



**İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**YAPAY SİNİR AĞLARI İLE FİYAT TAHMİN ANALİZİ**

**Mehdi ALİZADEH**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman**

**Prof. Dr. Sabri ARIK**

**Haziran, 2011**

**İSTANBUL**



**İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YÜKSEK LİSANS TEZİ**

**YAPAY SİNİR AĞLARI İLE FİYAT TAHMİN ANALİZİ**

**Mehdi ALİZADEH**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman**

**Prof. Dr. Sabri ARIK**

**Haziran, 2011**

**İSTANBUL**

Bu çalışma 06/07/2011 tarihinde ařağıdaki jüri tarafından Bilgisayar Mühendisliğı Anabilim Dalı Bilgisayar Mühendisliğı programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi

Prof. Dr. Sabri ARIK (Danışman)  
İstanbul Üniversitesi  
Mühendislik Fakültesi

Prof. Dr. Ahmet SERTBAŞ  
İstanbul Üniversitesi  
Mühendislik Fakültesi

Prof. Dr. Mukden UĞUR  
İstanbul Üniversitesi  
Mühendislik Fakültesi

Prof. Dr. A. Halim ZAİM  
İstanbul Ticaret Üniversitesi  
Fen Bilimleri Enstitüsü

Prof. Dr. Serdar ÖZOĞUZ  
İstanbul Teknik Üniversitesi (İTÜ)  
Mühendislik Fakültesi

## **ÖNSÖZ**

Bu çalışma İstanbul Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi olarak hazırlanan “**Yapay Sinir Ağları İle Fiyat Tahmin Analizi (Price-Prediction Analysis Using Neural Networks)**” isimli tezi içermektedir.

Yüksek lisans öğrenimim sırasında ve tez çalışmalarım boyunca gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Prof. Dr. Sabri ARIK’a en içten dileklerle teşekkür ederim.

Her konuda beni destekleyen ve yardımlarını, uzakta bile olsalar, benden esirgemeyen aileme teşekkür ediyorum.

**Haziran 2011**

**Mehdi ALİZADEH**

## İÇİNDEKİLER

ÖNSÖZ.....	i
İÇİNDEKİLER .....	ii
ŞEKİL LİSTESİ.....	iv
TABLO LİSTESİ .....	vii
ÖZET.....	ix
SUMMARY .....	x
1. GİRİŞ .....	1
2. GENEL KISIMLAR .....	2
2.1. TAHMİN (PREDICTION) .....	2
2.2. YAPAY SİNİR AĞLARININ TAHMİN İÇİN KULLANIMI .....	6
2.3. DAHA ÖNCE YAPILAN ÇALIŞMALAR.....	9
2.3.1. Birinci Modelin Tanımı .....	9
2.3.2. Birinci Modelin Dezavantajları.....	13
2.3.3. İkinci Modelin Tanımı (Başarısız Bir Deneme).....	14
2.3.4. İkinci Modelin Dezavantajları .....	15
2.4. NARX YAPAY SİNİR AĞLARI.....	16
3. MALZEME VE YÖNTEM .....	18
3.1. FİYAT GEÇMİŞİ VERİ FORMATI VE DÜZENLEMELER .....	18
3.1.1. Fiyat Geçmiş Veri Formatı.....	18
3.1.2. Verileri Düzleştirmek (Smoothing).....	19
3.2. YAPAY SİNİR AĞ MODELLEME .....	21
3.2.1. Giriş ve Çıktılar ve Çalışma Mantığı .....	21
3.3. MATLAB İLE GELİŞTİRİLEN UYGULAMANIN ANLATIMI.....	25
3.3.1. Veri Tabanı Yapısı .....	25
3.3.2. Fiyat Verisini Yükleme.....	27

3.3.3. Verileri Kullanım İçin Temizlemek ve Düzleştirme.....	27
3.3.4. Yapay Sinir Ağını Oluşturmak .....	29
3.3.5. Yapay Sinir Ağını Eğitmek.....	30
3.3.6. Tahmin Aşaması (Prediction Phase).....	33
3.3.7. Tahmin Sonuç Analizi.....	35
<b>4. BULGULAR.....</b>	<b>37</b>
4.1. UYGULAMA SONUÇLARI .....	37
<b>5. TARTIŞMA VE SONUÇ.....</b>	<b>68</b>
<b>KAYNAKLAR .....</b>	<b>73</b>
<b>EKLER.....</b>	<b>75</b>
<b>EK-1 : MATLAB ORTAMINDA GELİŞTİRİLEN UYGULAMANIN KAYNAK KODU .....</b>	<b>75</b>
<b>EK-2 : ZAMAN SERİSİNİ DİLİMLERE AYIRMA YÖNTEMİ İÇİN YAZILAN MATLAB KODU (BU YÖNTEM KULLANILMAMIŞTIR):.....</b>	<b>87</b>
<b>EK-3 : MATLAB'DEN MYSQL'E BAĞLANMAK .....</b>	<b>89</b>
<b>ÖZGEÇMİŞ.....</b>	<b>90</b>

## ŞEKİL LİSTESİ

Şekil 2.1	: Fiyat değişimlerini gösteren bir zaman serisi.....	3
Şekil 2.2	: Fiyat geçmişi ve gelecekteki zaman dilimi hakkında tahminde bulunmak.	5
Şekil 2.3	: Geri dönüşümlü yapay sinir ağ modeli (SRN).....	8
Şekil 2.4	: Zaman serisi tahmini için bir model ve eğitim anındaki veri girişleri. ....	10
Şekil 2.5	: Tahmin için bir yapay sinir ağ modeli ve etken değişkenlerin kullanımı.	11
Şekil 2.6	: Kaydırma yaparak eğitim gruplarını oluşturmak.....	11
Şekil 2.7	: Veri gruplandırması, eğitim, doğrulama ve deneme grupları.....	12
Şekil 2.8	: TimeSlicing metodu (verileri eş boyutlu dilimlere ayırmak). ....	15
Şekil 2.9	: Narx ANN model: SP –Mode [5].....	17
Şekil 2.10	: Narx ANN model: P –Mode [5].....	17
Şekil 3.1	: Örnek veri grafiği .....	19
Şekil 3.2	: Örnek veriler üzerinde düzleştirme işleminin sonucu. ....	20
Şekil 3.3	: Kullanılan yapay sinir ağ modeli (bu şema Matlab programı ile çıkarılmıştır).....	21
Şekil 3.4	: Kullanılan ağ modelini eğitmek için kullanılan vektörler ve fiyat tahmin çıkışı gösterilmektedir.....	23
Şekil 3.5	: Örnek olarak Şekil 3.4'te gösterilen fiyat verileri için sistemin ürettiği sonuç. ....	24
Şekil 3.6	: Veri tabanında ürün tablosu (product table).....	26
Şekil 3.7	: Veri tabanında fiyat geçmişi tablosu (productsPriceHistory Table) .....	26
Şekil 3.8	: Revize edilen verilerin örnek grafiği .....	28
Şekil 3.9	: Yapay sinir ağı eğitim aşaması (Bu şema Matlab programından alınmıştır). ....	32
Şekil 3.10	: Yapay sinir ağı eğitim sonrası hata grafiği.....	33
Şekil 3.11	: Örnek veri üzerinde tahmin sonucu bu grafikte gösterilmiştir.....	35
Şekil 3.12	: Örnek hedef ve çıkış vektörleri için tahmin analizi ve en iyi uyum hatları gösterilmiştir. ....	36
Şekil 4.1	: Sonuç 1 .....	38
Şekil 4.2	: Sonuç Analizi 1 .....	38
Şekil 4.3	: Sonuç 2 .....	39
Şekil 4.4	: Sonuç Analizi 2 .....	39
Şekil 4.5	: Sonuç 3 .....	40
Şekil 4.6	: Sonuç Analizi 3 .....	40

<b>Şekil 4.7</b>	: Sonuç 4 .....	41
<b>Şekil 4.8</b>	: Sonuç Analizi 4 .....	41
<b>Şekil 4.9</b>	: Sonuç 5 .....	42
<b>Şekil 4.10</b>	: Sonuç Analizi 5 .....	42
<b>Şekil 4.11</b>	: Sonuç 6 .....	43
<b>Şekil 4.12</b>	: Sonuç Analizi 6 .....	43
<b>Şekil 4.13</b>	: Sonuç 7 .....	44
<b>Şekil 4.14</b>	: Sonuç Analizi 7 .....	44
<b>Şekil 4.15</b>	: Sonuç 8 .....	45
<b>Şekil 4.16</b>	: Sonuç Analizi 8 .....	45
<b>Şekil 4.17</b>	: Sonuç 9 .....	46
<b>Şekil 4.18</b>	: Sonuç Analizi 9 .....	46
<b>Şekil 4.19</b>	: Sonuç 10 .....	47
<b>Şekil 4.20</b>	: Sonuç Analizi 10 .....	47
<b>Şekil 4.21</b>	: Sonuç 11 .....	48
<b>Şekil 4.22</b>	: Sonuç Analizi 11 .....	48
<b>Şekil 4.23</b>	: Sonuç 12 .....	49
<b>Şekil 4.24</b>	: Sonuç Analizi 12 .....	49
<b>Şekil 4.25</b>	: Sonuç 13 .....	50
<b>Şekil 4.26</b>	: Sonuç Analizi 13 .....	50
<b>Şekil 4.27</b>	: Sonuç 14 .....	51
<b>Şekil 4.28</b>	: Sonuç Analizi 14 .....	51
<b>Şekil 4.29</b>	: Sonuç 15 .....	52
<b>Şekil 4.30</b>	: Sonuç Analizi 15 .....	52
<b>Şekil 4.31</b>	: Sonuç 16 .....	53
<b>Şekil 4.32</b>	: Sonuç Analizi 16 .....	53
<b>Şekil 4.33</b>	: Sonuç 17 .....	54
<b>Şekil 4.34</b>	: Sonuç Analizi 17 .....	54
<b>Şekil 4.35</b>	: Sonuç 18 .....	55
<b>Şekil 4.36</b>	: Sonuç Analizi 18 .....	55
<b>Şekil 4.37</b>	: Sonuç 19 .....	56
<b>Şekil 4.38</b>	: Sonuç Analizi 19 .....	56
<b>Şekil 4.39</b>	: Sonuç 20 .....	57
<b>Şekil 4.40</b>	: Sonuç Analizi 20 .....	57



<b>Şekil 4.41</b>	: Sonuç 21 .....	58
<b>Şekil 4.42</b>	: Sonuç Analizi 21 .....	58
<b>Şekil 4.43</b>	: Sonuç 22 .....	59
<b>Şekil 4.44</b>	: Sonuç Analizi 22 .....	59
<b>Şekil 4.45</b>	: Sonuç 23 .....	60
<b>Şekil 4.46</b>	: Sonuç Analizi 23 .....	60
<b>Şekil 4.47</b>	: Sonuç 24 .....	61
<b>Şekil 4.48</b>	: Sonuç Analizi 24 .....	61
<b>Şekil 4.49</b>	: Sonuç 25 .....	62
<b>Şekil 4.50</b>	: Sonuç Analizi 25 .....	62
<b>Şekil 4.51</b>	: Sonuç 26 .....	63
<b>Şekil 4.52</b>	: Sonuç Analizi 26 .....	63
<b>Şekil 4.53</b>	: Sonuç 27 .....	64
<b>Şekil 4.54</b>	: Sonuç Analizi 27 .....	64
<b>Şekil 4.55</b>	: Sonuç 28 .....	65
<b>Şekil 4.56</b>	: Sonuç Analizi 28 .....	65
<b>Şekil 4.57</b>	: Sonuç 29 .....	66
<b>Şekil 4.58</b>	: Sonuç Analizi 29 .....	66
<b>Şekil 4.59</b>	: Sonuç 30 .....	67
<b>Şekil 4.60</b>	: Sonuç Analizi 30 .....	67
<b>Şekil 5.1</b>	: Sinüs fonksiyonu tipinde olan bir fiyat geçmişi vektörü.....	69
<b>Şekil 5.2</b>	: Sinüs fonksiyonu tipinde olan giriş vektörüne ait sonuç.....	70
<b>Şekil 5.3</b>	: Sinüs fonksiyonu tipinde olan giriş vektörüne ait sonuç analizi .....	70
<b>Şekil 5.4</b>	: Kosinüs fonksiyonu tipinde olan bir fiyat geçmişi vektörü.....	71
<b>Şekil 5.5</b>	: Kosinüs fonksiyonu tipinde olan giriş vektörüne ait sonuç.....	71
<b>Şekil 5.6</b>	: Kosinüs fonksiyonu tipinde olan giriş vektörüne ait sonuç analizi .....	72

## TABLO LİSTESİ

<b>Tablo 4.1</b>	: Analiz sonucu 1.....	38
<b>Tablo 4.2</b>	: Analiz sonucu 2.....	39
<b>Tablo 4.3</b>	: Analiz sonucu 3.....	40
<b>Tablo 4.4</b>	: Analiz sonucu 4.....	41
<b>Tablo 4.5</b>	: Analiz sonucu 5.....	42
<b>Tablo 4.6</b>	: Analiz sonucu 6.....	43
<b>Tablo 4.7</b>	: Analiz sonucu 7.....	44
<b>Tablo 4.8</b>	: Analiz sonucu 8.....	45
<b>Tablo 4.9</b>	: Analiz sonucu 9.....	46
<b>Tablo 4.10</b>	: Analiz sonucu 10.....	47
<b>Tablo 4.11</b>	: Analiz sonucu 11.....	48
<b>Tablo 4.12</b>	: Analiz sonucu 12.....	49
<b>Tablo 4.13</b>	: Analiz sonucu 13.....	50
<b>Tablo 4.14</b>	: Analiz sonucu 14.....	51
<b>Tablo 4.15</b>	: Analiz sonucu 15.....	52
<b>Tablo 4.16</b>	: Analiz sonucu 16.....	53
<b>Tablo 4.17</b>	: Analiz sonucu 17.....	54
<b>Tablo 4.18</b>	: Analiz sonucu 18.....	55
<b>Tablo 4.19</b>	: Analiz sonucu 19.....	56
<b>Tablo 4.20</b>	: Analiz sonucu 20.....	57
<b>Tablo 4.21</b>	: Analiz sonucu 21.....	58
<b>Tablo 4.22</b>	: Analiz sonucu 22.....	59
<b>Tablo 4.23</b>	: Analiz sonucu 23.....	60
<b>Tablo 4.24</b>	: Analiz sonucu 24.....	61
<b>Tablo 4.25</b>	: Analiz sonucu 25.....	62
<b>Tablo 4.26</b>	: Analiz sonucu 26.....	63
<b>Tablo 4.27</b>	: Analiz sonucu 27.....	64
<b>Tablo 4.28</b>	: Analiz sonucu 28.....	65
<b>Tablo 4.29</b>	: Analiz sonucu 29.....	66
<b>Tablo 4.30</b>	: Analiz sonucu 30.....	67

<b>Tablo 5.1</b>	: Sinüs fonksiyonu tipinde olan giriş vektörüne ait analiz sonucu.....	70
<b>Tablo 5.2</b>	: Kosinüs fonksiyonu tipinde olan giriş vektörüne ait analiz sonucu.....	72

## ÖZET

### YAPAY SİNİR AĞLARI İLE FİYAT TAHMİN ANALİZİ

Bu çalışmada yapay sinir ağları ile ürünlerin fiyat geçmişi bilgilerini kullanarak ilerleyen zaman birimlerinde fiyat tahminlerinin yapılması amaçlanmıştır.

Çalışmada öncelikle tahmin kavramı anlatıldıktan sonra yapay sinir ağlarının özelliklerine ve öğrenme kabiliyetine dayanarak, tahmin yürütmede uygun bir araç oldukları ve kullanımları anlatılmıştır. Devamında tahmin için genelde uygun olan yapay sinir ağ tipleri kısaca anlatılmıştır.

Çalışmanın devamında bu konuda daha önce yapılan çalışmalar arasından kullanılan bir yapay sinir ağ modeli ve bizim deneme olarak yaptığımız alternatif bir model yer almaktadır. Bu bölümde tanımlanan modellerin tahmin yürütmede ve çalışma performanslarında dezavantajları ve başarısız noktaları listelenmiştir.

Çalışmanın 3. bölümünde bu projede kullanılan yöntem açıklanmıştır. Öncelikle elde olan fiyat geçmişi veri formatları ve kullanım için verilerin düzenlenmesi anlatılmış ve ardından Narx yapay sinir ağ modeli kısaca tanıtılmıştır. Devamında bu projede kullanılan yapay sinir ağ modeli açıklanmış ve sistemin girişleri, çıkışları ve çalışma mantığı anlatılmıştır.

Devamında Matlab ortamında modelin yapılan uygulaması detaylı bir şekilde anlatılmıştır. veritabanının yapısı ve geliştirilen uygulamanın genel akışı bu bölümde yer almaktadır.

Çalışmanın 4. bölümünde veritabanından çekilen örnek 30 adet farklı ürün bilgisi için programın ürettiği çıkışlar gösterilmiştir. Devamında sonuç ve performans analizi yapılmıştır. Bu kısımda ilave olarak sistemin mantıklı sonuçlar verebildiğini denemek için sinüs ve kosinüs fonksiyonları formatında olan veri tipleri giriş olarak sisteme uygulanmış ve sistemin beklentiler doğrultusunda ürettiği çıkışlar, gösterilmiştir.

Geliştirilen Matlab programının kaynak kod çıktısı da Ekler bölümünde yerleştirilmiştir.

## **SUMMARY**

### **PRICE-PREDICTION ANALYSIS USING NEURAL NETWORKS**

In this study with the aid of artificial neural networks using price history information of products, we have tried to predict price changes in the upcoming time steps.

First of all we will have a brief definition of “prediction” and then artificial neural networks are mentioned as a suitable tool for prediction due to their learning abilities and other specifications. Subsequently artificial neural network types which are generally used and suitable for prediction are briefly discussed.

Subsequently earlier studies on price prediction using neural networks with an implemented model is discussed. And also an alternative model that we tested for this project but was not successful is included. The disadvantages and weak points of these models are listed afterwards.

In the section 3 the method used in our project is described. First price history data format and data preparation process for usage is explained. Subsequently Narx Artificial neural network model is briefly introduced. Then the artificial neural network model designed for this project and its input, output and system logic is described.

Subsequently the project implementation in Matlab environment is explained with details. Used database table formats and general program flow takes place in this part.

In the section 4, the system results generated for thirty different products fetched from database, is shown in figures. Subsequently conclusion and system performance analysis is done. In addition for testing that system generates logical outputs, system result for assumed price history in easily guessable sinus and cosinus function formats are displayed in figures which meet our expectations.

The developed Matlab program complete source code is also included in the appendix section.

## 1. GİRİŞ

Bu çalışmada ürünlerin geçmişte olan fiyat değişimlerine bakarak yapay sinir ağları ile ilerideki fiyat değişimleri hakkında tahminde bulunmak amaçlanmıştır. Fiyat tahminleri satıcılara satış politikası ve stok belirlemede, aynı zamanda tüketiciler için satın almaya karar vermede yararlı bilgiler sunabilir.

Burada yapay sinir ağları geçmişteki fiyat bilgileri ile eğitilip, bir sonraki zaman diliminde fiyatların ne gibi bir değişiklik göstereceği hakkında tahminde bulunmaya çalışılacaktır.

Yapay sinir ağlarının eğitilebilmesi ve insan beynine benzer bir şekilde tecrübe kazanma ve benzer durumlarla karşılaşıncı uygun cevaplar üretebilme özelliği sayesinde, tahmin yürütme gibi durumlarda da işe yarar bir araç olarak kullanılabilir.

Sistemin ürettiği çıkışlar sonuçta bir tahmin olduğu için ileride oluşacak gerçek fiyatlarla aynı olamaz ama en azından fiyatların gelecekteki durumu hakkında bir fikir verebilir.

Sonuçlar bölümünde göreceğimiz gibi yüksek seviyede değişim göstermeyen veya keskin iniş çıkışlara sahip olmayan giriş verileri için sistem mantıklı sonuçlar üretebiliyor. Bu çıkışlar sonuçlar bölümünde, ilgili dönemlerin gerçek fiyatları ile karşılaştırılmıştır.

Bu çalışmada sistemin giriş verileri olarak sadece fiyat geçmişi kullanılmıştır ve fiyat üzerinde etkisi olan diğer değişkenler (döviz kur değişimi gibi) göz ardı edilmiş sadece zaman serisi tahmin yöntemine yoğunlaşmıştır. Fiyat geçmişi üzerinde tüm diğer değişkenlerin zaman içerisinde zaten etkisi bulunduğu için diğer etkenler giriş bilgilerine dahil edilmeyip kullanımı 2. bölümde tartışılmıştır.

## 2. GENEL KISIMLAR

### 2.1. TAHMİN (PREDICTION)

Bir olayın oluşumu hakkında iddiada bulunmaya tahmin denir, ki genelde geçmişteki ve şu anki verilere dayanarak yapılır [1]. Herkes günlük hayatında çeşitli tahmin problemlerini belirli başarı seviyesi ile çözmektedir. Örneğin hava durumu, enerji tüketimi, döviz kur değişiklikleri, ürün stok paylaşımları ve bunlar gibi bir çok tahmin yürütmeye gerek duyan konular günlük yaşamda karşımıza çıkıyor.

Teknik olarak bir sistemin tahmin edilebilir parametreleri bazen denklemlerle anlatılıp değerlendirilebilir. Bu durumlarda denklemlerin çözülmesi ile tahmin işlemleri kolaylıkla gerçekleştirilebilir. Ama pratikte karşılaştığımız problemler için genelde bu şekilde matematiksel anlatımlar ve denklemler kurmak çok zor veya bazen hatta imkansız olur. Ayrıca bu yöntem ile çözümler çok komplike hesaplamalar haline gelip ve hatta bazen olay gerçekleştikten sonra sonuca ulaşırız. Burada değişik tahmini etkenler kullanılabilir. Örneğin tahmin edilecek değişkenin diğer olaylar üzerinde geri dönüşümde bağlılığı ki bu durumda yine tahmini etkenleri bulmak zor olabilir. Bu işlemler tahmin edilecek konu için modelleme yapma anlamına geliyor.

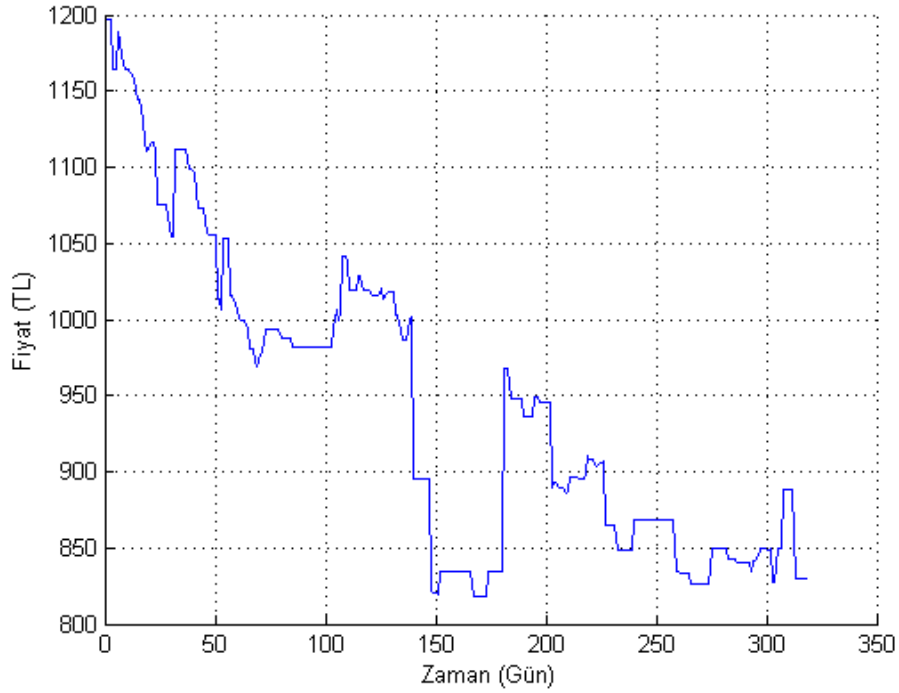
Yapay sinir ağları değişik başarı oranlarında tahminde bulunmak için kullanılabilir. Bunun avantajları ise bağlantılı olan etkenleri, ekstra bilgilere gerek kalmadan toplanmış verilerden otomatik öğrenmedir. Yapay sinir ağı geçmişteki verilerle eğitilerek verilerde olan gizli bağlılıkları bulup onları ileriye yönelik tahminde bulunmak için kullanılabilir. Diğer anlamda yapay sinir ağları açık bir model olarak belirtilmemektedir ve daha çok öğrenme kapasitesine sahip bir siyah kutudur (black box).

Değişik veri tiplerini kullanarak tahminde bulunmak mümkündür. Bu çalışmada Time Series (zaman serisi) tahminleri üzerinde çalışma yapılmıştır.

Zaman serileri deęişkenlerin zaman iersindeki deęişimini gosterir. Geri deęerlerin zerinde zaman dıřında bařka faktorlerin de etkisi vardır. Zaman serileri bir deęişkenin ayrık (discrete) gemiř deęerlerini gosterir ve rnekleme kullanılarak bir srekli (continuous) fonksiyonla elde edilebilir.

Zaman serileri tahmini, gemiř verileri analiz eder ve gelecek yaklařık deęerler iin projeksiyonda bulunur. Temel olarak bu metod doęrusal olmayan (nonlinear) bir fonksiyonu gemiř deęerlerden ıkarılan tekrarlanan iliřkileri modeller. Yinelenen iliřkiler yeni deęerlerin tahminini saęlayabilir [2].

Alttađı resimde bir zaman serisi gosterilmektedir. Bu zaman serisi zel bir rnn yaklařık 320 gn iersindeki fiyatlarını gstermektedir.



řekil 2.1: Fiyat deęişimlerini gosteren bir zaman serisi

Tahmin eřitleri deęişik kriterlere gore sınıflandırılabilir. Temel kriterler, tahmin eęitimi iin elde olan veriler ve neyin tahmin edileceđidir (deęer veya eęilim [trend]). Deęer tahminlerinde bir deęişkenin ilerleyen zamanda alabileceđi deęerler tahmin edilir. Ama eęilim (trend) tahmininde bir deęerin azalıp oęalacađı tahmin edilir ki



burada ne kadar azalıp veya çoğalacağı önemli değildir (örneğin borsa veya döviz kur değerleri).

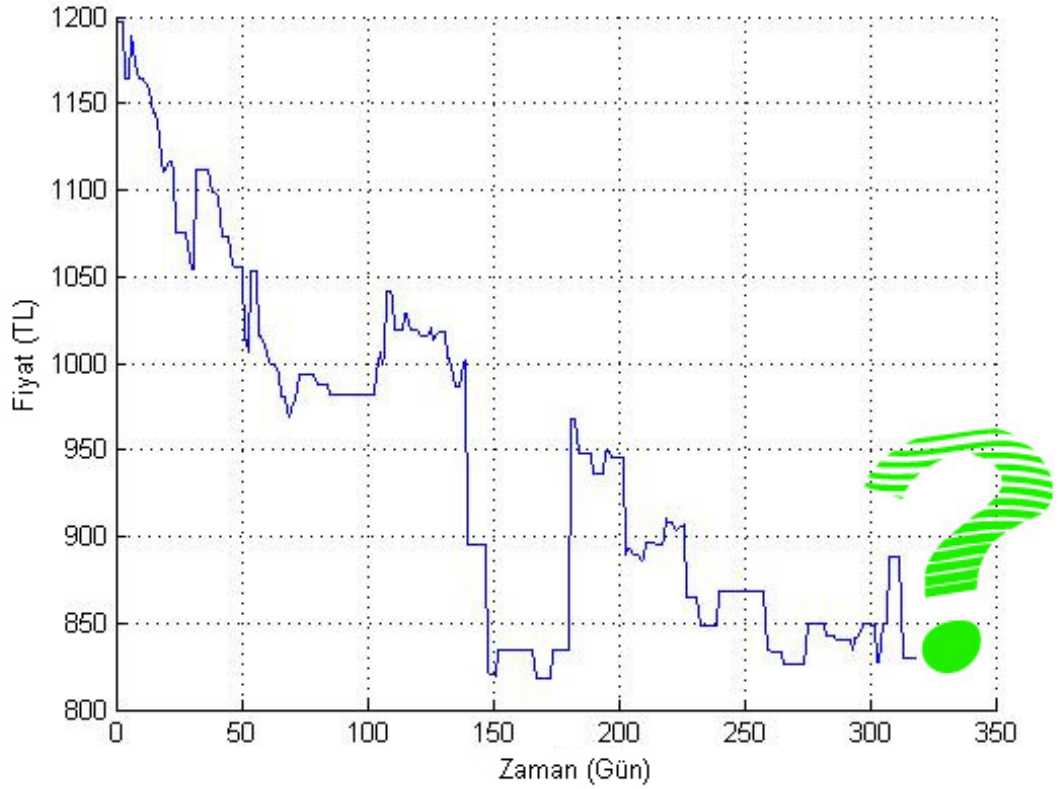
Genelde değer tahminleri eğilim (trend) tahminlerinden daha kolaydır. Ayrıca eğilim tahminlerinde belki eğilim ile alakalı ortalama değişiklik gibi diğer kısımları da tahmin etmemiz gerekebilir.

Veri tipi sınıflandırması açısından bakarsak zaman serileri için genelde bir değişkenin eşit aralıklarla alınmış değerleri elimizde mevcut olmalıdır. Bu durumda geçmişteki verilere dayanarak ilgili değişken hakkında ilerleyen zamanlarda tahmin yürütmeye çalışabiliriz. Bunu yapabilmek için geçmişe ait zaman serisi verileri yeteri kadar uzun bir döneme ait ve yeteri kadar sıkışık (dense) olmalıdır. Örneğin bu projedeki gibi bir ürünün bir aylık fiyat tahminini yapabilmek için 6 aylık veya bir yıllık bir fiyat geçmişi elimizde olmalıdır. Verilerin sıkışıklığı açısından mesela elimizdeki fiyat değerlerinin aylık olması yeteri kadar sıkışık (dense) bir veri geçmişi sayılamaz. Bu projede kullandığımız fiyat geçmişi bilgileri günlük değerlerden oluşmuştur ki bu da uygun bir sıkışıklık oranına sahiptir.

Sıkışıklık aralığını belirleme konusunda dikkat etmemiz gereken konu, ilgili değişkenin ne kadar sıklıkla değişime uğradığıdır. Örneğin ortalama ayda bir değişiklik gösteren değişken için günlük değer kaydı almak uygun değildir veya tam tersi her saat değişiklik gösteren bir değişken için günlük veya aylık veri kaydı yapmak tamamen yetersiz kalır ve değişim geçmişi ile alakalı yeterli veri sunamaz. Bu projede de ele alınan değişken ürün fiyatları olduğu ve bu durumda ürün fiyatları ortalama günlük değişimlere sahip oldukları için günlük değer kayıtları geçmişe ait uygun ve kabul edilebilir bir bilgi sunacaktır.

Zaman serisinin dışında türev gibi ekstra bilgiler de verilere eklenebilir. Bu bilgiler daha başarılı tahminler yürütmek için kullanılabilir. Tahmin edilecek zaman dilimine ait başka dahil olan etken değişkenler de yararlı olabilir. Mesela örneğin enerji tüketimi tahmini için tahmin işlemini Pazar günü veya Pazartesi için yapacağımızı bilmemiz tahmin yürütmede yardımcı olacaktır. Bu etken değişkenleri de tasarlanacak modele dahil edebilirsek sonuç üretmede yararlı olacaktır. Bu projenin konusu olan fiyat tahmini için fiyat değişimlerini etkileyen bir sürü dış etken olabilir. Döviz kur

değişiklikleri, borsa değerleri, siyasi ve ticari gelişmeler hepsi fiyatlar üzerinde etki yaratabilir. Bunların hepsini ele almak hesaplamayı gayet komplike hale getirebilir. Bu projede yapay sinir ağının tahminleri fiyat geçmişine dayanarak yapıldığı için dışarıdan etken olan değişkenler göz ardı edilmiştir ve sadece zaman serisinin bir sonraki zaman dilimindeki durumunu tahmin etmeğe yönelik çalışma yapılmıştır. Fiyat geçmişi üzerinde zaten bahsi geçen değişkenlerin hepsi geçmişte kendi etkisini fiyat değerleri üzerinde bırakmıştır. Dolayısı ile geçmişteki verilerin içinde bulunan bu etkenlerin etki değerlerini ayırmak ve dışarıdan bir etken gibi uygulamak mümkün olmayabilir veya fiyat geçmişine ait zaman dilimleri hakkında fiyat değerleri dışında diğer değişkenlerin de detaylı bilgilerinin bulunması ile mümkün olabilir.



Şekil 2.2: Fiyat geçmişi ve gelecekteki zaman dilimi hakkında tahminde bulunmak.

