



## FARKLI ÖRNEKLEME YÖNTEMLERİNİN YAPAY SİNİR AĞLARINDA PERFORMANSININ KARŞILAŞTIRILMASI

Ahmet ULUSLU\*

İstanbul Üniversitesi-Cerrahpaşa, Elektronik ve Otomasyon Bölümü, İstanbul, Türkiye

Anahtar Kelimeler	Öz
YSA, ÇKA, Latin Hiperküp, Monte Carlo, Modelleme.	Optimizasyon modelleme problemlerinde, kullanılacak verinin seçimi büyük önem arz etmektedir. Bu çalışmada farklı örnekleme yöntemlerinin Yapay Sinir Ağlarının başarısındaki etkisini, anten tasarımlarında sıkça kullanılan model olan Çok Katmanlı Algılayıcı kullanılarak mikroşerit yama anteni üzerinde performans karşılaştırılması yapılmaktadır. Seçilen Yapay Sinir Ağı kara kutu modeli 5 giriş ve 1 çıkış parametrelerinden oluşmaktadır. Mikroşerit yama anten modellenmesinde kullanılacak veri seçiminde Latin Hiperküp ve Monte Carlo örnekleme yönteminin uygulanması incelenmiştir. Öncelikle örnek sayısı (eğitim ve doğrulama veri setleri) giriş parametrelerinin belirlenen değer aralıklarında Latin Hiperküp ve Monte Carlo örnekleri kendilerine özgün oluşturulma yöntemine göre temin edilmiştir. Akabinde oluşturulan 2 farklı örnek sayısına sahip problemler kendi içlerinde %50 ve %33 oranında rastgele ayrılarak eğitim ve doğrulama verileri oluşturulmuştur. Performans karşılaştırmasında, 3 algoritma ile 4 farklı mimari yapıya sahip toplamda 12 farklı ağ ile birlikte 4 farklı sayıda eğitim ve doğrulama veri setleri kullanılmaktadır. Çıkan sonuçlar birbirleri ile mukayese edildiğinde, düşük veya yüksek eğitim örnek sayısına sahip modellemede de Monte Carlo örnekleme yönteminin performans olarak daha başarılı sonuçlar verdiği görülmektedir. Buna karşın kendi içinde Latin Hiperküp örnekleme yönteminin ise eğitim örnek sayısında artışa gidilmesi kısmi iyileşmeye neden olduğu görülmektedir. Fakat yine de daha az örnek sayısına sahip olan diğer Monte Carlo örnekleme yönteminin performans olarak gerisinde kalmıştır. Dolayısı ile Monte Carlo örnekleme yönteminin bu ve benzer problemler için daha uygulanabilir olduğuna kanaat getirilmiştir.

## COMPARISON OF PERFORMANCE OF DIFFERENT SAMPLING METHODS IN ARTIFICIAL NEURAL NETWORKS

Keywords	Abstract
ANN, MLP, Latin Hypercube, Monte Carlo, Modelling.	In optimization modeling problems, the selection of the data to be used is of great importance. In this study, the effect of different sampling methods on the performance of Artificial Neural Networks is compared on the microstrip patch antenna by using the Multilayer Perceptron, which is the model that is frequently used in antenna designs. The selected neural network black box model consists of 5 input and 1 output parameters. The application of Latin Hypercube and Monte Carlo sampling methods was examined in the selection of data to be used in microstrip patch antenna modeling. First of all, Latin Hypercube and Monte Carlo samples were obtained according to their unique creation method in the specified value ranges of the sample number (training and validation data sets) input parameters. Afterwards, the problems with 2 different sample numbers were randomly separated at a rate of 50% and 33%, and training and validation data were created. In the performance comparison, a total of 12 different networks with 3 algorithms and 4 different architectural structures, as well as 4 different training and validation data sets are used. When the results are compared with each other, it is seen that Monte Carlo sampling method gives more successful results in terms of

\* İlgili yazar / Corresponding author: auluslu@iuc.edu.tr, +90-212-866-3700

performance in modeling with low or high educational sample number. On the other hand, it is seen that the Latin Hypercube sampling method, on the other hand, increased the number of training samples and caused a partial improvement. However, it still lags behind the other Monte Carlo sampling method, which has less sample size, in terms of performance. Therefore, it was concluded that the Monte Carlo sampling method is more applicable for this and similar problems.

#### Alıntı / Cite

Uluslu, A., (2022). Comparison of Performance of Different Sampling Methods in Artificial Neural Networks, Mühendislik Bilimleri ve Tasarım Dergisi, 10(2), 412-428.

#### Yazar Kimliği / Author ID (ORCID Number)

A. Uluslu, 0000-0002-5580-1687

#### Makale Süreci / Article Process

<b>Başvuru Tarihi / Submission Date</b>	19.10.2021
<b>Revizyon Tarihi / Revision Date</b>	30.01.2022
<b>Kabul Tarihi / Accepted Date</b>	19.02.2022
<b>Yayın Tarihi / Published Date</b>	30.06.2022

## 1. Giriş (Introduction)

Teknolojini ilerlemesine karşın isteklerinde bununla birlikte hızla artmasından dolayı elektromanyetik (EM) simülasyonların uzun süreler aldığından, mikrodalga bileşen modellemesinin hesaplama maliyetini azaltmak için polinom regresyon (Chávez-Hurtado vd., 2016; Rayas-Sánchez vd., 2015; Rayas-Sánchez vd., 2010; Rayas-Sánchez vd., 2010), destek vektör makinesi (Khusro vd., 2018; Güneş vd., 2014; Geng vd., 2014; Jacobs vd., 2013; Tokan vd., 2008) ve Yapay Sinir Ağları (YSA) gibi çeşitli vekil modeller oluşturulmuştur (Yuan vd., 2020; Xiao vd., 2018; Kapetanakis vd., 2018; Xiao vd., 2021; Mahouti vd., 2017; Uluslu, 2021; Uluslu, 2021). Bunların arasından YSA, karmaşık doğrusal olmayan problemler için doğruluğu ve verimli modelleme yeteneği nedeniyle mikrodalga bileşen vekil modelinde yaygın olarak tercih edilmektedir. Mikrodalga bileşeni için YSA modellerinin araştırılması esas olarak modelleme yöntemlerine odaklanır ve farklı ağların kıyaslamaları mevcuttur (Uluslu, 2021; Güneş vd., 2017; Mahouti vd., 2014). Örneklem, modellemenin temelidir ve kalitesi, vekil modellemenin ve vekil destekli optimizasyonun verimliliğini ve doğruluğunu direk olarak etkilemektedir. Monte Carlo örneklem yöntemi, güncel bilimsel araştırmalarda yaygın olarak kullanılan bir örneklem yöntemidir, ancak düşük örneklem verimliliği, büyük miktarda hesaplama gibi çeşitli hataları vardır (Lei, 2003). Bununla birlikte McKay, örneklem verimliliği yüksek ancak örnek büyüklüğü küçük olan Latin Hiperküp örneklem yöntemini geliştirmiştir (McKay vd., 1979). Bu makalede, bu iki yöntemin matematiksel ilkeleri ve çalışma adımları kısaca tanıtılacak ve mikroşerit yama anten modelleme probleminde uygulanacaktır. Bunun yanı sıra, yazarın bildiği kadarıyla bu tür problemde bu iki örneklem yönteminin bu kadar detaylı karşılaştırılmasına rastlanmamıştır. Bu alana hizmet eden daha iyi olanı seçmek için bu iki yöntemin pratik etkileri birbiriyle karşılaştırılacaktır. Çoğunlukla anten tasarım problemlerinde daha yaygın olarak optimizasyon yöntemiyle istenilen özellikler elde edilmeye çalışılır. Ancak bu yöntem anten simülasyon süresi dikkate alındığında oldukça maliyetlidir. Çalışmada, birçok tasarım probleminde öncülük etmiş, 2-3GHz bant aralığında herhangi bir noktada çalışan mikroşerit yama antenin tasarım modelleme probleminin çözümü için, düşük maliyetli, hızlı, verimli, doğru ve güvenilir tabanlı bir vekil model kullanılmıştır. Anten EM simülasyonları için MATLAB 2021a programı kullanılmıştır. Tüm bu işlemler 8. nesil Intel Core i7 CPU, 3.20 GHz işlemcili, 8 GB RAM'e sahip bir bilgisayar tarafından gerçekleştirilmiştir. Tek/çift bant çalışan bir antenin  $S_{11}$  (dB) parametresinin farklı boyutlarda belirlenen frekans bandı boyunca değişimini tahmin etmek için bir YSA modelinde kullanılacak olan veri setinin farklı örneklem yöntemleri ile performansının karşılaştırılması yapılmıştır.

Bu çalışma, Latin Hiperküp örneklem yani yerel örneklem yöntemi ile Monte Carlo örneklem yöntemi yani küresel örneklem yönteminin performans karşılaştırması üzerine dayandırılmıştır. Yerel örneklem yöntemi, iyi örneklerin kullanımını artırmak için açgözlülük stratejisiyle tasarlanmıştır. Küresel örneklem yöntemi ise, tasarım alanını keşfetmek için örnek tekdüzeliğini güçlendirmek için minimum mesafe reddine sahip yöntemdir. Doğru örneklem yapılması bütün modelleme problemlerinde daha az hataya sahip sonuçlar bulunmasına neden olmaktadır. Bu nedenle araştırma ihtiyacını karşılamak için sınırları iyi seçmek gerekmektedir. Ayrıca bu çalışma, örneklemenin vekil model oluşturma sürecinde temel ve önemli bir çalışma olduğu anlamına gelebilir. Bu makalenin temel amacı, bilimsel ve makul örneklem yöntemini bulmaktır.

## 2. Kaynak Araştırması (Literature Survey)

Modelleme çalışmalarında veri setini elde etmek için, Latin Hiperküp örneklem (LHÖ) (McKay vd., 1979) ve Monte Carlo örneklem (MCÖ) vekil destekli mikrodalga bileşen modelleme ve optimizasyon problemlerinde yaygın olarak tercih edilmektedir. Belirtilen bu örneklem yöntemleri, tasarım alanında dağıtılmış örneklerin tek biçimde elde edilmesini geliştirmeye odaklanır. Mikrodalga bileşenleri filtre anten vb. başta olmak üzere genellikle hassas olduklarından, tasarım kriterlerini sağlayan iyi numuneler muhtemelen tasarım uzayının küçük

bölgelerinde bulunur (Uluslu, 2021; Mahouti vd., 2021; Mahouti vd., 2021). Tek tip örnekleme yöntemleri kullanılıyorsa, iyi örnekler (tasarım özelliklerini karşılayan) ve kötü örneklerden (tasarım özelliklerini karşılamayan) oluşur. Bu kötü örneklere dayanan vekil modeller, doğru optimizasyona izin vermeyebilir. Bu nedenle, vekil modelleme ve optimizasyon için hem örnek tekdüzeliğine hem de iyi örneklerle odaklanan yüksek kaliteli bir örnekleme yöntemi istenmektedir. Son yıllarda mikrodalga bileşenlerinin modelleme ve optimizasyon performansını iyileştirmek için çeşitli numune doldurma yöntemleri önerilmiştir (Liu vd., 2018; Güneş vd., 2017). Bir çalışmada, modelleme tekniği olarak interpolasyon ve ekstrapolasyon yöntemi önerilmiştir (Güneş vd., 2017). Bu yöntem genel olarak modelleme doğruluğunu artırır. Ancak eklenen örnekler iyi örneklerle ait olmayabileceğinden, bu tekniğe dayalı vekil model doğru optimizasyona izin vermeyebilir. Ayrıca Çok Katmanlı Algılayıcı (ÇKA) en temel modelleme yöntemlerinden biridir. Özellikle mikrodalga alanında, filtre ve anten modelleri olmak üzere birçok makalede yer almıştır (Uluslu, 2021; Mahouti, vd., 2020; Mahouti, vd., 2021; Uluslu, 2021). Ayrıca algoritması geliştirilmiş vekil model ile yapılmış çalışmada mevcuttur (Çalık, vd., 2019). Yapılan çalışmada kullanılacak olan katman ve nöron sayısı seçimi daha önceden yapılmış bir çalışmadaki sonuçlara dayanarak belirlenmiştir (Mahouti, P., 2019).

