Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 30, S. 12-16, Aralık 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 30, pp. 12-16, December 2021

Research Article

Salp Sürü Algoritması ile Öznitelik Seçimi ve Sınıflandırıcı Performans Değerlendirmesi

Celal Can[1*], Yasin Kaya[2], Fatih Kılıç[3]

[1*]Adana Alparslan Türkeş Bilim ve Teknoloji Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Adana, Türkiye, (ORCID: 0000-0002-7631-8934), ccan@atu.edu.tr

[2] Adana Alparslan Türkeş Bilim ve Teknoloji Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Adana, Türkiye (ORCID: 0000-0002-9074-0189), ykaya@atu.edu.tr

[3] Adana Alparslan Türkeş Bilim ve Teknoloji Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Adana, Türkiye (ORCID: 0000-0002-8550-1562), fkilic@atu.edu.tr

(2nd International Conference on Computer, Electrical and Electronic Sciences ICCEES 2021, September 1-3, 2021)

(**DOI:** 10.31590/ejosat.1005417)

ATIF/REFERENCE: Can, C., Kaya, Y., & Kılıç, F. (2021). Salp Sürü Algoritması ile Öznitelik Seçimi ve Sınıflandırıcı Performans Değerlendirmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (30), 12-16.

Öz

Son yıllarda doğadan esinlenen sürü tabanlı algoritmalar arasında yer alan Salp Sürü Algoritması oldukça popüler olmuştur. Bu çalışmada, Salp Sürü Algoritması kullanılarak farklı veri setleri üzerinde öznitelik seçimi yapılmış, farklı sınıflandırıcılar ile bazı performans metrikleri karşılaştırılmıştır. Deneysel sonuçların hesaplanması için UCI Makine Öğrenmesi Deposunda yer alan BreastCancer, Colon ve Ionosphere veri setleri kullanılmıştır. Sınıflandırıcı olarak k En Yakın Komşu Algoritması, Destek Vektör Makineleri ve Rastgele Orman Algoritması kullanılmıştır. Sayısal sonuçlar incelendiğinde, çalışma zamanı bakımından kNN algoritması ile yapılan testler genellikle en hızlı algoritma olmuştur. Seçilen öznitelik sayısı bakımından ise SVM ve RF algoritmaları daha iyi sonuç vermiştir.

Anahtar Kelimeler: Öznitelik seçimi, Salp sürü algoritması, Doğadan esinlenen algoritmalar, Sınıflandırıcı algoritmaları.

Feature Selection Using Salp Swarm Algorithm and Classifier Performance Evaluation

Abstract

Salp Swarm Algorithm, a nature-inspired swarm-based algorithm, has become very popular in recent years. This study uses Salp Swarm Algorithm for feature selection and tries different classifiers as fitness functions on various datasets. BreastCancer, Colon, and Ionosphere databases in the UCI Machine Learning Repository are used as test datasets. k Nearest Neighbor Algorithm (kNN), Support Vector Machines (SVM), and Random Forest Algorithm (RF) are used as classifiers. When the experimental results are examined, the kNN algorithm is generally the fastest in terms of runtime. However, considering the number of selected features, SVM and RF algorithms achieve better results.

Keywords: Feature selection, Salp swarm algorithm, Nature-inspired algorithms, Classification algorithms