## 2.2. YAPAY SİNİR AĞLARININ TAHMİN İÇİN KULLANIMI

Yapay sinir ağlarının tahmin için kullanılmasının avantaj ve dezavantajlarına önce bir göz atalım. Yapay sinir ağları örnek verilerle eğitilebilir ve eğitildikten sonra verilerde gizli ve linear olmayan ilişkileri verilerde belli miktarda gürültü (noise) olsa bile, yakalayabilirler.

Dezavantaj ise yapay sinir ağları belirli bir zaman süresi için eğitilebilirler ve genelde tahminlerde hata oranı tespit edilemez. Başka bir dezavantaj ise zaman serisi tahmin aralık süresi çoğaldıkça tahminlerin dikkat oranı şiddetle azalır.

Zaman serisi olarak  $x$  değişkenine ait geçmiş değerler elimizde mevcuttur.  $x$  değişkeni zaman içerisinde değişim halindedir  $[x_t (t=1,2,...)]$ , ve biz  $x$ 'in alabileceği değeri ilerleyen  $t+h$  zamanında tahmin etmek istiyoruz.

Yapay sinir ağları ile bu tahminleri yapmak, elimizde olan geçmişte belirli bir döneme ait verilerle yapay sinir ağını eğitmek ve eğitilmiş bilgileri geleceğe uygulamaktan oluşmaktadır. Geçmişe ait veriler yapay sinir ağına uygulanacak giriş (Input) verilerinde kullanılacaktır ve geleceğe ait beklenen değerler yapay sinir ağının çıkışında (Output) yer alacaktır.

Anlatılanlara göre yapay sinir ağı eğitim türü, eğitmen sinyali kullanma türünden olacaktır. Daha önce bahsedildiği gibi ekstra etken değişkenler de kullanılabilir ama daha fazla bilgi her zaman daha iyi tahmin sonucu anlamına gelmez. Bazen fazla bilgiler yapay sinir ağının eğitim işlemini kötüleştirir. Dolayısı ile sadece ilgili önemli etkenleri, elimizde olduğu takdirde, seçmekte fayda vardır.

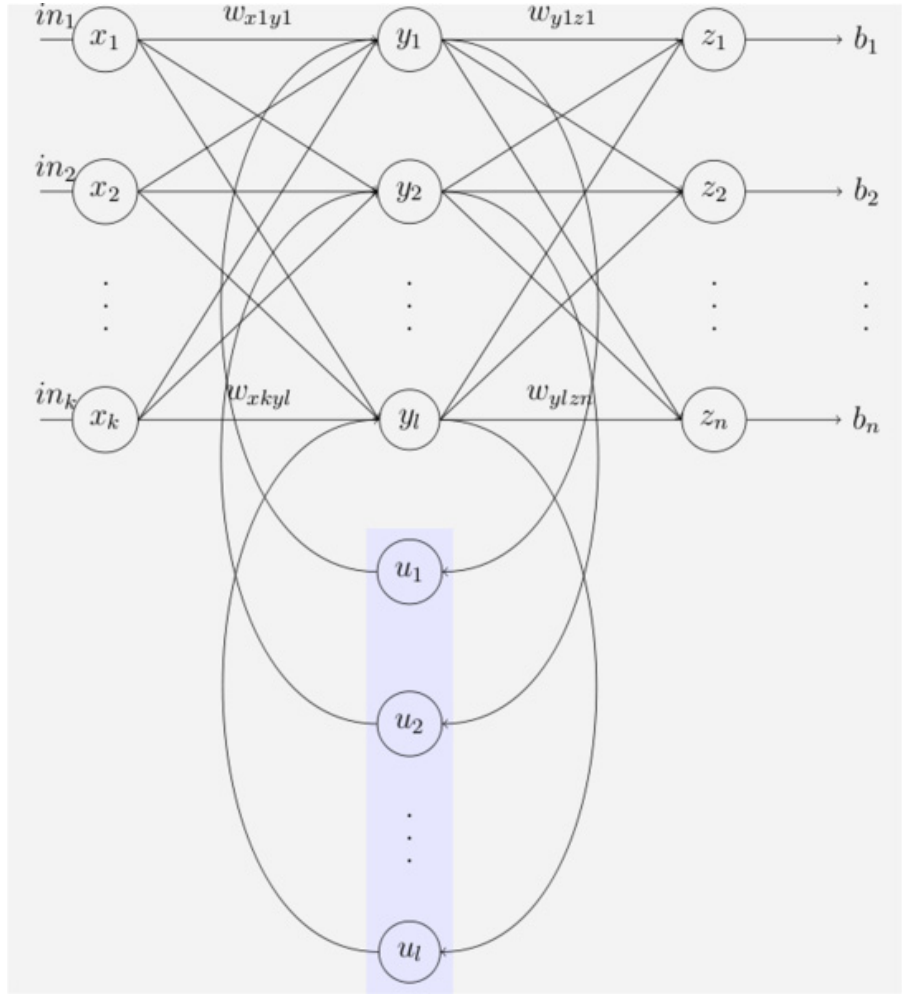
Çeşitli yapay sinir ağı modelleri tahmin için kullanılabilir. Örneğin backpropagation, ART (Adaptive Resonance Theory Neural network), Simple Recurrent Network ve Narx network. Bu projede Narx yapay sinir ağları fiyat tahminleri için kullanılmıştır. Tahmin için genellikle kullanılabilen bir kaç yapay sinir ağı burada kısaca özetlenmiştir. Bu projede kullanılan Narx yapay sinir ağı da ilerleyen kısımlardan kısaca anlatılmıştır.

**ART: The Adaptive Resonance Theory:** 1976 yılında Grossberg tarafından biyolojik olguların (biological phenomena) anlatımı için bu model tanıtıldı. Bu modelin üç önemli özelliği vardır [3].

- Toplam network aktivitelerini normalize edebilme özelliği: Biyolojik sistemler genelde çevredeki değişikliklere kendilerini adapte edebilirler. Örneğin insan gözü çevredeki ışık şiddetine göre kendisini adapte eder.
- Giriş şablonlarının (input patterns) arasında farklılık artırma (contrast enhancement): Giriş şablonlarının arasındaki ufak değişikliklerin farkında olmak, sistemin ayakta kalması için çok önem taşımaktadır.
- Farklılık artırma şablon (pattern) için kısa süreli hafıza (STM: Short Term Memory): Giriş şablonu deşifre (decode) edilmeden önce kısa süreli hafızada kaydedilmelidir.

**Simple Recurrent Network (SRN) Elman Style:** bir geri dönüşümlü (recurrent) yapay sinir ağı modeli ki bir bağlam (context) katmanı vardır [4]. Bağlam katmanı bir önceki çıkışı, çıkış katmanından alıp onu geriye doğru gizli katmanın girişine verir. Dolayısı ile gizli katman her zaman bir önceki döngüde oluşan çıkışı, giriş olarak alır. Elman yapay sinir ağları genelde genetik algoritma teknikleri, *simulated annealing* veya bir *propagation* tekniği ile eğitilir. Bu network tipleri genelde tahmin için kullanılır.

**Simple Recurrent Network (SRN) Jordan Style:** bir geri dönüşümlü (recurrent) yapay sinir ağı modeli olarak bir bağlam (context) katmana sahiptir [4]. Burada yine bir önceki döngüye ait çıkış verisi, giriş olarak gizli katmana uygulanır. Bu yapay sinir ağı modeli de genelde tahmin yapmak için kullanılır. Jordan Style modelinin Elman Style'dan farkı onun geri dönüşümünün gizli katman yerine çıkış katmanından olmasıdır.



Şekil 2.3: Geri dönüşümlü yapay sinir ağ modeli (SRN)

## 2.3. DAHA ÖNCE YAPILAN ÇALIŞMALAR

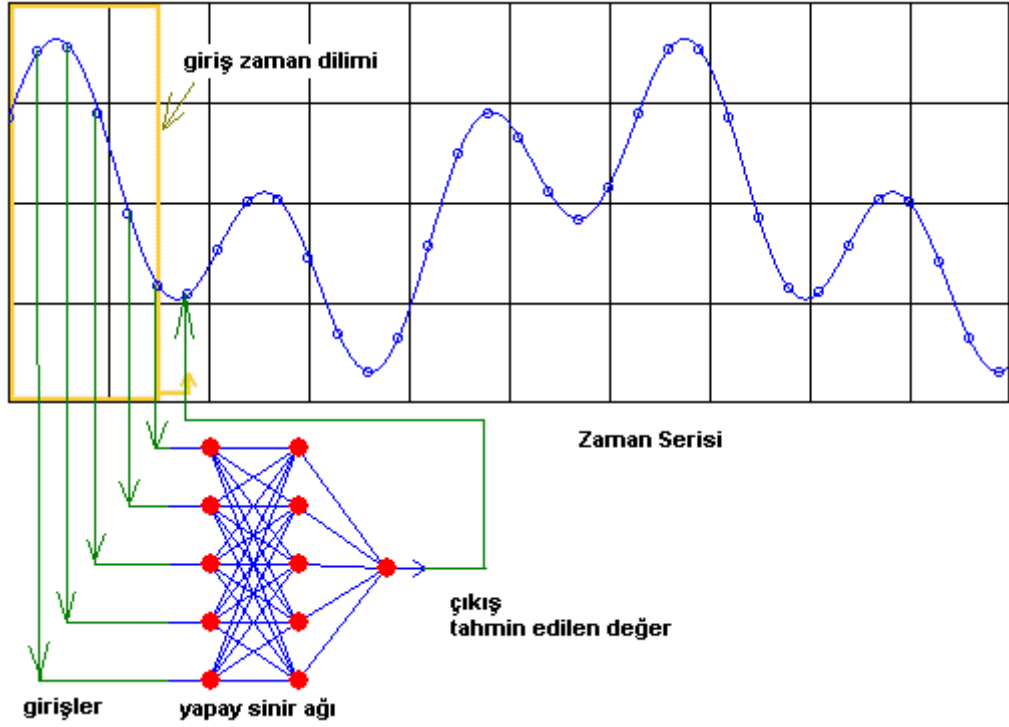
Araştırma esnasında bu konuda yapılan çalışmalar arasında bulunan bir modelin yapısı ve çalışma mantığı burada anlatılmıştır. Bu modelin dezavantajları da devamında tartışılmıştır.

### 2.3.1. Birinci Modelin Tanımı

Bu modelde de fiyat geçmişi bir zaman serisi şeklinde yapay sinir ağına giriş olarak verilmektedir. Zaman serisinde her bir veri noktası için giriş katmanında bir nöron belirlenmiştir. Yani tek seferde giriş olarak örneğin 30 eleman sayısında bir giriş verisi yapay sinir ağına uygularsak giriş katmanında 30 adet nörona ihtiyaç olacaktır. Bu network bir gizli katmana da sahiptir (giriş katmanla aynı veya ondan farklı nöron sayısından oluşabilir) ve çıkış katmanında bir adet nörondan oluşmaktadır. Bu modelin çıkışı bir tek sayıdır ve bir sonraki zaman biriminde oluşabilecek değeri tahmin etmeye çalışacaktır.

Şekil 2.4'te bu yapı gösterilmiştir. Grafikte her bir nokta zaman serisinde bir değeri (günlük fiyat) belirtmektedir.

Şekil 2.4'te görüldüğü gibi bu yapının tahmin edebileceği değer sadece bir sonraki zaman birimine ait değerdir. Örnek olarak bu yapıya 30 günlük veri uygulanırsa, 31. günde oluşabilecek değeri tahmin etmeğe çalışacaktır. Dolayısı ile ileriye doğru bir kaç günlük tahmin yürütmek için verileri kaydırma yöntemi kullanılmalıdır. Yani ilk günün verisi diziden atılır ve tahmin edilen son günün verisi diziyeye ilave edilir ve bir sonraki günün değeri tahmin edilir ve bu şekilde devam ederek bir kaç günlük ileriye doğru oluşabilecek değerleri sistemden alabiliriz.

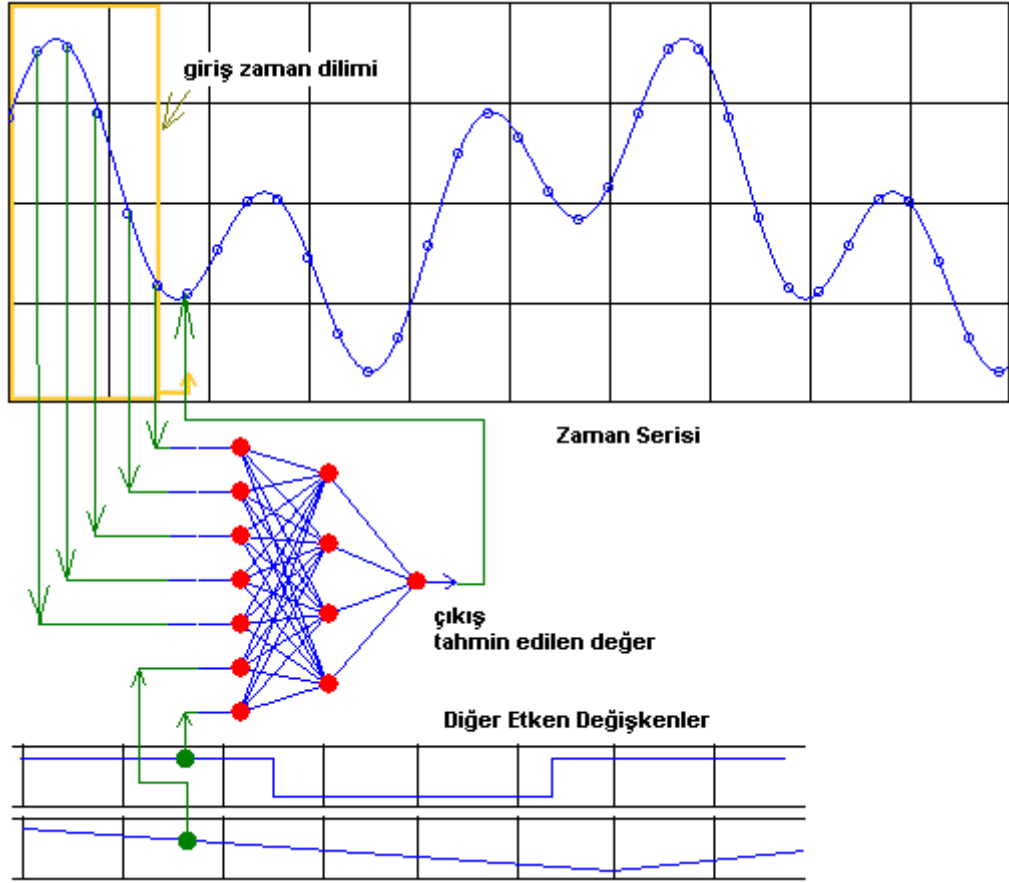


Şekil 2.4 : Zaman serisi tahmini için bir model ve eğitim anındaki veri girişleri.

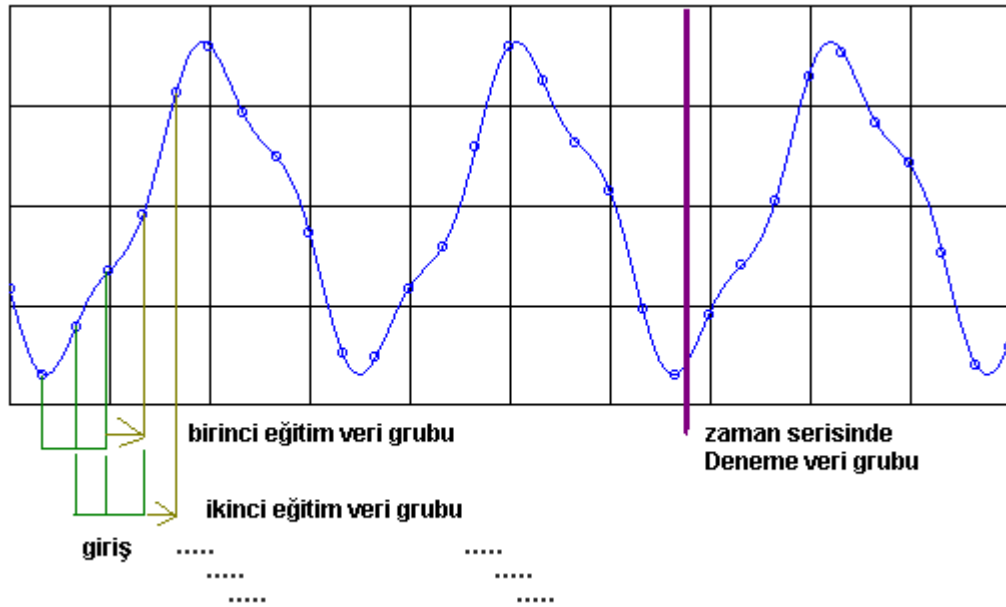
Bu modelin biraz gelişmiş hali dışardan etken değişkenleri de sisteme dahil ederek şekil 2.5'te gösterilmiştir. Burada eğer sistemi etkileyen dış değişkenlerde elde varsa onların her birisi için de giriş katmanda ayrı bir nöron yerleştirilmiştir ve bunlara aynı zaman diliminde bulunan etken değişkenlerin değeri uygulanmaktadır. Yani etken değişkenlerin hepsi için aynı zaman aralığında bir değer geçmişi, zaman serisi formatında elimizde bulunmalıdır.

Bu modellerde eğitim veri kümelerini oluşturmak için zaman ekseninde kaydırma (shifting) yaparak eğitim grupları oluşturabiliriz. Bu yöntemle oluşturulacak eğitim setleri kullanılacak network tipine göre gerekirse belirli aralıklara da uyarlanabilir (örneğin 0-1 aralığı).

Şekil 2.6'de eğitim setlerinin kaydırma yaparak oluşturulması gösterilmiştir. Bu şekilde window ibaresi giriş verilerini adlandırmak için kullanılmıştır.



Şekil 2.5: Tahmin için bir yapay sinir ağ modeli ve etken değişkenlerin kullanımı.



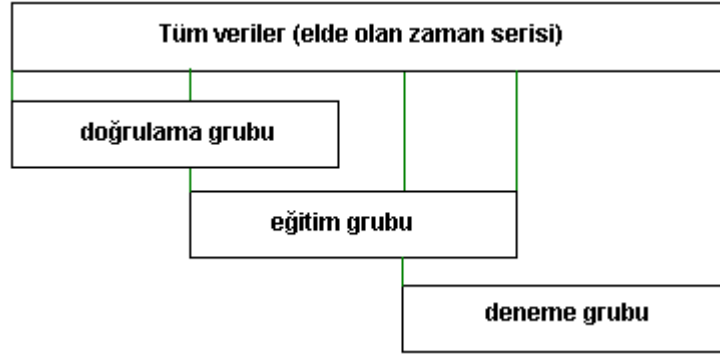
Şekil 2.6 : Kaydırma yaparak eğitim gruplarını oluşturmak.



Bu yapıda elde olan veri genelde 3 gruba ayrılır:

- Eğitim grubu (learning set)
- Doğrulama grubu (validation set)
- Deneme grubu (testing set).

Bu grupların bir biri ile küme kesişimi de olabilir ve ardardına olma zorunluluğu yoktur. Eğitim grubu adından da belli olduğu gibi, eğitim aşamasında kullanılacak veri dizisini içermektedir. İstenilen çıkışları üretmek için yapay sinir ağı bu gruba adapte edilir. Beklenen çıkışlarla sistemin ürettiği çıkışların arasındaki farkı ölçmek için, doğrulama grubu kullanılır ki bu fark ölçümü de eğitimin tamamlanabileceğine karar vermek için kullanılır. Deneme grubu da sisteme daha önce verilmemiş verilerle, son aşamada yapay sinir ağının doğru çalışıp çalışmadığını test etmek için kullanılır. Alttaki şekilde bu veri grupları gösterilmiştir.



Şekil 2.7: Veri gruplandırması, eğitim, doğrulama ve deneme grupları.

### 2.3.2. Birinci Modelin Dezavantajları

Bir önceki kısımda tanımlanan modelin bazı limitleri ve onlardan kaynaklanan dezavantajları ortaya çıkıyor. Kısaca dezavantajları bu şekilde listeleyebiliriz:

- her bir veri noktası bir giriş nöronuna bağlandığı için aynı anda yapay sinir ağına uygulanabilecek giriş verileri çok sınırlı sayıda olacaktır. Örneğin giriş katmanına bir senelik (365 güne ait fiyat verileri) veri uygulamak istersek giriş katmanına 365 adet nöron yerleştirmemiz gerekecek ve bu sayı arttıkça sistemin performansını kötü yönde etkileyecektir. Nöron sayısı arttıkça eğitim süresi de artacaktır. Bunun yanında sistemin hafıza (Memory) tüketimi de yüksek artış gösterecektir. Dolayısı ile nöron sayısını istediğimiz kadar arttıramayacağımıza göre her zaman belli bir sınır sayısı kadar veri tek seferde network'e uygulayabileceğiz.
- Bu modelde uygulanan giriş bilgileri karşılığında çıkış (output) olarak ilerdeki sadece bir zaman biriminin tahmin çıktısını üretebileceği için pek kullanışlı olmayacaktır. Örneğin 30 günlük bir veri bu sisteme uygulandıktan sonra tek sefer çalıştırmada sistem sadece 31. günün fiyatı hakkında tahmin sonucu üretebilecektir.
- İleride daha fazla bir süre için tahmin sonuçları üretebilmek için üretilen tek zaman birimine ait çıkışları bir sonraki aşamada giriş olarak kullanmak gerekir. Yani aynen eğitim aşamasındaki kullanılan kaydırma (shifting) yöntemi ile ilerleyen zaman birimleri hakkında sistem sonuç üretebilecektir ve bu yöntem de sağlıklı bir tahmini veriler üretmeyecektir. Çünkü sistemin kaydırma yapıldıktan sonraki girişleri gerçek veriler değil ve bir önceki zaman birimleri için çıkışlarda oluşan tahmini veriler olacaktır.
- Eğitim (Training) aşaması, her bir zaman noktasına kaydırma yaparak tekrarlanacaktır. Ayrıca modelde kullanılan nöronlar fazla sayıda gerekli olabilir. Bu nedenler ile, eğitim aşaması uzun sürecektir.

### 2.3.3. İkinci Modelin Tanımı (Başarısız Bir Deneme)

Bu model bir alternatif model olarak Narx yapay sinir ağlarını kullanarak denendi fakat yüksek hafıza tüketimine ihtiyaç duyduğu için uygulaması başarısız kaldı. Bu model sadece alternatif bir yöntem olarak denenmiştir ve bu projede kullanılan yöntem değildir.

Bu modelde fiyat geçmişi verilerini aylık kümelere ayırmaya çalışıldı. Bu kümeler kesişimi olmayacak şekilde düzenlendi. Her küme için bir sonraki küme, bir sonraki aya ait verileri taşıyor. Dolayısı ile elde olan kümenin bir sonraki zaman dilimindeki durumu olarak eğitimde kullanılabilir.

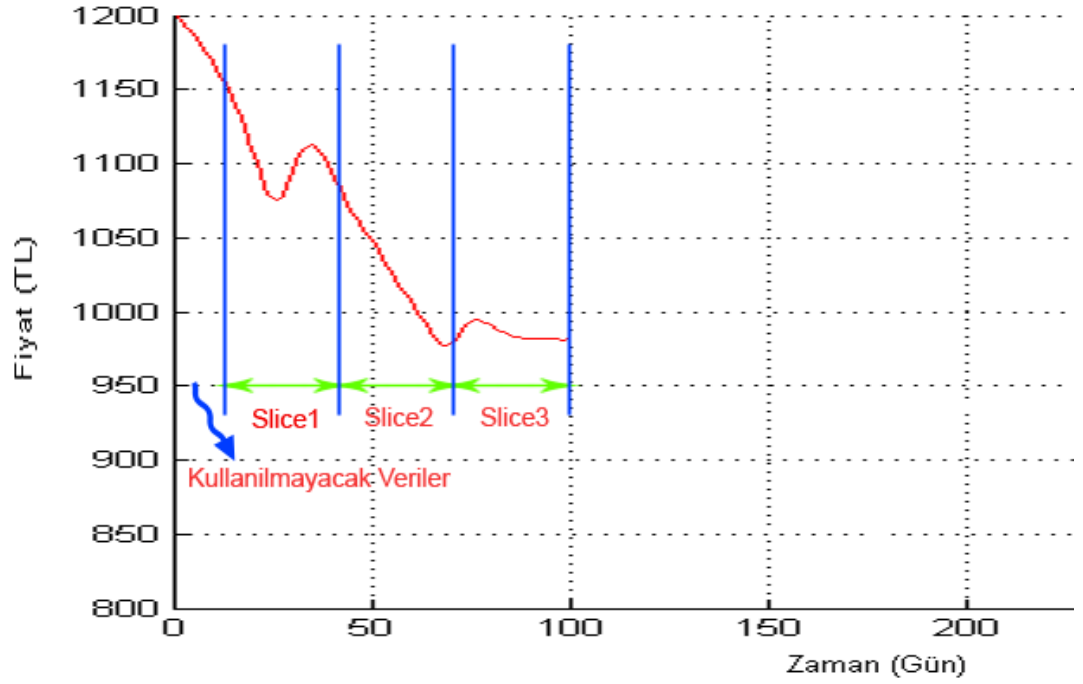
Veriler burada 30 günlük (aylık) dilimlere ayrıldı. Ekler bölümünün Ek-2 kısmında bu dilimleme (Slicing) işleminin nasıl yapıldığı Matlab ortamında yazılan kod ile gösterilmiştir. *dataCount* değişkeninde toplam elde olan veri adedi kayıtlıdır. Örneğin 300 günden oluşan bir fiyat geçmişi bilgisi olursa *dataCount* bu dizinin eleman sayısı yain 300 değerini alacaktır.

Fiyat veri bilgileri *prices* dizisine yüklenmiştir. *TimeSliceSize* değişkeni dilimlerin boyutunu belirliyor ki burada her dilim 30 günlük (1 aylık) veri içerecektir.

Dilimlere ayrılmış veriler *TimeSlices* matrisinde yüklenmiş olacak ve toplam oluşan dilim sayısı da *sliceCount* değişkeninde yer alacaktır.

Kod bloğunun başlangıcında yeterli sayıda verinin bulunması kontrol ediliyor. Minimum veri sayısı burada ( $3 * TimeSliceSize$ ) olarak belirlenmiştir. Yani her dilimi 30 gün (1 Aylık) olarak oluşturmak istersek, en az 90 günlük (3 Aylık) veri bilgisine ihtiyacımız olacaktır. Bu minimum 3 dilim sayısı bir önceki bölümde anlattığımız 3 veri gurubunu (Eğitim, doğrulama ve deneme grupları) oluşturmak için gerekli olacaktır.

Burada oluşturulan veri dilimlerinin boyutu eşit olmalıdır. Dolayısı ile ilgili kodda bir kontrol ile eğer yeterli sayıda veri yeni bir dilim oluşturmaya kalmadı ise döngü işlemine son veriliyor. Yani eğer toplam veri sayısını *TimeSliceSize* a bölersek, ve kalan R olursa, R sayıda veri kullanım dışı kalacaktır ve dilimlere dahil olmayacaktır.



Şekil 2.8 : TimeSlicing metodu (verileri eş boyutlu dilimlere ayırmak).

Bu modelin Matlab ortamında uygulaması yüksek hafıza tüketimi nedeni ile yapılamadı. Slicing metodu ile elde edilen matristeki bilgiler kullanılarak input ve target matrisleri oluşturuldu. Denemelerde yaklaşık 320 günlük veri için bu sistemi çalıştırdığımızda 10 adet aylık slice oluşuyor. Her ayın giriş vektörüne karşılık gelen target bir sonraki ayın verileridir. Bunu dikkate alarak eğitim aşaması için  $p$  (input) ve  $T$  (target) matrisleri alttaki kod bloğunda gösterildiği gibi üretilmiştir. Bu matrislerin boyutunun büyük olması nedeni ile eğitim aşamasında algoritmanın çalışabilmesi için yüksek memory ihtiyacı oluşuyor.

$p, t$  matrisleri hazırlandıktan sonra `newnrxsp()` fonksiyonu ile bir narx yapay sinir ağı oluşturulup eğitmek için `train` fonksiyonuna gönderiliyor.

#### 2.3.4. İkinci Modelin Dezavantajları

Bu modelin dezavantajlarını kısaca bu şekilde özetleyebiliriz:

- Yüksek memory tüketimi: Bu modelde elde olan veri sayısı arttıkça daha fazla memory'ye ihtiyaç oluşuyor. Ayrıca Bir sistemin işleme alabileceği

data miktarı bilgisayarın işlem yapma hızı ile sınırlıdır. dolayısı ile giriş verilerinin boyutu arttığı zaman yapay sinir ağını eğitme prosedürü hesaplama açısından maliyetli(expensive) bir işlem haline gelir [6].

- Training aşaması her grup (slice) için tekrarlanacağı için uzun sürecektir. Verilerin sayısının artması ile de bu zaman artış gösterecektir.
- Sağlıklı bir şekilde training işlemi gerçekleşmeyebilir. Dilimlerin bir biri ile alakasız iniş çıkışları training aşamasını zorlaştırıp performansın yakalanmasını büyük bir oranda azaltacaktır.
- Dilimlere ayırma yönteminde elde olan bilgilerin bir kısmı dilimlere dahil olmayıp kullanım dışı kalacaktır ( eğer toplam veri sayısını TimeSliceSize a bölersek, ve kalan R olursa, R sayıda veri kullanım dışı kalacaktır).

Bir sonraki kısımda bu projede kullanılan Narx yapay sinir ağlarının yapısı kısaca özetlenmiştir.

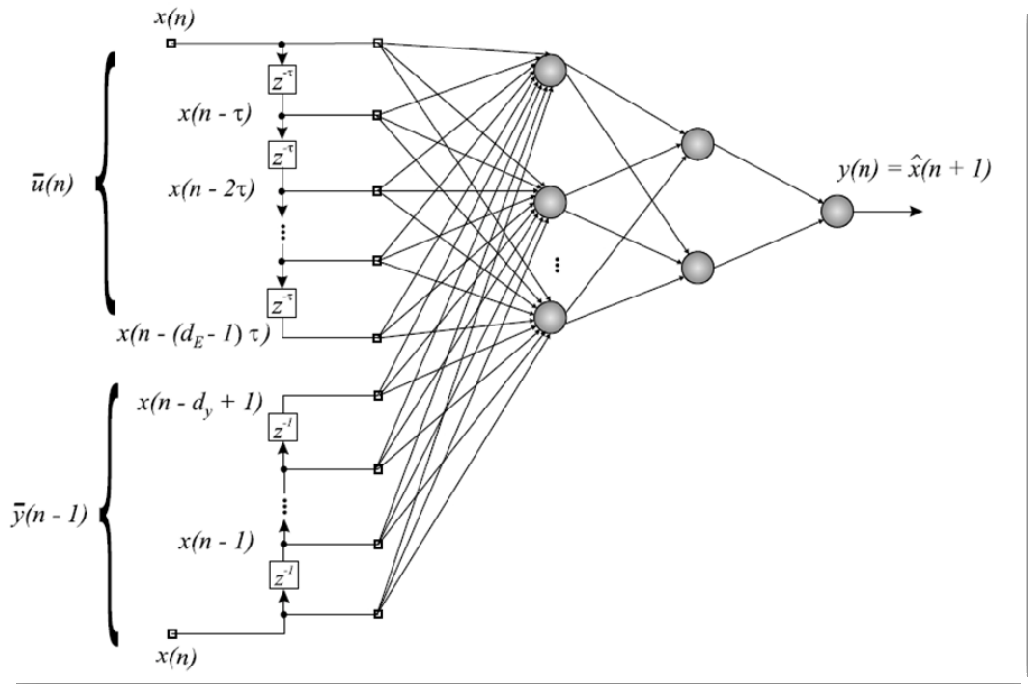
## **2.4. NARX YAPAY SİNİR AĞLARI**

Bu projede kullanılan yapay sinir ağ modelinden bahsetmeden önce Narx yapay sinir ağlarına kısaca bir göz atalım.

