. ECG veri kümesi tek eğitim ve test kümesine ayrıldığından dolayı, bu veri kümesindeki tahminleyici doğruluk aynı eğitim ve test veri kümesinde algoritmanın on kez çalıştırılmasıyla elde edilmiştir. Test kümesindeki tahminleyici doğruluk, on çalıştırmanın ortalaması alınarak hesaplanmış ve tüm veri kümeleri için tahminleyici doğruluk olarak sunulmuştur. Ayrıca ilgili tahminleyici doğrulukların standart sapmaları da hesaplanmıştır. Eğer bir kural değerlendirilirse, bu kural ile ilgili doğruluk ölçütü eşitlik (21) ile belirlenir, eğer bir kural kümesi ele alınırsa bu durumda doğruluk eşitlik (22) ile hesaplanır.

$$\text{Dogruluk} = \frac{tp + tn}{tp + tn + fp + fn} \quad (21)$$

$$\text{Dogruluk} = \frac{\text{Kural kumesi tarafından dogru sin iflandiril an ornek sayisi}}{\text{Test veri kume sin deki ornek sayisi}} \quad (22)$$

**Parametre değerleri:** Ele alınan bütün veri kümelerinde 5 gizli katman nörona sahip üç katmanlı ileri beslemeli YSA kullanılmıştır. DIFACONN-miner algoritmasının parametreleri Tablo 3'de gösterilmektedir. Bu tabloda  $NP$ ,  $G$ ,  $CR$ ,  $F$ ,  $x^{(l)}$  ve  $x^{(u)}$ ,  $M$ ,  $T$ ,  $f$ ,  $\rho$  ve  $Q$  sırasıyla popülasyon büyüklüğü, nesil sayısı, çaprazlama faktörü, mutasyon katsayısı, değişkenlerin alt ve üst sınırları, karınca sayısı, iterasyon sayısı, sıklık faktörü, buharlaşma parametresi ve  $Q$  sabitini ifade etmektedir.

**Tablo 3.** DIFACONN-miner algoritması parametre değerleri

$NP$	$G$	$CR$	$F$	$x^{(l)}$	$x^{(u)}$	$M$	$T$	$f$	$P$	$Q$
5	50	0.8	1.2	-1	1	50	50	2	0.8	10



**Performans karşılaştırmaları:** Tablo 4, DIFACONN-miner algoritmasının her bir veri kümesi için on-katlı-çapraz doğrulama sonuçlarını göstermektedir. Tabloda test veri kümesindeki minimum, ortalama ve maksimum tahminleyici doğruluklar, standart sapmalar ve kural sayıları verilmektedir.

**Tablo 4.** DIFACONN-miner algoritmasının tahminleyici doğrulukları

Veri kümesi	Min (%)	Ortalama (%)	Maks (%)	Standart sapma	Ortalama kural sayısı
ECG	96.00	97.45	100.00	0.021	8.45
Heart-C	90.00	94.08	96.67	0.030	38.10
LBC	92.86	97.88	100.00	0.025	26.60
Nursery	99.42	99.98	100.00	0.000	37.00
Pima	83.00	94.14	100.00	0.067	9.30
Vote	95.35	98.15	100.00	0.018	9.20

Tablo 4'den görülebileceği gibi DIFACONN-miner algoritması doğru ve özlu kurallar üretebilmektedir. ECG veri kümesi için çıkarılan örnek bir veri kümesi Tablo 5'de verilmektedir. Eşitlik (11) ile verilen kalite değerinin yanı sıra güvenilirlik ve destek ölçütleri de üretilen kuralların doğruluğunu değerlendirmek için tabloda verilmiştir. "Güvenilirlik ölçütü" bir kuralın kesinliğini, "destek ölçütü" ise bir kuralın kapsamını ölçer. Güvenilirlik ve destek ölçütleri eşitlik (23) ve (24) kullanılarak hesaplanır. Eşitlik (24)'deki  $N$  eğitim kümesindeki toplam örnek sayısını ifade etmektedir.