Benzer çalışmalardaki durumlara bakılacak olursa;

Yapılan bir çalışmada standart yapıya sahip bir mikroşerit anten YSA ailesine ait olan Radyal tabanlı fonksiyon (RBF) kullanılarak modellenmiştir. Çalışmada 9 giriş ve 2 çıkış parametresi kullanılmıştır. Elde edilen en başarılı sonuçtaki hata değeri 0,327 (MSE) olarak belirtilmiştir (Bose, vd., 2012).

Bir diğer çalışmada ise E şekilli bir yama antenin modellenmesi üzerinedir. Çalışmada kullanılan YSA 3 giriş, 2 gizli ve 1 çıkış katmanından oluşmaktadır. Toplamda 144 adet örnek sayısı kullanılmıştır. Bunların 130 tanesi eğitim, kalan 14 tanesi ise doğrulama verisi olarak seçilmiştir. Veri seçimi rastgele olarak yapılmaktadır. Çalışmada en başarılı sonuçtaki hata değeri 0,523 (MSE) olarak elde edildiği belirtilmiştir (Akdağlı, vd., 2013).

Yapılan bir başka çalışmada ise L şekilli yama anten modellemesi üzerine çalışılmıştır. Çalışmada 6 giriş, 3 gizli ve 1 çıkış katmanına sahip mimariden oluşan YSA kullanılmıştır. Toplam örnek sayısı 192 olarak belirtilmiştir. Bu örneklerin 172'si eğitim, kalan 20'si ise doğrulama verisi olarak ayrılmıştır. Verilerin seçimi rastgele olarak gerçekleştirilmiştir. Elde edilen en başarılı sonuca ait hata değeri ise 0,537 (MSE) olarak belirtilmiştir (Kayabaşı, vd., 2014).

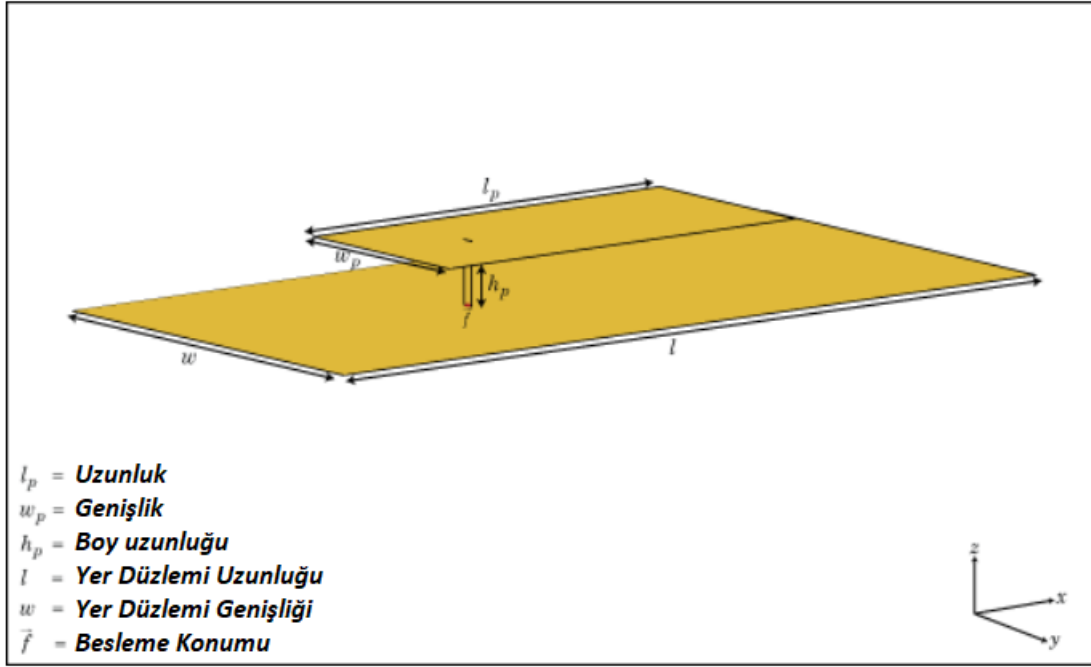
Yine aynı çalışma ekibi bu sefer A şekilli bir yama anteni için bir optimizasyon çalışması yapmıştır. Bu çalışmada 6 giriş, 4 gizli ve 1 çıkış katmanından oluşan bir YSA modeli tasarlanmıştır. Toplam örnek sayısı boyutu 144 olarak belirtilmiştir. Bunlardan 124 tanesi eğitim, kalan 20 tanesi ise doğrulama verisi olarak seçilmiştir. Veri seçimi rastgele olarak yapılmıştır. Çalışmada en başarılı sonuçtaki hata değeri 0,457 (MSE) olarak elde edildiği belirtilmiştir (Akdağlı, vd., 2015).

Ve son olarak yapılan bir tez çalışmasında YSA ve Destek Vektör Makinası (DVM) yöntemleri kullanılarak modelleme çalışması yapılmıştır. Aktivasyon fonksiyonu olarak Tansig, Logsig ve Purelin tercih edilmiştir. Çalışmada 4 giriş, 6 gizli ve 1 çıkış katmanından oluşan bir mimari kullanılmıştır. Ayrıca kullanılacak veri setindeki sonuçlarının elde edilmesinde HFSS programından yardım alınmıştır. Çalışmada 210 adet verinin 180 tanesi eğitim, 30 tanesi doğrulama amacıyla kullanılarak bir yama anten modelleme çalışması yapılmıştır. YSA'nın daha hızlı olduğundan bahsedilirken nümerik bir sonuç verilmemiştir. Ayrıca en başarılı sonuçtaki hata değeri 0,271 (MSE) olarak elde edildiği belirtilmiştir (Vuran, 2016).

### 3. Materyal ve Yöntem (Material and Method)

#### 3.1. Matematiksel Model (Mathematical Model)

Yaygın olarak kullanılan mikroşerit yama anten dielektrik zemin üzerine ışıyan metal yama ve altında toprak zeminden yapılmaktadır. Burada metal yama yaygın olarak bakır kullanılmaktadır. Bazı durumlarda bunun yerine iletken malzemelerden olan gümüş veya altında tercih edilmektedir. Yama şekli ise en yaygın olanı kare ve dikdörtgen olmak üzere E harfi U harfi vb. birçok özgün şekillerde görülmektedir (Uluslu, 2021; Uluslu, 2021). Bu çalışmada ise dikdörtgen yama şekli tercih edilmiştir. Yamanın uzunluğu ( $l_p$ ), genişliği ( $w_p$ ), yer düzlem uzunluğu ( $l$ ), yer düzlem genişliği ( $w$ ) ve boy uzunluğu ( $w_p$ ) ile sembolize edilmiştir. Burada dielektrik malzeme kalınlığı 0,05- 5,35mm arasında bir değer tercih edilebilmektedir (Balanis vd., 1982). Ayrıca düşük dielektrikli ve kalın malzeme seçilerek antenin performansında artış sağlanabilir (Safran vd., 2006). Bu çalışmada ise 1,6mm olarak tercih edilmiştir. Dikdörtgen şeklinde mikroşerit yama antenler çoğu zaman Şekil 1'de gösterildiği gibi bir taban üzerinde durmaktadır. Mikroşerit yama anten geometrisini tanımlamak için kullanılan parametreler, tanımları ve kullanılan tasarım değerleri ile birlikte Tablo 1'de verilmiştir.



Şekil 1. Mikroşerit yama antenin fiziksel parametreleri (Physical parameters of microstrip patch antenna)

Tablo 1. Mikroşerit yama antenin tasarım parametreleri (Design parameters of microstrip patch antenna)

Parametre	Değer	Tanım
Uzunluk (mm)	-	x eksenini boyunca yama uzunluğu
Genişlik (mm)	-	y eksenini boyunca yama genişliği
Yükseklik (mm)	1,6	Alt tabakanın yüksekliği
Yüzey	Hava	Dielektrik malzeme türü
Yer Düzlemi Uzunluğu (mm)	-	x eksenini boyunca yer düzlemi uzunluğu
Yer Düzlemi Genişliği (mm)	-	y eksenini boyunca yer düzlemi genişliği
İletken	PEC	Metal malzeme türü

### 3.2. Önerilen Yöntem (Proposed Method)

Bu çalışmada, 2 farklı örnekleme yöntemi önerilmiştir. Bunlar Latin Hiperküp ve Monte Carlo örnekleme yöntemidir. Şimdi bunları sırası ile detaylandıralım.

#### 3.2.1. Latin Hiperküp Örnekleme Yöntemi (Latin Hypercube Sampling Method)

Latin Hiperküp örnekleme yöntemi literatürde belirtildiği gibi tabakalı örnekleme sınıfına aittir (Hossain vd., 2005; Wan vd., 2008). Örnekleri, integral dağılımını etkili bir şekilde yansıtabilir. N'nin değişken türlerinin miktarı olduğunu varsayarsak,  $x_i$ , i değişkenini temsil eder,  $x_l^i$  ve  $x_u^i$ , i'nin minimum ve maksimum değeridir.

Çalıştırma adımları sırası ile aşağıdaki gibidir:

- 1) Numune boyutunu (H) belirleyin;
- 2) Nüfusu katmanlaştırın. Bu örneklemede aralığın uzunluğuna göre bölme stili seçilmiştir. Her  $x_i$  değişkeninin tanım aralığını ( $[x_l^i, x_u^i]$ ) ortalama olarak numune boyutunu (H) parçalarına bölün:

$$x_l^i = x_0^i < x_1^i < x_2^i < \dots < x_{j-1}^i < x_j^i < x_{j+1}^i < \dots < x_H^i = x_u^i \quad (1)$$

- 3) Sütunu  $\{1, 2, \dots, H\}$  dizisinin tümüyle rastgele bir düzenlemesi olan bir  $m \times n$  matrisi üretin;

- 4) Seçilen bir hiperküp bu matrisin her elemanına karşılık gelir, hiperküpde Monte Carlo örneklemesinin yapılacağı gibi rastgele örnek birer birer, numune boyutu (H) örnekleri üretilecektir.