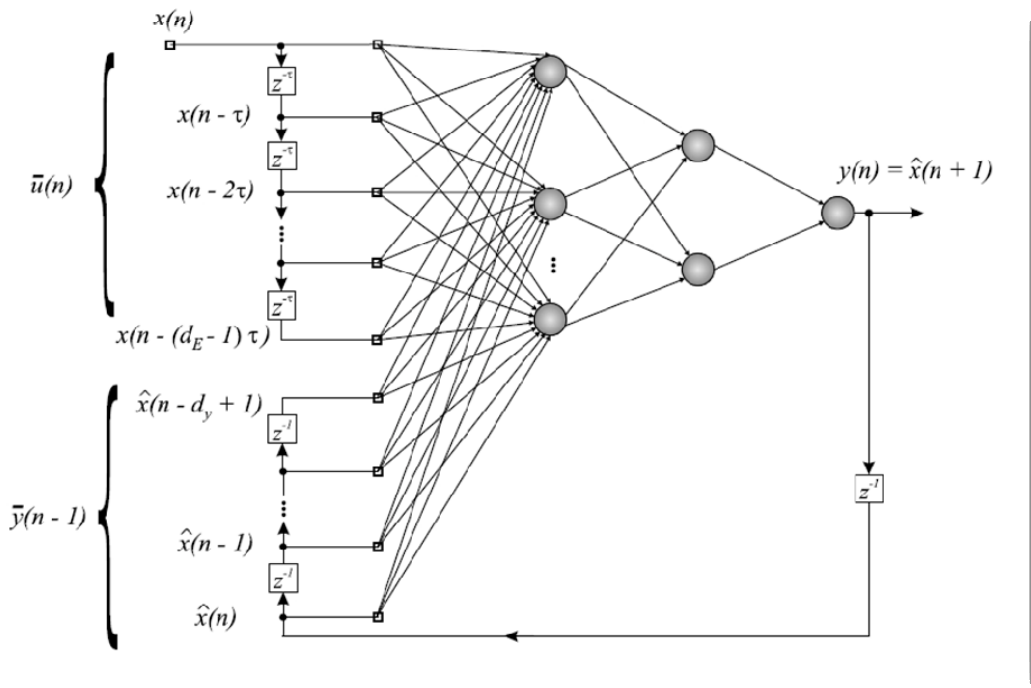
Narx bir dinamik yapay sinir ağ modelidir ve genelde linear olmayan sistemlerde giriş-çıkışların modellenmesinde kullanılır [5]. Time Series tahminleri için feedforward time delay şeklinde tasarlanıp kullanılabilir.

Narx modelinde 2 çeşit feedback kullanımı mevcuttur:

- Series-Parallel (SP) Mode:  
çıkışın geri dönüşümü (regressor) sistemin gerçek çıkış değerlerinden oluşmaktadır.
- Parallel (P) Mode:  
çıkışın geri dönüşümü (regressor) sistemin tahmini çıkış değerlerinden oluşmaktadır.



Şekil 2.9 : Narx ANN model: SP –Mode [5]



Şekil 2.10 : Narx ANN model: P –Mode [5]

### **3. MALZEME VE YÖNTEM**

Bu kısımda ürünlerin fiyatlarını ileleyen günlerde tahmin etmek için yapay sinir ağlarını kullanacağız. Çalışmanın devamında fiyat geçmişi elimizde olan örnek bir ürün üzerinde çalışacağız. Ürünün fiyat TL biriminde geçen X günde elimizdedir. Bu fiyat son tüketiciye alışveriş için sunulan fiyattır ve zaman içerisinde değişimlerinde piyasa fiyatlandırmanın yanı sıra kur değişiklikleri ve enflasyon oranları da etkin olmuştur. Dolayısı ile kur veya enflasyon oranını başka bir etken olarak bu tahminlere dahil etmek doğru olmayacaktır ve bu analizi, bütün etkenlerin bu son fiyat üzerinde bıraktığı etkiyi bir bütün olarak ele almamız gerekmektedir.

#### **3.1. FİYAT GEÇMİŞİ VERİ FORMATI VE DÜZENLEMELER**

##### **3.1.1. Fiyat Geçmiş Veri Formatı**

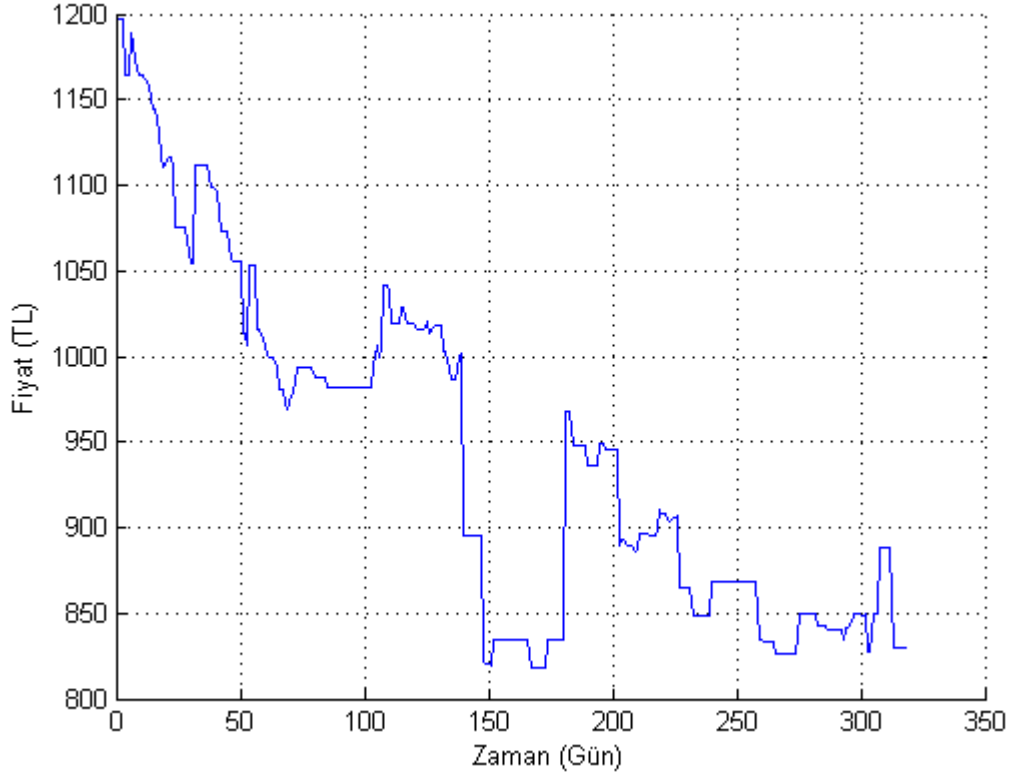
Bu projede fiyat verileri herhangi bir örnek ürün için TL fiyatı olarak veri dosyasından veya mysql veritabanından elde edilecektir. Ürünün günlük fiyatları virgül ile birbirinden ayrılmış şekilde bir karakter dizisi olarak kayıtlıdır.

Ör: 110,110,125,150,150,140,140,150,150,150

Bu projede dizide fiyat elemanları ters sırada yer almaktadır. Yani dizinin ilk elemanı son güne ait fiyatı ve bu sıra ile dizinin son elemanı ilk kayıt alınan günün fiyatını göstermektedir. Dolayısıyla zaman eksenine verileri yerleştirebilmek için önce dizinin eleman sıralamalarını ters halinden çıkarıyoruz ve zaman içerisinde ilk günden bu güne değişim gösteren bir değişken haline getiriyoruz.

Dizide hata veya günlük kayıt boşluğunu gösteren sıfır elemanlarını bir fonksiyonla temizliyoruz. Bu düzenlemeyi sıfırları onlardan bir önceki günün fiyatı ile değiştirebiliriz.

Şekil 3.1’de örnek bir ürün için yaklaşık olarak 320 günlük veri zaman serisi halinde gösterilmektedir.



Şekil 3.1: Örnek veri grafiği

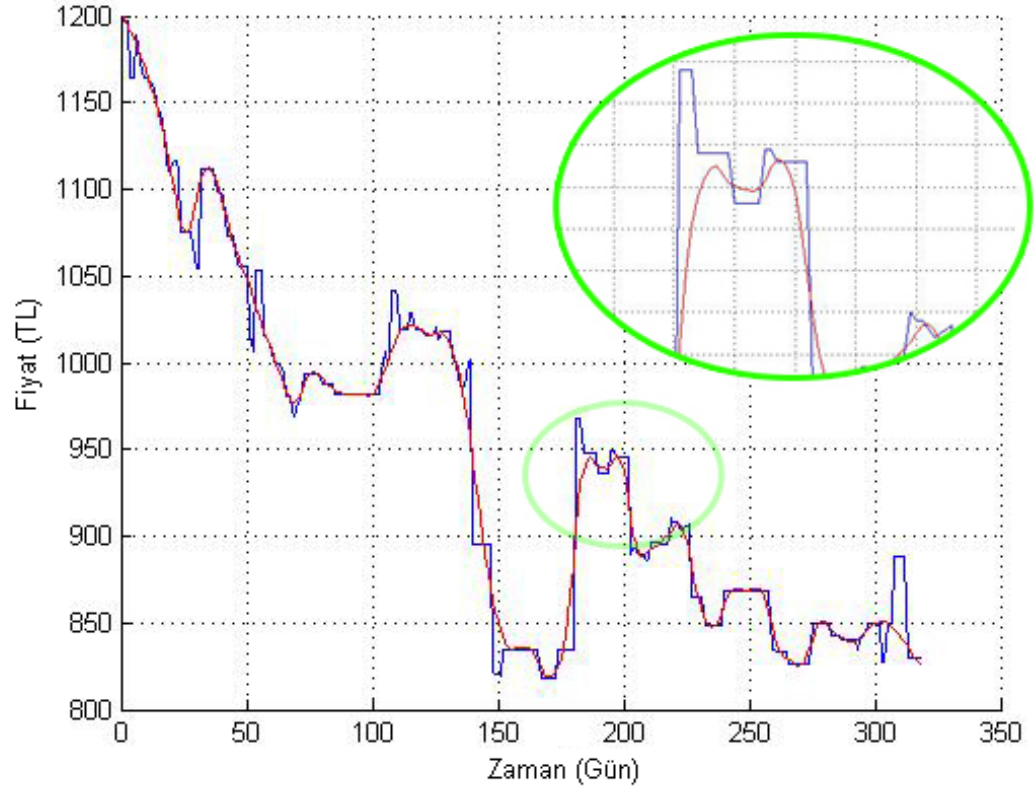
### 3.1.2. Verileri Düzleştirmek (Smoothing)

Verileri yapay sinir ağına giriş olarak kullanmadan önce düzleştirmemizde (smoothing) fayda vardır. Bu işlem yapay sinir ağını daha sağlıklı bir şekilde eğitmemize yardımcı olacaktır.

Bunu giriş verileri üzerinde bir çeşit noise giderme de diyebiliriz. Matlab programında dizi üzerinde smooth() fonksiyonu uygulanarak bu işlem gerçekleştirilmiştir.

Bir sonraki şekilde örnek verilerin üzerinde smoothing işleminin sonucu gösterilmiştir.

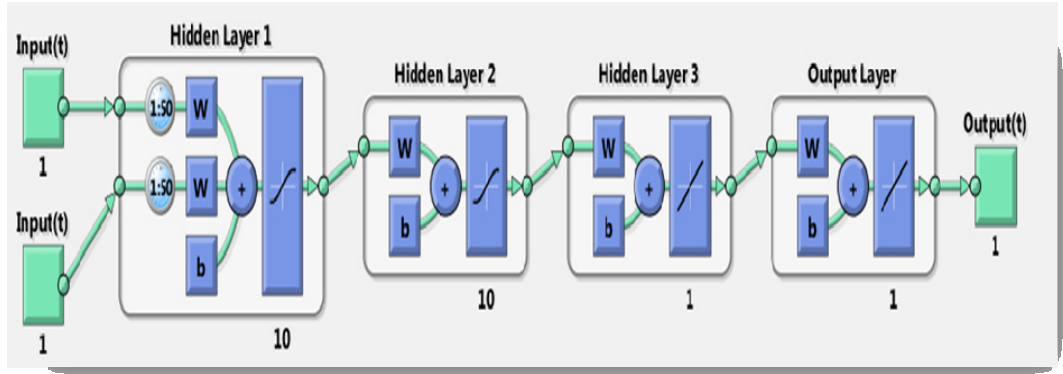




Şekil 3.2: Örnek veriler üzerinde düzleştirme işleminin sonucu.

### 3.2. YAPAY SİNİR AĞ MODELLEME

Bu projede fiyat tahmini için kullandığımız model 3 gizli katmandan (hidden layer) oluşan bir sp-mod narx yapay sinir ağıdır. Bu modelde 1. Ve 2. Gizli katmanda her birinde 10 nöron ve 3. gizli katmanda 1 nöron kullanılmıştır.



Şekil 3.3 : Kullanılan yapay sinir ağ modeli (bu şema Matlab programı ile çıkarılmıştır).

#### 3.2.1. Giriş ve Çıktılar ve Çalışma Mantığı

Belirli bir döneme ait fiyat geçmişini ele alarak, ilerdeki N gün hakkında fiyat tahmininde bulunmak için giriş ve çıkış vektörleri örnekleme yaparak bu şekilde anlatabiliriz. örnek olarak 300 günlük fiyat geçmişi ile ilerleyen 30 gün hakkında bir tahmin yürütmek istiyoruz. İlk 270 günün verilerini (1-270. güne ait veriler) seri halinde, giriş vektörü olarak network'a uyguluyoruz. Sistemden beklenen çıkış, ilerleyen 270-300. güne ait, yani 30 günlük veri olacaktır. Ama bu çıkış vektörünü 270 günlük olarak sistemden çıkmasını ayarlıyoruz yani 30. günden 300. güne kadar veriler uygun çıkış olarak sistemden beklenmektedir. Sistemi eğitmek için bu örnek çıkış vektörü eğitim verisi (Target) olarak kullanılabilir.

Fiyat geçmişi olarak elde bulunan günlük veri sayısı: dataCount

Tahminde bulunmak istediğimiz gün sayısı: N

Input= priceArray(1: dataCount-N)

Target= priceArray(N+1: dataCount)

Oluşturulan network bu Input ve Target ile eğitildikten sonra, Target vektörü bir sonraki aşamada Input olarak Network'e uygulanıyor ve elimizde olmayan ilerleyen N günü içinde bulunduran vektörü, üretmek için kullanılıyor:

Input= priceArray(N+1: dataCount)

Output= PredictedPriceArray(2\*N+1 : dataCount+N)

Örnek olarak: N=30;

Input= priceArray(31: 300)

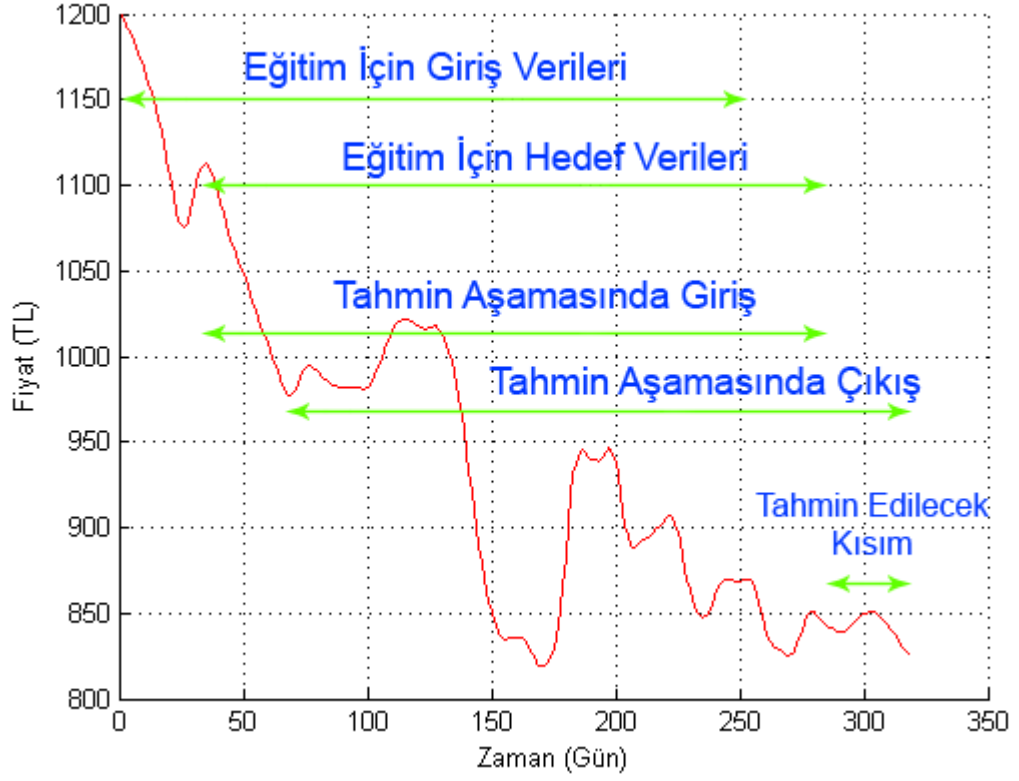
Output= PredictedPriceArray(61 : 330)

Not: yukarıdaki denklemlerde kullanılan vektör tanımlama ifadeleri Matlab programının ortamında geçerli olan kurallara tabidir.

Bu projede tahmin edilen verilerin gerçeğe yakınlığını ölçebilmek için, elde olan veri sayısını dataCount-N günlük farz edip Input ve Target'i ona göre ayarlıyoruz. Zaman dizisinde son N günün verilerini Network'a uygulamadan üretilen tahmini sonuçları son N günün gerçek fiyatları ile bir grafikte karşılaştırıyoruz.

Yapay sinir ağının ürettiği çıkış (output) vektörü üzerinde yine smooth() fonksiyonunu uygulayarak verileri düzleştiriyoruz.

Altaki grafikte (Şekil 3.4) modelin giriş çıkış vektörleri ve eğitmek için kullanılan giriş ve hedef vektörleri anlatılmıştır.

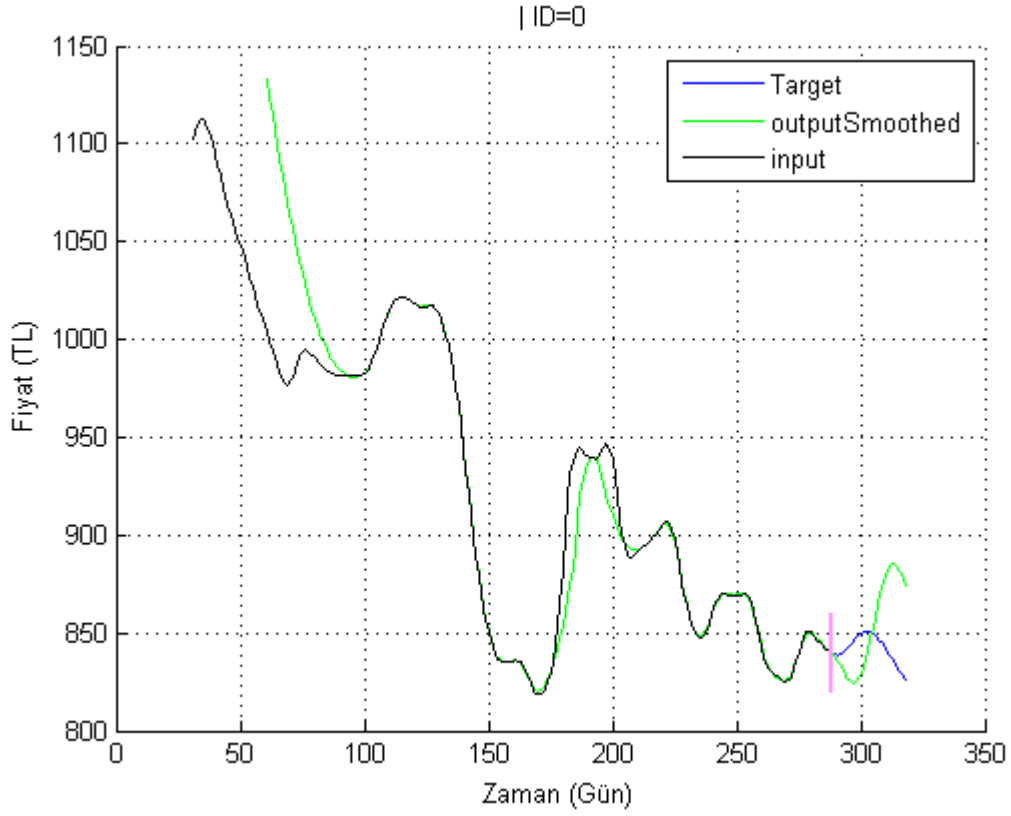


Şekil 3.4: Kullanılan ağ modelini eğitmek için kullanılan vektörler ve fiyat tahmin çıkışı gösterilmektedir.

Şekil 5.6’da gösterilen grafikte tahmin süresi  $N=30$  gün olarak farz edilmiştir. Desired Prediction başlığı ile belirtilen aralık, hakkında tahmin yürütmek istediğimiz son 30 günü belirtmektedir. Yapay sinir ağını şekilde gösterilen verilerle eğittikten sonra Prediction Input aralığında bulunan verileri giriş olarak kullandıktan sonra Prediction Output aralığı için networkten çıkış alınacak. Bu çıkış vektörünün son 30 güne ait verisi tahmin edilen 30 günlük fiyat olarak, Desired Prediction aralığında olan gerçek fiyatlarla kıyaslanabilir.

Bu mantığı kullanarak Şekil 3.4’te gösterilen veriler için programın ürettiği sonuç Şekil 3.5’te gösterilmektedir. Bu grafikte network’a verilen giriş vektörü siyah renkle gösterilmiştir. Beklenen gerçek sonuç ise, giriş vektöründe bulunmayan bir sonraki 30 günün gerçek fiyatını mavi renkle göstermektedir. Ki bu aralık daha önce anlatılan prediction veya tahmin aralığını belirtmektedir. Bu aralıkta bulunan mavi çizgiye yakın

yeşil çizgi ise sistemin bu 30 gün için tahmin ettiği fiyatları göstermektedir. Tahmin aralığının başlangıcı penbe rekte olan bir dikey çizgi ile grafikte işaretlenmiştir.



Şekil 3.5 : Örnek olarak Şekil 3.4'te gösterilen fiyat verileri için sistemin ürettiği sonuç.

Bulgular bölümünde, 30 adet ürünün fiyat geçmişi verilerine bu yapay sinir ağ modeli uygulanıp sonuçları grafiklerle incelenmiştir.

### **3.3. MATLAB İLE GELİŞTİRİLEN UYGULAMANIN ANLATIMI**

Bu bölümde projenin MATLAB ortamında uygulaması anlatılacaktır. Uygulamanın kaynak kodu ekler bölümünde yerleştirilmiştir.

Bu uygulamada veriler mysql veri tabanından çekilmektedir. Dolayısı ile uygulamayı çalıştırabilmek için local'de sisteme mysql servisi kurulmasının yapılması gerekmektedir.

Matlab'den Mysql'e bağlanabilmek için yapılması gereken işlemler ve ayarlar ekler kısmında "Matlab'den Mysql'e bağlanmak" bölümünde anlatılmıştır.

#### **3.3.1. Veri Tabanı Yapısı**

Kullanılacak olan veritabanında 2 tablo oluşturulmuştur. Birincisi "product" tablosu. Bu tabloda ürün IDsi ve ismi bulunmaktadır. Diğer tablo "productspricehistory" ismi ile ürünlerin fiyat geçmişini içermektedir. Bir önceki tablo ile productID alanı ile eşleşen bu tabloda fiyat geçmişi text formatında günlük fiyatları virgül ile ayrılmış şekilde barındırmaktadır. Bu tabloda fiyat geçmişlerinin hangi tarihe ait olduğunu anlayabilmek için "lastModifyDate" alanı en son güncelleme tarihini içerecek şekilde ayarlanmıştır. örnek olarak bu tarih alanında 2011-01-02 yazılı ise son fiyat kaydı 2011-01-02 gününde alınmıştır demektir ve bir önceki fiyat kaydı da doğal olarak 2011-01-01 gününe ait olmalıdır. Bu yapı tablonun boyutunun fazla büyümemesi için bu şekilde ayarlanmıştır. aksi durumda her gün için tarih ve o güne ait fiyat satırı tablyla eklendiği takdirde, kayıtlar arttığı zaman gereksiz bir yunluk tabloda oluşacaktır.

Bu tabloda günlük fiyatlar, minimum ve maximum fiyat olarak her ürün için kaydedilmiştir. Bu projede minimum price alanında olan veriler kullanılmıştır. Alternatif olarak maximum ve minimum fiyat ortalaması da alınarak kullanılabilir.

Table: product

	productID	productNameTitle
<input type="checkbox"/>	73569	SONY ERICSSON W910 RED CEP TELEFONU
<input type="checkbox"/>	73571	SONY SONY CYBERSHOT DSC W210P 12.1 MP Pembe Dijital Fotograf Makinesi
<input type="checkbox"/>	73572	FUJI FINEPIX J120 FOTOĞRAF MAKİNESİ
<input type="checkbox"/>	73573	PHILIPS HTS 3270/12 EV SİNEMA SİSTEMİ
<input type="checkbox"/>	73574	PHILIPS HTS 3377/12 EV SİNEMA SİSTEMİ
<input type="checkbox"/>	104788	JVC DLA-RS4000 Projeksiyon
<input type="checkbox"/>	104789	Epson EB-X7 Projeksiyon
<input type="checkbox"/>	104790	Epson EB-W8/D Projeksiyon
<input type="checkbox"/>	104791	Epson EB-W7 Projeksiyon
<input type="checkbox"/>	73580	GOLDMASTER KLC 843 MİNİ LCD TV
<input type="checkbox"/>	73581	PHILIPS 22PFL3404 LCD TV
<input type="checkbox"/>	73582	DELL 1545 64H3 T6400 3GB 250GB VHB 15.6 Notebook
<input type="checkbox"/>	73584	DELL XPS 1330 T6400 2GB 250GB VHB BLACK 13.3 Notebook
<input type="checkbox"/>	78747	Praktica DCZ 5.7 Dijital Fotoğraf Makinesi
<input type="checkbox"/>	73587	ASUS N51VG240DV T9550 4GB 320GB 15.6 512VGA Notebook
<input type="checkbox"/>	73588	ASUS X61SV-240DV Notebook
<input type="checkbox"/>	73589	XEROX 3300MFP/DN NW PR.SCAN.FOTO.FAX
<input type="checkbox"/>	104785	Epson EH-DM30 Projeksiyon

Şekil 3.6: Veri tabanında ürün tablosu (product table)

Table: productspricehistory

productID	minPrice_history	maxPrice_history	lastModifiyDate
185160	332,336,336,336,336,336,345,345,345,336,336,336,33...	481,360,360,360,360,481,481,481,481,481,362,362,36...	2011-02-03 02:54:03
185161	524,531,531,531,531,531,544,544,544,531,531,531,53...	761,569,569,569,569,761,761,761,761,761,571,571,57...	2011-02-03 02:54:03
185162	370,375,375,375,375,375,385,385,385,375,375,375,37...	537,402,402,402,402,537,537,537,537,537,403,403,40...	2011-02-03 02:54:03
185163	505,512,512,512,512,512,524,524,524,512,512,512,51...	733,548,548,548,548,733,733,733,733,733,550,550,55...	2011-02-03 02:54:03
185164	185,187,187,187,187,187,187,187,187,187,187,18...	201,201,201,201,201,201,201,201,201,187,187,187,18...	2011-02-03 02:54:03
185165	265,265,265,265,265,265,265,265,265,265,265,26...	370,278,278,278,278,370,370,370,370,370,278,278,27...	2011-02-03 02:54:03
185166	319,323,323,323,323,323,331,331,331,323,323,323,32...	463,347,347,347,347,463,463,463,463,463,348,348,34...	2011-02-03 02:54:03
185167	365,365,365,365,365,365,365,365,365,365,365,36...	510,381,381,381,381,510,510,510,510,510,382,382,38...	2011-02-03 02:54:03
185168	460,466,466,466,466,466,668,668,668,466,466,466,46...	668,466,466,466,466,668,668,668,668,668,466,466,46...	2011-02-03 02:54:03
185169	478,478,478,478,478,478,478,478,478,478,478,47...	668,499,499,499,499,668,668,668,668,668,501,501,50...	2011-02-03 02:54:03
185170	232,232,232,232,232,232,232,232,232,232,232,23...	323,243,243,243,243,323,323,323,323,323,243,243,24...	2011-02-03 02:54:03
185171	191,194,194,194,194,194,198,198,198,194,194,19...	277,208,208,208,208,277,277,277,277,277,208,208,20...	2011-02-03 02:54:03

Şekil 3.7: Veri tabanında fiyat geçmişi tablosu (productsPriceHistory Table)

### 3.3.2. Fiyat Verisini Yükleme

Programda fiyat geçmişi sorgulaması için kullanıcıdan bir ürünID'sini girmesi isteniliyor. ÜrünID'si boş bırakılırsa testlerde kullandığımız default test verisi prices.mat dosyasından yükleniyor. Kullanıcı tarafından girişi yapılan productID'ye sahip ürünün fiyat geçmişi bir karakter dizisi (string) olarak prices\_str değişkenine yükleniyor.

Bu aşamada prices\_str değişkenine yüklenmiş olan verinin üzerinde çalışabilmek için önce texti (karakter dizisini) okuyup sayılardan oluşan bir matris haline getirmemiz gerekiyor. Bu işlem strread() fonksiyonu ile yapılabilir ve ardından elde edilen bir satırlık matrisi sütun haline çevirmek için transpose operatörü (') kullanılabilir.

Daha önce belirttiğimiz gibi günlere ait fiyat verileri ters sıra ile kayıtlı olduğu için kullanıma hazırlamak için diziyi flip() fonksiyonu uygulanarak dizi ters sırada döndürülüp düzgün hali elde ediliyor.

İşlemlere devam etmek toplam elde ettiğimiz zaman serisi verilerinin sayısını, dizideki elemanları saydırarak çıkarmak gerekiyor. Bu işlem Matlab'de length() fonksiyonu ile yapılabilir.

### 3.3.3. Verileri Kullanım İçin Temizleme ve Düzleştirme

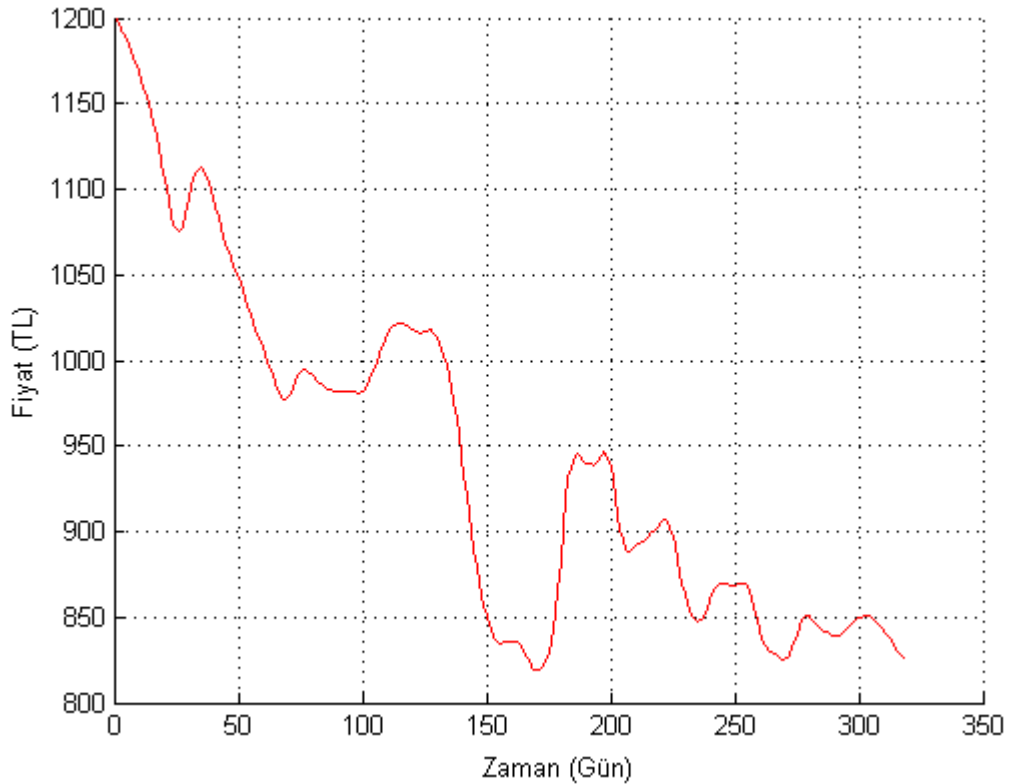
Bu aşamada çekilen verileri kullanım için uygun hale getirmemiz gerekiyor. Verilerin arasında sıfır gibi fiyat kayıtları varsa onları temizlemek gerekir. Sıfır fiyat, ilgili günde her hangi bir sebeple fiyat kaydının yapılmadığını ve elde edilmediğini gösteriyor. Bu sıfır elemanlar günlük fiyatların alınmaması durumunda diğer fiyatların tarihinin bozulmaması için gereklidir. Ama bu aşamada programda sıfır olarak kullanılırsa yapay sinir ağını tamamen yanıltacaktır. Bunu önlemek için en iyi çözüm olarak, sıfır fiyat kayıtları, bir önceki günün fiyat değerleri ile revize edilmiştir. Bu işi yapmak için Matlab ortamında reviseData() fonksiyonu yazılıp, ilave edildi. Bu fonksiyonun kaynak kodu Ekler bölümüne konulmuştur. Fiyat dizisi bu fonksiyona gönderilerek sıfır elemanlardan temizleniyor.



Bu aşamada dizi üzerinde yapılması gereken başka bir değişiklik, verileri düzleştirmek olacaktır. Başka bir tabir ile grafikte görünen keskin geçişleri yuvarlak hale getirmek gerekiyor. Bu işlem yapay sinir ağının eğitim aşamasında performansını yükseltmeye yardımcı olur.

