

Bitcoin Fiyat Tahmini için ARIMA Zaman Serisi Modeli ve LSTM Derin Öğrenme Modelinin Karşılaştırılması

Ebru Şeyma Karakoyun, Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği, Konya, ebruseymakarakoyun@gmail.com

Ali Osman Çıbıkdiken, Necmettin Erbakan Üniversitesi, Bilgisayar Mühendisliği Bölümü, Konya, aacdiken@konya.edu.tr

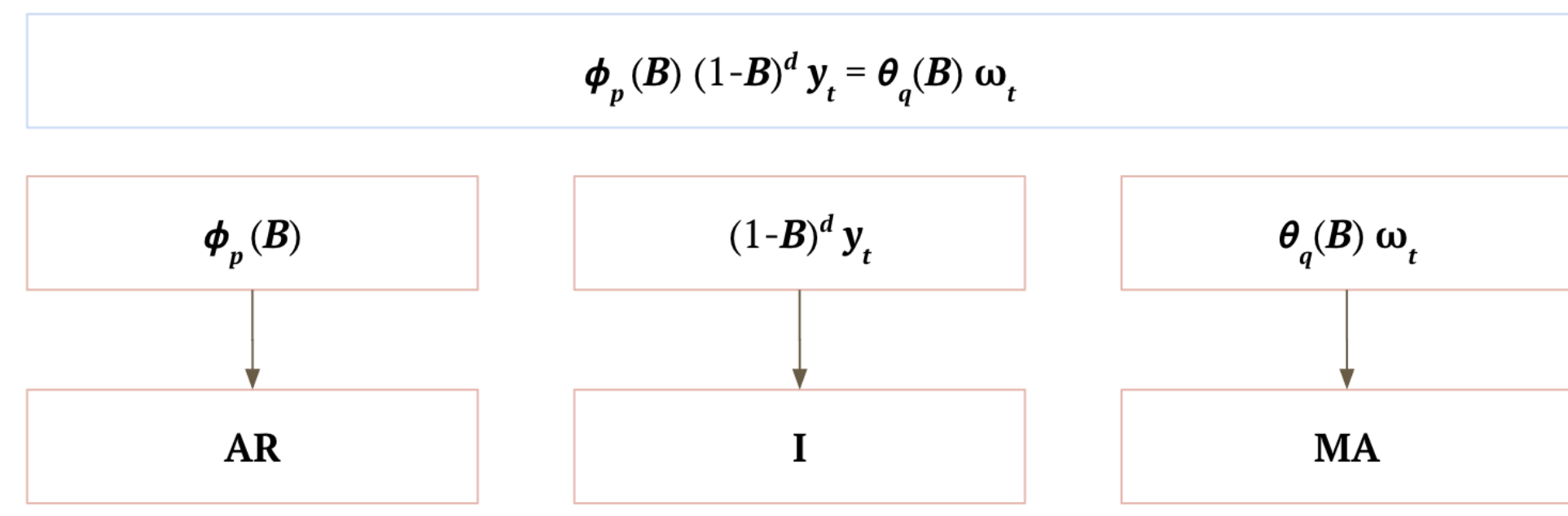
ÖZET

Bu çalışmada, ARIMA Zaman Serisi Modeli ve LSTM Derin Öğrenme Mimarisi, Bitcoin'in gelecekteki fiyatını tahmin etmek için karşılaştırılmıştır. Zaman serilerinin tahmininde yaygın olarak kullanılan ARIMA modeli R programlama dilinde ve LSTM modeli de Python'daki Keras framework kullanılarak oluşturulmuştur. ARIMA ve LSTM derin öğrenme modelleri kullanılarak Bitcoin'in gelecekteki 30 günün fiyatları tahmin edilmiştir. Sonuçlar yaklaşık olarak ARIMA için MAPE %11.86 ve LSTM için MAPE% 1.40 olarak elde edilmiştir. Sonuçlar karşılaştırıldığında LSTM Modelinin daha başarılı olduğu gözlenmiştir.

