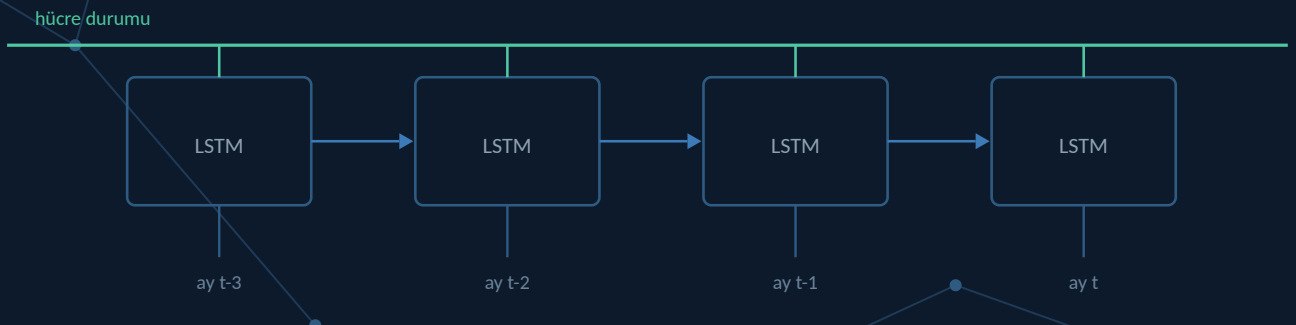
BCDataWorks

SIRALI VERİ MODELLERİ

LSTM

Long Short-Term Memory



· Hücre anatomisi · Veri kurgusu · Bankacılıkta kullanım

Bu doküman ne anlatıyor?

LSTM, sıralı veriyi işlemek için tasarlanmış bir sinir ağı hücresidir. Tek bir fikir üzerine kuruludur: **bilgiyi taşıyan ayrı bir kanal aç**, o kanala neyin gireceğine ve neyin kalacağına küçük kapılar karar versin. Bu doküman kapıların ne yaptığını, verinin nasıl şekillendirileceğini ve bankacılık analitiğinde LSTM'in gerçekten kazandırdığı yerleri gösterir.

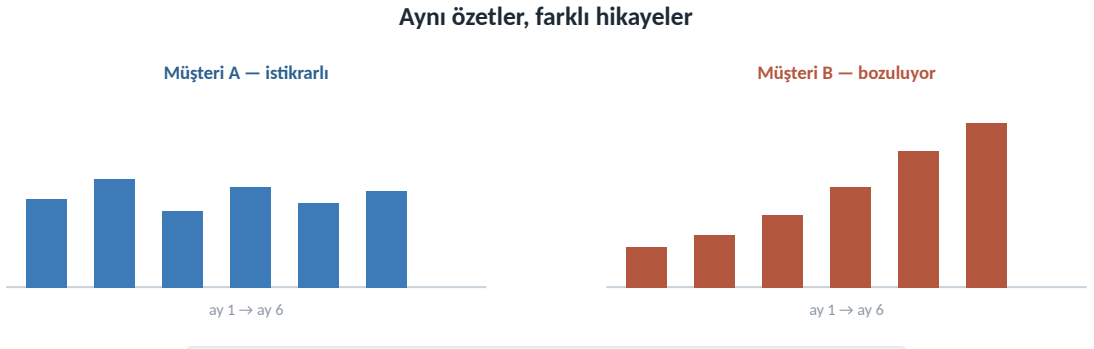
> **Üç cümlede LSTM**
>
> **Ne çözer?** Temel RNN'in uzun dizilerde gradyanı kaybetmesini. Hücre durumu, bilgiyi çarpımlar yerine toplamalarla taşır; gradyan bu yüzden sönmeden geriye akabilir.
>
> **Nasıl çalışır?** Üç kapı — unut, gir, çıkar — her adımda neyin saklanacağına ve neyin dışarı verileceğine karar verir. Kapıların hepsi öğrenilir, elle kurulmaz.
>
> **Ne zaman gerekir?** Sıranın kendisi bilgi taşıdığında. Sıra bilgi taşımıyorsa veya birkaç özetle temsil edilebiliyorsa, gradient boosting neredeyse her zaman daha ucuz ve daha savunulabilir bir cevaptır.

İçindekiler

Anlatım boyunca **tabular ve zaman serisi bankacılık verisi** esas alınmıştır; doğal dil ve ses uygulamalarına yalnız karşılaştırma gerektiğinde değinilir. Kod örnekleri Keras sözdizimiyle yazılmıştır, PyTorch karşılıkları kavramsal olarak aynıdır. Şekillerdeki performans ve süre sayıları ise tipik davranışın yönünü göstermek için konulmuştur; ölçüm sonucu değildir ve kendi verinde doğrulanmadan kullanılmamalıdır.

01 Neden sıralı model?

Klasik yaklaşım, diziyi sabit sayıda özete indirger: son ay bakiyesi, 3 ay ortalaması, 6 aylık maksimum. Bu çoğu zaman yeterlidir. Yetmediği yer, **sıranın kendisinin bilgi taşıdığı** yerdir.



Çözüm her zaman LSTM değildir — eğim ve değişim değişkeni de bu farkı yakalar (13. bölüm).

Şekil 1. Sıra bilgi taşıyorsa, özet değişkenler bilgi kaybeder.

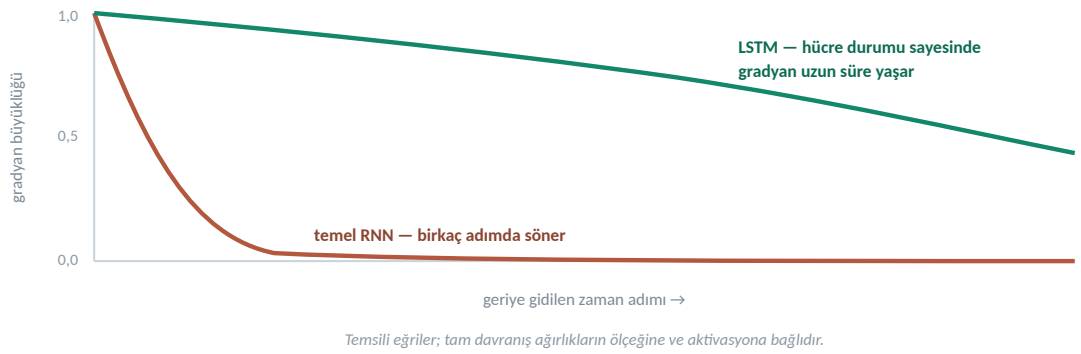
Sıralı modelin üç somut üstünlüğü

· **Değişken uzunluklu dizi.** 3 aylık müşteriyle 36 aylık müşteriyi aynı modelde, doldurma ve maskeleme ile işleyebilir. Sabit pencereli tabular kurguda bunun için ya kısaltma ya da eksik doldurma gerekir.
· **Paylaşılan ağırlıklar.** Aynı hücre her zaman adımında yeniden kullanılır; "ikinci ayda harcama artışı" ile "sekizinci ayda harcama artışı" aynı parametrelerle öğrenilir. Tabular kurguda her ay ayrı bir kolondur ve model aynı deseni her kolonda baştan öğrenmek zorundadır.
· **Uzun mesafeli bağımlılık.** Sekiz ay önceki bir olayın bugünkü davranışa etkisi, elle üretilmiş bir değişkende yakalanmamışsa tabular modelde kaybolur.

