Uludağ Üniversitesi Mühendislik Fakültesi Dergisi, Cilt 25, Sayı 2, 2020 ARAŞTIRMA

DOI: 10.17482/uumfd.695128

YAPAY SİNİR AĞLARI İLE KLASİK KRİPTOGRAFİ ALGORİTMALARININ ŞİFRELİ VERİLER ÜZERİNDEN SINIFLANDIRILMASI

Sevtap TÜRK [*]
Rüya ŞAMLI [**]

Alınma: 26.02.2020 ; düzeltme: 01.04.2020 ; kabul: 07.05.2020

Öz: Şifreleme bilimi olarak ifade edilebilecek olan kriptoloji için kullanılan algoritmaların temel amacı bir noktadan bir noktaya iletilen, ya da herhangi bir ortamda saklanan verilere izinsiz kişilerin erişmesini engellemek ve bu veriler ele geçirilse dahi verilerin anlaşılmasını imkânsız hale getirmektir. Günümüzde, birçok farklı tipteki şifreleme algoritmalarının temeli klasik simetrik şifreleme yöntemlerine dayanmaktadır. Gelişen teknolojiyle ortaya çıkan veri güvenliği sorununu çözmek için daha karmaşık matematiksel altyapıya sahip yöntemler denense de donanımsal gerçekleme zorlukları araştırmacıları farklı arayışlara yöneltmiştir. Bunlardan biri de YSA (Yapay Sinir Ağları – Artificial Neural Networks)'dır. Kriptoloji ve YSA'nın birleşimi ile oluşan ve "Nöral Kriptografi" olarak adlandırılan çalışma alanında hem şifreleme hem de kriptanaliz aşamalarında YSA modellerinden faydalanılmaktadır. Bu çalışmada, bir Nöral Kriptografi uygulaması ile klasik simetrik şifreleme yöntemlerinden birkaçıyla şifrelenen verilerin, YSA yöntemi ile klasik şifreleme algoritmalarından hangisine ait olduğu tahmin edilmeye çalışılmıştır.

Anahtar Kelimeler: Klasik Şifreleme, Simetrik Şifreleme Algoritmaları, Yapay Sinir Ağları

Classification of Classical Cryptography Algorithms Through Encrypted Data With Using Artificial Neural Networks

Abstract: The main aim of cryptography algorithms is to prevent unauthorized people from attaining data that transmitted from one node to another or stored in any environment, even if it is captured, making it impossible to decrypt. Today, basis of many different types of encryption methods is based on classical encryption algorithms. Although many methods which have more complex mathematical infrastructure are tried to solve the data security problem become important by the advancement of technology. The hardware implementation difficulties of these complex methods have led the researchers to the different areas. One of these areas is ANN (Artificial Neural Networks). In the study area called "Neural Cryptography" which is formed by the combination of cryptology and ANN, ANN models are used in both encryption and cryptanalysis phase. In this study, we prepared a Neural Cryptography application and have tried to determine which data is encrypted by which classical method with using ANN.

Keywords: Classical Encryption, Symmetric Cipher Algorithms, Artificial Neural Networks

[*] Bandırma Onyedi Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Yazılım Mühendisliği Bölümü, 10200, Bandırma/Balıkesir/Türkiye

[**] İstanbul Üniversitesi-Cerrahpaşa, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 34320, Avcılar/İstanbul/Türkiye

İletişim Yazarı: Sevtap Türk (sturk@bandirma.edu.tr)

1. GİRİŞ

Kriptoloji; kriptografi (şifre yazımı, şifre oluşturma) ve kriptanaliz (şifre kırma) alt dallarını bünyesinde toplayan şifreleme bilim dalıdır (Piper, 1997). Kriptografide, şifreleme sistemleri tasarlanırken kriptanalizde ise anahtar değeri bilinmeden şifreli metinden anlamlı bilgiler çıkarma işlemleri gerçekleştirilir. Kriptanalizde gerçekleştirilen yöntem ve anahtar belirleme çalışmalarındaki temel amaç, kullanılan algoritmanın zayıf yönünü ortaya çıkarmaya çalışmaktır (Sharif ve diğ., 2010). Bu işlem gerçekleştirilirken, Kaba Kuvvet Saldırısı (Brute Force Attack), Sözlük Saldırısı (Dictionary Attack), Gökkuşağı Tablosu Saldırısı (Rainbow Table Attack) gibi algoritmik yöntemlere ya da matematiksel yöntemlere başvurulmaktadır (Tan ve Ji, 2016). Bunun yanında literatürde makine öğrenmesi yöntemlerini kullanarak yapılan çalışmalar da mevcuttur.

Khadivi ve Momtazpour (2010), Gelişmiş Şifreleme Standardı (Advanced Encryption Standard - AES) gibi blok şifreleme yöntemlerinde Destek Vektör Makinesi (Support Vector Machine - SVM), En Yakın K Komşu (K Nearest Neighbor) ve Doğrusal Diskriminant Analizi (Linear Discriminant Analysis - LDA) sınıflandırıcılarını kullanarak şifreli metinleri sınıflandırmıştır. Tan ve Ji (2016) yine blok şifreleme yöntemlerinden Gelişmiş Şifreleme Standardı, Balon Balığı Şifrelemesi (Blowfish Cipher), Üçlü Veri Şifreleme Standardı (Triple Data Encryption Standards - 3DES), RC5 Şifreleme Algoritması ve Veri Şifreleme Standardı için Destek Vektör Makinesi ile sınıflandırma yapmaya çalışmış ve şifreli metni içeren dosyaların boyutu büyüdüğünde sınıflandırma başarı oranının da arttığını tespit etmiştir.

Dileep ve Sekhar (2006) çalışmasında, Elektronik Kod Defteri (Electronic Code Book) ile Veri Şifreleme Standardı, Şifre Blok Zincirlemesi (Cipher Block Chaining) ile Veri Şifreleme Standardı, Üçlü Veri Şifreleme Standardı, Balon Balığı Şifrelemesi, Gelişmiş Şifreleme Standardı ve RC5 yöntemleriyle şifrelenen veriler üzerinde Destek Vektör Makinesi tabanlı En Yakın K Komşu algoritmasıyla şifreli veriyi sınıflandırmaya çalışmıştır. Türk (2017), klasik şifreleme algoritmalarından Ötelemeli Şifreleme (Shift Cipher), Yerdeğiştirme Şifrelemesi (Substitution Cipher), Doğrusal Şifreleme (Affine Cipher), Hill Şifrelemesi (Hill Cipher), Vigenere Şifrelemesi (Vigenere Cipher) şifreleme yöntemleriyle şifrelenen metinleri, YSA ile sınıflandırmaya çalışmıştır. Ötelemeli Şifreleme için düz ve şifreli metinler girdi olarak alınmış, çıktı olarak belirlenen anahtar değerini YSA'nın tahmin etmesi sağlanmıştır. Doğrusal Şifreleme, Vigenere Şifrelemesi ve Hill Şifrelemesi yöntemleri için anahtar değeri sabit tutulup şifreli metinlere karşılık düz metin tahmini, Yerdeğiştirme Şifrelemesi içinse sabit bir permütasyon dizilimi seçilip yine şifreli metne karşılık YSA'nın düz metni tahmin etmesi sağlanmıştır.

Sharif ve diğ. (2010), çalışmasında blok şifreleme algoritmalarından Vigenere Şifrelemesi, Uluslararası Veri Şifreleme Algoritması (International Data Encryption Algorithm), Gelişmiş Şifreleme Standardı ve RC2 Şifreleme Algoritması'nın farklı bitlerdeki versiyonlarını Elektronik Kod Defteri şekliyle, herbiri farklı anahtar değeriyle şifrelenen veri dosyalarını, örüntü tanıma yöntemlerinden; Naive Bayes Sınıflandırıcısı, Destek Vektör Makinesi, YSA, Örnek Tabanlı Öğrenen (Instance Based Learner), Torbalama (Bagging – Bootstrap Aggregating), Uyarlamalı Yükseltme (Adaptive Boosting), Rotasyon Orman Sınıflandırıcı (Rotation Forest Classifier), C4.5 Ağacı (C4.5 Tree) ile sınıflandırmaya çalışmıştır.