“Tahmin zordur, özellikle geleceğe aitse...” Nils Bohr

ARIMA

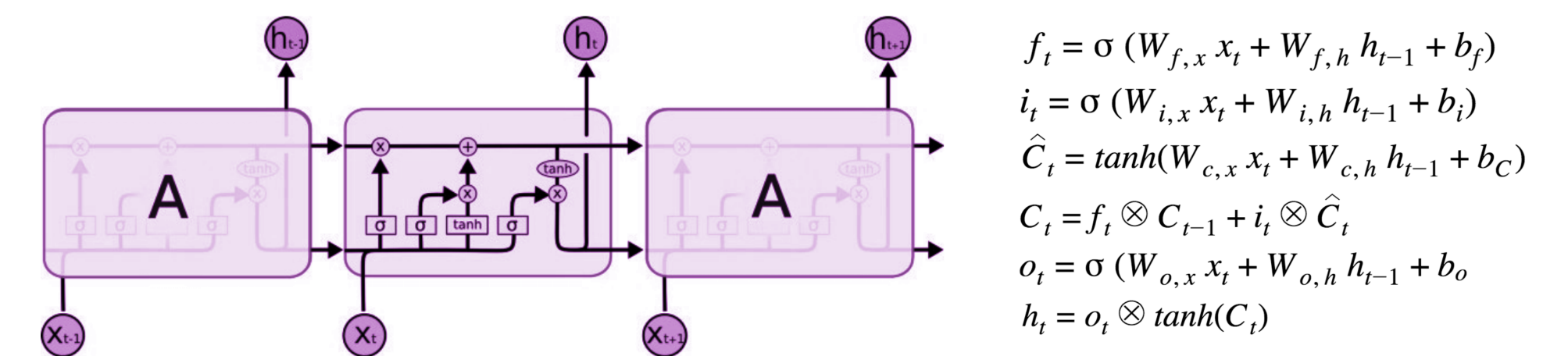
Otoregresif Bütünleşik Hareketli Ortalama Modeli (ARIMA) modeli, veri noktalarının sonlu farkları alınarak durağan olmayan bir zaman dizisi durağan hale getirilir ve ARIMA(p,d,q) şeklinde ifade edilir



Bitcoin, 2008 yılında Satoshi Nakamoto isimli gizemli bir geliştirici tarafından matematiksel kanıt ve kriptografik yöntemle dayanan bir elektronik ödeme sistemi olarak önerilmiştir. Önerilen ödeme sistemi herhangi bir merkezi otoriteden bağımsız olarak çalışabilir.

LSTM

Uzun - Kısa Süreli Hafıza (Long Short Term Memory - LSTM) uzun süreli öğrenme kapasitesine sahip bir Tekrarlamalı Sinir Ağı mimarisidir. Hochreiter ve Schmidhuber (1997) tarafından ortaya atılmıştır.



İstatistiksel

Derin Öğrenme

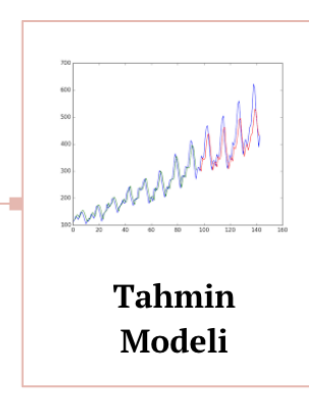
Veri Seti

ARIMA

LSTM

ARIMA Tahmin Modeli Adımları

Adım 0. Veri seti ve gerekli kütüphaneler yüklenir.
Adım 1. Zaman serisi verisi oluşturulur. Zaman serisi grafiği çizilir.
Adım 2. Durağanlık kontrol edilir.
Adım 3. Mevsimsellik kontrol edilir.
Adım 4. ARIMA modeli elde edilir.
Adım 5. Tahmin modeli oluşturulur.
Adım 6. Eğitim verileri oluşturulur ve veriler test edilir.
Adım 7. Test verilerinin doğruluğu için performans ölçümü yapılır.