Ne var ki bankacılık verisinde bu üç üstünlüğün üçüne birden nadiren ihtiyaç duyulur. Aylık hesap özeti gibi **kısa ve düzenli** dizilerde, iyi tasarlanmış değişim değişkenleri LSTM'in çıkardığı bilginin büyük kısmını çok daha ucuza çıkarır. Sıralı modelin gerçekten öne geçtiği yer, dizinin **uzun, düzensiz ve olay bazlı** olduğu problemlerdir: işlem akışı, tıklama akışı, çağrı kayıtları. Bu ayrımın nasıl ölçüleceği 12. ve 13. bölümlerde ele alınıyor.

02 Temel RNN ve kaybolan gradyan

LSTM'in varlık sebebi tek bir teknik sorundur. Kapısız, eklentisiz tekrarlayan sinir ağında — kütüphanelerdeki karşılığı `SimpleRNN` katmanıdır — bilgi her adımda aynı ağırlık matrisiyle **çarpılarak** ilerler; yeterince çok adımda bu çarpım ya sıfıra iner ya da patlar.



Şekil 2. Sorun ezberleme değil, öğrenme sinyalinin geriye ulaşamamasıdır.

Neden oluyor?

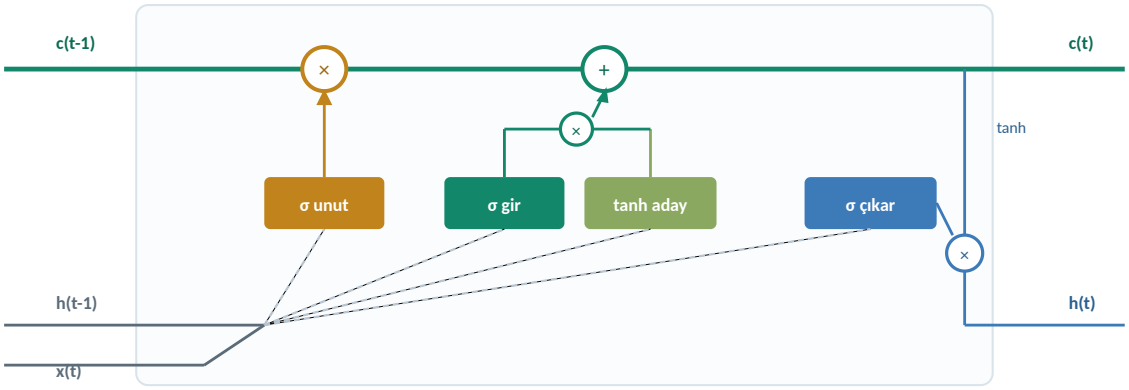
Geri yayılım, zincir kuralı gereği her zaman adımı için bir türev çarpar. Aynı matris otuz kez çarpıldığında öz değerleri 1'den küçükse sonuç üstel olarak sıfıra, 1'den büyükse üstel olarak sonsuza gider. Birincisi **kaybolan gradyan**, ikincisi **patlayan gradyan**dır. Patlayan gradyan görece kolay çözülür — kırpma (`clipnorm`) yeterlidir. Kaybolan gradyan ise sessizdir: model eğitilir, kayıp düşer, ama uzak geçmişten hiçbir şey öğrenilmemiştir.

> **LSTM'in çözümü: çarpmak yerine toplamak**
>
> LSTM, bilgiyi taşıyan hücre durumunu her adımda ağırlık matrisiyle çarpmaz; üzerine **toplama** yapar ve yalnız bir kapı ile ölçekler. Kapı açık kaldığı sürece bilgi ve gradyan neredeyse bozulmadan ilerler. Literatürde bu yapıya **sabit hata taşıyıcısı** (constant error carousel) denir ve LSTM'in tüm mimarisi bu tek fikrin etrafında kuruludur.

03 Hücrenin anatomisi

Bir LSTM hücresinde iki hafıza ve üç kapı vardır. Hepsi aynı iki girdiden beslenir: bir önceki gizli durum ve bu adımın verisi.

Bir zaman adımı içinde ne oluyor?



Dört kapının da girdisi aynıdır: h(t-1) ve x(t). Farklı olan öğrenilen ağırlıklardır.

Yesil yatay hat hücre durumudur; bilgi burada carpilmadan, toplanarak ilerler.

Şekil 3. İki hafıza (c ve h) ve üç kapı.

Bileşen	Görevi	Sezgisi
Hücre durumu c(t)	Uzun vadeli hafıza; kapılarla güncellenir	Defterin kendisi
Gizli durum h(t)	O adımda dışarı verilen çıktı	Defterin bu an okunan sayfası
Unutma kapısı	Eski bilginin ne kadarı kalsın?	Silgi
Giriş kapısı	Yeni bilginin ne kadarı yazılsın?	Kalem basıncı
Aday değer	Yazılabilecek yeni içerik	Yazılacak metin
Çıkış kapısı	Hafızanın ne kadarı dışarı verilsin?	Perde

04 Denklemler ve sezgisi

Altı satır. Hepsi aynı iki girdiden beslenir ve hepsinin öğrenilen ayrı bir ağırlık matrisi vardır.

$$f(t) = \sigma(\, W(f) \cdot [h(t\text{-}1),\, x(t)] + b(f) \,)$$ unutma kapısı · 0 ile 1 arası

$$i(t) = \sigma(\, W(i) \cdot [h(t\text{-}1),\, x(t)] + b(i) \,)$$ giriş kapısı

$$g(t) = \tanh(\, W(g) \cdot [h(t\text{-}1),\, x(t)] + b(g) \,)$$ aday değer · -1 ile 1 arası

$$o(t) = \sigma(\, W(o) \cdot [h(t\text{-}1),\, x(t)] + b(o) \,)$$ çıkış kapısı

$$c(t) = f(t) \odot c(t\text{-}1) + i(t) \odot g(t)$$ **asıl satır burası**

$$h(t) = o(t) \odot \tanh(\, c(t) \,)$$ dışarı verilen çıktı

Şekil 4. ⊙ eleman bazında çarpımdır; σ sigmoid, değerleri 0–1 aralığına sıkıştırır.