$$\text{güvenilirlik} = \frac{tp}{tp + fp} \quad (23)$$

$$\text{destek} = \frac{tp + fp}{N} \quad (24)$$

**Tablo 5.** ECG Veri Kümesi İçin Örnek Veri Kümesi

Kural	Kalite	Güvenilirlik	Destek
1 <b>EĞER</b> perikardiyal efüzyon=0 <b>VE</b> duvar hareket indeksi $\leq$ 2 <b>O HALDE</b> sınıf 0	0.476	0.818	0.673
2 <b>EĞER</b> perikardiyal efüzyon=0 ve (duvar hareket indeksi $\leq$ 1,5 <b>VEYA</b> 2<duvar hareket indeksi $\leq$ 3) <b>O HALDE</b> sınıf 0	0.425	0.774	0.663
3 <b>EĞER</b> perikardiyal efüzyon=1 <b>O HALDE</b> sınıf 1	0.494	0.545	0.224
4 <b>EĞER</b> duvar hareket indeksi $\leq$ 2 <b>O HALDE</b> sınıf 0	0.488	0.762	0.857
5 <b>EĞER</b> perikardiyal efüzyon=0 <b>VE</b> (Eps $\leq$ 9 <b>VEYA</b> 17<Eps $\leq$ 40) <b>VE</b> (4<Lvdd $\leq$ 6,73) <b>VE</b> (duvar hareket indeksi $\leq$ 1 <b>VEYA</b> 2<duvar hareket indeksi $\leq$ 3) <b>O HALDE</b> sınıf 1	0.497	0.444	0.184
6 <b>EĞER</b> perikardiyal efüzyon=0 <b>VE</b> (1<duvar hareket indeksi $\leq$ 3) <b>O HALDE</b> sınıf 1	0.443	0.280	0.510

Test Doğruluğu= 100 %

DIFACONN-miner algoritmasının referans veri kümeleri üzerindeki performansı iki aşamada gerçekleştirilmiştir. İlk aşamada algoritma, Weka 3.5 veri madenciliği yazılımındaki dört popüler kural tabanlı algoritma ile karşılaştırılmıştır. Bu algoritmalar "C4.5" [15], "PART" [16], "DecisionTable" [17] ve "NBTree" [18] algoritmasıdır. Tablo 6, test veri kümesindeki minimum, ortalama ve maksimum doğrulukları, standart sapma ve ortalama kural sayılarını göstermektedir. Tablodan görülebileceği gibi DIFACONN-miner algoritması daha yüksek doğruluklarla diğer algoritmalarla üstündür. DIFACONN-miner algoritmasının karşılaştırılan algoritmalarla göre ortalama doğrulukları oldukça yüksektir ve standart sapmaları daha düşüktür. Eğer çıkarılan ortalama kural sayısı dikkate alınırsa, DIFACONN-miner algoritması, Nursery veri kümesinde diğer algoritmalarla daha kısa kural kümeleri bulmaktadır.

**Tablo 6.** Karşılaştırma Sonuçları I

Veri kümesi		NBTree	DecisionTable	PART	C4.5	DIFACONN-miner
ECG	Minimum	57.69	48.00	60.00	60.00	<b>96.00</b>
	Ortalama	67.45	65.48	71.39	70.17	<b>97.45</b>
	Maksimum	80.77	76.00	80.77	80.77	<b>100.00</b>
	Standart sapma	6.88	8.97	6.62	7.08	<b>0.02</b>
	Ortalama kural sayısı	<b>2.3</b>	3.3	4.0	3.2	8.5
Heart-C	Minimum	35.48	50.00	38.71	41.94	<b>90.00</b>
	Ortalama	53.18	54.82	55.51	52.52	<b>94.08</b>
	Maksimum	63.33	63.33	73.33	63.33	<b>96.67</b>
	Standart sapma	9.09	4.28	9.51	6.80	<b>0.03</b>
	Ortalama kural sayısı	<b>7.9</b>	10.1	40.3	38.7	38.1
LBC	Minimum	58.62	67.86	65.52	67.86	<b>92.86</b>
	Ortalama	72.75	75.53	70.99	75.16	<b>97.88</b>
	Maksimum	86.21	82.76	72.41	86.21	<b>100.00</b>
	Standart sapma	6.96	4.64	2.18	5.10	<b>0.03</b>
	Ortalama kural sayısı	<b>2.1</b>	14.6	17.7	9.1	26.6
Nursery	Minimum	96.53	94.06	98.69	96.14	<b>99.42</b>
	Ortalama	97.49	94.78	99.21	97.05	<b>99.98</b>
	Maksimum	98.15	95.60	99.61	97.69	<b>100.00</b>
	Standart sapma	0.50	0.60	0.28	0.47	<b>0.00</b>
	Ortalama kural sayısı	149.1	1053	191.9	352.9	<b>37</b>
Pima	Minimum	66.23	67.53	64.94	64.94	<b>83.00</b>
	Ortalama	74.36	72.27	74.49	73.84	<b>94.14</b>
	Maksimum	85.53	77.92	79.22	81.82	<b>100.00</b>
	Standart sapma	6.68	3.56	5.09	5.66	<b>0.07</b>
	Ortalama kural sayısı	<b>3.2</b>	51.5	7.5	19.2	9.3
Vote	Minimum	88.37	84.09	84.09	88.94	<b>95.35</b>
	Ortalama	94.26	95.41	95.06	96.33	<b>98.15</b>
	Maksimum	97.73	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
	Standart sapma	3.12	4.92	4.21	3.42	<b>0.02</b>
	Ortalama kural sayısı	9.8	29.1	17.8	<b>5.8</b>	9.2