Düzleştirmeği uygulamak için Matlab'de `smooth()` fonksiyonu kullanılmıştır. Smoothing işleminde düzleştirme oranını belirlemek için `span` değişkeni kullanılmıştır. Bu değişken tek sayı olmalıdır. Bu programda `span` 19 olarak tanımlanmıştır. Duruma göre uygun sonuç elde edilemez ise bu oran değiştirilebilir. Ayrıca `smooth()` en son parametresi de uygulanacak olan smooth algoritma türünü belirtiyor. Bu uygulamanın amacına uygun opsiyon `rloess` olarak seçilmiştir. Matlab dökümanında bu parametrenin diğer opsiyonlarını (`lowess`, `moving`, `rlowess` gibi) görebilir ve uygulama sonuçlarını deneyebilirsiniz.

Veriler anlatılan şekilde revize edildikten sonra alttaki şekilde gösterilen örnekteki gibi bir zaman serisi hazırlanıyor.



Şekil 3.8 : Revize edilen verilerin örnek grafiği

### 3.3.4. Yapay Sinir Ağını Oluşturmak

Yapay sinir ağını oluşturmada gerekli olan maximum ve minimum fiyat sınırlarını belirlemek gerekecek. Burada dizide bulunan en yüksek fiyatın %30 üstü, üst limit ve en düşük fiyatın %30 altı, alt fiyat limiti olarak tanımlanmıştır:

```
minRange=round(min(prices)*0.70)
maxRange=round(max(prices)*1.30)
```

Bu projede ileriye dönük fiyat tahmini bir aylık (30 günlük) olarak yapılmıştır. Dolayısı ile *delay* adında bir değişken oluşturup 30 değerini ona kaydediyoruz. Yapay sinir ağına giriş verisi olarak kullanılacak verileri oluşturmak için *data\_prior* ve *data\_delayed* dizilerini de şu şekilde tanımlıyoruz:

```
delay=30;
data_prior    = prices(1:(end-delay))
data_delayed  = prices((delay+1):end)
```

Diğer anlatımda eşit boyutta olan iki dizi tanımlanıyor ki *data\_prior* dizisinde son 30 günün bilgileri mevcut değil ve *data\_delayed* dizisinde ilk 30 günün verileri mevcut değil ve verileri 30 gün kaydırarak 31. günden son güne kadar fiyat bilgisini içermektedir.

Yapay sinir ağı modelini oluşturmak için giriş (Input) ve hedef (Target) vektörlerini şu şekilde tanımlanmıştır:

```
p = data_prior(1:(end-delay)) % training input
T = data_delayed(1:(end-delay)) % training target
```

Diğer anlatımda *p* vektöründe son 60 gün (yani  $2*delay=2*30$ ) günün verisi yoktur ve *T* vektöründe ilk 30 gün ve son 30 günün verileri yoktur.

Bunu bu formatta yapmamızın sebebi son 30 günün verilerini hiç yokmuş gibi target vektöründe yapay sinir ağına vermemek ve elimizde olan son 30 günün gerçek verileri

ile real bir sonuç karşılaştırması yapabilmektir. Gerçek bir projede son 30 günün verileri de Target vektörüne dahil edilebilir şekilde uygulamanın bu kısmı revize edilmelidir.

Bu aşamada yapay sinir ağını oluşturmaya geliyoruz. Bu proje için düşünülen ANN modeli 1 giriş katmanı 3 gizli katman ve 1 çıkış katmanından oluşmaktadır. Birinci ve ikinci gizli katmanın aktivasyon fonksiyonu tangent sigmoid (*tansig*) ve 3. gizli katmanın fonksiyonu pureline (*purelin*) olarak belirlenmiştir.

Matlab ortamında *newnarxsp()* fonksiyonu ile yapay sinir ağı oluşturulmuştur:

```
d1 = [1:delay]; d2 = [1:delay];
narx_net = newnarxsp(p,T,d1,d2
    , [10 10 1], {'tansig', 'tansig', 'purelin'});
```

Bu şekilde daha önce anlatımı yapılan bir Series-Parallel (SP) Mode Narx yapay sinir ağı oluşturuluyor. İlk 2 gizli katmanda 10 adet nöron ve 3. gizli katmanda 1 adet nöron kullanılmıştır. *d1* ve *d2* vektörleri gecikme vektörü olarak tanımlanmıştır. *d1* giriş gecikmesi vektörü (Input delay vector) ve *d2* çıkış gecikme vektörü (Output delay vector) olarak belirtilir. Bu gecikme vektörlerinde sıfırdan büyük pozitif değerler kullanılır ve bu vektörler çıkış elemanlarının girişten ne kadar gecikmeli olduklarını gösterir.

### 3.3.5. Yapay Sinir Ağını Eğitmek

Uygulamada performans hedefi (*narx\_net.trainParam.goal*) 0.01 olarak tanımlanmış ve maximum eğitim döngü sayısı (*narx\_net.trainParam.epochs*) 500 olarak belirlenmiştir. Eğer eğitim aşamasında istenilen performans hedefine 500 döngü tekrarı sonucunda ulaşılamazsa ağın eğitimine son verilip program eğitim döngüsünden çıkıyor.

Bir önceki bölümde Narx ağını oluşturmak için kullanılan *p* giriş ve *T* hedef vektörleri bu aşamada ağı eğitmek için kullanılacaktır. Bu vektörler ağ oluşturma fonksiyonunda sadece giriş ve çıkış vektörlerin boyutunu belirlemek için kullanılır.

Yapay sinir ağını eğitmek için Matlab ortamında *train()* fonksiyonunu kullanabiliriz. Bu fonksiyonun kullanım şekline bir bakalım:

```
net=train(net,P,T,Pi,Ai)
```

Oluşturulan network objesi giriş vektörü ve hedef (target) vektörünü parametre olarak alıyor. *Pi* ve *Ai* opsiyonel parametre olarak kullanılabilir. *Pi* başlangıç için giriş gecikme değerlerini alabilir. Belirlenmediği takdirde default olarak bu değerler sıfır olarak alınır. *Ai* başlangıç için katmanın gecikme değerleri olarak belirlenebilir ki belirlenmediği takdirde default olarak bu değerler sıfır olarak alınır. Bu uygulamada sadece *Pi* parametresi altta gösterildiği gibi tanımlanıp kullanılmıştır:

```
for k=1:delay,  
    Pi{1,k}=data_prior{k};  
    Pi{2,k}=data_delayed{k};  
End
```

Böylece *Pi* değişkeni 2 satırlık bir matris olarak oluşuyor. Birinci satırında *data\_prior*'da olan ilk 30 (*delay*) günün verileri ve ikinci satırında *data\_delayed*'de olan ilk 30 (*delay*) günün verileri yerleştiriliyor.

Yapay sinir ağını eğitmek için *train()* fonksiyonuna alttaki parametreler verilmiştir:

```
narx_net = train(narx_net,[p;p],T,Pi);
```

Train parametresinde input olarak 2 satırlık bir matris verilmiştir. Her iki satırına da hazırladığımız *p* ardışık dizisi yerleştirilmiştir. İkinci satır gecikmeli geri dönüşüm (delayed) için paralel Narx ağlarında kullanılabilir ama kullandığımız serial-paralel modelinde ikinci satıra target vektörü yerine input vektörünün aynısı yerleştirilmiştir. Diğer kullanım amacımıza uygun sonuçlar vermemektedir.

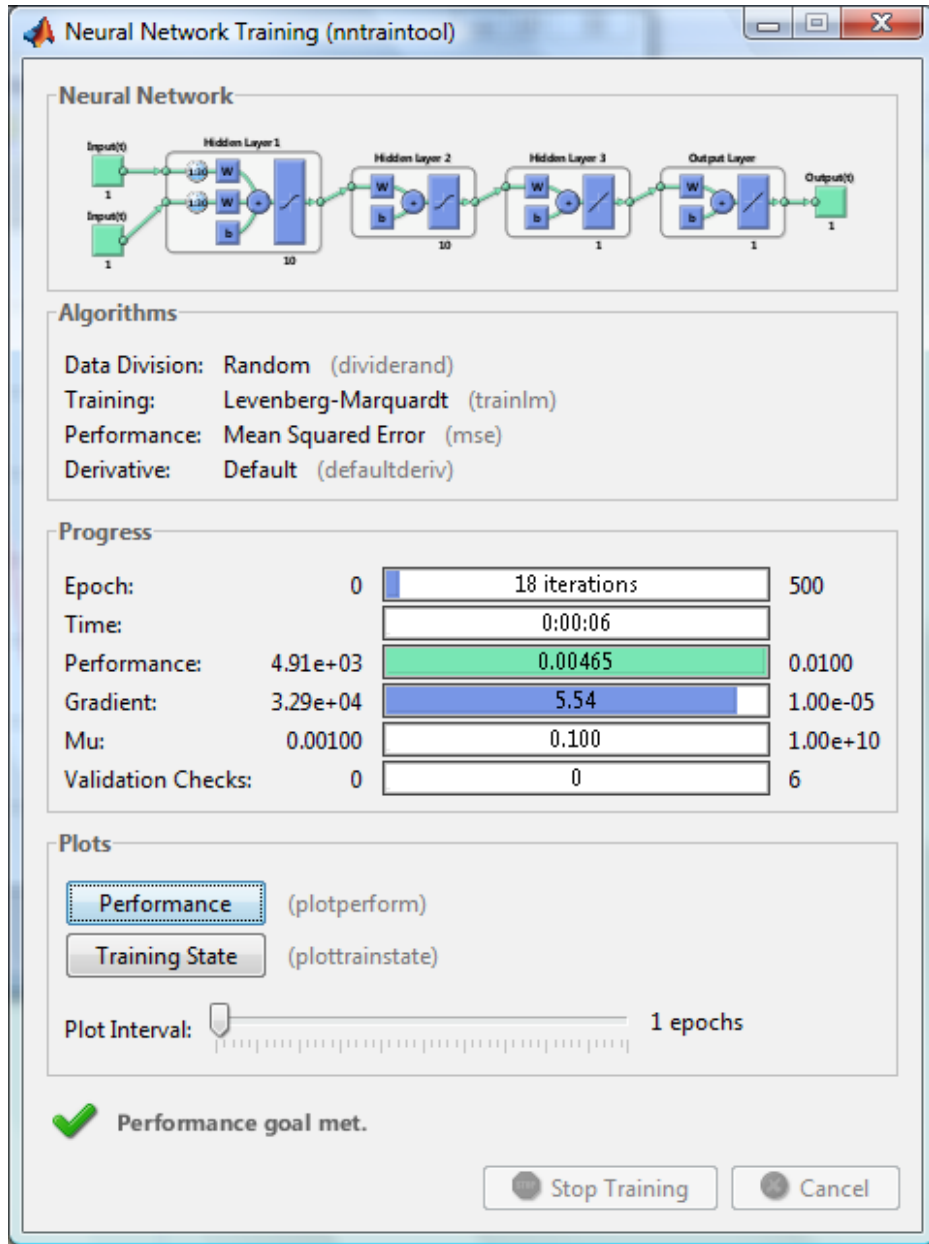
Bu aşamada yapay sinir ağı eğitim döngüsüne giriyor ve performansı yakalayana kadar veya maximum epoch limitini doldurana kadar eğitim döngüsü devam ediyor. Bir sonraki şekilde Matlab ortamında bu aşamada gösterilen ve eğitim bilgilerini içeren, kullanıcı arayüzü gösterilmiştir. Gösterilmiş olan örnekte 18 eğitim döngüsünde istenilen performans elde edilmiştir.

Eğitim aşaması bittikten sonra *p* girişini ağa uyguladığımız zaman gerçek hedef (target) ve yapay sinir ağının ürettiği cevabın farkını kontrol edebiliriz. Bu işlemi hedef ve çıkış

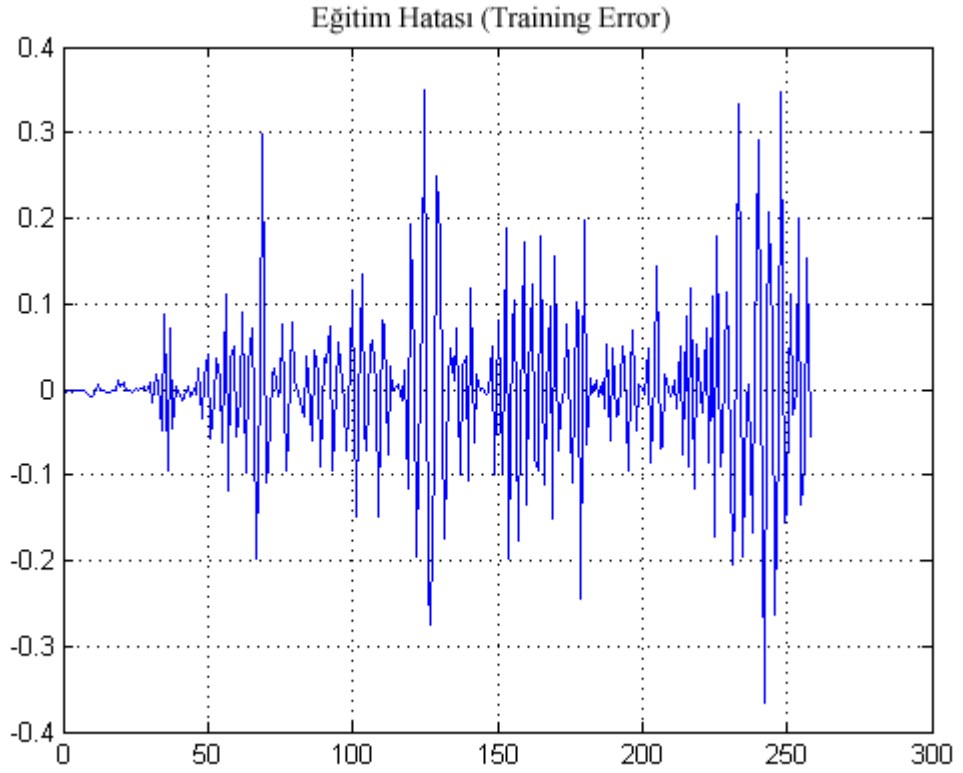
vektörlerini birbirinden çıkararak yapabiliriz. *sim()* fonksiyonu ile yapay sinir ağına giriş verileri uygulayıp çıkış alabiliriz.

```
output1 = sim(narx_net, [p;p], Pi);  
error = cell2mat(output1)-cell2mat(T);
```

Yukarıdaki hata hesaplamasında kullanılan *cell2mat()* fonksiyonu output değişkenini hücre türünden matris formatına çevirmek için kullanılmıştır.



Şekil 3.9 : Yapay sinir ağı eğitim aşaması (Bu şema Matlab programından alınmıştır).



Şekil 3.10 : Yapay sinir ağının eğitim sonrası hata grafiği.

### 3.3.6. Tahmin Aşaması (Prediction Phase)

Bu kısımda tahmin aşamasına gelmiş oluyoruz. Tahmin için giriş vektörünü ve amaç olan gerçek hedef vektörlerini şu şekilde oluşturuluyor:

```
prediction_input =data_prior(delay+1:end)
prediction_target=data_delayed(delay+1:end)
```

Eğitim aşamasında kullanılan input ve target vektörlerini *delay* (burada *delay*=30 gün) kadar ileriye kaydığımız zaman tahmin için giriş ve hedef vektörlerini elde ediyoruz.

*sim()* fonksiyonu için opsiyonel parametre olan *Pil* matrisi de başlangıçta girişin gecikme koşullarını (Initial input delay conditions) belirtmek için kullanılabilir. Bu

uygulamada giriş ve hedef vektörlerinin ilk 30 (*delay*) günlük elemanı bu matrise yüklenmiştir:

```
for k=1:delay,
    P11{1,k}=prediction_input{k};
    P11{2,k}=prediction_target{k};
end
```

Bu aşamada `sim()` fonksiyonunu kullanarak networke uygulanan veriden çıkış elde edebiliriz:

```
prediction_output =
sim(narx_net,[prediction_input;prediction_input],P11);
```

Elde edilen *prediction\_output* hücre değişkeni önce matris formatına çevirilir, sonrasında üzerinde düzleştirme yapılıyor.

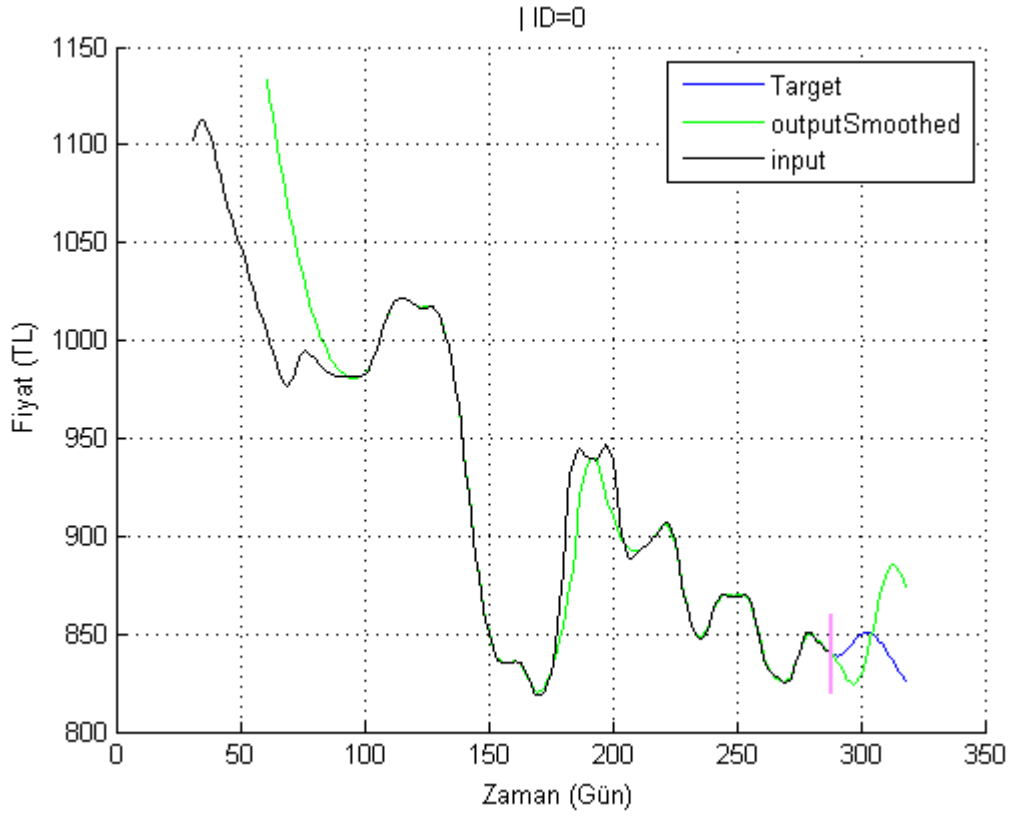
```
prediction_output_smoothed=smooth(
cell2mat(prediction_output),span,'rloess');
```

geliştirilen uygulamada sisteme uygulanan giriş, üretilen sonuç ve beklenen hedef (Target) kıyaslanabilmek için grafiklerle çizilmiştir.

Bu grafiklerde kullanılmış olan *x* eksen vektörleri, *y* ekseninde gösterilen fiyat değerlerinin ilgili gün sayılarının üzerine oturması için, değerleri ait olduğu bölüme göre ayarlanmıştır. mesela *prediction\_output\_smoothed* dizisinin fiyat değerleri  $2*delay+1$  yani 61. günden son güne ait verilere aittir.

`line()` fonksiyonu ile de hakkında tahmin yapılan, son 30 günlük bölümün başlangıcı bir dikey çizgi ile belirlenmiştir. Bu noktanın *x* eksenindeki değeri *dataCount-delay* olarak hesaplanıyor.

Altta şekilde örnek veri için üretilen çıkış grafiği gösterilmiştir.



Şekil 3.11 : Örnek veri üzerinde tahmin sonucu bu grafikte gösterilmiştir.

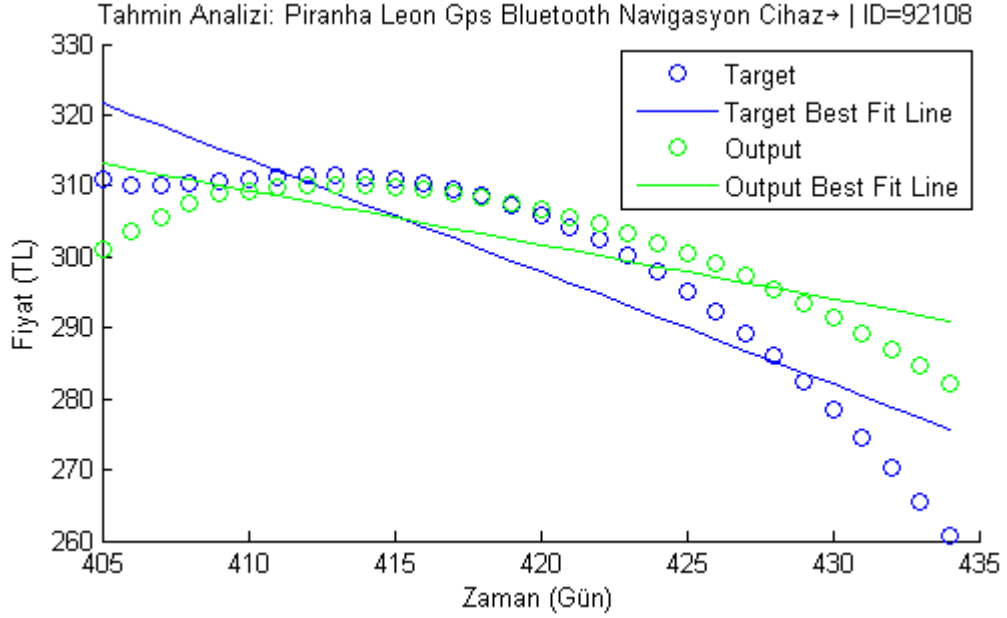
### 3.3.7. Tahmin Sonuç Analizi

Uygulamanın ürettiği tahmin sonucunu başarılı veya başarısız olarak değerlendirebilmek için burada eğilim (trend) analizi yapılmıştır. Başarı sorusuna cevap verebilmek için bu analiz sadece düşüş veya artış göstermeyi önemseyerek yapılmıştır. Burada sonuçlarda üretilen eğrilerin düşüş veya yükseliş eğilimine karar verebilmek için “en iyi uyum hattı (best fit line)” yöntemi kullanılmıştır. Bulunan uyum hattının eğilimi (düşüş veya artış) yapay sinir ağının ürettiği çıkışın eğilimi (düşüş veya artış) olarak kabul edilip analiz edilmiştir. Hedef vektörünün de eğilimi aynı yöntem ile hesaplanmıştır.

Matlab ortamında en iyi uyum hattını bulmak için polyfit() fonksiyonu kullanılmıştır. Bir sonraki şekilde örnek sonuç için bulunan en iyi uyum hatları gösterilmiştir. Bu örnekte hedef ve çıkış için bulunan en iyi uyum hatları her ikisi de aynı yönde (her ikisi



de düşüşü göstermekte) oldukları için tahmin sonucu bu örnek için başarılı olarak kabul edilir.



Şekil 3.12 : Örnek hedef ve çıkış vektörleri için tahmin analizi ve en iyi uyum hatları gösterilmiştir.

Üretilen çıkış ve hedef (Target) vektörleri için bulunan en iyi uyum hattına ait eğilimler aynı yönde değilse tahmin sonucu başarısız olarak kabul edilir.

Uygulamada bulunan  $y=ax+b$  denklemlili “en iyi uyum hattı”nın  $a$  çarpanının işaretine bakılarak hattın eğiliminin yönüne karar veriliyor.

## 4. BULGULAR

### 4.1. UYGULAMA SONUÇLARI

Bu kısımda, 30 adet ürün için bu sistemin uygulandıktan sonra ürettiği sonuçlar yer almaktadır.

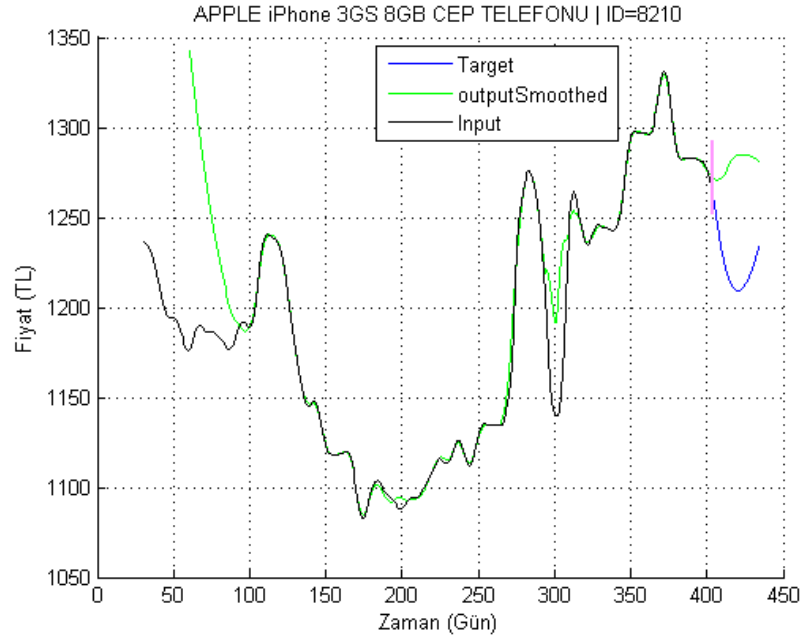
Örneklere yüksek iniş çıkışları olan, keskin çizgilere sahip olan veya yetersiz veri sayısı olanlar da dahil edilmiştir.

Burada kullanılan fiyat verileri grafiklerin başlıklarında yazılmış olan ürünlere aittir ve bu fiyatlar 2009-2011 yıllarında toplanmış olan piyasada bulunan gerçek fiyatlardır. Bu fiyatlar akillifiyat.com sitesinden temin edilmiştir.

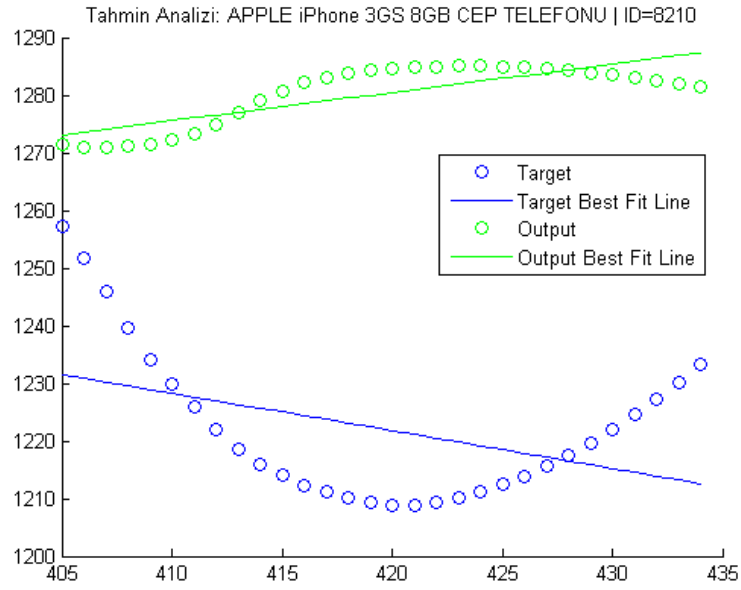
Bu denemelerde her ürün için sistemin ürettiği çıkış grafiği ve sonuç analizi grafiği verilmiştir. Grafiklerin altında sonuç analizi tablosu elde edilen sonuçlara göre çıkarılmıştır.

Burada sonucun başarılı veya başarısız olduğuna karar verebilmek için “en iyi düz çizgi uyarlama (Best Line Fit)” yöntemi kullanılmıştır. Bu işlem Matlab ortamında *polyfit()* fonksiyonu ile yapılmıştır. Eğer Hedef ve Çıkışa ait çıkarılan düz çizgilerin yönü aynı ise (her ikisi de yükseliş veya düşüşü gösteriyor ise) başarılı ve diğer koşullarda başarısız olarak var sayılmıştır.

Eğitim performansı yakalanmamış olarak belirtilen sonuçlarda eğitim aşamasında maksimum 500 tekrarlama (max epoch) ulaşıldığı için eğitim aşamasına son verilmiştir.