[*] Sorumlu Yazar: ccan@atu.edu.tr

1. Giriş

Veri madenciliği çalışmalarında karşılaşılan sorunlardan biri de veri setlerinde boyut sayısının çok fazla olmasıdır. Öznitelik seçimi, öğrenme algoritmasının performansını artırmak için bir veri kümesinden sınıflandırmada etkisi olmayan ve farklı özelliklerin kaldırılması işlemidir. Problemlerin çözümünde öznitelik seçimi için bazı metasezgisel algoritmaların ikili versiyonları geliştirilmiştir. Literatürde hayvanların doğada besin bulma mekanizmalarını temel alan sürü tabanlı birçok algoritma bulunmaktadır. Arı Kolonisi Algoritması (Karaboga ve Akay, 2007), Parçacık Sürü Algoritması (Eberhart ve Kennedy, y.y.; Kılıç, Kaya ve Yildirim, 2021), Balina optimizasyon algoritması (Mirjalili ve Lewis, 2016), Gri Kurt Optimizasyonu (Aswani, Ghrera ve Chandra, 2016; Kumar, Chhabra ve Kumar, 2017; Mirjalili, Mirjalili ve Lewis, 2014), Karınca Kolonisi Algoritması (Dorigo, Birattari ve Stutzle, 2006), Ateşböceği algoritması (X. S. Yang, 2010), Yarasa Algoritması (X.-S. Yang, 2010) ve Salp Sürü Algoritması (SSA) (Mirjalili ve diğerleri, 2017) doğadan esinlenen algoritmalardan bazılarıdır. Salp Sürü Algoritması basit, verimli ve esnektir, uygulanması kolaydır ve diğer algoritmalardan daha az sayıda parametreye sahiptir. Literatüre bakıldığında Salp sürü algoritması ile yapılan çalışmalar ile karşılaşmak mümkündür.

Örneğin, Sayed ve ark. çalışmalarında kaotik haritalar kullanarak SSA algoritmasında C2 parametresini değiştirmişlerdir. C2 parametresi için 10 kaotik harita türünü karşılaştırmışlar ve lojistik kaotik haritanın en iyisi olduğu sonucuna varmışlardır. Bu iyileştirme ile yakınsama hızını ve yerel optimum problemini çözmek için kullanmışlardır (Sayed, Khoriba ve Haggag, 2018). İbrahim ve ark. biyomedikal veri kümelerinde öznitelik seçimi için herhangi bir iyileştirme yapılmadan SSA uygulamışlardır. SSA'nın hem gerçek hem de sentetik veri kümeleri için seçilen diğer algoritmalara kıyasla daha az çalışma süresi ile en yüksek doğruluğu elde ettiğini göstermişlerdir (Ibrahim, H. T., Mazher, W. J., Ucan, O. N., and Bayat, 2017). Hegazy ve ark. yakınsama hızını iyileştirmek ve SSA sömürüsü ile keşif arasında bir denge kurmak için SSA pozisyon güncelleme denklemlerine yeni bir ağırlık parametresi eklemişlerdir (Hegazy, Makhlouf ve El-Tawel, 2020). Faris ve ark., SSA'nın ana döngü yinelemesi sırasında lider ve takipçi sayısı için zamanla değişen bir teknik önermişlerdir. Bu iyileştirme, yerel optimum ve erken yakınsama içeren SSA sorunlarını çözmek için dahil edilmiştir (Faris ve diğerleri, 2020).

Bu çalışmanın organizasyonu şu şekilde yapılmıştır: Bölüm 2, Salp Sürü Algoritmasının ayrıntılarını içeren Materyal ve Yöntemi sunmaktadır. Bölüm 3, yapılan deneysel sonuçların ayrıntılarını ve bu deneylerden elde edilen sonuçları sunmaktadır. Bölüm 4 ise çalışmanın ana sonuçların özetini sunmaktadır.

2. Materyal ve Metot

Bu bölümde çalışmada kullanılan Salp sürü algoritmasının ne olduğu, matematiksel modeli ve sözde kodu ile uygunluk fonksiyonu hakkında bilgilere yer verilmiştir.

Deneyler, Windows 10, Intel i7-4700HQ CPU 2.40 GHz, 12 GB belleğe sahip bir bilgisayarda MATLAB R2019b'de yürütülmüştür.