Karşılaştırmaların ikinci aşamasında DIFACONN-miner algoritmasının performansı güncel literatürde yer alan çeşitli kural tabanlı sınıflandırıcılarla karşılaştırılmıştır. Karşılaştırma sonuçları Tablo 7'de verilmektedir. Tablodan da görüldüğü gibi DIFACONN-miner algoritması altı veri kümesinde de diğer algoritmalarla daha iyi tahminleyici doğruluklara sahip kurallar elde etmiştir. Algoritma, ayrıca ortalama kural sayısı bakımından da karşılaştırılan algoritmalarla rekabetçi durumdadır. DIFACONN-miner algoritması diğer algoritmalarla karşılaştırıldığında ECG ve Vote veri kümelerinde oldukça kısa kural kümeleri elde etmiştir.

**Tablo 7.** Karşılaştırma Sonuçları II

Veri kümesi	Algoritma	Ortalama doğruluk (%)	Ortalama kural sayısı	Referans
ECG	AIS kullanarak adaptif YSA'lerden kural çıkarımı	94.59	24.0	Kahramanlı ve Allahverdi [10]
	TACO-miner	96.40	13.5	Özbakır vd. [9]
	<b>DIFACONN-miner</b>	<b>97.45</b>	<b>8.5</b>	
Heart-C	DOEA	79.84	<b>6.0</b>	Tan vd. [19]
	CORE	80.77	6.2	Tan vd. [20]
	RMR	84.74	75.0	Thabtah ve Cowling [21]
	GARC	81.11	12.0	Chen vd. [22]
	Unordered Rule Set Ant Miner	64.84	11.0	Smaldon ve Freitas [23]
	MEPAR-miner	87.78	-	Baykasoğlu ve Özbakır [24]
	TACO-miner	87.45	67.3	Özbakır vd. [9]
<b>DIFACONN-miner</b>	<b>94.08</b>	<b>38.1</b>		
LBC	AIS kullanarak adaptif YSA'lerden kural çıkarımı	92.31	59.0	Kahramanlı ve Allahverdi [10]
	DOEA	78.91	5.0	Tan vd. [19]
	GPc	70.65	12.0	"
	MEPAR-miner	90.63	-	Baykasoğlu ve Özbakır [24]
	Unordered Rule Set Ant Miner	78.42	6.1	Smaldon ve Freitas [23]
	CORE	75.41	<b>4.3</b>	Tan vd. [20]
	TACO-miner	95.12	35.6	Özbakır vd. [9]
<b>DIFACONN-miner</b>	<b>97.88</b>	<b>26.6</b>		
Nursery	MEPAR-miner	95.83	-	Baykasoğlu ve Özbakır [24]
	SGA	76.20	-	Dehuri ve Mall [25]
	NPGA	78.25	-	"
	INPGA	76.65	<b>7.0</b>	"
	TACO-miner	97.24	81.9	Özbakır vd. [9]
<b>DIFACONN-miner</b>	<b>99.98</b>	<b>37.0</b>		
Pima	RMR	78.05	65.0	Thabtah ve Cowling [21]
	CFS Approx	75.90	-	Hoffmann vd. [26]
	GFS Desc 5 FS	76.00	-	"
	GFS Desc 7 FS	76.60	-	"
	RN	73.92	6.5	Li ve Wang [27]
	RS	73.22	<b>6.0</b>	"
	GARC	73.83	<b>6.0</b>	Chen vd. [22]
TACO-miner	88.66	14.0	Özbakır vd. [9]	
<b>DIFACONN-miner</b>	<b>94.14</b>	<b>9.3</b>		
Vote	GARC	89.67	32.0	Chen vd. [22]
	TACO-miner	90.39	25.4	Özbakır vd. [9]
	<b>DIFACONN-miner</b>	<b>98.15</b>	<b>9.2</b>	