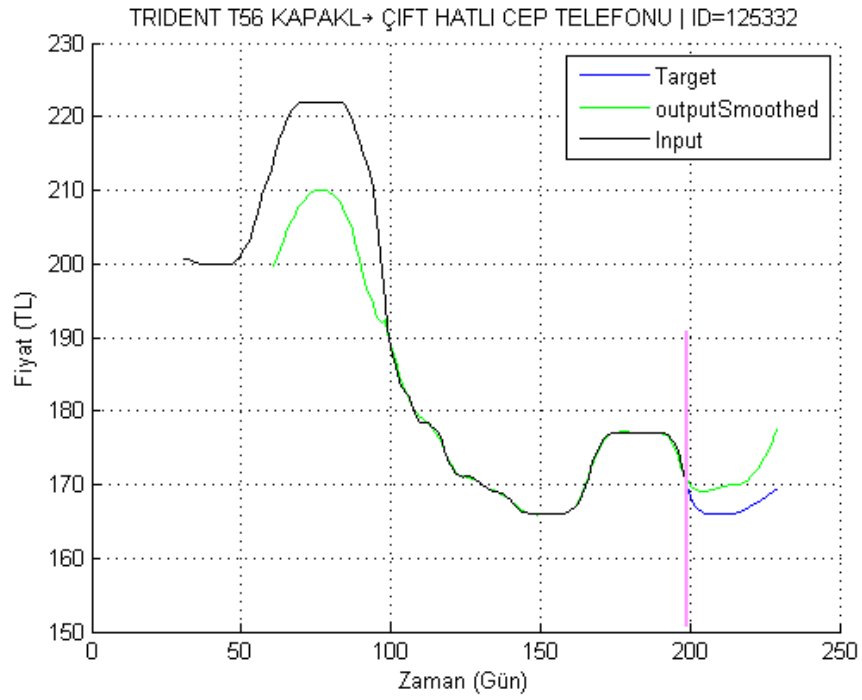
Şekil 4.1 : Sonuç 1



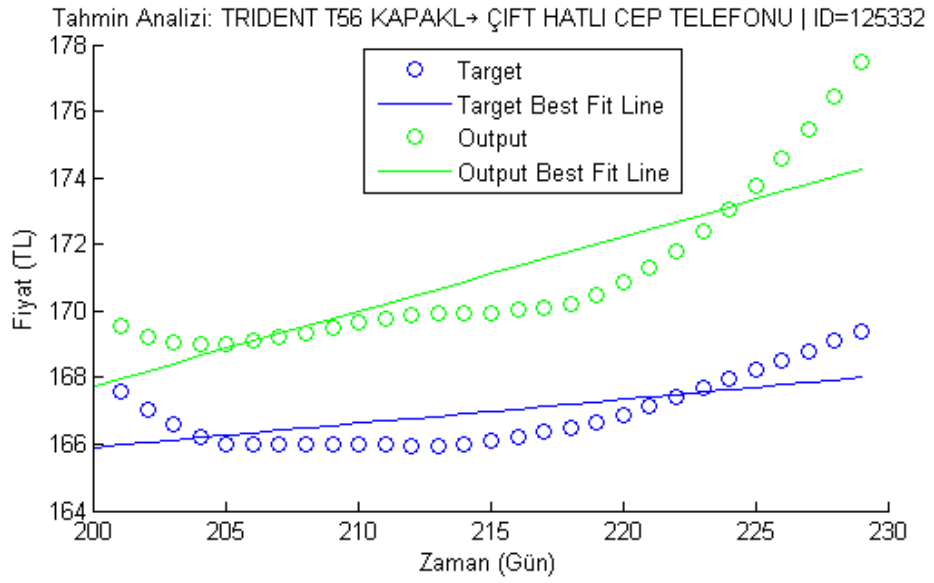
Şekil 4.2 : Sonuç Analizi 1

Kullanılan Veri Sayısı	404
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Düşüş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarısız
Not.	Verilerde yüksek iniş çıkışlar mevcuttur

Tablo 4.1 : Analiz sonucu 1



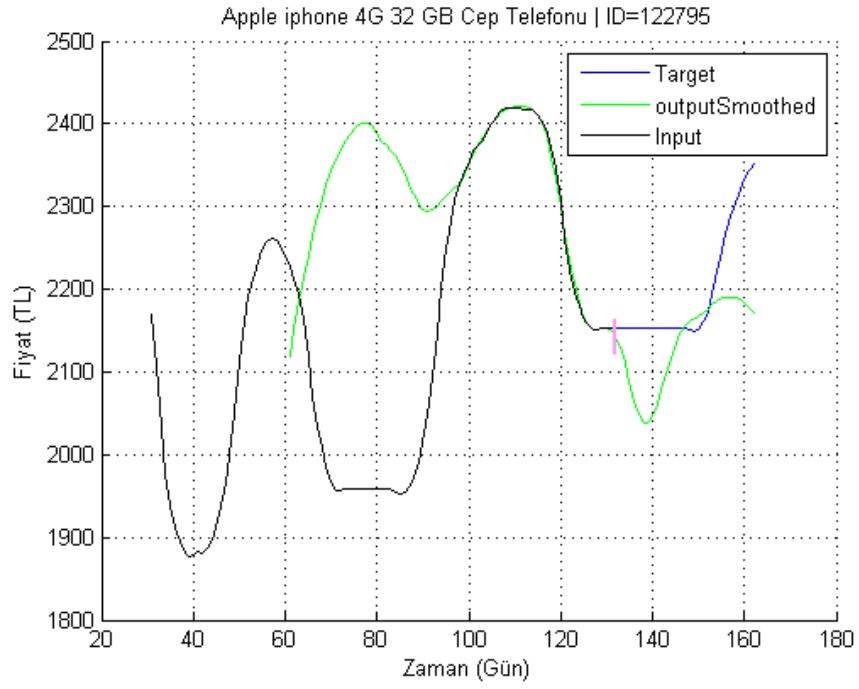
Şekil 4.3 : Sonuç 2



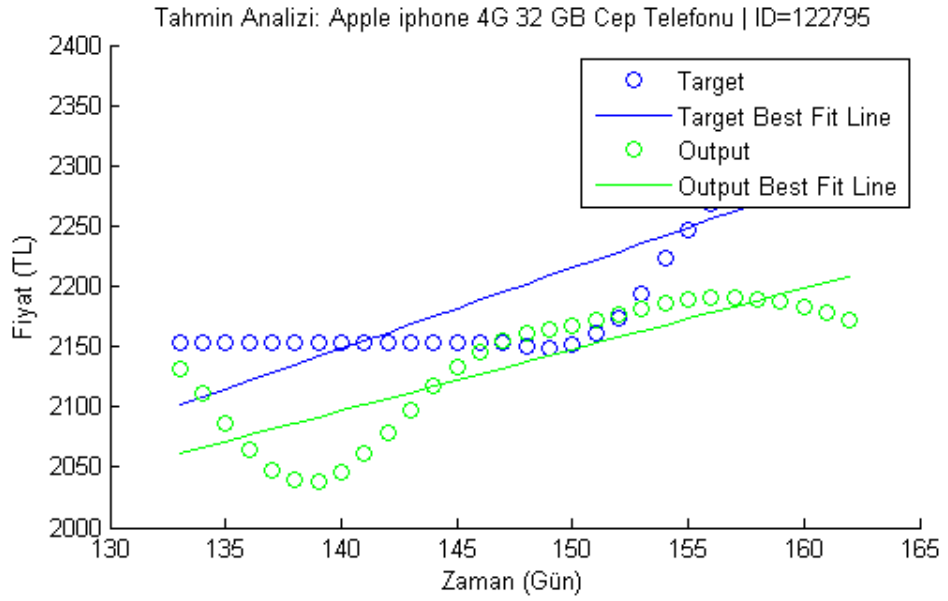
Şekil 4.4 : Sonuç Analizi 2

Kullanılan Veri Sayısı	199
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliliş
Çıkış Eğilimi	Yükseliliş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.2 : Analiz sonucu 2



Şekil 4.5 : Sonuç 3

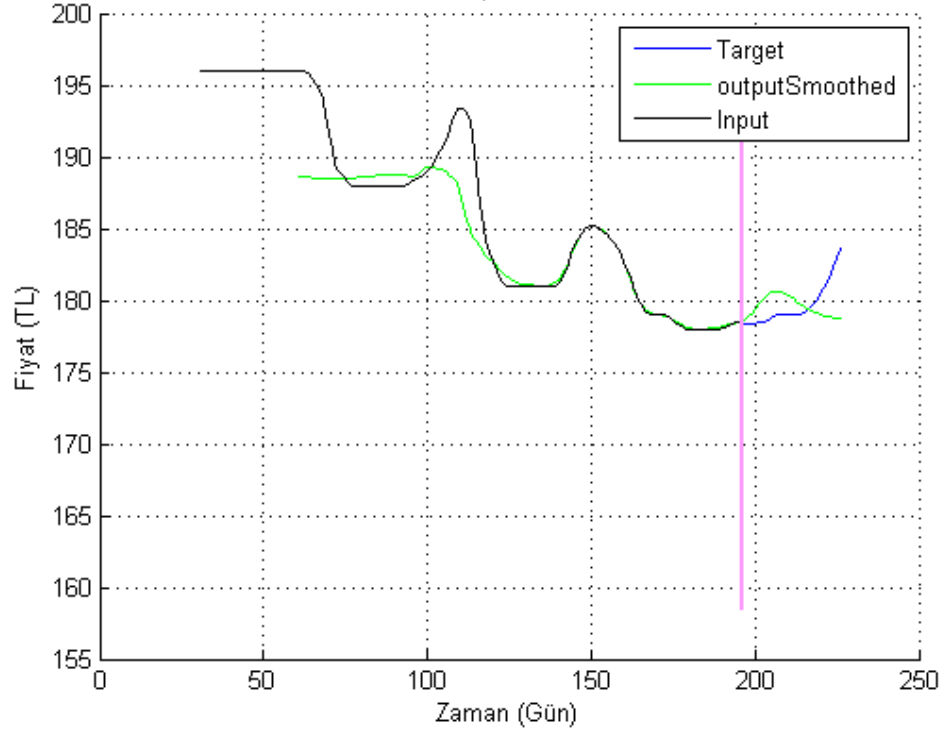


Şekil 4.6 : Sonuç Analizi 3

Kullanılan Veri Sayısı	132
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

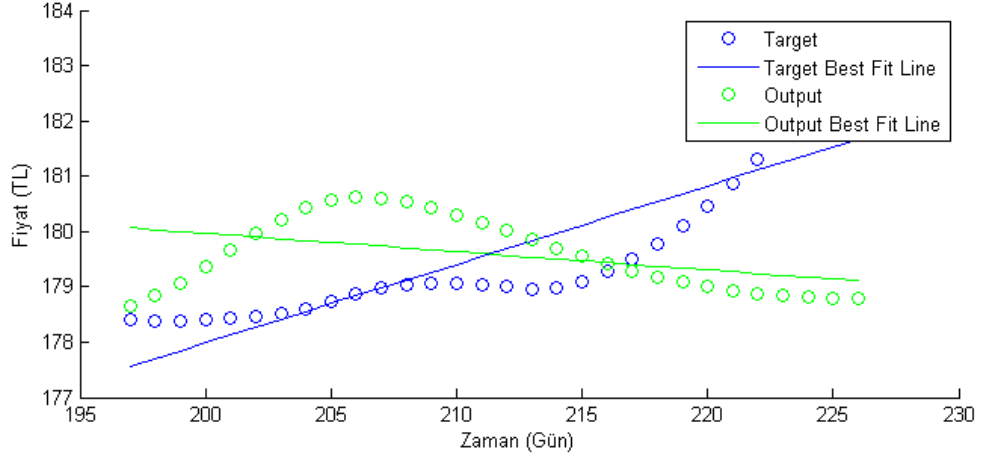
Tablo 4.3 : Analiz sonucu 3

Y DSC-S1900S Foto→raf Makinesi 10.1MP 3XOptik Zoom 2.7 LCD Ekran HD Gümü→ Renk | ID=



Şekil 4.7 : Sonuç 4

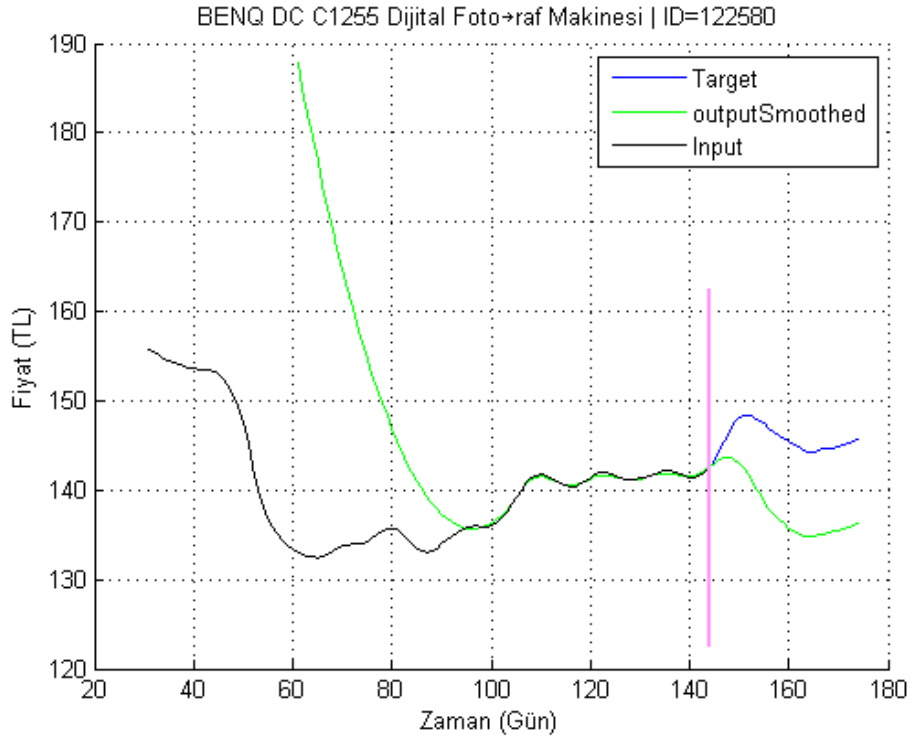
Tahmin Analizi: SONY DSC-S1900S Foto→raf Makinesi 10.1MP 3XOptik Zoom 2.7 LCD Ekran HD Gümü→ Renk | ID=125609



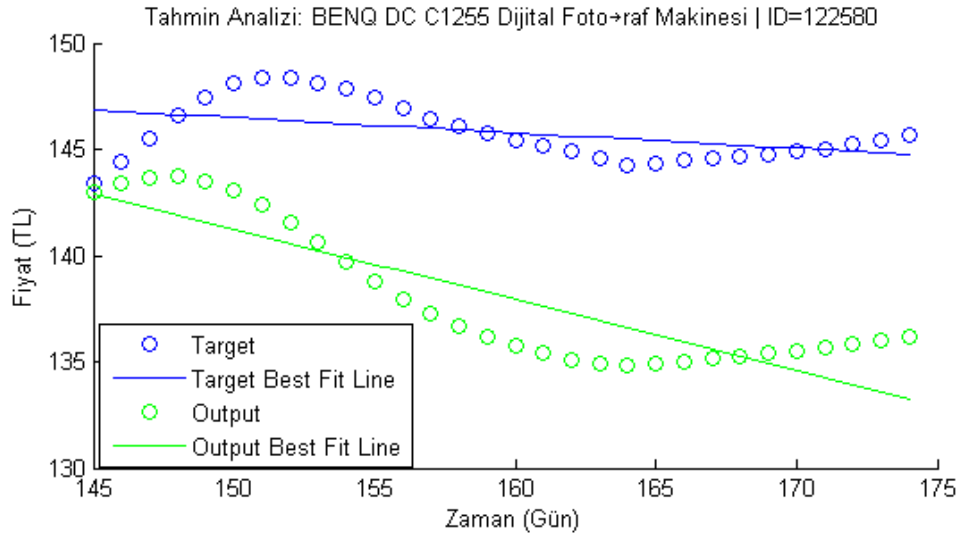
Şekil 4.8 : Sonuç Analizi 4

Kullanılan Veri Sayısı	196
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükselilş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarısız
Not.	

Tablo 4.4 : Analiz sonucu 4



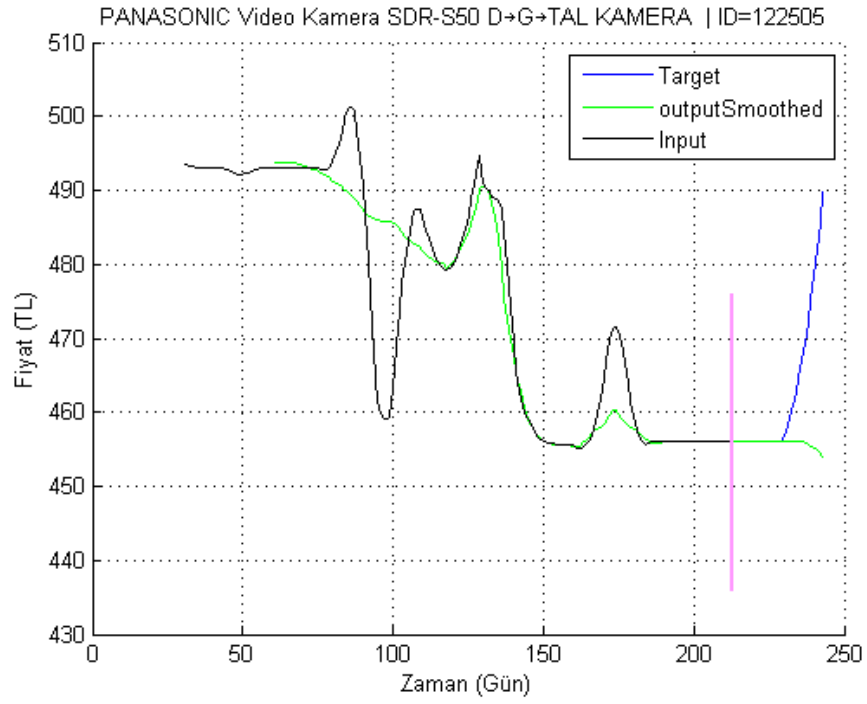
Şekil 4.9 : Sonuç 5



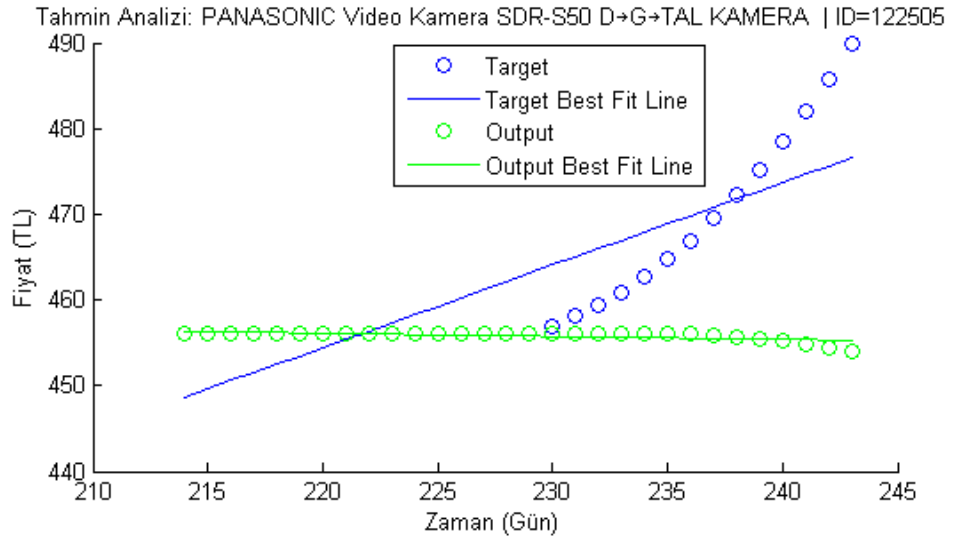
Şekil 4.10 : Sonuç Analizi 5

Kullanılan Veri Sayısı	144
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Düşüş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.5 : Analiz sonucu 5



Şekil 4.11 : Sonuç 6

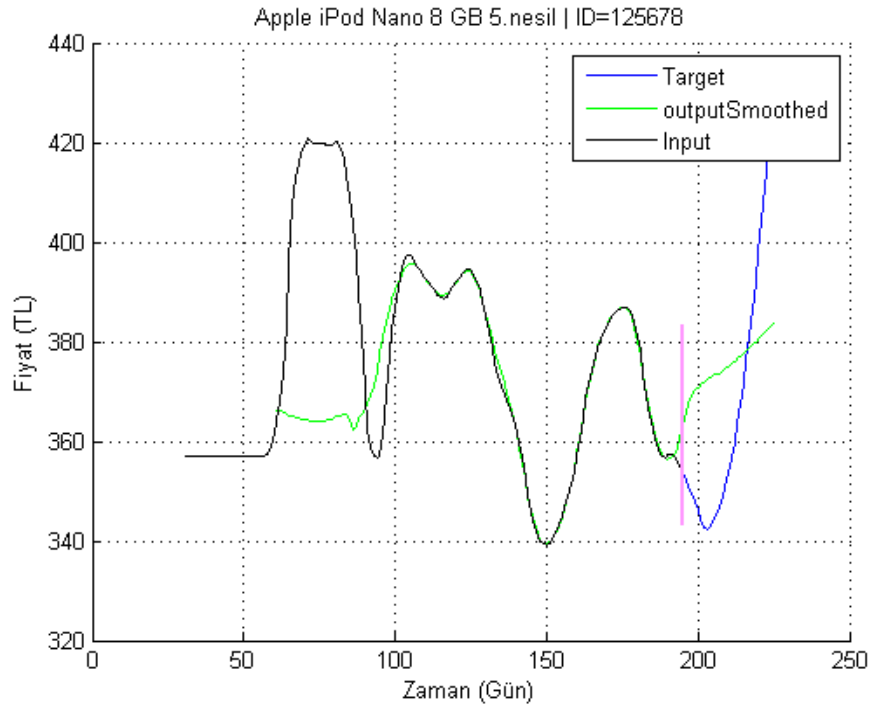


Şekil 4.12 : Sonuç Analizi 6

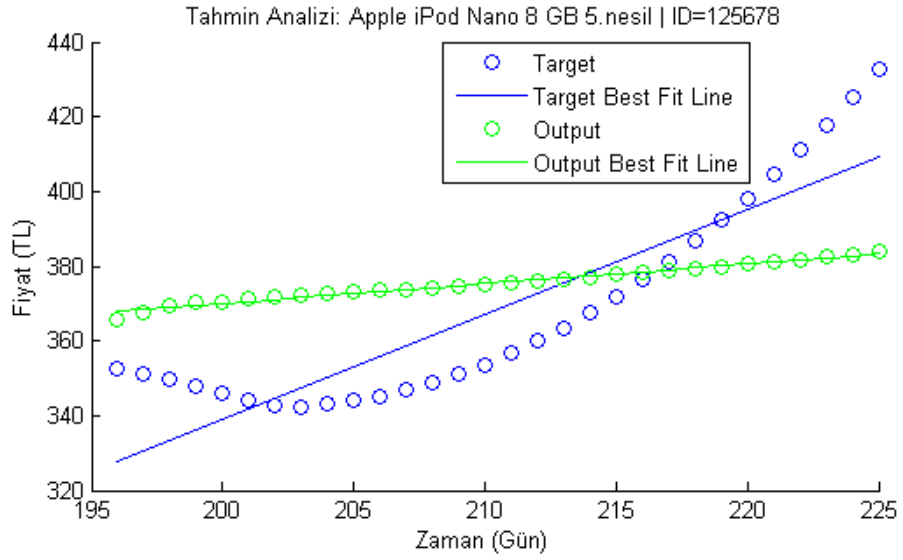
Kullanılan Veri Sayısı	213
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarısız
Not.	

Tablo 4.6 : Analiz sonucu 6





Şekil 4.13 : Sonuç 7

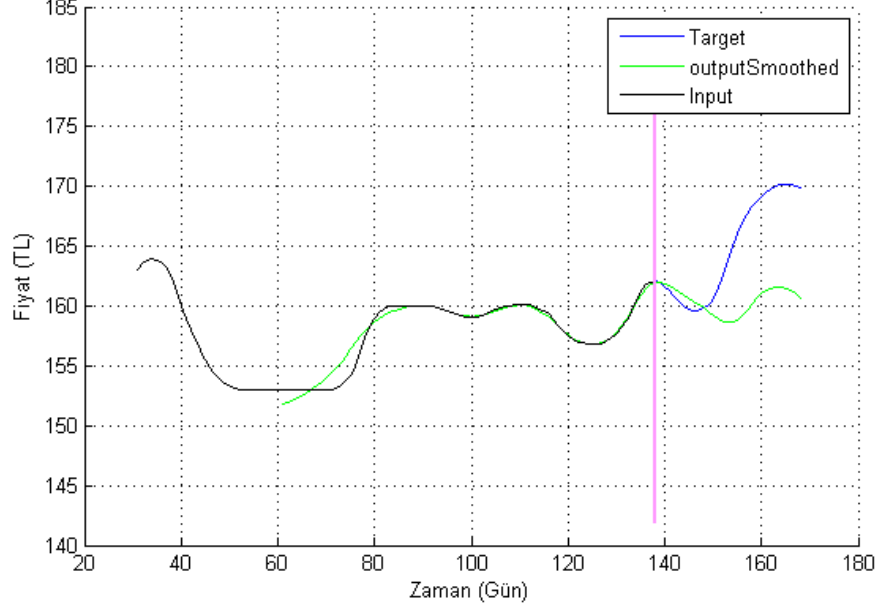


Şekil 4.14 : Sonuç Analizi 7

Kullanılan Veri Sayısı	195
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

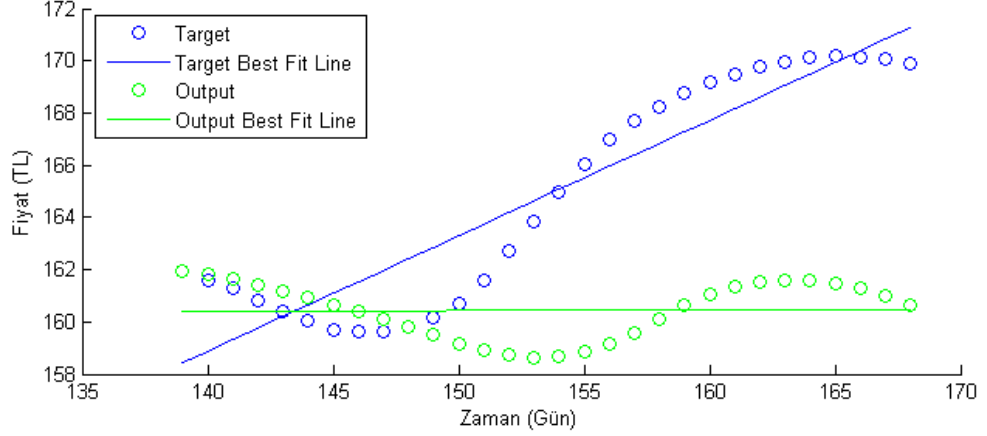
Tablo 4.7 : Analiz sonucu 7

SONY 2GB SES KAYIT C→HAZI 535 SAAT KAYIT USB 5 SATIR LCD EKРАН (ICD-UX200) | ID=165340



Şekil 4.15 : Sonuç 8

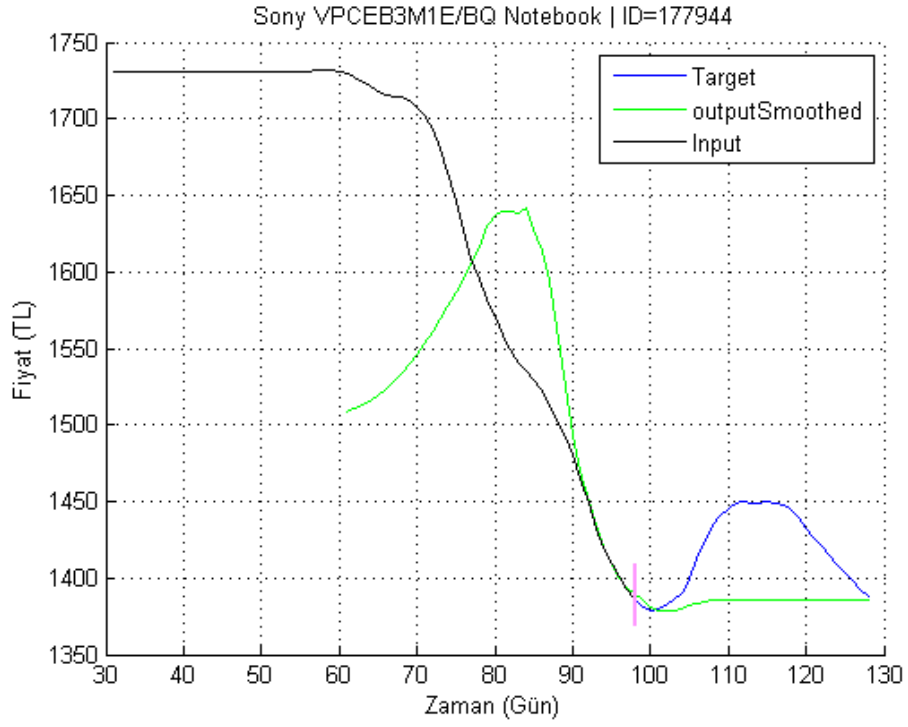
Tahmin Analizi: SONY 2GB SES KAYIT C→HAZI 535 SAAT KAYIT USB 5 SATIR LCD EKРАН (ICD-UX200) | ID=165340



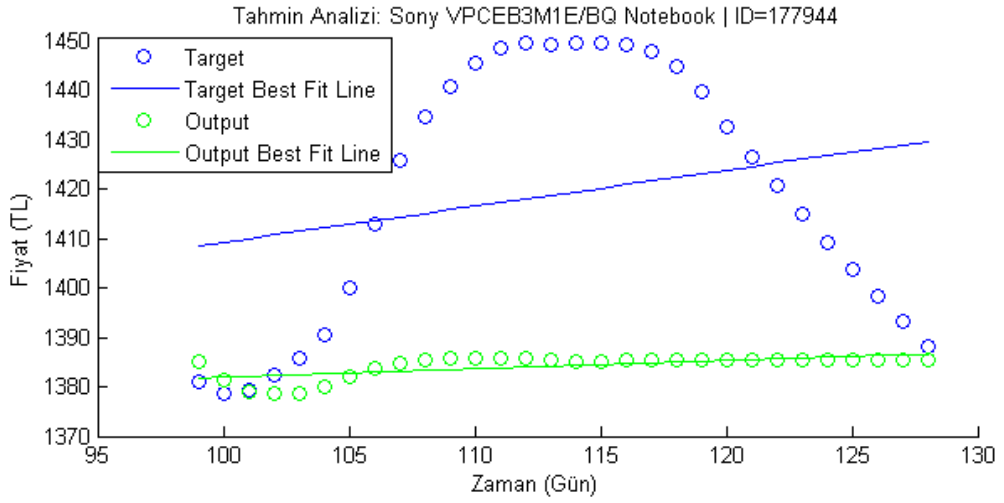
Şekil 4.16 : Sonuç Analizi 8

Kullanılan Veri Sayısı	138
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.8 : Analiz sonucu 8



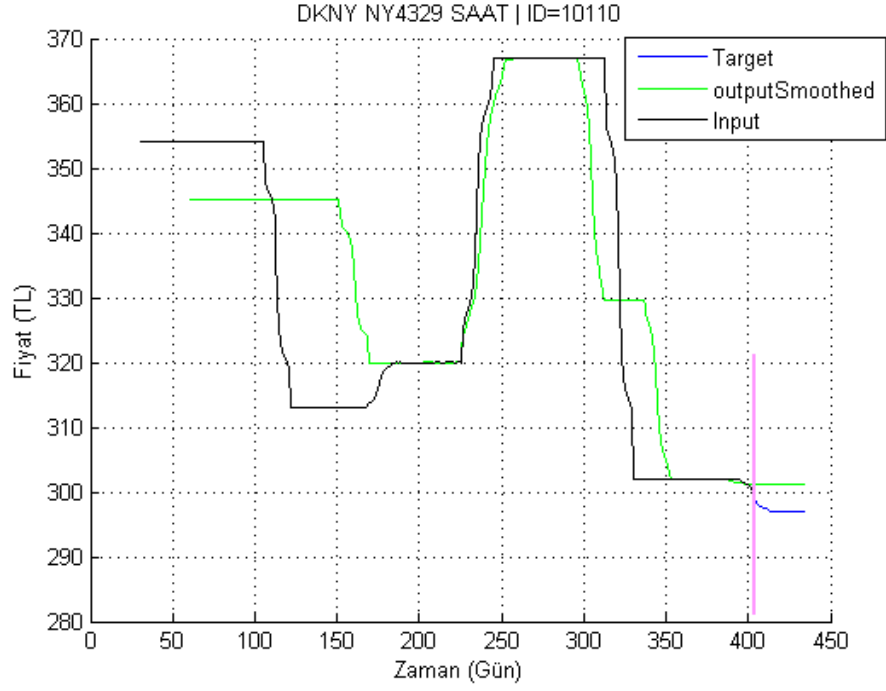
Şekil 4.17 : Sonuç 9



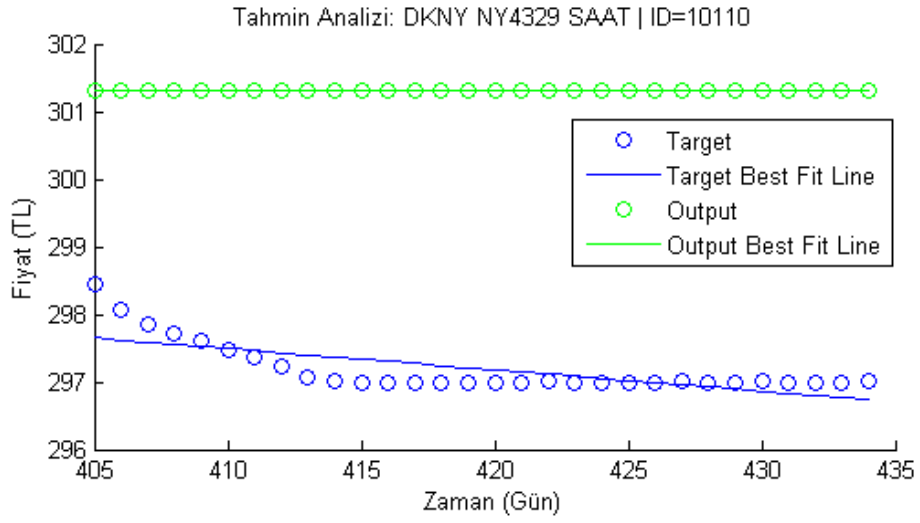
Şekil 4.18 : Sonuç Analizi 9

Kullanılan Veri Sayısı	98
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	Veri sayısı düşük

Tablo 4.9 : Analiz sonucu 9



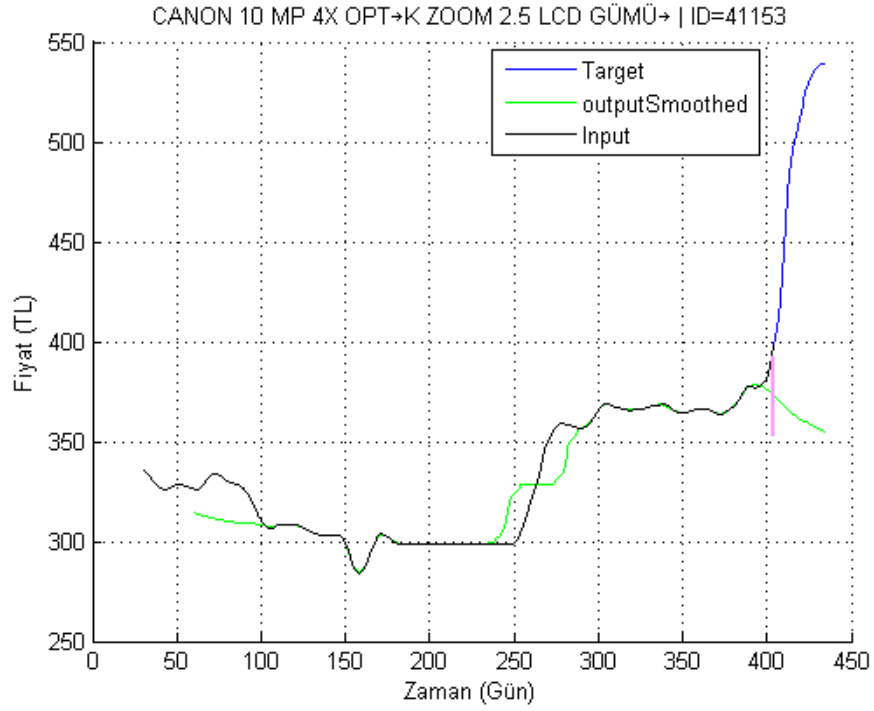
Şekil 4.19 : Sonuç 10



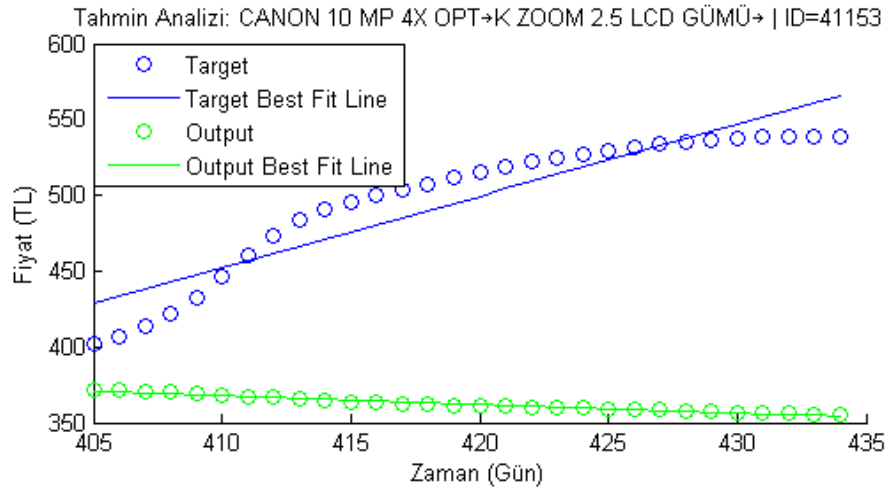
Şekil 4.20 : Sonuç Analizi 10

Kullanılan Veri Sayısı	404
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Düşüş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarısız
Not.	Grafikte keskin köşeler mevcuttur

Tablo 4.10 : Analiz sonucu 10



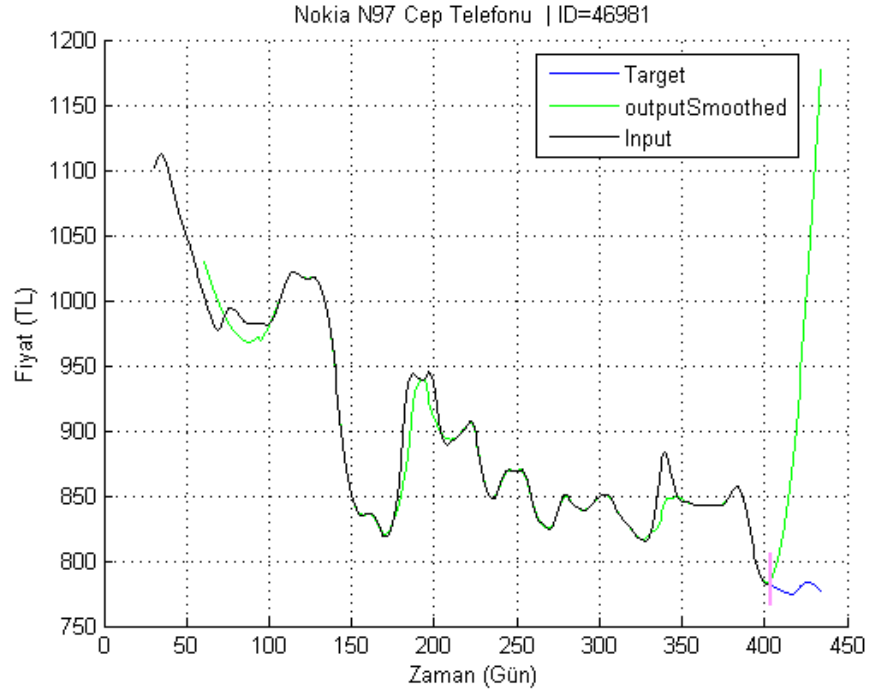
Şekil 4.21 : Sonuç 11



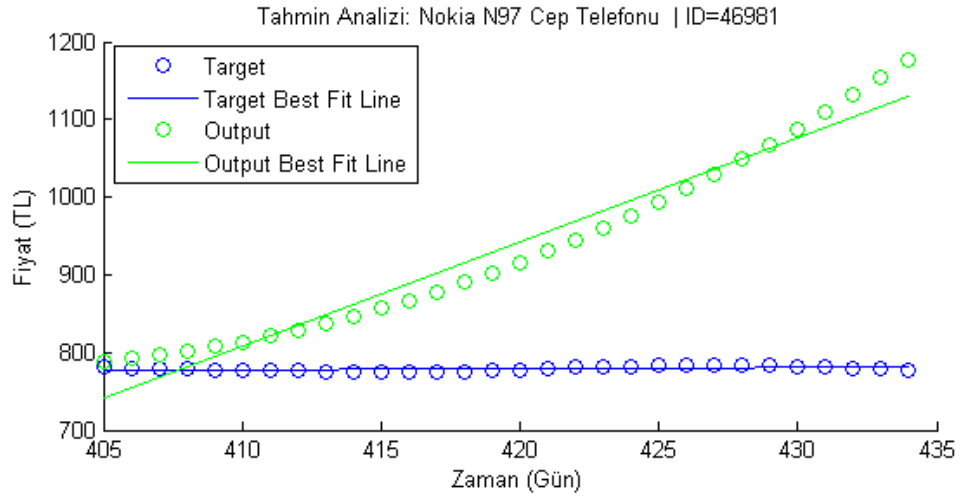
Şekil 4.22 : Sonuç Analizi 11

Kullanılan Veri Sayısı	404
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarısız
Not.	