2.1. Salp Sürü Algoritması

Salp sürüsü algoritması, Mirjalili ve ark. tarafından geliştirilmiş rastgele popülasyon tabanlı algoritmalardan bir tanesidir [11]. Salp'ler Salpidae ailesine ait bir deniz canlısıdır ve vücutları jöle kıvamında olan balıklar gibidir. Bununla birlikte hareketleri jöle balığına çok benzer. Genellikle Salp'ler okyanusun derinliklerinde salp zinciri oluşturarak bir sürü şeklinde davranış gösterirler. Salp sürü algoritmasında iki tür salp vardır. Birincisi lider olarak tanımlanan salp, ikincisi lideri takip eden Salp'lerdir. Salp'lerin salp zinciri adı verilen belirli davranışları vardır. Bu davranış, yiyecek arama için kullanılır. Lider Salp'in pozisyon değiştirme denklemi şu şekildedir [11]:

$$x_j^i = \begin{cases} F_j + c1\big((ubj - lbj)c2 + lbj\big)\ c3 > 0.5 \\ F_j - c1\big((ubj - lbj)c2 + lbj\big)\ c3 < 0.5 \end{cases} \quad (1)$$

Burada;

x_j^i: j'inci boyuttaki lider salp'in pozisyonunu,

F_j: j'inci boyuttaki yiyecek kaynağının pozisyonunu,

$c1, c2, c3$: [0,1] aralığında eşit olarak üretilen rastgele değişkenleri,

ubj, lbj: Sırasıyla j'inci boyuttaki üst ve alt sınırı ifade etmektedirler.

Bu denklemde $c1$ parametresi şu şekilde hesaplanmaktadır:

$$c1 = 2e^{(\frac{4l}{L})^2} \quad (2)$$

Burada l şimdiki iterasyonu ve L ise maksimum iterasyon sayısını belirtmektedir.

Lider salp'i takip eden salp'lerin pozisyonunu güncellemek için ise şu denklem kullanılmaktadır:

$$x_j^i = \frac{1}{2}at^2 + v_0 t \quad (3)$$

Burada v_0 başlangıç hızını, t zamanı ifade etmektedir. Bu denklemde $i \geq 2$ olmalıdır. $a = \frac{v_{final}}{v_0}$ ve $v = \frac{x - x_0}{t}$ ile hesaplanır. $v_0 = 0$ alınır ve denklem şu şekilde kullanılır.

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \quad (4)$$

Algoritma 1: SSA Algoritmasının sözde kodu

```
Rastgele x (i = 1, 2, . . ., n) salp popülasyonunu oluştur
    for 1:L do
     Popülasyondaki her bir salp'i değerlendir
         En iyi Salp'i F olarak belirle
         c1 değerini güncelle Eşt. (2)
    for (her salp (xi) için) do
     if xi lider ise (i=1) then
       Liderin pozisyonunu güncelle Eşt. (1)
     else
          Takipçilerin pozisyonunu güncelle Eşt. (4)
return F
```

Yiyecek kaynağı ister hareketli ister sabit olsun, lider salp yiyecek etrafında konumlanır ve diğer takipçi Salp'ler lider salp arkasında zincir olacak şekilde sıralanır. Salp Sürü algoritmasına ait sözde kod Algoritma 1'de verilmiştir.

2.2. Uygunluk Fonksiyonu

Sınıflandırma tahmininde, genellikle uygunluk değeri olarak sınıflandırma hatası kabul edilmektedir. Sadece sınıflandırma hatasının kullanılması öznitelik seçimi için yeterli olmayabilir. Sınıflandırma hatasına ek olarak, uygunluk fonksiyonunda seçilen öznitelik sayısı da kullanılır. Böylece sınıflandırma hatasına sahip iki çözümden daha az özniteliği olan çözüm seçilmiş olur. Yapılan bu çalışmada, aşağıdaki denklem uygunluk fonksiyonu olarak kullanılmıştır.

$$Fitness = \rho Err(D) + \varphi \frac{|n|}{|N|} \quad (5)$$

Burada,
$Err(D)$ sınıflandırma hata oranını, ρ ve φ sabit değerler, n tanımlanan öznitelik alt kümesinin boyutunu, N ise toplam öznitelik sayısını vermektedir. Bu çalışmada, $\rho = 0.99$ ve $\varphi = 0.01$ alınmıştır.