Abd ve Al-Janabi (2019), klasik şifreleme yöntemlerini yerine koyma, yer değiştirme ve bunların birleşimi olarak alt başlıklara ayırmış ve tanımladığı her sınıf için harflerin frekans dağılım oranları gibi özelliklerini kullandığı sınıflandırıcı etiketler hazırlamıştır. Bu etiketleri farklı aşamalarda kullanmış ve şifreli metinleri YSA ile sınıflandırmada başarılı sonuçlar elde ettiği görülmüştür. Dunham ve diğ. (2005) çalışmasında, "Cipher in Depth" olarak tanımlanan, aynı anahtarı kullanarak oluşturulan dizi (stream) şifrelemede, üç seviyeli YSA yapısı oluşturarak farklı dosya tiplerindeki verinin "in depth" mi yoksa "not in depth" mi olduğunu tahmin etmeye çalışmıştır.

Chandra ve diğ. (2007), seçtikleri blok ve dizi şifreleme yöntemleriyle şifrelenmiş verilerden oluşan dosyaları YSA ile ayırt etmeye çalışmışlardır. Blok şifrelemede; geliştirilmiş RC6 ve

Serpent algoritmalarını, dizi şifrelemede ise Lili-128 ve Rabbit algoritmalarını kullanmışlardır. Blok ve dizi şifrelemeyi önce kendi içlerinde sonrasında karışık ikili gruplarla YSA ile yöntem tespiti yapmaya çalışmışlardır. Farklı YSA modelleri kullanarak başarı oranının değişimini gözlemlemişlerdir.

Bazı çalışmalarda YSA'nın sadece kriptografik yöntemin tespiti ya da sınıflandırılmasında değil, verinin şifrelenmesi veya şifrenin çözülmesi işlemlerinde direkt olarak kullanıldığı görülmektedir. Yee ve Silva (2002) çalışmasında açık anahtarlı bir kriptografik sistem oluşturmada YSA'yı kullanmıştır. YSA modellerinden biri olan ÇKA (Çok Katmanlı Algılayıcı - Multilayer Perceptron) yapısını kullanmayı tercih etmiş ve eğitim frekansı ile kullanılan örneklere göre ağırlık değerlerinin değişmesini açık anahtarlı kriptografi sistemini oluşturmada önemli bir özellik olarak görmüşlerdir. Önerdikleri sistem; A ve B adında iki ayrı bölümden ve bu bölümlerin ÇKA modellerinden oluşmaktadır. Aynı ağırlık değerleriyle başlatılan ÇKA'lar için A ve B bölümleri kendi gizli anahtarlarını seçmektedirler. A bölümü 128 ASCII (American Standard Code for Information Interchange) karakterini seçmekte ve ÇKA ağı seçilen bu 128 girdi ile eğitilmektedir. Sonrasında A ve B bölümü kendi ÇKA'larını birbirleriyle değiştirmektedir. B bölümü A'dan aldığı ÇKA için yeni bir gizli anahtar seçerek aynı işlemleri tekrarlamaktadır. A ve B bölümlerinin ağırlık değerleri eşitlendiğinde, her iki bölümde bu ağırlık değerlerine sahip ÇKA'larla şifreleme ve şifre çözme işlemlerini gerçekleştirmektedirler. Bazı çalışmalarda gizli anahtar oluşturma ve şifreleme kısımlarının Boole Cebri ile yapıldığını ardından oluşturulan şifreli örneklerle YSA'nın eğitilerek şifre çözme işlemlerinde kullanıldığı görülmektedir (Shihab, 2006). Yu ve Cao (2006) çalışmalarında kaotik Hopfield ağ yapısını kullanarak şifreleme işlemini gerçekleştirmeye ve Hopfield modeliyle ikili (binary) diziler oluşturmuş kaotik harita seçimi ve rastgele değişim fonksiyonuyla veriyi maskelemeye çalışmışlardır.

Bir diğer çalışmada Volna ve diğ. (2012), YSA'ya dayalı bir şifreleme sistemi oluşturmayı denemiştir. Eğitim seti; küçük harflerden oluşan İngiliz alfabesindeki harfler, boşluk karakteri ve buna ek olarak noktalama işaretlerinin tümünü temsilen ASCII değerlerinden oluşmaktadır. Giriş, gizli ve çıkış katmanlarından oluşan ve her bir katmanda 6 nöron olacak şekilde ÇKA modeli kullanmışlardır. YSA'nın giriş değerleri olarak düz ve şifreli metinler 6 bitlik veri setlerine ayrılarak şifreleme ve şifre çözme işlemlerine sokulmuştur. Anahtar değeri ise YSA'nın topolojik yapısındaki parametrelerden katmanlar arası ağırlık değerleri ve her katmandaki nöron sayısı olarak belirtilmektedir. Buna benzer bir çalışmada Türk ve diğ. (2019), Türkçe karakterler için klasik simetrik şifreleme yöntemlerinden bazılarıyla şifrelenmiş metinlerde YSA ile karakter tahmini yapmaya çalışmışlardır. Çalışmalarında, veri seti oluşturmak ve verinin önişlem basamaklarında kullanılmak üzere bir program tasarlamış, şifreleme ve şifre çözme işlemleri için burada hazırladıkları verileri kullanmışlardır.

Bu çalışmada, gerçeklenen uygulama ile YSA'nın kriptoloji alanına da uygulanabildiği ve makul çözümler üretebildiği gösterilmeye çalışılmıştır. YSA lineer olmayan yapısı sayesinde finans, biyomedikal, robotik, lojistik gibi birçok farklı alandaki problemlere uygulanmış ve başarılı sonuçlar elde edildiği görülmüştür. Bununla birlikte kriptoloji alanında da kullanılmaya başlanmıştır. Bu çalışmayla kriptoloji verilerinin de uygulanabilirlik açısından YSA'ya uygun olduğu ve matematiksel işlemlere gerek kalmadan üzerinde çalışılabileceği gösterilmek istenmiştir.

2. MATERYAL VE YÖNTEM

2.1. Klasik Simetrik Şifreleme Yöntemleri

Kriptolojinin temel yapıtaşları olan gizlilik; kimlik doğrulama, inkar edememe, veri bütünlüğü gibi özellikler şifreleme algoritmaları ve çeşitli protokoller ile sağlanmaktadır (Kara, 2009). Şifrelemenin ve şifre çözme işlemlerinin aynı anahtarla yapıldığı yöntemlere simetrik şifreleme algoritmaları denilmektedir. Kullanılan anahtar sadece alıcı ve göndericide bulunduğundan yöntemin bir diğer adı gizli anahtarlı şifreleme olarak geçmektedir. Simetrik

şifrelemede algoritmalar blok ve dizi şifreleme algoritmaları olarak iki gruba ayrılmaktadır. Blok şifrelemede metin uzunlukları belli olan bloklar halinde şifreleme yapılırken dizi şifrelemede bir üreteç ve anahtar yardımıyla istenilen uzunlukta dizi üretilmektedir. Şifreleme yapılan anahtara açık, şifreyi çözen anahtara gizli anahtar denilen; şifreleme ve şifre çözme anahtarı birbirinden farklı olan yöntemlere asimetrik şifreleme algoritmaları denilmektedir. Açık anahtar herkesle paylaşılabilirken gizli anahtar sadece alıcıda bulunmaktadır. Anahtar kullanılmayan bazı yöntemler de mevcuttur. Bunlardan en bilineni özet (hash) fonksiyonlarıdır. Özet fonksiyonları girdi olarak rastgele uzunlukta metinleri alıp sabit uzunlukta özet çıktılar üretmektedir.