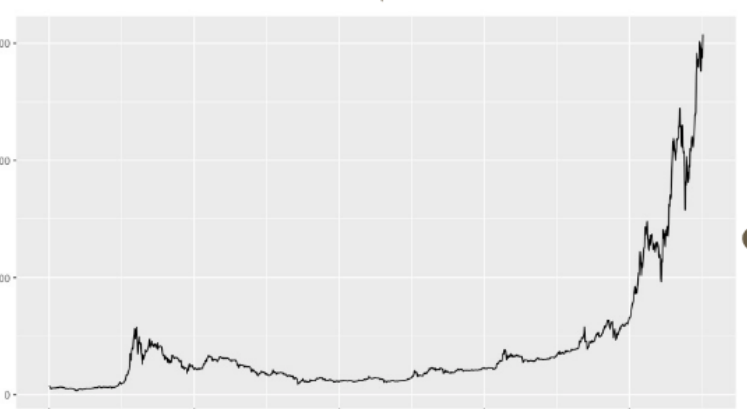
LSTM Tahmin Modeli Adımları

Adım 0. Zaman serisi veri seti yüklenir.
Adım 1. Derin öğrenme algoritması için ağ modeli oluşturulur.
Adım 2. Veri seti 30 günlük zaman serisi şeklinde düzenlenir.
Adım 3. Veri setinin %20 si test için, geri kalanı eğitim verisi için ayrılır.
Adım 4. Eğitim verileri kullanılarak model eğitilir.
Adım 5. Eğitilen model üzerinde test verileri tahmin edilir.
Adım 6. Test sonucu hata oranları ekrana yazdırılır.

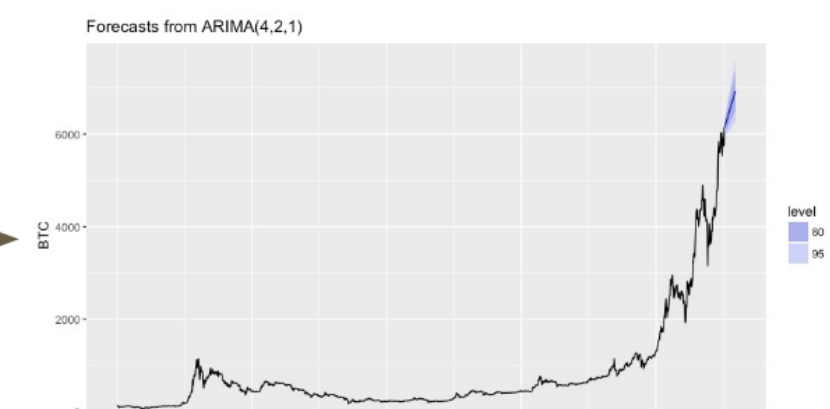
ARIMA Tahmin Modeli

Bitcoin Fiyatı
28.4.2013-29.10.2017
1646 adet günlük fiyat

RMSE	MAE	MPE	MAPE
1146.067	939.5819	10.86483	11.86484



ARIMA(4,2,1)



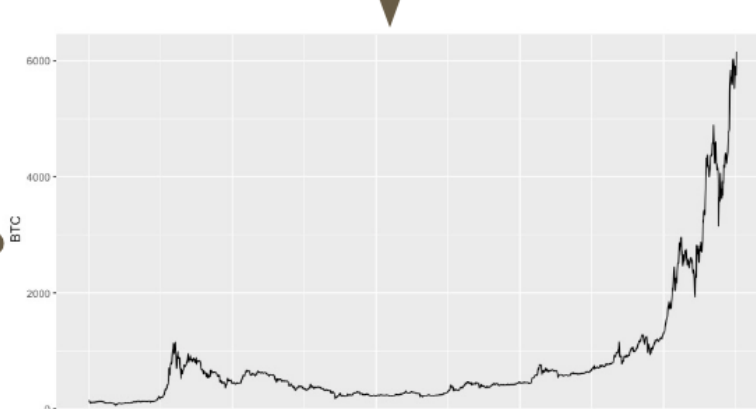
LSTM Tahmin Modeli

RMSE	MAE	MPE	MAPE
93.27	81.56	1.40	1.40



LSTM(6L)