Tablo 4.11 : Analiz sonucu 11



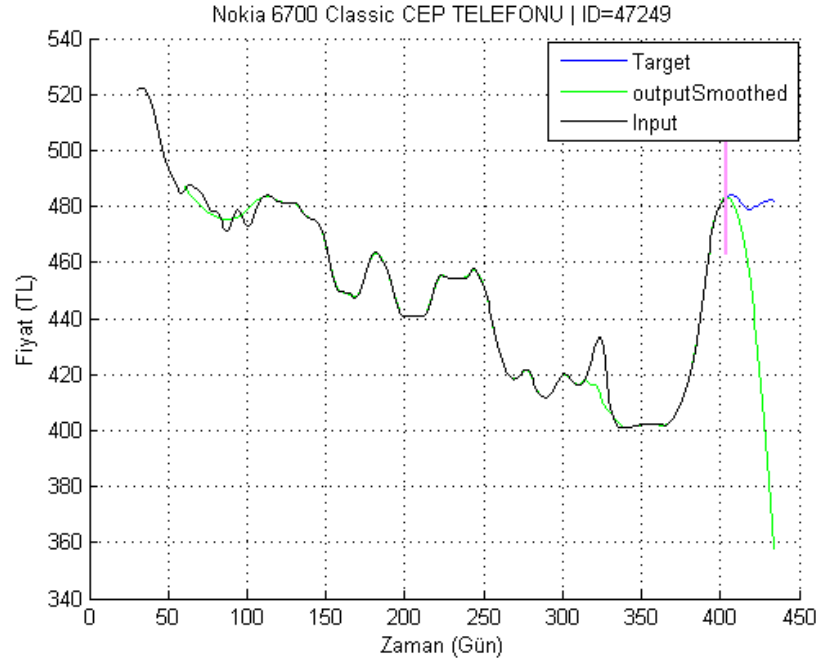
Şekil 4.23 : Sonuç 12



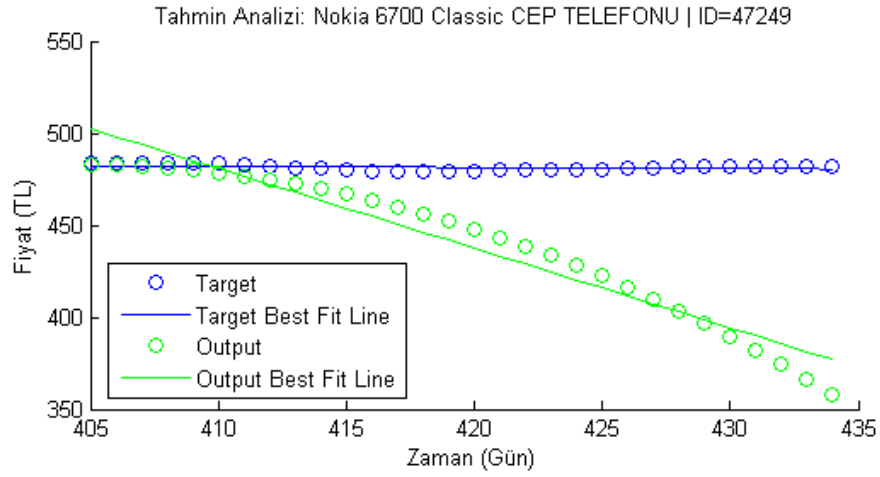
Şekil 4.24 : Sonuç Analizi 12

Kullanılan Veri Sayısı	404
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.12 : Analiz sonucu 12



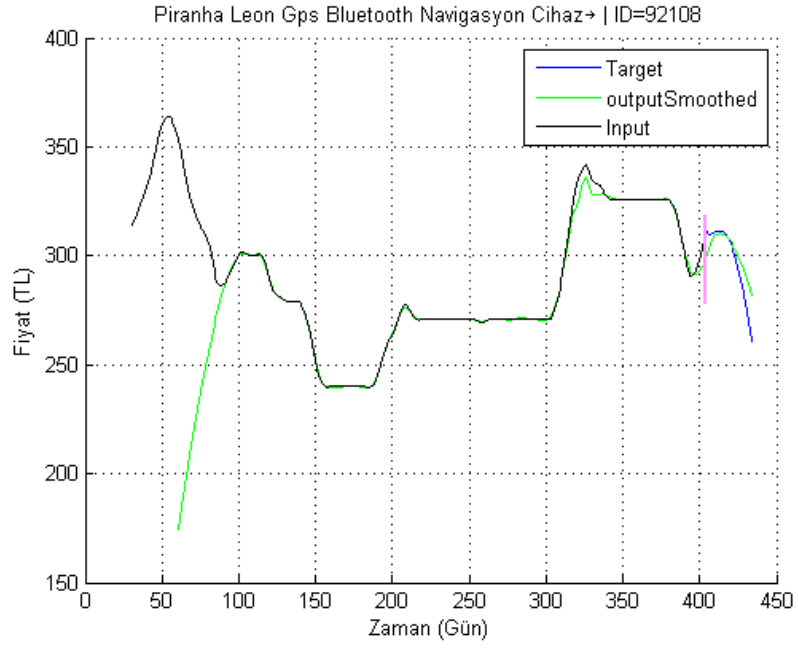
Şekil 4.25 : Sonuç 13



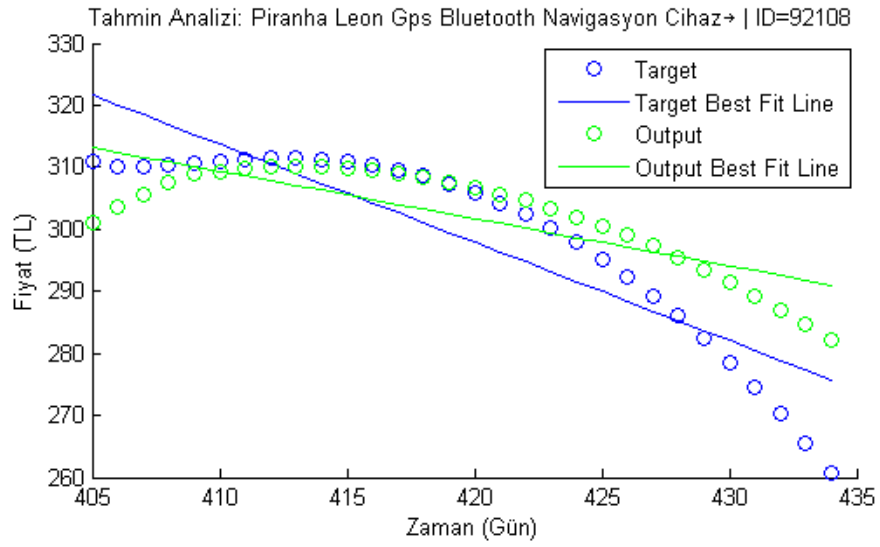
Şekil 4.26 : Sonuç Analizi 13

Kullanılan Veri Sayısı	404
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Düşüş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.13 : Analiz sonucu 13



Şekil 4.27 : Sonuç 14

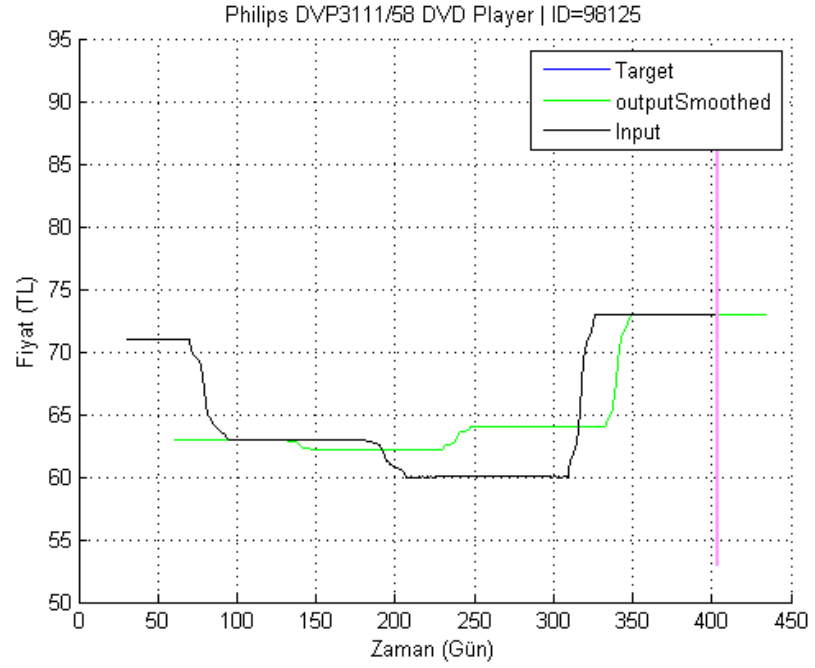


Şekil 4.28 : Sonuç Analizi 14

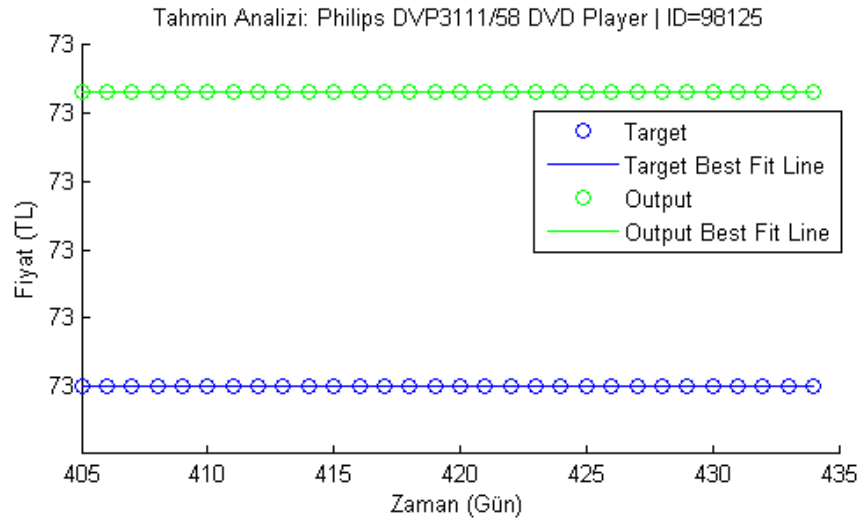
Kullanılan Veri Sayısı	404
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Düşüş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.14 : Analiz sonucu 14





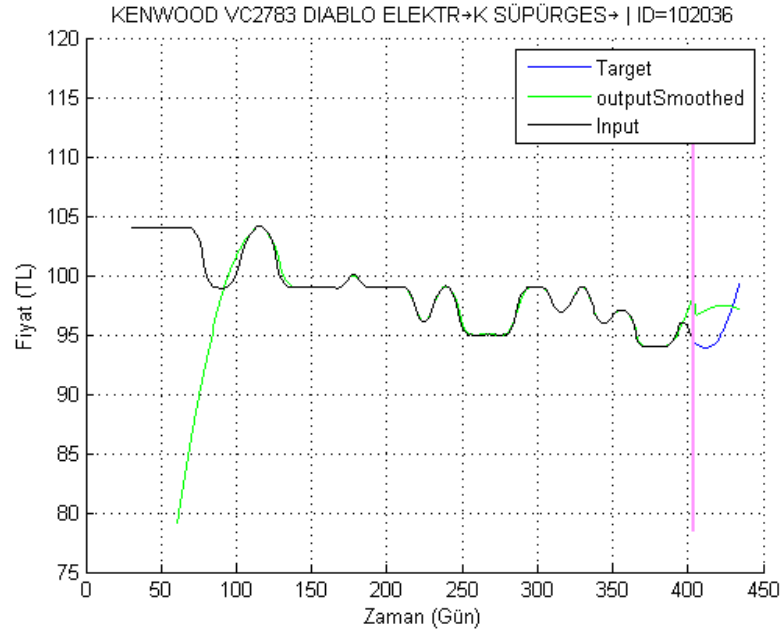
Şekil 4.29 : Sonuç 15



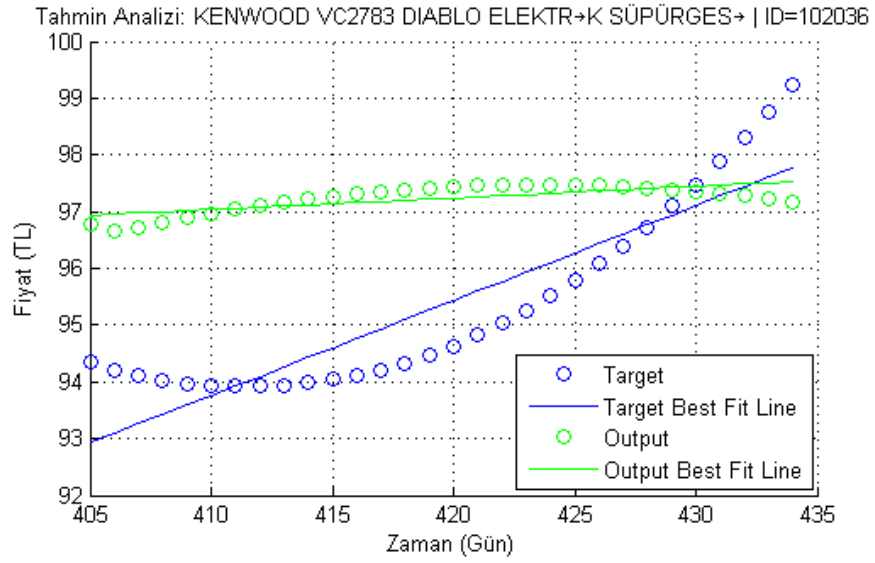
Şekil 4.30 : Sonuç Analizi 15

Kullanılan Veri Sayısı	404
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Düşüş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarısız
Not.	Giriş verileri sağlıklı değil ve keskin köşeleri vardır.

Tablo 4.15 : Analiz sonucu 15



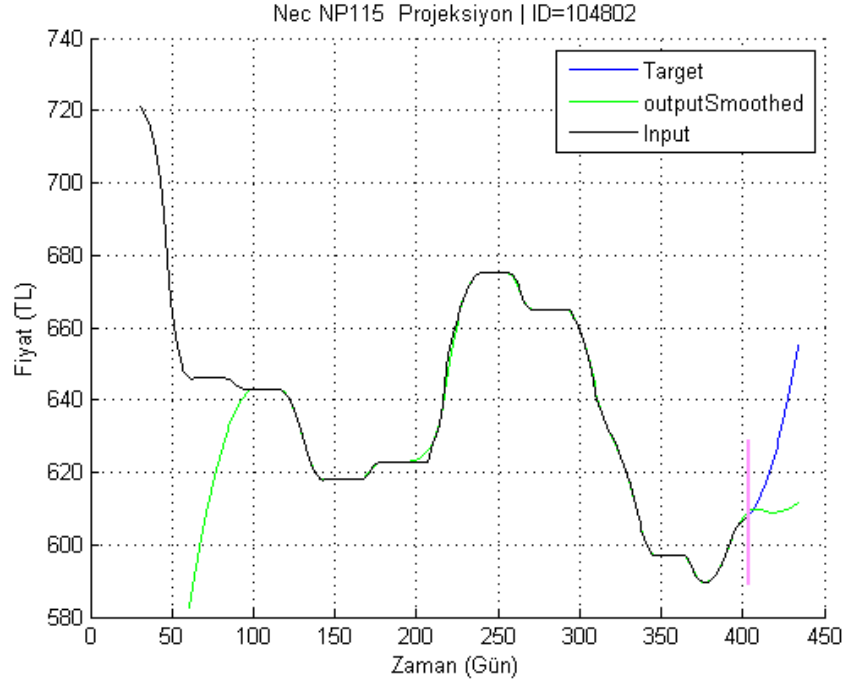
Şekil 4.31 : Sonuç 16



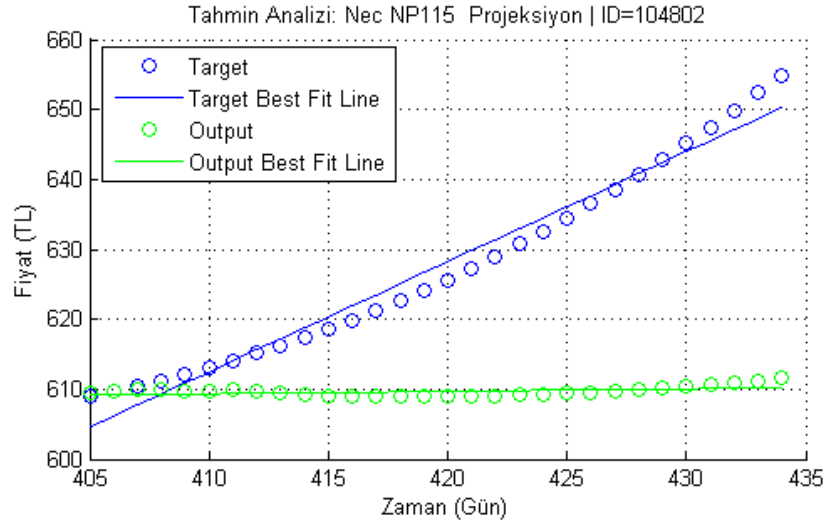
Şekil 4.32 : Sonuç Analizi 16

Kullanılan Veri Sayısı	404
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.16 : Analiz sonucu 16



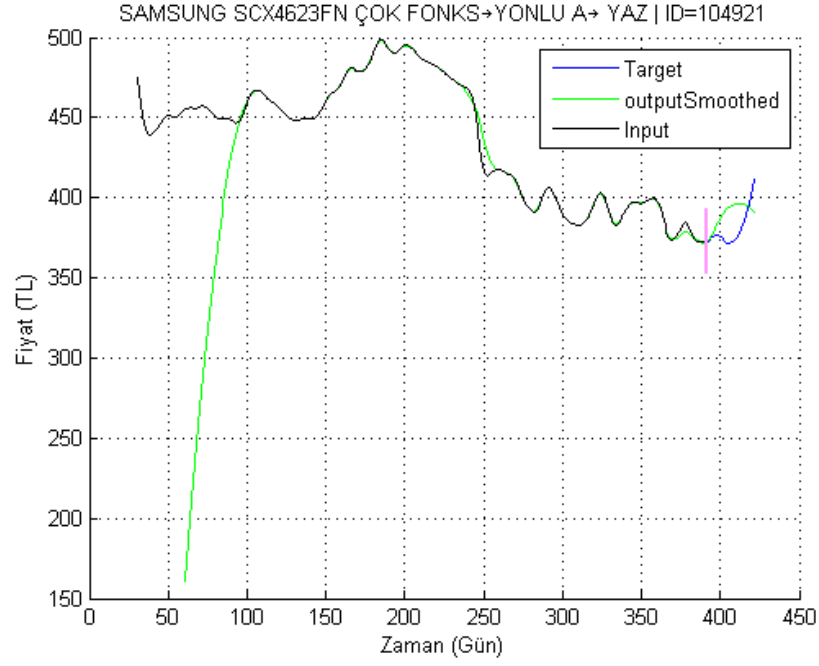
Şekil 4.33 : Sonuç 17



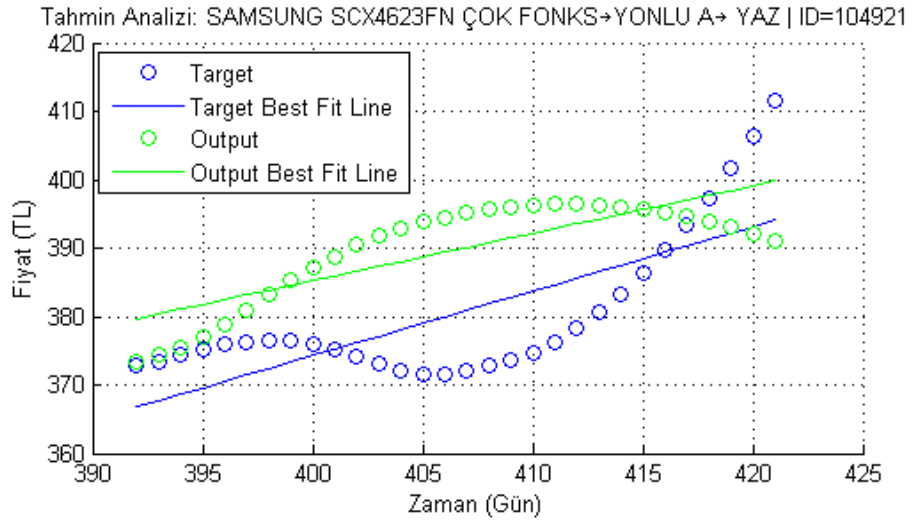
Şekil 4.34 : Sonuç Analizi 17

Kullanılan Veri Sayısı	404
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.17 : Analiz sonucu 17



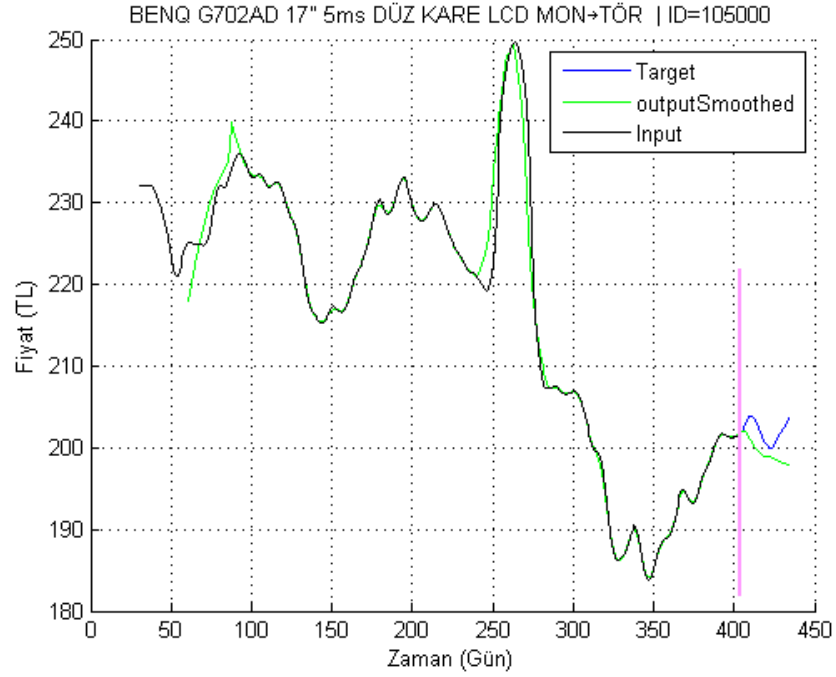
Şekil 4.35 : Sonuç 18



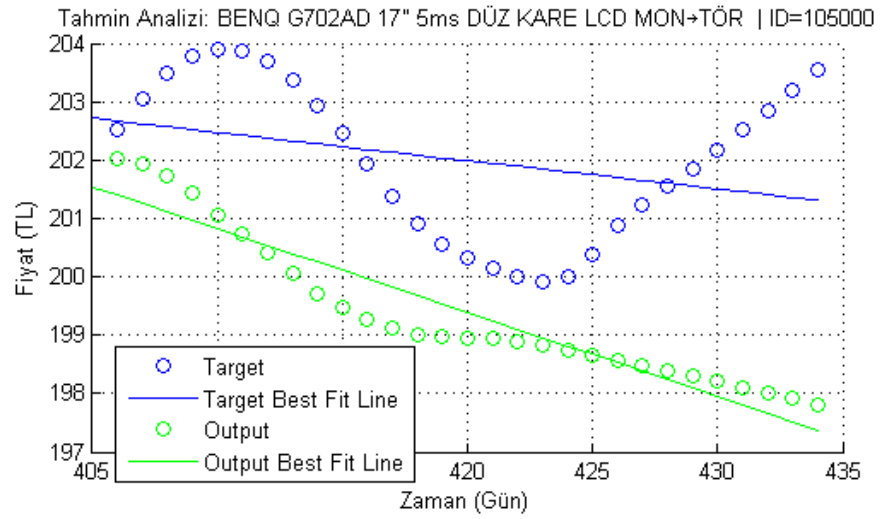
Şekil 4.36 : Sonuç Analizi 18

Kullanılan Veri Sayısı	391
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.18 : Analiz sonucu 18



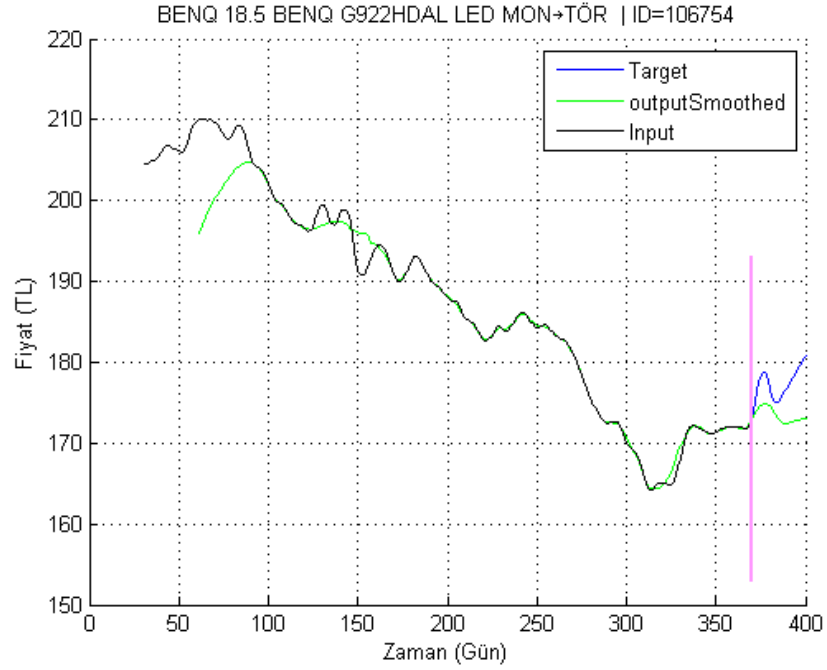
Şekil 4.37 : Sonuç 19



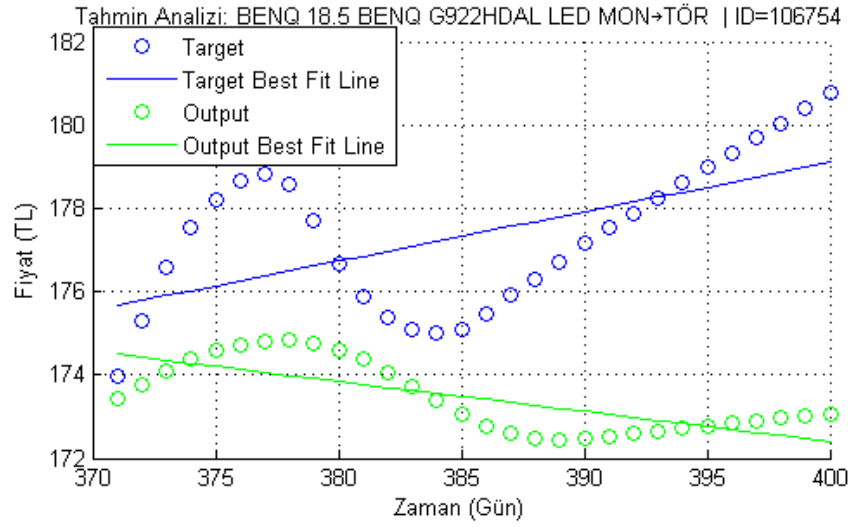
Şekil 4.38 : Sonuç Analizi 19

Kullanılan Veri Sayısı	404
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Düşüş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.19 : Analiz sonucu 19



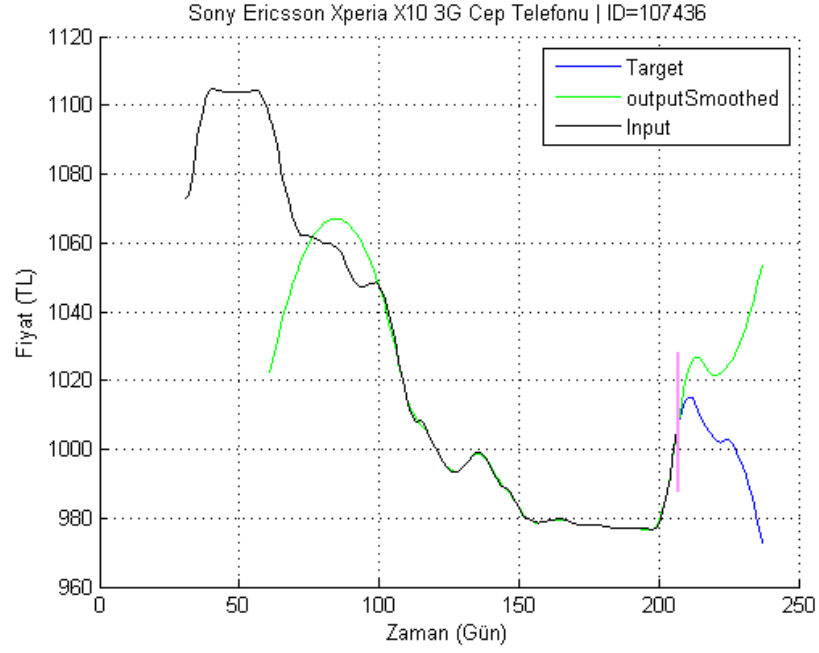
Şekil 4.39 : Sonuç 20



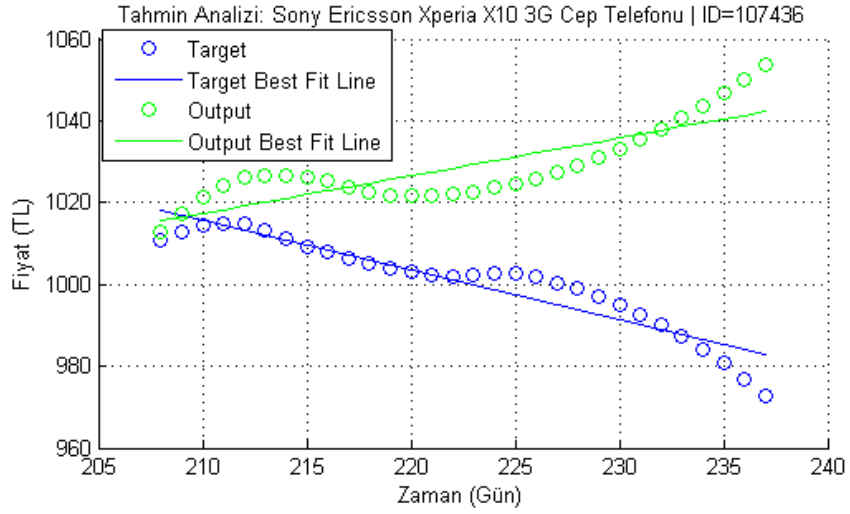
Şekil 4.40 : Sonuç Analizi 20

Kullanılan Veri Sayısı	370
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarısız
Not.	