### 3.2. Gerçek Hayat Uygulaması

DIFACONN-miner algoritması, referans sınıflandırma problemlerinin yanı sıra, büyük bir tekstil firmasındaki hata faktörlerini sınıflandırmak ve analiz etmek için uygulanmıştır [28]. Böylece, firmadaki hatalı ürün miktarını azaltarak veri madenciliği ile sürdürülebilir bir üretim sağlamak amaçlanmıştır. Tekstil firmasının profesyonel bir bilgi toplama sistemi vardır ve detaylı işlem verileri veritabanına gerçek zamanda transfer edilmektedir. Makinelerin teknik bilgileri, ürün tipleri, hata türleri ve oranları, makine duruşları ve zaman aralıkları, makine arızaları ve sebepleri gibi pek çok veri veritabanlarında depolanmaktadır.

Firmada en çok meydana gelen hatalardan “duruş-kalkış”, “düşüm” ve “kafes” kalite hataları, ortaya çıkış nedenlerini belirlemek amacıyla ele alınmıştır. Bu üç hata türüyle ilgili nitelikler üretim uzmanlarıyla yoğun görüşmeler yapılarak belirlenmiştir. Niteliklerle ilgili veriler farklı veritabanlarından toplanmış ve tek bir veritabanında birleştirilmiştir. Kalite hataları üzerinde etkisi olabilecek nitelikler Tablo 8’de gösterilmektedir.

**Tablo 8.** Nitelik Karakteristikleri

No	Faktör	Tip	İkili değişken sayısı
1	Tip kodu	Kesikli	10
2	Atkı kopuş	Sürekli	3
3	Çözümlü kopuş	Sürekli	3
4	Toplam atkı kopuş/100 000m	Sürekli	3
5	Haşıl makine	Kesikli	4
6	İndigo makine	Kesikli	4
7	Sarma kopuş	Sürekli	3
8	Sarma kopuş/1.000.000m	Sürekli	3
9	Açma kopuş	Sürekli	3
10	Açma kopuş/1.000.000m	Sürekli	3
11	Hamut	Kesikli	5
12	Çapraz	Kesikli	5
13	Makine grubu	Kesikli	3
14	Hata oranı /100 m <sup>2</sup>	Kesikli	4

Ele alınan hata tiplerinden duruş-kalkış ve düğüm veri kümeleri 230, kafes veri kümesi ise 143 örnek içermektedir. Verilerin rastgele seçilen % 66'sı eğitim kümesi, kalanı ise test kümesi olarak kullanılmıştır. Yapay sinir ağı yapısı olarak, 7 gizli katman nöronuna sahip, üç katmanlı ileri beslemeli yapay sinir ağı kullanılmıştır ve aktivasyon fonksiyonu olarak sigmoid fonksiyon tercih edilmiştir.

DIFACONN-miner algoritmasının gerçek hayat problemi üzerindeki performansı iki aşamada değerlendirilmiştir. İlk aşamada algoritma, yazarlar tarafından [9] 2009 yılında geliştirilen TACO-miner algoritması ile karşılaştırılmıştır. Tablo 9 ve Tablo 10 sırasıyla TACO-miner ve DIFACONN-miner algoritmalarının parametre değerlerini göstermektedir.

**Tablo 9.** TACO-Miner Algoritması Parametre Değerleri

Karınca sayısı (M)	İterasyon Sayısı(T)	Sıklık faktörü(f)	Buharlaşma parametresi (g)	Q
100	1000	2	0,8	10

**Tablo 10.** DIFACONN-Miner Algoritması Parametre Değerleri

NP	G	CR	F	$x^{(1)}$	$x^{(2)}$	M	T	f	$\rho$	Q
5	50	0.8	1.2	-1	1	100	250	2	0.8	10

Ele alınan üç hata tipine ait veri kümelerinde algoritmaları on kez çalıştırmanın, test veri kümeleri üzerindeki tahminleyici doğruluklarının ve kural sayılarının minimum, ortalama, maksimum değerleri ve standart sapmalar Tablo 11'de gösterilmektedir. Tablodan da görüldüğü gibi TACO-miner algoritması ile karşılaştırıldığında DIFACONN-miner algoritması daha az sayıda kuralla daha yüksek tahminleyici doğruluklar elde etmektedir.