Bu bölümde, bu çalışmada kullanılan klasik simetrik şifreleme yöntemlerinden bahsedilmiştir. Kullanılan denklemlerdeki sembollerden; P düz metni, C şifreli metni, K anahtar uzayını, e_k şifreleme kuralını, d_k şifre çözme kuralını, Z İngiliz alfabesindeki harfler kümesini, x şifrelenecek harfi, y ise şifresi çözülecek harfi temsil etmektedir. Örnek uzayı olarak İngiliz alfabesindeki 26 harf seçildiğinden ötürü işlemler mod 26'da yapılmaktadır. Her bir harfe karşılık 0-25 arası bir değer atanmakta ve anahtar değeriyle birlikte aritmetik işlemler uygulanmaktadır.

2.1.1. Ötelemeli Şifreleme (Shift Cipher)

Ötelemeli Şifreleme modüler aritmetiğe dayanan bir yöntemdir. Bu yöntemde, anahtar tekil bir sayıdır.

$P = C = K = Z_{26}$, $0 \leq K \leq 25, (x, y \, \epsilon \, Z_{26})$ iken,

şifreleme için;

$$e_k(x) = (x + K) \ (mod\ 26) \tag{1}$$

şifre çözme için;

$$d_k(y) = (y - K) \ (mod\ 26) \tag{2}$$

şeklinde ifade edilmektedir (Stinson, 2002).

Burada şifreli veri, (x+K)'nın mod 26'daki değeriyle temsil edilirken şifre çözme işleminde tam tersi işlem yapılmaktadır. Mod 26'da Ötelemeli Şifreleme ile şifrelenen bir veri, geniş kapsamlı anahtar araştırması (exhaustive key search) ile kırılabilmektedir.

2.1.2. Yerdeğiştirme Şifrelemesi (Substitution Cipher)

Yerdeğiştirme Şifrelemesi'nde anahtar alfabetik karakterlerin (ya da seçilen herhangi bir örnek uzayındaki elemanların) permütasyonlarıdır.

$P = C = K = Z_{26}$, $K\{0 - 25 \text{ arası sayıların permütasyonları}\}$ iken,

şifreleme için;

$$e_\pi(x) = \pi(x) \tag{3}$$

şifre çözme için;

$$d_\pi(y) = \pi^{-1}(y) \tag{4}$$

şeklinde ifade edilmektedir (Stinson, 2002).

Denklemlerdeki π gösterimi permütasyonu ifade etmektedir. İngiliz alfabesindeki harfler düşünüldüğünde örneğin "a"ya karşılık "w" harfinin kullanılması şeklinde şifreleme işlemi yapılabilmektedir. Şifre çözme işleminde ise tersten gidilerek "w" harfinin yerine "a"nın

yerleştirilmesi şeklindedir. 26 harfin permütasyonu 26! ve geniş kapsamlı anahtar aramasıyla kırmak pek mümkün olmasa da, şifreli metin içinde harflerin frekans değerleri ve bir araya gelişleri incelenerek çıkarımlar yapmak mümkün olmaktadır.

2.1.3. Doğrusal Şifreleme (Affine Cipher)

Doğrusal Şifreleme'de, anahtar a ve b şeklinde iki tamsayıdan oluşmaktadır. a belirlenirken 26 ile en büyük ortak böleninin (ebob) 1 olmasına dikkat edilmesi gerekmektedir.

$$P = C = K = Z_{26}\ ,\ K = \{(a,b)\epsilon Z_{26} \times Z_{26} : ebob(a,26) = 1\}, K = (a,b)\epsilon K, (x,y\epsilon Z_{26})$$ iken,

şifreleme için;

$$e_k(x) = (ax + b)\ \ (mod\ 26) \tag{5}$$

şifre çözme için;

$$d_k(y) = a^{-1}(y - b)\ (mod\ 26) \tag{6}$$

şeklinde ifade edilmektedir (Stinson, 2002).

ebob(a,26)=1 koşulunu sağlayan 12 farklı a değeri, b içinse 26 farklı değer mevcuttur. Bu işlem bilgisayar kullanılarak gerçekleştirildiğinde 312 (12×26) anahtarın şifre çözme için denenmesiyle şifre kırılabilmektedir (Trappe ve Washington, 2006).

2.1.4. Vigenere Şifrelemesi (Vigenere Cipher)

Vigenere Şifrelemesi'nde seçilen anahtar bir vektördür.

$P = C = K = (Z_{26})^m\ , K = (k_1, k_2, \dots, k_m)$ iken,

şifreleme için;

$$e_k(x_1, x_2, .., x_m) = (x_1 + k_1, x_2 + k_2, .., x_m + k_m) \tag{7}$$

şifre çözme için;

$$d_k(y_1, y_2, .., y_m) = (y_1 - k_1, y_2 - k_2, .., y_m - k_m) \tag{8}$$

şeklinde ifade edilmektedir (Stinson, 2002).

Ötelemeli Şifreleme'deki gibi şifrelenecek metnin değeri karşılık gelen anahtar değeri kadar mod 26'ya göre ötelenir, tek fark anahtarın bir vektör olmasıdır. Vigenere Şifrelemesi'nin kriptanalizinde öncelikle anahtar uzunluğu bulunmaya çalışılır sonrasında frekans analiziyle şifreli metin çözülmeye çalışılmaktadır (Trappe ve Washington, 2006).

2.2. Yapay Sinir Ağları

YSA, örnekler üzerinden öğrenen ve nasıl çıktı üreteceğine karar verebilen; öğrenme, ilişkilendirme, sınıflandırma, genelleme yapma, özellik belirleme ve optimizasyon konularında başarıyla uygulanabilen bilgisayar sistemleridir (Öztemel, 2006). Öğrenme ile elde ettiği bilgiyi nöronlar arası bağlantı değerlerinde saklamaktadır. Daha önce üzerinde işlem yapmadığı örnekler hakkında bilgi üretebilmeleri (adaptif öğrenme), eksik-belirsiz bilgiyle çalışabilmeleri, hata toleransına sahip olup yeni durumlara adapte olabilmeleri YSA'nın tercih edilme sebeplerindendir. YSA'nın Hopfield Ağları (Hopfield Networks), Vektör Kuantizasyon

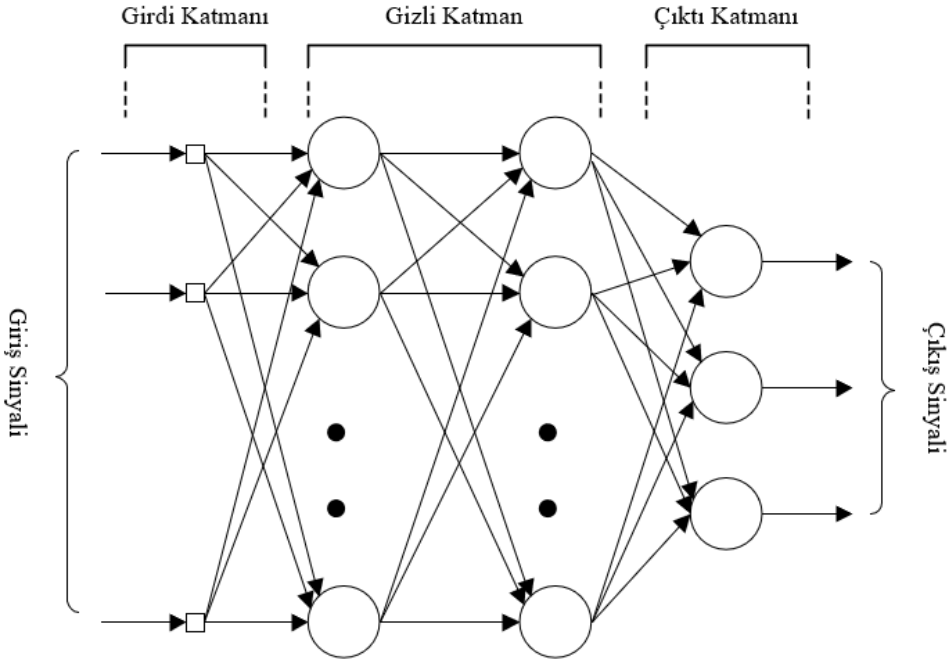
Modelleri (Learning Vector Quantization), Adaptif Rezonans Teorisi (Adaptive Resonance Theory), Kendi Kendine Organize Eden Harita (Self Organizing Map) gibi birçok farklı tipleri mevcuttur. Bu çalışmada Çok Katmanlı Algılayıcı modeli kullanılmıştır.