3. Araştırma Sonuçları ve Tartışma

Bu bölümde çalışmada kullanılan veri setleri ile performans metrikleri hakkında bilgiler verilmiştir. Ayrıca elde edilen nümerik sonuçlar tablolar halinde sunulmuştur.

3.1. Veri Seti

Bu çalışmada, University of California Irvine üniversitesinin makine öğrenmesi çalışmalarında kullanıma sunduğu, web üzerindeki en bilindik ve en çok kullanılan veri depolarından biri olan UCI (UCI Machine Learning Repository) veri deposu kullanılmıştır. Veri setleri seçilirken farklı boyutlarda ve farklı özniteliğe sahip veri setleri seçilmesine dikkat edilmiştir. UCI veri setlerinden 3 tanesi kullanılmıştır. "Breastcancer" veri setinde 9 öznitelik ve 699 örnek bulunmaktadır. "Ionosphere" veri setinde 34 öznitelik ve 351 örnek bulunmaktadır. "Colon" veri seti ise 2000 öznitelik ve 62 örneğe sahiptir.

Uygunluk fonksiyonun hesaplanması için üç sınıflandırıcı kullanılmıştır. kNN sınıflandırıcısı için k=5 olarak ayarlanmıştır ve 10-kat çapraz doğrulama tekniği eğitim ve test veri seti üretmek için veri setlerine uygulanmıştır. SVM sınıflandırıcı çekirdek fonksiyonu olarak RBF kullanılmıştır. Rastgele orman sınıflandırıcısı için ağaç sayısı 100 alınmıştır.

3.2. Performans Metrikleri

Farklı sınıflandırıcılar ile SSA algoritması aşağıdaki metrikler kullanılarak karşılaştırılmıştır;

- Sınıflandırma doğruluğu: test veri setinde seçilen öznitelikler kullanılarak 10 çalışmadan ortalama doğruluk hesaplanarak elde edilir.
- F ölçüsü: Bir testin doğruluğunun ölçüsüdür.
- AUC: işlem karakteristik eğrisinin (ROC) altında kalan alanı gösterir.
- Uygunluk değerleri: belirtildiği gibi her bir yaklaşımdan elde edilirler. Ortalama ve minimum uygunluk değerleri karşılaştırılır.
- Ortalama seçim boyutu: seçilen özniteliklerin ortalama sayısıdır.
- Ortalama yürütme süresi: Saniye cinsinden çalıştırma süresidir.

3.3. Sayısal Sonuçlar

Tüm sınıflandırıcılar tarafından 10 çalıştırma ve 100 iterasyon sonunda elde edilen BreastCancer veri setine ait performans sonuçları Tablo 1'de, Colon veri setine ait performans sonuçları Tablo 2'de, Ionosphere veri setine ait performans sonuçları Tablo 3'de verilmiştir. Elde edilen en iyi değerler tablolarda eğik ve altı çizili olarak vurgulanmıştır.

Tablo 1. BreastCancer veri seti performans sonuçları

Sınıflandırıcı	En iyi Uygunluk	Ortalama Uygunluk	Ortalama Doğruluk	F-Ölçütü	AUC	Çalıştırma Zamanı	Seçilen öznitelik sayısı
KNN	0.6287	0.7171	0.7815	0.6983	0.8382	*112.6270*	9.1
SVM	*0.6065*	*0.6660*	0.7301	0.5773	0.7623	453.3615	*8.6*
RF	0.7448	0.7803	*0.8461*	*0.7812*	*0.9178*	6953.7065	11.5

Tablo 2. Colon veri seti performans sonuçları

Sınıflandırıcı	En iyi Uygunluk	Ortalama Uygunluk	Ortalama Doğruluk	F-Ölçüsü	AUC	Çalıştırma Zamanı	Seçilen öznitelik sayısı
KNN	0.6032	0.6195	0.7364	0.6634	0.8080	*109.9235*	1012
SVM	0.6903	0.7010	*0.8223*	*0.7721*	*0.8641*	181.9140	*984.5*
RF	*0.4814*	*0.5423*	0.6401	0.6262	0.7044	6357.3490	1012.25