Rastgele sayıların bölünmüş aralığa ait olduğunu garanti etmek için, (j) aralığının rasgele sayısı  $U_j$  aşağıdaki gibi bir denklemi sağlamalıdır:

$$U_j = \frac{U}{H} + \frac{j-1}{H} \quad (2)$$

Bu denklemde  $U$ ,  $[0, 1]$  bölgesinde düzgün dağılmış bir rasgele sayıdır;  $U_j$ ,  $j(j=1, 2, \dots, H)$  bölgesine ait rastgele sayıdır.

### 3.2.2. Monte Carlo Örnekleme Yöntemi (Monte Carlo Sampling Method)

Literatüre göre (Wei vd., 2004; Jin vd., 2005; Zhou vd., 1997) Monte Carlo örnekleme yöntemi, rastgele problemleri rastgele deney yoluyla çözen rastgele simülasyon yöntemi olarak da adlandırılır. Rastgele  $A_i$  olayının olasılığının  $P_i$  ( $i=1, 2, \dots, n$ ) olduğunu varsayarsak. Örnekleme yapabilmek için öncelikle kümülatif olasılığı oluşturun:

$$P_0 = 0, P^{(l)} = \sum_{i=1}^l P_i, (i = 1, 2, \dots, n) \quad (3)$$

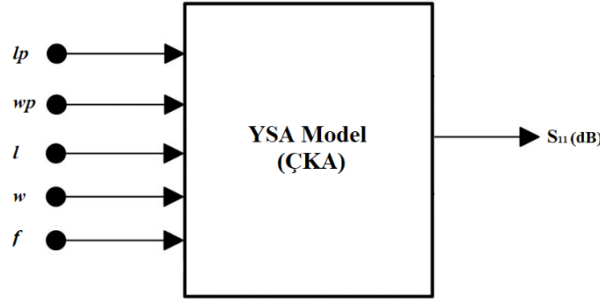
Rastgele bir  $r$  sayısı alın, eğer  $P^{l-1} < P^l \leq P^l$ , ( $l = 1, 2, \dots, n$ ) koşulu sağlanırsa,  $A_i$  olayının gerçekleştiği kabul edilir.

Kesikli rastgele değişkenlerden örnekleme adımları sırası ile aşağıdaki gibidir:

- 1) Kesikli rastgele değişkenleri boyutlarına göre sıralayın, görelî sıralama frekanslarını olasılık olarak alın;
- 2) Rastgele bir sayı alın;
- 3) Rastgele sayının değerini karşılık gelen aralıkla karşılaştırın ve örnek oluşturun.

### 3.3. Çok Katmanlı Algılayıcı ile Modelleme (Multilayer Perceptron Model)

ÇKA başta olmak üzere YSA'lar giriş, çıkış ve gizli katmanlar barındırmaktadırlar. Bu tür ağların yapısı gereği çoklu giriş/çıkış modellemeye imkân sağlamaktadır (Şen, 2004). Mikroşerit yama antenin modellemesi için kullanılacak olan ÇKA iki, üç ve dört gizli katmanlı olacak şekilde 10, 15 ve 20 nörondan oluşan durumlar için tasarlanmıştır. Benzer yapılar yapılan bir çalışmada başarılı sonuçlar vermiştir (Mahouti, 2019). Nöronlar giriş/çıkış katmanları farklı ağırlık katsayılarına sahiptir. Eğitim aşaması, çıkan eğitim hata oranı minimum seviyeye getirilene kadar modelleme çalışmayı sürdürür (Şen, 2004). Bu amaç için katmanlar arasında iki farklı aktivasyon fonksiyonunun kombinasyonu kullanılmıştır. Gizli katmandaki aktivasyon fonksiyonu olarak son nöronda logsis, diğer katmanlar için tansig tercih edilmiştir. Aktivasyon fonksiyonları yine literatürdeki çalışmalara benzer olarak tercih edilmiştir (Mahouti, 2019; Vuran, 2016). Ağ modeli oluşturulurken antenin 4 boyut parametresi yanı sıra frekans parametresi ile birlikte toplamda 5 giriş parametresi vardır. Önerilen ağ modelinin kara kutu modellemesi Şekil 2'de gösterilmiştir.



**Şekil 2.** Mikroşerit yama antenin  $S_{11}$  (dB) parametresini eğitmek için YSA modeli (ANN model for training  $S_{11}$  (dB) parameter of microstrip patch antenna)

## 4. Deneysel Sonuçlar (Experimental Results)

Deneysel çalışma kısmı 4 bölümden oluşmaktadır. Tüm bölümlerde ortak olarak kullanılacak ağ modeli Şekil 2'de verilmiştir. Bu ağ modelinde trainbr, trainlm, trainrp ve trainscg algoritmaları kullanılacaktır. Mimari olarak ise 2, 3 ve 4 gizli katmanlı yapılar kullanılacaktır. 2 gizli katmanlı ağ modeli 15 ve 20 nörona sahiptir. 3 gizli katmanlı ağ modeli ise 10, 15 ve 20 nörona sahiptir. 4 gizli katmanlı ağ modeli ise 10, 15, 15 ve 20 nörona sahiptir. Tüm bu seçilen algoritma ve mimaride kullanılan modeller literatür araştırmasında da belirtilen başarılı bir çalışmadan örnek alınarak tasarlanmıştır (Mahouti, P., 2019). Her bölümde değişik örnekleme boyutuna sahip farklı sayıdaki eğitim ve doğrulama verileri için Latin Hiperküp ve Monte Carlo örnekleme yöntemlerinin performansları nümerik olarak tablolar halinde verilecektir. Literatür taramasında en düşük örnekleme boyutu olarak 144 olarak tespit edilmiştir (Akdağlı, vd., 2015). Ayrıca eğitim data sayısı 124 olmakla birlikte doğru veri sayısı oranı %7,2 ye denk gelmektedir (Akdağlı, vd., 2015). Yapılacak çalışmada ise tüm literatür örneklerinden daha düşük olacak şekilde örnekleme boyutu 120 ve 100 olarak seçilmiştir. Daha sonra oluşturulan 2 farklı örnek sayısına sahip problemler

kendi içlerinde %50 ve %33 oranında rastgele ayrılarak eğitim ve doğrulama verileri oluşturulmuştur. Doğrulama veri sayısı oranı ise %50 ve %33 tüm literatür örneklerinden daha yüksektir. Böylelikle doğrulama hata gerçekliği daha yüksek olacaktır. Eğitim ve doğrulama verilerinin ayrımı verilen literatür örneklerinde de olduğu gibi rastgele olarak yapılmıştır. Ayrıca YSA modelleri rastgele süreçlerden geçerek eğitimlerini tamamladıklarından dolayı her bir modelleme için kod 10 defa çalıştırılacaktır. Tablolarda modelleme performans değerlendirilmesinde sıkça kullanılan ortalama kare hata (MSE), ortalama mutlak hata (MAE) ve korelasyon katsayısı ( $R^2$ ) kullanılacaktır. Ayrıca her bölümde bir algoritma için seçilen bazı sonuçlar grafik olarak gösterilecektir. Ek olarak tüm modelleme işlemleri 1 dakikanın altı bir sürede işlemi tamamlamıştır. En kısa optimizasyon süresi 10 saniye ve en uzun optimizasyon süresi ise 58 saniye sürmüştür. 4 bölümde kullanılacak olan eğitim ve doğrulama veri setleri detayları Tablo 2’de nümerik olarak verilmiştir.

**Tablo 2.** Eğitim ve doğrulama veri setleri (Training and validation datasets)

Parametre	Değer Aralığı	Örnekleme Yöntemi	Örnek Sayısı (Bölüm 4.1)	Örnek Sayısı (Bölüm 4.2)	Örnek Sayısı (Bölüm 4.3)	Örnek Sayısı (Bölüm 4.4)
Uzunluk (mm)	44 – 74	Latin Hiperküp veya Monte Carlo	(Eğitim: 60)	(Eğitim: 80)	(Eğitim: 50)	(Eğitim: 67)
Genişlik (mm)	59 – 98		Doğrulama: 60)	Doğrulama: 40)	Doğrulama: 50)	Doğrulama: 33)
Yer Düzlemi Uzunluğu (mm)	94 – 156	Monte Carlo	Toplam:120	Toplam:120	Toplam:100	Toplam:100
Yer Düzlemi Genişliği (mm)	94 – 156					
Frekans (GHz)	1,5 – 3,5	Lineer	100	100	100	100
Toplam Örnek	-	-	120x100	120x100	100x100	100x100

#### 4.1. Latin Hiperküp ve Monte Carlo Örnekleme Yöntemlerinin 60 Eğitim, 60 Doğrulama Verisi için Performans Karşılaştırması (Performance Comparison of Latin Hypercube and Monte Carlo Sampling Methods for 60 Training, 60 Validation Data)

Bu kısımda 60 eğitim ve 60 doğrulama verisi olmak üzere toplamda 120 örnek data kullanılacaktır. Eğitim ve doğrulama veri seti detayları Tablo 2’de nümerik olarak verilmiştir. Modelleme sonucu elde edilen Latin Hiperküp örnekleme yöntemi için 10 koşu arasından en iyi ve ortalama performans karşılaştırması sırası ile Tablo 3 ve Tablo 4’de verilmiştir. Monte Carlo örnekleme yöntemi için 10 koşu arasından en iyi ve ortalama performans karşılaştırması sırası ile Tablo 5 ve Tablo 6’da verilmiştir.