Bitcoin Fiyatı
28.4.2013-29.10.2017
1646 adet günlük fiyat



ARIMA(4,2,1)

```
> library(forecast)
> library(ggplot2)
> library(tseries)
> library(readr)
> btc_usd_gunluk <- read_csv("btc_usd_gunluk.csv")
> BTC <- ts(btc_usd_gunluk,frequency=365)
> kpss.test(BTC)
> autoplot(BTC)
> TestMevsimsellik <- tbats(BTC)
> Mevsimsellik <- !is.null(TestMevsimsellik$seasonal)
> Mevsimsellik
> Model<-auto.arima(Apple,seasonal=FALSE,
stepwise=FALSE,approximation=FALSE,trace=TRUE)
> Model
> autoplot(Model)
> Tahmin_30gun <- forecast(Model,h=30)
> autoplot(Tahmin_30gun)
> Egitimverisi <- data.frame(Tahmin_30gun)
> Egitimverisi <- Egitimverisi$Point.Forecast
> Testverisi <- c(6158.76, 6115.15, 6411.84, 6694.38, 7069.03, 7199.62,
7369.08, 7400.39, 7025.14, 7167.19, 7447.27, 7154.71, 6622.42, 6368.32,
5852.81, 6527.20, 6714.17, 7280.20, 7814.49, 7738.42, 7833.45, 8012.64,
8225.41, 8132.54, 8260.71, 8054.45, 8239.31, 8706.60, 9207.99, 9713.31)
> accuracy(Egitimverisi,Testverisi)
```

Keras

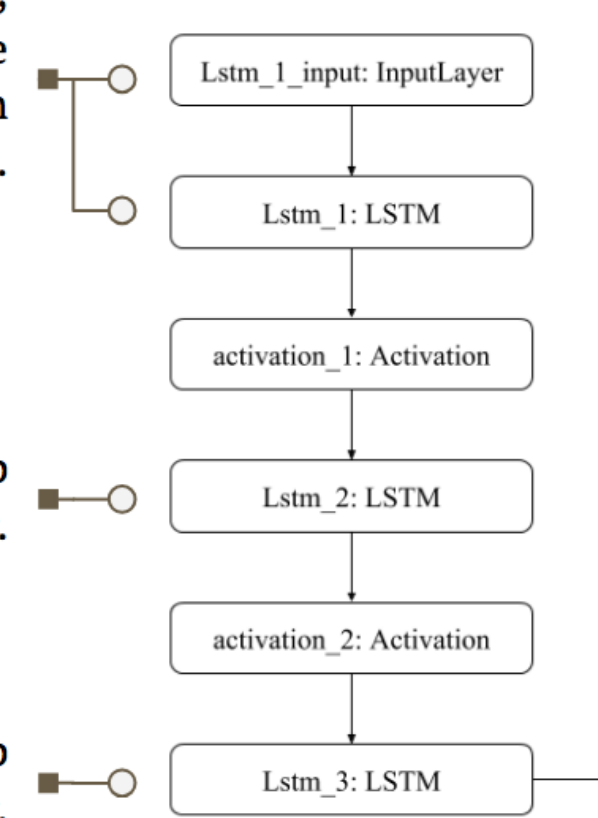
LSTM(6L)

Optimizasyon algoritması olarak ADAM kullanılmıştır. Algoritma üzerinde öğrenme oranı olarak 0.001 tercih edilmiştir. Öğrenme esnasında geri besleme işlemi (epoch) 1000 kez tekrarlanmıştır.

İlk LSTM katmanı, giriş katmanları ile tam olarak bağlanan 64 nörona sahiptir.

128 nörona sahip LSTM katmanındır.

128 nörona sahip LSTM katmanındır.



64 MLP nöronu vardır.

64 MLP nöronu vardır.

Tüm veri seti 0-1 arasında ölçeklendirildiği için çıkış katmanının değeri de tersine çevrilmiştir.