Tablo 3. Ionosphere veri seti performans sonuçları

Sınıflandırıcı	En iyi Uygunluk	Ortalama Uygunluk	Ortalama Doğruluk	F-Ölçüsü	AUC	Çalıştırma Zamanı	Seçilen öznitelik sayısı
KNN	0.7230	0.7468	0.8087	0.8683	0.9000	*107.1298*	12.6
SVM	*0.6270*	*0.7037*	0.7662	0.8360	0.7251	321.4078	12.4
RF	0.7960	0.8063	*0.8794*	*0.9084*	*0.9492*	6657.4206	*10.75*

Tablo 1 incelendiğinde, en iyi ve ortalama uygunluk değerini sırasıyla 0.6065 ve 0.6600 değerleri ile SVM sınıflandırıcısı ile elde edilmiştir. Ortalama doğruluk, F ölçütü ve AUC metriklerinde sırasıyla 0.8461, 0.7812 ve 0.9178 değerleri ile Rastgele orman sınıflandırıcı ile elde edilmiştir. Çalıştırma zamanı bakımından en hızlı sınıflandırıcı 112.627 saniye ile kNN sınıflandırıcı olmuştur. Seçilen öznitelik sayısına bakıldığında ise 8.6 ile SVM sınıflandırıcı ile elde edildiği görülmektedir.

Colon veri setine ait performans sonuçlarına bakıldığında (Tablo 2) en iyi ve ortalama uygunluk değerini sırasıyla 0.4814 ve 0.5423 değerleri ile RF sınıflandırıcısı ile elde edilmiştir. Ortalama doğruluk, F ölçütü ve AUC metriklerinde sırasıyla 0.8223, 0.7721 ve 0.8641 değerleri ile SVM sınıflandırıcı ile elde edilmiştir. Çalıştırma zamanı bakımından en hızlı sınıflandırıcı 109.9235 saniye ile kNN sınıflandırıcı olmuştur. Seçilen öznitelik sayısına bakıldığında ise 984.5 ile SVM sınıflandırıcı ile elde edildiği görülmektedir.

Tablo 3'te ise Ionosphere veri setine ait performans sonuçları verilmektedir. Burada da en iyi ve ortalama uygunluk değerini sırasıyla 0.627 ve 0.7037 değerleri ile SVM sınıflandırıcısı ile elde edilmiştir. Ortalama doğruluk, F ölçütü ve AUC metriklerinde bakıldığında ise sırasıyla 0.8794, 0.9084 ve 0.9492 değerleri ile RF sınıflandırıcı ile elde edilmiştir. Diğer veri setlerinde olduğu gibi çalıştırma zamanı bakımından en hızlı sınıflandırıcı 107.1298 saniye ile kNN sınıflandırıcı olmuştur. Seçilen öznitelik sayısına bakıldığında ise 10.75 ile RF sınıflandırıcı ile elde edildiği görülmektedir.

4. Sonuç

Salp sürü algoritması basit ve uygulanması kolay bir algoritma olmasından dolayı geniş bir uygulama alanına sahiptir. Literatürde yapılan çalışmalarda genel olarak testler tek bir sınıflandırıcı ile yapılmıştır. Burada ise farklı veri setleri üzerinde farklı sınıflandırma algoritmaları kullanılarak elde edilen sonuçlar karşılaştırılmalı olarak irdelenmiştir. Çalışmada üç tane UCI veri seti kullanılmıştır. Sınıflandırıcı olarak ise kNN algoritması, SVM ve RF algoritması kullanılmıştır. Farklı sınıflandırıcı ve farklı veri setleri üzerinden Salp Sürü Algoritması performansları gerçekleştirilmiş ve sonuçlar değerlendirilmiştir.