Beşinci satır neden kritik?

Hücre durumu güncellemesi bir **toplama**dır: eski değerin kapıyla ölçeklenmiş hali, artı yeni bilginin kapıyla ölçeklenmiş hali. Türev alındığında bu toplama, gradyanın geriye giden yolunu açık tutar — çünkü toplamanın türevi 1'dir. Temel RNN'de bu satırın yerinde bir matris çarpımı vardır ve gradyan her adımda o matristen geçmek zorundadır. Tüm fark budur.

Sigmoid neden kapı, tanh neden içerik?

Kapıların çıktısı 0 ile 1 arasında olmalıdır, çünkü bir **oran** ifade ederler: "bu bilginin %30'unu tut". Sigmoid tam olarak bunu verir. Aday değer ise bir **miktar**dır ve işaretli olmalıdır — hafızaya hem pozitif hem negatif katkı yapılabilmelidir; bu yüzden -1 ile 1 arasına oturan tanh kullanılır.

Unutma kapısı sapması: bilinmesi gereken tek başlangıç ayarı

Unutma kapısının sapma terimi (bias) 1 civarında başlatıldığında kapı eğitimin başında **açık** olur ve bilgi ilk turlardan itibaren geriye akabilir. Sıfırdan başlatılırsa kapı yarı kapalı davranır ve model uzun bağımlılıkları öğrenmeye çok daha geç başlar. Modern kütüphanelerde bu genellikle varsayılandır (Keras'ta `unit_forget_bias=True`), fakat hücreyi elle yazıyorsan atlanması kolay bir detaydır.

Parametre sayısı

Bir LSTM katmanının parametre sayısı `4 × ((girdi + gizli) × gizli + gizli)` formülüyle bulunur. 64 gizli birim ve 20 değişkenli bir girdi için bu yaklaşık **21.760** parametre demektir. Bu sayıyı elde ettiğin bad gözlem sayısıyla karşılaştırmak, mimariyi seçerken yapılabilecek en yararlı kontroldür — 8. bölüm.

05 GRU ve diğer alternatifler

LSTM sıralı modelleme dünyasının tamamı değildir. Aynı problemi çözen dört yaklaşım vardır ve bankacılık verisinde çoğu zaman en basiti yeterlidir.

	Ne yapar?	Ne zaman tercih edilir?
LSTM	Üç kapı, ayrı hücre durumu	Uzun ve karmaşık bağımlılıklarda; referans nokta budur.
GRU	İki kapı, tek durum; yaklaşık %25 daha az parametre	Veri sınırlıysa ve dizi kısaysa. Tabular bankacılık verisinde çoğu zaman LSTM ile ayırt edilemez.
1B Evrişim (CNN)	Kayan filtrelerle yerel desen yakalar	Kısa ve yerel desenler (ardışık 3 ayın şekli) yeterliyse. Çok hızlıdır, paralelleşir.
Transformer	Dikkat mekanizmasıyla tüm adımlara doğrudan bakar	Çok uzun diziler ve bol veri. Küçük veride LSTM'e üstünlük kurması zordur.
Tabular + değişim değişkeni	Diziyi elle özetler, GBDT'ye verir	Kısa ve düzenli dizilerde ilk tercih. 13. bölüm.

GRU'yu önce dene

Bankacılık dizileri genellikle kısadır (12–24 aylık hesap özeti) ve etiket sayısı sınırlıdır. Bu rejimde GRU'nun daha az parametreli olması bir dezavantaj değil, **düzenlileştirme**dir: daha hızlı eğitilir, daha az aşırı öğrenir ve çoğu zaman LSTM ile aynı performansı verir. LSTM'i ancak GRU'nun tavana vurduğunu gördüğünde dene.

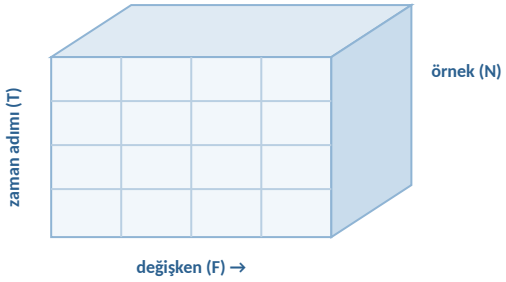
Çift yönlü (bidirectional) LSTM: dikkat

Çift yönlü katman diziyi hem ileri hem geri okur ve genellikle performansı artırır. Fakat bu, her zaman adımında **gelecekteki adımları da görmek** demektir. Tahmin problemlerinde (gelecek ay temerrüt, sonraki çeyrek nakit talebi) bu doğrudan sızıntıdır. Çift yönlü katman yalnız dizinin tamamının geçmişe ait olduğu problemlerde kullanılabilir: tamamlanmış bir işlem dizisinin sonradan sınıflandırılması gibi.

06 Girdi tensörü

LSTM ile çalışırken harcanan zamanın çoğu modelde değil, veriyi doğru şekle sokmakta geçer. Girdi her zaman **üç boyutludur**.

(örnek, zaman adımı, değişken)



Davranışsal risk örneği

N = 400.000 müşteri-ay gözlemi

T = 12 ay (son 12 hesap özeti)

F = 18 değişken (bakiye, harcama, ödeme oranı, gecikme, limit kullanımı…)

shape = (400000, 12, 18)

En sık hata: zaman adımı ile değişken eksenini yer değiştirmek. Model hata vermez, saçmalar.

Şekil 5. Şekli her zaman yazdır ve gözle doğrula.

Sessiz hatalar

· **Eksen karışması.** (N, F, T) verirsen kütüphane hata vermez; 18 zaman adımı ve 12 değişken varmış gibi öğrenir. Sonuç: anlamsız ama çalışan bir model.

· **Sıralamanın bozulması.** Veriyi müşteri ve tarih bazında sıralamadan pencerelemek, dizileri karıştırır. Kontrol: rastgele bir müşterinin dizisini elle yazdır, tarihleri artan mı bak.