**Tablo 3.** Latin Hiperküp örnekleme metodu kullanılarak oluşturulan 120 örnek veriden 60 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde en iyi performansların karşılaştırması (Comparison of 60 validation data from 120 sample data generated using the Latin Hypercube sampling method and the best performances out of 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata ( $R^2$ )	Doğrulama Hata ( $R^2$ )
trainbr	15 20	0,001973	1,328474	0,024389	0,153798	0,999525	0,598543
	10 15 20	0,000999	0,800414	0,018296	0,188743	0,999814	0,382493
	10 15 15 20	0,000998	0,246361	0,016918	0,164673	0,999887	0,653075
trainlm	15 20	0,002368	1,018362	0,027032	0,167077	0,996332	0,170747
	10 15 20	0,001132	0,734225	0,018973	0,210885	0,973320	0,011693
	10 15 15 20	0,000999	0,975303	0,017944	0,234827	0,983914	0,042827
trainrp	15 20	0,153190	0,305079	0,178794	0,241484	0,894082	0,798523
	10 15 20	0,084395	0,186416	0,125620	0,172891	0,934052	0,840864
	10 15 15 20	0,044988	0,222449	0,109446	0,184037	0,960946	0,528671
trainscg	15 20	0,056231	0,138097	0,103112	0,149750	0,940850	0,903255
	10 15 20	0,037208	0,185944	0,094866	0,141519	0,981819	0,931293
	10 15 15 20	0,021357	0,145068	0,072242	0,109607	0,985782	0,886969

**Tablo 4.** Latin Hiperküp örnekleme metodu kullanılarak oluşturulan 120 örnek veriden 60 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde ortalama performansların karşılaştırması (Comparison of 60 validation data from 120 sample data generated using the Latin Hypercube sampling method, and average performances among 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,003189	3,389983	0,030139	0,200252	0,999640	0,777088
	10 15 20	0,001205	5,372497	0,019846	0,279593	0,999864	0,718053
	10 15 15 20	0,000999	2,914812	0,017590	0,247307	0,999887	0,828619
trainlm	15 20	0,008538	56,81571	0,047729	0,691376	0,999035	0,494523
	10 15 20	0,025142	412,8871	0,053627	1,248521	0,997124	0,561795
	10 15 15 20	0,028783	749,3137	0,088201	3,924392	0,996732	0,364469
trainrp	15 20	0,506368	0,889192	0,349488	0,483596	0,940766	0,898481
	10 15 20	0,209331	0,758234	0,207459	0,340954	0,975889	0,918561
	10 15 15 20	0,145358	1,450832	0,180509	0,429328	0,983419	0,850833
trainscg	15 20	0,171518	0,398608	0,182402	0,230311	0,980314	0,956131
	10 15 20	0,086767	0,334069	0,134203	0,189602	0,990142	0,964652
	10 15 15 20	0,054943	0,418185	0,112227	0,183701	0,993780	0,955336

**Tablo 5.** Monte Carlo örnekleme metodu kullanılarak oluşturulan 120 örnek veriden 60 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde en iyi performansların karşılaştırması (Comparison of 60 validation data from 120 sample data generated using the Monte Carlo sampling method and the best performances out of 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,001313	0,049534	0,018744	0,078308	0,999753	0,962465
	10 15 20	0,000993	0,082836	0,015981	0,079942	0,999844	0,976793
	10 15 15 20	0,000996	0,086033	0,015675	0,085526	0,999868	0,966295
trainlm	15 20	0,001329	0,045046	0,019081	0,067007	0,999683	0,427380
	10 15 20	0,000998	0,052176	0,015809	0,073219	0,999528	0,371702
	10 15 15 20	0,000993	0,066269	0,016042	0,071179	0,999712	0,337340
trainrp	15 20	0,046070	0,069548	0,101068	0,115779	0,967993	0,963357
	10 15 20	0,020521	0,037888	0,064023	0,079406	0,977722	0,956007
	10 15 15 20	0,014878	0,040839	0,062132	0,083175	0,977871	0,926531
trainscg	15 20	0,016224	0,028126	0,064687	0,072597	0,971210	0,975758
	10 15 20	0,009670	0,024452	0,047320	0,059586	0,997118	0,992834
	10 15 15 20	0,007134	0,022210	0,040346	0,055654	0,998513	0,994423

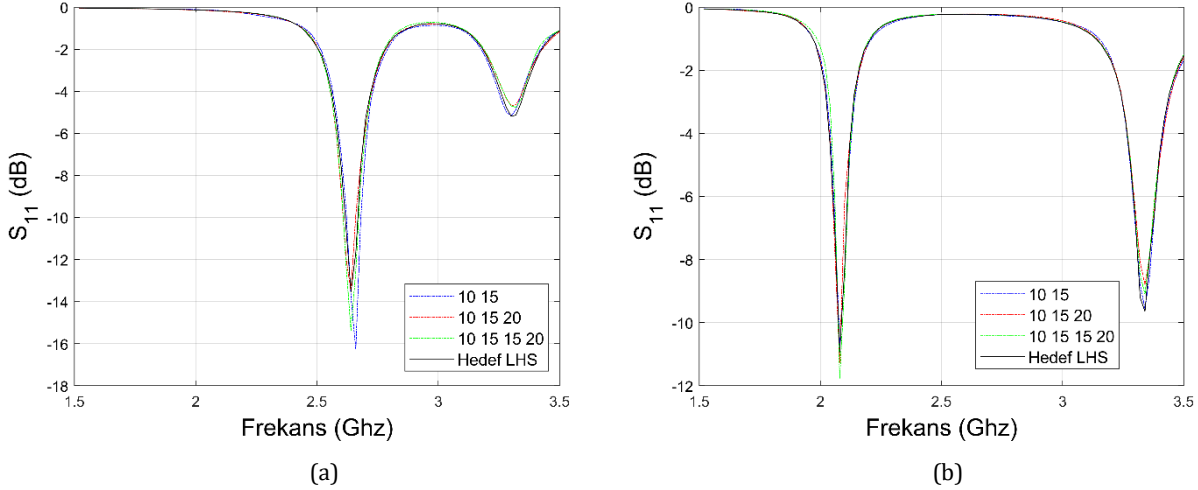
**Tablo 6.** Monte Carlo örnekleme metodu kullanılarak oluşturulan 120 örnek veriden 60 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde ortalama performansların karşılaştırması (Comparison of 60 validation data from 120 sample data generated using the Monte Carlo sampling method and average performances among 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,001599	0,124235	0,020684	0,095263	0,999788	0,984703
	10 15 20	0,001017	0,136837	0,016893	0,099227	0,999866	0,982683
	10 15 15 20	0,000999	0,138683	0,016523	0,094267	0,999868	0,982522
trainlm	15 20	0,001724	3,618635	0,021042	0,208700	0,999772	0,806605
	10 15 20	0,001256	8,384723	0,018797	0,248744	0,999834	0,877114
	10 15 15 20	0,001182	62,23466	0,020253	0,795375	0,999843	0,747122
trainrp	15 20	0,100741	0,149667	0,147481	0,177807	0,986586	0,980641
	10 15 20	0,050355	0,130917	0,102756	0,132432	0,993302	0,983419
	10 15 15 20	0,057007	0,241713	0,106461	0,168643	0,992403	0,968840
trainscg	15 20	0,047986	0,069377	0,092270	0,103391	0,993599	0,991075
	10 15 20	0,014805	0,031459	0,060296	0,071632	0,998039	0,995963
	10 15 15 20	0,009678	0,031212	0,049057	0,065171	0,998718	0,995992

Verilen Tablo 3, Tablo 4, Tablo 5 ve Tablo 6'dan yola çıkarak eğitim hata sonuçlarını her örnekleme yönteminin kendi içinde karşılaştıracak olursak, MSE ve MAE olarak trainbr algoritması her iki örnekleme yönteminde en başarılı sonuçların bulunduğu görülmektedir. Yine tablolardan örnekleme yöntemlerini performans olarak

birbirleri ile karşılaştırırsak MSE ve MAE olarak ise Monte Carlo örnekleme yönteminin daha başarılı sonuç verdiği görülmektedir. Tablolar üzerinden doğrulama hata sonuçlarını her örnekleme yönteminin kendi içinde karşılaştıracak olursak, MSE olarak trainscg algoritması için 2 gizli katmana sahip mimaride Latin Hiperküp örnekleme yöntemi en başarılı sonuca sahipken, trainscg algoritması için 4 gizli katmana sahip mimaride monte carlo örnekleme en başarılı sonuca sahiptir. MAE olarak ise trainbr algoritması için 4 gizli katmana sahip mimaride Latin Hiperküp örnekleme yöntemi en başarılı sonuca sahipken, trainscg algoritması için 4 gizli katmana sahip mimaride Monte Carlo örnekleme en başarılı sonuca sahiptir. Örnekleme yöntemlerini performans olarak birbirleri ile karşılaştırırsak MSE ve MAE olarak Monte Carlo örnekleme yönteminin daha başarılı sonuç verdiği görülmektedir.

Ayrıca Latin Hiperküp örnekleme yöntemiyle elde edilmiş trainbr algoritmalı 2, 3 ve 4 gizli katmanlı sonuçlardan seçilen 2 tanesi için  $S_{11}$  (dB) parametresi grafik olarak Şekil 3A ve Şekil 3B'de gösterilmiştir. Grafiklerdeki hedef ve modellenen çizgilerin örtüşmesi sonucun başarılı olduğunu grafiksel olarak ispatlamaktadır.



**Şekil 3.** Latin Hiperküp örnekleme metodu kullanılarak oluşturulan 120 örnek veriden 60 doğrulama datası içerisinde seçilen 2, 3 ve 4 gizli katmanlı “trainbr” algoritmalı ÇKA modellemesi sonucu oluşan  $S_{11}$  (dB) simülasyonu (a) lp=54, wp=88, l=112, w=109 (mm) (b) lp=69, wp=87, l=129, w=131 (mm) ( $S_{11}$  (dB) simulation, which is the result of MLP modeling with 2, 3 and 4 hidden layers “trainbr” algorithm, selected from 60 validation data out of 120 sample data created using the Latin Hypercube sampling method (a) lp=54, wp=88, l=112, w=109 (mm) (b) lp=69, wp=87, l=129, w=131 (mm))

#### 4.2. Latin Hiperküp ve Monte Carlo Örnekleme Yöntemlerinin 80 Eğitim 40 Doğrulama Verisi için Performans Karşılaştırması (Performance Comparison of Latin Hypercube and Monte Carlo Sampling Methods for 80 Training 40 Validation Data)

Bu kısımda 80 eğitim ve 40 doğrulama datası olmak üzere toplamda 120 örnek data kullanılacaktır. Eğitim ve doğrulama veri seti detayları Tablo 2’de nümerik olarak verilmiştir. Modelleme sonucu elde edilen Latin Hiperküp örnekleme yöntemi için 10 koşu arasından en iyi ve ortalama performans karşılaştırması sırası ile Tablo 7 ve Tablo 8’de verilmiştir. Monte Carlo örnekleme yöntemi için 10 koşu arasından en iyi ve ortalama performans karşılaştırması sırası ile Tablo 9 ve Tablo 10’da verilmiştir.