DIFACONN-miner algoritmasının gerçek hayat veri kümeleri üzerindeki performansını daha iyi değerlendirmek için algoritmanın elde ettiği tahminleyici doğruluklar ve kural sayıları Weka 3.5 veri madenciliği yazılımındaki dört popüler algoritma ile karşılaştırılmıştır. Bu algoritmalar "C4.5" [15], "PART" [16], "DecisionTable" [17] ve "NBTree" [18] algoritmalarıdır. Tablo 12, test veri kümeleri için minimum, ortalama ve maksimum doğrulukları, standart sapmaları ve ortalama kural sayılarını göstermektedir. Algoritmaları on kez çalıştırılırken aynı eğitim ve test veri kümeleri kullanıldığından, karşılaştırılmada kullanılan algoritmalar on çalıştırmada da aynı sonuçları vermiştir. Dolayısıyla test doğruluklarının ve kural sayılarının standart sapmaları 0 olarak hesaplanmıştır.

**Tablo 11.** TACO-Miner ve DIFACONN-Miner Algoritmalarının Tahminleyici Doğruluk ve Kural Sayıları

Algoritma	Veri Kümesi	Duruş-kalkış	Düğüm	Kafes	
TACO-miner	Test doğruluğu	Minimum	61.54	76.62	77.08
		Ortalama	83.08	90.52	93.96
		Maksimum	97.44	94.81	95.83
	Kural sayısı	Standart sapma	11.92	5.36	5.929
		Minimum	25.00	32.00	20.00
		Ortalama	34.80	39.40	27.00
		Maksimum	42.00	43.00	39.00
		Standart sapma	5.59	<b>4.20</b>	<b>5.48</b>
DIFACONN-miner	Test doğruluğu	Minimum	<b>92.01</b>	<b>92.21</b>	<b>95.83</b>
		Ortalama	<b>92.92</b>	<b>97.79</b>	<b>98.95</b>
		Maksimum	93.59	<b>100.00</b>	<b>100.00</b>
	Kural sayısı	Standart sapma	<b>0.71</b>	<b>2.88</b>	<b>1.48</b>
		Minimum	<b>9.00</b>	<b>10.00</b>	<b>7.00</b>
		Ortalama	<b>15.30</b>	<b>24.60</b>	<b>12.20</b>
		Maksimum	<b>25.00</b>	<b>34.00</b>	<b>26.00</b>
		Standart sapma	<b>5.46</b>	8.48	7.30

**Tablo 12.** DIFACONN-Miner Algoritmasının Klasik Sınıflandırıcılar ile Karşılaştırılması

Veri kümesi		C4.5	DecisionTable	NBTree	PART	DIFACONN-miner	
Duruş-kalkış	Test doğruluğu	Minimum	<b>93.59</b>	78.21	87.18	85.90	92.01
		Ortalama	<b>93.59</b>	78.21	87.18	85.90	92.92
		Maksimum	<b>93.59</b>	78.21	87.18	85.90	<b>93.59</b>
	Kural sayısı	Standart sapma	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	0.71
		Minimum	26.00	39.00	11.00	22.00	<b>9.00</b>
		Ortalama	26.00	39.00	<b>11.00</b>	22.00	15.30
		Maksimum	26.00	39.00	<b>11.00</b>	22.00	25.00
		Standart sapma	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	5.46
Düğüm	Test doğruluğu	Minimum	60.26	61.54	84.62	61.54	<b>92.21</b>
		Ortalama	60.26	61.54	84.62	61.54	<b>97.79</b>
		Maksimum	60.26	61.54	84.62	61.54	<b>100.00</b>
	Kural sayısı	Standart sapma	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	2.88
		Minimum	<b>3.00</b>	<b>3.00</b>	5.00	4.00	10.00
		Ortalama	<b>3.00</b>	<b>3.00</b>	5.00	4.00	24.60
		Maksimum	<b>3.00</b>	<b>3.00</b>	5.00	4.00	34.00
		Standart sapma	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	8.48
Kafes	Test doğruluğu	Minimum	<b>95.92</b>	<b>95.92</b>	<b>95.92</b>	<b>95.92</b>	95.83
		Ortalama	95.92	95.92	95.92	95.92	<b>98.95</b>
		Maksimum	95.92	95.92	95.92	95.92	<b>100.00</b>
	Kural sayısı	Standart sapma	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	1.48
		Minimum	<b>1.00</b>	4.00	4.00	<b>1.00</b>	7.00
		Ortalama	<b>1.00</b>	4.00	4.00	<b>1.00</b>	12.20
		Maksimum	<b>1.00</b>	4.00	4.00	<b>1.00</b>	26.00
		Standart sapma	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	7.30