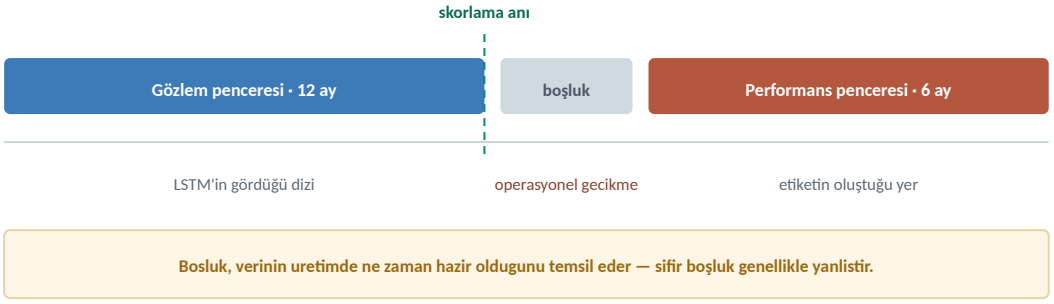
· **Kısa dizilerin doldurulması.** Yeni müşterilerde 12 ay yoktur. Doldurma (padding) yapıp **maskelemeyi unutmak**, modelin sıfırlarla dolu ayları gerçek gözlem sanmasına yol açar.

· **Ölçeklemenin tüm veride yapılması.** Standartlaştırma istatistikleri yalnız eğitim diliminden hesaplanmalıdır; aksi halde sessiz bir sızıntı olur.

07 Veri hazırlığı ve pencereleme

Sıralı modellerde sızıntı, tabular modellerdekinden daha kolay oluşur ve daha zor fark edilir. Pencere kurgusunu tek bir çizim üzerinde sabitlemek en güvenli yoldur.

Gözlem penceresi, boşluk ve performans penceresi



Bosluk, verinin uretimde ne zaman hazir oldugunu temsil eder — sifir boşluk genellikle yanlistir.

Şekil 6. Pencere kurgusu, skorun üretimde ne zaman üretilebileceğini yansıtmalıdır.

Konu	Yapılması gereken
Ölçekleme	Standartlaştırma veya robust ölçekleme zorunlu; istatistikler yalnız eğitim diliminden. Ağaç modellerinin aksine sinir ağı ölçeğe duyarlıdır.
Uzun kuyruklu tutarlar	Bakiye ve harcama gibi alanlarda log dönüşümü; aksi halde birkaç aykırı değer gradyanı domine eder.
Kategorik alanlar	Gömme (embedding) katmanı veya düşük kardinalitede one-hot. WOE de çalışır ve yorumlanabilirliği artırır.
Eksik dönem	Doldurma + maskeleme. Ayrıca "bu ay veri yok" bayrağını ayrı bir değişken olarak ver.
Değişken uzunluk	Baştan doldurma (pre-padding) tercih edilir; son adımlar dizinin gerçek sonu olur.
Bölme	Zaman bazlı. Aynı müşterinin farklı pencereleri hem eğitimde hem testte olmamalı.

Aynı müşteriden çok sayıda pencere

Kayan pencere ile bir müşteriden onlarca örnek üretmek veriyi büyütür ama **bağımsız gözlem sayısını artırmaz**. Bu pencerelerin bir kısmı eğitimde, bir kısmı testte kalırsa model aynı müşteriyi her iki tarafta görür ve performans iyimser çıkar. Bölmeyi her zaman **müşteri bazında** yap; kayan pencereyi yalnız eğitim tarafında kullan.

08 Mimari kararları

Bankacılık verisinde işe yarayan mimariler şaşırtıcı derecede küçüktür. Derinlik ve genişlik eklemek, veri miktarı sabitken neredeyse her zaman aşırı öğrenme demektir.

Karar	Tipik başlangıç	Not
Gizli birim sayısı	32–64	256 birim tabular bankacılık verisinde neredeyse her zaman fazladır.
Katman sayısı	1 (gerekirse 2)	İkinci katmanın kazancı çoğu zaman gürültü bandında kalır.
return_sequences	Son katmanda False	Dizi başına tek tahmin üretiyorsan yalnız son adım gerekir; katman üst üste bindirilecekse alt katmanda True.
Çift yönlü	Hayır (tahmin problemlerinde)	Sızıntı riski — 5. bölüm.
Dropout	0,1–0,3	Girdi ve çıktı bağlantılarında.
Recurrent dropout	0,0–0,2	Güçlü bir düzenlileştiricidir ama GPU hızlandırmasını kapatır; eğitim belirgin yavaşlar.
Çıkış katmanı	Dense(1, sigmoid)	İkili sınıflandırma için; çok adımlı tahminde Dense(ufuk).
Statik değişkenler	Ayrı girdi, LSTM çıktısıyla birleştir	Yaş, segment, ürün gibi zamana bağlı olmayan alanları her adıma tekrarlamak yerine sona ekle.

Parametre–gözlem dengesi

Mimariyi seçmeden önce iki sayıyı yan yana koy: modelin parametre sayısı ve elindeki **pozitif** gözlem sayısı. 20 bin parametreli bir modeli 3.000 bad gözlemle eğitmek, düzenlileştirme olmadan ezberden başka bir şey üretmez. Bankacılıkta pratik sıralama şudur: önce gizli birimi düşür, sonra dropout ekle, en son katman ekle — bu sıra tersine çevrildiğinde model hep aynı yere, aşırı öğrenmeye varır.

Hibrit mimari: dizi + tabular

Gerçek projelerde en çok işe yarayan kurgu genellikle saf LSTM değil, **hibrit** olanıdır: dizi verisi LSTM'den geçer, statik ve büro değişkenleri doğrudan yoğun (dense) katmana verilir, ikisi birleştirilip çıkış katmanına bağlanır. Bu yapı hem sıranın bilgisini kullanır hem de zaten güçlü olan tabular değişkenleri seyreltmeden korur.

09 Eğitim pratiği

LSTM eğitimi, gradient boosting eğitiminden farklı bir disiplin gerektirir: sonuç seed'e daha duyarlıdır, erken durdurma daha kritiktir ve öğrenme oranı en önemli tek düğmedir.