**Tablo 7.** Latin Hiperküp örnekleme metodu kullanılarak oluşturulan 120 örnek veriden 40 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde en iyi performansların karşılaştırması (Comparison of 40 validation data from 120 sample data generated using the Latin Hypercube sampling method and the best performances out of 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,003933	0,385420	0,030196	0,119281	0,999370	0,015992
	10 15 20	0,001428	0,685734	0,021152	0,136972	0,999740	0,583285
	10 15 15 20	0,000998	0,507913	0,016822	0,139642	0,999857	0,673131
trainlm	15 20	0,003485	0,647510	0,029434	0,129251	0,999199	0,124453
	10 15 20	0,001880	0,992843	0,023302	0,132946	0,937151	0,220892
	10 15 15 20	0,000993	0,394939	0,017729	0,124124	0,979294	0,224632
trainrp	15 20	0,138789	0,147121	0,161003	0,198407	0,934455	0,868007
	10 15 20	0,060710	0,144877	0,116404	0,152284	0,909801	0,851061
	10 15 15 20	0,045614	0,220835	0,107138	0,159026	0,927236	0,694099
trainscg	15 20	0,072538	0,153040	0,120014	0,165059	0,959037	0,928693
	10 15 20	0,037040	0,128874	0,089257	0,114068	0,987568	0,761788
	10 15 15 20	0,022477	0,140121	0,073786	0,110155	0,992649	0,904612

**Tablo 8.** Latin Hiperküp örnekleme metodu kullanılarak oluşturulan 120 örnek veriden 40 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde ortalama performansların karşılaştırması (Comparison of 40 validation data from 120 sample data generated using the Latin Hypercube sampling method, and average performances among 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,004585	13,75082	0,033413	0,228361	0,999489	0,726046
	10 15 20	0,001891	2,644887	0,023379	0,196905	0,999790	0,809997
	10 15 15 20	0,001042	2,554567	0,017548	0,208413	0,999884	0,823942
trainlm	15 20	0,005145	94,61056	0,035269	0,663166	0,999427	0,484484
	10 15 20	0,057190	75,57028	0,070630	0,701197	0,993436	0,552098
	10 15 15 20	0,031620	121,5421	0,083918	1,504551	0,996455	0,516331
trainrp	15 20	0,300752	0,424269	0,249937	0,290997	0,965973	0,952716
	10 15 20	0,169526	0,451246	0,176775	0,238819	0,980672	0,950263
	10 15 15 20	0,188475	0,895378	0,191922	0,366386	0,978503	0,901148
trainscg	15 20	0,153375	0,308190	0,174839	0,207016	0,982760	0,966075
	10 15 20	0,057242	0,715813	0,110606	0,165539	0,993612	0,923310
	10 15 15 20	0,038357	0,371930	0,092861	0,132661	0,995722	0,957607

**Tablo 9.** Monte Carlo örnekleme metodu kullanılarak oluşturulan 120 örnek veriden 40 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde en iyi performansların karşılaştırması (Comparison of 40 validation data from 120 sample data generated using the Monte Carlo sampling method and the best performances out of 10 runs for MLP modeling)

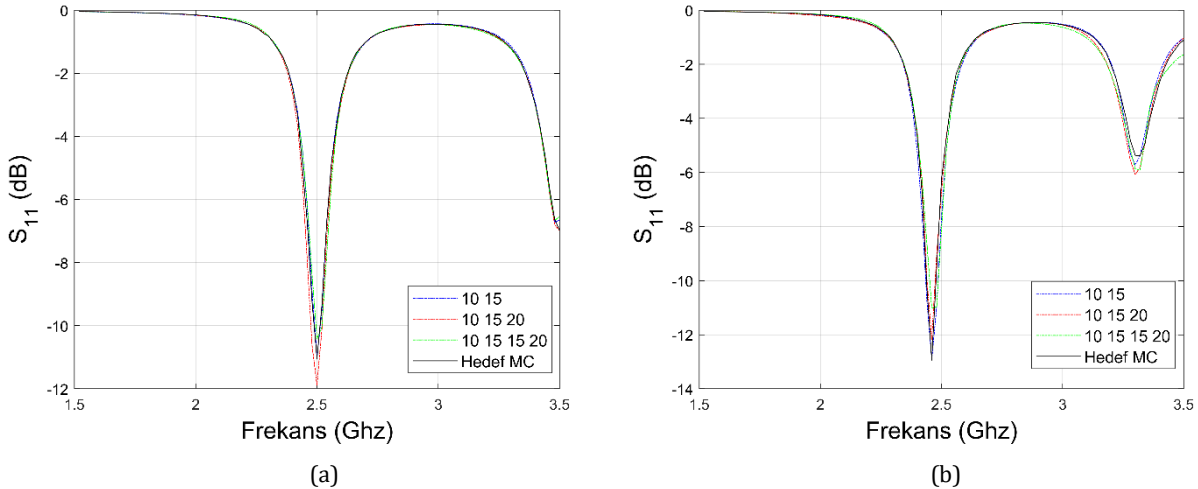
Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,001898	0,025394	0,021124	0,060418	0,999623	0,990613
	10 15 20	0,001069	0,026984	0,017165	0,061214	0,999789	0,964204
	10 15 15 20	0,000998	0,063060	0,016280	0,081095	0,999865	0,939123
trainlm	15 20	0,002026	0,033453	0,021327	0,066231	0,999551	0,151974
	10 15 20	0,001182	0,053091	0,017726	0,074440	0,999791	0,924177
	10 15 15 20	0,000988	0,047050	0,016284	0,070968	0,999872	0,691523
trainrp	15 20	0,043964	0,038318	0,106944	0,115254	0,974435	0,960766
	10 15 20	0,019773	0,024613	0,065131	0,077623	0,980629	0,975262
	10 15 15 20	0,012991	0,019287	0,054340	0,065211	0,996063	0,974934
trainscg	15 20	0,018469	0,021077	0,063946	0,071418	0,995968	0,995406
	10 15 20	0,012208	0,015730	0,052254	0,055649	0,995859	0,995302
	10 15 15 20	0,006967	0,011902	0,037465	0,045570	0,994910	0,991838

**Tablo 10.** Monte Carlo örnekleme metodu kullanılarak oluşturulan 120 örnek veriden 40 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde ortalama performansların karşılaştırması (Comparison of 40 validation data from 120 sample data generated using the Monte Carlo sampling method and average performances among 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,002448	0,046752	0,023514	0,072579	0,999686	0,993892
	10 15 20	0,001321	0,115126	0,018879	0,088402	0,999830	0,985119
	10 15 15 20	0,001005	0,184657	0,016612	0,100020	0,999871	0,979430
trainlm	15 20	0,002491	111,9872	0,023935	0,400002	0,999680	0,882946
	10 15 20	0,001367	0,293139	0,018779	0,119649	0,999824	0,965502
	10 15 15 20	0,000997	1,256360	0,016946	0,144696	0,999872	0,944520
trainrp	15 20	0,102595	0,104201	0,153403	0,168428	0,986808	0,985924
	10 15 20	0,046683	0,060383	0,098521	0,109308	0,993980	0,991939
	10 15 15 20	0,021084	0,054416	0,071944	0,092636	0,997290	0,992758
trainscg	15 20	0,023872	0,026329	0,076088	0,081863	0,996930	0,996474
	10 15 20	0,017050	0,021851	0,064604	0,070574	0,997808	0,997085
	10 15 15 20	0,013648	0,021609	0,054652	0,063394	0,998246	0,997141

Verilen Tablo 7, Tablo 8, Tablo 9 ve Tablo 10'dan yola çıkarak eğitim hata sonuçlarını her örnekleme yönteminin kendi içinde karşılaştıracak olursak, MSE ve MAE olarak trainbr algoritması için 4 gizli katmana sahip mimaride Monte Carlo örnekleme yönteminde en başarılı sonuçların bulunduğu görülmektedir. Latin Hiperküp örnekleme yönteminde ise MSE olarak trainlm algoritmasında, MAE olarak ise trainbr algoritmasının daha başarılı sonuçlara sahip olduğu görülmektedir. Yine tablolardan örnekleme yöntemlerini performans olarak birbirleri ile karşılaştırsak MSE olarak Latin Hiperküp örnekleme yönteminin, MAE olarak ise Monte Carlo örnekleme yönteminin daha başarılı sonuç verdiği görülmektedir. Tablolar üzerinden doğrulama hata sonuçlarını her örnekleme yönteminin kendi içinde karşılaştıracak olursak, MSE olarak trainscg algoritması için 3 gizli katmana sahip mimaride Latin Hiperküp örnekleme yöntemi en başarılı sonuca sahipken, trainscg algoritması için 4 gizli katmana sahip mimaride monte carlo örnekleme en başarılı sonuca sahiptir. MAE olarak ise trainscg algoritması için 4 gizli katmana sahip mimaride Latin Hiperküp ve Monte Carlo örnekleme en başarılı sonuca sahiptir. Örnekleme yöntemlerini performans olarak birbirleri ile karşılaştırsak MSE ve MAE olarak Monte Carlo örnekleme yönteminin daha başarılı sonuç verdiği görülmektedir.

Ayrıca Monte Carlo örnekleme yöntemiyle elde edilmiş trainlm algoritmalı 2, 3 ve 4 gizli katmanlı sonuçlardan seçilen 2 tanesi için  $S_{11}$  (dB) parametresi grafik olarak Şekil 4A ve Şekil 4B'de gösterilmiştir. Grafiklerdeki hedef ve modellenen çizgilerin benzerliği bu kısımda da gözükmemektedir.



**Şekil 4.** Monte Carlo örnekleme metodu kullanılarak oluşturulan 120 örnek veriden 40 doğrulama datası içerisinde seçilen 2, 3 ve 4 gizli katmanlı "trainlm" algoritmalı ÇKA modellemesi sonucu oluşan  $S_{11}$  (dB) simülasyonu (a) lp=57, wp=83, l=119, w=110 (mm) (b) lp=58, wp=88, l=121, w=98 (mm) ( $S_{11}$  (dB) simülasyonu, which is the result of MLP modeling with 2, 3 and 4 hidden layers "trainlm" algorithm, selected from 40 validation data out of 120 sample data created using Monte Carlo sampling method (a) lp=57, wp=83, l=119, w=110 (mm) (b) lp=58, wp=88, l=121, w=98 (mm))

#### 4.3. Latin Hiperküp ve Monte Carlo Örnekleme Yöntemlerinin 50 Eğitim 50 Doğrulama Verisi için Performans Karşılaştırması (Performance Comparison of Latin Hypercube and Monte Carlo Sampling Methods for 50 Training 50 Validation Data)

Bu kısımda 50 eğitim ve 50 doğrulama datası olmak üzere toplamda 100 örnek data kullanılacaktır. Eğitim ve doğrulama veri seti detayları Tablo 2’de nümerik olarak verilmiştir. Modelleme sonucu elde edilen Latin Hiperküp örnekleme yöntemi için 10 koşu arasından en iyi ve ortalama performans karşılaştırması sırası ile Tablo 11 ve Tablo 12’de, Monte Carlo örnekleme yöntemi için 10 koşu arasından en iyi ve ortalama performans karşılaştırması sırası ile Tablo 13 ve Tablo 14’de verilmiştir.