ÇKA, genel yapısı itibariyle katmanlardan oluşan, ileri beslemeli (feedforward) ve hata geri yayılım (error backpropagation) algoritmasını kullanan YSA modelidir (Fausett, 1994). Ağ; kaynak düğümlerden (source nodes) oluşan girdi katmanı, hesaplama düğümlerinden (computation nodes) oluşan bir veya daha fazla sayıda gizli katman ve çıktı katmanından oluşmaktadır (Haykin, 1998). Burada kaynak düğüm ya da hesaplama düğümlerinden kasdedilen ağdaki nöronlardır.

Ağın her katmanı kendi ağırlık matrisi, eşik (bias) değeri, girdi ve çıktı vektörlerine sahiptir ve her katman farklı sayıda nörona sahip olabilir (Hagan ve diğ., 2014).

Kullanılan hata geri yayılım algoritması temelde ileri ve geri yönlü olarak iki aşamada çalışmaktadır. İleri yönlü iletimde girdi vektörü giriş katmanına verilmekte, ağda ileri yönlü aktarımla hesaplamalar sonucunda çıktı değerleri üretilmektedir. Geri yönlü iletimde ise bağlantıların ağırlık değerleri hata sinyaline göre yeniden düzenlenmektedir. Hata sinyali, hedef çıktı ve ağın ürettiği arasındaki farktır.

ÇKA, Haykin (1998)'e göre üç karakteristik özelliğe sahiptir. Birincisi; ağdaki her nöronun doğrusal olmayan aktivasyon fonksiyonuna sahip olmasıdır. En yaygın kullanılanı sigmoid fonksiyonudur. İkincisi; ağ bir veya daha fazla gizli katman içerebilir ve bu gizli katmanlar girdi ya da çıktı katmanlarının bir parçası değildir. Üçüncüsü; ağın bağlantı derecesinin yüksek olmasıdır. Tam bağlantılı (fully-connected), iki gizli katmana sahip, ileri beslemeli bir ÇKA mimarisi örneği Şekil 1'de gösterilmiştir.



Şekil 1:
ÇKA mimarisi örneği

3. BULGULAR

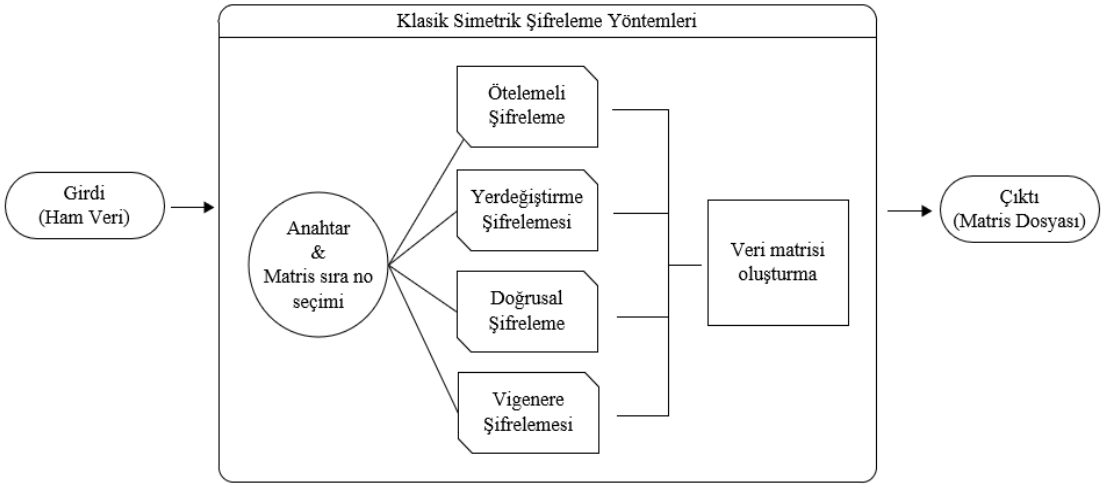
Bu bölümde, geliştirilen uygulamanın tasarım ve gerçekleme aşamaları ve elde edilen sonuçlar ile karşılaştırmalı grafikleri yer almaktadır.

3.1. Uygulama Tasarımı

Bu çalışmada verilen bilgilerin doğrulanması amacıyla veri şifreleme ve YSA işlemleri için Matlab platformunda bir uygulama hazırlanmıştır. Uygulama "Çevrim" ve "YSA" şeklinde 2 sekmeden oluşmaktadır.

Çevrim sekmesinde girilen düz metin ve anahtar değerlerine karşılık Ötelemeli Şifreleme, Yerdeğiştirme Şifrelemesi, Doğrusal Şifreleme ve Vigenere Şifrelemesi yöntemlerine göre şifrelenmiş veri dosyası oluşturulabilmektedir.

"YSA" sekmesinde ise "Çevrim" sekmesinde hazırlanan veri dosyaları kullanılarak YSA eğitim/test işlemleri, performans grafiği ve elde edilen sonuçlar gösterilmektedir. "Çevrim" sekmesinin işlem basamakları Şekil 2'de gösterilmiştir.

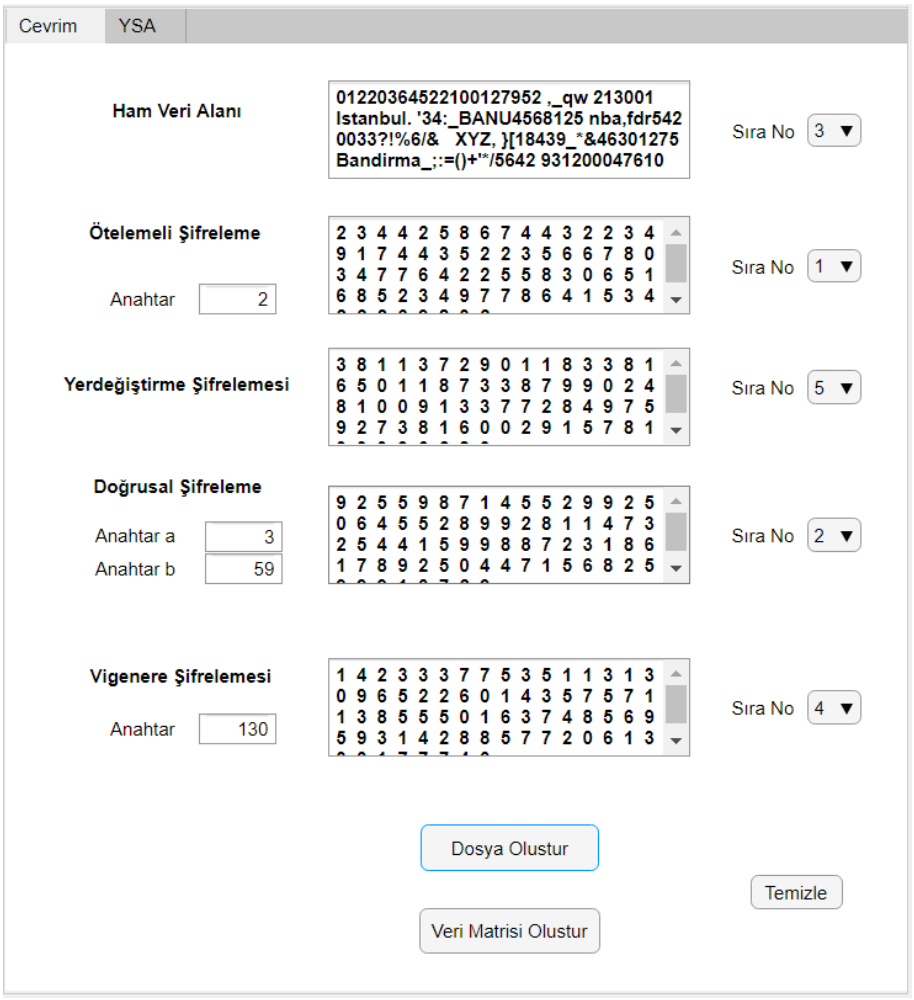


Şekil 2:
Çevrim sekmesinde yapılan işlemlerin blok diyagramı

Uygulamanın "Çevrim" sekmesi şu şekilde kullanılmaktadır:

- Veri matris dosyası oluşturulurken her bir yöntemin yanındaki "Sıra No" isimli açılan listelerden seçim yapılıp, oluşturulacak dosyadaki algoritma sıralaması değiştirilebilmektedir.
- Veri girişinin sadece "Ham Veri Alanı" kısmına yapılması ve algoritmaların anahtar değerlerinin girilmesi yeterli olmaktadır.
- "Dosya Oluştur" butonu, herbir yöntem için şifrelenmiş veriyi .txt dosyasına aktarırken, "Veri Matrisi Oluştur" butonu ise matris formunda .txt dosyası oluşturmaktadır.

"Çevrim" sekmesinin kullanıcı arayüzü Şekil 3'te gösterilmiştir.

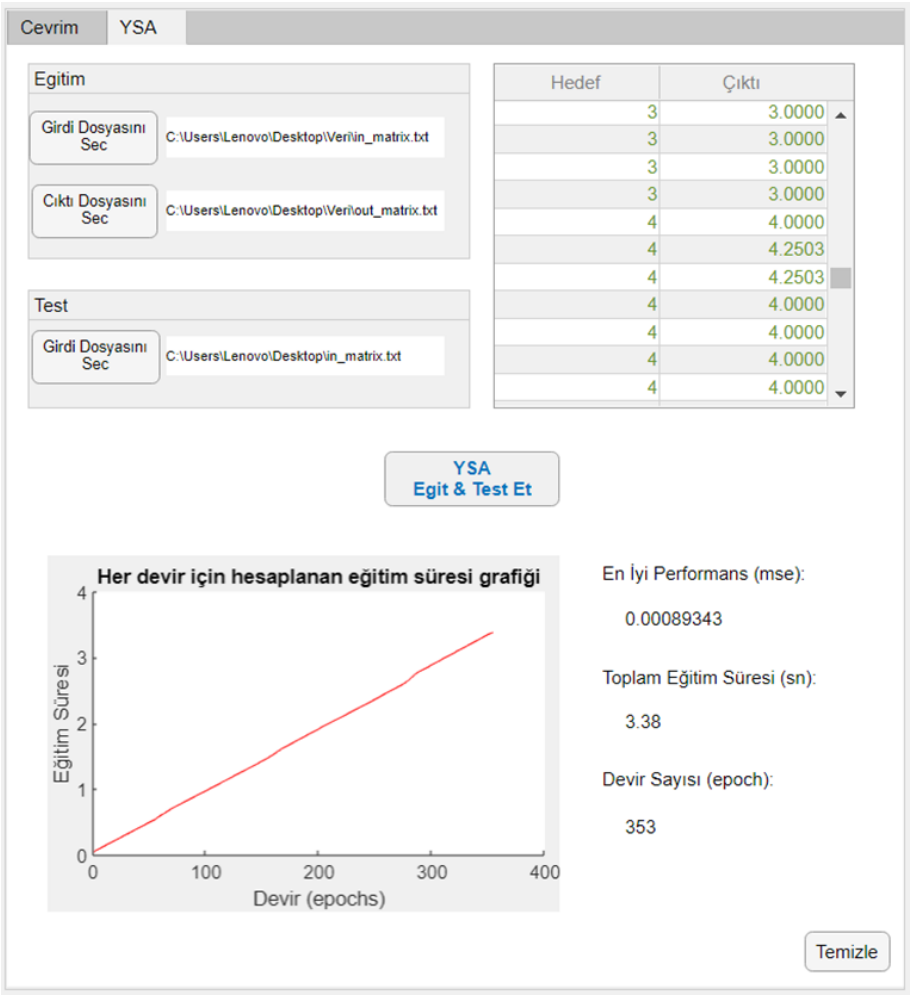


Şekil 3:
Çevrim sekmesinin kullanıcı arayüz görüntüsü

Kullanılan veri genel olarak şifreleme uygulamalarında olduğu gibi 0-9 arası tamsayılardan oluşmaktadır. Herhangi bir sembol, harf, boşluk, yeni satır, noktalama işareti dönüşüm sırasında göz ardı edilmektedir. Aynı şekilde anahtar değerlerinde de sadece tamsayılar geçerli olmaktadır. Doğrusal Şifrelemede, a anahtar değeri için tersi alınabilen tamsayılardan sadece {1, 3, 7, 9} değerleri geçerlidir.

"YSA" sekmesinde kullanıcı ağ eğitimi için girdi ve çıktı dosyalarını, test için ise sadece girdi dosyasını seçebilmektedir. "Çevrim" sekmesinde hazırlanan veri matris dosyaları burada kullanılmaktadır.

Girdi matris dosyalarının içinde 2 özellik değeri yer almaktadır. Bunlar şifreli ve düz metindeki karakterlerin sayısal karşılıklarıdır. Çıktı matris dosyasında ise şifreli metnin ait olduğu sınıf değeri yer almaktadır. Düz metin (ham veri) 1., Ötelemeli Şifreleme 2., Yerdeğiştirme Şifrelemesi 3., Doğrusal Şifreleme 4., Vigenere Şifrelemesi 5. sınıf olarak tanımlanmıştır. Ağ kısmında, ÇKA temel mimarisine sahip yüzeysel bir YSA modeli oluşturulmuştur. "YSA Eğit & Test Et" butonuna basıldığında seçilen girdi-çıktı dosyaları okunup ağ eğitilmeye başlanmakta, sonrasında test için seçilen girdi dosyası verileri ağa tanıtılıp ağın ürettiği çıktı tabloda gösterilmektedir. "YSA" sekmesinin kullanıcı arayüzü Şekil 4'te gösterilmiştir.



Şekil 4:
YSA sekmesinin kullanıcı arayüz görüntüsü

Grafikteki y ekseninde zaman (sn) ve x ekseninde devir (epoch) sayısı gösterilmektedir. Aynı zamanda en iyi performans değeri, toplam eğitim süresi (sn) ve devir sayısı da kullanıcı arayüzünde ayrıca belirtilmektedir. Performans değeri Ortalama Hata Kareleri (Mean Squared Error) formülüyle hesaplanmıştır.

Oluşturulan YSA modeli için ileri beslemeli ağ yapısı tercih edilmiş olup geri yayılımlı öğrenme algoritmalarından en bilinenleri denenmiş ve en uygun öğrenme algoritması tespit edilmeye çalışılmıştır. Bunlar; Levenberg-Marquardt, Bayesian Regularization, BFGS Quasi-Newton (BFGS: Broyden, Fletcher, Goldfarb, Shanno), Ölçeklenmiş Konjuge Gradyan (Scaled Conjugate Gradient), Fletcher-Powell Konjuge Gradyan, Bir Adım Sekant (One Step Secant), Gradyan Azalım (Gradient Descent) ve Momentumlu Gradyan Azalım (Gradient Descent with Momentum) algoritmalarıdır.