Konu	Öneri
Optimizer	Adam, öğrenme oranı 1e-3 ile başla; plato görürsen 3e-4'e düşür.
Kayıp	İkili sınıflandırmada binary crossentropy; nadir sınıfta sınıf ağırlığı veya focal loss.
Batch boyutu	256–1024. Büyük batch eğitimi hızlandırır, çok büyüğü genellemeyi bozabilir.
Gradyan kırpma	`clipnorm=1.0` — patlayan gradyana karşı ucuz sigorta.
Erken durdurma	Validation AUC üzerinden, sabır 5–10 epoch, en iyi ağırlıkları geri yükle.
Epoch sayısı	Üst sınır 50–100; gerçek durma noktasını erken durdurma belirler.
Öğrenme oranı planı	Plato görülünce yarıya indiren bir plan (ReduceLROnPlateau) genellikle yeterlidir.
Seed	Sonuç ağaç modellerine göre daha oynaktır; en az 5 seed ile medyan ± std raporla.

Nadir sınıfta iki seçenek

Sınıf ağırlığı basittir ve genellikle yeterlidir; fakat çıktının kalibrasyonunu bozar — skoru olasılık olarak kullanacaksan ayrı bir kalibrasyon katmanı gerekir. **Alt örnekleme** eğitimi hızlandırır ama diziler halinde çalışırken bilgi kaybettirir. Pratikte sınıf ağırlığı + sonradan kalibrasyon, kredi riski uygulamalarında en temiz yoldur.

Bir LSTM'in aşırı öğrendiği nasıl anlaşılır?

Eğitim kaybı düşerken validation kaybının yükselmeye başlaması klasik göstergedir; ancak nadir sınıfta bu grafik yanıltıcı olabilir. Daha güvenilir kontrol: **validation AUC** ve **üst dilim yakalama** eğrisini epoch bazında izlemek. AUC hala artarken üst dilim yakalamanın düşmeye başlaması, modelin ortalamayı iyileştirip riskli uçta bozulduğunu gösterir — kredi riskinde asıl önemsediğin bölge orasıdır.

10 Bankacılık analitiğinde kullanım

Bankada sıralı veri boldur: aylık hesap özetleri, kart işlemleri, kanal etkileşimleri, tahsilat temasları. Soru "nerede dizi var?" değil, **"sıranın kendisi bilgi taşıyor mu?"**dur.

Dizi uzunluğu ve düzenliliğine göre uygunluk haritası

Şekil 7. Temsili konumlandırma; kesin yer veri kalitesine ve dizi uzunluğuna göre değişir.

Kullanım alanı	Dizi neyi temsil eder?	LSTM buna değer mi?
Kart fraud tespiti	Ardışık işlemler: tutar, saat, MCC, kanal, coğrafya	**Evet.** Sıra ve zamanlama doğrudan sinyal; en güçlü kullanım alanı.
AML / şüpheli işlem	Hesap hareketleri, karşı taraflar, tutar deseni	**Evet**, ama açıklanabilirlik zorunluluğu mimariyi kısıtlar.
Davranışsal kredi riski	12–24 aylık hesap özeti	Kısmen. GBDT + değişim değişkeni çoğu zaman yakın sonuç verir.
Erken uyarı (EWS)	Bakiye, ödeme oranı, limit kullanımı dizisi	Kısmen. Bozulmanın **şekli** önemliyse LSTM katkı sağlar.
Churn / kullanım bırakma	Aylık işlem sayısı, kanal etkileşimi, temas kaydı	Evet — sessizleşme deseni sıralıdır.
Tahsilat aksiyon yanıtı	Temas–ödeme–söz dizisi	Evet, olay bazlı ve düzensiz olduğu için.
ATM / şube nakit talebi	Günlük çekim serisi, takvim etkileri	Evet, ama klasik zaman serisi yöntemleri güçlü rakiptir.
Başvuru skorlaması (FPD)	Dizi yok — tek anlık fotoğraf	**Hayır.** Sıralı model için uygun problem değil.

11 Üç kullanım alanı yakından

Aynı hücre, üç farklı problemde üç farklı veri kurgusuyla çalışır. Fark modelde değil, dizinin nasıl tanımlandığındadır.

1 · Kart fraud tespiti — LSTM'in en güçlü olduğu yer

Dizi, bir kartın son **N işlemi**dir ve zaman adımı sabit değildir: iki işlem arasında bir saat de olabilir bir hafta da. Model, tek başına olağan görünen bir işlemi **bağlamı içinde** değerlendirir — üç küçük test harcamasının ardından gelen yüksek tutarlı işlem, aynı tutarın tek başına taşıdığından çok farklı bir anlam taşır. Kritik değişkenler arasında işlemler arası geçen süre, tutarın müşterinin kendi geçmişine oranı ve MCC değişimi bulunur. Üretim tarafında zorluk gecikmedir: skor birkaç yüz milisaniyede üretilmelidir, bu da dizi uzunluğunu ve model boyutunu doğrudan sınırlar.

2 · Davranışsal kredi riski ve erken uyarı

Dizi, son 12–24 ayın **hesap özeti**dir: bakiye, asgari ödeme oranı, limit kullanımı, nakit avans payı, gecikme gün sayısı. Buradaki değer, bozulmanın **şeklini** yakalamaktır: altı ay boyunca kademeli tırmanan limit kullanımı ile tek ayda sıçrayan kullanım aynı ortalamayı verse de farklı risklerdir. Uyarı: bu problemde GBDT çok güçlü bir rakiptir ve çoğu portföyde iyi tasarlanmış eğim/değişim değişkenleriyle LSTM'e yakın sonuç verir. Karar, 15. bölümdeki protokolle ölçülmeden verilmemelidir.

3 · Nakit talebi ve operasyonel kapasite tahmini

ATM ve şube nakit talebi, çağrı merkezi hacmi, gişe yoğunluğu — hepsi düzenli aralıklı zaman serileridir. LSTM burada çoklu değişkeni (takvim, maaş günü, kampanya, hava durumu) tek modelde birleştirebilmesiyle değerlidir. Ancak dürüst olmak gerekirse mevsimsellik ve takvim etkisi baskın olduğunda klasik yöntemler (SARIMA, Prophet benzeri ayrıştırmalar, hatta iyi kurulmuş bir GBDT) çoğu zaman yeterlidir ve çok daha kolay savunulur. LSTM'i, çok sayıda ATM'yi **tek bir global modelde** öğrenmek istediğinde tercih et.

Üç örneğin ortak yanı, modelin değerinin doğruluğu kadar **zamanında üretilmesine** bağlı olmasıdır. Fraud skorunun işlem anında, erken uyarı skorunun ay kapanmadan, nakit tahmininin sevkiyat planından önce hazır olması gerekir. Bu yüzden sıralı bir modeli üretime alırken sorulacak ilk soru mimariyle ilgili değildir: dizinin son adımının ne kadar gecikmeyle hazır olduğu, modelin kullanılabilir olup olmadığını baştan belirler.