**Tablo 11.** Latin Hiperküp örnekleme metodu kullanılarak oluşturulan 100 örnek veriden 50 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde en iyi performansların karşılaştırması (Comparison of 50 validation data out of 100 sample data generated using the Latin Hypercube sampling method and the best performances out of 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,001585	2,333636	0,022464	0,349488	0,999672	0,231354
	10 15 20	0,000995	1,706196	0,017123	0,208349	0,999830	0,533939
	10 15 15 20	0,000995	1,557427	0,016606	0,253581	0,999872	0,548605
trainlm	15 20	0,001309	7,068730	0,020976	0,328642	0,997540	0,197531
	10 15 20	0,000996	0,896049	0,017822	0,266908	0,984509	0,177188
	10 15 15 20	0,000997	3,766801	0,019065	0,643363	0,816500	0,026961
trainrp	15 20	0,172894	0,557929	0,182967	0,307520	0,878103	0,673983
	10 15 20	0,061052	0,722574	0,114141	0,242127	0,928834	0,501273
	10 15 15 20	0,087372	0,878676	0,140899	0,395197	0,874162	0,305786
trainscg	15 20	0,089749	0,325379	0,142042	0,219289	0,958302	0,905458
	10 15 20	0,019838	0,227184	0,072324	0,137911	0,984989	0,927619
	10 15 15 20	0,019329	0,314340	0,069508	0,168582	0,995866	0,668651

**Tablo 12.** Latin Hiperküp örnekleme metodu kullanılarak oluşturulan 100 örnek veriden 50 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde ortalama performansların karşılaştırması (Comparison of 50 validation data out of 100 sample data generated using the Latin Hypercube sampling method and average performances within 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,002045	16,18066	0,025745	0,486400	0,999738	0,620771
	10 15 20	0,001030	6,287330	0,018091	0,383860	0,999868	0,737979
	10 15 15 20	0,000999	9,208221	0,017442	0,420673	0,999872	0,722814
trainlm	15 20	0,004260	142,2027	0,034630	1,011339	0,999453	0,399249
	10 15 20	0,017270	742,0637	0,061437	3,249868	0,997772	0,476083
	10 15 15 20	0,134075	1538,540	0,099892	6,129165	0,981118	0,300158
trainrp	15 20	0,446364	1,724528	0,339359	0,721611	0,940749	0,828826
	10 15 20	0,232362	5,715165	0,222861	0,634925	0,969580	0,791798
	10 15 15 20	0,301587	25,65133	0,268356	0,976683	0,960062	0,623100
trainscg	15 20	0,160376	0,688733	0,186898	0,294752	0,979160	0,935194
	10 15 20	0,049122	0,487672	0,104394	0,194591	0,993675	0,956274
	10 15 15 20	0,025658	1,233003	0,083262	0,241400	0,996704	0,893627

**Tablo 13.** Monte Carlo örnekleme metodu kullanılarak oluşturulan 100 örnek veriden 50 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde en iyi performansların karşılaştırması (Comparison of 50 validation data from 100 sample data generated using the Monte Carlo sampling method and the best performances out of 10 runs for MLP modeling)

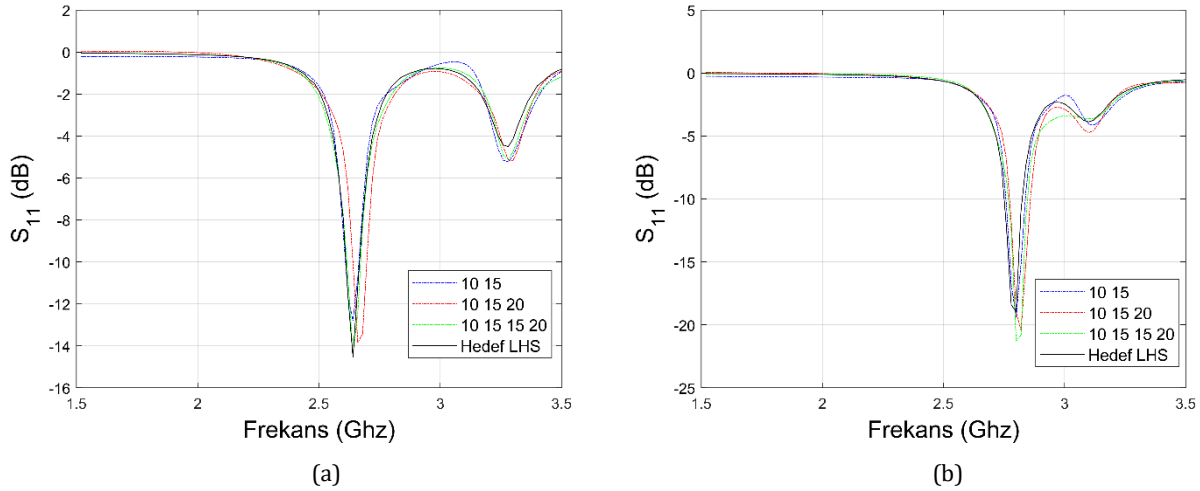
Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,001000	0,061366	0,016683	0,089762	0,999837	0,958224
	10 15 20	0,000991	0,062696	0,015678	0,079976	0,999867	0,969483
	10 15 15 20	0,000994	0,044752	0,015575	0,074735	0,999872	0,980504
trainlm	15 20	0,000998	0,029041	0,016895	0,064791	0,992677	0,127467
	10 15 20	0,000994	0,058094	0,015964	0,077985	0,280952	0,064943
	10 15 15 20	0,000979	0,090526	0,015929	0,104591	0,999848	0,170217
trainrp	15 20	0,049709	0,057373	0,103332	0,129045	0,952535	0,725290
	10 15 20	0,014054	0,027980	0,063896	0,080441	0,975563	0,860686
	10 15 15 20	0,011397	0,025918	0,051620	0,073160	0,981948	0,814580
trainscg	15 20	0,016816	0,025738	0,065055	0,082955	0,995021	0,992125
	10 15 20	0,008972	0,019613	0,046257	0,064350	0,997112	0,993617
	10 15 15 20	0,005937	0,014702	0,039023	0,056149	0,995639	0,985060

**Tablo 14.** Monte Carlo örnekleme metodu kullanılarak oluşturulan 100 örnek veriden 50 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde ortalama performansların karşılaştırması (Comparison of 50 validation data out of 100 sample data generated using Monte Carlo sampling method and average performances within 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,001131	0,147042	0,017825	0,131049	0,999855	0,981920
	10 15 20	0,001002	0,106380	0,016375	0,097173	0,999872	0,986572
	10 15 15 20	0,000997	0,088398	0,016180	0,096135	0,999872	0,988817
trainlm	15 20	0,006694	24,93457	0,031253	0,518730	0,999141	0,682751
	10 15 20	0,360822	0,745635	0,132281	0,259787	0,927980	0,867912
	10 15 15 20	0,001015	166,3106	0,018902	1,542591	0,999870	0,671264
trainrp	15 20	0,180577	0,833575	0,200131	0,412123	0,976583	0,898809
	10 15 20	0,071810	0,306512	0,125501	0,221286	0,990749	0,959279
	10 15 15 20	0,046469	0,559031	0,098857	0,252120	0,994027	0,932137
trainscg	15 20	0,023641	0,038412	0,075898	0,095058	0,996971	0,994933
	10 15 20	0,014456	0,030737	0,060377	0,078102	0,998149	0,995941
	10 15 15 20	0,012841	0,044033	0,057041	0,083581	0,998355	0,994206

Verilen Tablo 11, Tablo 12, Tablo 13 ve Tablo 14'den yola çıkarak eğitim hata sonuçlarını her örnekleme yönteminin kendi içinde karşılaştıracak olursak, MSE ve MAE olarak trainbr algoritması için Latin Hiperküp örnekleme yönteminde en başarılı sonuçların bulunduğu görülmektedir. Monte Carlo örnekleme yönteminde ise MSE olarak trainlm algoritmasında, MAE olarak ise trainbr algoritmasının daha başarılı sonuçlara sahip olduğu görülmektedir. Yine tablolardan örnekleme yöntemlerini performans olarak birbirleri ile karşılaştırsak MSE olarak aynı MAE olarak ise Monte Carlo örnekleme yönteminin daha başarılı sonuç verdiği görülmektedir. Tablolar üzerinden doğrulama hata sonuçlarını her örnekleme yönteminin kendi içinde karşılaştıracak olursak, MSE olarak trainscg algoritması için 3 gizli katmana sahip mimaride Latin Hiperküp örnekleme yöntemi en başarılı sonuca sahipken, trainscg algoritması için 4 gizli katmana sahip mimaride monte carlo örnekleme en başarılı sonuca sahiptir. MAE olarak ise trainscg algoritması her iki örnekleme yönteminde en başarılı sonuçların bulunduğu görülmektedir. Örnekleme yöntemlerini performans olarak birbirleri ile karşılaştırsak MSE ve MAE olarak Monte Carlo örnekleme yönteminin daha başarılı sonuç verdiği görülmektedir.

Ayrıca Latin Hiperküp örnekleme yöntemiyle elde edilmiş trainscg algoritmalı 2, 3 ve 4 gizli katmanlı sonuçlardan seçilen 2 tanesi için S<sub>11</sub> (dB) parametresi grafik olarak Şekil 5A ve Şekil 5B'de gösterilmiştir. Bu kısımdaki grafiklerde hedef ve modellenen çizgilerin örtüşmesi önceki bölümlere göre daha azdır.



**Şekil 5.** Latin Hiperküp örnekleme metodu kullanılarak oluşturulan 100 örnek veriden 50 doğrulama datası içerisinde seçilen 2, 3 ve 4 gizli katmanlı “trainscg” algoritmali ÇKA modellemesi sonucu oluşan  $S_{11}$  (dB) simülasyonu (a)  $l_p=54$ ,  $w_p=89$ ,  $l=137$ ,  $w=103$  (mm) (b)  $l_p=51$ ,  $w_p=94$ ,  $l=112$ ,  $w=107$  (mm) ( $S_{11}$  (dB) simulation, which is the result of MLP modeling with 2, 3 and 4 hidden layers “trainscg” algorithm selected from 50 validation data out of 100 sample data created using the Latin Hypercube sampling method (a)  $l_p=54$ ,  $w_p=89$ ,  $l=137$ ,  $w=103$  (mm) (b)  $l_p=51$ ,  $w_p=94$ ,  $l=112$ ,  $w=107$  (mm))

#### 4.4. Latin Hiperküp ve Monte Carlo Örnekleme Yöntemlerinin 67 Eğitim 33 Doğrulama Verisi için Performans Karşılaştırması (Performance Comparison of Latin Hypercube and Monte Carlo Sampling Methods for 67 Training 33 Validation Data)

Bu kısımda 63 eğitim ve 33 doğrulama datası olmak üzere toplamda 100 örnek data kullanılacaktır. Eğitim ve doğrulama veri seti detayları Tablo 2’de nümerik olarak verilmiştir. Modelleme sonucu elde edilen Latin Hiperküp örnekleme yöntemi için 10 koşu arasından en iyi ve ortalama performans karşılaştırması sırası ile Tablo 15 ve Tablo 16’da, Monte Carlo örnekleme yöntemi için 10 koşu arasından en iyi ve ortalama performans karşılaştırması sırası ile Tablo 17 ve Tablo 18’de verilmiştir.