Aktivasyon fonksiyonlarından; doğrusal, sigmoid ve hiperbolik tanjant transfer fonksiyonları, katman başlatma fonksiyonu olarak Nguyen-Widrow, ağırlık ve eşik değer (bias) katman başlatma fonksiyonları denenmiştir.

YSA yapısı üç katmanlı olacak şekilde oluşturulmuştur. Giriş katmanında 4 nöron, gizli katmanda 3 nöron ve çıkış katmanında 1 nöron tanımlanmış, ilk 2 katman için eşik değerler verilmiştir. Öğrenme katsayısı ve momentum katsayısı rastgele olarak sırasıyla 0.2 ve 0.3 olarak seçilmiştir.

Denemeler sonucunda YSA'da en iyi performanslar; Levenberg-Marquardt öğrenme algoritması, sigmoid aktivasyon fonksiyonu ve katmanlar için başlatma fonksiyonu olarak Nguyen-Widrow fonksiyonuyla yakalanmıştır. Bu denemeler, literatürdeki genel kullanım örnekleri dikkate alınarak rastgele eşleşmelerle gerçekleştirilmiştir.

Levenberg-Marquardt algoritmasının diğer algoritmalara göre daha hızlı çalıştığı bilinmektedir ve literatürde en çok tercih edilen algoritmalardan birisidir.

3.2. Uygulama Sonuçları

YSA'ya 5 farklı sınıfa ait veri tanıtımı yapılmış ve test aşamasında en başarılı tahminler ham veri yani düz metinde ve Ötelemeli Şifreleme'de görülmüştür. En başarısız sınıflandırma Yerdeğiştirme Şifrelemesi'nde gözlemlenmiştir. Şifreli metinde diğer yöntemlerle benzer değerler oluşabildiğinden daha başarılı sonuçlar için anahtar değerinin dikkatli seçilmesi gerekmektedir. Kullanılan anahtar değerlerinin sınıflandırma başarısına doğrudan etki ettiği görülmüştür. Veri dosyasının büyüklüğünün YSA başarısına olumlu ya da olumsuz kayda değer bir etkisi görülmemiştir.

Hata oranı hesaplamasında Ortalama Hata Kareleri ve Ortalama Yüzde Mutlak Hata (Mean Absolute Percentage Error) formüllerinden yararlanılmıştır. Ortalama Hata Kareleri formülü (9) denklemindeki şekliyle ifade edilmektedir (Gallant, 1993);

$$\text{Ortalama Hata Kareleri} = \frac{1}{N}\sum_{k=1}^{N}(W.E^k - C^k)^2 \tag{9}$$

Burada W; ağırlık değerini, N; örnek sayısını, E^k; k. örneğin değerini, C^k; k. çıktının gerçek değerini ifade etmektedir.

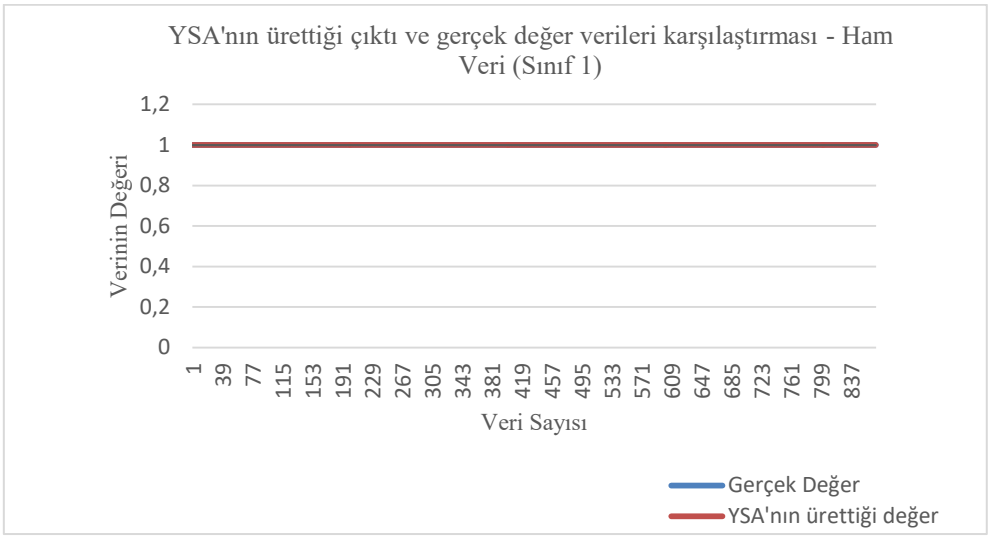
Ortalama Yüzde Mutlak Hata formülü (10) denklemindeki şekilde ifade edilmektedir (Soto ve diğ., 2018);

$$\text{Ortalama Yüzde Mutlak Hata} = \frac{100\%}{N}\sum_{t=1}^{N}\left|\frac{(A_t - P_t)}{A_t}\right| \tag{10}$$

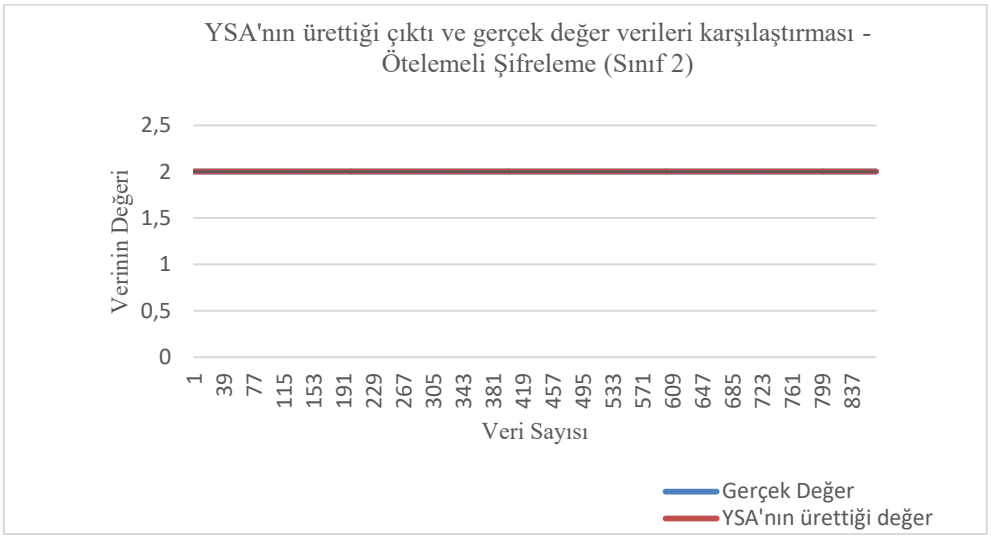
Burada N; örnek sayısını, A_t ; gerçek değeri, P_t; tahmin edilen değeri ifade etmektedir. Ağın ürettiği çıktı ve hedef değer arasında en fazla 1,5 değerinde sapma yakalanmıştır. Ortalama Yüzde Mutlak Hata değeri %2,726 olarak ölçülmüştür. YSA eğitimi sırasında elde edilen en iyi Ortalama Hata Kareleri değeri 0,000893 olarak tespit edilmiştir. Test verilerine karşılık ağın ürettiği değerler karşılaştırması Şekil 5'te gösterilmiştir. YSA'nın yöntemlere göre doğru/yanlış veri tahmini sayısı tablosu Tablo 1'de gösterilmiştir.