12 LSTM mi, GBDT mi?

Kredi riski ekiplerinde en sık sorulan soru budur ve cevabı çoğu portföyde beklenenden daha tutucudur: **diziyi düzgün özetleyebiliyorsan gradient boosting kazanır.**

Boyut	LSTM	GBDT + değişim değişkeni
Girdi	Ham dizi; elle değişken üretimi az	Elle üretilmiş özet ve eğim değişkenleri
Kısa düzenli dizide	Genellikle eşdeğer	Genellikle eşdeğer, çok daha ucuz
Uzun düzensiz dizide	Belirgin üstün	Özetleme bilgi kaybettirir
Kategorik zenginlik	Gömme katmanı gerekir	Native destek, WOE ile olgun pratik
Eğitim süresi	Saatler; GPU ile anlamlı hızlanma	Dakikalar; CPU yeterli
Seed duyarlılığı	Yüksek	Düşük
Yorumlanabilirlik	Zor; dikkat veya vekil model gerekir	TreeSHAP ile olgun
Model riski onayı	Ağır süreç	Yerleşik pratik
Bakım	Ortam, sürüm ve donanım bağımlılığı yüksek	Düşük

Karar kuralı

LSTM'i, GBDT baseline'ını **aynı bilgi setiyle** geçtiğini ölçtükten sonra seç. Fark, seed dağılımının genişliğinin yaklaşık iki katından küçükse, bakım maliyeti ve açıklanabilirlik gerekçesiyle GBDT'de kal. Farkın büyüdüğü tipik durum tek başına dizi uzunluğu değil, **dizinin düzensizliğidir**: olay bazlı, değişken aralıklı ve uzun akışlarda LSTM'in üstünlüğü gürültü bandını aşar.

Melez yol çoğu zaman en iyisi

Pratikte en verimli kurgu ikisini yarıştırmak değil, **birleştirmek**tir: LSTM'in ürettiği dizi temsilini (son gizli durum vektörünü) bir değişken bloğu olarak çıkar ve GBDT modeline diğer tabular değişkenlerle birlikte ver. Böylece sıranın bilgisi modele girer, karar mekanizması ise açıklanabilir kalır. Model riski tarafında savunması, uçtan uca bir sinir ağından belirgin biçimde kolaydır.

13 LSTM'siz alternatif

LSTM'e geçmeden önce yapılması gereken şey, diziyi elle özetlemenin sınırına gerçekten ulaşıp ulaşmadığını görmektir. Çoğu ekip bu sınıra hiç yaklaşmadan sinir ağına geçer.

Yakalanmak istenen	Elle üretilecek değişken
Seviye	Son değer, son 3/6/12 ay ortalaması, medyan
Eğim / yön	Son 3 ay ile önceki 3 ayın oranı; basit doğrusal eğim katsayısı
Oynaklık	Standart sapma, değişim katsayısı, maksimum–minimum farkı
Sıçrama	En büyük aylık artış, kendi ortalamasına göre z-skoru
Süreklilik	Ardışık artış/azalış ay sayısı, eşiğin üstünde geçirilen ay sayısı
Yakınlık	Son olaydan bu yana geçen süre (son gecikmeden bu yana kaç ay)
Şekil	Tırmanan mı, sıçrayan mı, dalgalı mı — basit desen bayrakları
Mevsimsellik	Aynı ayın geçen yılki değeriyle oranı

Bu sekiz başlık, kısa ve düzenli dizilerde LSTM'in çıkardığı bilginin büyük kısmını kapsar. Üstelik ürettiği değişkenler **iş tarafına anlatılabilir**: "son üç ayda limit kullanımı önceki üç aya göre %40 artmış" cümlesi, bir gizli durum vektörüyle kıyaslanamayacak kadar anlaşılırdır. Sıralı modelin gerçek katkısı, bu tür değişkenlerin **elle tasarlanamadığı** durumlarda ortaya çıkar: yüzlerce işlemlik düzensiz bir akışta hangi alt desenin önemli olduğunu önceden bilmek mümkün değildir. Listeyi tüketmeden LSTM'e geçmek, kolayca elde edilebilecek bir kazancı pahalı bir yoldan aramak anlamına gelir.

> **Ara bir yol: dizi temsilini öğrenmek**
>
> İki uç arasında bir seçenek daha var: diziyi denetimsiz bir modelle (otokodlayıcı) düşük boyutlu bir vektöre indirip bu vektörü tabular modele değişken olarak vermek. Etiket sayısı sınırlıyken bu yaklaşım, uçtan uca eğitilen bir LSTM'den daha az aşırı öğrenir ve dizi temsili birden fazla modelde yeniden kullanılabilir.

14 Python: davranışsal risk dizisi

12 aylık hesap özeti dizisi + statik değişkenler ile ikili sınıflandırma. Hibrit kurgu, 8. bölümde anlatılan yapıdır.

```
# ---------- VERI: (N, T, F) ----------
# X_seq : (n_musteri, 12, 18)  aylik hesap ozeti dizisi
# X_sta : (n_musteri, 25)      statik + buro degiskenleri
print(X_seq.shape, X_sta.shape)      # sekli HER ZAMAN dogrula

from sklearn.preprocessing import StandardScaler
sc = StandardScaler().fit(X_seq_tr.reshape(-1, X_seq_tr.shape[-1]))   # yalniz TRAIN
scale = lambda a: sc.transform(a.reshape(-1, a.shape[-1])).reshape(a.shape)
X_seq_tr, X_seq_va = scale(X_seq_tr), scale(X_seq_va)

# ---------- MODEL ----------
from tensorflow import keras
from tensorflow.keras import layers as L

seq_in = keras.Input(shape=(12, 18), name="dizi")
m      = L.Masking(mask_value=0.0)(seq_in)          # doldurulmus aylari maskele
m      = L.LSTM(48, dropout=0.2, unit_forget_bias=True)(m)

sta_in = keras.Input(shape=(25,), name="statik")
s      = L.Dense(32, activation="relu")(sta_in)

x   = L.Concatenate()([m, s])                       # hibrit birlesim
x   = L.Dropout(0.2)(x)
out = L.Dense(1, activation="sigmoid")(x)

model = keras.Model([seq_in, sta_in], out)
model.compile(
    optimizer=keras.optimizers.Adam(1e-3, clipnorm=1.0),
    loss="binary_crossentropy",
    metrics=[keras.metrics.AUC(name="auc"), keras.metrics.AUC(curve="PR", name="pr")]
)

cb = [
    keras.callbacks.EarlyStopping(monitor="val_auc", mode="max",
                                  patience=8, restore_best_weights=True),
    keras.callbacks.ReduceLROnPlateau(monitor="val_auc", mode="max",
                                      factor=0.5, patience=3)
]

model.fit([X_seq_tr, X_sta_tr], y_tr,
          validation_data=([X_seq_va, X_sta_va], y_va),
          epochs=60, batch_size=512,
          class_weight={0: 1.0, 1: neg/pos},        # cikti artik olasilik degil
          callbacks=cb, verbose=2)

# ---------- DIZI TEMSILINI CIKAR (12. bolum: melez yol) ----------
encoder = keras.Model(seq_in, m)
Z_tr = encoder.predict(X_seq_tr)      # (N, 48) -> GBDT'ye degisken blogu olarak ver
```