Tablo 4.20 : Analiz sonucu 20



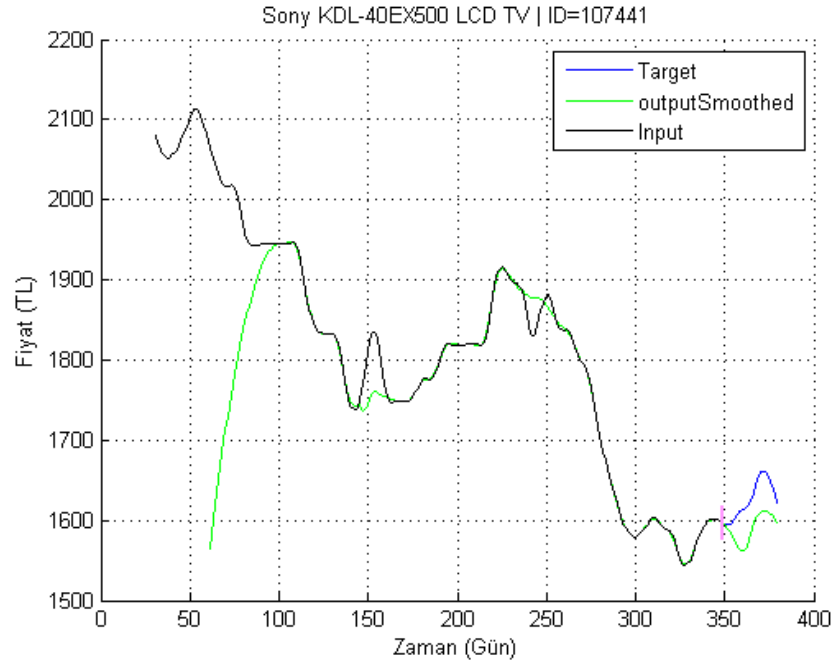
Şekil 4.41 : Sonuç 21



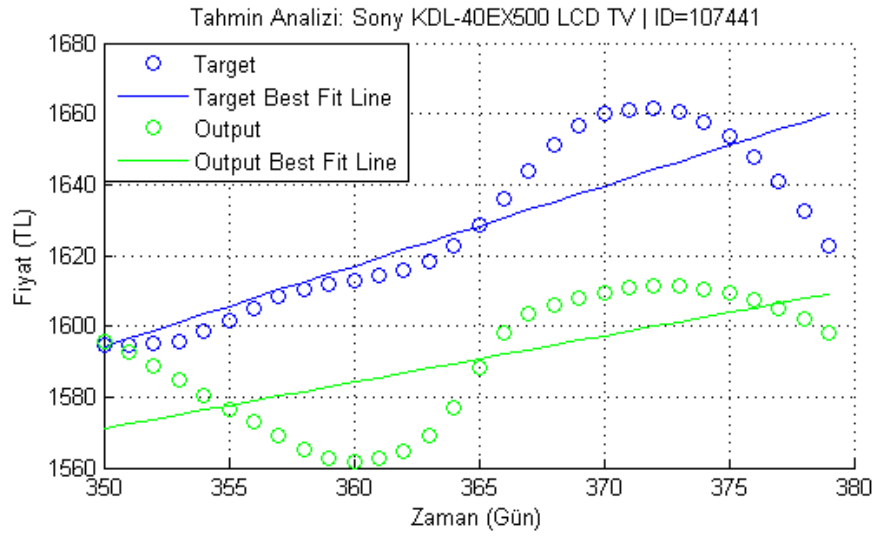
Şekil 4.42 : Sonuç Analizi 21

Kullanılan Veri Sayısı	207
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Düşüş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarısız
Not.	

Tablo 4.21 : Analiz sonucu 21



Şekil 4.43 : Sonuç 22

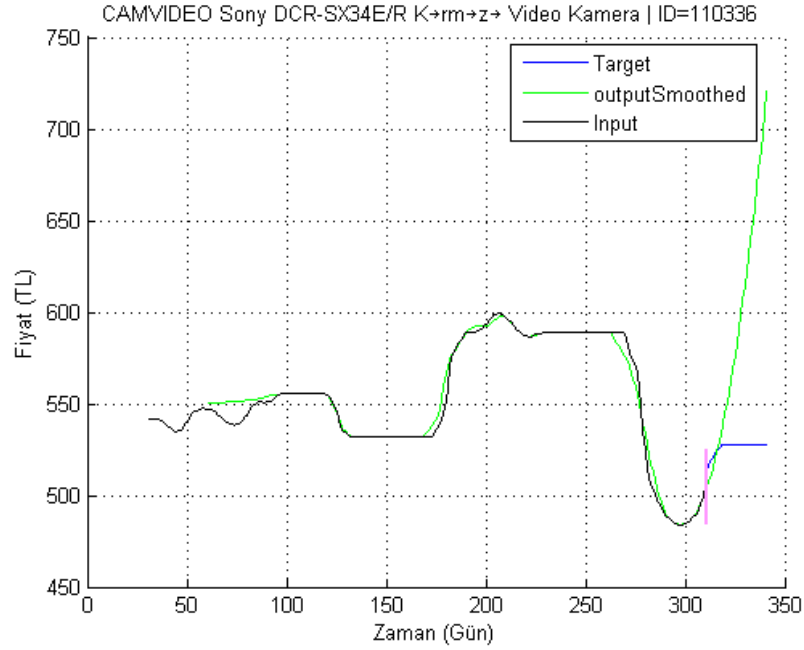


Şekil 4.44 : Sonuç Analizi 22

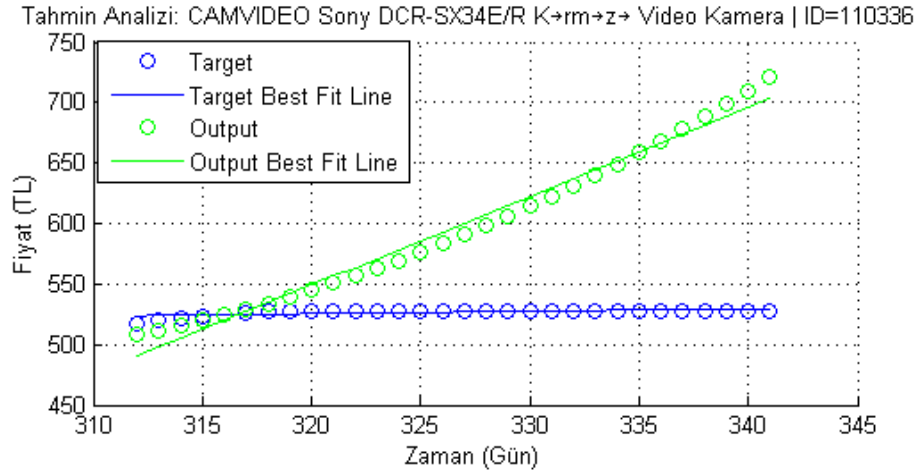
Kullanılan Veri Sayısı	349
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.22 : Analiz sonucu 22





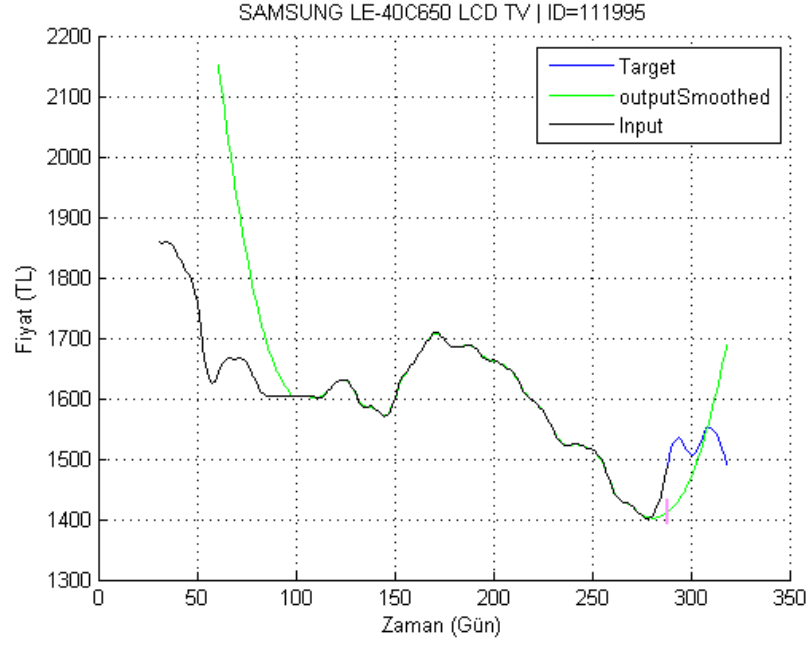
Şekil 4.45 : Sonuç 23



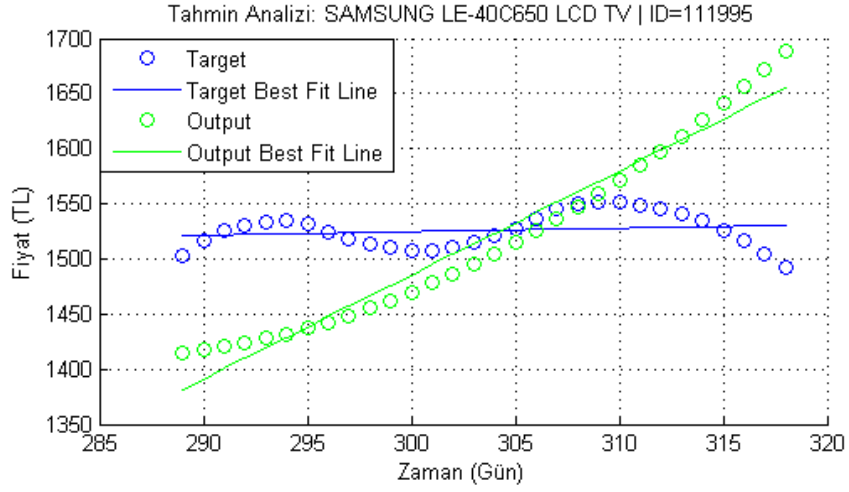
Şekil 4.46 : Sonuç Analizi 23

Kullanılan Veri Sayısı	311
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	Target Eğilimi sıfıra yakın pozitif bir değer çıktığı için Yükseliş olarak kabul edilmiştir.

Tablo 4.23 : Analiz sonucu 23



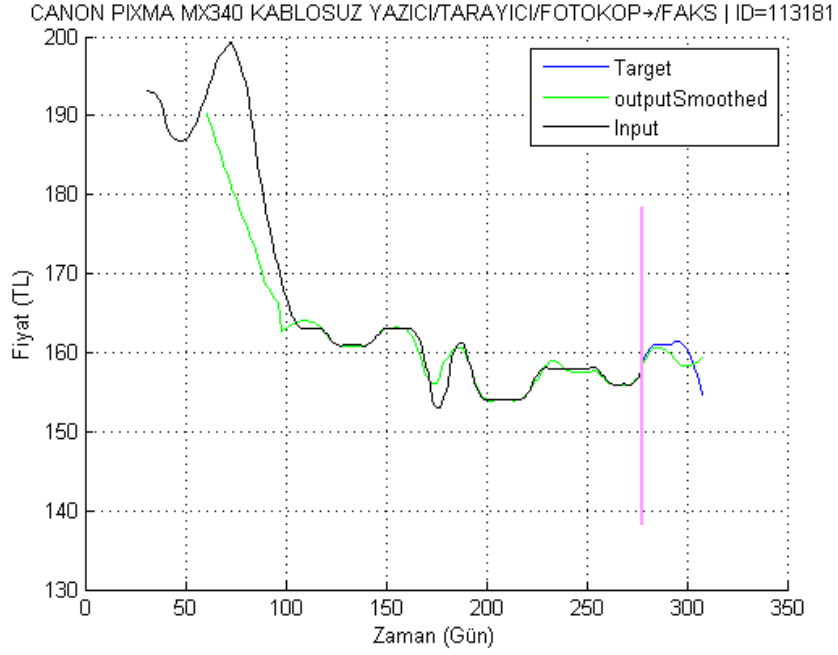
Şekil 4.47 : Sonuç 24



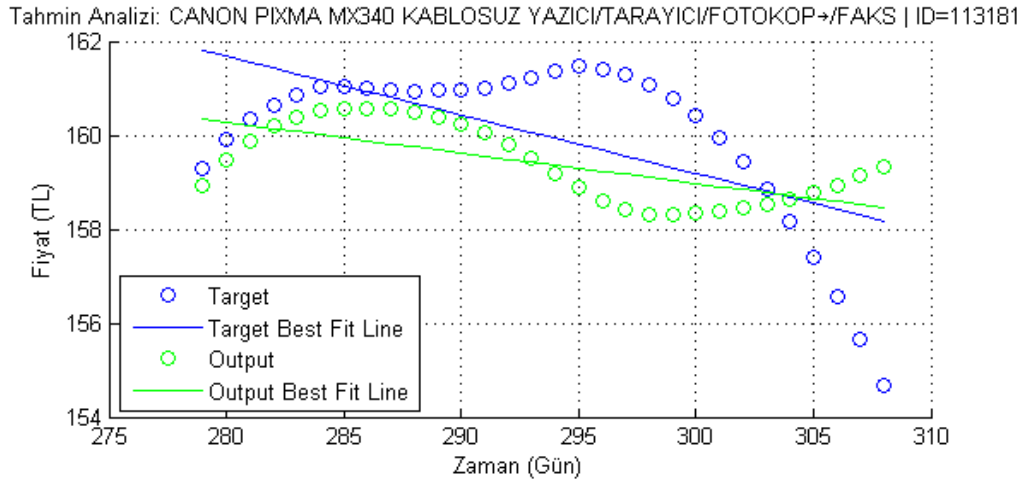
Şekil 4.48 : Sonuç Analizi 24

Kullanılan Veri Sayısı	288
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.24 : Analiz sonucu 24



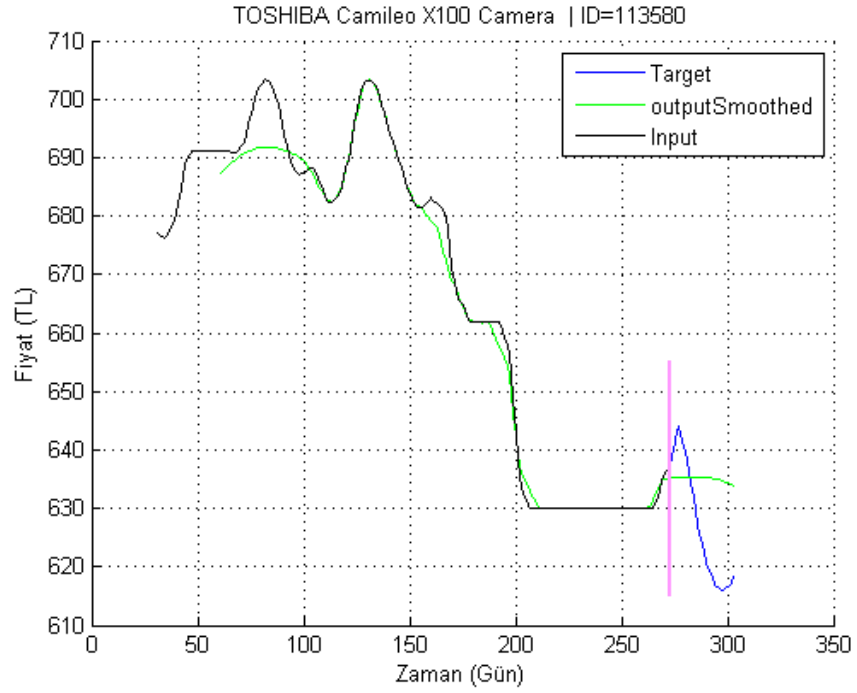
Şekil 4.49 : Sonuç 25



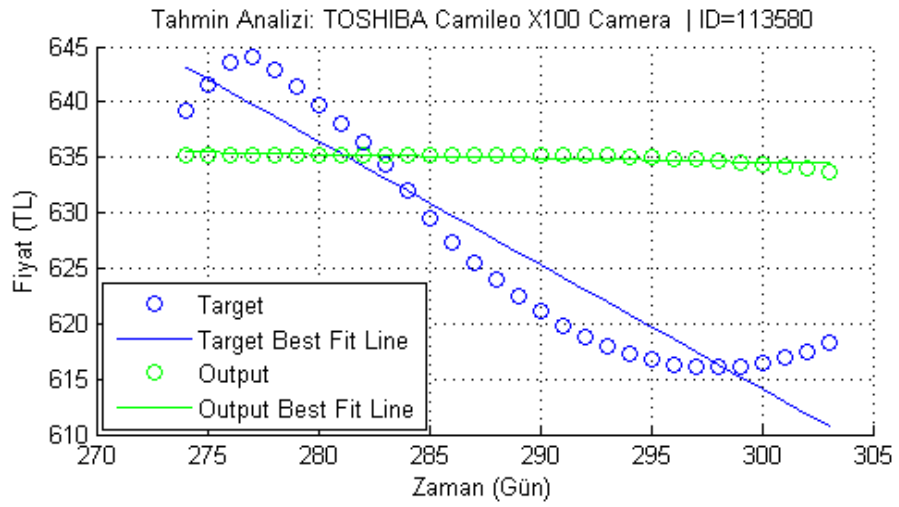
Şekil 4.50 : Sonuç Analizi 25

Kullanılan Veri Sayısı	278
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Düşüş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.25 : Analiz sonucu 25



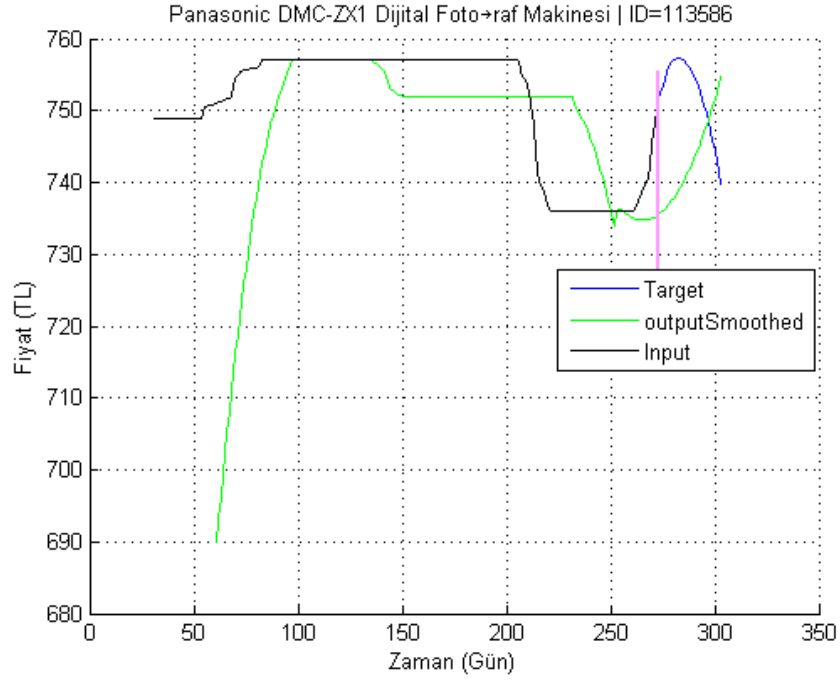
Şekil 4.51 : Sonuç 26



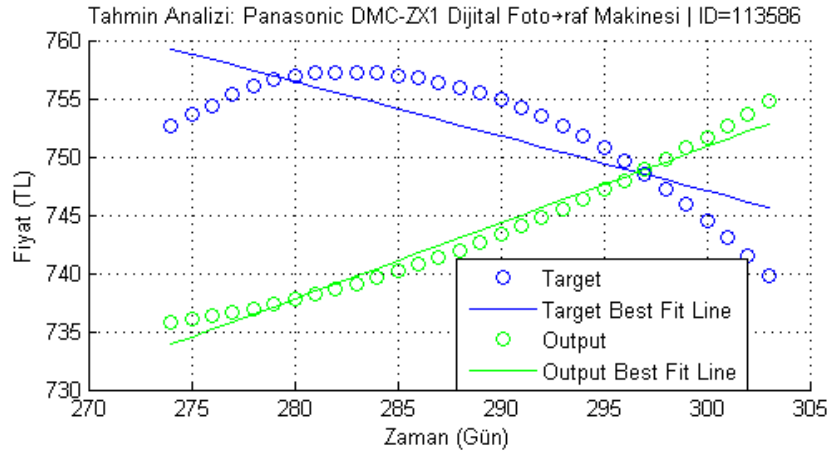
Şekil 4.52 : Sonuç Analizi 26

Kullanılan Veri Sayısı	273
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Düşüş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.26 : Analiz sonucu 26



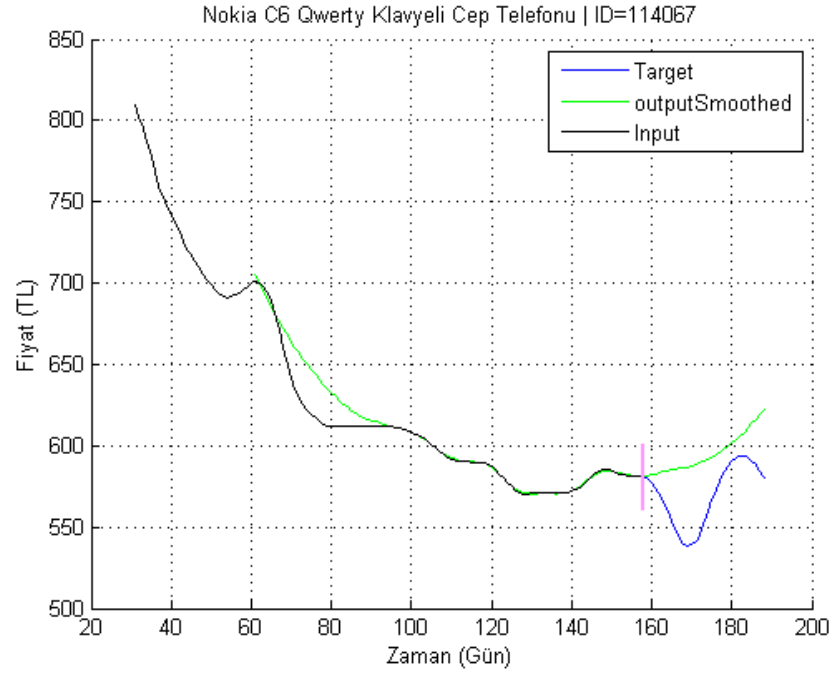
Şekil 4.53 : Sonuç 27



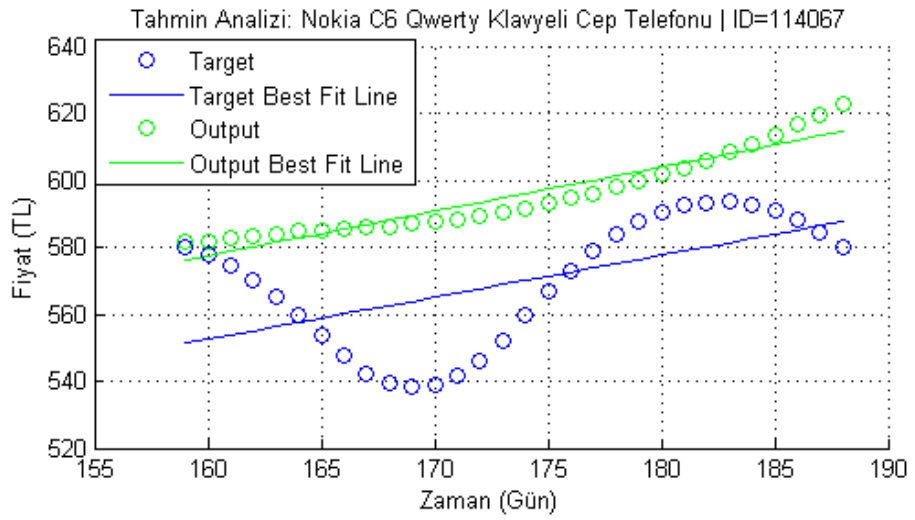
Şekil 4.54 : Sonuç Analizi 27

Kullanılan Veri Sayısı	273
Eğitim Performansı yakalandı mı	Hayır
Target Eğilimi	Düşüş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarısız
Not.	Veriler sağlıklı değil ve keskin köşeler mevcuttur.

Tablo 4.27 : Analiz sonucu 27



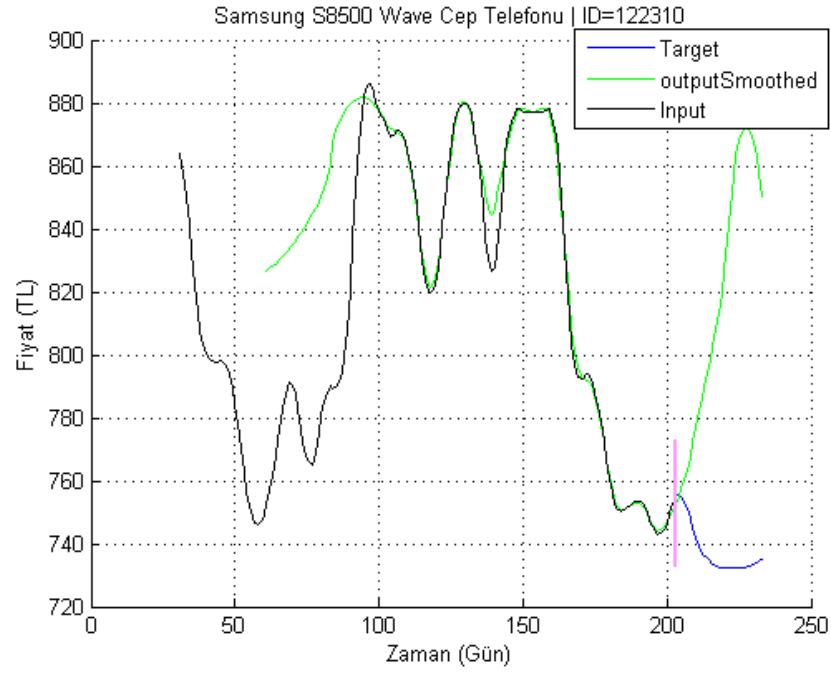
Şekil 4.55 : Sonuç 28



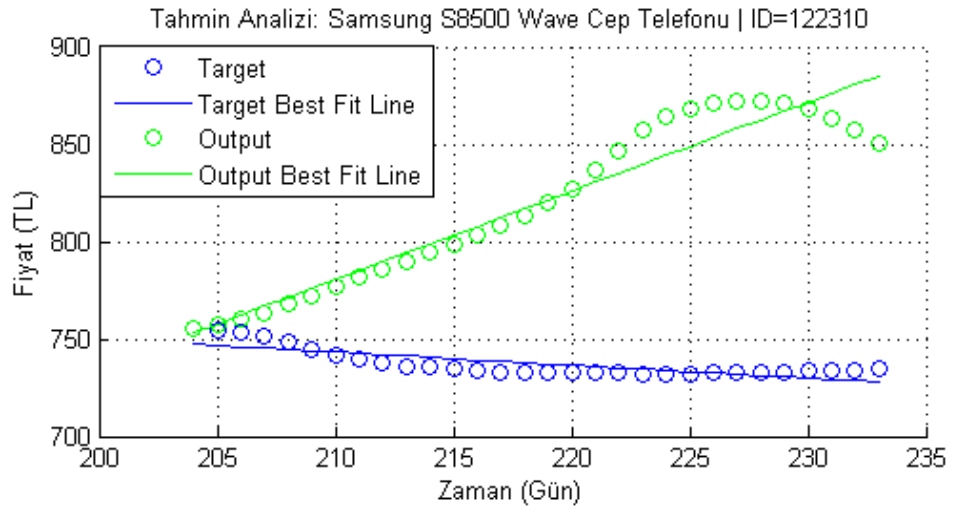
Şekil 4.56 : Sonuç Analizi 28

Kullanılan Veri Sayısı	158
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

Tablo 4.28 : Analiz sonucu 28



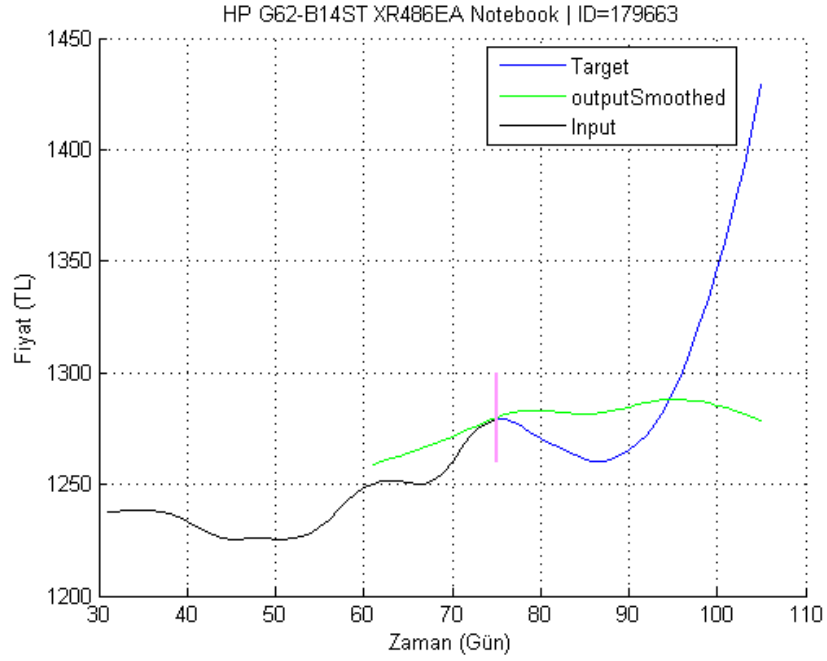
Şekil 4.57 : Sonuç 29



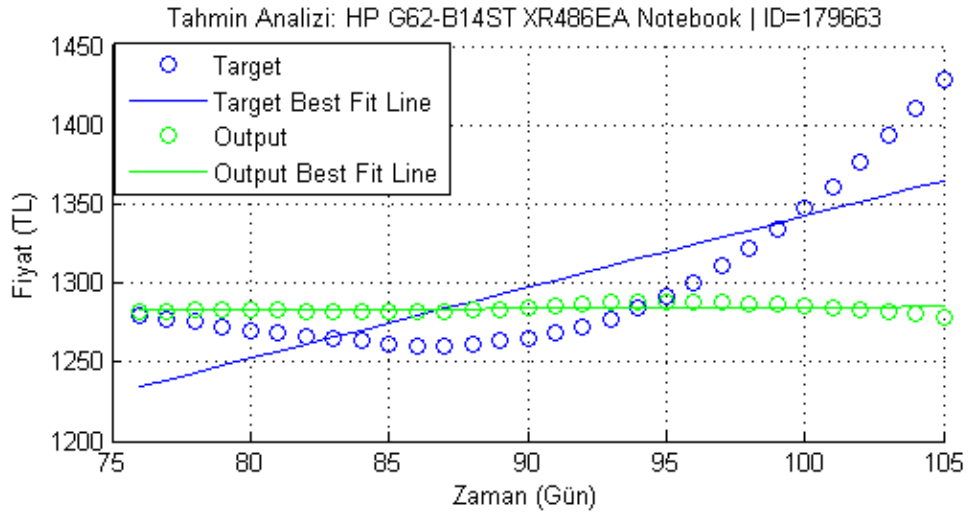
Şekil 4.58 : Sonuç Analizi 29

Kullanılan Veri Sayısı	203
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Düşüş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarısız
Not.	