Tablo 1. YSA'nın yöntemlere göre doğru/yanlış veri tahmini sayısı

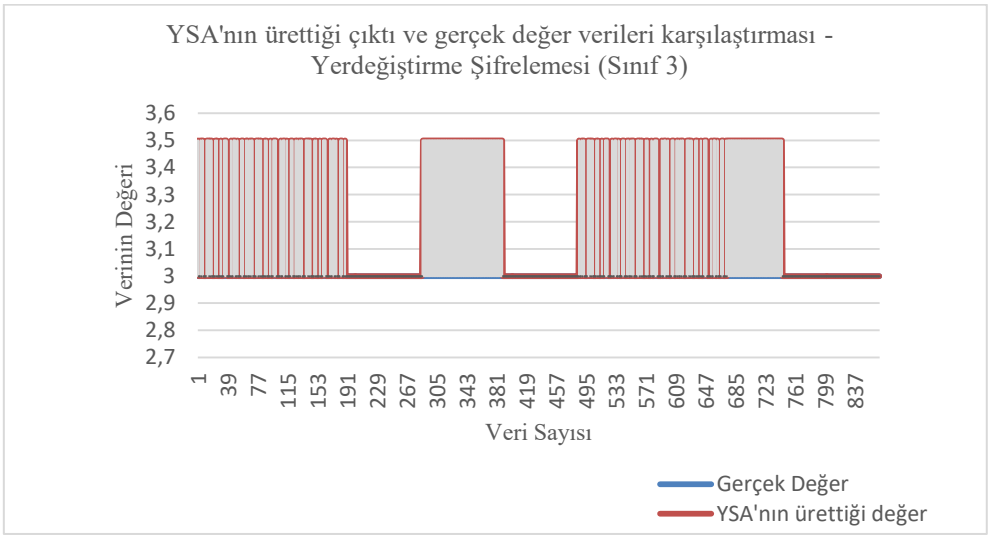
Yöntem	Sınıfı	Doğru Tahmin Sayısı	Yanlış Tahmin Sayısı	Yüzde Başarı	Toplam Veri Sayısı
Düz Metin (Ham Veri)	1	868	0	100	868
Ötelemeli Şifreleme	2	868	0	100	868
Yerdeğiştirme Şifrelemesi	3	546	322	62,90	868
Doğrusal Şifreleme	4	578	290	66,59	868
Vigenere Şifrelemesi	5	666	202	76,73	868


YSA'nın ürettiği çıktı ve gerçek değer verileri karşılaştırması - Ham Veri (Sınıf 1)
Verinin Değeri
1,2
1
0,8
0,6
0,4
0,2
0
1 39 77 115 153 191 229 267 305 343 381 419 457 495 533 571 609 647 685 723 761 799 837
Veri Sayısı
Gerçek Değer
YSA'nın ürettiği değer

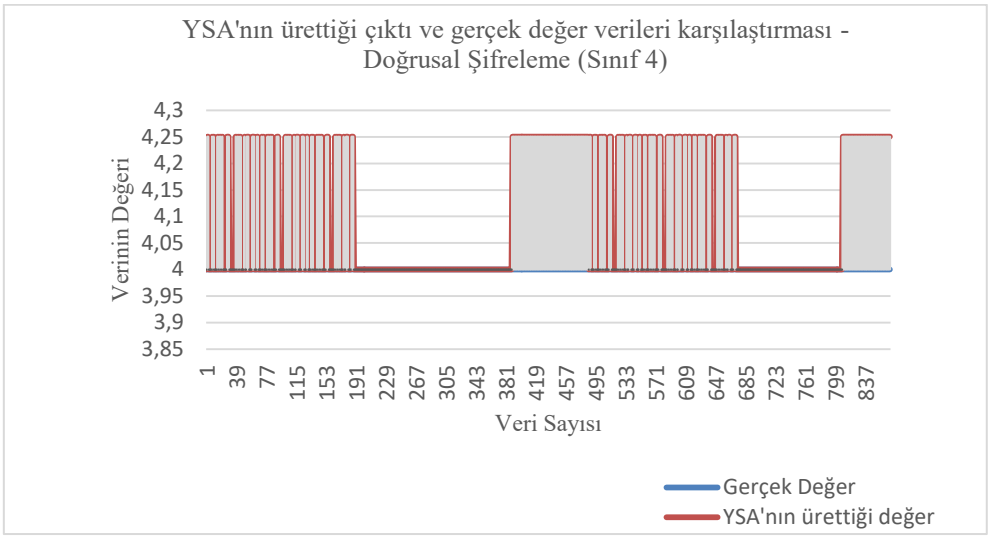
a.


YSA'nın ürettiği çıktı ve gerçek değer verileri karşılaştırması - Ötelemeli Şifreleme (Sınıf 2)
Verinin Değeri
2,5
2
1,5
1
0,5
0
1 39 77 115 153 191 229 267 305 343 381 419 457 495 533 571 609 647 685 723 761 799 837
Veri Sayısı
Gerçek Değer
YSA'nın ürettiği değer

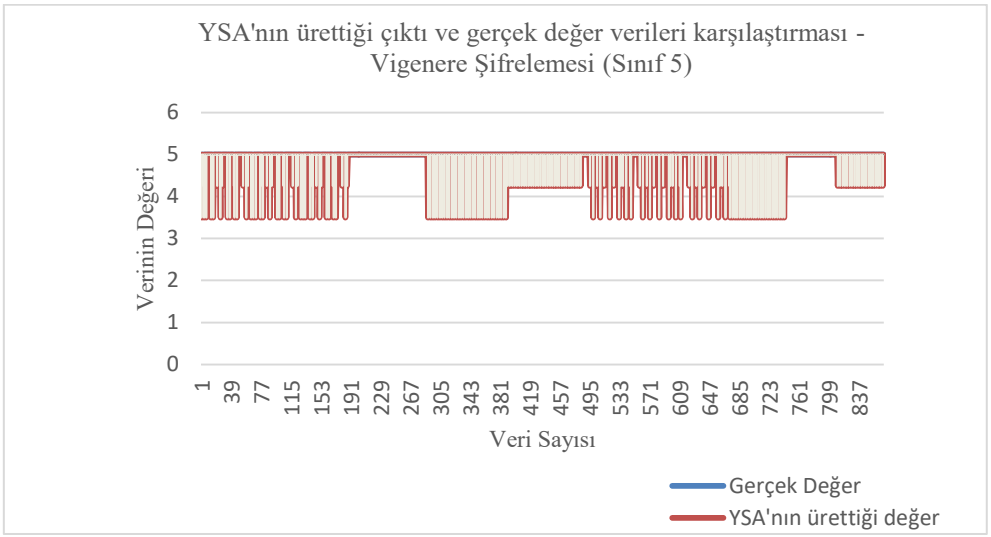
b.


YSA'nın ürettiği çıktı ve gerçek değer verileri karşılaştırması - Yerdeğiştirme Şifrelemesi (Sınıf 3)
Verinin Değeri
3,6
3,5
3,4
3,3
3,2
3,1
3
2,9
2,8
2,7
1 39 77 115 153 191 229 267 305 343 381 419 457 495 533 571 609 647 685 723 761 799 837
Veri Sayısı
Gerçek Değer
YSA'nın ürettiği değer

c.



d.



e.

Şekil 5:
Şifreleme yöntemleri için gerçek değer ve YSA'nın ürettiği çıktı karşılaştırma grafikleri;
a. *Sınıf 1* ***b.*** *Sınıf 2* ***c.*** *Sınıf 3* ***d.*** *Sınıf 4* ***e.*** *Sınıf 5 için*

4. DEĞERLENDİRME VE SONUÇ

Bu çalışmada; Ötelemeli Şifreleme, Yerdeğiştirme Şifrelemesi, Doğrusal Şifreleme, Vigenere Şifrelemesi ile şifrelenen metinler YSA ile sınıflandırılmaya çalışılmıştır. Şifreli metinlere ek olarak düz metin kategorisi de oluşturulmuştur. Düz metin, sınıf 1 olmak üzere her bir yöntem sırayla 5. sınıfa kadar ayrı bir sınıf ile temsil edilmektedir.

Veri seti oluşturma ve YSA ile gerçekleştirilecek işlemler için, hazır program eklentileri kullanılmadan bir kullanıcı arayüz programı geliştirilmiştir. Kullanıcı uygulamanın "Çevrim" sekmesinde .txt tipinde matris dosyası oluşturup başka bir programa ihtiyaç duymadan uygulamanın "YSA" sekmesinde hazırladığı bu dosyayı yükleyerek sonuçları sekmenin içindeki tablo alanında görebilmektedir. Ayrıca elde ettiği sonuçları .txt dosyası olarak alabilmektedir.