Tablo 12'den görüldüğü gibi DIFACONN-miner algoritması, DecisionTable, NBTree ve PART algoritmalarından daha iyi tahminleyici doğruluklar elde etmiştir. DIFACONN-miner algoritmasının ortalama doğrulukları, karşılaştırılan algoritmalarından oldukça yüksektir. Duruş-kalkış veri kümesinde C4.5 algoritmasının test doğruluğu az farkla DIFACONN-miner algoritmasından daha yüksektir, ancak C4.5 algoritmasının ortalama kural sayısı belirgin bir şekilde DIFACONN-miner algoritmasından yüksektir. DIFACONN-miner algoritması kafes ve düğüm veri kümelerinde daha yüksek tahminleyici doğruluklarla daha fazla kural üretmiştir. Tahminleyici doğruluğu düşürmeden, kural sayısı farklı uygunluk fonksiyonu kullanılarak düşürülebilir. Elde edilen sonuçlar, DIFACONN-miner algoritmasının bir tekstil işletmesinde en sık rastlanan kalite hatalarında etkin faktörleri belirlemede doğru ve anlaşılır sınıflandırma kuralları üretebildiğini göstermektedir.





## SONUÇ

Bu çalışmada bir sınıflandırma kural üretme algoritması olan DIFACONN-miner incelenmiştir. Algoritmanın performansı ilk olarak sınıflandırma referans veri kümeleri üzerinde değerlendirilmiş daha sonra ise bir tekstil işletmesindeki kalite hataları üzerine gerçek bir hayat uygulaması gerçekleştirilmiştir. Veri madenciliği, sürekli üretimi gerçekleştirmeyi amaçlayan araştırmacılar ve uygulamacıların yer aldığı bir takım olarak düşünülebilir. Her üretim işletmesinde devasa miktarda veri üretilmektedir. Bu devasa veriden kullanışlı bilginin çıkarımı, hata oranlarını düşürmek, enerji kullanımını iyileştirmek gibi çeşitli amaçlar için sürdürülebilirliği artırmakta yardımcı olacaktır. Bu çalışmada ele alınan DIFACONN-miner algoritmasının sınıflandırma veri kümeleri ve gerçek hayat veri kümesindeki elde ettiği sonuçlar, algoritmanın üretim gibi gerçek hayat problemlerini çözebileceğini; işletmelerde depolanan devasa boyutlu veri kümelerinden anlamlı ve kullanışlı bilgiler çıkarabileceğini göstermektedir.