**Tablo 15.** Latin Hiperküp örnekleme metodu kullanılarak oluşturulan 100 örnek veriden 33 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde en iyi performansların karşılaştırması (Comparison of 33 validation data out of 100 sample data generated using the Latin Hypercube sampling method and the best performances out of 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,002456	1,006949	0,026458	0,190577	0,999551	0,233098
	10 15 20	0,000999	0,863909	0,018016	0,171344	0,999752	0,472788
	10 15 15 20	0,000998	0,744676	0,016850	0,162948	0,999882	0,304711
trainlm	15 20	0,002638	2,576331	0,027338	0,195523	0,792954	0,202279
	10 15 20	0,001227	1,384747	0,019574	0,196208	0,982330	0,051905
	10 15 15 20	0,000998	1,098765	0,017763	0,207815	0,998606	0,146731
trainrp	15 20	0,129964	0,203759	0,158063	0,221754	0,940247	0,926071
	10 15 20	0,059178	0,123458	0,104575	0,131629	0,943105	0,717244
	10 15 15 20	0,023770	0,136946	0,077702	0,154237	0,957999	0,597428
trainscg	15 20	0,082545	0,177199	0,128622	0,171219	0,977717	0,915898
	10 15 20	0,031319	0,128665	0,084448	0,130037	0,988102	0,908936
	10 15 15 20	0,021335	0,197761	0,072470	0,124540	0,992639	0,867694

**Tablo 16.** Latin Hiperküp örnekleme metodu kullanılarak oluşturulan 100 örnek veriden 33 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde ortalama performansların karşılaştırması (Comparison of 33 validation data out of 100 sample data generated using the Latin Hypercube sampling method and average performances among 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,003152	9,295719	0,029142	0,286493	0,999626	0,578046
	10 15 20	0,001408	8,261951	0,020604	0,299841	0,999833	0,711752
	10 15 15 20	0,000999	3,504052	0,017474	0,257282	0,999882	0,797369
trainlm	15 20	0,159885	52,51200	0,101125	0,619414	0,978909	0,500023
	10 15 20	0,016328	113,1161	0,043864	1,163718	0,998049	0,607944
	10 15 15 20	0,002884	1942,713	0,030793	3,583872	0,999658	0,574024
trainrp	15 20	0,236083	0,364087	0,220032	0,285764	0,971713	0,959522
	10 15 20	0,154508	0,692181	0,169545	0,270353	0,981425	0,927013
	10 15 15 20	0,140619	1,155512	0,169647	0,385215	0,983103	0,874696
trainscg	15 20	0,123413	0,320150	0,156441	0,206112	0,985271	0,965468
	10 15 20	0,063031	0,352947	0,114308	0,167413	0,992505	0,961216
	10 15 15 20	0,037446	0,445158	0,093472	0,154624	0,995555	0,951359

**Tablo 17.** Monte Carlo örnekleme metodu kullanılarak oluşturulan 100 örnek veriden 33 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde en iyi performansların karşılaştırması (Comparison of 33 validation data out of 100 sample data generated using the Monte Carlo sampling method and the best performances out of 10 runs for MLP modeling)

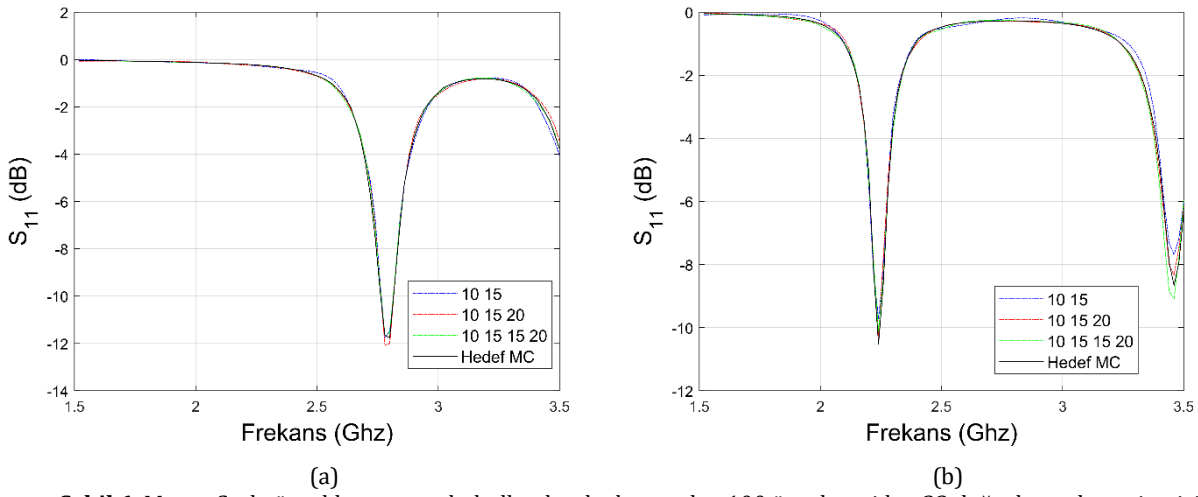
Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,001637	0,037076	0,020367	0,070632	0,999744	0,856150
	10 15 20	0,000999	0,041839	0,016082	0,065080	0,999825	0,967785
	10 15 15 20	0,000997	0,044214	0,016217	0,072575	0,999872	0,976190
trainlm	15 20	0,001382	0,067309	0,019376	0,086141	0,999675	0,938985
	10 15 20	0,000999	0,084011	0,015882	0,085029	0,999654	0,849073
	10 15 15 20	0,000991	0,099057	0,016134	0,096517	0,993428	0,181333
trainrp	15 20	0,044324	0,041718	0,101561	0,115132	0,952361	0,928530
	10 15 20	0,017365	0,024819	0,064081	0,079914	0,996160	0,993482
	10 15 15 20	0,011500	0,022744	0,055728	0,073306	0,996430	0,988059
trainscg	15 20	0,012346	0,021273	0,056525	0,071212	0,990393	0,987929
	10 15 20	0,009564	0,019689	0,046992	0,060763	0,996922	0,994757
	10 15 15 20	0,007089	0,014139	0,041558	0,051128	0,998653	0,995267

**Tablo 18.** Monte Carlo örnekleme metodu kullanılarak oluşturulan 100 örnek veriden 33 doğrulama datası ile ÇKA modellemesi için 10 koşu içerisinde ortalama performansların karşılaştırması (Comparison of 33 validation data out of 100 sample data generated using Monte Carlo sampling method and average performances among 10 runs for MLP modeling)

Algoritma	Mimari	Eğitim Hata (MSE)	Doğrulama Hata (MSE)	Eğitim Hata (MAE)	Doğrulama Hata (MAE)	Eğitim Hata (R <sup>2</sup> )	Doğrulama Hata (R <sup>2</sup> )
trainbr	15 20	0,001803	0,235575	0,021522	0,092728	0,999770	0,978725
	10 15 20	0,001111	0,125718	0,017228	0,104786	0,999858	0,984450
	10 15 15 20	0,000999	0,134136	0,016691	0,103023	0,999873	0,983951
trainlm	15 20	0,001844	0,222549	0,021850	0,109668	0,999765	0,972786
	10 15 20	0,001299	0,398032	0,018834	0,154045	0,999834	0,953409
	10 15 15 20	0,006668	334,1354	0,036147	2,103498	0,999147	0,670303
trainrp	15 20	0,134018	0,164554	0,160512	0,190653	0,982679	0,977327
	10 15 20	0,022331	0,033846	0,073418	0,086883	0,997148	0,995471
	10 15 15 20	0,017854	0,049051	0,067786	0,093423	0,997720	0,993500
trainscg	15 20	0,028605	0,036123	0,082779	0,093406	0,996342	0,995131
	10 15 20	0,013348	0,026543	0,057253	0,072128	0,998295	0,996467
	10 15 15 20	0,008866	0,021838	0,046537	0,062273	0,998868	0,997069

Verilen Tablo 15, Tablo 16, Tablo 17 ve Tablo 18'den yola çıkarak eğitim hata sonuçlarını her örnekleme yönteminin kendi içinde karşılaştıracak olursak, MSE ve MAE olarak trainbr algoritması için Latin Hiperküp örnekleme yönteminde en başarılı sonuçların bulunduğu görülmektedir. Monte Carlo örnekleme yönteminde ise MSE ve MAE olarak ise trainlm algoritmasının daha başarılı sonuçlara sahip olduğu görülmektedir. Yine tablolardan örnekleme yöntemlerini performans olarak birbirleri ile karşılaştırsak MSE olarak aynı MAE olarak ise Monte Carlo örnekleme yönteminin daha başarılı sonuç verdiği görülmektedir. Tablolar üzerinden doğrulama hata sonuçlarını her örnekleme yöntemini kendi içinde karşılaştıracak olursak, MSE olarak trainrp algoritması için 3 gizli katmana sahip mimaride Latin Hiperküp örnekleme yöntemi en başarılı sonuca sahipken, trainscg algoritması için 4 gizli katmana sahip mimaride Monte Carlo örnekleme en başarılı sonuca sahiptir. MAE olarak ise trainscg algoritması için 4 gizli katmana sahip mimaride Latin Hiperküp ve Monte Carlo örnekleme en başarılı sonuca sahiptir. Örnekleme yöntemlerini performans olarak birbirleri ile karşılaştırsak MSE ve MAE olarak Monte Carlo örnekleme yönteminin daha başarılı sonuç verdiği görülmektedir.

Ayrıca Monte Carlo örnekleme yöntemiyle elde edilmiş trainrp algoritmalı 2, 3 ve 4 gizli katmanlı sonuçlardan seçilen 2 tanesi için  $S_{11}$  (dB) parametresi grafik olarak Şekil 6A ve Şekil 6B'de gösterilmiştir. Verilen grafiklerdeki hedef ve modellenen çizgilerin örtüşmesi bir önceki bölüme göre daha iyidir.



**Şekil 6.** Monte Carlo örnekleme metodu kullanılarak oluşturulan 100 örnek veriden 33 doğrulama datası içerisinde seçilen 2, 3 ve 4 gizli katmanlı "trainrp" algoritmalı ÇKA modellemesi sonucu oluşan  $S_{11}$  (dB) simülasyonu (a)  $l_p=51$ ,  $w_p=81$ ,  $l=134$ ,  $w=132$  (mm) (b)  $l_p=64$ ,  $w_p=84$ ,  $l=121$ ,  $w=115$  (mm) ( $S_{11}$  (dB) simulation, which is the result of MLP modeling with 2, 3 and 4 hidden layers "trainrp" algorithm selected from 33 validation data out of 100 sample data created using Monte Carlo sampling method (a)  $l_p=51$ ,  $w_p=81$ ,  $l=134$ ,  $w=132$  (mm) (b)  $l_p=64$ ,  $w_p=84$ ,  $l=121$ ,  $w=115$  (mm))

## 5. Sonuç ve Tartışma (Result and Discussion)

Bu çalışmada, YSA modellemesi için farklı örnekleme metotlarının performansları anten optimizasyon tasarım problemlerinde sıkça kullanılan model olan ÇKA kullanılarak mikroşerit yama anten üzerinde performans karşılaştırılması yapılmıştır. Seçilen model toplamda 6 parametreye sahiptir. Bunlardan 5 tanesi giriş, 1 tanesi ise çıkış parametresidir. Parametre sayısının yüksek olması problemin zorluğunu arttırmıştır. Toplamda 12 farklı ağ kullanılmıştır. Bunlar 3 algoritma ve 4 mimariden oluşmaktadır. Kullanılan yapılar literatür taramasında karşılaşılan başarılı bir çalışmaya dayandırılarak seçilmiştir (Mahouti, 2019). Seçilen en başarılı örnekleme metodunun başarısı çok sayıda farklı ağ kullanılarak da test edilmiştir. Literatür taramasında karşılaşılan en düşük örnek sayısı 144 olarak tespit edilmiştir (Akdağlı, vd., 2015). Ayrıca tüm örnek sayısı içerisinde doğrulama veri sayısı oranı %7,2'dir (Akdağlı, vd., 2015). Yapılan çalışmada ise bahsedilen literatür örneklerinden daha düşük bir sayıda örnekleme boyutu olarak 120 ve 100 seçilmiştir. Doğrulama veri sayısı oranı ise yine bahsedilen literatür örneklerinden daha yüksek olacak şekilde %50 ve %33 olarak tercih edilmiştir. Böylelikle doğrulama hata gerçekliği tesadüfi olmadığı daha net bir şekilde teyit edilmektedir. Örnekleme metodunun başarısının sadece belirli bir sayı ile kısıtlı olmadığını göstermek için 4 farklı sayıda eğitim ve doğrulama veri seti kullanılarak performans karşılaştırılması ile kesin olarak teyit edilmiştir. Ek olarak tüm modelleme işlemleri 1 dakikanın altı bir sürede gerçekleşmiştir. En kısa optimizasyon süresi 10 saniye ve en uzun optimizasyon süresi ise 58 saniye sürmüştür. Performans değerlendirmesinde yine literatürdekilere benzer olarak ortalama kare hata (MSE), ortalama mutlak hata (MAE) ve korelasyon katsayısı (R2) kullanılmıştır. Ayrıca her bölümde bir algoritma için seçilen bazı sonuçlar grafik olarak gösterilmiştir. Tüm sonuçlar göz önüne alındığında Monte Carlo örnekleme yöntemi bu işte daha iyi uygulanabilirliğe sahip olduğu gözükmemektedir. Buradan Monte Carlo örnekleme yöntemini bu tarz bir modelleme için kullanılacak olan veri setini oluşturmada bir efekt algoritması olarak kabul

edilebilir. Dolayısı ile farklı modelleme problemlerinde veri seti oluşturmada Monte Carlo örnekleme yöntemini direk olarak tercih edilebilir ve bu çalışma referans olarak gösterilebilir.