Performans değerlerini de sonuç tablosuyla aynı anda grafik alanında gözlemleyebilmektedir. Bu özellikleri sayesinde çalışmada kullanıcı kolaylığı sağlanmıştır.

Uygulamanın "YSA" sekmesinde kullanılan YSA modeli ÇKA temel mimarisine sahiptir. Matlab platformunda kodlanan bu mimari, temel yapıları içermesinin yanısıra kullanılan YSA parametrelerinde (örneğin katmanların nöron sayısı, bağlantılar, katmanların aktivasyon ve başlangıç fonksiyonları gibi) farklı seçimler denenmiştir. Hazır program eklentileri çoğunlukla YSA'nın mimari yapısında kısıtlı değişime izin vermektedir. Geliştirilen uygulama ile bu kısıtlardan daha az etkilenilmiştir.

YSA'nın nümerik değerlerle çalıştığı göz önüne alındığından, üzerinde çalışılan örnek uzayı 0-9 arası tamsayılar olarak seçilmiştir. Örnek uzayı boyutunun küçüklüğünden faydalanılarak veriler üzerinde herhangi bir normalizasyon işlemine ihtiyaç duyulmamıştır.

Kullanılan algoritmaların matematiksel yapılarınında kimi zaman kısıtlayıcı olduğu görülmüştür. Bu çalışmada kullanılan Doğrusal Şifreleme yönteminin, ilk anahtar değeri (a değişkeni) tersi alınabilir olması gerekmektedir. Dolayısıyla seçilen örnek uzayında 0-9 arası tamsayıların sadece "1,3,7,9" değerlerinin tersi alınabilmektedir. Bu kurala uygun olarak veri seti oluşturulmuş ve YSA eğitiminde kullanılmıştır.

Yapılan denemelerde YSA'ya çok sayıda ve mümkün olduğunca birbirinden farklı örnekler tanıtıldığında sınıflandırma işlemindeki başarı oranının da arttığı görülmüştür. Şifreleme algoritmalarının matematiksel yapısı bilinmese dahi şifreli ve düz metin örnekleriyle YSA ile etkili tahminler yapılabileceği gözlemlenmiştir.

Literatürde, araştırmacıların çoğunun İngiliz alfabesi üzerinde çalışmayı tercih ettikleri görülmektedir. İleride yapılabilecek çalışmalarda; lokal diller veya birden fazla dilden oluşan veri setleri, semboller üzerinden ya da büyük-küçük harf duyarlılığı, noktalama işaretleri, boşluk karakteri, yeni satır gibi özel karakterlerin dahil edilip kullanıldığı farklı veri setleri üzerinde çalışılabilir. Simetrik şifreleme yöntemlerinin yanısıra asimetrik şifreleme yöntemleri ile YSA üzerinden tahminlerin yapıldığı Nöral Kriptografi çalışmaları yapılabilir.

5. TEŞEKKÜR/DESTEK

Bu çalışma Doç. Dr. Rüya ŞAMLI danışmanlığında gerçekleştirmiş olduğum ve 2017 yılında tamamladığım yüksek lisans tez çalışmamın bir bölümünü oluşturmaktadır. Ayrıca bu çalışma esnasında TÜBİTAK 118E682 numaralı projeden destek alınmıştır.

KAYNAKLAR

1. Abd, A.J., Al-Janabi, S.T.F. (2019) Classification and Identification of Classical Cipher Type Using Artificial Neural Networks, *Journal of Engineering and Applied Sciences*, 14(11), 3549-3556. doi: 10.36478/jeasci.2019.3549.3556

2. Chandra, B., Varghese, P.P., Saxena, P.K., Kant, S. (2007) Neural Networks for Identification of Crypto Systems, *The 3rd Indian International Conference on Artificial Intelligence (IICAI-07)*, 402-411.

3. Dileep, A.D., Sekhar, C.C. (2006) Identification of Block Ciphers Using Support Vector Machines, *The IEEE International Joint Conference on Neural Network Proceedings*, 2696-2701. doi: 10.1109/IJCNN.2006.247172

4. Dunham, J.G., Sun, M.T., Tseng, J.C.R. (2005) Classifying File Type of Stream Ciphers in Depth Using Neural Networks, *The 3rd ACS/IEEE International Conference onComputer Systems and Applications*. doi: 10.1109/AICCSA.2005.1387088

5. Fausett, L.V. (1994) *Fundamentals of Neural Networks: Architectures, Algorithms and Applications,* Prentice Hall, United States.

6. Gallant, S.I. (1993) *Neural Network Learning and Expert Systems*, MIT Press, London, England.

7. Hagan, M.T., Demuth, H.B., Beale, M.H., De Jesus, O. (2014) *Neural Network Design*, Martin Hagan.

8. Haykin, S. (1998) *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Delhi, India.

9. Kara, O. (2009) Kriptografinin Yapıtaşları Kriptografik Algoritmalar ve Protokoller, *Bilim ve Teknik*, 500, 34-41.

10. Khadivi, P., Momtazpour, M. (2010) Cipher-Text Classification With Data Mining, *The 4th IEEE International Symposium on Advanced Networks and Telecommunication Systems*, 64-66. doi: 10.1109/ANTS.2010.5983530

11. Öztemel, E. (2006) *Yapay Sinir Ağları,* Papatya Yayıncılık, İstanbul, Türkiye.

12. Piper, F. (1997) Introduction to Cryptology, *Information Security Technical Report*, 10-13.

13. Sharif, S.O., Kuncheva, L.I., Mansoor, S.P. (2010) Classifying Encryption Algorithms Using Pattern Recognition Techniques, *The IEEE International Conference on Information Theory and Information Security*, 1168-1172. doi: 10.1109/ICITIS.2010.5689769

14. Shihab, K. (2006) A Cryptographic Scheme Based on Neural Networks, *The 10th WSEAS International Conference on Communications*, 7-12.

15. Soto, J., Melin, P., Castillo, O. (2018) *Ensembles of Type 2 Fuzzy Neural Models and Their Optimization with Bio-Inspired Algorithms for Time Series Prediction*, Springer.

16. Stinson, D.R. (2002) *Cryptography: Theory and Practice*, Chapman & Hall/CRC Press.

17. Tan, C., Ji, Q. (2016) An Approach to Identifying Cryptographic Algorithm from Ciphertext, *The 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, 19-23. doi: 10.1109/ICCSN.2016.7586649

18. Trappe, W., Washington, L.C. (2006) *Introduction to Cryptography with Coding Theory*, Pearson Prentice Hall.

19. Türk, S. (2017). Yapay Sinir Ağları Kullanılarak Şifreleme Yöntemlerinin Performans Analizlerinin Gerçekleştirilmesi, *Yüksek Lisans Tezi,* İstanbul Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

20. Türk, S., Samli, R., Orman, Z. (2019) A Sample Substitution Cipher Data Processing Using Neural Networks, *The 4th International Conference on Theoretical and Applied Computer Science and Engineering (ICTACSE)*, 18-22.

21. Volna, E., Kotyrba, M., Kocian, V., Janosek, M. (2012) Cryptography Based On Neural Network, *The 26th European Conference on Modelling and Simulation (ECMS)*, 386-391.

22. Yee, L.P., Silva, L.C.D. (2002) Application of Multilayer Perceptron Networks in Public Key Cryptography, *The International Joint Conference on Neural Networks (IJCNN)*, 1439-1443. doi: 10.1109/IJCNN.2002.1007728

23. Yu, W., Cao, J. (2006) Cryptography Based on Delayed Chaotic Neural Networks, *Physics Letters A*, 356 (4-5), 333-338. doi: 10.1016/j.physleta.2006.03.069