Koddaki dört bilinçli tercih

Ölçekleyici yalnız train'de fit ediliyor — dizilerde en sık görülen sessiz sızıntı budur.

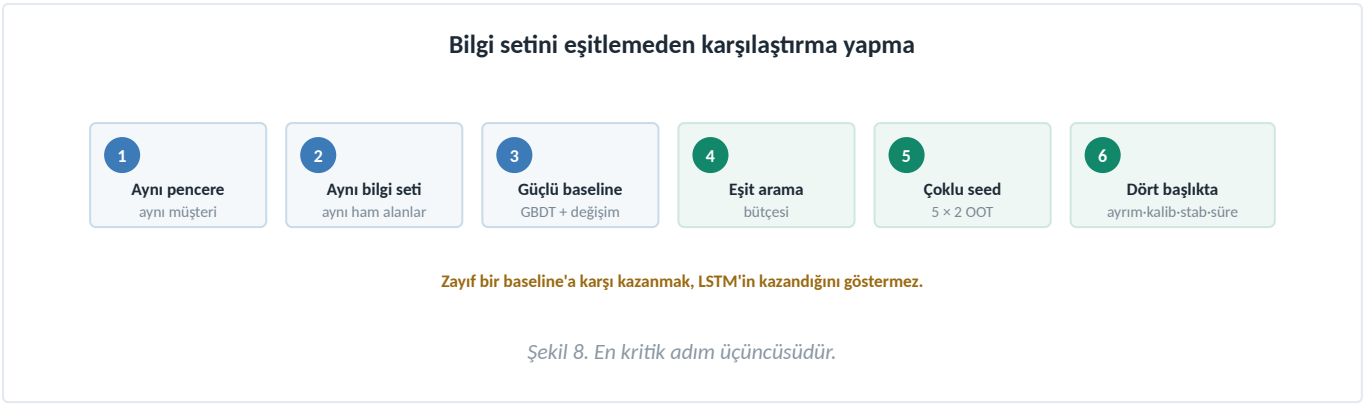
Masking katmanı var — doldurulmuş aylar gerçek gözlem sayılmıyor.

Erken durdurma AUC'yi izliyor, kaybı değil — nadir sınıfta kayıp eğrisi yanıltıcıdır.

Encoder ayrıca çıkarılıyor — aynı eğitimden hem uçtan uca model hem melez kurgu için değişken bloğu elde edilir.

15 Adil karşılaştırma protokolü

LSTM ile tabular baseline karşılaştırmalarının çoğu, model farkı değil **bilgi farkı** ölçer. Aşağıdaki sıra bunu engeller.



Şekil 8. En kritik adım üçüncüsüdür.

1. Aynı pencere ve aynı popülasyon. Aynı gözlem penceresi, aynı boşluk, aynı müşteri bazlı bölme.

2. Aynı ham bilgi seti. LSTM'e 24 ay verip GBDT'ye 3 aylık özet vermek karşılaştırma değildir. İki modele de aynı ham alanlar sunulmalı; fark yalnız bu alanların nasıl temsil edildiğinde olmalıdır.

3. Baseline gerçekten güçlü olmalı. Karşına koyacağın GBDT, 13. bölümdeki değişim değişkenleriyle beslenmiş ve kendi aramasıyla ayarlanmış olmalı. Literatürdeki "LSTM tabular modeli geçti" sonuçlarının önemli bir kısmı, zayıf bir baseline'a karşı alınmıştır.

4. Eşit arama bütçesi. Aynı sayıda deneme; sinir ağına 200, GBDT'ye 20 deneme vermek ayar farkı ölçer.

5. Çoklu seed ve çoklu OOT. LSTM'in seed varyansı ağaç modellerinden yüksektir; en az 5 seed zorunludur.

6. Dört başlıkta raporla: ayrım, kalibrasyon, stabilite, süre. Skorlama gecikmesini de ölç.

16 Model riski ve yorumlanabilirlik

Bankada bir modelin hayatı, geliştirme bittiğinde başlar. Sıralı modellerde doğrulama ve izleme, tabular modellere göre belirgin biçimde daha ağırdır.

Boyut	LSTM'de durum	Ne yapmalı?
Değişken önemi	Doğrudan karşılığı yok	Permutation importance (değişken veya zaman adımı bazında), gradyan tabanlı atıf
Zaman adımı katkısı	Gizli	Dikkat katmanı ekle veya adım bazında maskeleyip performans düşüşünü ölç
Monotonluk kısıtı	Yok	Kritik değişkenler için kısıt gerekiyorsa mimari tek başına yetmez; melez kurgu düşün
Tekrarlanabilirlik	Zor: seed, donanım, kütüphane sürümü, GPU determinizmi	Ortamı konteynerle sabitle; deterministik modu aç ve test et
Kalibrasyon	Sınıf ağırlığı ve dropout ile bozulur	Ayrı dilimde kalibrasyon katmanı, OOT'de doğrulama
İzleme	PSI skor üzerinde yeterli değil	Girdi dizilerinin uzunluk ve eksiklik dağılımını da izle

Vekil model: denetim için pratik bir köprü

Uçtan uca bir LSTM'i savunmanın en pratik yollarından biri, skorlarını hedef alan **yorumlanabilir bir vekil model** (sığ ağaç veya lojistik) eğitmektir. Vekilin uyum derecesi ve ürettiği kurallar, "bu model hangi durumlarda yüksek skor veriyor?" sorusuna somut cevap üretir. Vekil modeli asıl model yerine kullanma; yalnız açıklama ve tutarlılık kontrolü aracıdır.