## KAYNAKLAR

- [1] FU, X., WANG, L., "Rule Extraction By Genetic Algorithms Based on A Simplified RBF Neural Network", in: Proceedings of The 2001 IEEE Congress on Evolutionary Computation (CEC 2001) vol. 2, 753-758, 2001.
- [2] ANDREWS, R., DIEDERICH, J., TICKLE, A., B., "A Survey, Critique of Techniques for Extracting Rules From Trained Artificial Neural Networks", Knowledge Based Systems, vol. 8(6), 373-389, 1995.
- [3] ELALFI, E., HAQUE, R., ELALAMI, M. E., "Extracting Rules From Trained Neural Network Using GA for Managing E-business", Applied Soft Computing, vol. 4, 65-77, 2004.
- [4] MARKAWSKA-KACZMAR, U., WNUK-LIPINSKI, P., "Rule Extraction From Neural Network By Genetic Algorithm with Pareto Optimization", in: Proceedings of the 7th International Conference on Artificial Intelligence and Soft Computing –ICAISC 2004, Lecture Notes in Computer Science, vol. 3070, Springer, 450-455, 2004.
- [5] MARKOWSKA-KACZMAR U., "The Influence of Parameters in Evolutionary Based Rule Extraction Method From Neural Networks", in: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), 2005.
- [6] TOKINAGA, S., LU, J., IKEDA, Y., "Neural Network Rule Extraction By Using The Genetic Programming and Its Applications to Explonatory Classifications", IECE Transactions on Fundamentals, E88-A (10), 2627-2635, 2005.
- [7] HRUSCHKA, E. R., EBECKEN, N. F. F., "Extracting Rules From Multilayer Perceptrons in Classification Problems: A Clustering-based Approach", Neurocomputing, vol. 70, 384-397, 2006.
- [8] SETIONO, R., BAESENS, B., MUES, C., "A Note on Knowledge Discovery Using Neural Networks and Its Application to Credit Card Screening", European Journal of Operational Research, vol. 192(1), 326-332, 2008.
- [9] OZBAKIR, L., BAYKASOGLU, A., KULLUK, S., YAPICI, H., "TACO-miner: An Ant Colony Based Algorithm for Rule Extraction From Trained Neural Networks", Expert Systems with Applications, vol. 36(10), 12295-12305, 2009.
- [10] KAHRAMANLI, H., ALLAHVERDI, N., "Rule Extraction from Trained Adaptive Neural Networks Using Artificial Immune Systems", Expert Systems with Applications, vol. 36, 1513-1522, 2009.
- [11] OZBAKIR, L., BAYKASOGLU, A., KULLUK, S., "A Soft Computing-based Approach for Integrated Training and Rule Extraction From Artificial Neural Networks: DIFACONN-miner", Applied Soft Computing, vo. 10(1), 304-317, 2010.
- [12] ILLONEN, J., KAMARAINEN, J. K., LAMPINEN, J., "Differential Evolution Training Algorithm for Feed-forward Neural Networks", Neural Processing Letters, vol 17, 93-105, 2003.
- [13] PHAM, D.T., BIGOT, S., DIMOV, S. S., "Rule-5: A Rule Induction Algorithm for Classification Problems Involving Continuous Attributes", Proceedings of The Institution of Mechanical Engineers: Journal of Mechanical Engineering Science, vol. 217©, 1273-1286, 2003.
- [14] TABUCCANON, M. T., "Multiple Criteria Based Decision Making in Industry", Elsevier, New York, 1988.



- [15] QUINLAN, R., "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [16] FRANK, E., WITTEN, I. H., "Generating Accurate Rule Sets Without Global Optimization", in: J. Shavlik (Ed.), Machine Learning: Proceedings of the 15<sup>th</sup> International Conference, Morgan Kaufmann Publishers, 144-151, 1998.
- [17] KOHAVI, R., "The Power of Decision Tables", in: N. Lavrac, S. Wrobel (Eds.), Machine Learning: Proceedings of the 8<sup>th</sup> European Conference on Machine Learning (ECML95), Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 914, 174-189, 1995.
- [18] JOHN, G. H., LANGLEY, P., "Estimating Continuous Distributions in Bayesian Classifiers", in: Proceedings of the 11<sup>th</sup> Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, 338-345, 1995.
- [19] TAN, C., YU, Q., ANG, J. H., "A Dual-Objective Evolutionary Algorithm for Rules Extraction in Data Mining", Computational Optimization and Applications, vol. 34, 273-294, 2006a.
- [20] TAN, C., YU, Q., ANG, J. H., "A Coevolutionary Algorithm for Rules Discovery in Data Mining", International Journal of System Science, vol. 37(12), 835-864, 2006b.
- [21] THABTAH, F. A., COWLING, P. I., "A Greedy Classification Algorithm Based on Association Rule", Applied Soft Computing, vol. 7, 1102-1111, 2007.
- [22] CHEN, G., LIU, H., YU, L., WEI, Q., ZHANG, X., "A New Approach to Classification Based on Association Rule Mining", Decision Support Systems, vol. 42, 674-689, 2006.
- [23] SMALDON, J., FREITAS, A. A., "A New Version of The Ant-Miner Algorithm Discovering Unordered Rule Sets", In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006), 43-50, 2006.
- [24] BAYKASOĞLU, A., ÖZBAKIR, L., "MEPAR-Miner: Multi-Expression Programming for Classification Rule Mining", European Journal of Operational Research, vol. 183, 767-784, 2007.
- [25] DEHURI, S., MALL, R., "Predictive and Comprehensible Rule Discovery Using A Multi-Objective Genetic Algorithm", Knowledge-Based Systems, vol. 19, 413-421, 2006.
- [26] HOFFMANN, F., BAESSENS, B., MUES, C., GESTEL, T. V., VANTHIENEN, J., "Inferring Descriptive and Approximate Fuzzy Rules for Credit Scoring Using Evolutionary Algorithms", European Journal of Operational Research, vol. 177, 540-555, 2007.
- [27] LI, R., WANG, Z., "Mining Classification Rules Using Rough Sets and Neural Networks", European Journal of Operational Research, vol. 157, 439-448, 2004.
- [28] BAYKASOĞLU, A., ÖZBAKIR, L., KULLUK, S., "Classifying Defect Factors in Fabric Production via DIFACONN-miner : A Case Study", Expert Systems with Applications, vol. 38, 11321-11328, 2011.