Elde edilen bulgular değerlendirildiğinde her üç veri seti performans metriklerinde farklı sonuçlar ile karşılaşılmıştır. Çalıştırma zamanı bakımından kNN algoritması ile yapılan testler genellikle en hızlı algoritma olmuştur. Seçilen öznitelik sayısı bakımından ise SVM ve RF algoritmaları daha iyi sonuç vermiştir. Diğer metrikler bakımından değerlendirme yapıldığında ise veri setinin büyüklüğü, öznitelik sayısı, örnek sayısı gibi parametrelere göre farklı sınıflandırıcılar daha başarılı olmuştur.

Bu çalışma farklı veri setleri üzerinde yapılan öznitelik seçimi çalışmaları için kullanılan sınıflandırıcı seçiminde fikir edinme anlamında örnek bir çalışmadır. Bu çalışma ile başka veri setleri üzerinde de sınıflandırıcılar uygulanabilir ve öznitelik seçimi için en uygun sınıflandırıcılar seçilebilir.

Kaynakça

Aswani, R., Ghrera, S. P. ve Chandra, S. (2016). A Novel Approach to Outlier Detection using Modified Grey Wolf Optimization and k-Nearest Neighbors Algorithm. *Indian Journal of Science and Technology*, *9*(44). doi:10.17485/ijst/2016/v9i44/105161

Dorigo, M., Birattari, M. ve Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*(4), 28–39. doi:10.1109/MCI.2006.329691

Eberhart, R. ve Kennedy, J. (y.y.). A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science* içinde (ss. 39–43). IEEE. doi:10.1109/MHS.1995.494215

Faris, H., Heidari, A. A., Al-Zoubi, A. M., Mafarja, M., Aljarah, I., Eshtay, M. ve Mirjalili, S. (2020). Time-varying hierarchical chains of salps with random weight networks for feature selection. *Expert Systems with Applications*, *140*, 112898. doi:10.1016/j.eswa.2019.112898

Hegazy, A. E., Makhlouf, M. A. ve El-Tawel, G. S. (2020). Improved salp swarm algorithm for feature selection. *Journal of King Saud University - Computer and Information Sciences*, *32*(3), 335–344. doi:10.1016/j.jksuci.2018.06.003

Ibrahim, H. T., Mazher, W. J., Ucan, O. N., and Bayat, O. (2017). Feature Selection using Salp Swarm Algorithm for Real Biomedical Datasets. *International Journal of Computer Science and Network Security*, *17*(12), 13–20.

Karaboga, D. ve Akay, B. (2007). Artificial Bee Colony (ABC) Algorithm on Training Artificial Neural Networks. *2007 IEEE 15th Signal Processing and Communications Applications* içinde (ss. 1–4). IEEE. doi:10.1109/SIU.2007.4298679

Kılıç, F., Kaya, Y. ve Yildirim, S. (2021). A novel multi population based particle swarm optimization for feature selection. *Knowledge-Based Systems*, *219*, 106894. doi:10.1016/j.knosys.2021.106894

Kumar, V., Chhabra, J. K. ve Kumar, D. (2017). Grey Wolf Algorithm-Based Clustering Technique. *Journal of Intelligent Systems*, *26*(1), 153–168. doi:10.1515/jisys-2014-0137

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H. ve Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, *114*, 163–191. doi:10.1016/j.advengsoft.2017.07.002

Mirjalili, S. ve Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, *95*, 51–67. doi:10.1016/j.advengsoft.2016.01.008

Mirjalili, S., Mirjalili, S. M. ve Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, *69*, 46–61. doi:10.1016/j.advengsoft.2013.12.007

Sayed, G. I., Khoriba, G. ve Haggag, M. H. (2018). A novel chaotic salp swarm algorithm for global optimization and feature selection. *Applied Intelligence*, *48*(10), 3462–3481. doi:10.1007/s10489-018-1158-6

Yang, X.-S. (2010). A New Metaheuristic Bat-Inspired Algorithm (ss. 65–74). doi:10.1007/978-3-642-12538-6_6

Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, *2*(2), 78. doi:10.1504/IJBIC.2010.032124