Üretim maliyeti gerçekçi hesaplanmalı

LSTM'i üretime almak, bir GBDT modelini almaktan yapısal olarak farklıdır: dizi verisinin skorlama anında hazır edilmesi gerekir, bu da genellikle yeni bir veri boru hattı demektir. Toplam maliyetin büyük kısmı modelde değil **bu hattın kurulması ve izlenmesindedir**. Karar verirken kazanılan Gini puanının yanına bu maliyeti de yaz.

17 Yaygın tuzaklar

Modeli ya da karşılaştırmayı sessizce bozan hatalar.

Tuzak	Ne olur?	Doğrusu
Eksenleri karıştırmak (N, F, T)	Kütüphane hata vermez; model anlamsız bir yapı öğrenir.	Şekli yazdır, bir müşterinin dizisini elle doğrula.
Ölçekleyiciyi tüm veride fit etmek	Sessiz sızıntı; validation iyimser çıkar.	Yalnız eğitim diliminde fit et.
Doldurup maskelememek	Sıfırlar gerçek gözlem sayılır; kısa dizili müşteriler bozulur.	Masking katmanı + "veri yok" bayrağı.
Tahmin probleminde çift yönlü katman	Gelecek bilgisi sızar.	Tek yönlü kullan.
Aynı müşterinin pencerelerini bölmek	Model aynı müşteriyi iki tarafta görür.	Müşteri bazlı bölme.
Zayıf baseline ile karşılaştırmak	LSTM kazanmış gibi görünür.	GBDT'yi değişim değişkenleriyle güçlendir.
Tek seed ile karar vermek	LSTM'de seed varyansı yüksektir; gürültü bulgu sanılır.	En az 5 seed, medyan ± std.
Sınıf ağırlığı sonrası çıktıyı olasılık sanmak	Skor sistematik olarak kayar.	Ayrı dilimde kalibrasyon.
Çok büyük mimariyle başlamak	Az veride hemen ezberler; teşhis zorlaşır.	32–64 birim, tek katmanla başla.
Skorlama gecikmesini hesaba katmamak	Model üretimde yetişemez.	Dizi hazır olma süresini baştan ölç.

Hepsinin ortak paydası

Listedeki hataların çoğu modelde değil, **verinin şekillendirilmesinde**. Sıralı modellerde zamanın büyük kısmı burada geçer ve hataların büyük kısmı da buradan çıkar.

Hızlı referans kartı

Tek sayfada LSTM.

Ne çözer	Temel RNN'de kaybolan gradyan; uzun mesafeli bağımlılık
Anahtar fikir	Hücre durumu bilgiyi çarparak değil toplayarak taşır
Kapılar	Unut (ne kalsın) · Gir (ne yazılsın) · Çıkar (ne gösterilsin)
İki hafıza	c(t) uzun vadeli defter · h(t) o adımın çıktısı
Girdi şekli	(örnek, zaman adımı, değişken) — üç boyutlu
Tipik mimari	1 katman · 32–64 birim · dropout 0,1–0,3
Optimizer	Adam 1e-3 · clipnorm 1.0 · erken durdurma AUC üzerinden
Parametre sayısı	4 × ((girdi + gizli) × gizli + gizli)
İlk alternatif	GRU — daha az parametre, kısa dizilerde çoğu zaman eşdeğer
Bankada en güçlü olduğu yer	Kart fraud, AML, churn, tahsilat yanıtı — uzun ve düzensiz akışlar
Bankada gereksiz olduğu yer	Başvuru skorlaması; kısa düzenli hesap özeti dizileri
En güçlü rakip	GBDT + değişim/eğim değişkenleri
En sık hata	Eksen karışması ve ölçekleyicinin tüm veride fit edilmesi
Model riski yükü	Yüksek: yorumlanabilirlik, tekrarlanabilirlik, veri hattı

Tek cümlelik özet

LSTM, **sıranın kendisinin bilgi taşıdığı** problemlerde satın aldığın araçtır. Bankacılık verisinin büyük kısmında bu koşul sağlanmaz ve iyi tasarlanmış değişim değişkenleriyle beslenen bir gradient boosting modeli hem daha ucuz hem daha savunulabilir sonuç verir. LSTM'i, bu alternatifin sınırına gerçekten ulaştığını ölçtükten sonra devreye al.

18 Kaynaklar

Yöntemi tanıtan özgün makaleler ve pratik başvuru kaynakları.

1 **Hochreiter & Schmidhuber — Long Short-Term Memory (Neural Computation, 1997)**
LSTM'in tanıtıldığı özgün makale

2 **Gers, Schmidhuber & Cummins — Learning to Forget (Neural Computation, 2000)**
Unutma kapısının eklendiği çalışma

3 **Cho et al. — Learning Phrase Representations using RNN Encoder-Decoder (2014)**
GRU'nun tanıtıldığı makale

4 **Greff et al. — LSTM: A Search Space Odyssey (IEEE TNNLS, 2017)**
Hücre varyantlarının sistematik karşılaştırması

5 **Pascanu, Mikolov & Bengio — On the difficulty of training RNNs (ICML 2013)**
Kaybolan ve patlayan gradyan; gradyan kırpma

6 **Grinsztajn et al. — Why do tree-based models still outperform deep learning on tabular data? (NeurIPS 2022)**
12. bölümdeki tutucu çerçevenin dayanağı

7 **Keras Documentation — LSTM, Masking, Functional API**
keras.io

8 **BCDataWorks Çalışma Notları № 04 ve № 05**
Ağaç toplulukları ve first payment default modellemesi

Buradaki değerlendirmelerin tamamı **tabular ve zaman serisi bankacılık verisi** için geçerlidir. Doğal dil, ses ve görüntü uygulamalarında LSTM'in konumu farklıdır; bu alanlarda dikkat tabanlı mimariler büyük ölçüde öne geçmiştir. Dokümanın sıralı modellere temkinli yaklaşması, veri miktarının sınırlı ve açıklanabilirlik beklentisinin yüksek olduğu bir ortamı esas almasından kaynaklanıyor.

> **Son söz**
>
> LSTM güçlü bir araçtır, fakat bir bankacılık projesinde başarıyı belirleyen sıralama değişmez: hedef tanımı, veri kurgusu, validasyon tasarımı ve izleme. Sıralı model bu sıralamanın üstüne çıkmaz — yalnız doğru problemde, doğru kurulmuş bir veri hattının üzerine oturduğunda değer üretir.

BCDataWorks Çalışma Notları № 06 · Ağustos 2026

— ÇALIŞMA SONU —