Tablo 4.29 : Analiz sonucu 29



Şekil 4.59 : Sonuç 30



Şekil 4.60 : Sonuç Analizi 30

Kullanılan Veri Sayısı	75
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	Kullanılan veri sayısı düşük ve yetersiz.

Tablo 4.30 : Analiz sonucu 30



## 5. TARTIŞMA VE SONUÇ

Bir önceki bölümde denemesi yapılan veriler ve sonuçlar incelendiğinde veri grafiğinde keskin çizgileri olmayan veri tipleri için sistemin daha mantıklı sonuçlar ürettiği gözlemlenmiştir. Keskin çizgilere sahip olan grafiklerde, yapay sinir ağında eğitim aşaması, istenilen performansı yakalayamamakta olup, belirlenmiş olan 500 eğitim döngü limitini (max epoch=500) genelde aşıyor ve dolayısı ile uygun bir sonuç üretemiyor.

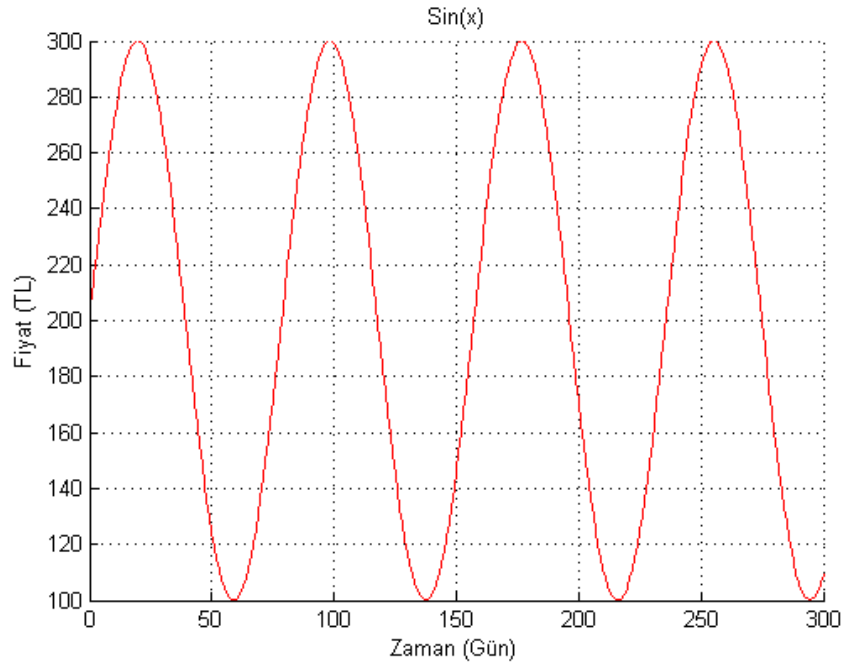
Keskin köşesi olmayan grafikler için sistem genelde eğitim aşamasında maksimum döngü sayısına (max epoch'a) ulaşmadan istenilen performans aralığında eğitimi tamamlayabiliyor. Tahmin Sonuçları bunlarda gerçeğe daha yakın gözüküyor. Burada bir tahmin yürütüldüğü için sonuçlar hiçbir zaman gerçek sonuçlarla aynı olamıyor, ama en azından fiyatların ileride nasıl olacağı hakkında bir fikir verebilir.

Eğitim aşamasında performansı yakalama konusunda, ağırlık kat sayılarının ilk random değerleri de etkilidir.

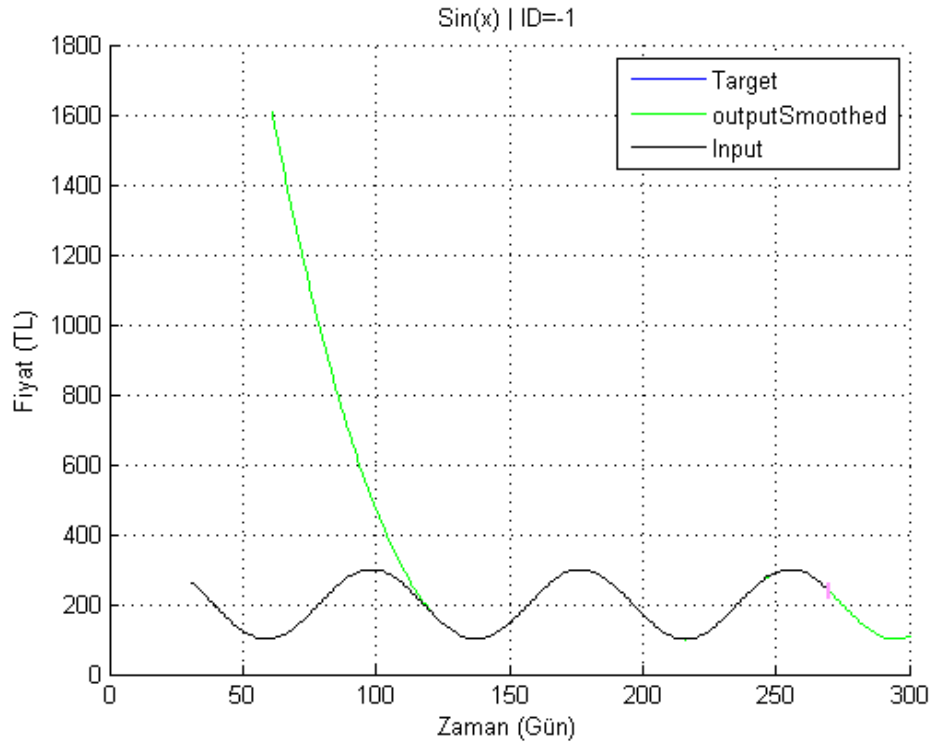
Yeterli sayıda veriye sahip ve keskin iniş çıkışları olmayan denenen örneklerin sonucunu, düşüş veya artışı göstermeyi önemseyerek değerlendirirsek, yaklaşık olarak %65 oranında sistemin başarılı sonuç verdiği gözüküyor. Bu değerlendirme çıkış ve gerçek (Target) veriler için bulunan “en iyi uyum hattı”nın (best fit line), eğilimine göre yapılmıştır. Bulunan “en iyi uyum hattı”nın eğilimi target ve çıkış vektörleri için aynı yönde ise sonuç başarılı ve diğer koşullarda sonuç başarısız olarak değerlendirilmiştir. Bu değerlendirme bulgular bölümündeki denenen 30 farklı fiyat verisi için yapılmıştır ve sistemin 20 adet başarılı sonuç ve 10 adet başarısız sonuç ürettiği gözükmektedir.

Bu sistemin mantıklı sonuçlar ürettiğini denemek için son olarak özel bir veri tipinde de bu deneme yapıldı. Bunun için Sinüs ve kosinüs fonksiyonlarının davranışına sahip veri dizileri üretildi. Bu dizilerin şekline baktığımız zaman ilerde nasıl olacaklarını basit bir mantıkla tahmin edebiliriz. Sisteme sinüs ve kosinüs fonksiyonları tipinde olan bu veriler verildiğinde, beklenildiği gibi mantıklı bir sonuç ürettiği gözüküyor. Bu sonuçlar bu bölümün devamındaki şekillerde gösterilmiştir.

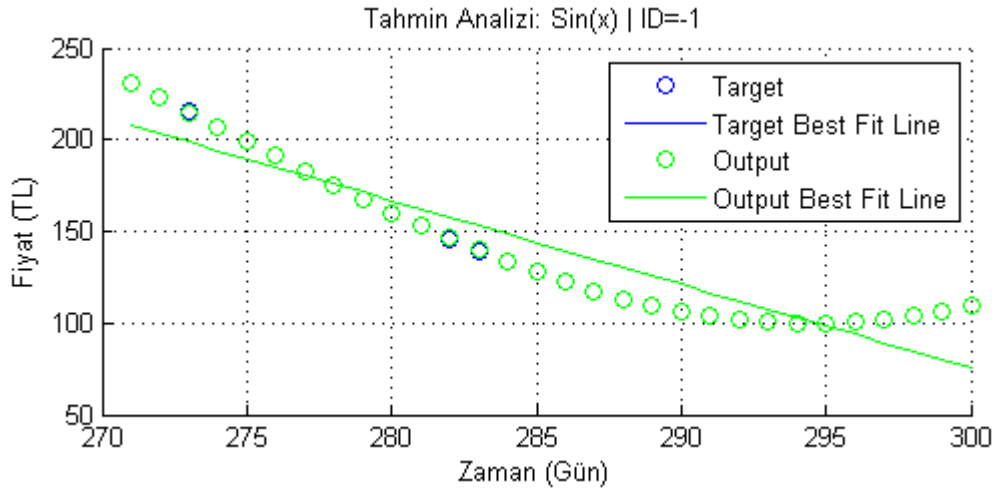
Şekil 5.2’de yeşil çizgi ile belirtilen sistemin çıkışı, *Target*’e birebir eşit çıktığı için, grafikte *Target* vektörüne ait mavi çizginin üzerini kapatmış olup mavi çizgi gözükmemektedir.



Şekil 5.1 : Sinüs fonksiyonu tipinde olan bir fiyat geçmişi vektörü.



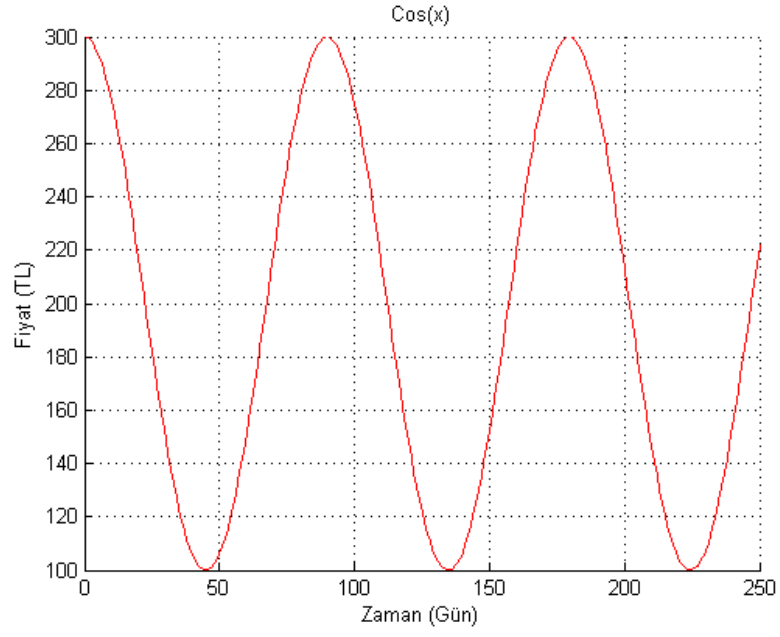
Şekil 5.2: Sinüs fonksiyonu tipinde olan giriş vektörüne ait sonuç



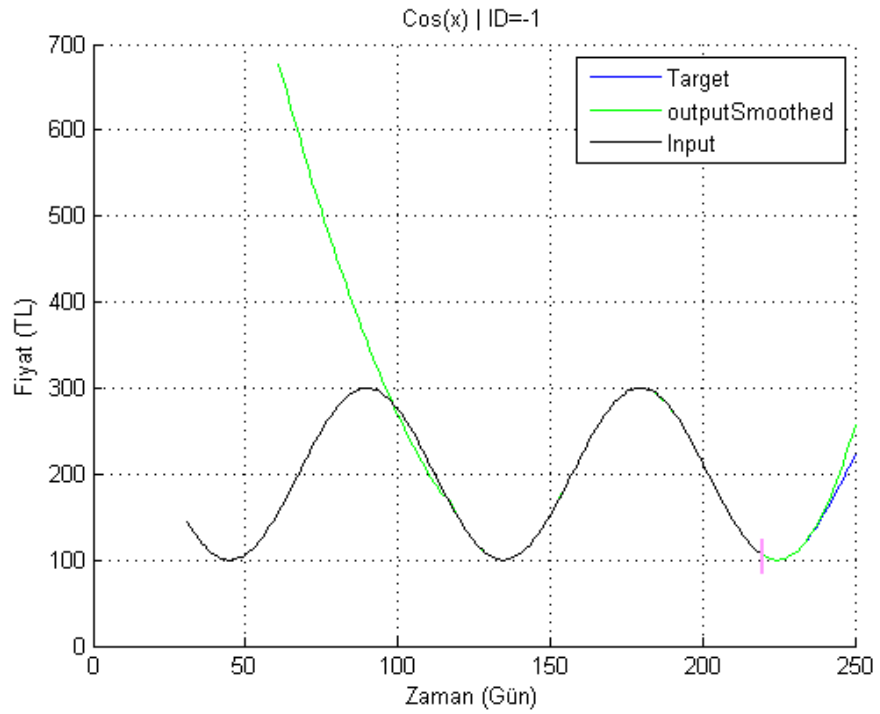
Şekil 5.3 : Sinüs fonksiyonu tipinde olan giriş vektörüne ait sonuç analizi

Kullanılan Veri Sayısı	270
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Düşüş
Çıkış Eğilimi	Düşüş
Tahmin Sonucu	Başarılı
Not.	Çıkış ve Target'in aynı çıkması nedeni ile Target'e ait mavi çizgi gözükmemektedir.

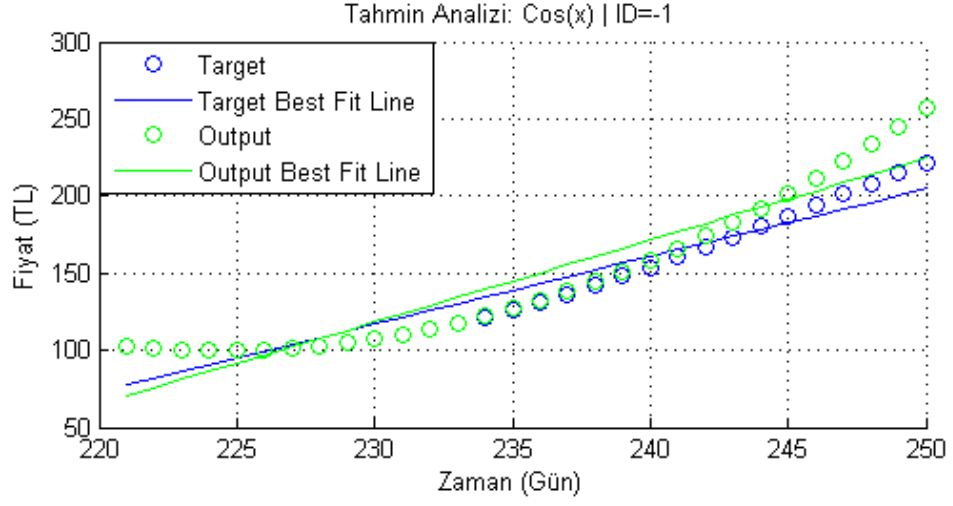
Tablo 5.1 : Sinüs fonksiyonu tipinde olan giriş vektörüne ait analiz sonucu



Şekil 5.4 : Kosinüs fonksiyonu tipinde olan bir fiyat geçmişi vektörü.



Şekil 5.5: Kosinüs fonksiyonu tipinde olan giriş vektörüne ait sonuç



Şekil 5.6 : Kosinüs fonksiyonu tipinde olan giriş vektörüne ait sonuç analizi

Kullanılan Veri Sayısı	220
Eğitim Performansı yakalandı mı	Evet
Target Eğilimi	Yükseliş
Çıkış Eğilimi	Yükseliş
Tahmin Sonucu	Başarılı
Not.	

Tablo 5.2 : Kosinüs fonksiyonu tipinde olan giriş vektörüne ait analiz sonucu

## KAYNAKLAR

1. OBITKO, Marek, 1999, *Neural Network Prediction* [online], Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, <http://www.obitko.com/tutorials/neural-network-prediction/> [Ziyaret Tarihi: 12 Mart 2010].
2. ÇAYLI, Hüseyin, 2008, *Finansal Tahminde Yapay Sinir İletim Ağlarının Kullanımı ve Bir Örnek Uygulama*, Yüksek Lisans Tezi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Ankara Üniversitesi.
3. RIOS, Daniel., 2007, *The Adaptive Resonance Theory: ART* [online], <http://www.learnartificialneuralnetworks.com/art.html> [Ziyaret Tarihi: 2 Haziran 2010].
4. HEATON, Jeff., 2009, *A Brief Summary of Neural Network Types* [online], <http://www.heatonresearch.com/content/brief-summary-neural-network-types> [Ziyaret Tarihi: 2 Haziran 2010].
5. P. JUNIOR, Jose Maria., A. BARRETO, Guilherme. 2007, Long-Term Time Series Prediction with the NARX Network: An Empirical Evaluation, *submitted to Elsevier Science. 9 March 2007*, 1-14.
6. LAWRENCE, Ramon., December 12, 1997, *Using Neural Networks to Forecast Stock Market Prices*, Department of Computer Science, University of Manitoba.
7. Mathworks, Matlab Product Help And Online Documentation, *NARX Network (narxnet, closeloop) : Dynamic Networks (Neural Network Toolbox)* [online], <http://www.mathworks.com/help/toolbox/nnet/ug/bss36dv-1.html> [Ziyaret Tarihi: 3 Kasım 2010].
8. Mathworks, Matlab Product Help And Online Documentation, *Dynamic Networks (Neural Network Toolbox)* [online], <http://www.mathworks.com/help/toolbox/nnet/ug/bss35yt.html> [Ziyaret Tarihi: 3 Kasım 2010].
9. WHITE, Halbert., 2007, *Economic Prediction Using Neural Networks: The Case Of IBM Daily Stock Returns*, Department of Economics, University of California, San Diego.

10. BOD'EN, Mikael., November 13, 2001, A guide to recurrent neural networks and backpropagation, School of Information Science, Computer and Electrical Engineering, Halmstad University.
11. JABBOUR, Georges., MALDONADO, Luciano., Prediction of financial time series with Time-Line Hidden, Markov Experts and ANNs, Escuela de Ingeniería de Sistemas e Instituto de Estadística Aplicada y Computación Universidad de Los Andes, Núcleo La Hechicera, Issue 9, Volume 4, Edif. B, 2° piso, Ala Sur, Mérida 5101, VENEZUELA ,September 2007.
12. CHAN Man-Chung, WONG Chi-Cheong, Financial Time Series Forecasting by Neural Network, Using Conjugate Gradient Learning Algorithm and Multiple Linear Regression Weight Initialization, LAM Chi-Chung, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong.

## EKLER

### EK-1 : MATLAB ORTAMINDA GELİŞTİRİLEN UYGULAMANIN KAYNAK KODU

```
%-----  
% Price Prediction Analysis Using Neural Networks  
% By Mehdi ALIZADEH  
% Supervisor: Prof. Dr. Sabri ARIK.  
% Istanbul University - 2011  
%-----  
  
clc;  
clear;  
  
%% request productID FROM user  
  
defaultProductID={'0'};  
  
inp = inputdlg('Enter ProductID','ProductID  
Input',1,defaultProductID);  
productID=str2num(inp{1});  
  
%% Load Data  
  
if( strcmp(inp{1},'sin') )  
  
    % create sinus price format:  
    X=(1:300)  
    prices=200+100*sin(X*0.08)
```



```

        prices =prices'
        productName='Sin(x)';
        productID=-1;

elseif( strcmp(inp{1},'cos') )

        % create cosinus price format:
        X=(1:250)
        prices=200+100*cos(X*0.07)
        prices =prices'
        productName='Cos(x)';
        productID=-1;

elseif(productID>0)

conn=database('myneuralnet','root','','com.mysql.jdbc.
Driver','jdbc:mysql://localhost:3306/myneuralnet')
        run_query=exec(conn,sprintf('SELECT
minPrice_history FROM `productspricehistory` WHERE
productID=%d',productID))

        result=fetch(run_query)
        prices_str=result.data{1}

        run_query_name=exec(conn,sprintf('SELECT
productNameTitle FROM `product` WHERE
productID=%d',productID))
        result_name=fetch(run_query_name)
        productName=result_name.data{1}

close(conn);

prices = strread(prices_str,'%d','delimiter',' ,')

```

```

    prices =prices'

    if(length(prices)<1)
        fprintf('Failed to import data from mysql.
check out the correct productID, your entered
productID=%d \n',productID);
        return
    end
else
    load('prices');

    productName='';
    productID=0;
end

% reverse the arrays: alternative for reversing:
x(end:-1:1)
prices=fliplr(prices)

dataCount=length(prices)

%% Plot Raw Data
figure(1);
clf;
hold on;
grid on;
xlabel('Time (days)');
ylabel('Product Price (TL)');

```

```

%plot Raw and not revised data
%plot (prices, 'DisplayName', 'prices', 'YDataSource',
'prices'); figure(gcf)

%delete Zeros Function Goes Here ro revise data
prices=reviseData(prices);

%plot revised data
%plot (prices, 'DisplayName', 'prices', 'YDataSource',
'prices'); figure(gcf)

%% smoothing
span=19; % span must be odd. %length(prices)/25;

%eger sin(x) veya cos(x) ise smooth yapilmamasi icin
burada if() eklendi
if(productID>0 || productID==0)%if sin or cos don't do
smoothing

    prices=smooth(prices,span,'rloess');
end

plot (prices, 'r'); figure(gcf)
title(productName);

%% prepare data for network creation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

minRange=round(min(prices)*0.70)

```

```

maxRange=round(max(prices)*1.30)

delay=30;

data_prior=prices(1:(end-delay))
data_delayed=prices((delay+1):end)
data_prior=data_prior'
data_delayed=data_delayed'

data_prior = con2seq(data_prior);
data_delayed = con2seq(data_delayed);

p = data_prior((1):(end-delay)) % training input
T = data_delayed((1):(end-delay)) % training target

% create network:
d1 = [1:delay]; d2 = [1:delay];
narx_net = newnarxsp(p,T,d1,d2,[10 10
1],{'tansig','tansig','purelin'});
narx_net.trainFcn = 'trainlm';
narx_net.trainParam.show = 10;
narx_net.trainParam.goal = 0.01;

narx_net.trainParam.epochs = 500;

view(narx_net);
% create initial delays:
for k=1:delay,
    Pi{1,k}=data_prior{k};
end

```

```

for k=1:delay,
    Pi{2,k}=data_delayed{k};
end

% train network:
narx_net = train(narx_net,[p;p],T,Pi);

% simulate network response:
output1 = sim(narx_net,[p;p],Pi);

% evaluate error:
e = cell2mat(output1)-cell2mat(T);
figure(3);
clf;

plot(e);
title('Training Error');
grid on;

% yapay sinir agin 1. katman bilgilerini gosterir
narx_net.layers{1}

%% prediction:-----

%- alttaki kodla serial-paralel narx networku parallel
%narxa cevirebiliriz.
%narx_net2 = sp2narx(narx_net);
%view(narx_net2);
%-----
prediction_input=data_prior(delay+1:end)
prediction_target=data_delayed(delay+1:end)

```

```

%Pi1: Initial input delay conditions
for k=1:delay,
    Pi1{1,k}=prediction_input{k};
    Pi1{2,k}=prediction_target{k};
end

%prediction_input

prediction_output =
sim(narx_net,[prediction_input;prediction_input],Pi1);

%eger parallel form kullanilirs a narx_net2
calistirilmalidir
%prediction_output =
sim(narx_net2,[prediction_input;prediction_input],Pi1)
;

size(prediction_output)
%output smoothing
prediction_output_smoothed=smooth(cell2mat(prediction_
output),span,'rloess');

%% PLOT OUTPUT
figure(2);clf;
hold on
grid on;
xlabel('Time (days)');
ylabel('Product Price (TL)');

```

```

plot((2*delay+1:dataCount),cell2mat(prediction_target)
', '-b')

%figure(2);
%smooth olmamis outputu bu satiri acarak gorebiliriz
%plot((2*delay+1:dataCount),cell2mat(prediction_output
)', '-r')

figure(2);
plot((2*delay+1:dataCount), (prediction_output_smoothed
)', '-g')

figure(2);
plot((delay+1:(dataCount-
delay)),cell2mat(prediction_input), '-k')

hleg1 = legend('Target','outputSmoothed','input');
title([productName ' | ID=' num2str(productID)]);

%dikey bi cizgi ile tahmin baslangicini belirlemek
icin
line([dataCount-delay ; dataCount-delay]
, [prediction_output_smoothed(end-delay)-20 ;
prediction_output_smoothed(end-delay)+20]
, 'Color', [1 .6 1], 'LineWidth', 2)
hold off;

%-----
%% Sonuç Analizi:
mat_prediction_target= cell2mat(prediction_target);
mat_prediction_target= mat_prediction_target(end-
delay+1:end);

```

```

mat_prediction_output_smoothed=
prediction_output_smoothed';
mat_prediction_output_smoothed=
mat_prediction_output_smoothed(end-delay+1:end);

%BEST FIT LINE
x=(dataCount-delay+1:dataCount);
target_best_fit_line =
polyfit(x,mat_prediction_target,1);
output_best_fit_line =
polyfit(x,mat_prediction_output_smoothed,1);
f_target = polyval(target_best_fit_line,x);
f_output = polyval(output_best_fit_line,x);
figure(8);clf;
hold on;
grid on;
xlabel('Zaman (Gün)');
ylabel('Fiyat (TL)');
plot(x,mat_prediction_target,'ob',...
      x,f_target,'-b',...
      x,mat_prediction_output_smoothed,'og',...
      x,f_output,'-g');
hleg2 = legend('Target','Target Best Fit
Line','Output','Output Best Fit Line');
title(['Tahmin Analizi: ' productName ' | ID='
num2str(productID)]);

fid = fopen([num2str(productID) '.txt'], 'w');
fprintf(fid, ['Tahmin Analizi: ' productName ' | ID='
num2str(productID)]);

```



```

fprintf('\nÜrün Adı: %s\n',productName);
fprintf(fid,'\r\nÜrün Adı: %s\n',productName);
fprintf('\nKullanılan Veri Sayısı: %d\n',dataCount-
delay);
fprintf(fid,'\r\nKullanılan Veri Sayısı:
%d\n',dataCount-delay);

if( tr.perf(end) <= narx_net.trainParam.goal )
    fprintf('\nEğitim Performansı yakalandı mi:
Evet\n');
    fprintf(fid,'\r\nEğitim Performansı yakalandı mi:
Evet\n');
else
    fprintf('\nEğitim Performansı yakalandı mi:
Hayır\n');
    fprintf(fid,'\r\nEğitim Performansı yakalandı mi:
Hayır\n');
end

%Eğilim Hesaplama
if( target_best_fit_line(1)>0 )
    TargetEğilim= 1;% 'Up'
    fprintf('\nHedef Eğilimi: Yükselis\n');
    fprintf(fid,'\r\nHedef Eğilimi: Yükselis\n');
elseif( target_best_fit_line(1)<0 )
    TargetEğilim= -1;% 'Down';
    fprintf('\nHedef Eğilimi: Düsüs\n');
    fprintf(fid,'\r\nHedef Eğilimi: Düsüs\n');
else
    TargetEğilim= 0;% 'None';
    fprintf('\nHedef Eğilimi: Yok\n');

```

```

        fprintf(fid, '\r\nHedef Egilimi: Yok\n');
end
%-----
if( output_best_fit_line(1)>0 )
    OutputEgilim= 1;% 'Up'
    fprintf('\nÇikis Egilimi: Yükselis\n');
    fprintf(fid, '\r\nÇikis Egilimi: Yükselis\n');
elseif( output_best_fit_line(1)<0 )
    OutputEgilim= -1;% 'Down';
    fprintf('\nÇikis Egilimi: Düsüs\n');
    fprintf(fid, '\r\nÇikis Egilimi: Düsüs\n');
else
    OutputEgilim= 0;% 'None';
    fprintf('\r\nÇikis Egilimi: Yok\n');
end

if(OutputEgilim == TargetEgilim )
    fprintf('\nTahmin Sonucu: Basarili\n');
    fprintf(fid, '\r\nTahmin Sonucu: Basarili\n');
else
    fprintf('\nTahmin Sonucu: Basarisiz\n');
    fprintf(fid, '\r\nTahmin Sonucu: Basarisiz\n');
end

fclose(fid);
% program end

```

Fonksiyon: reviseData() tanımlaması:

```
%replaces zero values in array
function reviseData=reviseData(data)
    len=length(data);
    if(len>=3 && data(1)>0 )

        for i=2:(len-1)
            if( data(i)==0 && data(i+1)>0 )
                data(i)= round((data(i-
1)+data(i+1))/2);
            elseif( data(i)==0 )
                data(i)= data(i-1);
            end
        end %end for

        if(data(end)==0)
            data(end)= data(end-1);
        end
        elseif(len<3)
            printf('\nError in reviseData: array length
should be more than 2 elements!');
        elseif(data(1)==0)
            printf('\nError in reviseData: first element
of array can not be zero!');
        end %end if

    reviseData=data;
    % reviseData fonkisyon bitis
```

## **EK-2 : ZAMAN SERİSİNİ DİLİMLERE AYIRMA YÖNTEMİ İÇİN YAZILAN MATLAB KODU (BU YÖNTEM KULLANILMAMIŞTIR):**

Not: Bu yöntem alternatif bir model olarak sadece denenmiştir ve bu projede kullanılan yöntem değildir.

```
%% Process input data and make Time Data Slices

TimeSliceSize=30;
if(dataCount<(3*TimeSliceSize))
    fprintf('The input data count is less than
required size. the minimum size required is
%d', (2*TimeSliceSize));
    return
end

sliceCount=1;
col=TimeSliceSize;
for i=dataCount:-1:1

    if(col==0)
        if( i< TimeSliceSize )
            break;
        end
        col=TimeSliceSize;
        sliceCount=sliceCount+1;
    end
    TimeSlices(sliceCount,col)=prices(i);
    col=col-1;
end

%% Prepare data for network creation:
delay=30;% one month
```

```

%prepare training input and target
j=1;
for i=sliceCount:-1:(sliceCount-1)
    u=TimeSlices(i,:);
    y=TimeSlices(i-1,:);
    u=u';
    y=y';
    y = con2seq(y); u = con2seq(u);
    p(1,j)=[u];
    t(1,j)=[y];
    j=j+1;
end

d1 = [1:delay]; d2 = [1:delay];
narx_net = newnarxsp(p,t,d1,d2
,[10 10 1],{'tansig','tansig','purelin'});
narx_net = train(narx_net,[p;p],t);

```

### **EK-3 : MATLAB'DEN MYSQL'E BAĞLANMAK**

Matlab programından mysqlle bağlanabilmek için önce jdbc sürücüsünü (driver) yüklemek gerekiyor. Bunun için şu adımların takip edilmesi gerekiyor:

- 1- jdbc driver dosyasını bu adresten indirebiliriz:  
<http://dev.mysql.com/downloads/connector/j/3.1.html>
- 2- [matlab path]/toolbox/local/ dizininde bulunan classpath.txt dosyasına şu satırı eklememiz lazım: [jdbc driver package unzip ettiğimiz dizin]/mysql-connector-java-3.1.12-bin.jar (dosya ismi versiyona göre değişebilir)
- 3- Şimdi işlemleri başarılı olmasını alttaki komut ile bir bağlantı denemesi yaparak kontrol edebiliriz:

```
conn=database('dbName','user','password','com.mysql.jdbc.Driver','jdbc:mysql://localhost:3306/dbName')
```

## ÖZGEÇMİŞ

Mehdi ALİZADEH 05.07.1983 tarihinde Tabriz/İran'da doğdu. İlk, orta ve lise eğitimini Tabriz'de tamamladı. 2001 yılında Tabriz Üniversitesi, Matematik Fakültesi, Bilgisayar Bilimleri bölümünde lisans eğitimine başladı. 2006 yılında mezun oldu. Lisans döneminde yapay sinir ağları ile çalışmaları oldu ve lisans projesinde bir classification projesi C++ ve MATLAB ortamında geliştirdi. 2008 yılın Ekim ayında İstanbul Üniversitesi, fen Bilimleri Fakültesi, Bilgisayar Mühendisliği Ana Bilim Dalında Yüksek Lisans eğitimine başladı.

Okul eğitimine paralel, 11 yaşından müzik eğitimini almaya başladı. 5 sene santoor müzik aleti ve ardından keman eğitimini İran klasik müziği dalında aldı.

İş hayatında ise web yazılım alanında online alışveriş sistemleri ve fiyat analiz siteleri programlama konusunda 5 senelik profesyonel deneyime sahiptir. Php/mysql uygulamalarında deneyimli ve linux işletim sistemleri ile çalışan sunucular hakkında deneyim sahibidir.

İngilizce, Farsça, Azerice ve Türkçe dillerine hakimdir.

Email: mehdi.alizadeh@gmail.com