### Çıkar Çatışması (Conflict of Interest)

Yazar tarafından herhangi bir çıkar çatışması beyan edilmemiştir. No conflict of interest was declared by the author.

### Kaynaklar (References)

- Chávez-Hurtado, J. L., Rayas-Sánchez, J. E., Brito-Brito, Z., 2016. Multiphysics polynomial-based surrogate modeling of microwave structures in frequency domain. IEEE MTT-S Latin America Microwave Conference (LAMC).
- Rayas-Sánchez, J. E., Chávez-Hurtado, J. L., Brito-Brito, Z., 2015. Enhanced formulation for polynomial-based surrogate modeling of microwave structures in frequency domain. IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO).
- Rayas-Sánchez, J. E., Vargas-Chávez, N., 2010. Design optimization of microstrip lines with via fences through surrogate modeling based on polynomial functional interpolants. 19th Topical Meeting on Electrical Performance of Electronic Packaging and Systems, 125-128.
- Rayas-Sánchez, J. E., Aguilar-Torrentera, J., Jasso-Urzúa, J. A., 2010. Surrogate modeling of microwave circuits using polynomial functional interpolants. IEEE MTT-S International Microwave Symposium, 197-200.
- Khusro, A., Hashmi, M. S., Ansari, A. Q., 2018. Exploring Support Vector Regression for Modeling of GaN HEMT. IEEE MTT-S International Microwave and RF Conference (IMaRC).
- Güneş F., Demirel S., Mahouti P., 2014. Design of a Front-End Amplifier for the Maximum Power Delivery and Required Noise by HBMO with Support Vector Microstrip Model. Radioengineering, 23(1).
- Geng, L., Gao, X., 2014. Support vector machine dynamic modeling method and its application in the fermentation process. The 26th Chinese Control and Decision Conference (2014 CCDC), 4478-4482.
- Jacobs, J. P., Koziel, S., Ogurtsov, S., 2013. Computationally Efficient Multi-Fidelity Bayesian Support Vector Regression Modeling of Planar Antenna Input Characteristics. IEEE Transactions on Antennas and Propagation, 61(2), 980-984.
- Tokan, N. T., Güneş, F., 2008. Analysis and Synthesis of the Microstrip Lines Based on Support Vector Regression. 38th European Microwave Conference, 1473-1476.
- Yuan, L., Yang, X-S., Wang, C., Wang, B-Z., 2020. Multibranch Artificial Neural Network Modeling for Inverse Estimation of Antenna Array Directivity. IEEE Transactions on Antennas and Propagation, 68(6), 4417-4427.
- Xiao, L-Y., Shao, W., Jin, F-L., Wang, B-Z., 2018. Multiparameter Modeling with ANN for Antenna Design. IEEE Transactions on Antennas and Propagation, 66(7), 3718-3723.
- Kapetanakis, T. N., Vardiambasis, I. O., Ioannidou, M. P., Maras, A., 2018. Neural Network Modeling for the Solution of the Inverse Loop Antenna Radiation Problem. IEEE Transactions on Antennas and Propagation, 66(11), 6283-6290.
- Xiao, L-Y., Shao, W., Jin, F-L., Wang, B-Z., Liu, Q. H., 2021. Inverse Artificial Neural Network for Multiobjective Antenna Design. IEEE Transactions on Antennas and Propagation, 69(10), 6651-6659.
- Mahouti, P., Güneş, F., Belen, M. A., Demirel, S., 2017. Symbolic Regression for Derivation of an Accurate Analytical Formulation using "Big Data" An Application Example. Applied Computational Electromagnetics Society Journal, 32(5), 372-380.
- Uluslu, A., 2021. Rekabetçi Evrimsel Algoritmalar ile Yuvarlak Papyon Anten Tasarımı. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 12(4), 551-564, DOI: 10.24012/dumf.1001870.
- Uluslu, A., 2021. Triangular Bowtie Antenna Design and Modelling - Research & Reviews in Engineering. Gece Kitaplığı Yayınevi, 93-106.
- Güneş, F., Mahouti, P., Demirel, S., Belen, M. A., Uluslu, A., 2017. Cost-effective GRNN-based modeling of microwave transistors with a reduced number of measurements. International journal of numerical modelling: electronic networks, devices and fields, 30(3-4).
- Mahouti, P., Güneş, F., Demirel, S., Uluslu, A., Belen, M. A., 2014. Efficient scattering parameter modeling of a microwave transistor using Generalized Regression Neural Network. 20th International Conference on Microwaves, Radar and Wireless Communications (MIKON).
- Liu, B., Yang, H., Lancaster, M. J., 2018. Synthesis of coupling matrix for duplexers based on a self-adaptive differential evolution algorithm. IEEE Transactions on Microwave Theory and Techniques, 66(2), 813-821.
- Uluslu, A., 2021. Süper Geniş Bantlı Antenin Uygun Maliyetli MLP Tabanlı Modellenmesi. 4th International Conference on Data Science and Applications (ICONDATA), 1(21), 138-141.
- Mahouti, M., Kuşkonmaz, N., Mahouti, P., Belen M. A., Palandoken, M., 2020. Artificial neural network application for novel 3D printed nonuniform ceramic reflectarray antenna. International journal of numerical modelling: electronic networks, devices and fields, 33(6).
- Uluslu, A., 2021. Design of Microstrip Filter by Modeling with Reduced Data. The Applied Computational Electromagnetics Society Journal (ACES), 36(11), 1453-1459.
- Calik, N., Belen, M. A., Mahouti, P., 2019. Deep learning base modified MLP model for precise scattering parameter prediction of capacitive feed antenna. International journal of numerical modelling: electronic networks, devices and fields, 33(2).
- Mahouti, P., 2019. Design optimization of a pattern reconfigurable microstrip antenna using differential evolution and 3D EM simulation-based neural network model. International Journal of RF and Microwave Computer-Aided Engineering, 29(8).
- Lei, G.Y., 2003. Study on Monte Carlo and quasi-Monte Carlo methods, Zhejiang Province.
- McKay, M. D., Beckman, R. J., Conover, W. J., 1979. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics, 21(2), 239-245.
- Uluslu, A., 2021. Microstrip Low Pass Filter Analysis and Design. Research & Reviews in Engineering- Gece Kitaplığı Yayınevi, 35-52.



- Mahouti, T., Yıldırım, T., Kuşkonmaz, N., 2021. Artificial intelligence-based design optimization of nonuniform microstrip line band pass filter. *International Journal of Numerical Modelling*.
- Mahouti, T., Kuşkonmaz, N., Yıldırım, T., 2019. Modelling of Non-Planar Microstrip Lines via Artificial Neural Networks. *Innovations in Intelligent Systems and Applications Conference (ASYU)*.
- Bose, T., Gupta, N., 2012. Design of an aperture-coupled microstrip antenna using a hybrid neural network. *IET Microwaves, Antennas & Propagation*, 6(4), 470-474.
- Akdagli, A., Toktas, A., Kayabasi, A., Develi, İ., 2013. An Application of Artificial Neural Network to Compute the Resonant Frequency of E-Shaped Compact Microstrip Antennas. *Journal of Electrical Engineering*, 64(5), 317-322.
- Kayabasi, A., Toktas, A., Akdagli, A., Mustafa B. Bicer, B.M., Ustun, D., 2014. Applications of ANN and ANFIS to Predict the Resonant Frequency of L-Shaped Compact Microstrip Antennas. *The Applied Computational Electromagnetics Society Journal (ACES)*, 29(6), 460-469.
- Kayabasi, A., Akdagli, A., 2015. A Comparative Study on ANN, ANFIS and SVM Methods for Computing Resonant Frequency of A-Shaped Compact Microstrip Antennas. *17th International Conference on Communication and Information Technology and Engineering*, 9(8).
- Vuran, S., 2016. Dikdörtgen mikroşerit antenin rezonans frekansının hesaplanmasında yapay sinir ağı ve destek vektör makinası yöntemlerinin karşılaştırılması. Yüksek lisans tez.
- Uluslu, A., 2021. Kültürel Algoritma Tabanlı 24 GHz ISM Bant Uygulamaları için H Şeklinde Yama Anten Tasarımı. *10th Union Radio-Scientifique Internationale/International Union of Radio Science (URSI'2021)*.
- Uluslu, A., 2021. Triangular Patch Antenna Optimization for 77 GHz Automotive Radar Based on Genetic Algorithm. *II. International Conference on Innovative Engineering Applications (CIEA'2021)*.
- Balanis, C.A., 1982. *Antenna Theory Analysis and Design*, John Wiley and Sons, Arizona State University, 4-6.
- Safran, M.I., Aydın, E., 2006. Pek Geniş Bant Anten Tasarımı ve İmalatı, Atılım Üniversitesi.
- Hossain, F., Anagnostou, E. N., Bagtzoglou, A. C., 2005. On Latin Hypercube sampling for efficient uncertainty estimation of satellite rainfall observations in flood prediction, 32, 776-792.
- Wan, Y., Lv, Z. Z., Yuan, X. K., 2008. Latin Hypercube sampling and updated Latin Hypercube sampling method for reliability sensitivity and its variance analysis, 30(6), 927-934.
- Wei, Q., Lu, W. X., Tian, Z. J., 2004. Application of Monte-Carlo to annual precipitation forecast. *Journal of Arid Land Resources and Environment*, 4(18), 144-146.
- Jin, C., 2005. Study on random number generator and random sampling in Monte Carlo method. *Dalian University of Technology*.
- Zhou, Y., 1997. Study of Sampling Methods in Monte Carlo Simulation. *Structure Environment Engineering*, (3), 14-18.
- Şen, Z., 2004. *Yapay Sinir Ağları İlkeleri*, Su Vakfı Yayınları, İstanbul.