## ÖZGEÇMİŞ

### Sinem KULLUK

1977 yılı Kayseri doğumludur. 2000 yılında Erciyes Üniversitesi Endüstri Mühendisliği Bölümünü bitirmiştir. Gazi Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalından 2003 yılında Yüksek Mühendis ve Erciyes Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliği Anabilim Dalından 2009 yılında Doktor ünvanını almıştır. 2001-2009 yılları arasında Araştırma Görevlisi olarak görev yapmıştır. 2009 yılından beri Erciyes Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği Bölümü'nde Yrd. Doç. Dr. olarak görev yapmaktadır. Veri madenciliği, yapay zeka teknikleri, tesis planlama, üretim planlama ve kontrol ilgi alanlarıdır.

### Lale ÖZBAKIR

1971 yılı Kayseri doğumlu Yrd. Doç. Dr. Lale Özbakır, Lisans öğrenimini 1992 yılında Bilkent Üniversitesi Bilgisayar ve Enformatik Mühendisliği Bölümünde tamamlamıştır. Erciyes Üniversitesi Sosyal Bilimler Enstitüsü Yönetim ve Organizasyon Anabilim Dalında 1997 yılında yüksek lisans derecesini, Üretim Yönetimi ve Pazarlama Anabilim Dalında 2004 yılında doktora derecesini almıştır. Erciyes Üniversitesi Endüstri Mühendisliği Bölümüne 1997 yılında araştırma görevlisi, 2004 yılında Yardımcı Doçent olarak atanmış olup halen aynı bölümde Öğretim Üyesi olarak görev yapmaktadır.



Yazarın uluslararası bilimsel dergilerde 30'un üzerinde, ulusal ve uluslararası kongrelerde 50'nin üzerinde bilimsel yayını bulunmaktadır. Yrd. Doç. Dr. Lale Özbakır çok sayıda ulusal ve uluslararası dergide hakemlik görevi yapmakta olup, çalışma alanları içerisinde veri madenciliği, yapay zeka ve meta-sezgisel yaklaşımlar, evrimsel algoritmalar, yöneylem araştırması yer almaktadır.

### **Adil BAYKASOĞLU**

Prof. Dr. Adil Baykasoğlu Isparta Teknik Lisesi Makina bölümünden mezun olduktan sonra Lisans ve Yüksek Lisans derecelerini Makina Mühendisliği alanında 1993 ve 1995 yıllarında Gaziantep'te, doktora derecesini ise YÖK bursu ile gittiği Nottingham Üniversitesinden 1999 yılında Endüstri Mühendisliği alanında almıştır. 1993-2010 yılları arasında Gaziantep Üniversitesi Endüstri Mühendisliği Bölümünde çalışan Prof. Baykasoğlu halen Dokuz Eylül Üniversitesi Endüstri Mühendisliği bölümünde çalışmaktadır. Prof. Baykasoğlu ulusal ve uluslar arası bilimsel dergi ve kongrelerde 300 civarında bilimsel makale yayımladı. Yazarın ayrıca üç adet yayımlanmış kitabı, düzenleyip editörlüğünü yaptığı çeşitli ulusal ve uluslar arası kongre kitapları bulunmaktadır. Yazarın çalışma alanları genelde yöneylem araştırması, bilişimsel yapay zekâ, zeki etmenler, lojistik ve üretim sistemleri yönetimi/tasarımı, bilgisayar destekli üretim, kalite ve benzetim konuları üzerinde yoğunlaşmaktadır. Prof. Baykasoğlu çok sayıda uluslararası dergide hakem ve yayın kurulu üyesi olarak görev yapmakta olup aynı zamanda Turkish Journal of Fuzzy Systems dergisinin eş-editörlüğünü yürütmektedir. Prof. Baykasoğlu'na 2007 yılında Türkiye Bilimler Akademisi Üstün Başarılı Genç Bilim İnsanı ödülü, 2008 yılında ODTÜ M. Parlar araştırma teşvik ödülü, 2010 yılında ise Tübitak Teşvik ödülü verilmiştir.